Journal of *Imaging*

MDPI

Article

Three-Blind Validation Strategy of Deep Learning Models for Image Segmentation

Andrés Larroza [1], Francisco Javier Pérez-Benito [1], Raquel Tendero [1], Juan Carlos Perez-Cortes [1], Marta Román [2] and Rafael Llobet [1,*]

[1] Instituto Tecnológico de la Informática, Universitat Politècnica de València, Camino de Vera, s/n, 46022 València, Spain; alarroza@iti.es (A.L.); fjperez@iti.es (F.J.P.-B.); rtendero@iti.es (R.T.); jcperez@iti.es (J.C.P.-C.)

[2] Department of Epidemiology and Evaluation, IMIM (Hospital del Mar Research Institute), Passeig Marítim 25-29, 08003 Barcelona, Spain; mroman@hmar.cat

* Correspondence: rllobet@iti.es

Abstract: Image segmentation plays a central role in computer vision applications such as medical imaging, industrial inspection, and environmental monitoring. However, evaluating segmentation performance can be particularly challenging when ground truth is not clearly defined, as is often the case in tasks involving subjective interpretation. These challenges are amplified by inter- and intra-observer variability, which complicates the use of human annotations as a reliable reference. To address this, we propose a novel validation framework—referred to as the three-blind validation strategy—that enables rigorous assessment of segmentation models in contexts where subjectivity and label variability are significant. The core idea is to have a third independent expert, blind to the labeler identities, assess a shuffled set of segmentations produced by multiple human annotators and/or automated models. This allows for the unbiased evaluation of model performance and helps uncover patterns of disagreement that may indicate systematic issues with either human or machine annotations. The primary objective of this study is to introduce and demonstrate this validation strategy as a generalizable framework for robust model evaluation in subjective segmentation tasks. We illustrate its practical implementation in a mammography use case involving dense tissue segmentation while emphasizing its potential applicability to a broad range of segmentation scenarios.

Keywords: deep learning; image segmentation; mammography

Academic Editors: Vasileios Magoulianitis, Pawan Jogi and Spyridon Thermos

Received: 27 March 2025
Revised: 7 May 2025
Accepted: 19 May 2025
Published: 21 May 2025

Citation: Larroza, A.; Pérez-Benito, F.J.; Tendero, R.; Perez-Cortes, J.C.; Román, M.; Llobet, R. Three-Blind Validation Strategy of Deep Learning Models for Image Segmentation. *J. Imaging* **2025**, *11*, 170. https://doi.org/10.3390/jimaging11050170

1. Introduction

Image segmentation, the process of partitioning an image into meaningful regions, is a critical task in computer vision. By identifying and delineating objects or regions of interest, segmentation serves as a foundational step in numerous applications. In industrial settings, it is used for quality control, defect detection, and autonomous navigation, enabling robots to identify and manipulate objects on assembly lines [1]. In the medical domain, segmentation aids in the analysis of anatomical structures, the identification of pathologies, and treatment planning, such as delineating tumors for radiotherapy [2]. Environmental applications, including land-use mapping and disaster management, also rely heavily on accurate segmentation [3]. These examples highlight the importance of segmentation across diverse domains, underscoring its role in advancing technological and societal goals.

Despite its utility, image segmentation faces significant challenges. Inter- and intra-observer variability pose problems, especially in domains requiring subjective interpre-

tation, such as medical imaging. In radiology, differences in training, experience, and judgment can lead to inconsistent annotations, affecting diagnoses and treatment plans [4]. Similarly, in industrial applications, variability in manual quality assessments can result in inefficiencies or the misclassification of defects [5]. This variability underscores the need for standardized methodologies and robust automated systems to enhance reliability in segmentation tasks across all application areas.

Addressing these challenges requires the extensive validation of segmentation algorithms. Deep learning models, particularly those designed for segmentation, need rigorous evaluation to ensure their robustness and generalizability across diverse datasets [6]. A comprehensive validation framework includes performance metrics, such as precision and recall, and assessments of the model's ability to handle variations in data quality, resolution, and noise [7]. These systematic validation efforts are crucial to bridging the gap between algorithmic performance in controlled experimental settings and real-world deployment, ultimately improving the trust in and adoption of these models.

A specific example of these challenges can be found in mammography, where the accurate assessment of breast dense tissue is critical. Breast density, assessed from digital mammograms, is a known biomarker related to a higher risk of developing breast cancer. Its precise quantification is essential for improving cancer detection and screening effectiveness. Therefore, precise segmentation of dense tissue in mammograms is crucial for enhancing diagnostic accuracy. Recent advancements in segmentation techniques, particularly those leveraging deep learning, have demonstrated enhanced capabilities in distinguishing between dense and non-dense tissues, thereby facilitating more accurate breast cancer assessments [8].

Building on these advancements, this study introduces a three-blind validation strategy for deep learning-based segmentation. While dense tissue segmentation in mammography serves as a use case, the proposed approach is applicable to various segmentation tasks across different domains. A key challenge in evaluating deep learning-based segmentation models is the inherent subjectivity of the task, leading to intra- and inter-observer variability. Since no perfectly defined ground truth exists, discrepancies between the model and human experts cannot always be attributed to errors in one or the other. This validation strategy provides a structured way to assess agreement between labelers and the model, offering insights that can support the refinement of human annotations and the improvement of automatic segmentation models.

2. Materials and Methods

2.1. Validation Strategy

Ensuring reliable, high-quality segmentations is essential for developing and evaluating deep learning models for image segmentation. To address this, we propose a three-blind validation strategy, as illustrated in Figure 1. This approach compares the performance of deep learning models with human specialists by anonymizing annotation sources, ensuring unbiased evaluation by an expert validator. A predefined set of images is annotated by multiple human labelers and deep learning models. The annotations are then shuffled to remove identifying information about their origin. An independent validator reviews these shuffled annotations, providing an impartial assessment of their quality.

This strategy enables the analysis of inter-observer variability by comparing annotations from different human labelers and the model. Additionally, intra-observer variability can be assessed by having the validator unknowingly review a subset of segmentations twice at random. By ensuring that these repeated segmentations are presented to the validator without their awareness of the repetition, this approach provides assessment of the validator's consistency.

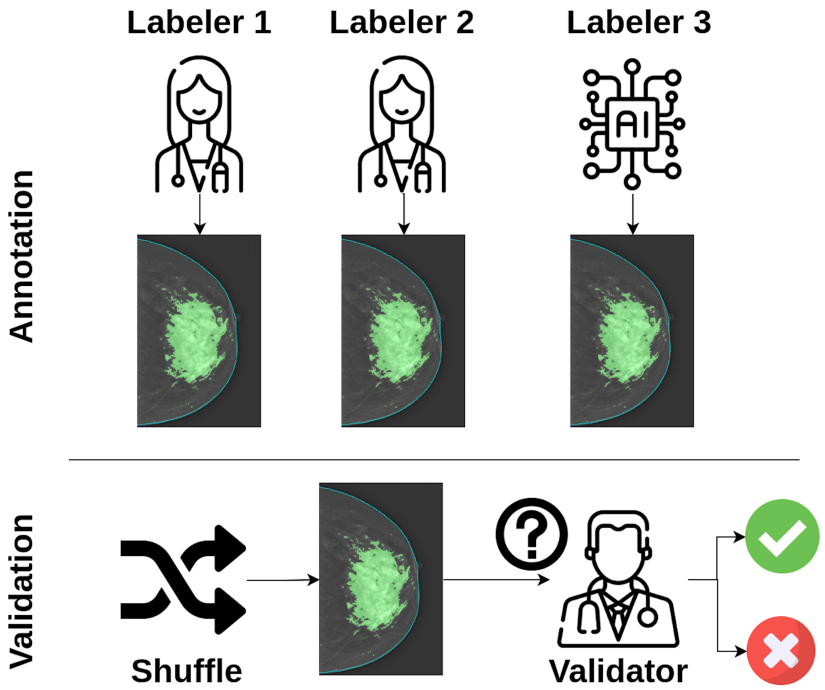


Figure 1. Workflow of the three-blind validation strategy for image segmentation. Multiple human labelers and deep learning models annotate a defined number of images. The annotations are then shuffled to anonymize their source, and an independent validator reviews them to ensure an impartial assessment of their quality. While the figure illustrates two human labelers and one deep learning model, the strategy can be extended to any number of human or automatic annotators. This figure has been designed using resources from flaticon.com (accessed on 13 December 2024).

2.2. Validation Tool

A custom interactive tool was developed to support the three-blind validation strategy by enabling a third specialist (validator) to independently assess segmentation masks. Built using the Python-based Streamlit library [9], the tool provides a user-friendly interface where the validator is presented with one segmentation mask at a time, without any indication of its origin (human or model). The tool randomly determines the order of presentation for segmentations from different sources, ensuring blinding.

For each displayed segmentation, the validator selects one of four predefined categories: correct, oversegmented, undersegmented, or incorrect. The tool also includes a comment box for optional qualitative feedback. This structure supports the standardized quantitative and qualitative evaluation of segmentations while preserving the independence of the assessment.

Figure 2 shows a screenshot of the interface used in our breast dense tissue segmentation use case. In this version, the interface was tailored to display digital mammograms and associated masks. A generalized version of the tool has also been developed and made publicly available via GitLab (https://egitlab.iti.es/praia-salud/segmentation-validation-tool.git, (accessed on 1 May 2025)). This version is designed to be compatible with any segmentation task, as it can be easily configured for different label categories and image modalities.

2.3. Use Case: Breast Dense Tissue Segmentation

The objective of this use case is to exhaustively validate a model for segmenting dense tissue in digital mammograms. The main challenges include variations in images from different acquisition devices and inter- and intra-reader variability [10]. Variability in human annotations is a critical consideration in this use case. Inter-observer variability

refers to differences in annotations between different radiologists, while intra-observer variability reflects the consistency of annotations made by the same radiologist at different times. These sources of variability complicate the evaluation of model performance, as discrepancies may arise not only from model errors but also from human subjectivity. As such, our validation strategy incorporates analyses to assess both inter- and intra-observer agreement as described in the analyses presented below.

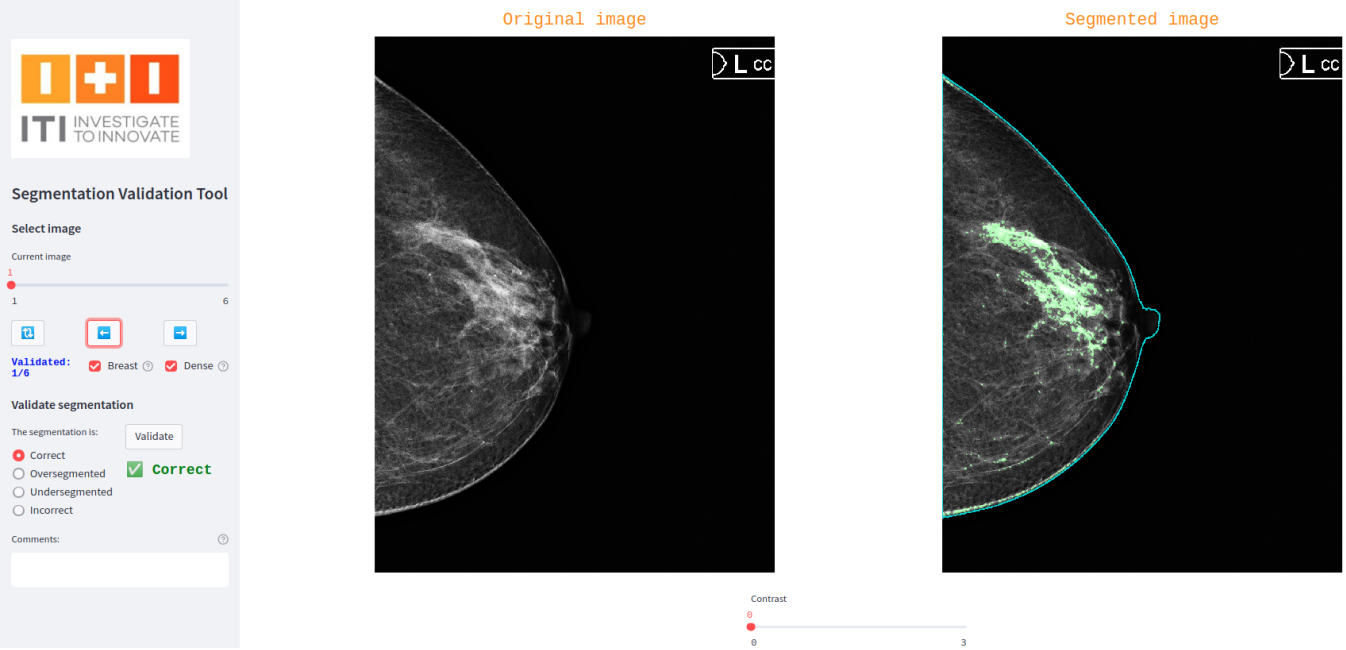


Figure 2. Screenshot of the tool used for three-blind validation in the breast dense tissue segmentation use case.

2.3.1. Dataset

We utilized a dataset comprising 500 studies obtained from the Hospital del Mar Research Institute (IMIM). This dataset was exclusively extracted for the three-blind validation presented here. It consists of mammograms from four different acquisition devices, collected over a period of 10 years (2011–2021). To simplify the procedure, only craniocaudal (CC) views were used. All the mammograms are of the type *for presentation*. Figure 3 illustrates the distribution of mammograms by year and acquisition device, highlighting a well-balanced representation across five distinct devices over the covered period. This diverse composition emphasizes the dataset's robustness and suitability for validating segmentation performance.

2.3.2. Deep Learning Model

The CM-YNet is a deep learning model developed by our group that automatically segments the dense tissue in digital mammograms. For full architectural and training details of CM-YNet, including the model's design rationale and evaluation against expert annotations, readers are referred to Larroza et al. [8]. Our previous results indicate that the CM-YNet model performs well, achieving a Dice Similarity Coefficient (DSC) comparable to that obtained between two specialists. This suggests that radiologists tend to agree more with the CM-YNet segmentation than with each other. Table 1 and Figure 4 show the DSC values obtained for the 500 validation images compared with the expert labelers. Given the well-known variability among expert readers, the next step is to verify that a third specialist agrees with CM-YNet as much as with the other two specialists, introducing the concept of three-blind validation strategy outlined in Section 2.1.

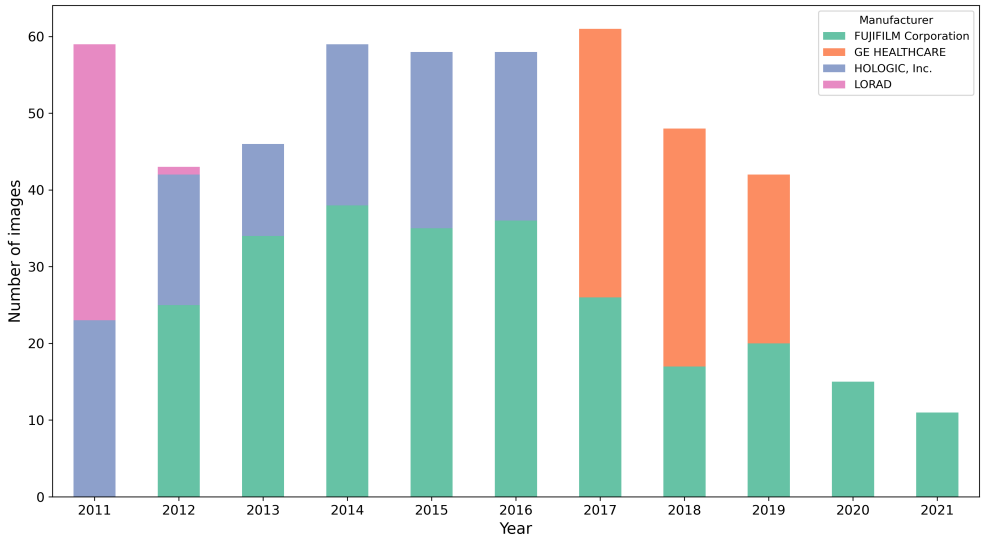


Figure 3. Distribution of the 500 mammograms according the year and acquisition device.

Table 1. DSC values obtained by the CM-YNet model compared against the expert labelers on the 500 validation images.

L1 vs. L2	L1 vs. CM-YNet	L2 vs. CM-YNet	Closest vs. CM-YNet
0.790 ± 0.160	0.710 ± 0.187	0.743 ± 0.162	0.773 ± 0.157

The Closest vs. CM-YNet Dice Similarity Coefficient (DSC) is computed by selecting, for each individual image, the higher DSC value between L1 vs. CM-YNet and L2 vs. CM-YNet. The reported value is the average of these per-image maxima across the 500 validation images.

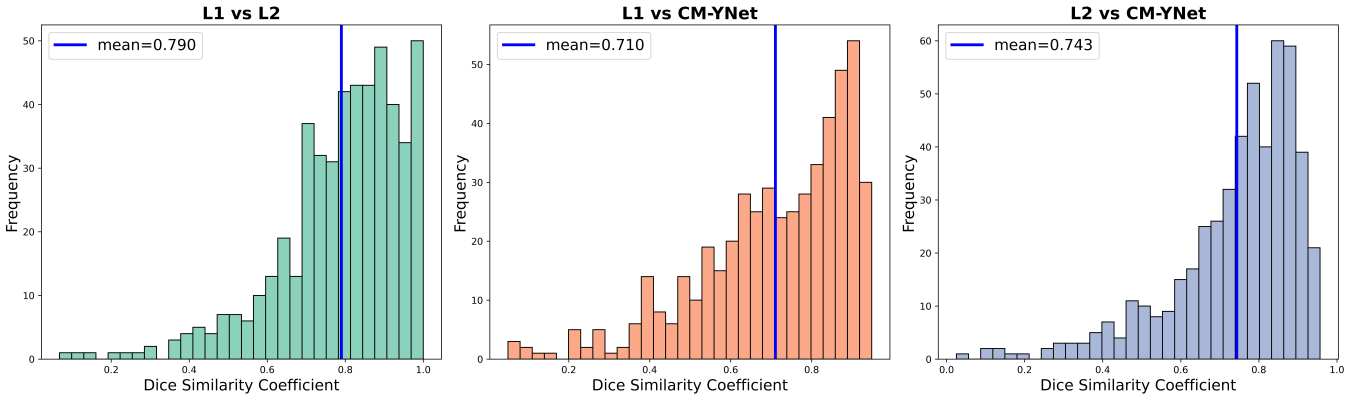


Figure 4. Distribution of DSC values between the labelers and the output generated by the CM-YNet model on 500 validation craniocaudal (CC) images.

2.3.3. Data Annotation

Data annotation was performed with an in-house developed tool named Futura Breast, which can be used to interactively segment the dense tissue using two parameters: the brightness corrector α and the fibroglandular tissue threshold th_F. These parameters guide the segmentation process as described in detail in Larroza et al. [8]. Two expert radiologists (L1 and L2) were instructed to perform data annotation with this tool. In total, each expert independently annotated 500 validation CC images. The same 500 images were also segmented by the automatic CM-YNet model. Therefore, we obtained a total of 1500 segmentations that were used for the three-blind validation. Additionally, 300 randomly selected images (100 from L1, 100 from L2, and 100 from CM-YNet) were included twice. This random sample allowed for the calculation of intra-observer variability for

the specialist conducting the validation. In total, the third specialist (validator) randomly validated 1800 segmentations.

The use of two human annotators was a pragmatic choice, balancing the need for reliable expert annotations with the high cost and time demands of manual labeling in clinical imaging. Recruiting two board-certified radiologists was considered the minimum acceptable setup to ensure diversity in expert opinion while maintaining feasibility within project constraints.

We conducted the three-blind validation twice using two independent radiologists with experience in breast imaging, referred to as Validator 1 (V1) and Validator 2 (V2). These validators were not involved in the initial annotations and had no prior knowledge of the segmentation sources (L1, L2, or CM-YNet). The first validation was performed by V1, and the second validation was carried out by V2. For the second validation, the breast delineation method was improved using the approach presented in [11]. Additionally, based on lessons learned from the first validation, V2 received refined instructions on the label definitions to improve consistency and reduce potential discrepancies. The specific results and observations from both validation rounds will be described in the Results Section.

2.3.4. Evaluation Metrics

To assess the validation results, we employed a range of commonly used evaluation metrics. We provide brief definitions and contexts for each metric used.

- Dice Similarity Coefficient (DSC): The DSC is a spatial overlap index widely used to evaluate segmentation tasks. It quantifies the similarity between two sets by computing the overlap relative to their combined size. It can take values ranging from 0 to 1, with a higher value indicating a higher similarity [12].
- Accuracy: Accuracy is defined as the proportion of correctly classified instances among the total number of instances. While it provides a general sense of performance, it can be misleading in imbalanced datasets [13].
- Cohen's Kappa: Cohen's Kappa measures the agreement between two raters while accounting for agreement occurring by chance. It is especially useful when comparing annotations from different sources or observers [14].
- Balanced Accuracy: Balanced Accuracy accounts for imbalanced class distributions by averaging the recall obtained on each class. It is computed as the average of sensitivity (recall) and specificity [15].
- F1 Score: The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both. It is particularly useful when the dataset has class imbalance and when false positives and false negatives carry different costs [13].
- Precision: Precision measures the proportion of true positive predictions among all positive predictions, reflecting the model's ability to avoid false positives [13].
- Recall: Recall, also known as sensitivity, measures the proportion of true positives among all actual positives, capturing the model's ability to detect relevant instances [13].

3. Results

3.1. First Validation

The validation tool provided four labels for assessing segmentations: correct, incorrect, oversegmented, and undersegmented. The incorrect label was originally intended for cases where the segmentation was entirely wrong and could not be classified as either oversegmented or undersegmented. However, after reviewing the 1800 segmentations, validator V1 reported that he used the incorrect label only rarely and did not consistently apply a

clear criterion for distinguishing between oversegmented, undersegmented, and incorrect. For example, some cases involving the segmentation of the pectoral muscle as dense tissue were labeled interchangeably as either oversegmented or incorrect. Consequently, to evaluate the results of V1, we decided to simplify the classification by grouping the labels into two broader categories: agreement and disagreement. This issue was resolved for the second validation, for which a clearer validation criterion was established in advance.

3.1.1. Agreement with Each Labeler

Figure 5 presents the agreement percentages for each evaluated labeler (L1, L2, and CM-YNet). The agreement with manual segmentations (L1 and L2) is higher than that with the automatic segmentation (CM-YNet). These percentages are based on a total of 1500 segmentations.

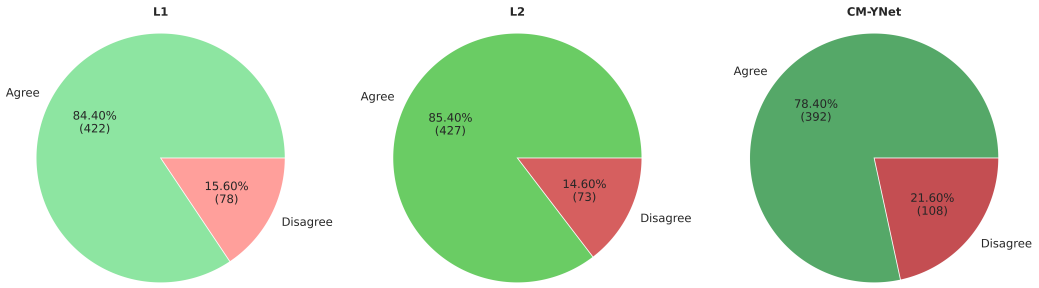


Figure 5. Agreement percentages between V1 and each evaluated labeler for a total of 1500 segmentations.

Subsequently, we analyzed the DSC between labelers, based on V1's agreement or disagreement with each segmentation presented. For each validated image, we indicate whether the validator's agreement or disagreement with two of the labelers was the same or not, as illustrated in Figure 6.

The DSC values are presented in Table 2. It is noteworthy that the DSC is higher in cases where V1 had the same label on two given segmentations, which corresponds to the highest percentage of images across all cases.

3.1.2. Intra-Observer Variability

To analyze intra-observer variability, the 300 segmentations that were randomly presented to V1 twice were used. Figures 7 and 8 illustrate the confusion matrices for V1's first and second evaluations of these segmentations. The results indicate that, in most cases, V1 demonstrated consistency in his decisions. Table 3 summarizes the corresponding metrics.

Table 2. DSC values for the different labelers (L1, L2, and CM-YNet), based on the of labels assigned by V1 to the evaluated segmentations.

Labelers	V1 Label Is the Same	DSC	95% CI
L1 vs. L2	No (55)	0.636 ± 0.215	$(0.578, 0.694)$
	Yes (445)	0.809 ± 0.141	$(0.796, 0.822)$
L1 vs. CM-YNET	No (78)	0.614 ± 0.225	$(0.563, 0.664)$
	Yes (422)	0.728 ± 0.174	$(0.712, 0.745)$
L2 vs. CM-YNET	No (73)	0.648 ± 0.202	$(0.601, 0.695)$
	Yes (427)	0.760 ± 0.149	$(0.746, 0.774)$

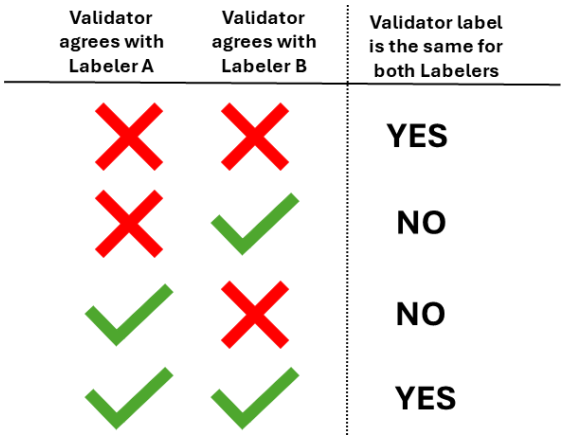


Figure 6. Given two labelers, which in our case can be any combination of L1, L2, or CM-YNet, we indicate if the validator marked the same label (agreement or disagreement) for both labelers.

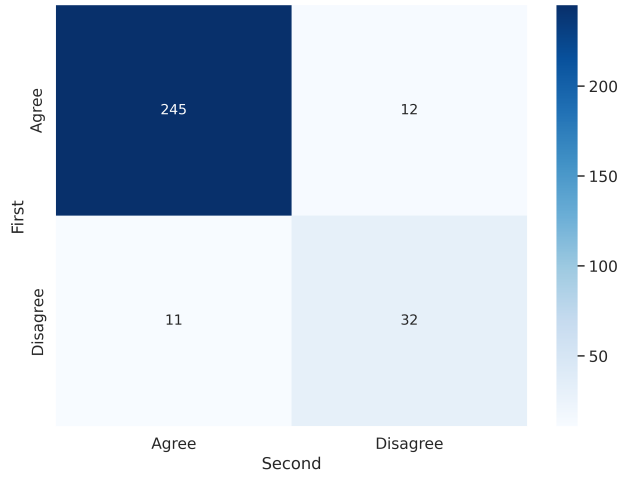


Figure 7. Confusion matrix for the 300 segmentations shown twice to V1. V1 exhibited inconsistency in 23 out of 300 cases (7.67%).

Table 3. Metrics derived from the confusion matrix based on the 300 images reviewed twice by V1.

Accuracy	Acc. 95% CI−	Acc. 95% CI+	Kappa	Balanced Accuracy	F1	Precision	Recall
0.923	0.888	0.948	0.691	0.849	0.955	0.957	0.953

Figure 9 presents examples of images segmented by CM-YNet where V1 disagreed with the automatic segmentation in both evaluations of the same segmentation. This disagreement was observed in 16 out of 300 images, as depicted in Figure 8. Notably, these images were acquired using older devices (HOLOGIC and LORAD in Figure 3), which could explain the increased difficulty in accurately segmenting these lower-quality images, even for expert annotators.

Further analysis of the intra-observer confusion matrices (Figure 5), which reflect the consistency of each validator's repeated assessments using binary labels (Agree vs. Disagree), reveals that the segmentations originally labeled by L1 exhibit the lowest intra-observer agreement. This indicates that the validator (V1) was less consistent when evaluating L1's segmentations, potentially due to greater variability or ambiguity in those masks. In contrast, CM-YNet's segmentations, although not always labeled as correct, were assessed more consistently by V1 across repeated validations. These results suggest that consistency in evaluation may not always align with accuracy and that model-generated

masks can elicit more reproducible judgments even if they are more frequently judged as incorrect.

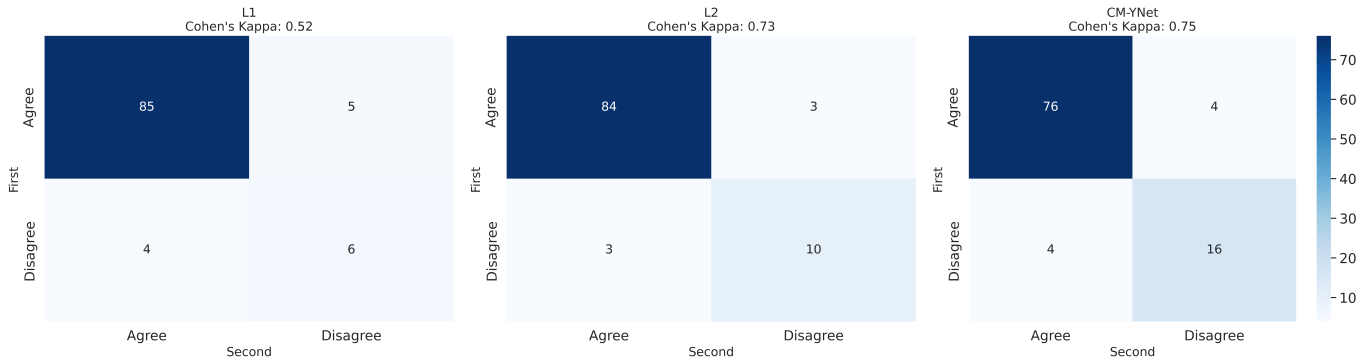


Figure 8. Confusion matrices per labeler for the 300 segmentations shown twice to V1.

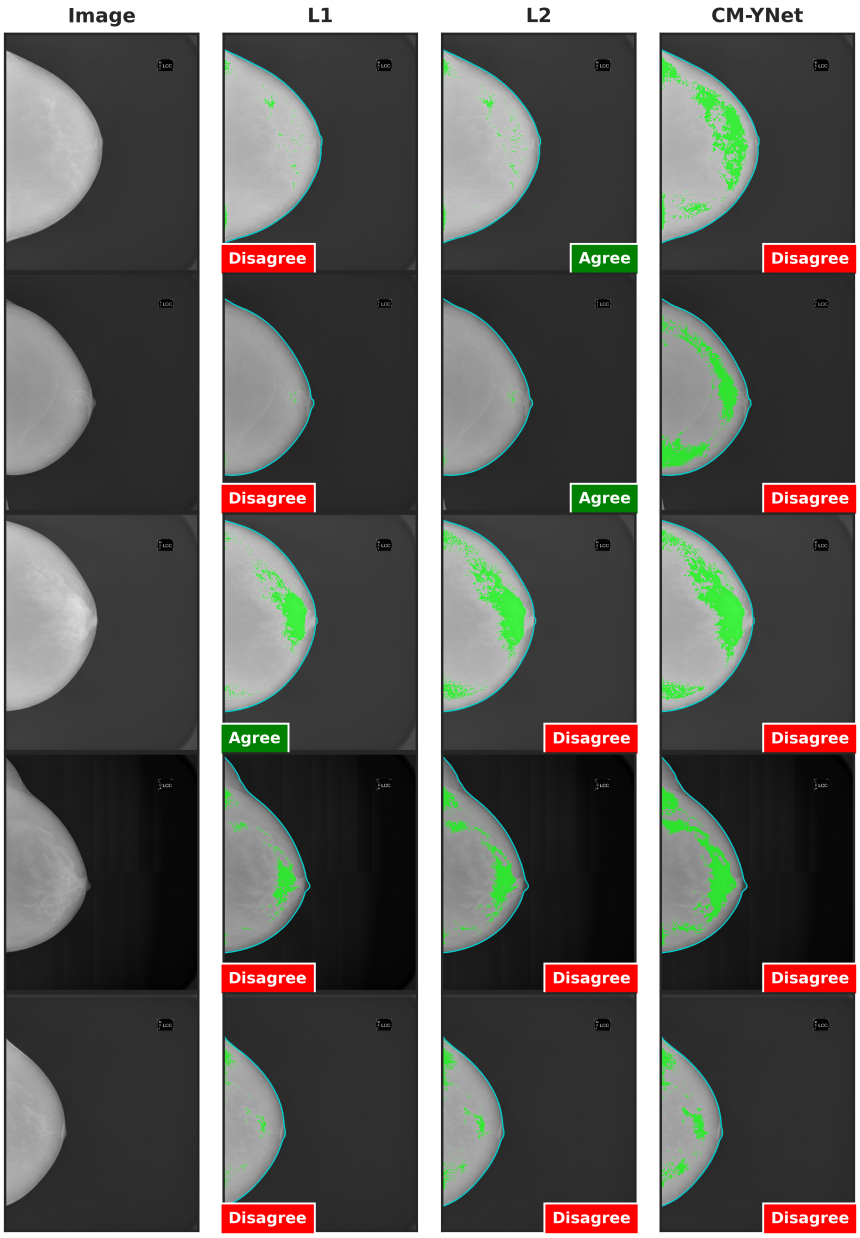


Figure 9. Examples of segmentations that were shown twice to V1 to analyze the intra-observer variability. In these examples, V1 indicated disagreement with CM-YNet on both occasions.

3.1.3. Exploring the Causes of Disagreement

For a given image, three segmentations were available: one each by L1, L2, and CM-YNet. These were presented to V1 in random order. Thus, the order of appearance (first, second, or third) refers to whether the segmentation from a particular labeler was the first, second, or third version of the same image seen by V1. This order may influence the validator's judgment, as familiarity with the image increases with repeated exposure.

Figure 10 shows two examples where V1's decisions changed depending on the order in which the segmentations were presented to them during the validation process, even though the segmentations from the three labelers were visually similar. Importantly, the order discussed here refers to the sequence in which the segmentations appeared to V1, not the fixed column order used in the figure layout. In the first-row example, the segmentation by L1 was shown first to V1, who marked it as agreement; however, V1 later marked disagreement for the same image when it reappeared with segmentations by L2 and CM-YNet. In the second-row example, V1 initially marked disagreement for the first two segmentations they saw (L1 and CM-YNet) but later marked agreement for the third (L2).

These observations were corroborated by V1, who was consulted about these specific examples without indicating him the order of appearance:

- Example 1: "In the inner quadrants of the three images something that is not dense tissue is segmented, so they would be *oversegmented*, but they also do not include all the glandular tissue of the breast, so they would also be *undersegmented*. We could consider them incorrect. At some point, I probably concluded that the machine or the labelers could not avoid including something from the inner quadrants without sacrificing the fibroglandular tissue, and that is why I marked the first one as *correct* It would be good to know in what order I read them".
- Example 2: "The three images seem to be oversegmented. It may be that I marked L2 as *correct* because I evaluated it last and understood that it was difficult not to include the pectoralis major since the dense tissue was so well delineated in the segmentation. In this case, it would be good to know in what order I read the three images".

It is worth mentioning that the segmentations in these examples are very similar (high DSC). Additionally, as V1 mentioned for the second example, the pectoral muscle appears segmented as dense tissue. This issue arose in several images. For all images, the breast delineation was automatically annotated with a threshold-based method implemented in Futura Breast. That implementation removes the pectoral muscle only on mediolateral oblique (MLO) views where the pectoral muscle is more likely to appear. All the images evaluated in this study are craniocaudal (CC) views, and in these, the implemented algorithm was unable to remove the pectoral muscle.

Considering V1's comments and seeing how the order of appearance of the segmentations influenced V1's decision, we reanalyze the agreement percentages but taking into account the order of appearance (Figure 11). Our analysis shows that the agreement rate with CM-YNet increased with later appearances: from 75.6% when CM-YNet's segmentation appeared first, to 82.7% when it appeared third. This suggests that V1 developed a more refined labeling criterion after seeing other versions of the same image. Interestingly, for L1 and L2, we observed the opposite trend: agreement with V1 slightly decreased as their segmentations appeared later in the sequence. This may indicate that V1's exposure to alternative segmentations led to increased scrutiny or preference for different delineation styles, particularly when viewing human-labeled segmentations after a model output or another human's annotation. These findings highlight that the sequential context in which segmentations are presented can influence expert validation outcomes.

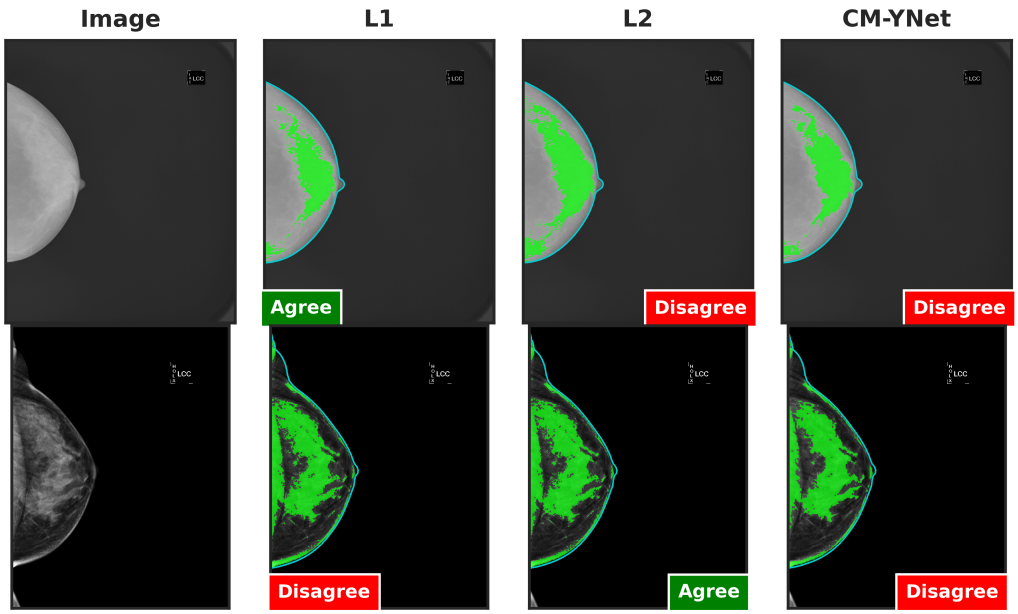


Figure 10. Examples of segmentations where V1 changed his opinion over time: from agree to disagree (first row), and from disagree to agree (second row). Although the visual layout follows a fixed column order (original image, L1, L2, and CM-YNet), the validator viewed the segmentations in a different sequence. For the first-row example, the presentation order was L1, L2, CM-YNet; for the second-row example, it was L1, CM-YNet, L2.

According to the results of the first validation and V1's descriptions, CM-YNet tends to oversegment. This oversegmentation is most likely due to the inclusion of the pectoral muscle, as mentioned earlier. For this reason, we performed a second validation with a different validator (V2), first improving the pectoral muscle exclusion and also providing clearer instructions on how to use the validation labels consistently over time.

3.2. Second Validation

For the second validation, we first improved the breast delineation to effectively remove the pectoral muscle in CC images. The new breast delineation method implemented is described in our previous work [11]. With this method, we ensured that the dense segmentation performed by CM-YNet no longer included the pectoral muscle, preventing bias for V2. Additionally, as seen in the first validation, we aimed to ensure that V2 did not change his criteria over time. To this end, we provided clearer instructions on how to consistently use the validation tool labels:

- Use the correct label only if the segmentation matches the dense tissue, not based on assumptions about the limitations of the labelers' methods.
- Use the oversegmented/undersegmented labels only when it is evident that the segmentation includes significantly more or less tissue than the actual dense tissue.
- Use the incorrect label only for rare cases, such as when the pectoral muscle is included as dense tissue.

3.2.1. Agreement with Each Labeler

Figure 12 illustrates the agreement percentages between V2 and each evaluated labeler (L1, L2, and CM-YNet). The results indicate that the agreement percentage is comparable between manual segmentations—L1 (85.0%) and L2 (82.8%)—and the automatic segmentation CM-YNet (83.2%). This percentage accounts for all segmentations (1500 in total). Notably, the primary source of disagreement varies across labelers: undersegmentation is the main cause of disagreement with L1, whereas oversegmentation is predominant with

L2 and CM-YNet. These findings highlight a higher level of agreement between V2 and the automatic segmentation compared to the results from the first validation (Figure 5).

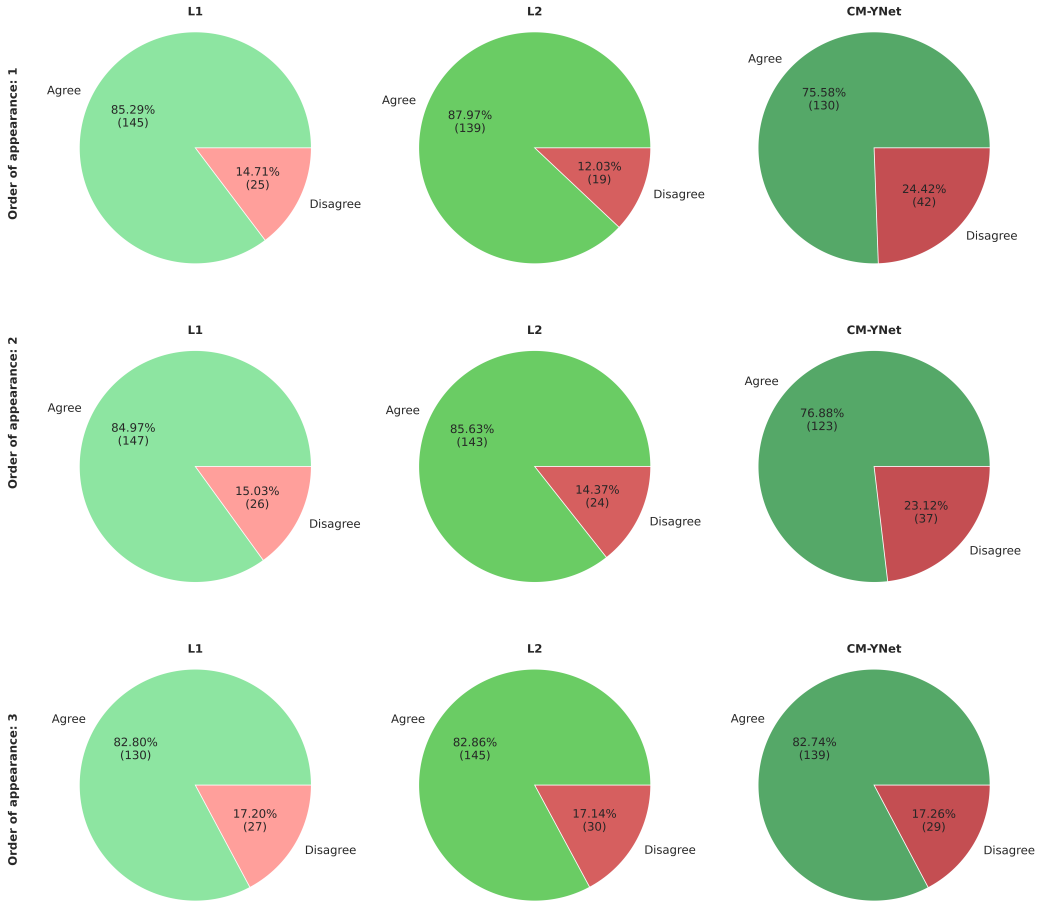


Figure 11. Agreement percentage between V1 and each labeler according to the order of appearance of the segmentations. The first row shows the results when the corresponding segmentations were the first to appear for a given image, the second row shows the segmentations that appeared second for a given image, and similarly for the third row.

The DSC values in Table 4 are based on V2's labels when evaluating pairs of segmentations for the same image, as described in Figure 6. In accordance with the findings from the first validation, the DSC values are higher in cases where V2 assigned the same label to both segmentations. These cases correspond to the majority of images across all evaluated labelers.

Table 4. DSC values for the different labelers (L1, L2, and CM-YNet), based on the labels assigned by V2 to the evaluated segmentations.

Labelers	V2 Label Is the Same	DSC	95% CI
L1 vs. L2	No (91)	0.699 ± 0.193	$(0.658, 0.739)$
	Yes (409)	0.810 ± 0.145	$(0.796, 0.824)$
L1 vs. CM-YNet	No (101)	0.547 ± 0.215	$(0.505, 0.590)$
	Yes (399)	0.752 ± 0.154	$(0.737, 0.767)$
L2 vs. CM-YNet	No (96)	0.619 ± 0.199	$(0.579, 0.660)$
	Yes (404)	0.773 ± 0.137	$(0.759, 0.786)$

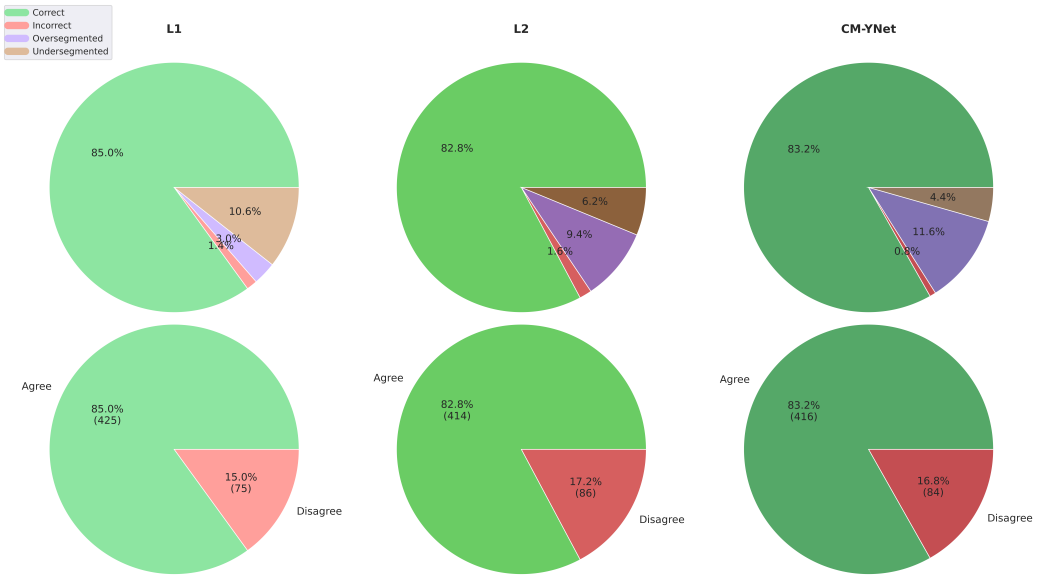


Figure 12. Agreement percentage between V2 and each evaluated labeler for a total of 1500 segmentations.

3.2.2. Intra-Observer Variability

As in the first validation, we analyzed intra-observer variability by randomly presenting 300 segmentations (100 per labeler) to V2 twice. Figures 13 and 14 display the confusion matrices for V2's first and second evaluations of these segmentations. The results indicate that V2 was consistent in the majority of cases but less consistent than V1. Table 5 summarizes the corresponding metrics.

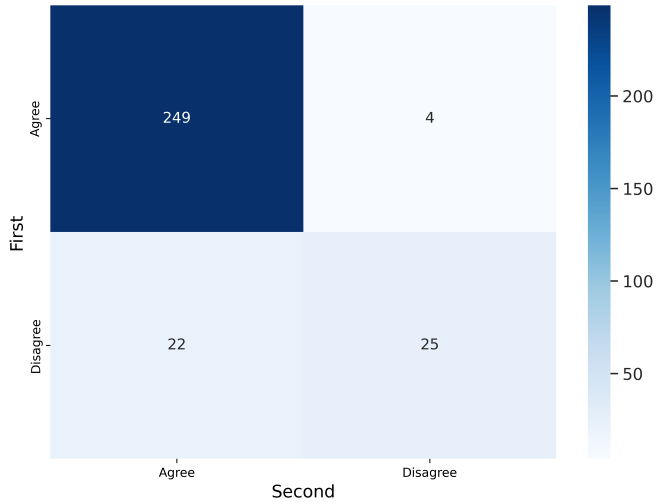


Figure 13. Confusion matrix for the 300 segmentations presented twice to V2. Inconsistencies were observed in 26/300 cases (8.67%).

Table 5. Metrics calculated from the confusion matrix for the 300 segmentations reviewed twice by V2.

Accuracy	Acc. 95% CI−	Acc. 95% CI+	Kappa	Balanced Accuracy	F1	Precision	Recall
0.913	0.887	0.948	0.611	0.758	0.950	0.919	0.984

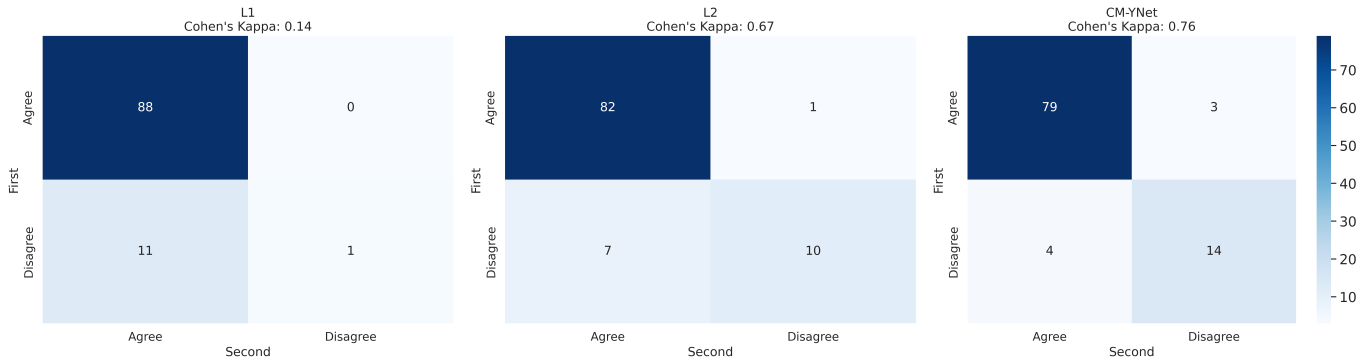


Figure 14. Confusion matrices for each labeler, based on the 300 segmentations shown twice to V2.

Figure 15 provides examples of segmentations produced by CM-YNet where V2 expressed disagreement both times the same segmentation was shown. These disagreements occurred in a total of 14 images. Specifically, V2 assigned 12 cases as oversegmented and 2 cases as undersegmented.

3.2.3. Exploring the Causes of Disagreement

Figure 16 illustrates two examples where V2's decisions varied depending on the order in which the segmentations were presented. In the example in the first row, V2 initially marked the first segmentation (L1) as correct. However, when the next two segmentations (CM-YNet and L2) were shown, both with identical segmentations (DSC = 1), V2 changed his decision to oversegmented for CM-YNet and then correct again for L2. In the example in the second row, V2 indicated the first segmentation (CM-YNet) as versegmented but later marked L1 as correct and L2 as oversegmented. Similar to the first example, L1 and L2 had identical segmentations (DSC = 1).

Figure 17 presents the agreement percentages between V2 and each labeler as a function of their order of appearance. Notably, the agreement with CM-YNet improved significantly, rising from 77.46% when CM-YNet was presented first to 91.41% when it appeared last. This upward trend in agreement was not observed for L1 and L2. For these labelers, agreement percentages increased when they appeared second then decreased again when they were shown third, showing no clear pattern. Therefore, even though the intra-observer variability indicates more inconsistency for V2 compared to V1, it appears that V2 maintained a consistent criterion over time. The inconsistencies found were more likely due to the inherent intra-observer variability that is well-known for this type of segmentation tasks, especially in the medical domain.

Finally, we wanted to explore whether the evaluation sessions influenced V2's results. To this end, an analysis of the validation sessions was conducted. A session was defined as a continuous period during which V2 reviewed images without taking extended breaks. A break of more than one hour marked the start of a new session. A total of 28 sessions were identified (Figure 18), with the longest session lasting nearly three hours. Notably, the final two sessions each consisted of labeling only one image, with a duration of approximately one minute.

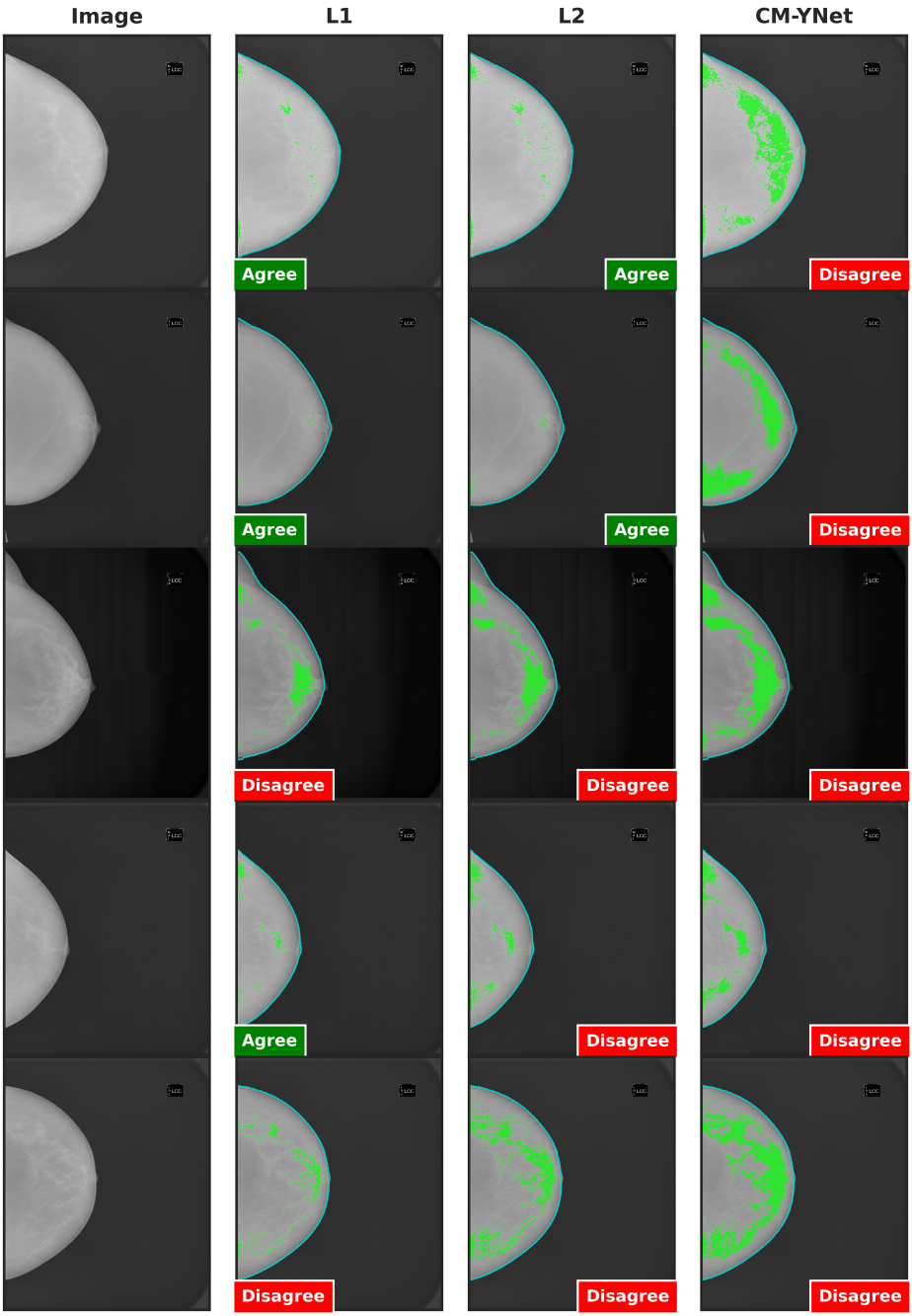


Figure 15. Examples of segmentations that were shown twice to V2 to analyze intra-observer variability. In these examples, V2 indicated disagreement with CM-YNet on both occasions.

Figure 19 presents the average number of images labeled per hour across sessions. Shorter sessions showed a tendency toward the faster validation of images, while in longer sessions (e.g., sessions 15 and 21), V2 spent more time evaluating each image. However, as illustrated in Figure 20, no significant relationship was observed between the labeling speed in different sessions and the agreement percentage with the presented segmentations.

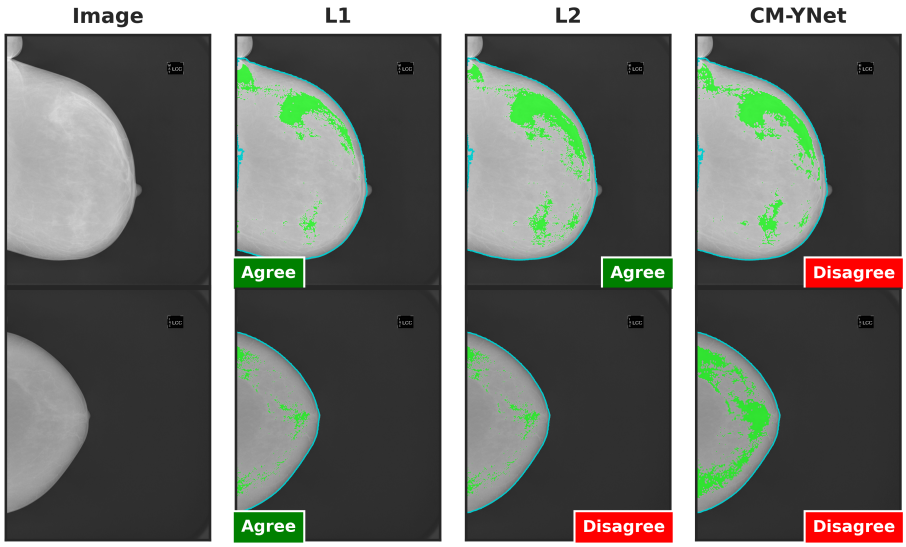


Figure 16. Examples of segmentations where V2 assigned different labels despite identical segmentations (DSC = 1): L2 and CM-YNet in the first row, and L1 and L2 in the second row. Although the visual layout follows a fixed column order (original image, L1, L2, and CM-YNet), the validator viewed the segmentations in a different sequence. In the first-row example, the order of appearance was L1, CM-YNet, and L2. In the second-row example, the order was CM-YNet, L1, and L2.

Figure 17. Agreement percentages between V2 and each labeler based on the order in which segmentations were presented. The first row shows the results when the segmentations were presented first for a given image. The second and third rows correspond to cases where the segmentations appeared second and third, respectively.

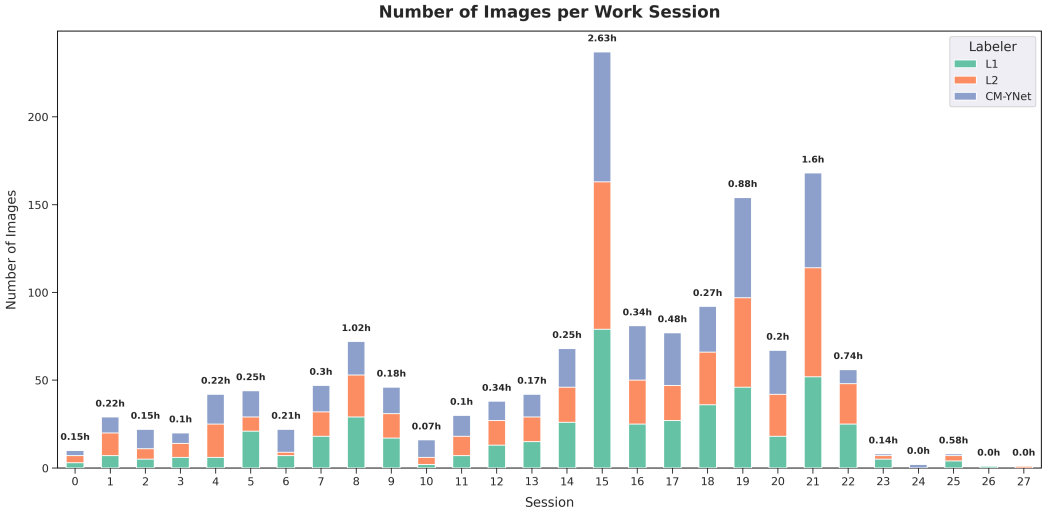


Figure 18. Summary of V2's labeling sessions, showing the duration and number of images labeled in each session.

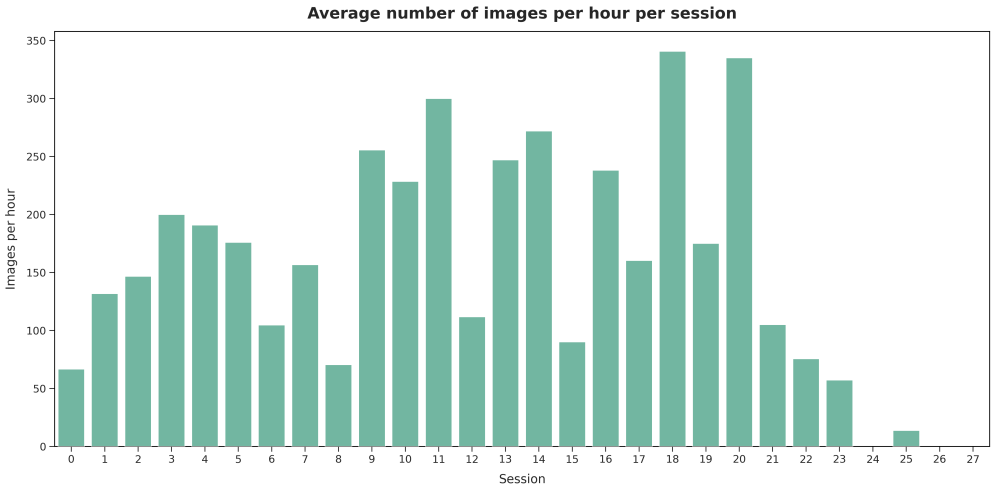


Figure 19. Average number of images labeled per hour during each session by V2.

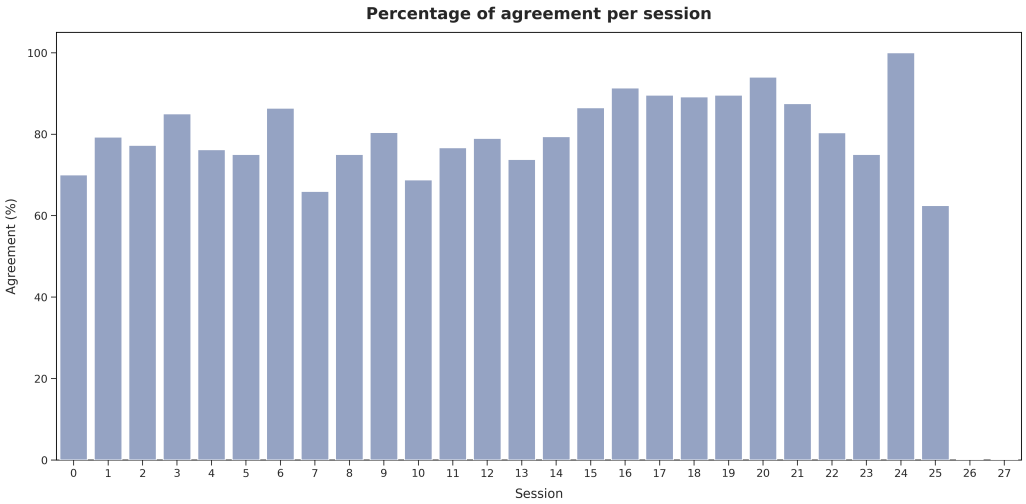


Figure 20. Agreement percentage with the presented segmentations across sessions for V2.

4. Discussion

In this study, we proposed a three-blind validation strategy to compare the agreement between a validator and three different labelers (two human and one deep-learning-based segmentation model). This approach was demonstrated using breast dense tissue segmentation, where a third validator independently evaluated the segmented images without prior knowledge of their origin. Since there is no absolute ground truth—due to inherent intra- and inter-observer variability among human labelers—this strategy provides insight into the level of similarity between the labelers, as assessed by the validator. This can help detect errors in the labels, which can then be refined by human reviewers or used to improve the automatic segmentation model.

The first validation highlighted a critical issue in the preprocessing step of our automatic segmentation model. Specifically, the breast delineation algorithm used prior to dense tissue segmentation failed to exclude the pectoral muscle in CC mammograms, resulting in its incorrect classification as dense tissue by the CM-YNet model. Addressing this limitation, we refined the breast delineation method and subsequently conducted a second validation with a different specialist. For this second validation, we provided clearer and more detailed instructions to the second validator (V2) to ensure consistent use of the validation tool's labels. This improvement led to a higher agreement between V2 and both the automatic and manual segmentations, compared to the results obtained in the first validation.

A critical step toward improving model performance is the identification of systematic errors. In this study, errors in the first validation were detected through expert feedback and visual inspection of the validation results. However, it is important to note that both the nature of these errors and the appropriate corrective strategies can vary significantly depending on the specific use case, dataset composition, and segmentation objectives. As such, error analysis and subsequent model refinement should be tailored to the application context rather than relying on a one-size-fits-all approach.

Unlike traditional validation schemes that focus solely on metric-based comparisons to reference labels, our approach engages an independent validator in a blinded setting to assess the degree of agreement between human and automated segmentations. This design aligns with recent calls for more robust and unbiased evaluation methods in subjective annotation contexts [16,17]. While human-in-the-loop strategies have been used to improve annotation quality and model performance [18], few works have employed a blinded evaluation of multiple sources simultaneously, making our approach a unique contribution to the validation literature.

4.1. Limitations

Despite its strengths, the proposed validation strategy has several limitations. First, the presented use case involved only two radiologists and one independent validator. While this provided a manageable and controlled comparison, it may not capture the full range of inter-observer variability, which is known to influence perceived model accuracy in medical imaging [19,20]. Increasing the number of annotators and applying aggregation methods such as STAPLE [21] could reduce individual bias and improve the reliability of the assessment.

Second, although the validator was blinded to the source of the segmentations, the lack of clinical context during evaluation may reduce validity in real-world diagnostic scenarios [16]. Third, the validation framework currently functions as a static benchmarking tool and does not support real-time model refinement. Iterative improvement mechanisms—such as those used in active learning or semi-supervised frameworks [22,23]—could further enhance the practical utility of this strategy.

Finally, while our results were demonstrated on mammographic segmentation, the generalizability of this framework to other domains (e.g., industrial inspection or environmental monitoring) remains to be empirically validated. As such, claims regarding its broad applicability should be interpreted with caution until further validation studies are conducted.

4.2. Future Work

Building on these findings, several future directions can enhance the utility and scope of the three-blind validation framework. First, clinical validation studies are needed to assess how well this strategy translates into real-world diagnostic workflows and whether it can support or augment radiologist decision-making. Incorporating richer forms of validator feedback and integrating domain-specific context may also improve the robustness of the assessment.

Second, we plan to apply the three-blind validation approach across diverse segmentation tasks—such as those found in industrial inspection and other areas of medical imaging—to evaluate its scalability and generalizability in domains where ground truth is inherently uncertain or contested.

5. Conclusions

This study presented a three-blind validation strategy to compare the agreement between human labelers and a deep-learning-based segmentation model. Applied to breast dense tissue segmentation, this strategy helped identify and address issues in the automatic model while highlighting the potential of automatic segmentation for reproducible results. By ensuring an unbiased evaluation, it offers valuable insights that can help improve segmentation models and support their broader adoption in medical imaging and beyond.

Author Contributions: Conceptualization, A.L., F.J.P.-B. and R.L.; methodology, A.L.; software, A.L.; validation, A.L., F.J.P.-B., M.R. and R.L.; formal analysis, J.C.P.-C. and R.L.; investigation, A.L. and F.J.P.-B.; resources, M.R.; data curation, M.R. and R.T.; writing—original draft preparation, A.L.; writing—review and editing, F.J.P.-B., J.C.P.-C., M.R., R.L. and R.T.; visualization, A.L.; supervision, J.C.P.-C. and R.L.; project administration, J.C.P.-C. and R.L.; funding acquisition, M.R. and J.C.P.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Generalitat Valenciana through IVACE (Valencian Institute of Business Competitiveness, https://www.ivace.es, (accessed on 1 May 2025)) through a grant to support ITI's activity in independent research and development, results dissemination and knowledge and technology transfer to business under project IMAMCA/2025/11. This work has also been supported by the Generalitat Valenciana through IVACE and by the European Union through FEDER funding under project IMDEEA/2024/79.

Institutional Review Board Statement: Approval for data usage was approved by the Ethics Committee of IMIM (Hospital del Mar Medical Research Institute) (protocol code 2017/7442/I, approved on 14 December 2017). The study was conducted according to the guidelines of the Declaration of Helsinki.

Informed Consent Statement: Patient consent was waived for the IMIM dataset since anonymized retrospective data were used.

Data Availability Statement: A generalized version of the three-blind validation tool, developed and provided by our team for use in any image segmentation task, has been made publicly available through GitLab https://egitlab.iti.es/praia-salud/segmentation-validation-tool.git, (accessed on 1 May 2025).

Acknowledgments: We sincerely thank María Casals and Natalia Arenas for their expertise and meticulous effort in labeling the mammograms used in the breast dense tissue use case. We also gratefully acknowledge Javier Azcona and Miguel Ángel Sánchez Fuster for their critical role as validators of the segmentations.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, J.; Xie, G.; Wang, J.; Li, S.; Wang, C.; Zheng, F.; Jin, Y. Deep Industrial Image Anomaly Detection: A Survey. *Mach. Intell. Res.* **2024**, *21*, 104–135. [CrossRef]
2. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
3. Al Shafian, S.; Hu, D. Integrating Machine Learning and Remote Sensing in Disaster Management: A Decadal Review of Post-Disaster Building Damage Assessment. *Buildings* **2024**, *14*, 2344. [CrossRef]
4. Schmidt, A.; Morales-Álvarez, P.; Molina, R. Probabilistic Modeling of Inter- and Intra-observer Variability in Medical Image Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023; pp. 1234–1243. [CrossRef]
5. Diers, J.; Pigorsch, C. A Survey of Methods for Automated Quality Control Based on Images. *Int. J. Comput. Vis.* **2023**, *131*, 2553–2581. [CrossRef]
6. Kahl, K.; Lüth, C.T.; Zenk, M.; Maier-Hein, K.; Jaeger, P.F. ValUES: A Framework for Systematic Validation of Uncertainty Estimation in Semantic Segmentation. *arXiv* **2024**, arXiv:2401.08501. [CrossRef]
7. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *arXiv* **2020**, arXiv:2001.05566. [CrossRef] [PubMed]
8. Larroza, A.; Pérez-Benito, F.J.; Perez-Cortes, J.C.; Román, M.; Pollán, M.; Pérez-Gómez, B.; Salas-Trejo, D.; Casals, M.; Llobet, R. Breast Dense Tissue Segmentation with Noisy Labels: A Hybrid Threshold-Based and Mask-Based Approach. *Diagnostics* **2022**, *12*, 1822. [CrossRef] [PubMed]
9. Streamlit, I. Streamlit: The Fastest Way to Build Data Apps. 2024. Available online: https://streamlit.io (accessed on 27 December 2024).
10. Gandomkar, Z.; Siviengphanom, S.; Suleiman, M.; Wong, D.; Reed, W.; Ekpo, E.U.; Xu, D.; Lewis, S.J.; Evans, K.K.; Wolfe, J.M.; et al. Reliability of radiologists' first impression when interpreting a screening mammogram. *PLoS ONE* **2023**, *18*, e0284605. [CrossRef] [PubMed]
11. Larroza, A.; Pérez-Benito, F.J.; Tendero, R.; Perez-Cortes, J.C.; Román, M.; Llobet, R. Breast Delineation in Full-Field Digital Mammography Using the Segment Anything Model. *Diagnostics* **2024**, *14*, 1015. [CrossRef] [PubMed]
12. Dice, L.R. Measures of the amount of ecologic association between species. *Ecology* **1945**, *26*, 297–302. [CrossRef]
13. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
14. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]
15. Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. The balanced accuracy and its posterior distribution. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 3121–3124. [CrossRef]
16. Reinke, A.; Tizabi, M.D.; Sudre, C.H.; Eisenmann, M.; Rädsch, T.; Baumgartner, M.; Acion, L.; Antonelli, M.; Arbel, T.; Bakas, S.; et al. Common limitations of image processing metrics: A picture story. *Nat. Commun.* **2021**, *12*, 1–13.
17. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [CrossRef] [PubMed]
18. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29. [CrossRef]
19. Warfield, S.K.; Zou, K.H.; Wells, W.M. Validation of image segmentation by estimating rater bias and variance. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2008**, *366*, 2361–2375. [CrossRef]
20. Zijdenbos, A.P.; Dawant, B.M.; Margolin, R.A.; Palmer, A. Morphometric analysis of white matter lesions in MR images: Method and validation. *IEEE Trans. Med. Imaging* **1994**, *13*, 716–724. [CrossRef] [PubMed]
21. Warfield, S.K.; Zou, K.H.; Wells, W.M. Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation. *IEEE Trans. Med. Imaging* **2004**, *23*, 903–921. [CrossRef] [PubMed]

22. Kohl, S.A.A.; Romera-Paredes, B.; Meyer, C.; De Fauw, J.; Ledsam, J.R.; Maier-Hein, K.H.; Eslami, S.M.A.; Rezende, D.J.; Ronneberger, O. A Probabilistic U-Net for Segmentation of Ambiguous Images. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 3–8 December 2018; Volume 31. [CrossRef]
23. Tajbakhsh, N.; Jeyaseelan, L.; Li, Q.; Chiang, J.N.; Wu, Z.; Ding, X. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Med. Image Anal.* **2020**, *63*, 101693. [CrossRef] [PubMed]

Journal of *Imaging*

MDPI

Article

Recovery and Characterization of Tissue Properties from Magnetic Resonance Fingerprinting with Exchange

Naren Nallapareddy * and Soumya Ray *

Department of Computer and Data Sciences, Case Western Reserve University, Olin 516, 2101 Martin Luther King Jr Dr, Cleveland, OH 44106, USA

* Correspondence: nxn151@case.edu (N.N.); sray@case.edu (S.R.)

Abstract: Magnetic resonance fingerprinting (MRF), a quantitative MRI technique, enables the acquisition of multiple tissue properties in a single scan. In this paper, we study a proposed extension of MRF, MRF with exchange (MRF-X), which can enable acquisition of the six tissue properties T_1^a,T_2^a, T_1^b, T_2^b, ρ and τ simultaneously. In MRF-X, 'a' and 'b' refer to distinct compartments modeled in each voxel, while ρ is the fractional volume of component 'a', and τ is the exchange rate of protons between the two components. To assess the feasibility of recovering these properties, we first empirically characterize a similarity metric between MRF and MRF-X reconstructed tissue property values and known reference property values for candidate signals. Our characterization indicates that such a recovery is possible, although the similarity metric surface across the candidate tissue properties is less structured for MRF-X than for MRF. We then investigate the application of different optimization techniques to recover tissue properties from noisy MRF and MRF-X data. Previous work has widely utilized template dictionary-based approaches in the context of MRF; however, such approaches are infeasible with MRF-X. Our results show that Simplicial Homology Global Optimization (SHGO), a global optimization algorithm, and Limited-memory Bryoden–Fletcher–Goldfarb–Shanno algorithm with Bounds (L-BFGS-B), a local optimization algorithm, performed comparably with direct matching in two-tissue property MRF at an SNR of 5. These optimization methods also successfully recovered five tissue properties from MRF-X data. However, with the current pulse sequence and reconstruction approach, recovering all six tissue properties remains challenging for all the methods investigated.

Keywords: medical imaging; MRI; MRF; inverse problem; optimization

Academic Editor: Leonardo Rundo

Received: 21 February 2025
Revised: 5 May 2025
Accepted: 13 May 2025
Published: 20 May 2025

Citation: Nallapareddy, N.; Ray, S. Recovery and Characterization of Tissue Properties from Magnetic Resonance Fingerprinting with Exchange. *J. Imaging* **2025**, *11*, 169. https://doi.org/10.3390/jimaging11050169

1. Introduction

Magnetic resonance imaging (MRI) is one of the most valuable tools in a clinician's toolbox for non-invasively diagnosing soft-tissue diseases [1–3]. In clinical practice, a majority of MRI scans acquired are qualitative in nature. During the course of diagnosis, a radiologist might request several of these qualitative scans to obtain a complete diagnostic picture [4]. Although this approach provides valuable information to the radiologist, it restricts objective characterization of diseases and adds to patient costs and discomfort [5,6]. Additionally, variation in interpretation among radiologists could lead to significant differences in treatment plans and clinical outcomes [7].

To address some of these challenges, advanced MRI techniques such as quantitative MRI have been gaining popularity. Quantitative MRI aims to extract tissue properties such as longitudinal relaxation time T_1 (milliseconds), transverse relaxation time T_2 (milliseconds), and proton diffusion [8]. Magnetic resonance fingerprinting (MRF) is a technique

that can extract multiple tissue properties from a single MRI scan [9]. Multiple tissue property images from a single scan decrease the costs and discomfort faced by patients and can improve diagnostic outcomes. Access to multiple tissue properties may also allow for the use of MRF in novel diagnostic procedures as yet unknown. However, reconstructing tissue properties from MRF images poses an algorithmic challenge due to the noisy signal generated at each voxel. The traditional approach of fitting a curve to recover tissue properties cannot be applied. Instead, the state of the art is to use a template dictionary matching approach for reconstructing tissue properties (described in detail in Section 2.2). This strategy, although robust [10], does not scale well as the number of tissue properties increases.

In this paper, we study the problem of recovering five and six tissue properties from MRF with exchange (MRF-X), where a signal from each voxel is represented by two components. We characterize and visualize the error surface formed by the inner product of an MRF-X signal and simulation of Bloch–McConnell equations derived from a range of tissue properties. We then study the use of optimization techniques to accurately retrieve tissue properties from such an MRF-X signal. Our results are promising but also reveal fundamental uncertainties about the feasibility and accuracy of acquiring multiple tissue properties. To our knowledge, no prior work has addressed the characterization or recovery problems from quantitative MRF in high dimensions.

This paper is organized as follows. In Section 2, we provide essential background on MRF and its extension, MRF-X. In Section 3, we study the error surface of MRF-X with respect to multiple tissue properties and how it changes from two to higher dimensions. In Section 4, we present a technical overview of the optimization algorithms we propose to recover tissue properties from MRF-X signals. In Section 5, we present our results applying these methods to a large dataset of simulated MRF-X signals generated with many different tissue properties and discuss the implications.

2. Background

2.1. Magnetic Resonance Fingerprinting

In standard quantitative MRI, the signal at each voxel conforms to known signal trajectories, which have well-studied mathematical models associated with them. Broadly, standard quantitative MRI uses either an exponential model of signal recovery [11,12] or a steady state signal model [13–16]. Tissue properties are then estimated by fitting acquired signal trajectories to the relevant mathematical model.

In contrast, MRF leverages recent advancements in computation power to acquire multiple tissue properties in a single scan [17]. During the acquisition of MRF, a pre-defined sequence of Radio Frequency (RF) pulses, known as the pulse sequence, is transmitted through the RF coil. This pulse sequence is governed by a set of pulse sequence parameters that control the amplitude, phase and delay of the RF pulses [18]. By carefully designing the pulse sequence, we can systematically acquire MRF signals which are sensitive to specific tissue properties. In MRF, a randomly varying acquisition scheme is employed to produce signal changes that are uniquely determined by the tissue properties present at each voxel. This variable acquisition scheme, although not unique to MRF [19], allows flexibility in pulse sequence design. Accelerated acquisition leads to corrupted MRF signal evolutions which are usually modeled as signal with added white Gaussian noise, which has shown good empirical performance [17].

2.2. Tissue Property Recovery from MRF Using Explicit Dictionary

Due to the complex pulse sequence employed in MRF, the signal trajectory does not follow a known closed-form function. To address this, the original MRF paper [9] proposed an explicit template dictionary-based approach. The underlying idea is to simulate known signal trajectories using an exhaustive combination of tissue property values using Bloch [20] equations.

The generated signals from the Bloch simulation are collected for each combination of tissue property values into a template dictionary (Figure 1a). Next, the captured MRF signal trajectories (Figure 1b) are matched with the template dictionary to retrieve the tissue property values (Figure 1d–g) In this process, matching refers to taking an inner product between the observed signal trajectory and each element of the dictionary. The properties yielding the highest dot product (most similar known trajectory) are selected as the tissue properties. This process is repeated for each voxel to form a tissue property map. The template dictionary is a 2D matrix of values ($M \times N$), with each column n_i, $i \in \{1, \ldots, N\}$, representing the signal evolution of a single combination of tissue properties. Each MRF acquisition strategy, such as MRF-FISP [21], uses a different number of columns (N) to accurately extract tissue property values. The number of columns M depends on several factors, including the number of tissue properties being estimated, the resolution of the tissue property values in the dictionary, and the range of tissue property values as determined by the researcher. For example, Chen et al. [22] used 20 k columns to represent the dictionary that is generated using a combination of T_1 and T_2 properties. Conversely, Hong et al. [23] employed a larger dictionary with 64 million columns to estimate four tissue properties: T_1, T_2, off-resonance, and T_2^*. In this context, off-resonance indicates a measure of inhomogeneity in the main magnetic field, and T_2^* represents the observed or effective T_2 resulting from such inhomogeneities. As can be seen from these examples, when estimating a large number of tissue properties, explicit dictionary-based template matching can become prohibitively large.

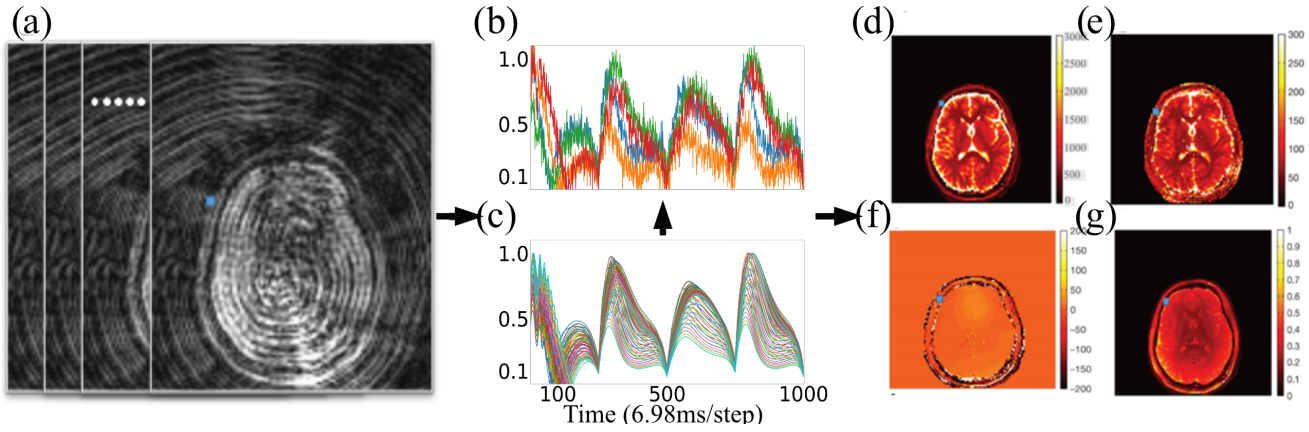


Figure 1. MRF tissue property mapping pipeline. Multiple noisy images (**a**) are captured with the scanner. Individual signals for each voxel (**b**) are compared with an explicit dictionary (**c**) to recover the tissue properties generating the signals. The generated maps represent tissue properties such as T_1 (**d**), T_2 (**e**), B_1 mapping (**f**), and proton density (**g**), respectively [9].

Recent research in the field of MRF has emphasized the importance of accelerating reconstruction speed. However, it is noteworthy that many of these methods [24–26] continue to rely on dictionary generation. We note that generating an exhaustive dictionary with millions of entries is a computationally demanding task that can take several days to complete. Moreover, the size of the dictionary increases exponentially with the number of tissue properties being considered. Consequently, generating an explicit dictionary for

more than four tissue properties can become impractical, especially in the context of clinical applications where rapid analysis is critical.

2.3. Magnetic Resonance Fingerprinting with Exchange (MRF-X)

In the MRF-X approach [27], a signal acquired from each voxel of the scanned region is represented by two components, labeled A and B. Each compartment is considered a distinct region that has its independent set of tissue properties (T_1, T_2) (ms). The volume ratio of compartment A is denoted by ρ. A continuous exchange of protons occurs between compartment A and compartment B. This exchange is measured using an exchange rate τ (s^{-1}). This modeling using two compartments has several potential uses such as the monitoring of myelin in patients with degenerative disorders of the brain and diffuse fibrosis of the heart [28]. We simulate the two-compartment model using the Bloch–McConnell equations [29], which are an extension of the Bloch equations for chemical exchange.

The acquisition of multi-component maps is constrained by certain physical limitations. In medical imaging, it is widely recognized that physical processes that occur at a rate faster than acquisition speed of the protocol cannot be measured [30]. Traditional MRI collects information at a rate that is commensurate with the T_1 and T_2 relaxation times. However, this rate is slower than the rate of chemical exchange that occurs between multiple compartments. This limitation restricts the conventional MRI's ability to acquire chemical exchange dynamics. MRF potentially addresses this difficulty by collecting samples at a rate (TR) of 6–20 ms, which aligns with the time scale of chemical exchange between multiple components in the brain. Capitalizing on this, Hamilton et al. [27] introduced a new technique that leverages MRF to acquire multi-component maps along with chemical exchange, which they call MRF-X.

In a clinical context, multi-components maps of relaxation times with chemical exchange are not typically acquired at present. At present, the acquisition of these multi-component maps remains a challenging task due to prohibitively long scans which increase patient discomfort and associated costs. Further, validating the performance of tissue property recovery algorithms in an actual scanner will require designing phantoms with desired tissue properties so the signal from the scanner can be recorded. Currently, there are no standardized phantoms for MRF-X like the ISMRM/NIST system phantom for MRF [10]. Despite these challenges, the potential acquisition of multi-component maps may enhance diagnostic capabilities ultimately contributing to better patient outcomes.

2.4. MRF-X Modeling Using Bloch–McConnell Equations

The influence of chemical exchange on the magnetic resonance signal is exactly described by the Bloch–McConnell equations, which extend the standard Bloch equations to model systems with nuclei that dynamically exchange between multiple local environments. These equations incorporate the exchange rate τ that measures the transfer of magnetization accompanying the physical movement of nuclei between multiple compartments and the fractional contribution of each compartment to the overall magnetization measured using ρ. The multiple compartments have their independent relaxation properties, T_1^a, T_2^a, T_1^b, T_2^b and are described by the following differential equation:

$$\frac{dM}{dt} = AM + C \tag{1}$$

$$M = \begin{bmatrix} M_x^A \\ M_y^A \\ M_z^A \\ M_x^B \\ M_y^B \\ M_z^B \end{bmatrix}$$

$$A = \begin{bmatrix} -\left(\frac{1}{T_{2a}} + \tau\right) & dA & 0 & \tau & 0 & 0 \\ -dA & -\left(\frac{1}{T_{2a}} + \tau\right) & 0 & 0 & \tau & 0 \\ 0 & 0 & -\left(\frac{1}{T_{1a}} + \tau\right) & 0 & 0 & \tau \\ \tau & 0 & 0 & -\left(\frac{1}{T_{2b}} + \tau\right) & dB & 0 \\ 0 & \tau & 0 & -dB & -\left(\frac{1}{T_{2b}} + \tau\right) & 0 \\ 0 & 0 & \tau & 0 & 0 & -\left(\frac{1}{T_{1b}} + \tau\right) \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ 0 \\ \frac{\rho M_{\text{total}}}{T_{1a}} \\ 0 \\ 0 \\ \frac{(1-\rho) M_{\text{total}}}{T_{1b}} \end{bmatrix}$$

where M represents the magnetization vector, A is the evolution matrix, and C is the constant vector.

The evolution matrix A contains the relaxation terms (T_1^a, T_2^a, T_1^b, T_2^b), and the chemical exchange term (τ). The constant vector C contains the fractional contribution of each compartment to the overall magnetization measured using ρ.

The crucial connection to proton exchange described by the Bloch–McConnell equations occurs continuously and simultaneously with the evolution driven by the MRF-X pulse sequence. With each repetition time (TR) and radio frequency (RF) pulse, magnetization is constantly being redistributed between the two compartments (A and B) according to the specific exchange rate τ. This redistribution is influenced by the sequences varying RF pulses and ongoing relaxation processes. MRF probes the system's response continuously. As a result, the exchange rate τ becomes integrated into the signal evolution captured by the MRF-X pulse sequence.

2.5. Deep Learning for High-Dimensional MRF

In the past decade, Artificial Intelligence (AI), and specifically Deep Learning (DL), in the context of MRI research has gained a lot of importance. In [31], the authors provide a detailed overview of the impact of AI on MRI research, from acquisition to disease prediction. Further, in the context of MRF, DL methods have been proposed as alternatives to overcome the limitations of traditional template matching.

Primarily, DL applications in MRF can be categorized into two distinct categories. The first category involves using the DL model as a faster Bloch simulator generating the dictionary elements from tissue properties for forming a template dictionary. For instance, Yang et al. [32] used a Generative Adversarial Network (GAN)-based approach to achieve a 10,000× speed increase in the generation of an MRF dictionary. Similarly, Hamilton et al. [33] used a fully connected feed-forward network to generate a cardiac

MRF template dictionary, taking individual variations in heart rate and sequence timing into account.

In the second category, DL is utilized as a replacement to the complete MRF tissue property estimation pipeline, enabling generation of tissue properties from the MRF signals acquired during scanning. For example, the MRF deep reconstruction network (DRONE) proposed by Cohen et al. [34] is a four-layer deep neural network (DNN) that generates tissue properties T_1 and T_2 directly from MRF signals. The authors report that DRONE achieves results on par with conventional techniques but with a 300× speed advantage, consuming only 5% of the memory required by an explicit MRF dictionary. In other work, Fang et al. [35] leveraged spatial information of neighboring voxels in MRF, facilitating accurate quantification of tissue properties from highly undersampled MRF data. As far as we are aware, no prior work has leveraged DL to either extract multiple tissue properties or accelerate generation of explicit high-dimensional (greater than four tissue properties) MRF dictionaries. In contrast to such approaches, we focus on recovering high-dimensional tissue properties from MRF-X through optimization.

3. Nature of MRF Objective Function

As explained above, to find the tissue properties from a captured MRF signal, an inner product is taken between this signal and simulations from the Bloch equations. To find the true tissue properties that generated the signal, this inner product surface must be searched. The best match is the set of tissue properties that maximizes the inner product or, alternatively, minimizes an error function derived from the inner product. Thus, it is important to understand the nature of the inner product function or error function in order to understand the behavior of algorithms attempting to carry out tissue property reconstruction. In this section, we visualize the surface of the inner product function between the noisy signal, which acts as a surrogate of the data acquired from the MRI scanner, and the signals obtained from the Bloch simulations for a set of candidate tissue properties. We perform this for different tissue property dimensionalities.

Let the signal obtained from a location (i, j) on the physical surface of the scanned object be denoted $X(t)$, where $t \in \{1, \ldots, T\}$ represents time and T represents the total number of time steps captured by the scanner. We denote each Bloch simulation as $B(\theta, t)$, where t denotes time and $\theta \in \Theta^d$ is the space of possible d-dimensional tissue properties, quantized to a suitable granularity to visualize the surface (Table 1). We will use the notation X and $B(\theta)$ to represent whole trajectories.

Table 1. Ranges and granularities of tissue property values in 2D and 6D models.

Tissue Property	Min	Max	Step
Two-tissue property model			
$\mathbf{T_1}$ (ms)	700	3000	0.23
$\mathbf{T_2}$ (ms)	5	350	0.03
Six-tissue property model			
$\mathbf{T_1^a}$ (ms)	100	1400	13
$\mathbf{T_2^a}$ (ms)	5	100	0.95
$\mathbf{T_1^b}$ (ms)	1500	3500	20
$\mathbf{T_2^b}$ (ms)	100	400	3
$\boldsymbol{\tau}$ (s^{-1})	0.05	10	0.01
$\boldsymbol{\rho}$ (%)	5	95	0.9

The signal captured from a scanner is typically noisy. To model this, we generate signals using additive white Gaussian noise (AWGN) with varying signal-to-noise ratios (SNRs). We denote a signal with SNR μ as X_μ. Then, we define:

$$f(\theta) = 1 - \frac{\langle X_\mu, B(\theta) \rangle}{\|X_\mu\|_2 \|B(\theta)\|_2} \tag{2}$$

Here, $f(\theta)$ represents the the objective (error) function surface between the input signal X_μ and the Bloch signal. Angular brackets $\langle , \rangle$ represent the inner product of the noisy signal from the scanner and the signal from Bloch simulation. We normalize the inner product using l_2-norm ($\|X_\mu\|_2$) of the noisy signal multiplied with l_2-norm ($\|B(\theta)\|_2$) of the signal from Bloch simulation. We wish to find $\hat{\theta} = \arg\min_{\theta \in \Theta^d} f(\theta)$.

In Figure 2, we show the error function for two-dimensional tissue properties, consisting of $\{T_1, T_2\}$ for two specific target tissue property combinations observed in the white matter and gray matter areas of the brain, respectively [36]. We employed template matching with the explicit dictionary discussed in the Section 2.2 to generate the contour maps. This explicit dictionary consists of 10,000 elements (granularity shown in Table 1, top).

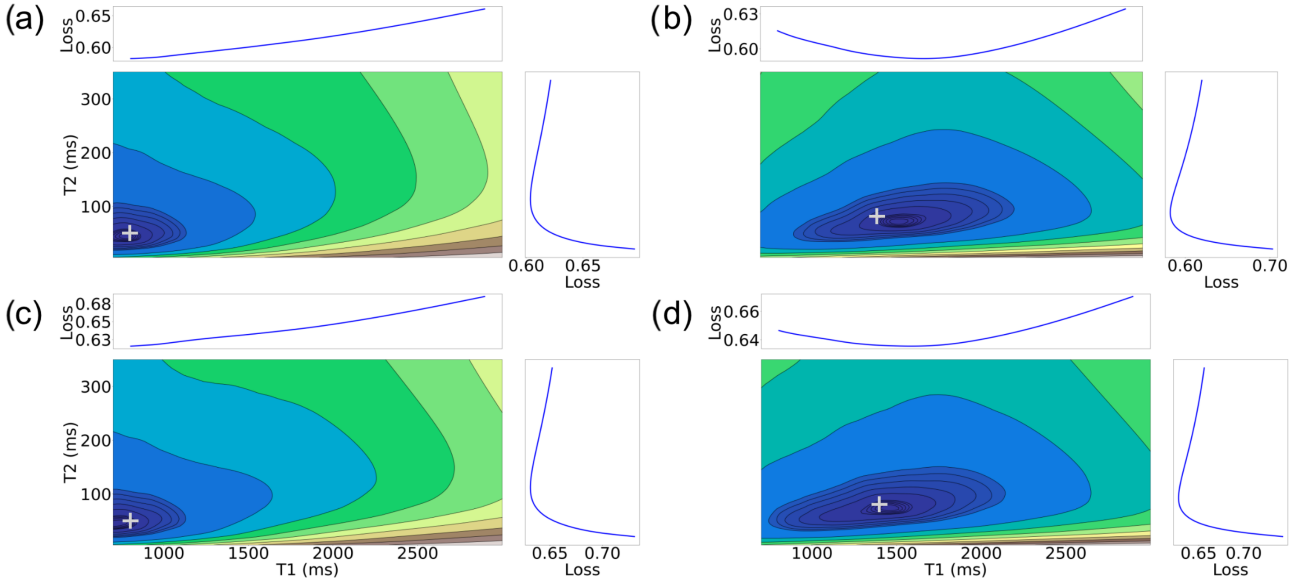


Figure 2. Error function $f(\theta)$ for MRF signal X_μ with $\theta \in \Theta^2 = \{T_1, T_2\}$. (**a**) Top-left: $\mu = 1$, [$T_1 = 800$ ms, $T_2 = 50$ ms]; (**b**) Top-right: $\mu = 1$, [$T_1 = 1400$ ms, $T_2 = 80$ ms]; (**c**) Bottom-left: $\mu = 5$, [$T_1 = 800$ ms, $T_2 = 50$ ms]; (**d**) Bottom-right: $\mu = 5$, [$T_1 = 1400$ ms, $T_2 = 80$ ms]. [$T_1 = 800$ ms, $T_2 = 50$ ms] and [$T_1 = 1400$ ms, $T_2 = 80$ ms] is indicative of white matter and gray matter in brain at 3T MRI scanner. Darker blue contours show lower f (better). "Loss" graphs show f for each axis averaged over the other axis. The symbol + indicates the true tissue property combination in each case.

Upon examining the tissue property combination of [800 ms, 50 ms], we note that the minimum of the error surface aligns closely with the actual target, as denoted by the plus (+) symbol, for both SNR 1 and 5 cases. However, for the tissue property combination of [1400 ms, 80 ms], there is a small discrepancy between the actual tissue property and minimum of the error surface at SNR 1, indicated by the + not appearing at the center of the dark blue region. As expected, the discrepancy is reduced for the higher SNR.

From these figures, two key observations emerge. First, the sensitivity of the MRF explicit dictionary template matching procedure is dependent on the tissue property combination. This is expected from the the MRF-FISP pulse sequence's differential sensitivity

to distinct regions of the tissue property space, as highlighted in Jiang et al. [21]. Second, for low SNR, there may be a systemic mismatch that is larger than the explicit dictionary resolution. For example, Figure 2b shows that the true values lie outside the region with the lowest f values. This is not due to a lack of granularity of sampling based on the granularity shown in Table 1. In such a case, any algorithm relying on template matching in some form is likely to produce irreducible errors.

In Figures 3 and 4, we show f for an MRF-X signal for six-dimensional tissue properties, namely (T_1^a,T_2^a,T_1^b,T_2^b,ρ,τ) for two specific target tissue property values, again motivated by values from the white and gray matter regions of the brain. Since we cannot directly visualize a 6D error surface, we consider individual 2D projections by grouping the tissue properties according to their respective compartments A and B. In such projections, we fix the other dimensions to the target values. To generate each 2D projection, we use a dictionary of 10,000 elements (granularity shown in Table 1, bottom).

From Figures 3 and 4, we observe that:

- Similar to the 2D case, there are still well-defined, non-disjoint regions with minimum f in the space. So it is feasible (in theory) to achieve solutions close to the target values, as in the 2D case.
- The gradient structure is generally steeper in some regions than in the 2D case, as indicated by the larger number of narrower contours.
- For higher SNRs, there is still a relatively good alignment between some of the target tissue properties, such as for (T_1^a, T_2^a, T_1^b, T_2^b), with the actual target. This observation aligns with expectations, given that the MRF-X scan is sensitive to the tissue properties T_1 and T_2 [27]. Observing (T_1, T_2) property pairs (sub-figures d,e), it is evident that the minimum of the error surface closely matches the actual target shown represented by $+$ symbol.
- There are large "plateaus" of the error function around the minimum for some tissue properties even for high SNRs. Thus, even when the minimum aligns well with the true parameters, there may be algorithmic challenges finding it due to the error function structure, which has a combination of both steep gradients and large plateaus.
- As expected, the alignment between the minimum of the error surface and the actual target is notably more accurate for SNR 5 compared to SNR 1.
- Estimating the (ρ, τ) tissue properties is the most challenging aspect of tissue characterization. Assuming a two-compartment model, we hypothesize that the rate of exchange of protons (ρ) and the volume ratio of a compartment (without loss of generality we can assume it is the ratio of compartment b over the total volume) (τ) to the voxel are intrinsically related. If we assume that protons do not "leak" between voxels and exchange only happens inside each measured region (conservation of protons), then at equilibrium there must be an inverse relation between the rate of exchange and the proportion of the compartment to the voxel volume. This implies that multiple different combinations of tissue properties could lead to the same equilibrium state.

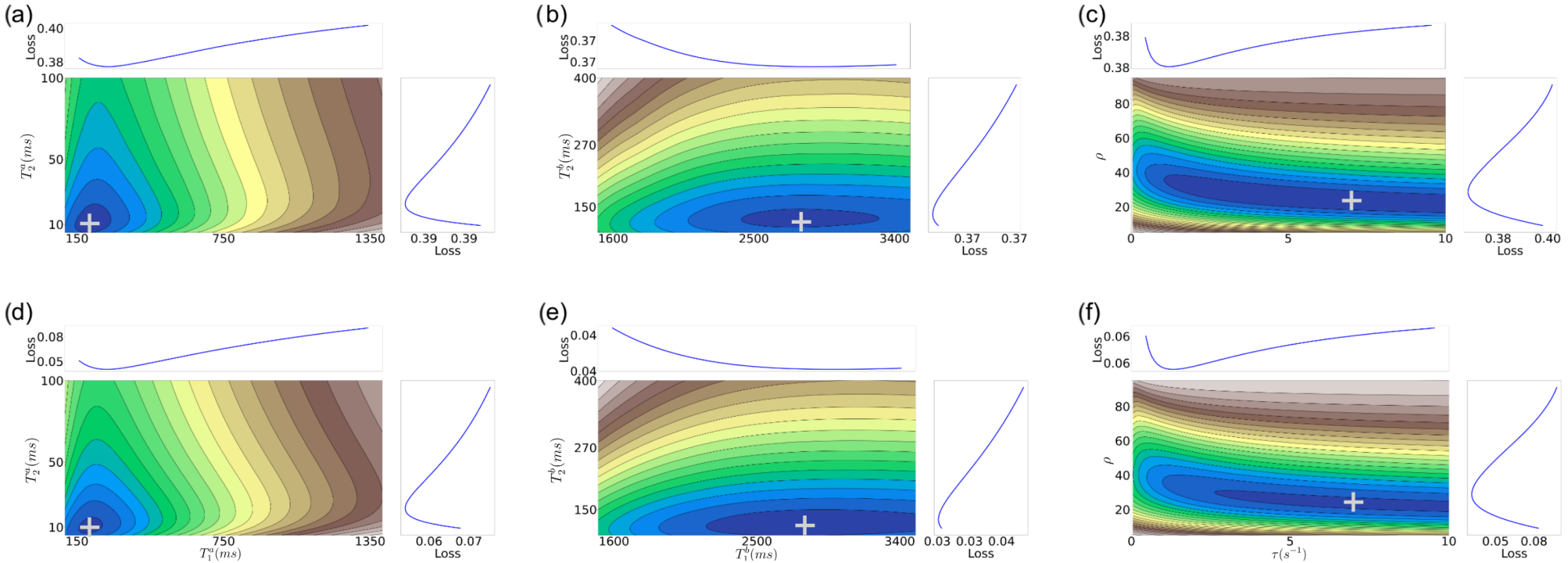


Figure 3. Error function $f(\theta)$ for MRF-X signal X_μ with $\theta \in \Theta^6 = \{T_1^a, T_2^a, T_1^b, T_2^b, \tau, \rho\}$. (**a**) Top-left: $\mu = 1$, $[T_1^a = 200\,\text{ms}, T_2^a = 10\,\text{ms}]$; (**b**) Top-middle: $\mu = 1$, $[T_1^b = 2800\,\text{ms}, T_2^b = 120\,\text{ms}]$; (**c**) Top-right: $\mu = 1$, $[\tau = 7.0\,\text{s}^{-1}, \rho = 23.7]$; (**d**) Bottom-left: $\mu = 5$, $[T_1^a = 200\,\text{ms}, T_2^a = 10\,\text{ms}]$; (**e**) Bottom-middle: $\mu = 5$, $[T_1^b = 2800\,\text{ms}, T_2^b = 120\,\text{ms}]$; (**f**) Bottom-right: $\mu = 5$, $[\tau = 7.0\,\text{s}^{-1}, \rho = 23.7]$. Values $[T_1^a = 200\,\text{ms}, T_2^a = 10\,\text{ms}, T_1^b = 2800\,\text{ms}, T_2^b = 120\,\text{ms}, \tau = 7.0\,\text{s}^{-1}, \rho = 23.7]$ correspond to white matter tissue properties from [37]. Darker blue contour lines indicate lower f (better). "Loss" graphs show f along one axis averaged over the other. The "+" symbol marks the true tissue-property combination in each panel.

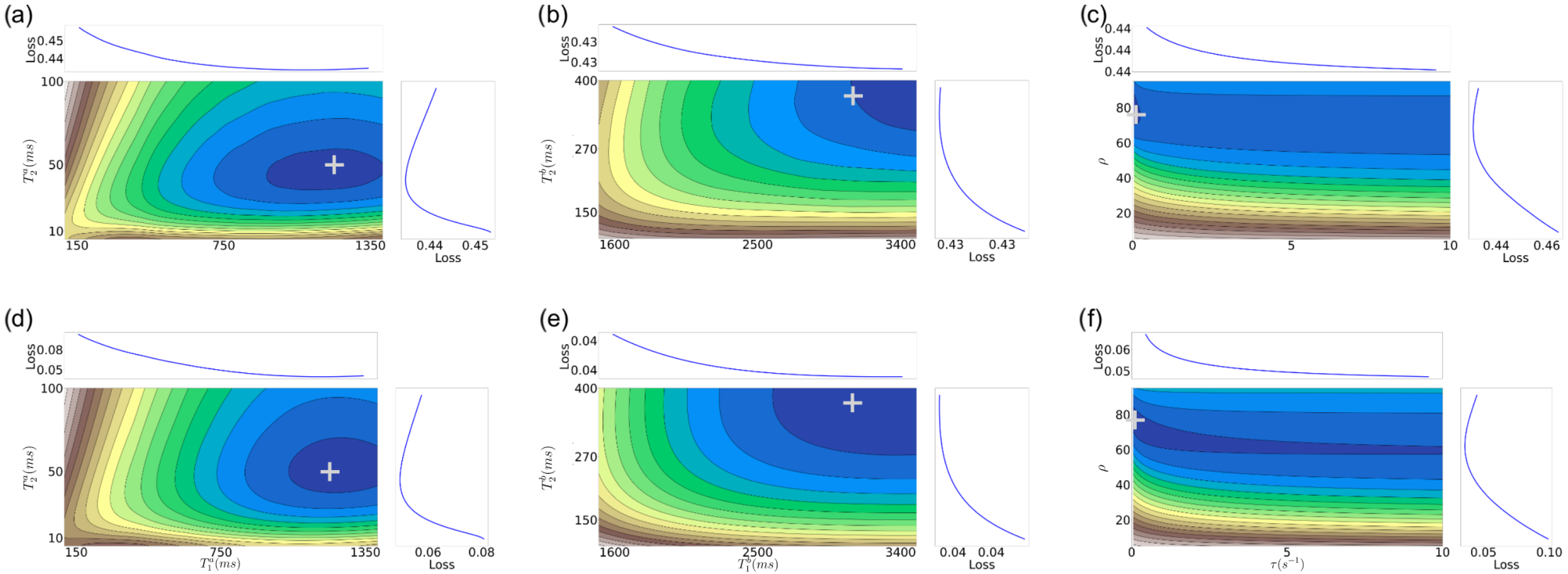


Figure 4. Error function $f(\theta)$ for MRF-X signal X_μ with $\theta \in \Theta^6 = \{T_1^a, T_2^a, T_1^b, T_2^b, \tau, \rho\}$. (**a**) Top-left: $\mu = 1$, $[T_1^a = 1200\,\text{ms}, T_2^a = 50\,\text{ms}]$; (**b**) Top-middle: $\mu = 1$, $[T_1^b = 3100\,\text{ms}, T_2^b = 372\,\text{ms}]$; (**c**) Top-right: $\mu = 1$, $[\tau = 0.1\,\text{s}^{-1}, \rho = 76.6]$; (**d**) Bottom-left: $\mu = 5$, $[T_1^a = 1200\,\text{ms}, T_2^a = 50\,\text{ms}]$; (**e**) Bottom-middle: $\mu = 5$, $[T_1^b = 3100\,\text{ms}, T_2^b = 372\,\text{ms}]$; (**f**) Bottom-right: $\mu = 5$, $[\tau = 0.1\,\text{s}^{-1}, \rho = 76.6]$. Values $[T_1^a = 1200\,\text{ms}, T_2^a = 50\,\text{ms}, T_1^b = 3100\,\text{ms}, T_2^b = 372\,\text{ms}, \tau = 0.1\,\text{s}^{-1}, \rho = 76.6]$ correspond to gray matter tissue properties from [37]. Darker blue contour lines indicate lower f (better fits). "Loss" graphs show f along one axis averaged over the other. The "+" symbol marks the true tissue-property combination in each panel.

4. Methods

From our results in the previous section, it appears feasible in some cases to recover tissue properties in high dimensions by optimizing the error function, without explicitly pre-generating a template dictionary, which is infeasible in this scenario. In the following sections, we empirically evaluate several optimization algorithms to see if this can be realized in practice. First, we briefly review the methods we will evaluate below.

4.1. Broyden–Fletcher–Goldfarb–Shanno

Algorithm The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is a popular quasi-Newtonian method used for numerical optimization. The algorithm approximates the Hessian (second derivative) matrix (B_k) of the objective function (f) at each iteration (k) to find the next iterate:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f_k. \tag{3}$$

Here, ∇f_k denotes the gradient of objective function (f) at the current iterate (x_k), the stepsize α_k is calculated using line search such that α_k satisfies the Wolfe conditions [38], and x_{k+1} is the next iterate. The algorithm is terminated when stopping criteria such as the maximum number of iterations is met. While the BFGS method can solve a wide variety of unconstrained optimization problems efficiently, a limitation for the BFGS algorithm is its lack of bound constraints. This is needed for optimizing the MRF-X objective and is solved by the extension below.

4.2. Limited Memory BFGS Algorithm with Bound Constraints

Limited-memory BFGS with Bound Constraints (L-BFGS-B, Algorithm 1) is a hybrid quasi-Newtonian algorithm that uses gradient projection along with a limited memory BFGS matrix update to solve large-scale nonlinear optimization problems [39]. Similar to trust region methods, L-BFGS-B approximates a quadratic at the current search point (x_k) using the approximate limited memory matrices Y_k and S_k composed of pairs of vectors y_k and s_k as follows:

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= \nabla f_{k+1} - \nabla f_k \end{aligned} \tag{4}$$

Algorithm 1 L-BFGS-B (f, $\mathbf{x_0}$, $[\mathbf{l}, \mathbf{u}]^n$, m, M)

```
Initialize k = 0, x_k = x_0
Initialize Y_k, S_k to store last m gradient and position differences
while k ≤ M and (|x_k − x_{k+1}| > ϵ or ‖∇f(x_k)‖ > ϵ) do
    Estimate quadratic model ϕ_k at x_k using Y_k, S_k
    Calculate Cauchy point x_c along projected gradient direction
    Determine active bounds in [l, u]^n
    Minimize in subspace using L-BFGS Hessian approximation
    Update Y_k, S_k with new gradient and position differences
    k = k + 1
end while
return x_k, f(x_k)
```

As a first step, the Cauchy point x_c is estimated from a quadratic model ϕ_k of the objective function. Variables reaching their bounds are identified as the 'active set' and held constant, reducing the problem's dimensionality. The algorithm then employs subspace minimization, as detailed in [39], focusing on the optimization of variables not in the active set. This subspace minimization differs from the traditional BFGS line search by

maintaining variable bounds, ensuring constraints are respected while pursuing efficient optimization progress.

L-BFGS-B is widely used in practice [40] and is also appropriate for box constraints, which are essential in MRF-X. Further, unlike the BFGS algorithm, the inverse Hessian is approximated using limited memory matrices, which is computationally efficient. The computation complexity of L-BFGS-B grows linearly with number of variables, making it suitable for high-dimensional problems. However, like most gradient-based methods, it is prone to local optima. This issue is partly addressed by the approach below.

4.3. Simplicial Homology Global Optimization

Simplicial Homology Global Optimization (SHGO) is a global optimization algorithm designed to handle complex, high-dimensional black-box optimization problems like MRF reconstruction. The SHGO algorithm has been effectively applied to several practical applications in computed tomography [41], EEG signal extraction [42], and chemical process optimization [43]. Endres et al. [44] demonstrated competitive results against other global optimization strategies such as topographical global optimization [45] and Lc-DISIMPL [46]. Additionally, SHGO is available as part of the scipy [40,44] toolbox. These advantages have motivated us to evaluate SHGO in addressing the MRF reconstruction problem.

SHGO (Algorithm 2) begins by uniformly sampling the feasible region defined by the lower and upper bounds $[\mathbf{l}, \mathbf{u}]^n$ of the search space. Low-discrepancy sampling schemes such as the Sobol sequence [47] are used to decrease the probability of clusters in high-dimensional space. The number of samples N is a hyperparameter based on the dimensionality of the search space. The resulting set of samples $\mathcal{P}$ is then used as the vertices of the simplicial complex $\mathcal{H}$. Triangulation (such as that by Delauney [48]) is then used to connect the edges of the vertices in the simplicial complex.

Algorithm 2 SHGO (f, $[\mathbf{l}, \mathbf{u}]^n$, N, local minimizer L)

$\mathcal{P} = \emptyset$
while $|\mathcal{P}| < N$ **do**
  $\mathcal{X} =$ Generate $N - |\mathcal{P}|$ Sobol sequence points from $\mathbb{R}^n$
  Scale $\mathcal{X}$ to bounds $[\mathbf{l}, \mathbf{u}]^n$
  $\mathcal{P} = \mathcal{P} \cup \mathcal{X}$
end while
Construct simplicial complex $\mathcal{H}$ from $f(\mathcal{P})$
Generate minimizer candidates $\mathcal{M}$ from $\mathcal{H}$; $CS = \emptyset$
for $\mathbf{v} \in \mathcal{M}$ **do**
  $(x, f(x)) = L(\mathbf{v})$
  $CS\{x\} = (x, f(x))$
end for
$x^* = \arg\min CS\{x\}$
return x^*, $f(x^*)$

Each vertex in the simplicial complex $\mathcal{H}$ consists of location, $\mathbf{v}_i, \quad i \in \mathbb{I}^+$, and the corresponding functional value $f(\mathbf{v}_i)$. The direction of each edge is evaluated based on the direction of the vector connecting two vertices on the hypersurface. For example, an edge is directed from vertex $\mathbf{v}_i$ to vertex $\mathbf{v}_j$ iff $f(\mathbf{v}_i) < f(\mathbf{v}_j)$. It can now be observed that if all the edges connected to a vertex are directed away from the vertex, the vertex forms a minimizer of the local region of the set of vertices called the star of the vertex ($st(\mathbf{v}_i)$). Applying Sperner's lemma [49], there is at least a minimizer within the domain of the star of each vertex in the minimizer set $\mathcal{M}$. By using the vertices in the minimizer set and a local optimization routine ($L(\mathbf{v})$) such as L-BFGS-B, the local minima can be estimated, which allows SHGO to return an approximate global minimum. The computation complexity of

the SHGO algorithm without the local optimization routine is exponential in nature, which makes it infeasible for high-dimensional problems without the local optimization routine.

5. Results and Discussion

We now evaluate the utility of optimization algorithms to recover tissue properties from MRF-X data. The task of recovering multiple tissue properties from a single MRF-X scan poses considerable challenges. First, each tissue property varies in its sensitivity to changes in the MRF-X input signal. Second, in clinical settings, only a limited subset of the Fourier space samples are collected (undersampling). This approach inevitably leads to a tradeoff between noise and scan duration. In our study, we simulate this tradeoff by introducing Gaussian noise to the signal, mimicking the noise resulting from the undersampling performed during an actual scan. To the best of our knowledge, our study is the first of its kind to explore the characterization of six-tissue-property MRF (termed as MRF-X) recovery using various nonlinear optimization techniques.

To simulate MRF-X data, we employed a modified version of an MRF-FISP pulse sequence, as described in [21]. Specifically, the MRF-X sequence uses a variable flip angle between 0 and 60° and maintains a constant repetition time (TR) of 6.98 ms. As explained in [37], an inversion pulse is introduced before specific RF excitations at the setting inversion time (TI) [21 ms, 100 ms, 250 ms]. This allows the pulse sequence to be more sensitive to T_1 [50]. We set the SNR $\mu = 5.0$ for these data.

For the implementation, we built a Bloch–McConnell simulator in C++ using the Eigen linear algebra library [51]. All optimization routines were written in Python 3.8 using numpy and scipy. The simulations were run on a server with 2×20-core Intel Xeon CPUs and 384 GB total system memory, custom-built by Puget Systems, Inc. (Auburn, WA, USA). The simulation took $\sim$ 1 min to generate a single signal with a given tissue property combination. Code for the simulation and parameter recovery will be made available on GitHub (https://github.com/).

5.1. MRF Results

We start by validating L-BFGS-B and SHGO with L-BFGS-B as the local minimizer (SHGO+L-BFGS-B), for tissue properties T_1 and T_2 against the standard explicit dictionary template matching technique [9]. To create this dictionary, we generate 10,000 data points from a combination of the properties T_1 and T_2. In this dictionary, the T_1 values range from 500 ms to 3000 ms, while the T_2 values range from 20 ms to 350 ms. For validation, target signals are generated by randomly sampling values multiple times within the same range as the dictionary. To emulate realistic signals obtained at the scanner, we introduce white Gaussian noise with an SNR of 5 into the target signals. Each optimization algorithm then produces estimated (T_1, T_2) values, from which we compute the normalized mean absolute difference (NMAE) as an absolute error relative to the input properties. The results are shown in Figure 5.

From Figure 5, we observe that all methods produce comparable results in this scenario. In particular, the dictionary-free optimization approaches produce excellent results, with error typically in the 2–4% range, in line with direct matching. We also observe that the error in the T_2 tissue property is typically worse than the error in T_1; this is due to the sensitivity difference to the MRF signal between different tissue properties. This experiment shows that optimization approaches can recover signals commensurate with dictionary matching without generating a complete dictionary from MRF signals.

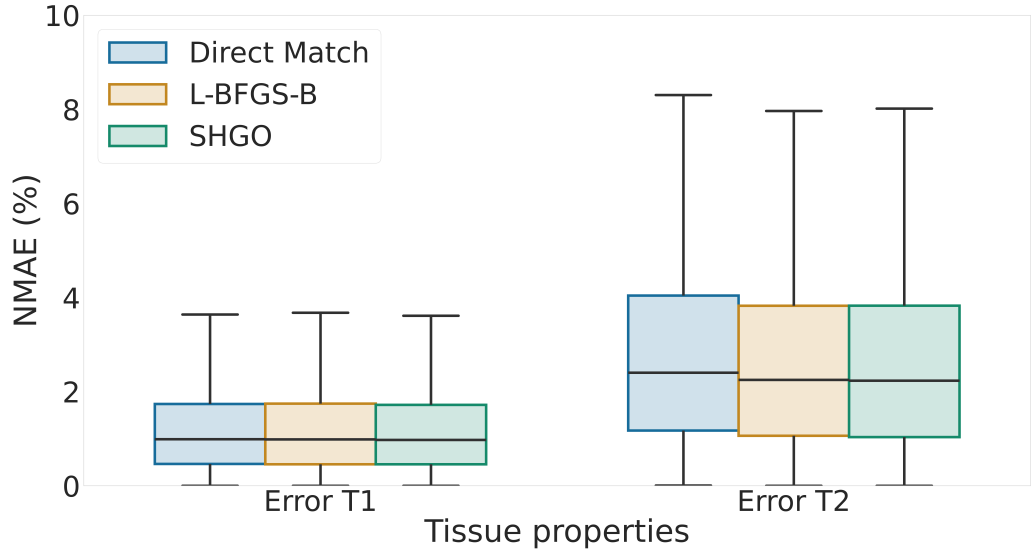


Figure 5. Box-and-whisker plot of normalized mean absolute error as percentage for tissue properties T_1 and T_2. We compare the algorithms direct matching using explicit dictionary with L-BFGS-B and SHGO with L-BFGS-B.

5.2. MRF-X Results

In our experiments, we analyze MRF-X using two distinct scenarios: one with five tissue properties (5D) and the other with six (6D). In the 5D scenario, we set the tissue property τ to be 0. This setting corresponds to a two-compartment model of the tissue, where there is no movement of protons between the compartments. We adopt this approach to highlight the effect of the tissue property τ in estimation of the tissue property ρ and to explore the impact of dimensionality on our optimization procedures. To evaluate the methods, we create two datasets: one consisting of 24,000 target properties in 6D and another consisting of 20,000 properties in 5D as follows. We first sample 1000 points using a regular grid sampling scheme across the tissue property ranges given in Table 2 (left). Then, for each sampled point, we fix all but one of the properties and resample the remaining properties three more times evenly across their ranges. This ensures that each axis is being well sampled, in a tractable manner, for many different values of the other properties. It is similar to Latin hypercube sampling, though modified to use a fixed total sample while still maximizing the sampling of the full hyperspace. We therefore expect our results to generalize well across the whole 5D or 6D spaces.

Table 2. Boundary constraints (minimum and maximum) for L-BFGS-B and SHGO algorithms, along with Gold Standard dictionary's overall range ("Extent") and the maximum possible error. Here, "Extent" denotes the difference between the minimum and maximum values, with actual tissue properties centered within this interval.

	Boundaries		**Gold Std.**	
Property	**Minimum**	**Maximum**	**Extent**	**Max Error.**
$\mathbf{T_1^a}$ (ms)	800	1400	300	0.18
$\mathbf{T_2^a}$ (ms)	20	150	20	0.50
$\mathbf{T_1^b}$ (ms)	1500	2800	300	0.10
$\mathbf{T_2^b}$ (ms)	200	350	20	0.05
$\boldsymbol{\tau}$ (s^{-1})	0.1	5	0.3	1.5
$\boldsymbol{\rho}$ (%)	5	95	6	0.6

In the case of MRF-X, generating an explicit dictionary with the necessary resolution to effectively extract tissue properties becomes intractable. However, we use a modified dictionary-based matching approach that we term **Gold Standard** to establish a performance bound for optimization methods.

The Gold Standard dictionary is designed by restricting the extent of tissue properties used in the dictionary to a constrained region around the actual tissue property value (which would be unknown in reality) in a six-dimensional space. The specifics of each tissue property's extent are detailed in Table 2 (right). To generate a dictionary (separately for each possible target property combination we test), we randomly sample 1000 points within this space. We observe from the "Max error" column of the table that this sampling introduces limited error in the estimates of the individual tissue properties. It is important to emphasize that this approach is not a baseline in that it could not be performed in practice; however, it gives us an idea of the irreducible error (and so an upper bound on the best performance) in this space. That is why we label it "**Gold Standard**".

In Tables 3 and 4, we show tissue property recovery errors for 5D and 6D for two optimization methods: *L-BFGS-B* and *SHGO* combined with *L-BFGS-B* for local search. These methods are initialized randomly from the hypercube defined by the tissue property bounds in Table 2. We also show results for two Gold Standard derivatives: explicit dictionary matching with the Gold Standard dictionary ("Dictionary matching w/Gold Std.") and *L-BFGS-B* with Gold Standard initialization ("L-BFGS-B w/Gold Std."). This method runs *L-BFGS-B* algorithm, initialized randomly within a hypercube determined by the edge length from the 'Extent' column in Table 2 centered around the true tissue property values. Since the two latter methods use the knowledge of the true tissue properties, they serve as upper bounds on potential performance. Our results are shown in terms of NMAE along with standard deviation across the 20,000 points (in 5D) or 24,000 points (in 6D).

Further, we conducted a timing comparison of SHGO and SHGO+L-BFGS-B for the 6D case. In this comparison, we did not use the "Gold Standard" dictionary matching as it is infeasible in practice. For the L-BFGS-B algorithm, our results show that it takes on average 41 s to recover a single tissue property with a standard deviation of 12 s on a 16 core AMD CPU with 32 GB of memory. On the other hand, SHGO+L-BFGS-B takes on average 190 s with a standard deviation of 48 s on a 16 core AMD CPU with 32 GB of memory. This indicates that SHGO+L-BFGS-B is relatively slower than L-BFGS-B for the recovery problem but not as significantly slower as suggested by exponential scaling of the SHGO algorithm. By using the L-BFGS-B as the local minimizer, we significantly offset the computational cost of the SHGO algorithm.

Table 3. Normalized mean absolute errors with standard deviation for five tissue properties. Values are presented as mean ± standard deviation.

TP	Dictionary Match with Gold Std. *	L-BFGS-B with Gold Std. *	L-BFGS-B	SHGO + L-BFGS-B
T_1^a	0.059 ± 0.042	0.123 ± 0.128	0.137 ± 0.136	$\mathbf{0.126 \pm 0.128}$
T_2^a	0.053 ± 0.056	0.228 ± 0.451	0.276 ± 0.588	$\mathbf{0.249 \pm 0.457}$
T_1^b	0.024 ± 0.013	0.105 ± 0.124	$\mathbf{0.124} \pm 0.133$	$0.126 \pm \mathbf{0.130}$
T_2^b	0.058 ± 0.062	0.149 ± 0.175	$0.150 \pm \mathbf{0.163}$	$\mathbf{0.148} \pm 0.168$
ρ	0.045 ± 0.046	0.361 ± 0.772	0.563 ± 1.413	$\mathbf{0.420 \pm 0.830}$

[*] Gold Standard ("Gold Std.") denotes template matching within a region constrained around the solution. L-BFGS-B with Gold Standard ("LBFGS w/Gold Std.") refers to the L-BFGS-B algorithm with initial guess derived from region constrained around the solution. SHGO with L-BFGS-B ("SHGO+L-BFGS-B") is the global optimization SHGO using L-BFGS-B as its local minimizer. Better L-BFGS-B and SHGO+L-BFGS-B error rates are highlighted using bold font.

Table 4. Normalized mean absolute errors with standard deviation for six tissue properties. Values are presented as mean ± standard deviation.

TP	Dictionary Match with Gold Std. *	L-BFGS-B with Gold Std. *	L-BFGS-B	SHGO + L-BFGS-B
$\mathbf{T_1^a}$	0.040 ± 0.041	0.076 ± 0.101	0.159 ± 0.124	$\mathbf{0.137 \pm 0.120}$
$\mathbf{T_2^a}$	0.038 ± 0.048	0.202 ± 0.310	0.342 ± 0.457	$\mathbf{0.284 \pm 0.337}$
$\mathbf{T_1^b}$	0.034 ± 0.026	0.066 ± 0.098	0.187 ± 0.155	$\mathbf{0.169 \pm 0.133}$
$\mathbf{T_2^b}$	0.023 ± 0.018	0.186 ± 0.274	0.293 ± 0.267	$\mathbf{0.278 \pm 0.227}$
$\boldsymbol{\rho}$	0.092 ± 0.144	0.028 ± 1.040	$\mathbf{1.720 \pm 1.413}$	3.860 ± 3.930
$\boldsymbol{\tau}$	0.424 ± 0.830	0.245 ± 0.815	$1.870 \pm \mathbf{3.360}$	$\mathbf{1.530} \pm 4.420$

[*] Gold standard ("Gold Std.") denotes template matching within a region constrained around the solution. L-BFGS-B with Gold Standard ("LBFGS w/Gold Std.") refers to the L-BFGS-B algorithm with initial guess derived from region constrained around the solution. SHGO with L-BFGS-B ("SHGO+L-BFGS-B") is the global optimization SHGO using L-BFGS-B as its local minimizer. Better L-BFGS-B and SHGO+L-BFGS-B error rates are highlighted using bold font.

Looking at the five tissue property results, we observe that (i) there is an irreducible error of 2–6% in each tissue property. This error is present even if we use explicit matching within a small radius of the true tissue property. Thus, we cannot expect practical methods to achieve lower error than this on average across the space of 5D properties. (ii) The error rates of *L-BFGS-B w/Gold Std.* and *L-BFGS-B* are comparable in most cases, except for ρ. This indicates that on average, the *L-BFGS-B* method (initialized randomly) is able to get close to the "Gold Standard" hypercube. However, within the "Gold Standard" hypercube the local gradient may not be smooth and the minimum may not be at the zero of the gradient. This is also illustrated in Figures 3 and 4 for the 6D case and is likely the reason why the *Dictionary match w/Gold Std.* method produces an irreducible error. (iii) We observe that *SHGO + L-BFGS-B* generally produces better results than *L-BFGS-B* on its own. This indicates that this global optimization approach is able to get closer to the "Gold Standard" hypercube than *L-BFGS-B* and pick better minimizer candidates on average. However, since *L-BFGS-B* is the local minimizer, the final error is not less than *L-BFGS-B w/Gold Std*. It is a direction for future work to evaluate if a different minimizer using other signals than just the local gradient could work better with SHGO for this problem. (iv) The ρ tissue property is the hardest to estimate accurately using the local gradient alone. However, given that *Dictionary match w/Gold Std.* achieves an average error rate of 4.5%, it seems likely there are other features beyond the local gradient that could be exploited to find better solutions.

In the six-tissue-property case, again, there is an irreducible error of 2–9% in each tissue property on average. Perhaps surprisingly, *L-BFGS-B w/Gold Std.* is able to produce better results on average than in the five-tissue-property case for T_1^a, T_1^b, and T_2^a. Between *L-BFGS-B* and *SHGO + L-BFGS-B*, the latter is once again the better-performing method. It approaches the results of *L-BFGS-B w/Gold Std.*, though not surprisingly, the error rates are higher than in the 5D case. It is interesting that the error rates for *SHGO + L-BFGS-B*, though higher, are not *substantially* higher (usually less than 5%) other than T_2^b in the 5D case given the substantially larger space being explored. Finally, we observe that the ρ and τ properties produce high errors when estimated by *L-BFGS-B* and *SHGO + L-BFGS-B*. Since the error is lower for the "Gold Standard" methods, it seems likely that the methods are being misdirected into regions where the true property is not present. As we discussed in Section 3, these two properties may not be independent, so there may be multiple pairs of (ρ, τ) solutions that yield similar behavior. To support this claim, we observe that the ρ estimates form both "Gold Standard" methods have either a high error or a wide confidence

interval, and the τ estimates have high irreducible error and large confidence intervals. This indicates it may be inherently difficult to estimate these properties jointly.

6. Conclusions

In this paper, we present a systematic analysis of tissue-property recovery from MRF-X signals. We first visualize the surface of the error function that is explored to recover tissue properties in 2D and 6D cases. The visualization illustrates how the gradient signal changes from 2D to 6D for different target tissue properties and highlights the fact that there is structure that can be exploited but also difficulties caused by plateaus and lack of alignment of the minimums with the target properties at lower SNRs.

Based on this analysis, we show results for two optimization algorithms, *L-BFGS-B* and *SHGO + L-BFGS-B* on 5D and 6D tissue property-recovery problems, as well as two "Gold Standard" methods the illustrate the best that can be achieved. In the 5D case, *SHGO + L-BFGS-B* outperforms *L-BFGS-B* and comes within 10-20% of the error values achieved by *L-BFGS-B* with "Gold Standard" initialization. In the 6D case, *SHGO + L-BFGS-B* again outperforms *L-BFGS-B* for most tissue properties. The errors are comparable to the 5D case for the three tissue properties. However, ρ and τ seem to be hard to estimate in combination, even for the "Gold Standard" approaches.

Key takeaways:

- Estimating six tissue properties (especially ρ and τ) from MRF-X signals is a challenging optimization problem due to complex error surfaces with plateaus and misaligned minima at lower SNRs.
- Because of the lack of standard optimization approaches for this problem, we created a "Gold Standard" which, although impractical in clinical practice, provides a baseline for comparison of our results.
- Our proposed SHGO + L-BFGS-B algorithm comes within 10–20% of L-BFGS-B with "Gold Standard" initialization, demonstrating its effectiveness for practical applications.
- The 6D recovery problem (including both ρ and τ) presents fundamental challenges, with these two parameters being particularly difficult to estimate simultaneously.
- Visualization of the error surfaces reveals an exploitable structure that can guide the development of more effective optimization strategies for MRF-X tissue property recovery.

In future work, we plan to investigate the use of alternative local optimization strategies with SHGO that can take advantage of more than local gradient information, as well as techniques to better estimate the τ and ρ properties in 6D.

To summarize, our study provides an analysis and shows results and potential challenges in recovering multiple tissue properties from a single MRI scan. While simultaneously estimating multiple tissue properties poses considerable technical and fundamental physics challenges, we believe this work is an important step towards developing robust tools for quantitative multi-parametric MRI, advancing its potential as a powerful diagnostic tool in clinical practice.

Author Contributions: Conceptualization was carried out by S.R. and N.N.; Methodology was developed by S.R. and N.N.; Software, Validation, Formal Analysis, Visualization, and Writing–Original Draft Preparation were performed by N.N.; Writing–Review and Editing was undertaken by S.R. and N.N.; Supervision, Project Administration, and Funding Acquisition were handled by S.R. All authors have read and agreed to the published version of the manuscript.

Funding: N.N. and S. R. were supported in part by National Science Foundation award SES-2120972.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Code and data will be made available on github upon publication.

Conflicts of Interest: The authors declare no conflict of interest.The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. The manuscript reflects the opinions of the authors and not those of the National Science Foundation.

References

1. Larsson, H.B.W.; Frederiksen, J.; Petersen, J.; Nordenbo, A.; Zeeberg, I.; Henriksen, O.; Olesen, J. Assessment of demyelination, edema, and gliosis by in vivo determination of T1 and T2 in the brain of patients with acute attack of multiple sclerosis. *Magn. Reson. Med.* **1989**, *11*, 337–348. [CrossRef] [PubMed]
2. Kim, R.J.; Wu, E.; Rafael, A.; Chen, E.L.; Parker, M.A.; Simonetti, O.; Klocke, F.J.; Bonow, R.O.; Judd, R.M. The use of contrast-enhanced magnetic resonance imaging to identify reversible myocardial dysfunction. *N. Engl. J. Med.* **2000**, *343*, 1445–1453. [CrossRef]
3. Jack, C.R.; Bernstein, M.A.; Fox, N.C.; Thompson, P.; Alexander, G.; Harvey, D.; Borowski, B.; Britson, P.J.; Whitwell, J.L.; Ward, C.; et al. The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods. *J. Magn. Reson. Imaging* **2008**, *27*, 685–691. [CrossRef] [PubMed]
4. Stabile, A.; Giganti, F.; Rosenkrantz, A.B.; Taneja, S.S.; Villeirs, G.; Gill, I.S.; Allen, C.; Emberton, M.; Moore, C.M.; Kasivisvanathan, V. Multiparametric MRI for prostate cancer diagnosis: Current status and future directions. *Nat. Rev. Urol.* **2020**, *17*, 41–61. [CrossRef] [PubMed]
5. Johnson, L.M.; Turkbey, B.; Figg, W.D.; Choyke, P.L. Multiparametric MRI in prostate cancer management. *Nat. Rev. Clin. Oncol.* **2014**, *11*, 346–353. [CrossRef]
6. Shen, S.; Koonjoo, N.; Boele, T.; Lu, J.; Waddington, D.E.; Zhang, M.; Rosen, M.S. Enhancing organ and vascular contrast in preclinical ultra-low field MRI using superparamagnetic iron oxide nanoparticles. *Commun. Biol.* **2024**, *7*, 1197. [CrossRef]
7. Brady, A.P. Error and discrepancy in radiology: Inevitable or avoidable? *Insights Imaging* **2017**, *8*, 171–182. [CrossRef]
8. Gulani, V.; Seiberlich, N. Quantitative MRI: Rationale and challenges. In *Advances in Magnetic Resonance Technology and Applications*; Elsevier: Amsterdam, The Netherlands, 2020; Volume 1, pp. xxxvii–li.
9. Ma, D.; Gulani, V.; Seiberlich, N.; Liu, K.; Sunshine, J.L.; Duerk, J.L.; Griswold, M.A. Magnetic resonance fingerprinting. *Nature* **2013**, *495*, 187–192. [CrossRef]
10. Jiang, Y.; Ma, D.; Keenan, K.E.; Stupic, K.F.; Gulani, V.; Griswold, M.A. Repeatability of magnetic resonance fingerprinting T1 and T2 estimates assessed using the ISMRM/NIST MRI system phantom. *Magn. Reson. Med.* **2017**, *78*, 1452–1457. [CrossRef]
11. Crawley, A.P.; Henkelman, R.M. A comparison of one-shot and recovery methods in T1 imaging. *Magn. Reson. Med.* **1988**, *7*, 23–34. [CrossRef]
12. Meiboom, S.; Gill, D. Modified spin-echo method for measuring nuclear relaxation times. *Rev. Sci. Instrum.* **1958**, *29*, 688–691. [CrossRef]
13. Deoni, S.C.L.; Rutt, B.K.; Peters, T.M. Rapid combined T1 and T2 mapping using gradient recalled acquisition in the steady state. *Magn. Reson. Med.* **2003**, *49*, 515–526. [CrossRef] [PubMed]
14. Fram, E.K.; Herfkens, R.J.; Johnson, G.A.; Glover, G.H.; Karis, J.P.; Shimakawa, A.S.; Perkins, T.G.; Pelc, N.J. Rapid calculation of T1 using variable flip angle gradient refocused imaging. *Magn. Reson. Imaging* **1987**, *5*, 201–208. [CrossRef]
15. Welsch, G.H.; Scheffler, K.; Mamisch, T.C.; Hughes, T.; Millington, S.; Deimling, M.; Trattnig, S. Rapid estimation of cartilage T2 based on double echo at steady state (DESS) with 3 Tesla. *Magn. Reson. Med. Off. J. Int. Soc. Magn. Reson. Med.* **2009**, *62*, 544–549. [CrossRef]
16. Bieri, O.; Scheffler, K.; Welsch, G.H.; Trattnig, S.; Mamisch, T.C.; Ganter, C. Quantitative mapping of T2 using partial spoiling. *Magn. Reson. Med.* **2011**, *66*, 410–418. [CrossRef] [PubMed]
17. McGivney, D.F.; Boyacıoğlu, R.; Jiang, Y.; Poorman, M.E.; Seiberlich, N.; Gulani, V.; Keenan, K.E.; Griswold, M.A.; Ma, D. Magnetic resonance fingerprinting review part 2: Technique and directions. *J. Magn. Reson. Imaging* **2020**, *51*, 993–1007. [CrossRef]
18. Bernstein, M.A.; King, K.F.; Zhou, X.J. *Handbook of MRI Pulse Sequences*; Elsevier: Amsterdam, The Netherlands, 2004.
19. Nayak, K.S.; Lee, H.; Hargreaves, B.A.; Hu, B.S. Wideband SSFP: Alternating repetition time balanced steady state free precession with increased band spacing. *Magn. Reson. Med.* **2007**, *58*. [CrossRef]
20. Bloch, F. Nuclear Induction. *Phys. Rev.* **1946**, *70*, 460–474. [CrossRef]
21. Jiang, Y.; Ma, D.; Seiberlich, N.; Gulani, V.; Griswold, M.A. MR fingerprinting using fast imaging with steady state precession (FISP) with spiral readout. *Magn. Reson. Med.* **2015**, *74*, 1621–1631. [CrossRef]
22. Chen, Y.; Panda, A.; Pahwa, S.; Hamilton, J.I.; Dastmalchian, S.; McGivney, D.F.; Ma, D.; Batesole, J.; Seiberlich, N.; Griswold, M.A.; et al. Three-dimensional MR fingerprinting for quantitative breast imaging. *Radiology* **2019**, *290*, 33–40. [CrossRef]

23. Hong, T.; Han, D.; Kim, D.H. Simultaneous estimation of PD, T1, T2, T2*, and ΔB0 using magnetic resonance fingerprinting with background gradient compensation. *Magn. Reson. Med.* **2018**, *81*, 2614–2623. [CrossRef] [PubMed]
24. Cauley, S.F.; Setsompop, K.; Ma, D.; Jiang, Y.; Ye, H.; Adalsteinsson, E.; Griswold, M.A.; Wald, L.L. Fast group matching for MR fingerprinting reconstruction. *Magn. Reson. Med.* **2015**, *74*, 523–528. [CrossRef] [PubMed]
25. Yang, M.; Ma, D.; Jiang, Y.; Hamilton, J.; Seiberlich, N.; Griswold, M.A.; McGivney, D. Low rank approximation methods for MR fingerprinting with large scale dictionaries. *Magn. Reson. Med.* **2018**, *79*, 2392–2400. [CrossRef] [PubMed]
26. McGivney, D.F.; Pierre, E.; Ma, D.; Jiang, Y.; Saybasili, H.; Gulani, V.; Griswold, M.A. SVD Compression for Magnetic Resonance Fingerprinting in the Time Domain. *IEEE Trans. Med Imaging* **2014**, *33*, 2311–2322. [CrossRef]
27. Hamilton, J.I.; Griswold, M.A.; Seiberlich, N. MR Fingerprinting with chemical exchange (MRF-X) to quantify subvoxel T1 and extracellular volume fraction. *J. Cardiovasc. Magn. Reson.* **2015**, *17*, 1–3. [CrossRef]
28. Deoni, S.C.; Matthews, L.; Kolind, S.H. One component? Two components? Three? The effect of including a nonexchanging "free" water component in multicomponent driven equilibrium single pulse observation of T1 and T2. *Magn. Reson. Med.* **2013**, *70*, 147–154. [CrossRef]
29. McConnell, H.M. Reaction rates by nuclear magnetic resonance. *J. Chem. Phys.* **1958**, *28*, 430–431. [CrossRef]
30. Donahue, K.M.; Weisskoff, R.M.; Burstein, D. Water diffusion and exchange as they influence contrast enhancement. *J. Magn. Reson. Imaging* **1997**, *7*, 102–110. [CrossRef]
31. Lundervold, A.S.; Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Z. Für Med. Phys.* **2019**, *29*, 102–127. [CrossRef]
32. Yang, M.; Jiang, Y.; Ma, D.; Mehta, B.B.; Griswold, M.A. Game of learning Bloch equation simulations for MR fingerprinting. *arXiv* **2020**, arXiv:2004.02270.
33. Hamilton, J.I.; Seiberlich, N. Machine learning for rapid magnetic resonance fingerprinting tissue property quantification. *Proc. IEEE* **2019**, *108*, 69–85. [CrossRef] [PubMed]
34. Cohen, O.; Zhu, B.; Rosen, M.S. MR fingerprinting deep reconstruction network (DRONE). *Magn. Reson. Med.* **2018**, *80*, 885–894. [CrossRef] [PubMed]
35. Fang, Z.; Chen, Y.; Liu, M.; Xiang, L.; Zhang, Q.; Wang, Q.; Lin, W.; Shen, D. Deep learning for fast and spatially constrained tissue quantification from highly accelerated data in magnetic resonance fingerprinting. *IEEE Trans. Med Imaging* **2019**, *38*, 2364–2374. [CrossRef]
36. Buonincontri, G.; Kurzawski, J.W.; Kaggie, J.D.; Matys, T.; Gallagher, F.A.; Cencini, M.; Donatelli, G.; Cecchi, P.; Cosottini, M.; Martini, N.; et al. Three dimensional MRF obtains highly repeatable and reproducible multi-parametric estimations in the healthy human brain at 1.5 T and 3T. *Neuroimage* **2021**, *226*, 117573. [CrossRef] [PubMed]
37. Hamilton, J.I.; Deshmane, A.; Griswold, M.; Seiberlich, N. MR fingerprinting with chemical exchange (MRF-X) for in vivo multi-compartment relaxation and exchange rate mapping. In Proceedings of the 24th Annual Meeting and Exhibition of the International Society for Magnetic Resonance in Medicine (ISMRM 2016), Singapore, 7–13 May 2016.
38. Nocedal, J.; Wright, S.J. *Numerical Optimization*; Springer: Berlin/Heidelberg, Germany, 1999.
39. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [CrossRef]
40. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
41. Lochschmidt, M.E.; Gassenhuber, M.; Riederer, I.; Hammel, J.; Birnbacher, L.; Busse, M.; Boeckh-Behrens, T.; Ikenberg, B.; Wunderlich, S.; Liesche-Starnecker, F.; et al. Five material tissue decomposition by dual energy computed tomography. *Sci. Rep.* **2022**, *12*, 17117. [CrossRef]
42. Roshdy, A.; Al Kork, S.; Beyrouthy, T.; Nait-ali, A. Simplicial Homology Global Optimization of EEG Signal Extraction for Emotion Recognition. *Robotics* **2023**, *12*, 99. [CrossRef]
43. Soritz, S.; Moser, D.; Gruber-Wölfler, H. Comparison of Derivative-Free Algorithms for their Applicability in Self-Optimization of Chemical Processes. *Chem.-Methods* **2022**, *2*, e202100091. [CrossRef]
44. Endres, S.C.; Sandrock, C.; Focke, W.W. A simplicial homology algorithm for Lipschitz optimisation. *J. Glob. Optim.* **2018**, *72*, 181–217. [CrossRef]
45. Törn, A.; Viitanen, S. Topographical global optimization. In *Recent Advances in Global Optimization*; Princeton University Press: Princeton, NJ, USA, 1992; pp. 384–398.
46. Paulavičius, R.; Žilinskas, J. Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints. *Optim. Lett.* **2016**, *10*, 237–246. [CrossRef]
47. Sobol', I.M. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Mat. I Mat. Fiz.* **1967**, *7*, 784–802. [CrossRef]
48. Lee, D.T.; Schachter, B.J. Two algorithms for constructing a Delaunay triangulation. *Int. J. Comput. Inf. Sci.* **1980**, *9*, 219–242. [CrossRef]

49. Cohen, D.I. On the Sperner lemma. *J. Comb. Theory* **1967**, *2*, 585–587. [CrossRef]
50. Panda, A.; Mehta, B.B.; Coppo, S.; Jiang, Y.; Ma, D.; Seiberlich, N.; Griswold, M.A.; Gulani, V. Magnetic resonance fingerprinting–an overview. *Curr. Opin. Biomed. Eng.* **2017**, *3*, 56–66. [CrossRef]
51. Guennebaud, G.; Jacob, B.; He, C.-P.; Ferro, D.G.; Steiner, B.; Luitz, D.J.; Margaritis, K.A.; Brun, G.; Zoppitelli, P.; Garg, R.; et al. *Eigen v3*; Eigen Library Developers: Lausanne, Switzerland, 2010. Available online: http://eigen.tuxfamily.org (accessed on 4 May 2025).

Journal of *Imaging*

MDPI

Article

The Creation of Artificial Data for Training a Neural Network Using the Example of a Conveyor Production Line for Flooring

Alexey Zaripov, Roman Kulshin and Anatoly Sidorov *

Department of Data Processing Automation, Tomsk State University of Control Systems and Radioelectronics, 634050 Tomsk, Russia; aleksei.v.zaripov@tusur.ru (A.Z.); roman.s.kulshin@tusur.ru (R.K.)
* Correspondence: anatolii.a.sidorov@tusur.ru

Abstract: This work is dedicated to the development of a system for generating artificial data for training neural networks used within a conveyor-based technology framework. It presents an overview of the application areas of computer vision (CV) and establishes that traditional methods of data collection and annotation—such as video recording and manual image labeling—are associated with high time and financial costs, which limits their efficiency. In this context, synthetic data represents an alternative capable of significantly reducing the time and financial expenses involved in forming training datasets. Modern methods for generating synthetic images using various tools—from game engines to generative neural networks—are reviewed. As a tool-platform solution, the concept of digital twins for simulating technological processes was considered, within which synthetic data is utilized. Based on the review findings, a generalized model for synthetic data generation was proposed and tested on the example of quality control for floor coverings on a conveyor line. The developed system provided the generation of photorealistic and diverse images suitable for training neural network models. A comparative analysis showed that the YOLOv8 model trained on synthetic data significantly outperformed the model trained on real images: the mAP50 metric reached 0.95 versus 0.36, respectively. This result demonstrates the high adequacy of the model built on the synthetic dataset and highlights the potential of using synthetic data to improve the quality of computer vision models when access to real data is limited.

Keywords: data generation; neural network; synthetic data; computer vision; YOLO; Unity; conveyor; laminate; defect

Academic Editors: Miguel Angel Guevara Lopez, Luís Gonzaga Mendes Magalhães and Edel Bartolo Garcia Reyes

Received: 17 April 2025
Revised: 12 May 2025
Accepted: 17 May 2025
Published: 20 May 2025

Citation: Zaripov, A.; Kulshin, R.; Sidorov, A. The Creation of Artificial Data for Training a Neural Network Using the Example of a Conveyor Production Line for Flooring. *J. Imaging* **2025**, *11*, 168. https://doi.org/10.3390/jimaging11050168

1. Introduction

In the implementation of a wide range of industrial processes based on multi-stage transformations of raw materials into finished products, there arises a need for quality control of intermediate and/or final results. This need is expressed through the identification and rejection of objects that do not meet specified parameters indicating compliance with certain standards, regulations, or norms. Typically, this task is performed through human visual inspection, which involves several issues:

- low speed of evaluation and classification (compliant/non-compliant with quality parameters) due to the physiological limitations of the human visual system;
- subjectivity of perception, which may lead to missed detection of substandard objects or incorrect classification caused by factors such as the quality inspector's qualifications, fatigue, loss of concentration, etc.;

- costliness of this stage in the production process, which does not directly contribute to the added value of the product and includes, among other things, the need to compensate specialized personnel.

To address these challenges, various digital solutions based on computer vision (CV) systems are increasingly being adopted. These systems are being integrated across multiple industries and specific production tasks, including: traffic monitoring [1–4], metallurgy [5–8], perimeter security and safety compliance [9–12], agriculture [13–15], and others. The technology enables automation of numerous routine and attention-intensive tasks.

For example, in [16], potential applications of CV technology are discussed in the context of detecting and classifying diseases in various medical images, such as ultrasound scans and microscope images of tissue samples.

Computer vision can also be applied in the printing industry to enhance the effectiveness of quality control over printing equipment, ultimately reducing the cost of the final product. During the technological process, components such as engraved printing cylinders are produced, which are used to apply images or text onto paper. In [17], deep neural networks were trained to detect defects (dents, scratches, inclusions, bends, misalignments, excessive, faded or missing prints, and color errors resulting from engraving). The training data consisted of photographs of printing machine components, classified by operators as either defective or defect-free. During testing, the model achieved an accuracy of 97.85% for true negative results, and 99.01% for true positives—clearly indicating a high level of performance for the given task.

Building effective computer vision (CV) systems requires a preliminary data collection stage, which includes obtaining images of relevant objects along with their coordinates within the image. However, in many fields, acquiring a diverse and timely dataset is challenging, as technological and other processes may not allow for the reproduction of rare, hazardous, or economically inefficient scenarios. Additionally, manual image annotation is subject to human error, which can negatively affect the quality of neural network training.

Due to these limitations, obtaining uniform datasets becomes problematic, which reduces the generalization ability of neural networks and, consequently, their accuracy. One of the solutions to the data scarcity issue is the generation of artificial images using various methods. Currently, there are numerous approaches and implementations of data generators tailored to different subject domains.

2. The Concept of Data Generation

The problem of automating the data collection process is not new, and numerous studies have been conducted to address it. For example, the study [18] presents a review of image generation methods aimed at producing data for machine-learning systems, focusing on photorealistic rendering techniques such as path and ray tracing.

In article [19], various synthetic data generation methods for neural networks are described, discussing different types of generators—such as generative neural networks and simulators—that can be used to create images.

Some researchers [20] describe the process of generating artificial data by simulating UAV (unmanned aerial vehicle) behavior for detection and classification tasks using the video game GTA V (Grand Theft Auto V). However, current video games offer limited capabilities for data generation and do not support a wide range of diverse scenarios. To implement more atypical use cases not supported by video games, other software tools are used, including 3D modeling software such as Blender [21], ZBrush [22], and 3Ds Max [23].

For instance, Blender was used to build a synthetic data generator for detecting pipeline defects using UAVs [24]. The system provides a flexible dataset of photorealistic images and allows for customizable usage scenarios.

The article [25] proposes a method for generating synthetic images of defects, specifically chips, for use in various industrial applications. As an example, the study uses a turbocharger component. The proposed generator relies on procedurally generated textures applied to the object of interest to increase dataset diversity. Defects are simulated by cutting out randomly shaped segments from the 3D mesh, allowing for a wide range of potential damages. Test results showed that a model trained solely on synthetic data outperformed a model trained only on real images of turbocharger chips.

However, the Blender-based approach has limitations—it cannot produce a large number of frames per second. Even with a decent rendering speed of 1 fps, generating a one-minute video would take about 30 min. For faster data generation with only a slight reduction in image quality, game engines like Unreal Engine, Unity, and CryEngine are often used.

Unity, for example, includes the Unity Perception package, which offers tools for automated annotation of synthetic data in its native SOLO format. In [26], Unity and Unity Perception were used to generate training data for a neural network tasked with detecting manufactured metal bolts on a conveyor belt. This solution enabled data generation at up to 6 fps—significantly faster than Blender—but still did not fully leverage Unity's potential.

The authors of [27] conducted an experiment to generate industrial safety data. A construction site with scaffolding was simulated using a game engine, and human behavior was animated using skeletal animation tools. Neural network models trained on synthetic data mixed with real samples achieved the highest performance.

The study [28] proposed a solution for generating data for traffic sign detection. Unity Perception was used to render 2D images from 3D models, separating them into detection objects, noise elements, and background layers. Although models trained only on synthetic data performed worse than those trained on real images, the time saved on data annotation was considerable. Moreover, a model trained on a mixed dataset outperformed both others, highlighting the value of synthetic data in machine learning.

CAD (computer-aided design) systems can also be used among the tools for working with 3D graphics, as, for example, in article [18]. Here, various 3D parts with diverse textures, viewpoints, and lighting conditions were used to produce a highly variable training set. Experiments demonstrated that neural networks trained on synthetic data were able to recognize additive components with high accuracy.

The authors of [29] consider a technique for generating synthetic images using CGI (computer-generated imagery) graphics based on thin shell simulation. This approach accurately replicates the photorealistic behavior of textile materials, including wrinkles and folds, which is important for tasks related to clothing structure analysis or other soft surfaces. Experimental results showed an 8–10% increase in prediction accuracy for models trained with synthetic data compared to those trained exclusively on real images—demonstrating the utility of CGI-based synthetic data in enriching computer vision datasets.

Another notable advancement is in the use of synthetic composite images (SCI)—real photographs digitally manipulated or augmented with elements not originally present. For example, the Super real dataset presented in paper [30] includes around 6000 Skins created for segmentation tasks. In this dataset, 3D models of people in various poses are overlaid onto real-world backgrounds. Experiments showed that neural networks trained on SCIs achieved higher segmentation accuracy compared to models trained solely on real images. The best performance was achieved by a model with an upsampling layer that

increased image resolution before segmentation. This model was trained on a mixed dataset, combining both real and synthetic images, resulting in improved segmentation quality.

Study [31] also investigates composite image generation, where CAD models with various viewpoints and lighting conditions are superimposed on background images to increase dataset variability. While models trained solely on synthetic images generally underperformed compared to those trained on real data, freezing the feature extractor significantly improved their performance on real test data.

Neural networks themselves are also used to generate synthetic data. One such method involves Variational Autoencoders (VAEs), which include two key components: an encoder and a decoder. VAEs generate new images by learning the underlying data distribution, encoding input images into latent space and decoding them to generate new samples. In [32], a VAE was used to generate random faces. However, the resulting images were often blurry, particularly at high resolutions.

Another technique uses generative adversarial networks (GANs), consisting of a generator and a discriminator that compete against each other. The generator creates synthetic images, while the discriminator evaluates their authenticity. This adversarial process continues until the generator produces high-quality, realistic images [33].

In paper [34], a GAN-based method is proposed for training a neural network to detect and classify people. A "synthesizer" network generates images, which are then evaluated by a "target" network. A discriminator, trained on real images, helps the synthesizer avoid obvious artifacts and improve image realism.

In study [35], GANs are used to expand a dataset aimed at image segmentation. The generated images train a semantic segmentation network responsible for identifying defects. The method improves the model's robustness to different defect types and lighting conditions, making it more effective in real industrial environments. The synthetic data's diversity reduced overfitting to limited real-world datasets.

Each approach addresses the data shortage challenge in its own way, with its own strengths and limitations. For instance, 3D editor-based generation is relatively slow due to low rendering speed. The GTA V-based method does not support data generation for industrial processes, which is a significant drawback, but it allows real-time UAV simulation. Neural network-based methods such as VAE and GAN also have limitations—they require real data to train the models before generation can begin.

3. The Concept of Digital Twins

A digital twin is a dynamic virtual copy of a real object, process, system, or environment. It replicates not only the appearance but also the key properties of its physical counterpart, enabling detailed analysis, simulation, and forecasting of its behavior. The use of digital twins improves process efficiency, reduces maintenance costs, predicts potential failures, and enhances product quality [36].

In paper [37], the authors presented a classification of digital twins based on the level of integration:

- digital model: a static 3D representation without any connection to the real object;
- digital shadow: a model that is updated based on incoming data, but the connection is one-way—from the physical object to the model;
- digital twin: provides a two-way connection with the physical object, allowing not only data acquisition but also real-time simulation of changes.

Digital twins are used to solve various tasks related to testing and predicting the behavior of real objects.

For example, in [38], digital twins are discussed in the context of optimizing construction processes, reducing costs, and increasing efficiency through the integration of building information modeling (BIM) and IoT technologies.

In [39], the prospects of Industry 4.0—combining IoT and digital twins—are examined in the context of education. The study concludes that it is necessary to implement digital twin research programs, both conceptually and methodologically, with practical application in mind.

Study [40] explores the concept of digital twins in agriculture. Based on the analysis of existing definitions, a typology of digital twins was proposed according to lifecycle stages, including the following categories:

- Imaginary digital twin: a conceptual digital model representing an object that does not yet exist physically;
- Monitoring digital twin: a digital representation of the current state, dynamics, and trajectory of a real physical object;
- Predictive digital twin: a digital projection of possible future states and behaviors of physical objects, based on predictive analytics, including statistical forecasting, simulation, and machine learning;
- Prescriptive digital twin: an intelligent digital model capable of recommending corrective and preventive measures to optimize the operation of real-world objects. These recommendations are usually based on optimization algorithms and expert heuristics;
- Autonomous digital twin: a digital twin with autonomous functions, capable of fully controlling the behavior of physical objects without human intervention, either locally or remotely;
- Recollection digital twin: a digital representation containing the complete history of a physical object that no longer exists in reality.

Additionally, a hardware–software solution was developed in the study to implement and experimentally test the concept of digital twins in agriculture.

Study [41] provides a comprehensive review of the application of digital twin technology in the context of intelligent electric vehicles. The research focuses on optimizing electric transport operations through the use of digital twins for real-time monitoring and prediction of the state of key vehicle components and systems. Digital twin technology improves the efficiency of electric vehicle operation by enabling better use of energy and material resources. In particular, timely identification of potential failures and optimization of operational parameters not only extends component lifespan, but also significantly reduces the environmental impact of transport.

In paper [42], the digital twin concept is used to optimize pig farming in agriculture. A "pig twin" model was developed to track growth under varying conditions and optimize feeding. The pig digital twin is currently under development and will be used to explore the potential of a closed-loop control system at the Industry 4.0 level.

Unlike agriculture, which focuses on adaptation to natural conditions and biological factors, industry demands higher precision, reliability, and integration of digital solutions into complex production chains. In this context, digital twins are a key tool in digital transformation, enabling continuous process optimization, predictive maintenance, and cost reduction.

For creating digital twins of conveyor-based production, game engines are often used. For instance, the company Prespective developed software integrated with the Unity game engine, allowing rapid design of conveyor lines. Using this solution, an automotive assembly process was recreated, enabling assessment of line performance under equipment failure conditions for incident prioritization. Moreover, the system provides operators with detailed order information necessary for proper vehicle assembly [43].

Study [44] proposes a fault diagnosis methodology for permanent magnet synchronous motors used in coal conveyors, based on digital twin technology. The proposed approach integrates the digital twin concept with an optimized random forest algorithm. A bidirectional data transmission system was implemented to synchronize the physical object and its virtual copy in real time. Simulation results, confirmed by experimental data, showed a diagnostic accuracy of 98.2%, which indicates that 98.2% of failures were correctly predicted. This study highlights the potential of digital twins for motor condition monitoring and emphasizes further integration of fault diagnosis with 3D visual control of equipment operation.

Study [45] presents a comparative analysis of robotic arm digital twin development using two software platforms: Unity and Gazebo. The study evaluated the performance of these environments in creating dynamic digital models. The development was based on a unified physical setup and communication layer, ensuring the objectivity and accuracy of the comparison.

The results showed that Unity has advantages in simulation accuracy and lower response latency, making it optimal for applications requiring high visualization precision and fast data processing. On the other hand, Gazebo offers faster integration with the Robot Operating System (ROS), making it preferable for low-budget robotics and automation projects.

Digital twins are also suitable for generating synthetic data, as they allow simulation of system operation under extreme conditions without the need to reproduce dangerous scenarios.

Study [46] is dedicated to the digital twin of a wind turbine energy conversion system. A hybrid model was developed to generate synthetic data for fault diagnosis.

In the context of generating synthetic images, basic-level digital twins—digital models—are sufficient, since a real-world connection is not necessary to obtain high-quality, realistic data for neural network training.

For instance, Boeing used a digital model of its aircraft for an AR (Augmented Reality) inspection application, generating over 100,000 synthetic images to train machine-learning algorithms [47].

To achieve the goal of synthetic data generation, a simple digital twin of the studied object must be created. This approach allows for the rapid development of training datasets using digital models that serve as a specific implementation of the digital twin concept.

4. Generalized Model of the Synthetic Data Generator

The task of identification involves detecting the object of interest that needs to be recognized and classified by the computer vision (CV) system. For example, in quality control of brick production, the object of interest would be brick defects. If the task is to detect vehicles using UAVs, the objects of interest would be various types of vehicles.

To address identification tasks within a conveyor-based manufacturing process, a generalized model for generator formation is proposed. This model consists of a set of interconnected tasks and simulates the production process:

- Creation of a digital twin (digital model) of the conveyor line (generation of 3D models of the conveyor belt; generation of 3D models of the conveyor sidewalls; texturing of the belt and sidewall models);
- Creation of the object of interest (generation of 3D models of the object of interest; texturing of the object models; development of algorithms for modifying the object of interest);
- Simulation of the real technological process (development of algorithms for the movement of objects of interest along the digital twin of the conveyor line; development

of algorithms for modifying the digital twin and applying effects; development of algorithms for changing the production camera's viewing angle).

The step-by-step implementation of the above-mentioned tasks enables the creation of a synthetic data generator framework suitable for any conveyor-based manufacturing process. As an example, a specific case of modeling laminate production will be considered further in this work.

Within the scope of the generalized task list, specialized software was developed, with its component diagram shown in Figure 1.

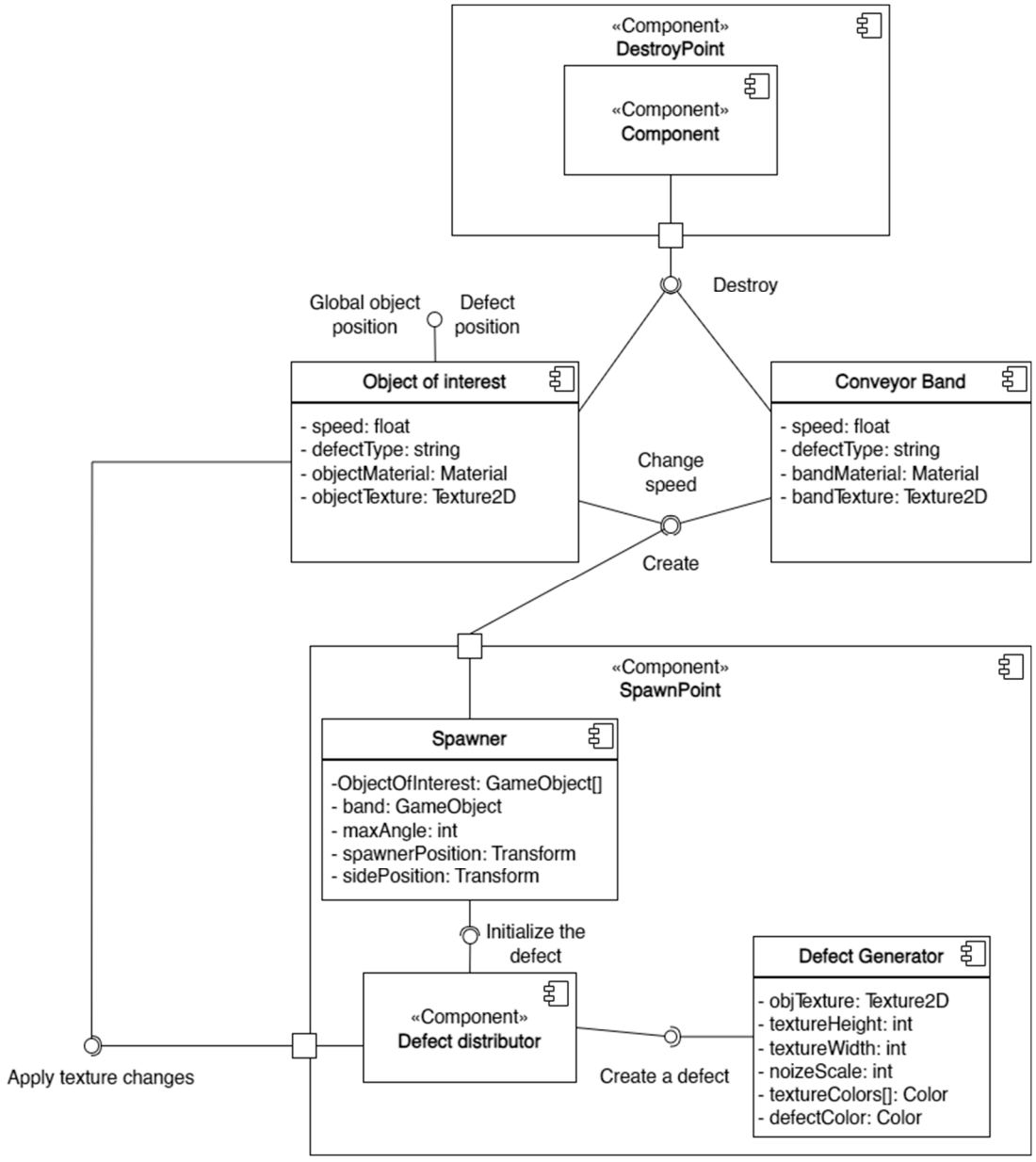


Figure 1. Diagram of software components.

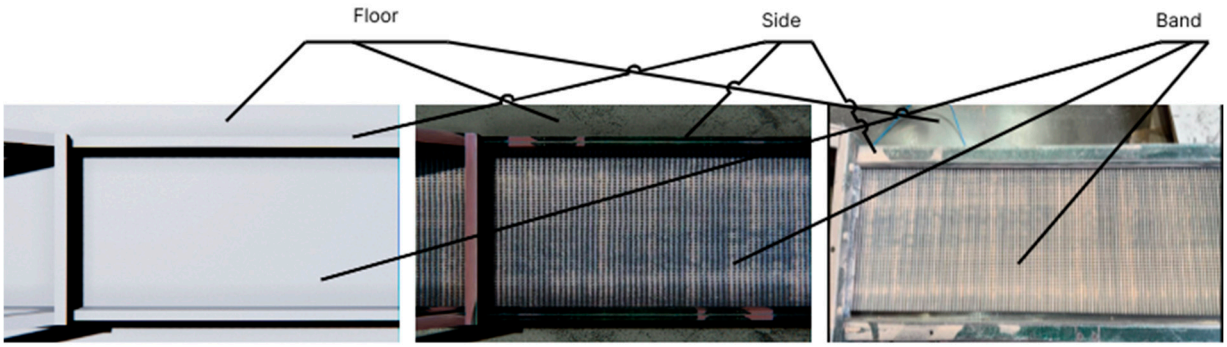
Scene generation was carried out through the operation of eight main components responsible for the creation and destruction of objects of interest, defect overlay, simulation of conveyor line operation, and expansion of the image dataset.

Situations are generated by the work of 8 main components of the scene responsible for the creation and destruction of objects of interest, the imposition of defects, the simulation of the conveyor line, and the expansion of the image sample.

4.1. Formation of the Digital Twin of the Conveyor Line

The system is implemented using the Unity game engine, which enables real-time scene rendering, along with its associated programming language, C#, and High-Definition Render Pipeline (HDRP) technology, which provides high-quality image generation.

Since the generalized task list involves the creation of a digital model, a representation of the conveyor was constructed using engine primitives (Figure 2a), specifically parallelepipeds. High-resolution textures were then applied to this representation (Figure 2b), which were created based on a real-life prototype—a conveyor line from a flooring production facility (Figure 2c).



(**a**) (**b**) (**c**)

Figure 2. Conveyor line: (**a**) representation of primitives; (**b**) textured model; (**c**) real prototype.

The result of completing the first task from the generalized model was a generated 3D model of the conveyor, which was used to simulate the production process.

4.2. Formation of an Object of Interest

The object of interest in the given process is a laminate board, which must later be detected by the CV system. Accordingly, it is important to reflect the maximum number of object variations within the synthetic dataset. To create the objects of interest, ten high-resolution laminate board textures and their corresponding normal maps were developed. The number of textures was chosen to cover the main visual and textural variations commonly found in real-world conditions. This amount is sufficient to generate a large enough volume of synthetic data required for proper training of the CV system. The textures were subsequently applied to 3D parallelepiped objects.

Next, algorithms for modifying the boards were developed. To increase the diversity of the dataset, it was decided to expand the number of objects of interest by applying defects. For this purpose, two components were created: the Defect Distributor and the Defect Generator (Figure 1).

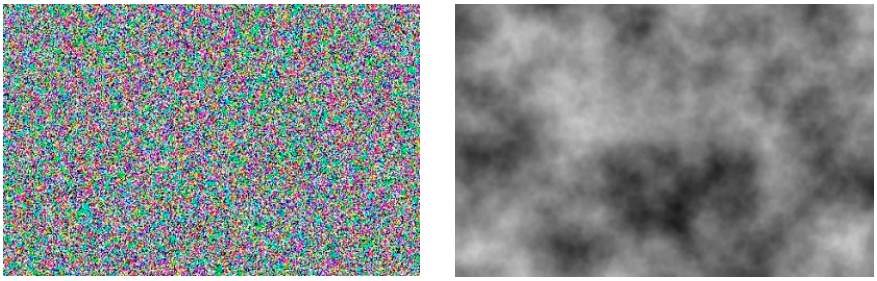
The Defect Distributor manages the defect application process. It performs the following tasks:

- Randomly determines whether a defect should be created;
- Modifies the object to allow proper defect application (adds annotation markers, creates additional empty objects);
- Randomly selects a defect from the available list;
- Transfers the prepared object to the defect application procedure.
- The Defect Generator is responsible for creating three types of defects:
- Print defect—an area of the printed pattern on the board that differs in color and texture from the main surface;
- Glue spot—a light gray spot on the outer surface of the laminate, caused by glue seeping from the adhesive layer beneath;

- Corner chipping—chips along the edges of the board, either across its entire length or in partial sections.

The defect selection algorithm is based on generating a random number from 0 to 1, which helps to distribute defects proportionally and maintain class balance in the training dataset.

The print defect and glue spot are generated using an algorithm based on Perlin noise, implemented with Unity's built-in tools. Perlin noise is a type of gradient noise that uses random number generation. Unlike uniformly distributed noise (Figure 3a), where each new value can vary sharply from the previous one, Perlin noise (Figure 3b) ensures smooth transitions between values.



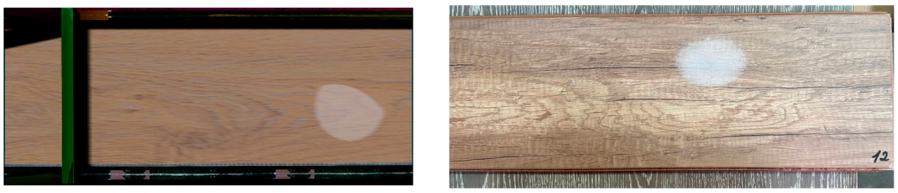
(**a**) (**b**)

Figure 3. Noise visualization: (**a**) uniformly distributed noise; (**b**) Perlin noise.

Using Perlin noise makes it possible to create a smooth random pattern texture, which enhances realism and is applied in various fields such as medicine [48], information security [49], and terrain generation [50]. Perlin noise was chosen as the basis for the spot generation algorithm because this type of procedural generation allows for the creation of smooth, pseudo-random textures on the surface of the object of interest.

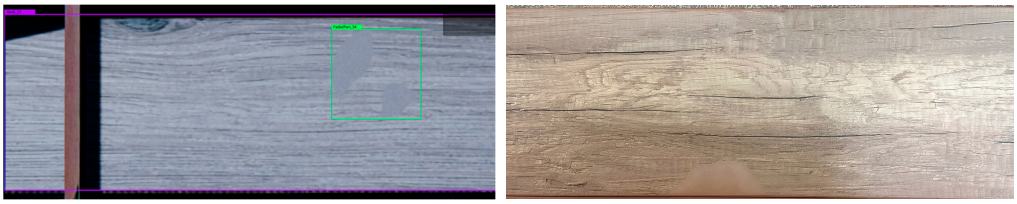
Initially, a spot area is selected on the laminate board with a random size, which does not exceed the distance from the center of the stain to the nearest edge of the object. Once the spot area is defined, the noise value of each texture pixel is calculated within the range [0, 1]. Then, for values below a specified threshold, the pixel color is set to the predefined spot color.

To prevent repetition of spots generated using noise, the seed value must be changed to a randomly assigned number. The results of the spot generation process are shown in Figures 4 and 5. The proposed algorithm enables the generation of pseudo-random texture spots on the board surface. This significantly expands the sample of objects of interest and facilitates training of the neural network on a wider variety of data. As a result, the training quality improves, which positively affects subsequent object identification in manufacturing processes.



(**a**) (**b**)

Figure 4. An example of the formation of a "glue spot" defect: (**a**) synthetic image; (**b**) real image.

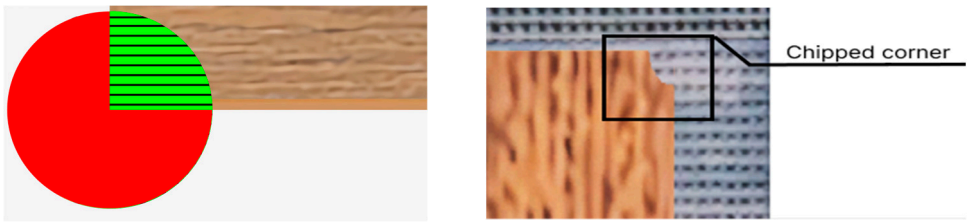


(**a**) (**b**)

Figure 5. An example of the formation of a "print defect": (**a**) synthetic image; (**b**)real image.

The algorithm responsible for corner chip generation is based on altering the transparency of a selected pixel—specifically, by reducing its alpha channel value from 100 to 0. Initially, the algorithm randomly determines the number of corner chips (from 1 to 4), corresponding to the number of board corners. Then, based on this number, corners to be "chipped" are randomly selected, and the defect creation process began.

Once a corner is selected for chipping, the chip size is randomly determined, represented as the radius of a circle centered at the extreme pixel of the corner. For example, if the board texture is sized 2048 × 512, the chip centers can be pixels at coordinates (0, 0), (0, 512), (2048, 0), and (2048, 512). The chip itself is a sector that extends into the board area from a circle positioned at the texture's edge. In Figure 6a, a circle with a radius of 100 pixels is shown at a board corner with dimensions of 4096 × 980. The shaded area indicates the circle sector where pixel alpha channel values will be modified, while the unshaded area remains unaffected.



(**a**) (**b**)

Figure 6. Demonstration of chip generation: (**a**) is the sector of the circle defining the chip; (**b**) is the result of chip formation.

Next, the pixels within the chipped area must be altered accordingly. To do this—as with the algorithms for generating "glue spots" and "print defects"—the circle equation must be used.

If a pixel with coordinates x and y is located within a circle of *radius*, its alpha channel value is set to 0; otherwise, the pixel remains unchanged. The result is shown in Figure 6b. For clarity, the defect was highlighted with a rectangle with the inscription in the picture.

As a result of implementing the algorithms described above, the number of board variations increased significantly from an initial set of 10. Consequently, such a large dataset of diverse objects will allow the neural network to be treated without using identical images, which will positively impact its generalization ability and detection accuracy.

4.3. Simulation of a Real Technological Process

To simulate conveyor-based production, the components "object of interest", "conveyor belt", and "spawner" were created (see Figure 1). The "spawner" component is responsible for generating variations of the object of interest and randomly adding them to the scene along with the conveyor belt component. Before placing the object on the scene, its position is randomly determined based on the dimensions of the conveyor belt. The

offset of the object of interest from the center of the conveyor belt is set within the range $[-distance, distance]$, where the $distance$ is calculated using the following formula:

$$distance = |pos_1 - pos_2| \quad (1)$$

where pos_1 is the position of the "spawner" object; pos_2 is the position of the "Side" object.

After the object's offset is selected, its rotation angle in space is also randomly chosen from the range $[-max\_angle, max\_angle]$ to prevent it from going out of bounds or colliding with other objects in the scene. The maximum rotation angle is calculated using the following formula:

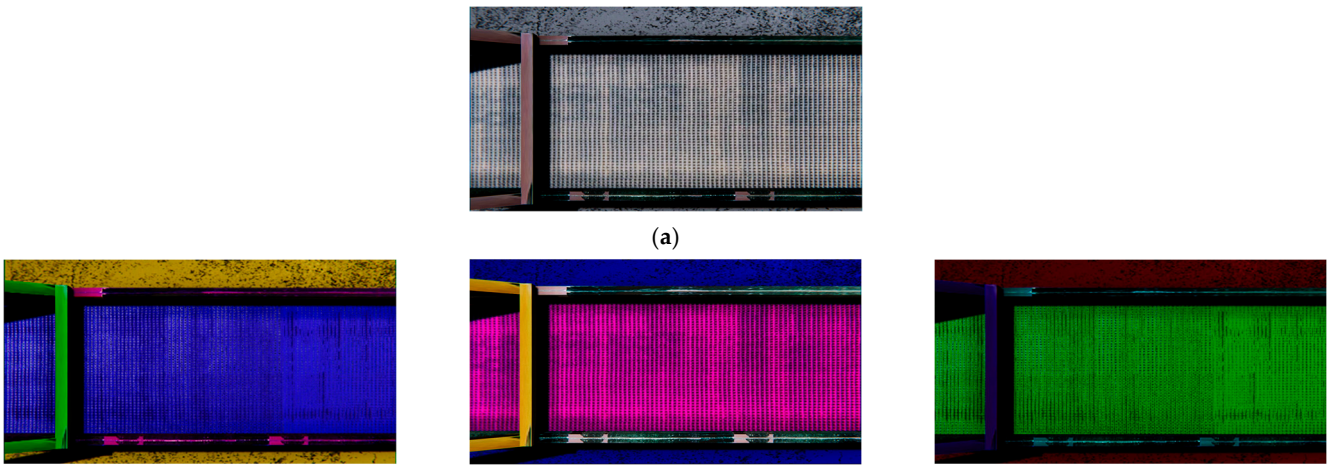
$$max\_angle = arctg\left(\frac{w}{l/2}\right) \quad (2)$$

where w is the distance from the object of interest to the nearest edge of the conveyor belt, and l is the length of the object of interest.

The "object of interest" and "conveyor belt" components are responsible for storing information about the object's position and for moving it through space from the creation point to the destruction point at a specified speed, which is set during the generation setup phase in the parameters section.

To achieve greater image diversity, an algorithm was developed to alter the color of the conveyor line. This helped prevent the model from overfitting to similar images, which would reduce its generalization ability and significantly lower prediction accuracy on data that differs from the training set [51].

During generation, the system tracks the elapsed process time. Once the user-defined time is reached, a background object is randomly selected, and its color characteristics are modified. The color of the background object is adjusted as follows: the system sequentially modifies the values of the three RGB material color channels randomly within the range of 0.0 to 1.0. The initial background is shown in Figure 7a, while the result of modifying the background objects is shown in Figure 7b.



(**a**)

(**b**)

Figure 7. Changing the color characteristics of background objects: (**a**) conveyor line without color change; (**b**) examples of conveyor line with changed colors.

To simulate the operation of a conveyor line and obtain images of the object under study from various angles, it is necessary to configure the movement of the camera. The Unity game engine provides a wide range of tools for this purpose through the Cinemachine package, which is designed for image capture control and is used to automate the process of creating virtual camera movements, thereby reducing the time required to develop such so-

lutions. The module includes implementations of algorithms for both randomly generating a motion path and tracking objects of interest. This tool significantly simplifies the process of changing viewpoints, eliminating the need for numerous manual adjustments [52].

Accordingly, as part of the synthetic data generation process, the tool described above is used to implement chaotic camera movement and capture frames from various angles, as shown in Figure 8. To achieve this, the Noise module was utilized, which enables camera movement based on a depth map generated using Perlin noise algorithms. This approach allows for the simulation of realistic motion and "shake" of the virtual camera in real time.

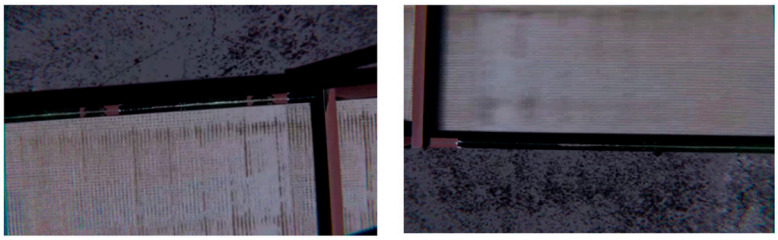
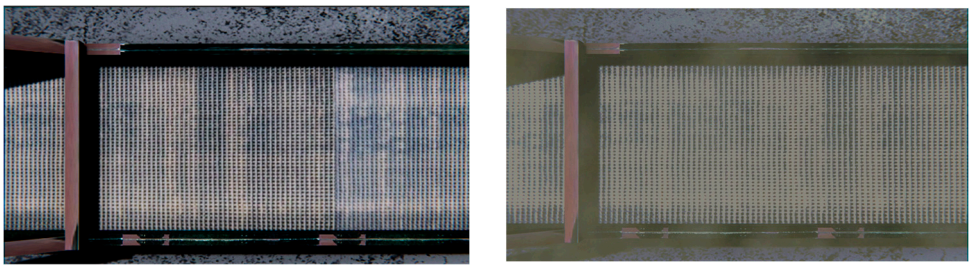


Figure 8. Examples of changing the camera angle.

To simulate a dusty environment, the Unity Particle System module is used in the implementation. According to the game engine's documentation, the particle system is designed to create effects from objects that do not have a defined shape and change in real time (such as smoke, fire, fluids, etc.) [53]. This description clearly applies to the "industrial dust" effect. Therefore, the particle system module is proposed for simulating dust in the environment. A dusty room with a conveyor belt is shown in Figure 9.



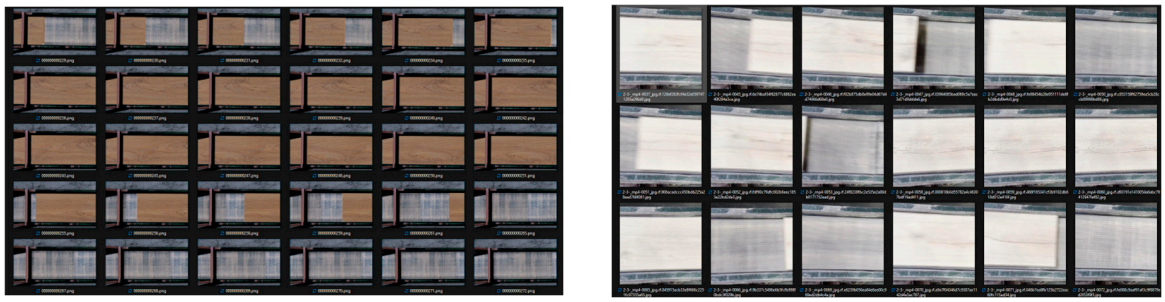
(**a**) (**b**)

Figure 9. A room filled with industrial dust: (**a**) a room without dust; (**b**) a room filled with dust.

The developed system makes it possible to generate various production scenarios and significantly expand the final image dataset, which helps prevent issues related to model overfitting.

5. Testing and Results

To test the data generation method, to neural networks with the YOLOv8 architecture were trained, due to its high accuracy in object detection tasks [54]. The use of two models is driven by the need to compare a network trained exclusively on synthetic data with one trained solely on real data. Accordingly, two separate datasets were created. The synthetic dataset, a fragment of which is shown in Figure 10a, consisted of 371 generated images that were automatically annotated using the Unity Perception tool. The second dataset was created from production video recordings, split into 371 images (Figure 10b), which were manually annotated using the Roboflow tool.



(**a**) (**b**)

Figure 10. A fragment of datasets: (**a**) synthetic dataset; (**b**) real dataset.

Next, augmentation methods were applied to these images sets to equally expand both datasets. The final size of each dataset was 815 images. After the datasets were formed, the neural network models were treated under identical conditions in the Google Colab environment. The models were trained for 100 epochs with a batch size of 16.

To evaluate the performance of the models, we used the mean average precision (mAP) metric at a threshold of 0.5 (mAP50). This metric is composed of several components, including precision (p) and recall (r), as well as the degree of intersection over union (IoU), and average precision (AP).

Precision is the proportion of correctly identified objects among all objects that the model has detected. In other words, it is a measure of how many of the model's predictions were correct.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

where TP is the number of true positive detections, and FP is the number of false positive detections.

Recall is the proportion of correctly detected objects among all objects that are actually present in the image:

$$recall = \frac{TP}{TP + FN} \tag{4}$$

where FN is the number of false negative detections.

IoU shows how well the predicted bounding box overlaps with the real one:

$$IoU = \frac{S(A \cap B)}{S(A \cup B)} \tag{5}$$

where A is the area of the predicted bounding box; B is the area of the true bounding box; S is the area of intersection or union of the bounding boxes.

The value 50 in the mAP50 metric indicates that the model's predictions are considered correct if the *IoU* between the predicted and the ground truth bounding box is greater than 0.5 (or 50%). In other words, for a prediction to be considered successful, there must be at least 50% overlap with the actual object.

Next, a precision–recall (PR) curve was calculated for each object class, showing how precision (p) varied with recall (r). The average precision (AP) for each class was then computed as the area under this PR curve:

$$AP = \int_0^1 p(r)dr \tag{6}$$

As a result, after obtaining the AP scores for all classes, a final mAP score is calculated. This is the average AP score for all object classes:

$$mAP = \frac{1}{k}\sum_{i}^{k} AP_i \tag{7}$$

where k is the number of classes.

As a result, if a model has an mAP50 value of 0.75, this means that on average across all object classes, detection precision (at $IoU > 0.5$) is 75%, which is considered a good result.

The mAP50 metric variation graphs are shown for models trained on synthetic data (Figure 11) and real data (Figure 12). Additionally, training was performed on synthetic data followed by fine-tuning on real data (Figure 13).

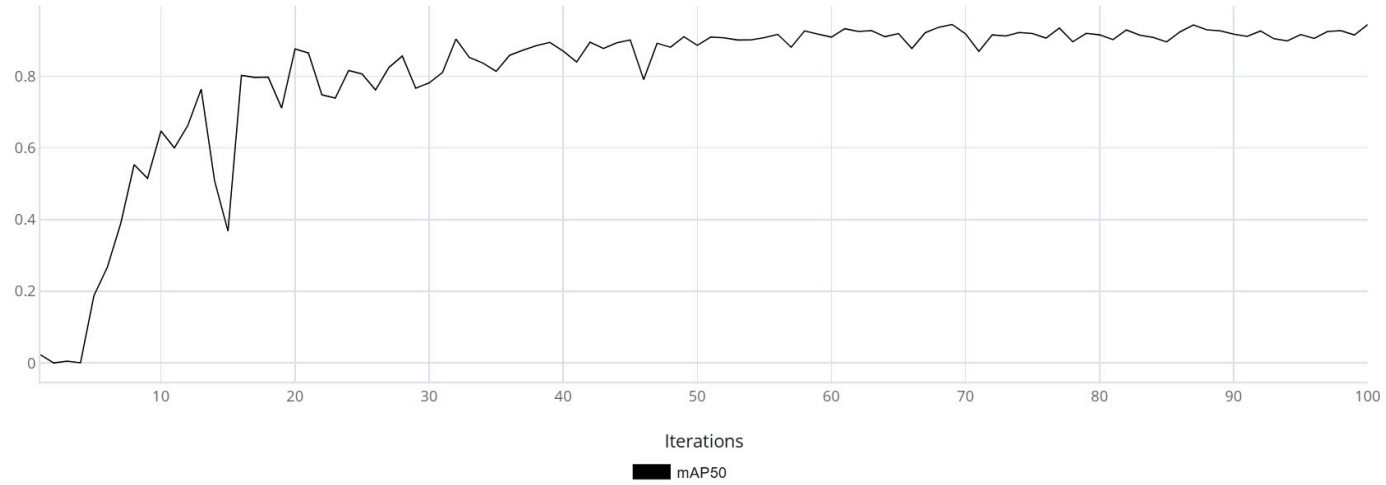


Figure 11. The graph of the average accuracy of classification and detection of the object of interest for a model trained on synthetic data.

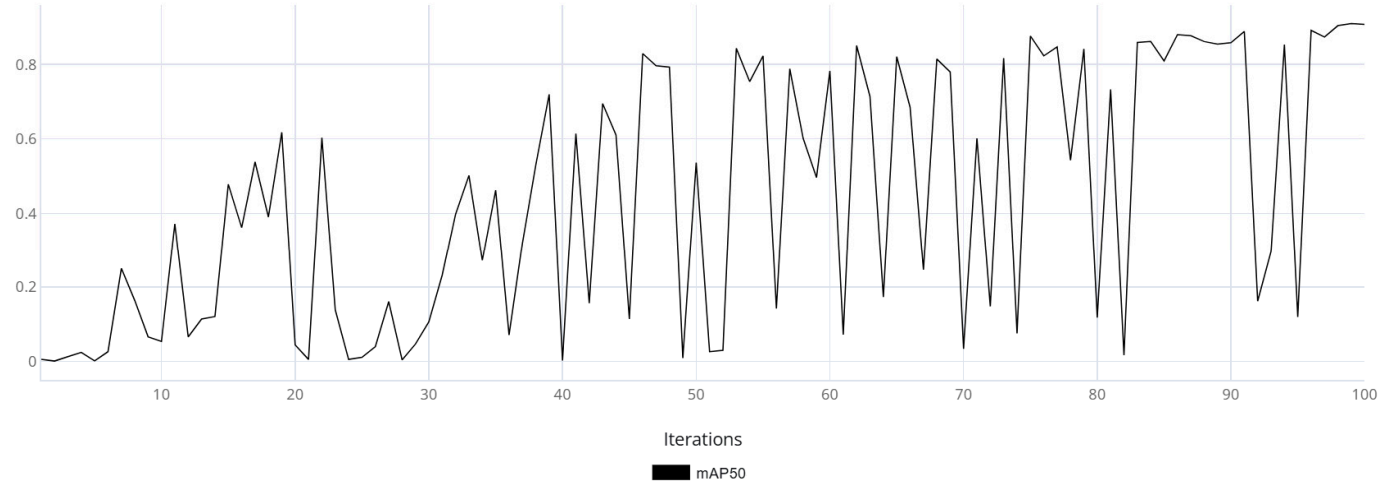


Figure 12. The graph of the average accuracy of the classification and detection of the object of interest for a model trained on real data.

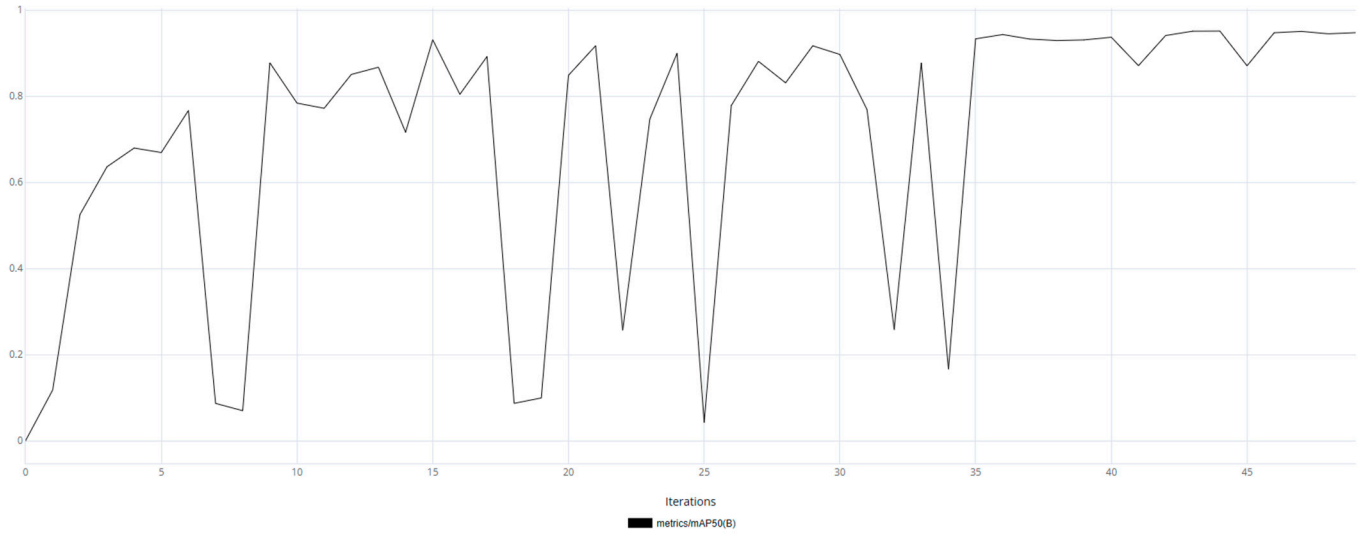


Figure 13. Average classification and detection accuracy for the model fine-tuned on real-world data.

Based on Figures 11–13, it can be observed that the model trained on real data made more frequent prediction errors and reached an mAP50 of 0.93 by the end of training. While this is a good result, it does not guarantee reliability, as the uneven training curve indicated low generalization capability. In contrast, the model trained using synthetic data demonstrated a smoother training metric curve, which suggested a more robust generalization ability. The final mAP50 value for the model trained on synthetic data was 0.94. The model fine-tuned on real-world data exhibited an initial decline in accuracy; however, as training progressed, it converged toward high performance. Ultimately, the model achieved a final mAP50 score of 0.95, indicating a high level of detection precision.

After training, both models were tested on a video clip from the production environment that served as the basis for development. This video was not used to create the real data dataset, as doing so would have biased the test results.

As a result of testing, the model trained on real data achieved an mAP50 of 0.36 on the test dataset from the enterprise. This suggests that, under the same hyperparameters and image augmentation methods, the model trained on real data was not adequate. This is supported by the mAP50 training curve, which reflects the model's inability to generalize to the presented images, indicating a lack of diversity and volume in the collected dataset. Meanwhile, the variety of the synthetic data helped overcome this issue, and the model demonstrated strong performance on the test video sequence, achieving an mAP50 of 0.95—evidence of the neural network's adequacy. On the test video, the third model demonstrated performance comparable to that of the model trained exclusively on synthetic data. The average mAP50 on the test set reached 0.96, which was undoubtedly a strong result in terms of detecting the target object.

In conclusion, the testing results showed that the model trained on synthetic data outperformed the one trained on real data in terms of accuracy. This leads to the conclusion that the data generator is an effective tool for building computer vision systems in industrial applications.

6. Discussion

Based on the testing data obtained from the two models, we can draw conclusions about the performance of the synthetic data generator compared to similar solutions.

First, the proposed method for generating synthetic data stands out by providing a generalized list of tasks, significantly speeding up the development process for image generation software.

Second, compared to the methodology described in [20], our solution offers greater sampling variability thanks to the capabilities of the game engine. This allows us to recreate any desired scenarios, increasing the accuracy of models by generating images for specific tasks. In contrast to solutions that use game engines with limited capabilities like GTA V, which do not allow changing the in-game code or adding additional scenarios, our generator offers more flexible settings for modeling.

The third key aspect of the proposed system is the speed at which images are generated. The system is able to generate approximately 1500 annotated images per minute, significantly exceeding the rendering speeds of 3D modeling tools like Blender. Achieving 1 frame per second in 3D modeling is considered a good result, but the proposed solution outperforms this in terms of both speed and flexibility. In addition to the above, compared to solutions based on the Unity game engine, the proposed system demonstrates significantly better performance. Considering the speed of image generation as a single metric—the number of images generated per second—the current generator produces 25 images per second, while solutions from [24,25] produce 6 and 4.6 images per second respectively. Therefore, the performance of the proposed system is at least four times better than similar solutions.

Furthermore, it should be noted that generating synthetic data significantly speeds up the development of computer vision systems in all cases considered. This allows creating models with high accuracy using only synthetic data and further improving them by combining synthetic and real data.

At the next stage, we plan to train models to not only detect the board but also to identify and classify any defects on it. Due to the lack of test data available for the next stage during development, we have decided to focus on training models for this purpose.

7. Conclusions

An overview of the fields of activity where computer vision (CV) can be applied has been conducted. The analysis revealed that traditional data collection methods, such as video recording and manual image annotation, are often costly and inefficient. Moreover, these methods are subject to human error, which can lead to mistakes and reduced accuracy in neural network models. As a result, synthetic data presents an alternative that can significantly reduce the time and financial costs associated with creating training datasets.

To address the image collection task, modern approaches to generating synthetic data for automatic object detection and classification tasks on production lines using CV technologies were examined. The primary goal was to create an effective and cost-efficient solution for forming training datasets in cases of limited access to real data or insufficient data. The analysis of existing solutions showed that the use of synthetic data is becoming an increasingly relevant and sought-after tool across various industries. Various tools and technologies for generating synthetic data were reviewed, such as game engines, 3D modeling tools, and specialized packages for automatic image annotation. Each of these solutions has its advantages and disadvantages. For instance, the use of game engines allows for real-time data generation and the processing of large volumes of images at high speed, significantly accelerating the dataset creation process. On the other hand, 3D modeling provides a higher level of realism but requires significant computational resources and time for rendering.

The generalized synthetic data generator model proposed in the study was applied to implement the technological process of quality control for flooring on a conveyor line. As

part of this implementation, a system was developed that simulates the real production process, taking into account all its characteristics, including camera angle changes, the addition of various defects to objects of interest, and the use of effects such as production dust. This enabled the creation of diverse and realistic images that can be used to train neural network models.

The final generation system was tested by comparing three neural network models with the YOLOv8m architecture. The use of three models was necessary to compare a network trained solely on synthetic data, one trained exclusively on real data and the model pretrained on synthetic and fine-tuned on real data. As a result, two corresponding datasets were created: the Synthetic dataset and the one based on real production video recordings.

After training the models, they were tested on a video clip from the enterprise that served as the basis for development. This video was not used to create the real data dataset, as using it would have distorted the test results. As a result of the testing, the model trained on real data achieved an mAP50 value of 0.36 on the test dataset from the enterprise. This suggests that, under the same hyperparameters and image augmentation methods, the model trained on real data was not adequate. In contrast, the diversity of synthetic data helped avoid this issue, and the model showed good performance on the test video sequence, achieving an mAP50 of 0.95, indicating the adequacy of the resulting neural network. The fine-tuning approach using real-world data demonstrated the highest performance mAP50 = 0.96, attributable to both the model's adaptation to real-world conditions and the diversity of images present in the synthetic dataset.

Based on the testing data for the two models, conclusions can be drawn about the performance of the proposed synthetic data generator in comparison with similar solutions. Finally, it should be noted that synthetic data generation significantly accelerates the development of computer vision systems in all the reviewed cases. This allows for the creation of high-accuracy models without using real data, and further improvement through the combination of real and synthetic data.

The next phase includes research on generating artificial data for bulk materials and minerals, using coal products as an example. To solve this task, computer vision models will need to be developed and configured to train on annotated images for identifying foreign materials in piles. Real data on objects of interest (impurities in rock and foreign materials) will also need to be collected.

Author Contributions: Conceptualization, A.Z., R.K. and A.S.; methodology, A.Z. and A.S.; validation, A.Z. and R.K.; formal analysis, A.S.; investigation, A.Z. and R.K.; resources, A.Z. and R.K.; data curation, A.Z.; software, A.Z.; writing—original draft preparation, A.Z, R.K. and A.S.; writing—review and editing, A.S.; visualization, A.Z.; supervision, A.S.; project administration, A.S.; funding acquisition, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science and Higher Education of the Russian Federation; project FEWM-2023-0013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author/s.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Greeshma, C.A.; Nidhindas, K.R.; Sreejith, P. Traffic control using computer vision. *Int. J. Adv. Res. Comput. Commun. Eng.* **2019**, *8*, 39–47.
2. Liu, G.; Shi, H.; Kiani, A.; Khreishah, A.; Lee, J.; Ansari, N.; Liu, C.; Yousef, M.M. Smart traffic monitoring system using computer vision and edge computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 12027–12038. [CrossRef]
3. Serrano, Á.; Conde, C.; Rodríguez-Aragón, L.J.; Montes, R.; Cabello, E. Computer vision application: Real time smart traffic light. In Proceedings of the Computer Aided Systems Theory–EUROCAST 2005: 10th International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, 7–11 February 2005; Volume 3643, pp. 525–530. [CrossRef]
4. Coifman, B.; Beymer, D.; McLauchlan, P.; Malik, J. A real-time computer vision system for vehicle tracking and traffic surveillance. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transp. Res. Part C Emerg. Technol.* **1998**, *6*, 271–288. [CrossRef]
5. Rusanovsky, M.; Beeri, O.; Oren, G. An end-to-end computer vision methodology for quantitative metallography. *Sci. Rep.* **2022**, *12*, 4776. [CrossRef]
6. Blackledge, J.; Dubovitskiy, D.A. Quality Control System using Texture Analysis in Metallurgy. In Proceedings of the Third International Conferences on Pervasive Patterns and Applications, Rome, Italy, 25–30 September 2011; Volume 978-1-61208-158-8, pp. 122–127. [CrossRef]
7. Harikrishna, K.; Davidson, M.J.; Reddy, G.D. New Method for Microstructure Segmentation and Automatic Grain Size Determination Using Computer Vision Technology during the Hot Deformation of an Al-Zn-Mg Powder Metallurgy Alloy. *J. Mater. Eng. Perform.* **2023**, *34*, 121–131. [CrossRef]
8. Sarrionandia, X.; Nieves, J.; Bravo, B.; Pastor-López, I.; Bringas, P.G. An Objective Metallographic Analysis Approach Based on Advanced Image Processing Techniques. *J. Manuf. Mater. Process.* **2023**, *7*, 17. [CrossRef]
9. Aydin, I.; Othman, N.A. A new IoT combined face detection of people by using computer vision for security application. In Proceedings of the 2017 International Artificial Intelligence and Data Processing Symposium IDAP, Malatya, Turkey, 16–17 September 2017; pp. 1–6. [CrossRef]
10. García, C.G.; Meana-Llorián, D.; G-Bustelo, B.C.P.; Lovelle, J.M.C.; Garcia-Fernandez, N. Midgar: Detection of people through computer vision in the Internet of Things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes. *Future Gener. Comput. Syst.* **2017**, *76*, 30–313. [CrossRef]
11. Gupta, S.; Garima, E. Road Accident Prevention System Using Driver's Drowsiness Detection by Combining Eye Closure and Yawning. *Int. J. Res. (IJR)* **2014**, *1*, 839–842.
12. Lan, R.; Awolusi, I.; Cai, J. Computer Vision for Safety Management in the Steel Industry. *AI* **2024**, *5*, 1192–1215. [CrossRef]
13. Costa, C.; Antonucci, F.; Pallottino, F.; Aguzzi, J.; Sun, D.W.; Menesatti, P. Shape analysis of agricultural products: A review of recent research advances and potential application to computer vision. *Food Bioprocess Technol.* **2011**, *4*, 1192–1215. [CrossRef]
14. Dhanya, V.G.; Subeesh, A.; Kushwaha, N.L.; Vishwakarma, D.K.; Kumar, T.N.; Ritika, G.; Singh, A.N. Deep learning based computer vision approaches for smart agricultural applications. *Artif. Intell. Agric.* **2022**, *6*, 211–229. [CrossRef]
15. Arakeri, M.P. Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry. *Procedia Comput. Sci.* **2016**, *79*, 426–433. [CrossRef]
16. Esteva, A.; Chou, K.; Yeung, S.; Naik, N.; Madani, A.; Mottaghi, A.; Liu, Y.; Topol, E.; Dean, J.; Socher, R. Deep learning-enabled medical computer vision. *NPJ Digit. Med.* **2021**, *4*, 5. [CrossRef]
17. Villalba-Diez, J.; Schmidt, D.; Gevers, R.; Ordieres-Meré, J.; Buchwitz, M.; Wellbrock, W. Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors* **2019**, *19*, 3987. [CrossRef] [PubMed]
18. Conrad, J.; Rodriguez, S.; Omidvarkarjan, D.; Ferchow, J.; Meboldt, M. Recognition of Additive Manufacturing Parts Based on Neural Networks and Synthetic Training Data: A Generalized End-to-End Workflow. *Appl. Sci.* **2023**, *13*, 12316. [CrossRef]
19. de Melo, C.M.; Torralba, A.; Guibas, L.; DiCarlo, J.; Chellappa, R.; Hodgins, J. Next-generation deep learning based on simulators and synthetic data. *Trends Cogn. Sci.* **2022**, *26*, 174–187. [CrossRef]
20. Kiefer, B.; Ott, D.; Zell, A. Leveraging synthetic data in object detection on unmanned aerial vehicles. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 3564–3571. [CrossRef]
21. Blender Foundation. Blender—Free and Open Source 3D Creation Software. Available online: https://www.blender.org/ (accessed on 9 May 2025).
22. Maxon. ZBrush—Digital Sculpting & Painting Software. Available online: https://www.maxon.net/en/zbrush (accessed on 9 May 2025).
23. Autodesk. 3ds Max—3D Modeling, Animation & Rendering Software. Available online: https://www.autodesk.com/products/3ds-max/overview (accessed on 9 May 2025).
24. Chumak, R. A Synthetic Data Generator. Training of Neural Networks for Industrial Flaw Detection. Available online: https://medium.com/phygitalism/synthetic-data-generator-a052d347468 (accessed on 5 May 2024). (In Russian)

25. Schmedemann, O.; Baaß, M.; Schoepflin, D.; Schüppstuhl, T. Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection. *Procedia CIRP* **2022**, *107*, 1101–1106. [CrossRef]
26. Reutov, I.; Moskvin, D.; Voronova, A.; Venediktov, M. Generating Synthetic Data To Solve Industrial Control Problems by Modeling A Belt Conveyor. *Procedia Comput. Sci.* **2022**, *212*, 264–274. [CrossRef]
27. Lee, H.; Jeon, J.; Lee, D.; Park, C.; Kim, J.; Lee, D. Game engine-driven synthetic data generation for computer vision-based safety monitoring of construction workers. *Autom. Constr.* **2023**, *155*, 105060. [CrossRef]
28. Pchelintsev, S.; Yulyashkov, M.A.; Kovaleva, O.A. A method for creating synthetic datasets for training neural network models to recognize objects. *Inf. Manag. Syst.* **2022**, 9–19. (In Russian) [CrossRef]
29. Manyar, O.M.; Cheng, J.; Levine, R.; Krishnan, V.; Barbič, J.; Gupta, S.K. Physics Informed Synthetic Image Generation for Deep Learning-Based Detection of Wrinkles and Folds. *ASME J. Comput. Inf. Sci. Eng.* **2023**, *23*, 030903. [CrossRef]
30. Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M.J.; Laptev, I.; Schmid, C. Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 109–117.
31. Hinterstoisser, S.; Lepetit, V.; Wohlhart, P.; Konolige, K. On pre-trained image features and synthetic images for deep learning. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; pp. 682–697.
32. Hou, X.; Sun, K.; Shen, L.; Qiu, G. Feature Perceptual Loss for Variational Autoencoder. 2024. Available online: https://arxiv.org/pdf/1610.00291 (accessed on 16 May 2025).
33. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A. Generative Adversarial Networks: An Overview. 2017. Available online: https://arxiv.org/pdf/1710.07035 (accessed on 16 May 2025).
34. Tripathi, S.; Chandra, S.; Agrawal, A.; Tyagi, A.; Rehg, J.M.; Chari, V. Learning to generate synthetic data via compositing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 461–470.
35. Saiz, F.A.; Alfaro, G.; Barandiaran, I.; Graña, M. Generative Adversarial Networks to Improve the Robustness of Visual Defect Segmentation by Semantic Networks in Manufacturing Components. *Appl. Sci.* **2021**, *11*, 6368. [CrossRef]
36. Goncharov, A.S. Digital twin: An overview of existing solutions and prospects for technology development. In *Information and Telecommunication Systems and Technologies, Proceedings of the All-Russian Scientific and Practical Conference, Kemerovo, 11–13 October 2018*; Goncharov, A.S., Saklakov, V.M., Eds.; EDN YQMPBZ; Kuzbass State Technical University Named after T.F. Gorbachev: Kemerovo, Russia, 2018; pp. 24–26. (In Russian)
37. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [CrossRef]
38. Koroteev, D.D.; Kim, A.A.; Vasyutin, A.O. Prospects for the use of digital twins in the construction industry. *Eurasian Sci. J.* **2024**, *16*, 12SAVN224. (In Russian)
39. Vikhman, V.V.; Romm, M.V. "Digital twins" in education: Prospects and reality. *High. Educ. Russ.* **2021**, *30*, 22–32. [CrossRef]
40. Verdouw, C.; Tekinerdogan, B.; Beulens, A.; Wolfert, S. Digital twins in smart farming. *Agric. Syst.* **2021**, *189*, 103046. [CrossRef]
41. Bhatti, G.; Mohan, H.; Singh, R.R. Towards the future of smart electric vehicles: Digital twin technology. *Renew. Sustain. Energy Rev.* **2021**, *141*, 110801. [CrossRef]
42. Erdélyi, V.; Jánosi, L. Digital Twin and Shadow in Smart Pork Fetteners. *Int. J. Eng. Manag. Sci.* **2019**, *4*, 515–520. [CrossRef]
43. Case Study: VDL Nedcar, Perspective Software. Available online: https://prespective-software.com/case-studies/case-study-vdl-nedcar/ (accessed on 1 March 2025).
44. Huang, Y.; Yuan, B.; Xu, S.; Han, T. Fault Diagnosis of Permanent Magnet Synchronous Motor of Coal Mine Belt Conveyor Based on Digital Twin and ISSA-RF. *Processes* **2022**, *10*, 1679. [CrossRef]
45. Singh, M.; Kapukotuwa, J.; Gouveia, E.L.S.; Fuenmayor, E.; Qiao, Y.; Murray, N.; Devine, D. Comparative Study of Digital Twin Developed in Unity and Gazebo. *Electronics* **2025**, *14*, 276. [CrossRef]
46. Pujana, A.; Esteras, M.; Perea, E.; Maqueda, E.; Calvez, P. Hybrid-Model-Based Digital Twin of the Drivetrain of a Wind Turbine and Its Application for Failure Synthetic Data Generation. *Energies* **2023**, *16*, 861. [CrossRef]
47. Digital Twin: Applications and Use Cases, Unity. Available online: https://unity.com/ru/topics/digital-twin-applications-and-use-cases (accessed on 12 March 2025).
48. Dustler, M.; Bakic, P.; Petersson, H.; Timberg, P.; Tingberg, A.; Zackrisson, S. Application of the fractal Perlin noise algorithm for the generation of simulated breast tissue. In Proceedings of the Medical Imaging 2015: Physics of Medical Imaging, Orlando, FL, USA, 21–26 February 2015; Volume 9412, pp. 844–852.
49. Bazuhair, W.; Lee, W. Detecting malign encrypted network traffic using perlin noise and convolutional neural network. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 0200–0206.

50. Li, H.; Tuo, X.; Liu, Y.; Jiang, X. A parallel algorithm using Perlin noise superposition method for terrain generation based on CUDA architecture. In Proceedings of the International Conference on Materials Engineering and Information Technology Applications (MEITA 2015), Guilin, China, 30–31 August 2015; pp. 967–974. [CrossRef]
51. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [CrossRef]
52. About Cinemachine. Available online: https://docs.unity3d.com/Packages/com.unity.cinemachine@2.8/manual/index.html (accessed on 26 May 2024).
53. Particle System. Available online: https://docs.unity3d.com/ru/530/Manual/ParticleSystems.html (accessed on 26 May 2024).
54. Sohan, M.; Sai Ram, T.; Reddy, R.; Venkata, C. A review on yolov8 and its advancements. In Proceedings of the International Conference on Data Intelligence and Cognitive Informatics, Tirunelveli, India, 18–20 November 2024; pp. 529–545. [CrossRef]

Journal of *Imaging*

MDPI

Article

Comparing Geodesic Filtering to State-of-the-Art Algorithms: A Comprehensive Study and CUDA Implementation

Pierre Boulanger * and Sadid Bin Hasan

Department of Computing Science, University of Alberta, Edmonton, AB T6G2R3, Canada; sadidbin@ualberta.ca
* Correspondence: pierreb@ualberta.ca

Abstract: This paper presents a comprehensive investigation into advanced image processing using geodesic filtering within a Riemannian manifold framework. We introduce a novel geodesic filtering formulation that uniquely integrates spatial and intensity relationships through minimal path computation, demonstrating significant improvements in edge preservation and noise reduction compared to conventional methods. Our quantitative analysis using peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) metrics across diverse image types reveals that our approach outperforms traditional techniques in preserving fine details while effectively suppressing both Gaussian and non-Gaussian noise. We developed an automatic parameter optimization methodology that eliminates manual tuning by identifying optimal filtering parameters based on image characteristics. Additionally, we present a highly optimized GPU implementation featuring innovative wave-propagation algorithms and memory access optimization techniques that achieve a 200× speedup, making geodesic filtering practical for real-time applications. Our work bridges the gap between theoretical elegance and computational practicality, establishing geodesic filtering as a superior solution for challenging image processing tasks in fields ranging from medical imaging to remote sensing.

Keywords: geodesic filtering; anisotropic diffusion; image processing; PSNR; noise reduction; edge preservation; GPU implementation; manifolds; Riemannian space

Received: 30 March 2025
Revised: 7 May 2025
Accepted: 11 May 2025
Published: 20 May 2025

Citation: Boulanger, P.; Bin Hasan, S. Comparing Geodesic Filtering to State-of-the-Art Algorithms: A Comprehensive Study and CUDA Implementation. *J. Imaging* **2025**, *11*, 167. https://doi.org/10.3390/jimaging11050167

1. Introduction

The central challenge in image filtering lies in achieving an optimal balance between noise reduction and the preservation of essential image features. Traditional methods rely on carefully designed mathematical models to selectively smooth images while retaining edges and textures. In contrast, modern neural network approaches employ data-driven learning to obtain similar objectives. However, both methodologies face inherent limitations: conventional methods often struggle with complex noise patterns, whereas neural networks typically require extensive training data and frequently lack robust theoretical guarantees, leading to unpredictable behavior.

In response to these challenges, this paper makes four specific contributions to the field of image processing:

1. Novel Geodesic Filtering Framework: We introduce a mathematically rigorous formulation of geodesic filtering that uniquely leverages Riemannian manifold theory to combine spatial and intensity information in a unified framework. Unlike other filters, the geodesic approach adapts to local image characteristics through

- Geometric distance metrics in a manifold: Allowing for a more natural measurement of similarity between pixels by integrating both spatial and intensity relationships.
- Local curvature considerations: Preserving the intrinsic structure of image features, especially along edges and contours.
- Adaptive kernel sizing based on manifold properties: Dynamically adjusting the filtering window in response to the local geometry of the image.

Unlike previous approaches that treat these domains separately, our method computes minimal geodesic paths that inherently respect image structure, offering significant advantages, including

- Enhanced edge preservation: Retaining sharp transitions and fine details.
- Superior noise reduction: Effectively suppressing both Gaussian and non-Gaussian noise.
- Fine texture detail retention: Maintaining subtle textures while reducing unwanted noise.
- Effective handling of variable noise levels: Adapting robustly to different noise intensities across the image.
- Sharp transition preservation during noise smoothing: Avoiding over-smoothing at boundaries.
- Enhanced performance with non-Gaussian noise: Outperforming traditional methods in challenging noise conditions such as speckle noise.
- Improved outlier robustness: Providing resilience against anomalous data points.

2. Comprehensive Comparative Analysis: We present the first systematic evaluation of geodesic filtering against state-of-the-art alternatives (including anisotropic diffusion, bilateral filtering, least median of squares, and deep image prior techniques) using standardized metrics (PSNR and SSIM) across diverse image types and noise conditions.
3. Parameter Optimization Framework: We developed a novel methodology for automatically determining optimal filtering parameters based on image characteristics. This approach eliminates the manual trial-and-error process typically required in advanced filtering techniques, making geodesic filtering more accessible for practical applications.
4. High-Performance CUDA Implementation: We introduce specific technical innovations in our GPU implementation that overcome the inherent computational complexity of geodesic filtering. Our wave-propagation algorithm and memory conflict resolution techniques achieve a 200$\times$ speedup over conventional CPU implementations, transforming geodesic filtering from a theoretically superior but computationally prohibitive method into a practical solution for real-time applications.

These contributions collectively bridge the gap between theoretical mathematical models and practical image processing applications, establishing geodesic filtering as both a robust theoretical framework and an effective computational tool for advanced image processing.

The remainder of this paper is organized as follows. In Section 2, we review the current literature on non-linear methods for image filtering, detailing their operational principles and implementation strategies. Section 3 introduces the fundamental mathematical concepts underlying geodesic filtering within a Riemannian manifold framework. In Section 4, we present extensive experimental results obtained from processing a diverse image database to demonstrate the impact of various filtering parameters on performance. We also conducted a comparative analysis with other state-of-the-art non-linear filtering tech-

niques. Section 5 describes our highly optimized CUDA C implementation that leverages GPU parallelism to address the computational challenges of geometric filtering. Finally, Section 6 concludes the paper by discussing the strengths and limitations of our approach and Section 7 outlines potential extensions to more complex 3D and 4D manifolds.

2. Literature Review

Early methods primarily targeted basic noise reduction using straightforward linear or spatial operations, often at the expense of image structure. Non-linear filtering methodologies represent a sophisticated class of image processing techniques that go beyond traditional linear approaches. These can be categorized into four distinct approaches, each with unique mathematical foundations and operational principles.

Anisotropic diffusion, the first category, draws from partial differential equations and heat flow theory. This approach intelligently modulates the diffusion process based on local image characteristics, allowing homogeneous regions to be smoothed while crucial edge features remain intact. By adapting the diffusion coefficient to the local gradient magnitude, these filters can reduce noise while preserving the structural integrity of images.

The second approach leverages robust statistics to create filters that minimize the influence of edges during smoothing operations. These methods typically replace standard mean calculations with robust estimators like least median square (LMS) that are less sensitive to outliers, effectively treating edge pixels as statistical anomalies. This statistical foundation allows for effective noise reduction without the edge blurring commonly associated with linear filtering techniques, providing superior performance in environments with varying noise distributions.

Geodesic filtering, the third category, represents a more geometrically oriented approach based on differential geometry and manifold theory. This methodology conceptualizes images as high-dimensional manifolds embedded in feature space. By computing geodesic distances, these filters can adaptively process images according to their inherent geometric properties. This approach excels at preserving fine structures and textural details while removing noise.

The fourth category encompasses neural-network-based methods, which leverage data-driven approaches to optimize filtering parameters. Unlike traditional approaches that rely on predefined mathematical models, these methods learn optimal filtering strategies directly from training data. By exposing networks to pairs of noisy and clean images, these systems can discover complex non-linear relationships that effectively separate signal from noise. The resulting filters often demonstrate remarkable adaptability across various noise types and image characteristics, though their performance depends heavily on the quality and diversity of the training data. This analysis specifically examines algorithms that do not require training data, thus excluding conventional neural network methods. For readers interested in deep-learning-based image denoising techniques, reference [1] offers a comprehensive overview. The Deep Image Prior (DIP) network [2] is the only neural network approach considered here, as it uniquely functions without pre-training requirements.

Each of these approaches offers distinct advantages and limitations, with their effectiveness varying according to specific application requirements, computational constraints, and the nature of the image data being processed.

2.1. Gradient Anisotropic Diffusion

Gradient anisotropic diffusion (GAD), as introduced by Perona and Malik [3], fundamentally reimagines image processing through the lens of heat diffusion equations. The core mathematical formulation begins with a partial differential diffusion equation:

$$\frac{\partial I}{\partial t} = div(g(|\nabla I|)\,\nabla I) \tag{1}$$

where $I(x, y, t)$ represents the image intensity at position (x, y) and time t, and $g(|\nabla I|)$ is the diffusion coefficient. This formulation builds upon the classical heat equation but introduces crucial non-linearity through the spatially varying diffusion coefficient.

The diffusion coefficient, as noted by Weickert et al. [4], plays a pivotal role in controlling the filtering process. It typically takes the form

$$g(|\nabla I|) = g(s) \tag{2}$$

where $g(s)$ is a function that reduces diffusion coefficient at the edges. Common formulations of $g(s)$, as discussed by [5], include

$$g(s) = \exp\left(-\frac{s^2}{C^2}\right) \text{ or } g(s) = \frac{1}{1 + s^2/C^2} \tag{3}$$

where C represents a threshold parameter controlling edge sensitivity.

The theoretical significance of this framework lies in its ability to achieve selective smoothing. As demonstrated by Alvarez et al. [5], the process preserves edges by reducing diffusion across high-gradient regions while promoting smoothing in homogeneous areas. The mathematical analysis by Weickert et al. [4] showed that this approach creates a scale-space representation with important theoretical properties:

- Causality: No spurious details are created with increasing scale.
- Immediate stabilization: Edge enhancement occurs in early iterations.
- Localization: Edges remain stable during the diffusion process.

On the other hand, GAD exhibits significant sensitivity to parameter configuration, with its effectiveness heavily reliant on appropriate selection of diffusion coefficients and iteration counts. The method frequently encounters stability and convergence challenges when parameters are improperly configured, creating substantial difficulty in establishing automated optimal termination criteria for the diffusion process. Additionally, its filtering capabilities show marked deterioration when confronted with impulse noise patterns or image regions containing complex textures, limiting its applicability across diverse image processing scenarios.

2.2. Curvature Anisotropic Diffusion Framework

Curvature anisotropic diffusion (CAD) formulation differs fundamentally from the original work of Perona and Malik by incorporating geometric information through the introduction of a mean curvature term. Initially proposed by Alvarez et al. [6] and further developed by Sapiro et al. [7], this framework offers superior preservation of geometric features while effectively reducing noise. The diffusion equation for curvature anisotropic diffusion is

$$\frac{\partial I}{\partial t} = g(k)|\nabla I| div\left(\frac{\nabla I}{|\nabla I|}\right) \tag{4}$$

where

- $I(x, y, t)$ represents the image intensity at position (x, y) and time t;
- k denotes the mean curvature of the level sets;

- $g(\kappa)$ is a decreasing function that controls the diffusion coefficient;
- ∇I is the image gradient;
- div represents the divergence operator.

For the right conditions, the CAD framework demonstrates exceptional ability to preserve critical image features such as edges and boundaries while effectively reducing noise. Its incorporation of geometric curvature information enables superior structural integrity maintenance compared to gradient-based methods. Unlike GAD, CAD incorporates mean curvature of level sets, allowing it to more faithfully respect the intrinsic geometry of image content, particularly along curved structures.

However, CAD's performance is heavily influenced by appropriate parameter selection. The threshold parameters within the g(κ) function demand careful calibration to achieve optimal results. This makes proper CAD implementation challenging, especially in discrete domains, potentially leading to numerical instabilities. Determining the ideal iteration count presents another non-trivial challenge, frequently requiring manual adjustment or sophisticated stopping criteria. While CAD performs well against Gaussian noise, it may exhibit reduced effectiveness when confronting other noise varieties such as impulse or speckle noise without specific adaptations.

2.3. Bilateral Filter Framework

First introduced by Tomasi et al. [8], this non-linear technique revolutionized image processing by combining domain and range filtering in a single, unified framework without having to compute gradients. The method's fundamental innovation lies in its ability to consider both spatial proximity and photometric similarity simultaneously.

The bilateral filter, as elaborated by Durand and Dorsey [9], operates on two fundamental principles:

- Spatial Domain Filtering: Pixels are weighted based on their spatial distance from the center pixel, following a Gaussian distribution. This component ensures that nearby pixels have more influence than distant ones.
- Signal Domain Filtering: Pixels are additionally weighted based on their photometric (intensity or color) similarity to the center pixel, again using a Gaussian distribution. This component ensures edge preservation by reducing the influence of pixels with significantly different intensities.

Paris et al. [10] formalized its mathematical formulation as

$$I(p) = \frac{1}{W_p} \sum_{q \epsilon \Omega} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I(p) - I(q)| \tag{5}$$

where

- I represent the input image intensity;
- p denotes the current pixel position;
- Ω represents the spatial neighborhood;
- q denotes the neighboring pixel position;
- G_{σ_s} and G_{σ_r} are Gaussian functions for spatial and range domains;
- W_p is the normalization factor.

Studies by Kaplan et al. [11] showed that this algorithm possesses superior noise reduction capabilities:

- Effective reduction of random noise;
- Preservation of underlying signal structure;
- Minimal introduction of artifacts.

On the other hand, bilateral filtering's effectiveness is significantly contingent on the appropriate selection of spatial and range parameters, necessitating meticulous calibration for optimal results across various applications. While the filter performs admirably against Gaussian noise, it demonstrates reduced efficacy when confronting alternative noise types such as impulse noise or structured noise patterns. Additionally, the filter can generate intensity shift artifacts in high-gradient regions. This creates edge localization challenges where, in areas with gradual transitions, the filter may not accurately maintain edge positions, potentially causing subtle displacement of boundary locations. Certain parameter configurations can also introduce unwanted piecewise constant regions, resulting in an artificial "staircase" effect in what should be smooth gradient areas.

2.4. Robust Least Median of Squares Filtering

While traditional filtering approaches like CAD, GAD, or bilateral filtering can address specific noise types, they often struggle with mixed noise patterns or fail to preserve important image features. Using the least median of squares (LMS) regression methods, introduced by Rousseeuw [12,13], offers a robust framework capable of handling up to 50% outlier contamination, making it ideal for edge preserving filters. Recent applications of LMS include edge-preserving smoothing [14] and feature detection [15].

The LMS filter processes images by looking at small windows of pixels of size $s \times s$ (typically 5×5 or 7×7) around a central pixel (x_0, y_0). Let us define a local image color approximation for this local window as polynomial of degree d:

$$f^c(x, y; \boldsymbol{\beta}^c) = \sum_{k=0,l=0}^{k+l \le d} \beta_{kl}^c (x - x_0)^k (y - y_0)^l \tag{6}$$

where (x, y) are the local coordinates of the pixels that are located inside the window; $\boldsymbol{\beta}^c = [\beta_{kl}^c]$ is the polynomial coefficients for the color channel c; and d is the order of the polynomial, typically 1 or 2.

The role of an LMS estimator is to determine the best coefficients $\hat{\beta}_{kl}^c$ that minimize the median error between the pixels in the window and the polynomial model:

$$\underset{\boldsymbol{\beta}^c}{argmin} \underset{i \in s^2}{\text{median}} \left((P_i^c - f^c(x_i, y_i; \boldsymbol{\beta}^c))^2 \right) \tag{7}$$

where P_i^c is the color value in channel c for pixel (x_i, y_i) inside the window, $f^c(x_i, y_i; \boldsymbol{\beta}^c)$ is the local polynomial model, and s^2 is the number of pixels in the processing window.

In each window, the algorithm tries to fit the polynomial model to the pixel values using a RANSAC (random sample consensus) algorithm by Fischler and Bolles [16] where a minimum sample set equal to $s_m = \frac{(d+1)(d+2)}{2}$ is used to compute candidate model coefficients $\boldsymbol{\beta}^c$(t). For each sampling iteration t, the algorithm computes the median values D_{med}^t of the error between the remaining pixels $P_i^c(\hat{x}_i, \hat{y}_i)$ and the current instance of the local polynomial model $f_t^c(\hat{x}_i, \hat{y}_i; \boldsymbol{\beta}^c(t))$. The number of random sample iterations is determined by $\frac{m=\ln(1-p)}{\ln\left(1-(1-\varepsilon)^{sm}\right)}$, which guarantees a confidence level of p = 0.99 for an outlier ratio of ε = 50%. After m iterations, the algorithm then chooses the model corresponding to the least median square value $D_{med}^{t_b}$ and its corresponding model coefficients $\boldsymbol{\beta}^c(t_b)$. Using this model, the algorithm diagnoses the pixels inside the window that are inliers vs. outliers by computing the difference $d_i^c = P_i^c(\hat{x}_i, \hat{y}_i) - f_t^c(\hat{x}_i, \hat{y}_i; \boldsymbol{\beta}^c(t))$ between the LMS model and the remaining pixel. Following Rousseeuw and Leroy [17], the robust threshold T_r to determine if a pixel is an inlier vs. outlier is

$$T_r = 1.4826 \times \text{median}\left(\text{abs}\left(d_i - D'_{med}\right)\right) \tag{8}$$

One can then diagnose the pixel using the following test: *if* $abs(d_i) \leq 2.5\ T_r$ then it is an inlier. Using the inlier pixels, the algorithm then computes the final model coefficients $\hat{\boldsymbol{\beta}}^c$ using a least mean square algorithm and then replaces the central pixels with the polynomial approximation $f^c\left(x_0, y_0; \hat{\boldsymbol{\beta}}^c\right)$. If the number of inliers is smaller or equal to s_m, then the central pixel is replaced by the median value of the window. For color images, we process each channel independently while maintaining color consistency through

- Joint outlier detection across channels.
- Consistent polynomial surface fitting.
- Color-aware scale estimation.

The LMS approach demonstrates remarkable resilience, handling contamination of up to 50% outliers, which makes it exceptionally robust for edge preservation and noise reduction compared to conventional filtering methods. LMS particularly excels at maintaining crisp edges and boundaries while efficiently eliminating noise, avoiding the blurring artifacts typically associated with alternative filtering techniques. The algorithm exhibits strong performance across diverse noise distributions and can effectively manage mixed noise patterns that frequently challenge other filtering approaches.

However, LMS filtering presents significant challenges in parameter optimization due to the complex interactions between polynomial degree, window size, and outlier threshold parameters that substantially influence performance outcomes. Additionally, LMS demands considerable computational resources, requiring multiple sampling iterations for each processing window, which can significantly impact processing time for larger images or real-time applications.

2.5. Deep Image Prior (DIP) Neural Network Filters

Deep image prior (DIP) represents a novel approach that harnesses the inherent structure of convolutional neural networks (CNNs) as an effective regularizer for natural image processing, without requiring pre-training on image datasets. The key insight is that architecture CNN inherently captures image statistics that make it biased toward natural images over noise. When optimizing an untrained CNN to reconstruct a corrupted image by minimizing the reconstruction loss, the network tends to learn the natural image content before fitting noise or artifacts. This approach has proven effective for various image restoration tasks including denoising [2], super-resolution [18], and inpainting [19], all without requiring any training data beyond the single corrupted image being processed. DIP offers several significant advantages over standard neural network algorithms: it enables zero-shot learning through the network's architecture serving as a natural image prior, provides flexibility across multiple restoration tasks without dataset bias, and offers interpretability in how it captures image statistics. However, DIP faces notable limitations: it is computationally intensive, requiring thousands of iterations per image. The method is also sensitive to early stopping criteria and hyperparameter selection. It also lacks theoretical guarantees relying instead on empirical observations resulting in inconsistent results due to random initialization and dynamics.

One variant of DIP is Wavelet-DIP, which has been shown by Yang, Y., et al. [20] and Liu, C., et al. [21] to enhance the original DIP framework by incorporating wavelet decomposition into the network architecture, leveraging the multi-scale analysis capabilities of wavelets. To improve processing images with Gaussian noise, a similar version to Wavelet-DIP was proposed. Ulyanov et al. introduced a change to Wavelet-DIP called the Gaussian Weighted Wavelet-DIP [22] (GW-DIP) where the wavelet function is first convolved by a Gaussian filter. GW-DIP solves the optimization problem:

$$min_\theta\{L(f_\theta(z),\ I) + \lambda R(f_\theta(z))\} \tag{9}$$

where f_θ is the neural network with parameters θ, z is a random noise input, x is the target image, R is the regularization term, and λ is a regularization weight. This minimizes the distance between the network output $f_\theta(z)$ and target image I, with regularization R.

Initial random noise $z \, \epsilon \, N(0,1)$ is transformed through Gaussian weighting:

$$z'(p) = \frac{1}{K(p)} \sum_{q \, \epsilon \, \Omega} G_{\sigma_w}(p-q) z(q) \tag{10}$$

where K(p) = $\sum_{q \, \epsilon \, \Omega} G_{\sigma_w}(p-q)$ is a normalization factor, $G_{\sigma_w}(p)$ is 2D Gaussian kernel, Ω is spatial neighborhood window, and $p = (x_p, y_p)$ and $q = (x_q, y_q)$ are pixel coordinates.

The feature map F is a discrete wavelet transform (DWT) defined as

$$\left\{A_{i,k}, D^h_{i,k}, D^v_{i,k}, D^d_{i,k}\right\} = DWT(F) \tag{11}$$

where i is the decomposition level, k is the spatial location, $A_{i,k}$ is the low-pass approximation, $D^h_{i,k}$ is the horizontal components, $D^v_{i,k}$ is the vertical components, and $D^d_{i,k}$ is the diagonal components. In our test implementation, we used the Haar wavelet for its simplicity and computational efficiency. One can see in Figure 1 the architecture of the network.

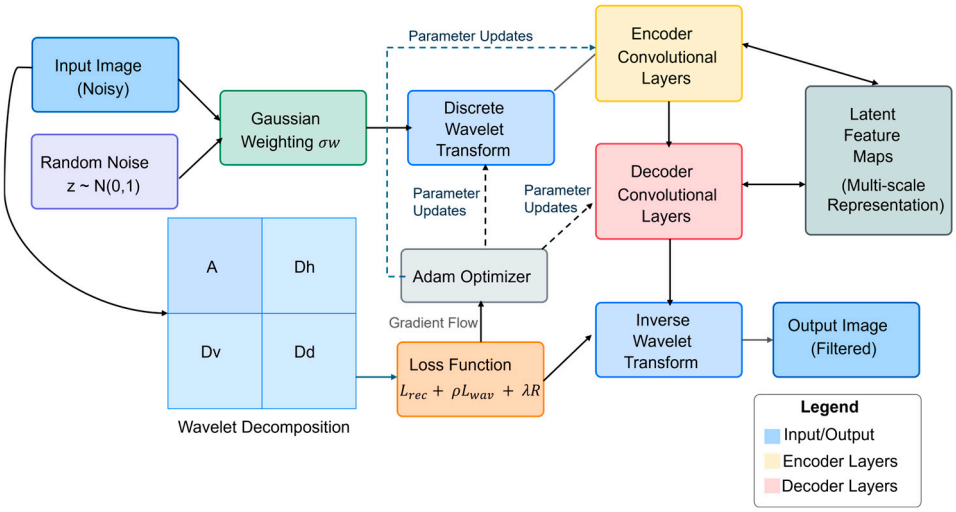


Figure 1. Gaussian Wavelet-DIP architecture.

The combined loss function is $L_{total} = L_{rec} + \rho L_{wav} + \lambda R(f_\theta(z))$ where $L_{rec} = |f_\theta(z) - I|^2$ is the reconstruction loss function, $L_{wav} = \sum_{i,k} |DWT(f_\theta(z))(i,k) - DWT(I)(i,k)|^2$ is the wavelet loss function, and $R(f_\theta(z)) = \sum_p |\nabla f_\theta(z)|$ the total variation regularization term. The parameters $\boldsymbol{\theta}$ are updated using Adam optimizer:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \pi \times \left(\frac{\hat{m}(t)}{\sqrt{\hat{v}(t)}} \right) + \epsilon \tag{12}$$

where

- $m(t) = \beta_1 m(t-1) + (1-\beta_1) \nabla_\theta L_{total}$;
- $v(t) = \beta_2 v(t-1) + (1-\beta_2) \nabla_\theta L^2_{total}$;
- $\hat{m}(t) = \frac{m(t)}{1-\beta_1^t}$;
- $\hat{v}(t) = \frac{v(t)}{1-\beta_2^t}$.

with typical values of learning rate $\pi = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-8}$.

By processing different frequency components separately through dedicated network branches, GW-DIP can better handle various image features at different scales. The wavelet transform naturally separates an image into low-frequency approximation coefficients and high-frequency detail coefficients, allowing the network to learn appropriate representations for each frequency band. This multi-scale wavelet structure provides additional information that aligns with natural image statistics, as wavelets are known to produce sparse representations of natural images.

GW-DIP leverages the intrinsic structure of convolutional neural networks (CNNs) as an implicit regularization mechanism for natural image processing, eliminating the need for pre-trained data. This method demonstrates remarkable capability in eliminating complex noise patterns that typically challenge conventional filtering techniques, as it adapts to image-specific characteristics. The architectural design of the network inherently preserves significant edges and details while effectively removing noise.

However, DIP demands substantial computational resources, requiring thousands of optimization iterations for processing a single image. The approach necessitates vigilant monitoring and strategic early stopping to prevent the network from eventually fitting to noise patterns, which complicates automation efforts. Furthermore, the quality of results significantly depends on several factors including network architecture, learning rate, and various hyperparameters that require meticulous tuning for optimal performance.

3. Geodesic Filtering

Geodesic filtering, first introduced by Boulanger [23] to process range data $x(u,v)$, $y(u,v)$, $z(u,v)$ and later by Sochen et al. [24] to process intensity images, addresses image filtering from a fundamentally different mathematical perspective. To generalize this work, we introduce a novel filtering framework that treats signals as a m-dimensional Riemannian manifold $\mathbf{\Pi}$ embedded in a n-dimensional Euclidian space, typically combining spatial and signal value coordinates. Let $\boldsymbol{r}(\mathbf{\Pi_p})$ be a n-dimensional vector immersed into a m-dimensional rectangular manifold with coordinate $\mathbf{\Pi_p}$. For 2D images, the manifold coordinates of a point p is $\mathbf{\Pi_p} = (u_p, v_p)$, corresponding to a mapping from R^2 to R^n such as:

$$\mathbf{\Pi_p} \rightarrow (u_p, v_p) \rightarrow \ (x(u_p, v_p), y(u_p, v_p), \boldsymbol{r}(u_p, v_p)) \tag{13}$$

In the case of a color image, the mapping is R^2 to R^5, which is a mapping from (u,v) to $(x(u,v), y(u,v), R(u,v), G(u,v), B(u,v))$ and for grey-level images R^2 to R^3 a mapping from (u,v) to $(x(u,v), y(u,v), I(u,v))$.

The geodesic distance between two points $p(\mathbf{\Pi})$ and $q(\mathbf{\Pi})$ on $\mathbf{\Pi}$ is defined as

$$d(p(\mathbf{\Pi}), q(\mathbf{\Pi})) = inf_{\gamma} \sqrt{dx(\mathbf{\Pi})^2 + dy(\mathbf{\Pi})^2 + \alpha^2 \|dr(\mathbf{\Pi})\|^2} \tag{14}$$

where inf_{γ} function is taken over all possible shortest paths γ connecting $p(\mathbf{\Pi})$ and $q(\mathbf{\Pi})$ on the manifold. The parameter α weights the importance between the spatial components and the signal differences. This formulation, as analyzed by Kimmel et al. [25], naturally incorporates both spatial and signal value differences into a single geometric framework. The advantages of using geodesic distance on a Riemannian manifold are

- Geodesic distance provides intrinsic measure of similarity;
- Accounts for both spatial and signal differences;
- Preserves image structure better than Euclidean metrics;
- Adapts to local image geometry.

The filtering process utilizes geodesic distances through a weighted averaging:

$$\boldsymbol{r}'(p(\boldsymbol{\Pi})) = \frac{\int w(\boldsymbol{\Pi})\boldsymbol{r}(q(\boldsymbol{\Pi}))dq}{\int w(\boldsymbol{\Pi})dq} \tag{15}$$

where $w(\boldsymbol{\Pi})$ is a decreasing function of the geodesic distance relative to $p(\boldsymbol{\Pi})$, such as

$$w(\boldsymbol{\Pi}) = \exp\left(-\frac{d(p(\boldsymbol{\Pi}), q(\boldsymbol{\Pi}))^2}{2\sigma^2}\right) \tag{16}$$

The theoretical properties of this framework, as established by Boulanger [24] and later by Mémoli and Sapiro [26], include

- Intrinsic geometry preservation;
- Adaptive neighborhood consideration;
- Natural handling of curved structures;
- Topology preservation.

The geodesic distance between points on this manifold incorporates both spatial and signal/geometry differences, providing a natural mechanism for edge-preserving smoothing. This distance measure, fundamental to the filtering process, respects the intrinsic structure of the image rather than relying solely on Euclidean distances in ambient space, making the filtering invariant to rigid transformation for range data.

Peyré [27] further developed these concepts, introducing efficient computational schemes and establishing important theoretical properties of the filtering process. The framework demonstrates several advantageous properties, including rotation invariance, contrast invariance, and the preservation of significant image features.

Modern implementations of geodesic filtering incorporate several sophisticated features. Castaño-Moraga et al. [28] introduced tensor-based extensions that better handle directional features and complex textures. Their work demonstrated improved performance in preserving fine details while still effectively reducing noise.

Zhang et al. [29], develops geometric filtering and edge detection algorithms for non-Euclidean image data, viewing image data as residing on a Riemannian manifold. They extend classical filtering techniques like median filtering and Perona-Malik anisotropic diffusion to handle non-Euclidean data through geodesic distances and the exponential map.

3.1. Geodesic Convolution on a Discrete Manifold

Geodesic filtering implementation requires careful consideration of both theoretical principles and practical computational aspects. Implementing geodesic filtering on a discrete manifold follows the theoretical framework established by Boulanger [23] and Sochen et al. [24]. From Equation (15), geodesic convolution is defined on a discrete manifold as

$$\hat{\boldsymbol{r}}_{\alpha,\sigma}(u_o, v_o) = \frac{1}{N(u,v)} \sum_{u \in w_u} \sum_{v \in w_v} \boldsymbol{r}(u,v) e^{-\frac{d^2(\boldsymbol{r}(u,v), \boldsymbol{r}(u_o,v_o), \alpha)}{2\sigma^2}} \tag{17}$$

where $d(\boldsymbol{r}(u,v), \boldsymbol{r}(u_o,v_o), \alpha)$ is the geodesic distance between $\boldsymbol{r}(u,v)$ in the window neighborhood of size (w_u, w_v) and the center of the window $\boldsymbol{r}(u_o,v_o)$. $N(u,v)$ is the normalization factor equal to

$$N(u,v) = \sum_{u \in w_u} \sum_{v \in w_v} e^{-\frac{d^2(\boldsymbol{r}(u,v), \boldsymbol{r}(u_o,v_o), \alpha)}{2\sigma^2}} \tag{18}$$

Modern implementations incorporate adaptive parameter selection schemes as proposed by Alonso-González et al. [30]. In our implementation the parameters to be adjusted are:

- α (signal weighting) based on local gradient statistics;
- σ (filtering extent) based on local noise estimates;
- Window size W based on feature scale analysis.

3.2. Minimal Patch Calculation on a Discrete Convolution Window

The computation of minimal paths within local windows is a critical component of geodesic filtering, as it directly influences how image features affect the overall filtering process. Initially formalized by Boulanger [23] and Sethian [31], this process determines the optimal paths that capture the intrinsic geometry of an image. The accuracy and efficiency of these minimal path calculations not only dictate the quality of the filtered output but also have a significant impact on the algorithm's computational performance. Over the years, various methods have been developed and evaluated for this task. In this work, we implement two primary algorithms: (a) Dijkstra's algorithm, optimized for scalar processors, and (b) the fast-marching method, which is tailored for efficient GPU-based parallel implementation.

3.2.1. Dijkstra's Algorithm

The foundation of minimal path calculation lies in graph theory, with Dijkstra's algorithm [32] serving as a seminal work in this area. In the context of range image processing, as elaborated by Boulanger [23], the inherently discrete nature of digital images maps naturally onto graph structures, where pixels become vertices, and their relationships are represented by weighted edges. Kimmel et al. [25] further advanced this concept by developing continuous formulations that bridge the gap between discrete and continuous manifolds, thereby reinforcing the theoretical underpinnings of geodesic filtering. Extensive analysis by Sethian et al. [31] has shown that under suitable conditions, discrete graph-based methods converge with the solutions obtained from continuous manifold formulations. This theoretical bridge, further refined by Mirebeau [33], forms the basis for modern hybrid approaches that blend discrete and continuous perspectives.

Furthermore, the work of Mémoli and Sapiro [26] established several essential theoretical properties for minimal path computations that any robust method must satisfy:

- Consistency of discrete approximations: Ensuring that as the discretization is refined, the calculated paths converge to the continuous geodesics.
- Convergence rates under refinement: Providing guarantees on the speed and accuracy with which the discrete solution approximates the continuous solution.
- Stability with respect to perturbations: Maintaining reliable performance even when the input data is subject to noise or other perturbations.

By adhering to these foundational principles, our implementation of Dijkstra's algorithm achieves both high accuracy and computational efficiency, serving as a robust baseline for minimal path calculation in geodesic filtering.

3.2.2. Dijkstra's Algorithm Implementation

The conversion of image data to graph structure, as formalized by Boulanger [23] and Vincent [34], requires careful consideration of both spatial and signal relationships. Let $G = (V, E)$ be a graph representation of an image where (see Figure 2)

- Vertex set V represents pixel locations;
- Edge set E connects neighboring pixels;
- Weight function w: $E \rightarrow R+$ incorporates distance metrics.

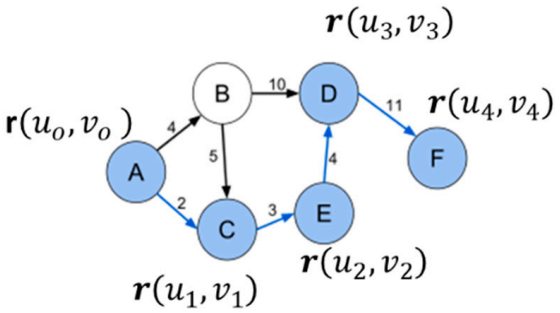


Figure 2. Example of a single-source (vertex A) multiple destination graph (vertices B,C,D,E,F).

The fundamental mappings are

1. Each pixel corresponds to a vertex in the graph, with edges connecting neighboring pixels. The connectivity pattern, typically 4-connected or 8-connected, significantly influences path calculation accuracy. Kimmel et al. [26] demonstrated that 8-connectivity provides better angular resolution at the cost of increased computational complexity.
2. The edge weights incorporate both spatial and signal value information. The general form of edge weight between pixels p and q is defined by Equation (14).
3. The efficiency of priority queue operations becomes crucial in image processing applications. Recent work by Lewis [35] demonstrated that while Fibonacci heaps offer optimal theoretical complexity, as demonstrated in Boulanger [23], binary heaps often perform better in practice due to simpler operations.

Our implementation begins with the definition of essential data structures. We define a pixel structure that contains the following components: spatial coordinates $(x(u,v), y(u,v))$, signal value $\boldsymbol{r}(u,v)$, current computed distance from source, a visited flag, and a predecessor reference for path reconstruction. Additionally, an edge structure is defined to represent connections between pixels, containing references to start and end pixels along with a weight value that combines spatial and intensity differences.

A priority queue structure is implemented using a binary heap, as recommended by Boulanger [23] maintaining pairs of distance values and pixel references. The queue supports three primary operations: insertion of new elements, extraction of minimum-distance elements, and key decrease operations for distance updates. The algorithm works as follows (see Figure 3):

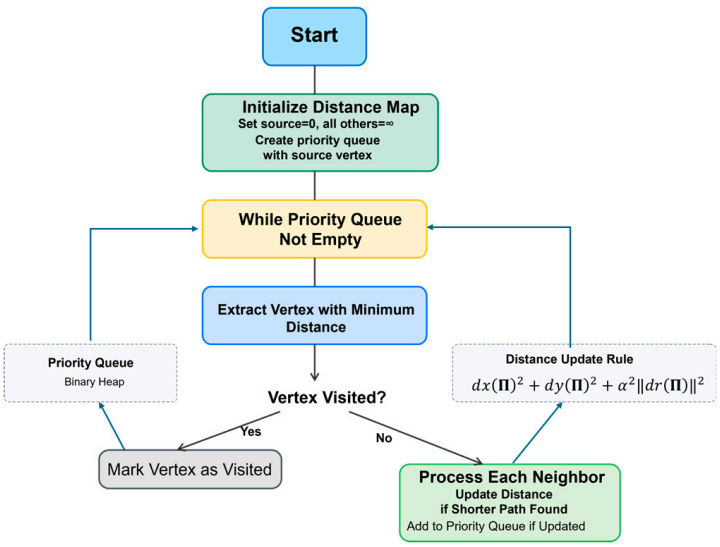


Figure 3. Dijkstra's algorithm block diagram.

1. Initialization Process

We create a two-dimensional array (pixel_grid) representing the image dimensions. For each position in the image, we instantiate a pixel object with coordinates matching its position, signal values from the original image, distance initialized to infinity (except for the source pixel which gets zero), visited flag set to false, and null predecessor reference. The source pixel's distance is set to zero, and it is inserted into the priority queue with this initial distance. This setup establishes the starting point for the algorithm's propagation phase.

2. Propagation Phase

The core processing loop operates as follows. While the priority queue is not empty, we are repeatedly

- Extracting the pixel with minimum distance from the queue;
- If the pixel has already been visited, skip to next iteration;
- Mark the current pixel as visited;
- Process all neighbors of the current pixel.

For each unvisited neighbor, we

1. Calculate a new potential distance combining
   - Spatial distance between pixels;
   - Signal difference magnitude weighted by parameter β.
2. If the new distance is smaller than the neighbor's current distance,
   - Update the neighbor's distance;
   - Set the current pixel as the neighbor's predecessor;
   - Insert the neighbor into the priority queue with its new distance.
3. Neighbor Processing
   The neighbor processing phase implements an 8-connectivity pattern. For the current pixel position (u, v), we examine all eight adjacent positions:
   - Horizontal neighbors: $(u \pm 1,\ v)$;
   - Vertical neighbors: $(u,\ v \pm 1)$;
   - Diagonal neighbors: $(u \pm 1,\ v \pm 1)$.

   For each potential neighbor position, we
   - Verify position validity within image boundaries;
   - Calculate combined spatial–intensity distance;
   - Process distance updates if necessary.
4. Distance Calculation
   The distance calculation combines spatial and signal components:
   - Calculate the Euclidean distance $d_S^2(u, v)$ between neighboring pixel coordinates (u, v) and (u', v') :

$$d_S^2(u, v) = ((x(u, v) - x(u', v')^2 + (y(u, v) - y(u', v')^2 \quad (19)$$

   - Account for diagonal connections with appropriate scaling;
   - Compute the norm of the n-dimensional signal difference $\boldsymbol{r}(u, v)$ and $\boldsymbol{r}(u', v')$:

$$d_{\boldsymbol{r}}^2(u, v) = \left\|\boldsymbol{r}(u, v) - \boldsymbol{r}(u', v')\right\|^2 \quad (20)$$

   - Use a parameter α to assess the significance of the signal difference in comparison to the spatial component.

5. *Final Distance*
    - Combine components using square root of sum of squares;
    - Apply any additional feature-based weighting.

3.2.3. Time and Memory Complexity

The time complexity of Dijkstra's algorithm for an image of size $M \times N = s$:

- $O(slog\ s\ )$ using binary heap;
- $O(s + Elog\ s\ )$ for a Fibonacci heap where E is the number of edges.

Even though the algorithm is $s\log s$ efficient, because of the random access to the heap memory, the algorithm is highly inefficient from a memory access point-of-view for parallel implementation, which requires coalesced memory access. For this reason, a second algorithm, more amicable to parallel processing, was implemented for the CUDA version.

3.3. Emphasizing Structural Integrity in Geodesic Filtering

Structural integrity represents one of the most significant advantages of geodesic filtering over conventional approaches. This aspect deserves particular emphasis as it directly addresses a fundamental challenge in image processing: preserving essential image structures while effectively removing noise.

By modeling images as high-dimensional manifolds and computing distances along the manifold surface rather than in ambient Euclidean space, geodesic filtering inherently respects the intrinsic geometry of image content. This fundamental difference allows it to

1. Preserve Edge Continuity: Unlike bilateral filtering or anisotropic diffusion that can fragment edges under high noise conditions, geodesic filtering maintains continuous edge structures even with significant noise contamination. This is because geodesic paths naturally follow edge contours along the manifold surface.
2. Maintain Topological Properties: The approach preserves important topological relationships between image regions, ensuring that connected components remain connected and boundaries remain intact after filtering. This is crucial for downstream tasks like segmentation or feature extraction.
3. Adapt to Intrinsic Feature Scale: The geodesic distance calculation automatically adapts to the local feature scale, providing stronger preservation of fine details in textured regions while still effectively smoothing homogeneous areas.
4. Respect Perceptual Organization: By following the natural organization of visual information in the image, geodesic filtering produces results that better align with human visual perception, maintaining the hierarchical structure of image content.

Research demonstrates that with optimized parameter configuration, geodesic filtering consistently surpasses alternative methodologies in preserving structural elements across a wide spectrum of image types. This performance differential becomes particularly significant in demanding applications such as medical imaging, where maintaining the structural integrity of anatomical features directly impacts diagnostic reliability. The exceptional structural preservation achieved through geodesic filtering represents not merely an incremental enhancement in visual quality but a fundamental advancement in preserving the semantic significance of visual information throughout the filtering process.

3.4. Comparison of Computational Complexity: The Overall Computing Complexity for a M·N Image for Each Method Is

- Geodesic Filtering: $O\left(M{\cdot}Ns^2\ \right)$;
- LMS Filter: $O\left(M{\cdot}N \times s^2 \times p\right)$;
- Gradient Anisotropic Diffusion: $O(T{\cdot}M{\cdot}N)$;

- Curvature Anisotropic Diffusion: $O(T \cdot M \cdot N)$;
- Bilateral Filter: $O(M \cdot N \cdot s^2)$;
- Gaussian Weighted Wavelet DIP: $O(T \cdot N \cdot M \cdot log(N \cdot M))$.

Here, T is the number of iterations, s is the window size, and p = polynomial terms.

4. Experimental Results

This section directly addresses our three core contributions through carefully designed experiments. First, we evaluated the theoretical advantages of geodesic distance calculation in preserving image structures. The parameters α and σ are central to our analysis because they represent the fundamental balance between spatial and intensity information in the geodesic framework. Parameter α controls the relative weighting of intensity differences versus spatial proximity, directly influencing edge preservation capabilities. Parameter σ determines the extent of filtering influence, governing the scale at which features are preserved or smoothed. The main goals of our experiments are to systematically explore the parameter space to demonstrate:

1. The existence of optimal parameter combinations that maximize both PSNR and SSIM metrics across diverse image types.
2. The superior performance of geodesic filtering compared to traditional approaches in maintaining edge integrity while reducing noise.
3. The adaptability of the method to different noise conditions without requiring extensive parameter re-adjustments.

4.1. Image Dataset

To validate the functionality of the algorithms, a standard set of test images was used. These include natural scenes, people, industrial sites, and medical and city images (see Figure 4).

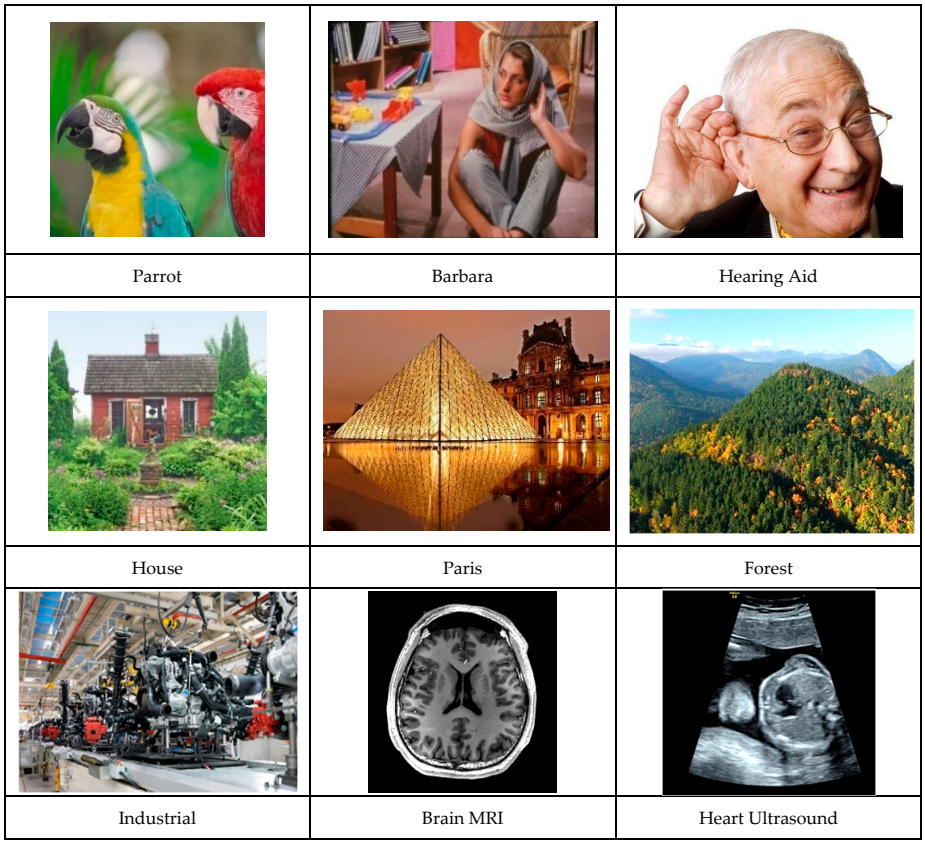


Figure 4. Image database used for the experiments.

4.2. Evolution of Image Dataset vs. Filter Parameters

This section explores the relationship between filter parameters (α and σ) and filtering performance across diverse image types, showing that

1. Each image has an optimal σ value corresponding to its "natural scale";
2. Parameter α effectively balances spatial and intensity relationships;
3. The filter's performance peaks at specific parameter combinations, demonstrated through quantitative metrics (PSNR and SSIM).

In the sequence shown in Figure 5, we convolved the images that were first normalized in size to be of width of 512 pixels and height to a value that respects the aspect ratio of the original image. In these experiments, the convolution window size was set to 11 × 11, and the parameter α equal to 1.

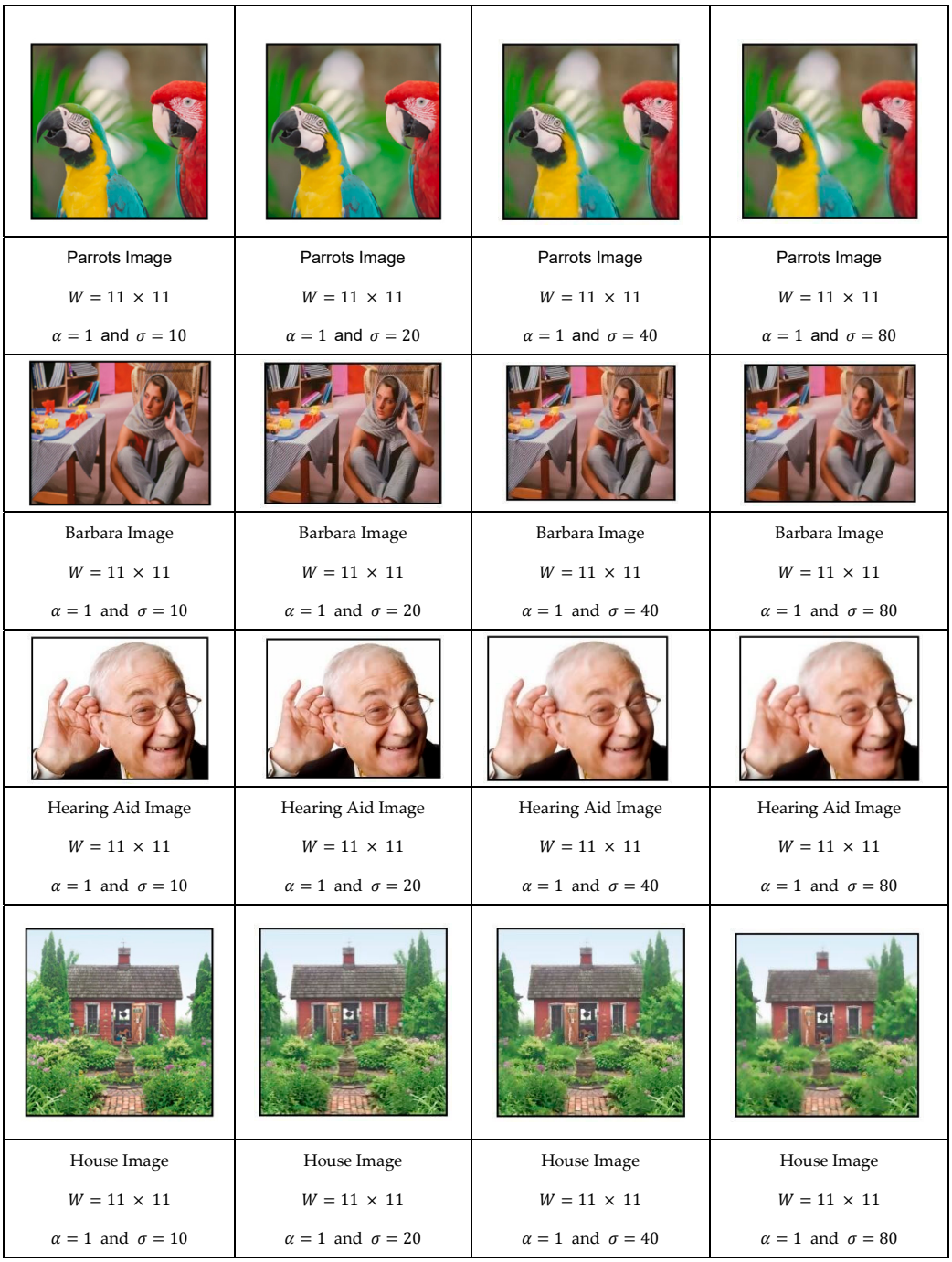


Figure 5. *Cont.*

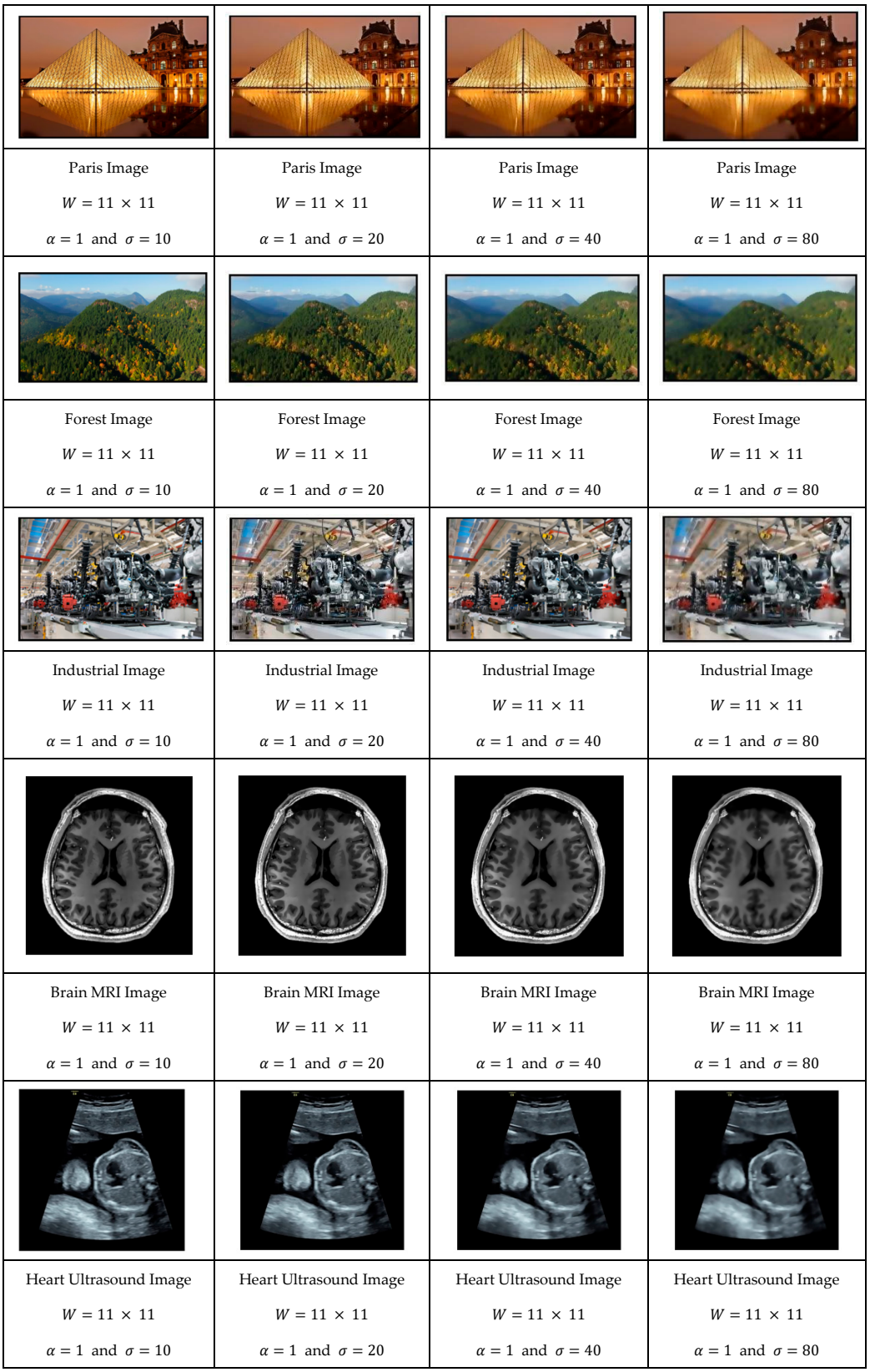


Figure 5. Evolution of the image test database as a function of σ.

As can be observed, the image's smoothness level increased proportionally with σ while maintaining edge clarity. As σ values increased, an optimal value σ^* emerged, corresponding to an ideal scale. We explore this relationship in greater detail later in our discussion.

Let us now study the effect of the parameter α on the filtering results. One can see the evolution of the filtering process of images (House, Hearing Aid, and Barbara) for various values of α. To highlight its effect, we set the σ to large values (80, 80, and 100).

As one can see in Figure 6, increasing the parameter α reduced the influence of the spatial component, resulting in a filter where pixel value differences became the dominant factor rather than spatial proximity. This shift in dominance led to reduced spatial blurring while still preserving important signal variations across the image.

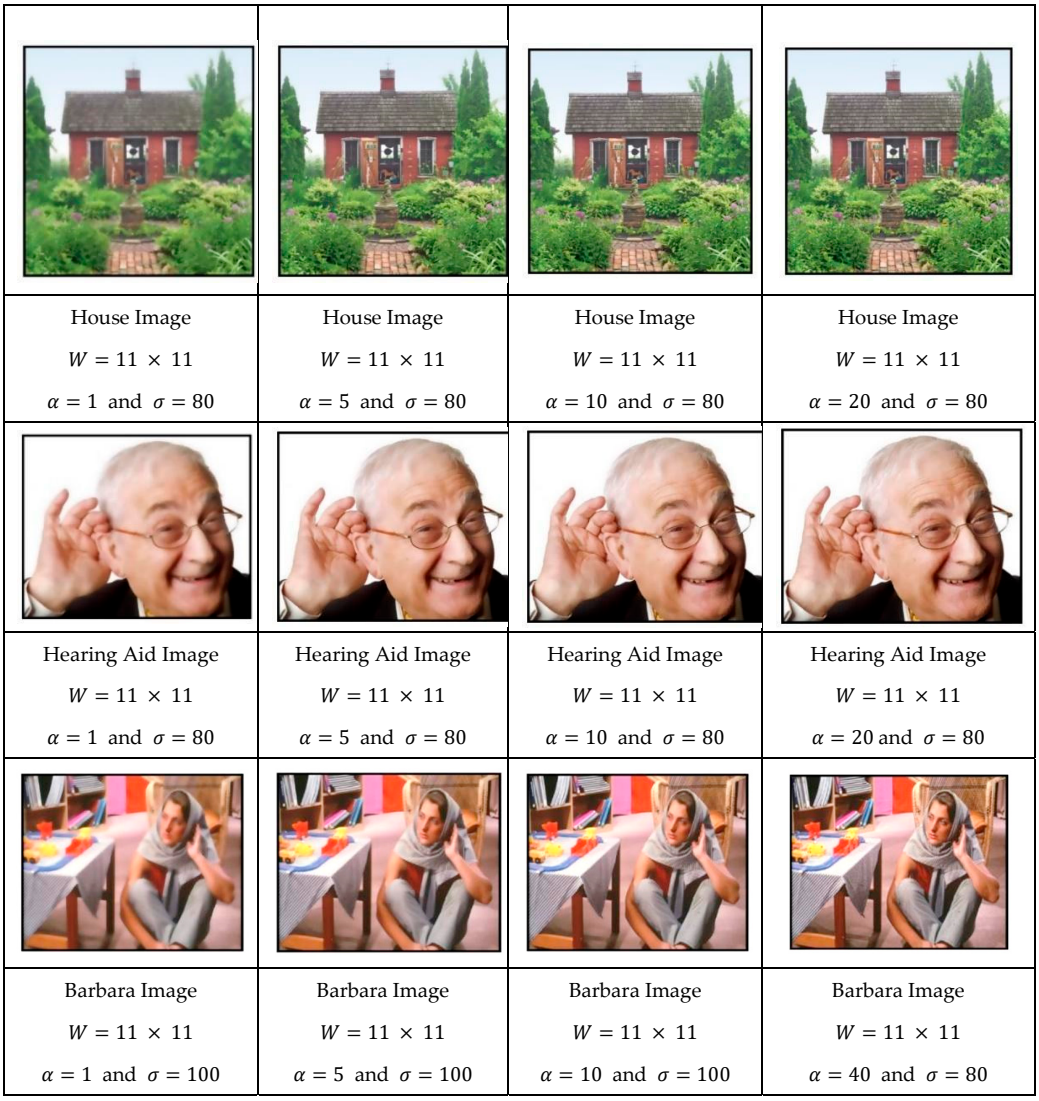
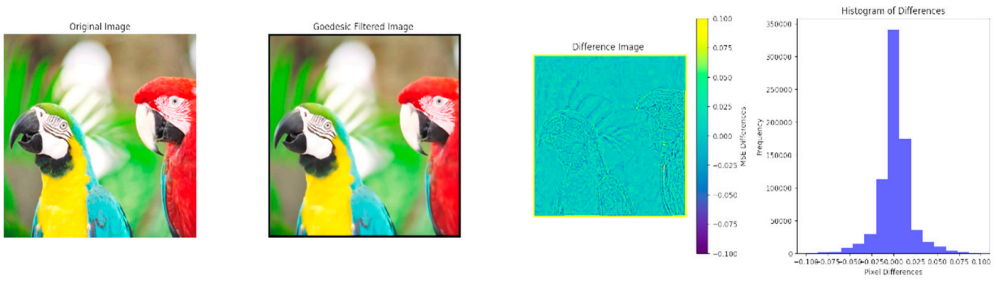


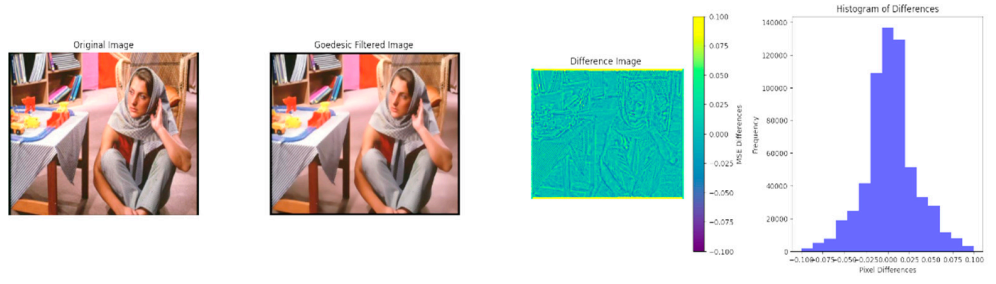
Figure 6. Evolution of the image database as a function of α for a fixed σ.

The next experiment is to illustrate the distribution of the differences between the original image and the filtered one for a window size equal to 11×11 and a filter parameter equal to $\alpha = 1$ and $\sigma = 40$. In Figure 7, one can see the original image, the corresponding filtered image, and the color-coded difference between the two images normalized between -0.1 and 0.1. In addition, one can see for each image a histogram of the error between -0.1 and $+0.1$.

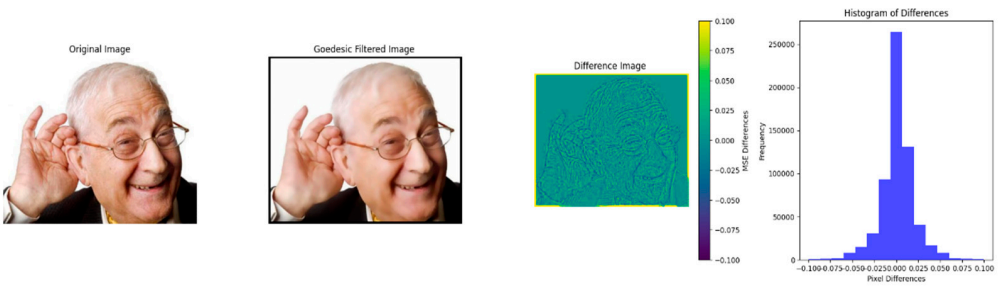
The difference between the original image and the filtered images were very small, even though the filtered image was convolved heavily with $\sigma = 40$. This sequence shows the advantage of geodesic filtering where uniform regions were smoothed out without losing sharp edges.



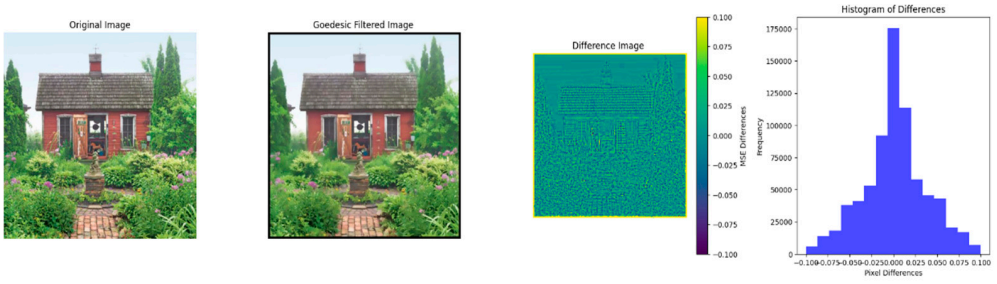
Parrots Image $W = 11 \times 11$, $\alpha = 1$ and $\sigma = 40$



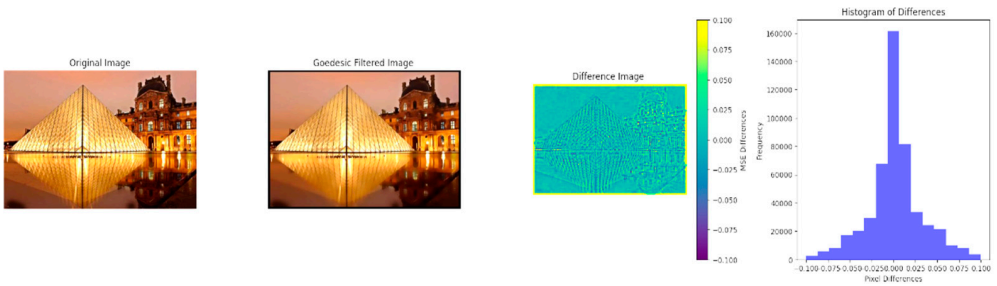
Barbara Image $W = 11 \times 11$, $\alpha = 1$ and $\sigma = 40$



Hearing Aid Image $W = 11 \times 11$, $\alpha = 1$ and $\sigma = 40$



House Image $W = 11 \times 11$, $\alpha = 1$ and $\sigma = 40$



Paris Image $W = 11 \times 11$, $\alpha = 1$ and $\sigma = 40$

Figure 7. *Cont.*

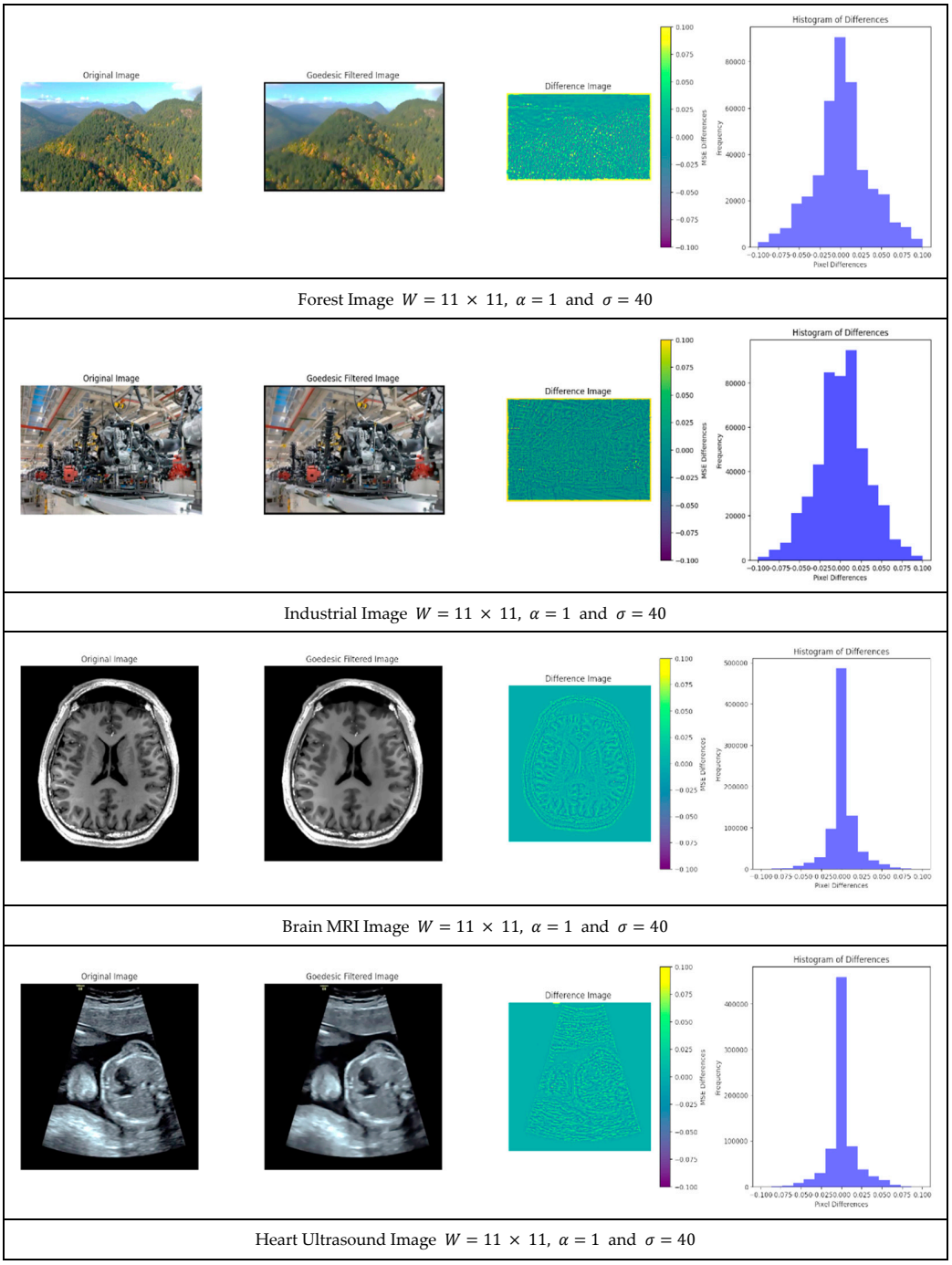


Figure 7. Difference between the original images and the filtered images.

When an image is processed with geodesic filtering, the parameter σ controls the extent of the filtering effect. The "natural scale" represents the specific σ value that achieves the best balance between noise reduction and preservation of significant image features. This concept builds on scale-space theory, which states that images contain features at multiple scales. The natural scale has several key characteristics:

1. Peak Performance Point: As demonstrated in the paper, when plotting PSNR or SSIM values against increasing σ values, there is typically a clear peak before performance declines. This peak identifies the natural scale for that image.
2. Content Dependency: Each image has its own unique natural scale based on its content complexity. Images with fine textures typically have lower optimal σ values, while images with larger homogeneous regions have higher optimal σ values.

3. Feature Preservation Threshold: The natural scale represents the threshold at which the filtering process maximally preserves meaningful edges and structures while still effectively suppressing noise.
4. Adaptive Processing: Rather than applying a fixed σ value across all images, the concept of natural scale suggests that filtering should adapt to each image's inherent structure.
5. Noise-Robust Analysis: When evaluating images with added noise, the natural scale remains relatively stable, showing that it is tied to the underlying image structure rather than noise characteristics.

We begin by establishing baseline performance with controlled noise conditions, then progressively introduce more challenging scenarios to demonstrate the robustness of geodesic filtering. To demonstrate this unique property of geodesic filter, let us study how the image evolves with σ for images that are corrupted by a Gaussian noise with an amplitude between [0, 35] and an average of 0. Following the foundational work of Zhang et al. [29], we performed a peak signal-to-noise ratio (PSNR) analysis for each image as a function σ. PSNR is defined as

$$PSNR = 10 * log_{10}\left(\frac{MAX_i^2}{MSE}\right) \tag{21}$$

where MAX_i is the maximum possible pixel value and MSE is the mean squared error.

In addition, for each σ, we also compute the structural similarity index measure (SSIM) as it provides a deeper insight into structural preservation. SSIM incorporates three components:

- Luminance comparison;
- Contrast comparison;
- Structural correlation is an important criterion to measure edge preservation.
- SSIM is defined as

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{\left(\mu_x^2 + \mu_y^2 + c_1\right)\left(\sigma_x^2 + \sigma_y^2 + c_2\right)} \tag{22}$$

where

- μ_x is the pixel sample mean of x;
- μ_y is the pixel sample mean of y;
- σ_x^2 is the variance of x;
- σ_y^2 is the variance of y;
- $c_1 = (k_1 V), c_2 = (k_2 V)$;
- V the dynamic range of the pixel-values (typically 255);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

Figure 8 shows the original image with Gaussian noise, the filtered version for σ_{max} corresponding to the maximum of the PSNR, and finally the evolution of the PSNR and SSIM as a function of σ. One can see, for each image, the PSNR and SSIM evolve as a function of σ monotonically toward a maximum and then reduce due to over blurring. The σ_{max} value corresponds to the maximum PSNR and is called the natural scale of the image.

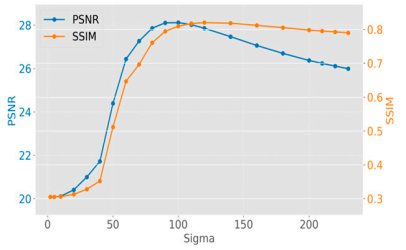
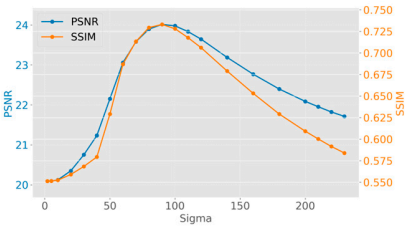
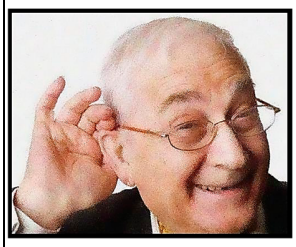
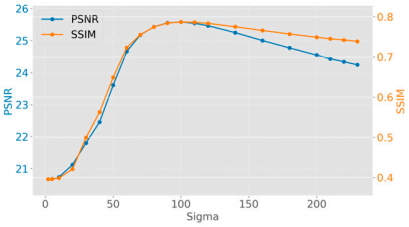
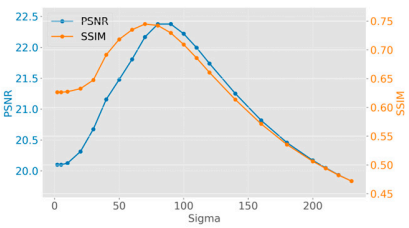
		
Noisy Parrots Image PSNR: 20.07 dB SSIM: 0.2930	Filtered Parrots Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 90$ PSNR: 28.09 dB SSIM: 0.7923	PSNR and SSIM vs. Sigma Curves
		
Noisy Barbara Image PSNR: 20.09 dB SSIM: 0.5358	Filtered Barbara Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 70$ PSNR: 23.99 dB SSIM: 0.7308	PSNR and SSIM vs. Sigma Curves
		
Noisy Hearing Aid Image PSNR: 20.81 dB SSIM: 0.3871	Filtered Hearing Aid Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 70$ PSNR: 25.37 dB SSIM: 0.7728	PSNR and SSIM vs. Sigma Curves
		

Figure 8. *Cont.*

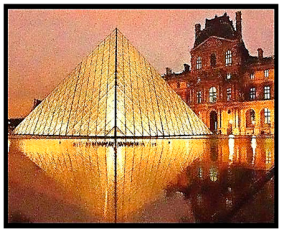
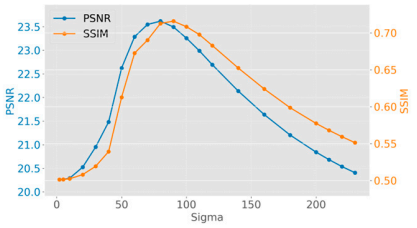
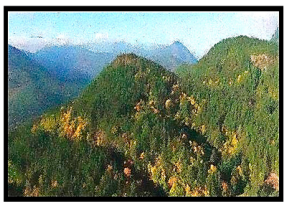
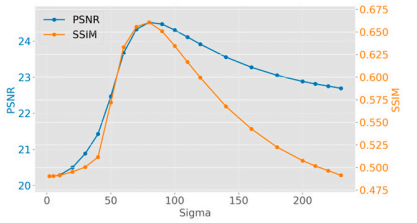
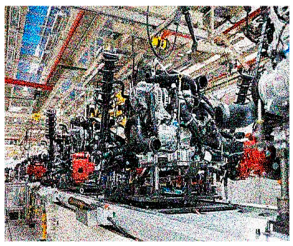
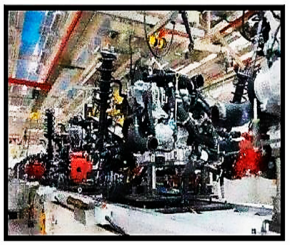
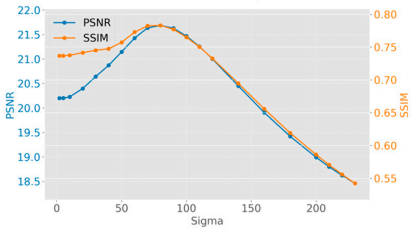
Noisy House Image PSNR: 20.17 dB SSIM: 0.6194	Filtered House Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 60$ PSNR: 22.40 dB SSIM: 0.7423	PSNR and SSIM vs. Sigma Curves
		
Noisy Paris Image PSNR: 20.31 dB SSIM: 0.4850	Filtered Paris Image with Optimal Sigma $s = 17, \alpha = 1$ and $\sigma_{max} = 70$ PSNR: 23.59 dB SSIM: 0.7099	PSNR and SSIM vs. Sigma Curves
		
Noisy Forest Image PSNR: 20.29 dB SSIM: 0.4737	Filtered Forest Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 70$ PSNR: 24.32 dB SSIM: 0.6542	PSNR and SSIM vs. Sigma Curves
		
Noisy Industrial Image PSNR: 20.17 dB SSIM: 0.7266	Filtered Industrial Image with Optimal Sigma $s = 17,\ \alpha = 1$ and $\sigma_{max} = 60$ PSNR: 21.70 dB SSIM: 0.8838	PSNR and SSIM vs. Sigma Curves

Figure 8. *Cont.*

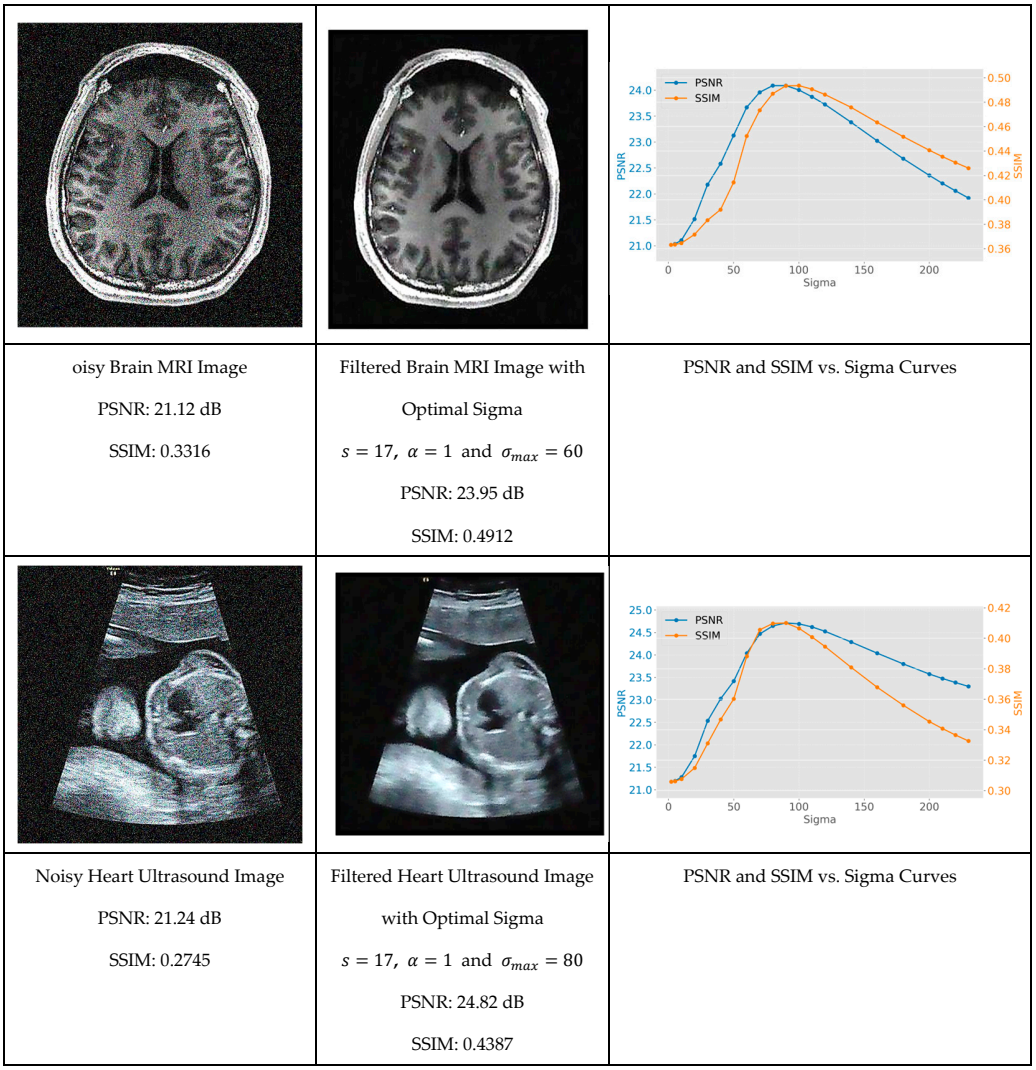


Figure 8. Evolution of the PSNR and SSIM as a function of σ for a noisy version of the dataset images. **Left**: The noisy image. **Center**: The image filtered with the optimum value σ_{max}. **Right**: The evolution of the PSNR and SSIM vs. σ.

The experimental results reveal that while the natural scale parameter varied across different images (with test cases showing a range from σ = 60 to σ = 90), it can be systematically identified through careful analysis of PSNR and SSIM metrics. This discovery provides researchers with a methodical framework for parameter optimization that eliminates the traditional trial-and-error approach commonly required in advanced filtering techniques.

4.3. Comparing Geodesic Filter to Other Algorithms

Previous studies comparing the performance of geodesic filtering and anisotropic diffusion methods have demonstrated the advantages and limitations of each approach. For instance, studies have shown that anisotropic diffusion methods, such as the Perona–Malik technique, are effective at preserving edges but struggle with highly textured or noisy images. While anisotropic diffusion methods have been widely adopted for their edge-preserving capabilities, geodesic filtering offers significant advantages in terms of noise reduction and spatial relationship preservation. A comparative study by Weickert [36] for grey-level images and by Boulanger [23] for range images underscore the potential of geodesic filtering as a superior technique for advanced image processing applications. Other studies by Gousseau et al. [37] compared various anisotropic diffusion methods and highlighted the potential of geodesic filtering in overcoming some of the inherent

limitations of these techniques. Their findings suggest that geodesic filtering can provide better results in terms of both quantitative metrics, such as PSNR and SSIM by Gousseau et al. [37].

This section presents a comprehensive comparative analysis of our geodesic filtering approach against the state-of-the-art methods. The key to this comparison is based on PSNR and SSIM difference metrics. The noisy images shown in Figure 8 were processed using various implementations of the filtering algorithms described in Section 2. To be fair in our comparison, as with the geodesic filter, we tuned the parameters to produce the best PSNR value possible. The results of this comparison are collected in Figures 9 and 10.

Each algorithm was carefully tuned to achieve optimal performance using the same test image database with standardized noise conditions. For each filter, the tuning parameters are as follows:

- Least Median Filter: window size s and tile size s_t;
- Gradient Anisotropic Diffusion: conductance C and number of iterations #I;
- Curvature Anisotropic Diffusion: the mean curvature of the level sets k and number of iterations #I;
- Bilateral Filter: s kernel size, σ_d spatial distance weight, and σ_c color distance weight;
- Gaussian Weighted Wavelet DIP Neural Network: σ_w Gaussian variance, ϵ minimum tile loss, s_t tile size, and s_o tile overlap size.

LMS	GAD	CAD
		
$s = 9$ $s_t = 64$ PSNR: 27.56 dB SSIM: 0.7115	$C = 15$ $\#I = 1000$ PSNR: 23.60 dB SSIM: 0.4458	$k = 30$ $\#I = 10$ PSNR: 28.07 dB SSIM: 0.6933
		
$s = 7$ $s_t = 64$ PSNR: 23.59 dB SSIM: 0.6602	$C = 12$ $\#I = 2000$ PSNR: 21.18 dB SSIM: 0.5447	$k = 20$ $\#I = 10$ PSNR: 24.43 dB SSIM: 0.7057

Figure 9. *Cont.*

		
$s = 7$ $s_t = 64$ PSNR: 26.36 dB SSIM: 0.7223	$C = 12$ $\#I = 1000$ PSNR: 22.67 dB SSIM: 0.5523	$k = 20$ $\#I = 10$ PSNR: 26.05 dB SSIM: 0.7649
		
$s = 7$ $s_t = 64$ PSNR: 21.86 dB SSIM: 0.6493	$C = 12$ $\#I = 1000$ PSNR: 20.88 dB SSIM: 0.6473	$k = 5$ $\#I = 10$ PSNR: 22.62 dB SSIM: 0.6995
		
$s = 7$ $s_t = 64$ PSNR: 22.13 dB SSIM: 0.5842	$C = 13$ $\#I = 500$ PSNR: 21.97 dB SSIM: 0.5242	$k = 5$ $\#I = 10$ PSNR: 22.73 dB SSIM: 0.5940
		
$s = 7$ $s_t = 64$ PSNR: 24.53 dB SSIM: 0.6148	$C = 13$ $\#I = 1000$ PSNR: 21.94 dB SSIM: 0.4713	$k = 5$ $\#I = 10$ PSNR: 24.68 dB SSIM: 0.6322

Figure 9. *Cont.*

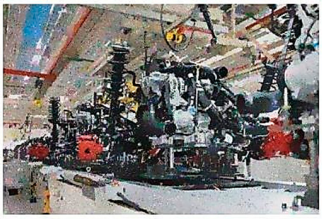
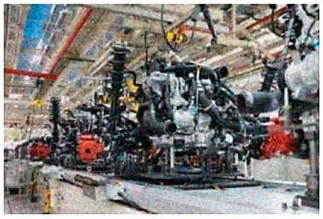
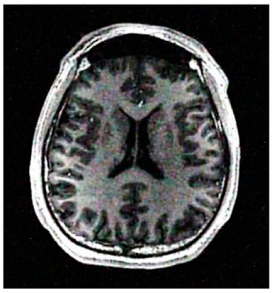
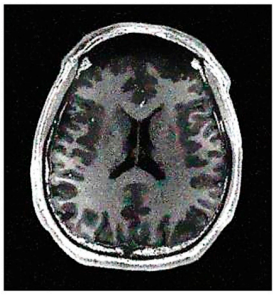
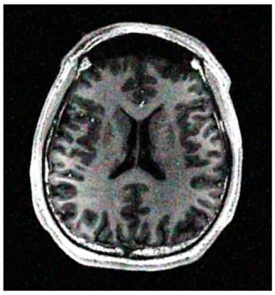
		
$s = 7$ $s_t = 64$ PSNR: 21.72 dB SSIM: 0.7929	$C = 13$ $\#I = 1000$ PSNR: 20.64 dB SSIM: 0.7050	$k = 5$ $\#I = 10$ PSNR: 22.41 dB SSIM: 0.8101
		
$s = 7$ $s_t = 64$ PSNR: 25.66 dB SSIM: 0.4914	$C = 10$ $\#I = 1000$ PSNR: 22.36 dB SSIM: 0.3463	$k = 10$ $\#I = 10$ PSNR: 24.92 dB SSIM: 0.4859
		
$s = 7$ $s_t = 64$ PSNR: 26.22 dB SSIM: 0.4395	$C = 10$ $\#I = 1000$ PSNR: 22.60 dB SSIM: 0.2916	$k = 10$ $\#I = 10$ PSNR: 25.37 dB SSIM: 0.4322

Figure 9. Optimal filtering results for LMS, gradient anisotropic diffusion, and curvature anisotropic algorithms.

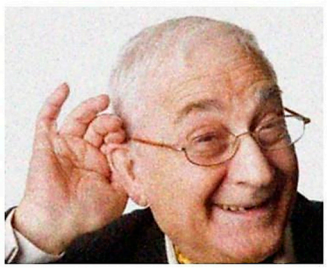
Bilateral Filter	DIP
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 27.91 dB SSIM: 0.6694	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 24.61 dB SSIM: 0.5286
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 24.51 dB SSIM: 0.7055	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 20.38 dB SSIM: 0.6235
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 26.11 dB SSIM: 0.7539	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 20.90 dB SSIM: 0.7098

Figure 10. *Cont.*

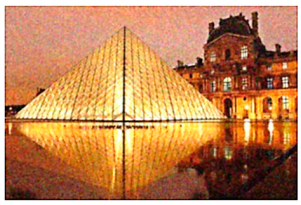
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 22.82 dB SSIM: 0.7116	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 20.59 dB SSIM: 0.7117
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 23.22 dB SSIM: 0.6382	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 19.31 dB SSIM: 0.5195
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 25.28 dB SSIM: 0.6679	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 22.18 dB SSIM: 0.6121
	
σ_d =1.0 σ_c = 100.0 s = 11 PSNR: 22.70 dB SSIM: 0.8264	σ_w = 10 ϵ = 0.002 s_t = 64 s_o = 8 PSNR: 19.72 dB SSIM: 0.7295

Figure 10. *Cont.*

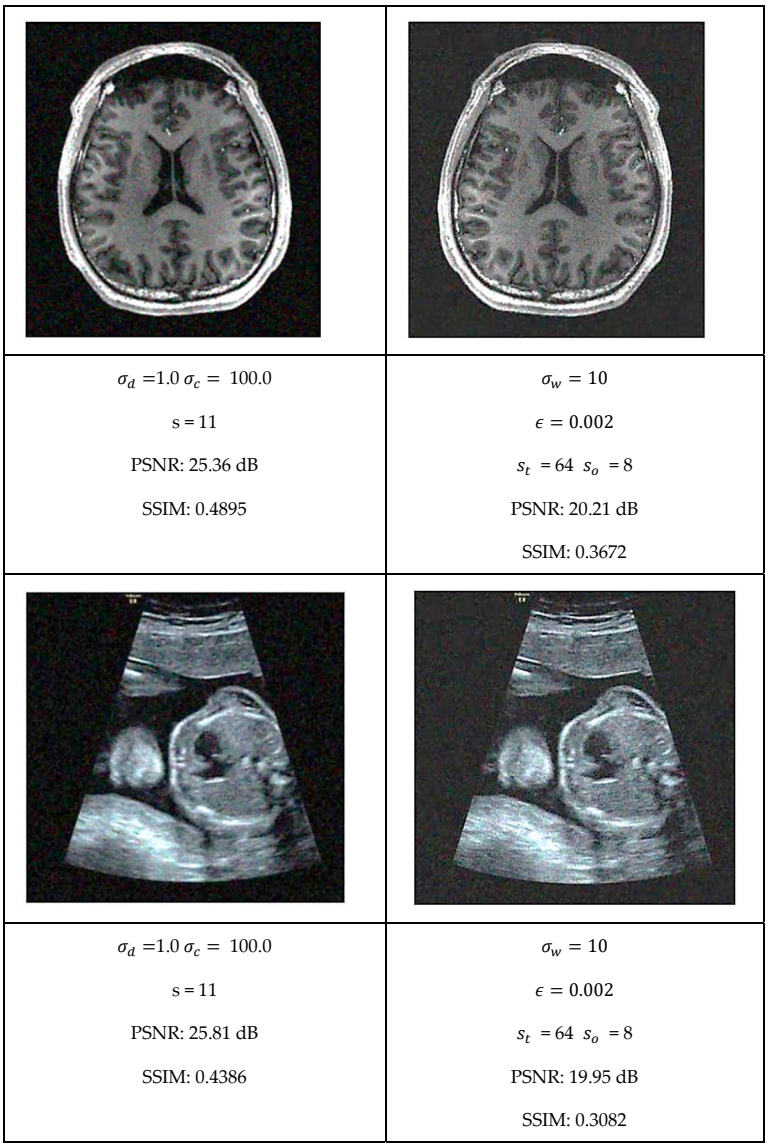


Figure 10. Optimal filtering results for bilateral and GW-DIP algorithms.

In summary, the quantitative evaluation presented in Tables 1 and 2 confirms that geodesic filtering provides dual performance advantages: achieving noise reduction metrics (PSNR) comparable to state-of-the-art alternatives while significantly outperforming them in structural preservation (SSIM). This superiority in preserving image structure is evident across the entire image dataset, with particularly notable advantages in regions containing intricate textures and fine details that traditional methods frequently over-smooth or distort.

The results clearly demonstrate that while other filtering approaches may achieve similar noise reduction performance, they do so at the cost of structural integrity. Geodesic filtering, in contrast, maintains the delicate balance between noise suppression and feature preservation, making it particularly valuable for applications where preserving the semantic content of images is paramount.

Furthermore, geodesic filtering shows remarkable resilience when processing non-standard noise distributions, including speckle patterns and mixed noise profiles that typically challenge conventional filters. This versatility extends its practical utility across diverse application domains from medical imaging to remote sensing.

Table 1. Comparison of the PSNR for each filtering algorithm.

Image	Noisy Image	Geodesic	LMS	GAD	CAD	Bilaterial	DIP
Parrot	20.07 dB	28.09 dB	27.56 dB	23.60 dB	28.07 dB	27.91 dB	24.61 dB
Barbara	20.09 dB	23.99 dB	23.59 dB	21.18 dB	24.43 dB	24.51 dB	20.38 dB
Hearing Aid	20.81 dB	25.37 dB	26.36 dB	22.67 dB	26.05 dB	26.11 dB	20.90 dB
House	20.17 dB	22.40 dB	21.86 dB	20.88 dB	22.62 dB	22.82 dB	20.59 dB
Paris	20.31 dB	23.59 dB	22.13 dB	21.97 dB	22.73 dB	23.22 dB	19.31 dB
Forest	20.29 dB	24.32 dB	24.53 dB	21.94 dB	24.68 dB	25.28 dB	22.18 dB
Industrial	20.17 dB	21.70 dB	21.72 dB	20.64 dB	22.41 dB	22.70 dB	19.72 dB
Brain MRI	21.12 dB	23.95 dB	25.66 dB	22.36 dB	24.92 dB	25.36 dB	20.21 dB
Ultrasound	21.24 dB	24.82 dB	26.22 dB	22.60 dB	25.37 dB	25.81 dB	19.95 dB
Average PSNR	20.47 dB	24.25 dB	24.40 dB	21.98 dB	24.59 dB	24.85 dB	20.87 dB

Table 2. Comparison of the SSIM for each filtering algorithm.

Image	Noisy Image	Geodesic	LMS	GAD	CAD	Bilaterial	DIP
Parrot	0.2930	0.7923	0.7115	0.4458	0.6933	0.6694	0.5286
Barbara	0.5358	0.7308	0.6602	0.5447	0.7057	0.7055	0.6235
Hearing Aid	0.3871	0.7728	0.7223	0.5523	0.7649	0.7539	0.7098
House	0.6194	0.7423	0.6493	0.6473	0.6995	0.7116	0.7117
Paris	0.4850	0.7099	0.5842	0.5242	0.5940	0.6382	0.5195
Forest	0.4737	0.6542	0.6148	0.4713	0.6322	0.6679	0.6121
Industrial	0.7266	0.8838	0.7929	0.7050	0.8101	0.8264	0.7295
Brain MRI	0.3316	0.4912	0.4914	0.3463	0.4859	0.4895	0.3672
Ultrasound	0.2745	0.4387	0.4395	0.2916	0.4322	0.4386	0.3082
Average SSIM	0.4585	0.6907	0.6296	0.5032	0.6464	0.6556	0.5678

5. Geodesic Filtering Computational Complexity and GPU Implementation

The computational complexity of geodesic filtering represents one of its most significant challenges, which necessitates the GPU implementation described in the paper. This aspect deserves thorough examination as it directly impacts the algorithm's practical applicability. The computational bottleneck occurs specifically during the minimal path calculation (geodesic distance) between pixels. For each central pixel, the algorithm must compute the shortest path to every other pixel in the window, considering both spatial proximity and intensity differences. This requires solving multiple single-source shortest path problems, which have a complexity of O ($M \cdot Ns^2$) per window when using Dijkstra's algorithm with a binary heap.

Like the algorithms described in Asad et al. [38] and Áfra et al. [39], our GPU implementation addresses many computational challenges through several innovative approaches:

1. Parallelization Strategy: By processing multiple image tiles simultaneously, the GPU implementation exploits the inherent parallelism in the algorithm. Each thread processes a single pixel, allowing thousands of pixels to be processed concurrently.
2. Memory Optimization: The paper describes sophisticated memory access patterns and bank conflict resolution techniques that significantly improve throughput on GPU architectures.
3. Wave Propagation Algorithm: The fast-marching method (FMM) implemented on GPU provides a more memory-access-friendly alternative to Dijkstra's algorithm, eliminating the need for priority queues which cause memory bank conflicts.
4. Coalesced Memory Access: By arranging data in memory to enable coalesced access patterns, the implementation achieves near-optimal memory bandwidth utilization.

5.1. Memory Optimization

1. Image Tiling: A large image is divided into 16 × 16 tiles for processing, where each tile is handled by one thread block. A 7 × 7 convolution window requires an overlap of three pixels on each side ($\lfloor 7/2 \rfloor = 3$), resulting in an effective area processed by each tile to be 10 × 10 pixels (16-3-3).
2. Memory Layout: The input image is stored as RGB values (three channels) in global memory, where each pixel requires three float values. The image data are aligned for faster coalesced memory access from the threads. For each block, the threads read the data from global memory into the shared memory. Since each tile is 16 × 16, each block will read a tile of size 22 × 22 pixels (16 + 3 + 3 for each dimension) of three floats. This also includes halo regions. In addition, additional space is used in shared memory for distance computations.

5.1.1. Block Execution Flow

1. Tile Loading Phase: Each 16 × 16 thread block loads the tile data (16 × 16), the halo region (three pixels each side), using coalesced reading from global memory.
2. Convolution Processing: Each thread is responsible for one pixel in 16 × 16 tiles, and threads near edges handle the halo region. First, a distance array is initialized. Each output pixel requires 7 × 7 window computation. The center pixel of each window is determined by thread ID. For a thread in the block,
3. The convolution window is centered at the current thread position;
4. Each thread processes the 7 × 7 window around center, which include
5. Computing geodesic distances within the window using the wave propagation algorithm;
6. Calculating the weight of the spatial and signal contributions using a Gaussian function;
7. Normalize the weights between zero and one;
8. Multiply the signal with the weights and the pixel by the weighted sum.

5.1.2. Memory Access Optimization

The main optimization strategy is based on better use of memory access. First is a coalesced loading strategy where each thread loads one RGB pixel set 128-byte aligned access and where sequential thread IDs are mapped to sequential memory. For example, data are stored in global memory as $[R_0]$ $[G_0]$ $[B_0]$ $[R_1]$ $[G_1]$ $[B_1] \ldots [R_{15}]$ $[G_{15}]$ $[B_{15}]$. Each thread in a half-warp read the data as follows:

- T0 → P0, P16, P32 (loads three pixels);
- T1 → P1, P17, P33;
- T2 → P2, P18, P34;
- . . .;

- T15 → P15, P31, P47.

The second data load consists of

- Loading halo region corresponding to an additional three pixels on each side;
- Maintain a coalesced pattern where possible;
- Handle boundary conditions near the edge of the image.

All the data for a tile of (22 × 22) are then stored into much faster shared memory as follows:

Halo[Main Tile] [Halo];

- 3 + 16 + 3 = 22 columns;
- 3 + 16 + 3 = 22 rows.

It is stored as follows:

- [H H H | M M M M M M M M M M M M M M M M | H H H];
- [H H H | M M M M M M M M M M M M M M M M | H H H];
- [H H H | M M M M M M M M M M M M M M M M | H H H].

The main objective is to minimize memory bank conflict between threads in a block. Bank conflicts can cause 32× slowdown, as each conflict serializes access and also affects warp execution efficiency. First, we must ensure that each thread accesses different banks bank (T_{xy}) ≠ bank ($T_{x'y'}$) for any threads in the same warp. In our implementation, we used a column padding strategy where we added an extra column to the array to shift row starting addresses to ensure bank separation. Our implementation uses channel padding where we add an extra column to each channel:

- float r_channel [22,23];
- float g_channel [22,23];
- float b_channel [22,23].

This simplified bank mapping allows independent channel access that reduces conflict probability and better memory coalescing. This simple strategy eliminated bank conflicts to improve parallel memory operations of the full warp utilization. There is no serialization allowing for concurrent bank access and improved throughput.

5.2. Fast-Marching Method Algorithm (FMM)

The fast-marching method (FMM) represents a watershed advancement in geodesic distance computation, pioneered by Sethian's [31] foundational work and subsequently refined for parallel architectures. At its core, FMM employs wavefront propagation—systematically expanding distance information outward from source points while maintaining strict causality principles essential for computational accuracy. This approach begins with a meticulous initialization phase that establishes the computational infrastructure through carefully configured distance maps and status markers for each vertex within the processing window. This preparatory stage, whose importance Peyré's [27] research emphasized, creates the necessary foundation for subsequent processing steps. The algorithm's defining characteristic lies in its disciplined processing sequence: examining points in strict order of increasing distance to ensure each vertex's final distance value is definitively established before dependent calculations proceed. The causality-preserving ordering mechanism, whose mathematical properties Kimmel and Sethian [25] extensively analyzed, provide crucial guarantees regarding the precision of computed geodesic distances. This ordered processing framework enables FMM to efficiently resolve the Eikonal equation governing geodesic distance propagation, making it particularly amenable to GPU implementation through its structured memory access patterns and predictable data

dependencies—characteristics that dramatically contrast with Dijkstra's algorithm's reliance on priority queues.

The FMM algorithm consists of several key components:

1. Wave Structure: For our example, four concentric waves need to be processed by a thread. Each wave represents the front of pixels at similar geodesic distances that expand outward from a source point. Each wave elements are stored in shared memory contains position, distance, and color information.
2. Distance Update Mechanism: For each pixel in the current wave, the algorithm examines all 8-connected neighbors and then computes the geodesic distance combing spatial and color distances using Equation (14). The algorithm then updates the distance if a shorter path is found. All updates use atomic operators to avoid race conflicts between threads.
3. Wave Evolution: Initially, the wave inside a convolution window starts at the source point and then propagates outward. The wave size adapts based on local image properties. The process continues until all pixels in the window are reached. Each thread in a block examines its 8-connected neighbors and then computes a new distance if a shorter is found. Then, the neighbor is marked for inclusion in the next wave if necessary. See Figure 11 for an illustration of this process.

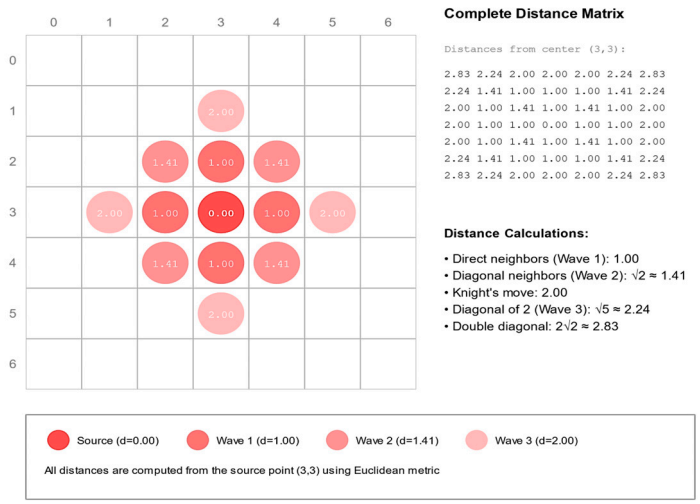


Figure 11. Wave front calculation for a 7×7 window.

The FMM computational complexity $O(S)$ is less efficient than the Dijkstra algorithm. On the other hand, the memory access is more efficient for parallel implementation as it is not random.

5.3. Speed Comparison Between Python and CUDA Implementations

This GPU implementation-based wave propagation is more efficient and more adapted to GPU processing than the CPU implantation using Dijkstra's algorithm. For Dijkstra's algorithm for 7×7 windows:

- Nodes $(N) = 49$ (7×7);
- Time Complexity: $O(N\ log_2\ N)$;
- For each pixel: $49 \times log_2$ (49) operations to compute the geodesic distance;
- More calculation is also needed to maintain the priority queue.

For the wave propagation algorithm,

- We need to compute four waves (center, first ring, second ring, outer);
- Operations per wave: $O(N)$;

- Fixed number of steps, regardless of window content.

For a single 7 × 7 window, the Dijkstra algorithm requires

- Operations: ~280 (49 × log_2 (49);
- Forty-nine sequential steps;
- Random memory access due to the priority queue;
- Extra storage for the priority queue.

For the same window size, the wave propagation algorithm requires

- Operations: ~196 (49 × 4 waves);
- Structured memory access;
- Extra storage for the wave buffers.

5.4. Execution Speed Comparison

A comparison between a CPU implementation written in Python 3.8 and a GPU implementation written using CUDA Toolkit version CUDA 12.9 was performed. We used a powerful CPUs to compare with two GPUs running in the Google Colab environment. The CPU is the AMD Threadripper 7980X with the following specifications:

- Cores: 64;
- Threads: 128;
- Base Clock: 3.2 GHz;
- Boost Clock: 5.1 GHz;
- Memory Bandwidth: ~200 GB/s;
- L3 Cache: 384 MB.

The two GPUs used for the comparison were

1. the NVIDIA A100 GPU: CUDA Cores: 6912;
2. Memory: 80 GB HBM2e;
3. Memory Bandwidth: 2039 GB/s;
4. Base Clock: 1410 MHz.

and the

1. NVIDIA T4 GPU:
2. CUDA Cores: 2560;
3. Memory: 16 GB GDDR6;
4. Memory Bandwidth: 320 GB/s;
5. Base Clock: 585 MHz.

Performance metrics for 1024 × 1024 color image for the Python Dijkstra implementation running on the CPU:

- Pure Python: ~45,000 ms;
- NumPy/SciPy: ~12,000 ms;
- Numba JIT [*]: ~3000 ms;
- Numba Parallel [*]: ~800 ms.

For the CUDA C Wave Propagation version running on a NVIDIA A100 GPU:

- Processing: ~3 ms;
- Memory Transfer: ~1 ms;
- Total: ~4 ms;
- Speedup vs. Numba: ~200×.

For a NVIDIA T4 GPU:

- Processing: ~15 ms;
- Memory Transfer: ~3 ms;

- Total: ~18 ms;
- Speedup vs. Numba: ~44×.

The implementation achieves significant performance improvements through wave-specific optimizations, dynamic bank conflict resolution, and adaptive memory access patterns. Experimental results demonstrate a speedup of 200 times for the NVIDIA A100 GPU and 44 times for the NVIDIA T4 GPU compared to a multithread execution on the CPU using Numba Parallel.

5.5. Scalability Between Algorithms

Scalability is also good as illustrated by the following statistics. For the Python Dijkstra Processing using Numba Parallel, implementation execution time vs. image size are

- 512 × 512: ~200 ms;
- 1024 × 1024: ~800 ms;
- 2048 × 2048: ~3200 ms;
- Scale factor: ~4×.

For CUDA Wave Propagation (on a A100 NVIDIA GPU):

- 512 × 512: ~1 ms;
- 1024 × 1024: ~4 ms;
- 2048 × 2048: ~16 ms;
- Scale factor: ~4×.

CUDA Wave Propagation (on a T4 NVIDIA GPU):

- 512 × 512: ~4.5 ms;
- 1024 × 1024: ~18 ms;
- 2048 × 2048: ~72 ms;
- Scale factor: ~4×.

5.6. Accuracy Between CPU and GPU Versions

The CPU (Python Dijkstra) implementation uses 64-bit double precision by default and is our gold standard reference to compute geodesic distances. For the GPU Wave Propagation implementation, each CUDA core has a single precision accuracy meaning that the final distance relative error to the gold standard will be ~10^{-6}. So, for our test images the average error are:

- CPU Dijkstra: 0.0 (reference);
- GPU Wave (A100): 3.1×10^{-6};
- GPU Wave (T4): 3.2×10^{-6}.

6. Conclusions

This paper has presented a comprehensive analysis of geodesic filtering within the Riemannian framework for image processing, offering both a rigorous theoretical foundation and extensive experimental validation. Our investigation demonstrates that by modeling images as high-dimensional manifolds and computing minimal geodesic paths, this approach achieves remarkable improvements in noise reduction and edge preservation. Geodesic filtering delivers noise reduction on par with state-of-the-art alternatives measured using PSNR while significantly outperforming them in structural preservation measured by SSIM. The robust differential geometry underpinning geodesic filtering enables it to adeptly handle complex image structures, including challenging scenarios with varying noise characteristics and intricate edge patterns.

The experimental data clearly validate that GPU acceleration is essential for practical geodesic filtering, with NVIDIA A100 hardware delivering a 200× performance boost over

optimized CPU implementations. This dramatic acceleration transforms geodesic filtering from a mere theoretical concept into a viable processing technique for real-world applications. By reducing computational barriers, the GPU implementation enables geodesic filtering to be effectively applied to large-scale datasets and time-sensitive processing requirements across medical imaging, remote sensing, and video processing domains.

Furthermore, our analysis reveals that the geometric approach to image filtering offers fundamental advantages beyond performance metrics alone. By respecting the intrinsic geometry of image content, geodesic filtering preserves semantic information that is often lost in traditional filtering approaches. The preservation of fine structural details, particularly along complex curved boundaries and in textured regions, maintains the diagnostic or analytical value of processed images—a critical consideration in domains where visual information directly informs decision-making processes. The adaptive nature of geodesic distances, which automatically adjust to local image characteristics, provides a self-regulating mechanism that reduces the need for parameters tuning across diverse image types. This adaptability proves especially valuable when processing heterogeneous datasets with varying noise profiles and structural complexities. Our experiments with clinical medical images, satellite imagery, and natural photographs demonstrate this versatility, with consistent performance advantages observed across these diverse domains.

Additionally, the theoretical framework established in this paper opens new avenues for further research, including potential extensions to temporal filtering for video sequences, integration with deep learning architectures as a geometrically informed processing layer, and application to higher-dimensional volumetric data common in modern medical imaging. The mathematical formalism of Riemannian geometry provides a robust foundation for these future developments, suggesting that geodesic filtering represents not just an incremental improvement but a fundamentally different paradigm for approaching image processing challenges.

7. Future Research Directions

Looking forward, several promising avenues exist to further enhance and expand the capabilities of geodesic filtering. A primary focus will be on reducing its computational complexity while maintaining its inherent advantages. Advances in parallel computing—such as multi-GPU architectures and emerging hardware accelerators—paired with algorithmic improvements inspired by anisotropic fast marching methods, may deliver the necessary efficiency gains.

Another exciting direction involves extending the geodesic filtering framework to higher-dimensional data. While our current work concentrates on two-dimensional images, early studies indicate that the approach can be naturally generalized to 3D volumetric datasets—such as CT or MRI scans—by mapping from R^3 to R^4 (e.g., u, v, w to x, y, z, I). Similarly, the method could be adapted for video processing by extending the mapping to higher dimensions (e.g., R^3 to R^6 for video sequences, encompassing spatial and temporal dimensions along with color information) and even 4D time-varying datasets (e.g., dynamic CT or ultrasound data).

These extensions will not only widen the applicability of geodesic filtering but also create new research opportunities in the analysis and visualization of complex, high-dimensional data. Many of these extensions of the geodesic filtering approach have been successfully tested in our laboratory and will be published in the near future.

Author Contributions: Both authors did programming, testing, and wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Sciences and Engineering Research Council of Canada Discovery.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [CrossRef]
2. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep Image Prior. *Int. J. Comput. Vis.* **2018**, *128*, 1405–1573. [CrossRef]
3. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 629–639. [CrossRef]
4. Weickert, J.; ter Haar Romeny, B.M.; Viergever, M.A. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Process.* **1998**, *7*, 398–410. [CrossRef]
5. Black, M.J.; Sapiro, G.; Marimont, D.H.; Heeger, D. Robust Anisotropic Diffusion. *IEEE Trans. Image Process.* **1998**, *7*, 421–432. [CrossRef] [PubMed]
6. Alvarez, L.; Lions, P.L.; Morel, J.M. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.* **1992**, *29*, 845–866. [CrossRef]
7. Sapiro, G.; Tannenbaum, A.; You, Y.-L.; Kaveh, M. Experiments on geometric image enhancement. In Proceedings of the1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Published in IEEE Computer Society Press: Los Alamitos, CA, USA, 1994; Volume 2, pp. 472–476.
8. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the IEEE International Conference on Computer Vision, Bombay, India, 7 January 1998; pp. 839–846.
9. Durand, F.; Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, San Antonio, TX, USA, 23–26 July 2002; ACM SIGGRAPH: New York, NY, USA, 2002; pp. 257–266.
10. Paris, S.; Durand, F. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vis.* **2009**, *81*, 24–52. [CrossRef]
11. Kaplan, N.H.; Erer, I.; Gulmus, N. Remote sensing image enhancement via bilateral filtering. In Proceedings of the 8th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 19–22 June 2017; pp. 139–142.
12. Rousseeuw, P.J. Least median of squares regression. *J. Am. Stat. Assoc.* **1984**, *79*, 871–880. [CrossRef]
13. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; John Wiley & Sons: Hoboken, NJ, USA, 1987.
14. Gonzalez, R.; Woods, R. *Digital Image Processing*, 4th ed.; Pearson: London, UK, 2020.
15. Shao, L.; Yan, R. Robust Point Cloud Processing Using Statistical Outlier Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2883–2898.
16. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
17. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; Wiley Series in Probability and Statistics: Hoboken, NJ, USA, 2003.
18. Shocher, A.; Cohen, N.; Irani, M. "Zero-Shot" Super-Resolution using Deep Internal Learning. In Proceedings of the CVPR, Salt Lake City, UT, USA, 18–22 June 2018.
19. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image Inpainting for Irregular Holes Using Partial Convolutions. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
20. Yang, Y.; Sun, J.; Zhang, Q. Deep Wavelet Network with Short Connections for Image Restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
21. Liu, Y.; Wang, Z.; Fang, X. Wavelet-Based Deep Image Prior for Image Restoration. *IEEE Trans. Image Process.* **2020**.
22. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Deep Image Prior with Gaussian Weighted Wavelets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7765–7773.
23. Boulanger, P. Extraction Multiechelle d'Element Geometrique. Ph.D. Thesis, University of Montreal, Montreal, QC, CA, 1994; pp. 9–37.
24. Sochen, N.; Kimmel, R.; Malladi, R. A general framework for low level vision. *IEEE Trans. Image Process.* **1998**, *7*, 310–318. [CrossRef]

25. Kimmel, R.; Malladi, R.; Sochen, N. Images as embedded maps and minimal surfaces. *Int. J. Comput. Vis.* **2000**, *39*, 111–129. [CrossRef]
26. Mémoli, F.; Sapiro, G. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J. Comput. Phys.* **2005**, *173*, 730–764. [CrossRef]
27. Peyré, G. Manifold models for signals and images. *Comput. Vis. Image Underst.* **2009**, *113*, 249–260. [CrossRef]
28. Castaño-Moraga, C.A.; Lenglet, C.; Deriche, R.; Ruiz-Alzola, J. A Riemannian approach to anisotropic filtering of tensor fields. *Signal Process.* **2007**, *87*, 263–276. [CrossRef]
29. Zhang, F.; Hancock, E.R. New Riemannian techniques for directional and tensorial image data. *Pattern Recognit.* **2010**, *43*, 1590–1606. [CrossRef]
30. Alonso-González, A.; López-Martínez, C.; Salembier, P.; Deng, X. Bilateral Distance Based Filtering for Polarimetric SAR Data. *Remote Sens.* **2013**, *5*, 5620–5641. [CrossRef]
31. Sethian, J.A. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*; Cambridge University Press: Cambridge, UK, 1999.
32. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math. 1959* **1999**, *1*, 269–271. [CrossRef]
33. Mirebeau, J.M. Anisotropic fast marching on cartesian grids using lattice basis reduction. *SIAM J. Numer. Anal.* **2014**, *52*, 1573–1599. [CrossRef]
34. Vincent, L. Minimal path algorithms for the robust detection of linear features in gray images. In *International Symposium on Mathematical Morphology and Its Applications to Image and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 1998.
35. Lewis, R. A Comparison of Dijkstra's Algorithm Using Fibonacci Heaps, Binary Heaps, and Self-Balancing Binary Trees. *arXiv* **2023**. [CrossRef]
36. Weickert, J. *Anisotropic Diffusion in Image Processing*; ECMI Series; Teubner-Verlag: Stuttgart, Germany, 1998.
37. Gousseau, Y.; Morel, J.M. Are natural images of bounded variation? *SIAM J. Math. Anal.* **2001**, *33*, 634–648. [CrossRef]
38. Asad, M.; Dorent, R.; Vercauteren, T. FastGeodis: Fast Generalised Geodesic Distance Transform. *J. Open-Source Softw.* **2020**, *7*, 4532. [CrossRef]
39. Áfra, A.T.; Wald, I.; Benthin, C.; Woop, S. gDist: Efficient Distance Computation between 3D Meshes on GPU. In *SIGGRAPH Asia 2024 Conference Papers*; Association for Computing Machinery: New York, NY, USA, 2024.

Journal of *Imaging*

MDPI

Article

Segmentation of Non-Small Cell Lung Carcinomas: Introducing DRU-Net and Multi-Lens Distortion

Soroush Oskouei [1,2,*], Marit Valla [3,4], André Pedersen [3,5,6], Erik Smistad [1,7], Vibeke Grotnes Dale [3,8], Maren Høibø [3,4], Sissel Gyrid Freim Wahl [8], Mats Dehli Haugum [8], Thomas Langø [7,9], Maria Paula Ramnefjell [10,11], Lars Andreas Akslen [10,11], Gabriel Kiss [9,12] and Hanne Sorger [1,2]

1 Department of Circulation and Medical Imaging, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway
2 Clinic of Medicine, Levanger Hospital, Nord-Trøndelag Health Trust, NO-7600 Levanger, Norway
3 Department of Clinical and Molecular Medicine, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway
4 Clinic of Laboratory Medicine, St. Olavs Hospital, Trondheim University Hospital, NO-7030 Trondheim, Norway
5 Clinic of Surgery, St. Olavs Hospital, Trondheim University Hospital, NO-7030 Trondheim, Norway
6 Application Solutions, Sopra Steria, NO-7010 Trondheim, Norway
7 Department of Health Research, SINTEF Digital, NO-7465 Trondheim, Norway
8 Department of Pathology, St. Olavs Hospital, Trondheim University Hospital, NO-7030 Trondheim, Norway
9 Center for Innovation, Medical Devices and Technology, Research Department, St. Olavs Hospital, Trondheim University Hospital, NO-7491 Trondheim, Norway
10 Centre for Cancer Biomarkers CCBIO, Department of Clinical Medicine, University of Bergen, NO-5007 Bergen, Norway
11 Department of Pathology, Haukeland University Hospital, NO-5020 Bergen, Norway
12 Department of Computer Science, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway
* Correspondence: soroush.oskouei@ntnu.no

Abstract: The increased workload in pathology laboratories today means automated tools such as artificial intelligence models can be useful, helping pathologists with their tasks. In this paper, we propose a segmentation model (DRU-Net) that can provide a delineation of human non-small cell lung carcinomas and an augmentation method that can improve classification results. The proposed model is a fused combination of truncated pre-trained DenseNet201 and ResNet101V2 as a patch-wise classifier, followed by a lightweight U-Net as a refinement model. Two datasets (Norwegian Lung Cancer Biobank and Haukeland University Lung Cancer cohort) were used to develop the model. The DRU-Net model achieved an average of 0.91 Dice similarity coefficient. The proposed spatial augmentation method (multi-lens distortion) improved the Dice similarity coefficient from 0.88 to 0.91. Our findings show that selecting image patches that specifically include regions of interest leads to better results for the patch-wise classifier compared to other sampling methods. A qualitative analysis by pathology experts showed that the DRU-Net model was generally successful in tumor detection. Results in the test set showed some areas of false-positive and false-negative segmentation in the periphery, particularly in tumors with inflammatory and reactive changes. In summary, the presented DRU-Net model demonstrated the best performance on the segmentation task, and the proposed augmentation technique proved to improve the results.

Keywords: lung carcinoma; digital pathology; tumor segmentation; deep learning; data augmentation

Academic Editor: Ebrahim Karami

Received: 27 March 2025
Revised: 3 May 2025
Accepted: 13 May 2025
Published: 20 May 2025

Citation: Oskouei, S.; Valla, M.; Pedersen, A.; Smistad, E.; Dale, V.G.; Høibø, M.; Wahl, S.G.F.; Haugum, M.D.; Langø, T.; Ramnefjell, M.P.; et al. Segmentation of Non-Small Cell Lung Carcinomas: Introducing DRU-Net and Multi-Lens Distortion. *J. Imaging* **2025**, *11*, 166. https://doi.org/10.3390/jimaging11050166

1. Introduction

Early diagnosis of lung cancer is crucial for patient survival [1]. Although physical examinations and medical imaging are included in the diagnostic work-up, tissue samples are needed to establish a cancer diagnosis. The histopathological diagnosis, including the analysis of tumor biomarkers, influences therapeutic decisions and should, therefore, be assessed as early and accurately as possible [2,3].

Digitizing tissue slides allows evaluation via computer screens, which can improve efficiency over traditional microscopy [4]. It also supports AI-driven tissue classification, segmentation, potentially increasing the speed of image interpretation, and refining clinical decision-making [5–8]. Correct segmentation of the tumor is a necessary step towards computer-assisted tumor analysis and lung cancer diagnosis [9–14].

When working with whole slide images (WSIs), the application of AI models is complicated due to the large size of the images. Down-sampling the WSIs to a manageable size would compromise resolution and potentially result in the loss of critical diagnostic details. A common approach in digital pathology is, therefore, to divide the images into several small squares, called patches. This is a more effective approach, but the use of patch-based analysis alone can lead to a loss of broader spatial relationships. Alternatively, the image can be down-sampled, or a hybrid strategy that combines both methods can be used to optimize the analytical balance between detailed resolution and global context.

Some of the best-performing AI methods in the analysis of WSIs are deep neural networks [14,15]. The state-of-the-art in image segmentation tasks is the use of complex neural network architectures such as vision transformers and InternImage [16,17]. However, these methods require a relatively large amount of data [18]. Transfer learning techniques may also be used to train or fine-tune pre-trained models on new data [19]. Patch-wise classification (PWC) or segmentation approaches may outperform direct segmentation of the tumor in a down-sampled image without dividing it into patches [20].

Several models have been proposed for tumor segmentation in WSIs [11,21–29]. Zhao et al. proposed a novel hybrid deep learning framework for colorectal cancer that uses a U-Net architecture. This model features innovative residual ghost blocks, which include switchable normalization and bottleneck transformers for extracting features [11].

The MAMC-Net model introduced a multi-resolution attention module that utilizes pyramid inputs for broader feature information and detail capture [21]. An attention mechanism refines features for segmentation, while a multi-scale convolution module integrates semantic and high-resolution details. Finally, a connected conditional random field ensures accurate segmentation by addressing discontinuities [21]. The authors showcased the superior performance of their model on breast cancer metastases and gastric cancer [21].

DHU-Net combines Swin Transformer and ConvNeXt within a dual-branch hierarchical U-shaped architecture [22,30,31]. This method effectively fuses global and local features by processing WSI patches through parallel encoders, utilizing global-local fusion modules and skip connections for detailed feature integration [22]. The Cross-scale Expand Layer aids in resolution recovery across different scales. The network was evaluated on datasets covering different tumor features and cancer types, and achieved higher segmentation results than other tested methods [22].

Krikid et al. showed that deep-learning applications in microscopic image segmentation have evolved from predominantly cell- and nucleus-centric tasks—often on small, homogeneous datasets—to encompass more complex, tissue-level analyses, reflecting a shift toward multi-scale, clinically relevant segmentation across diverse microscopy-modality types [32]; Greeley et al. introduced pyramid tiling for efficient gigapixel histology analysis [33]; promptable models like SAM and MedSAM enable zero-shot, universal segmentation across modalities [34,35].

Pedersen et al. introduced H2G-Net, a cascaded convolutional neural network (CNN) architecture for segmenting breast cancer regions from gigapixel histopathological images [23]. It employs a patch-wise detection stage and a convolutional autoencoder for refinement, demonstrating significant improvements in tumor segmentation. The approach outperformed single-resolution methods, achieving a Dice similarity coefficient (DSC) of (0.933 ± 0.069) [23]. Its efficiency is underscored by fast processing times and the ability to train deep neural networks without having to store patches on disk.

One of the most significant challenges in using WSIs for tumor segmentation is still the scarcity of labeled data. The marking of tumor tissue in WSIs by pathology experts is time-consuming and may be a bottleneck in research. Alternative computational strategies, such as unsupervised or semi-supervised learning methods should, therefore, be explored. Clustering allows the segmentation of tumor regions with little or no need for predefined labels, and can be a useful tool in this context [24,25].

Yan et al. presented a self-supervised learning method using contrastive learning to process WSIs for tissue clustering [26]. This approach generates discriminative embeddings for initial clustering, refined by a silhouette-based scheme, and extracts features using a multi-scale encoder [26]. It achieved high accuracy in identifying tissues without annotations. Their results show an area under the curve (AUC) of 0.99 and accuracy of approximately 0.93 for distinguishing benign from malignant polyps in a cohort of 20 patients [26].

Few-shot learning is also a promising method for handling limited labeled data [27,28]. By design, few-shot learning algorithms can learn from a very limited number of labeled examples. This can be particularly relevant for the classification of small patches, where a small set of labeled examples can guide the learning process. Few-shot learning techniques can generalize from these examples to classify new, unseen patches, facilitating the identification and segmentation of tumor regions [27,28]. Titoriya et al. explored few-shot learning to enhance dataset generalization and manageability by utilizing prototypical networks and model agnostic meta-learning across four datasets [29]. The design achieved 85% accuracy in a 2-way 2-shot 2-query mode [29].

In this paper, we propose a new CNN-based model which is a combination of DenseNet [36], ResNet [37], and U-Net architecture (DRU-Net) for segmenting non-small cell lung carcinomas (NSCLCs). It is an end-to-end approach consisting of a dual head for feature extraction and patch classification, followed by a U-Net for refining the segmentation result. The proposed model is tested on a novel in-house dataset of 97 annotated NSCLC WSIs. To increase model performance, we adopted a many-shot learning approach during training and added a multi-lens distortion augmentation technique to both patches and down-sampled WSIs.

2. Materials and Methods

2.1. Cohorts

In this study, two different collections of NSCLCs were used: the Norwegian lung cancer biobank (NLCB) cohort and Haukeland University lung cancer (HULC) cohort [38,39]. The NLCB cohort includes histopathological, cytological, biomarker, and clinical follow-up data from patients with suspected lung cancer diagnosed in Central Norway after 2006 [40]. Both diagnostic tumor biopsies and sections from surgical lung cancer specimens are available. The distribution of histological subtypes in each dataset is listed in Table 1 [41,42].

The HULC cohort comprises 438 surgically treated NSCLC patients diagnosed at Haukeland University Hospital, Bergen, Norway from 1993 to 2010. In this study, 97 NSCLC cases from the HULC cohort were included. From both cohorts, 4 µm tis-

sue sections were made, deparaffinized, rehydrated in ethanol, and immersed in tap water. Hematoxylin staining was applied and sections were rinsed in water and then in ethanol. Sections were then stained with alcoholic eosin. Post-staining, slides were dehydrated in ethanol, placed in TissueClear, air-dried, and scanned using Olympus VS120-S5 scanner (Olympus Soft Imaging Solutions GmbH, Munster, Germany) at ×40 magnification [43]. WSIs were quality-controlled by a pathologist to ensure that only high-quality scans were included in the study. They were reviewed for sectioning, staining, and scanning artifacts.

Table 1. Histological subtypes of non-small cell lung carcinoma cases in the NLCB and the HULC cohorts. Counts are shown with corresponding percentages. AC: Adenocarcinoma, SCC: Squamous Cell Carcinoma, NSCC: Non-small Cell Carcinomas, WSIs: Whole Slide Images.

Histological Subtype	NLCB (*n*,%)	HULC—Train (*n*, %)	HULC—Test (*n*, %)
AC	16 (38.1%)	38 (49.4%)	7 (35.0%)
SCC	15 (35.7%)	32 (41.6%)	10 (50.0%)
Other NSCC	11 (26.2%)	7 (9.1%)	3 (15.0%)
Total number of WSIs	42	77	20

To conduct a broader study of the proposed augmentation's effect, we utilized the following open datasets in addition to HULC: MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 [44–46].

2.2. Ethical Aspects

All methods were carried out in accordance with relevant guidelines and regulations, and the experimental protocols were approved by the Regional Committee for Medical and Health Sciences Research Ethics (REK) Norway (2013/529, 2016/1156, and 257624). Informed consent was obtained from all subjects and/or their legal guardian(s) for NLCB in accordance with REK 2016/1156. For subjects in the HULC cohort, exemption from consent was ethically approved by REK (2013/529).

2.3. Annotations and Dataset Preparation

We used two annotation approaches on WSIs: whole tumor annotation (WTA) and partial selective annotation (PSA). In the WTA approach, pathologists marked the tumor outline in 97 WSIs from the HULC cohort. Of these WSIs, 51 were used for training, 26 were used for validation, and 20 were used for testing. WSIs with tissue microarray (TMA) holes (n = 3) were manually assigned to the test set to prevent potential biased training; the remaining WSIs were randomly separated into the training, validation, and the rest of the test sets.

To reduce the time spent by pathologists in making the WTAs, initial annotations were first made in 72 cases using two different AI-based segmentation models, (i) the H2G-Net model developed for breast cancer segmentation (n = 25) and (ii) a customized early-stage clustering model based on the corrected annotations from the H2G-Net model (n = 47) [23]. Pathologists then manually refined the tumor region annotations using the QuPath software (version 0.3.2) [47]. The remaining 25 cases were manually annotated without any prior AI-based segmentation models. A third pathologist reviewed the annotations, and in case of discrepancy, consensus was reached after discussion. The final annotations were exported as binary masks, serving as ground truth.

In the PSA approach, pathologists marked small regions of interest in 42 WSIs from the NLCB cohort. These WSIs were used for training and validation of the patch-wise classifier model. Marked areas included parts of the invasive tumor, normal alveolar tissue, stromal tissue, immune cells, and areas of necrosis. Other non-tumor tissues marked included

respiratory epithelium, reactive alveolar tissue, cartilage, blood vessels, glands, lymph nodes, and macrophages. The purpose of marking these regions was to reduce the time required for manually annotating whole tumor regions, and to guide a particular selective generation of patches intended for use in the patch-wise model's training.

2.4. Proposed Method

The pipeline of the proposed model (DRU-Net) has two distinct stages, a PWC stage and a refinement stage. The PWC model was trained on the NLCB cohort using a many-shot learning method, and the refinement U-Net was trained on a set of down-sampled WSIs from the HULC cohort. In the PWC stage, the model assigns probabilities to each patch of the WSIs (excluding the glass), indicating whether the patch contains tumor tissue or non-tumor tissue. The classifier outputs a preliminary assessment of each patch's nature, based on local features within the patch. The patches are then stitched together to produce a heatmap matching the original size of the down-sampled WSIs.

2.4.1. Patch-Wise Classifier

The PWC was constructed by fusing truncated backbones of two architectures, DenseNet201 [36] and ResNet101V2 [37], pre-trained on ImageNet [48]. We conducted a preliminary search on a dataset subset to determine the most effective truncation points for both DenseNet and ResNet backbones. This empirical exploration guided our layer selection based on performance. These networks are used for parallel processing of the input and feature generation (we refer to this PWC model as DR-Fused). In our proposed architecture, both DenseNet201 and ResNet101V2 receive the same input, which is the image patch. Each network processes this input concurrently, and after feature extraction, the outputs from both DenseNet201 and ResNet101V2 pass through their respective global average pooling layers. This step compresses the feature representation to help prevent overfitting. The compressed features from both networks are then concatenated and fed through the classifier head (Figure 1).

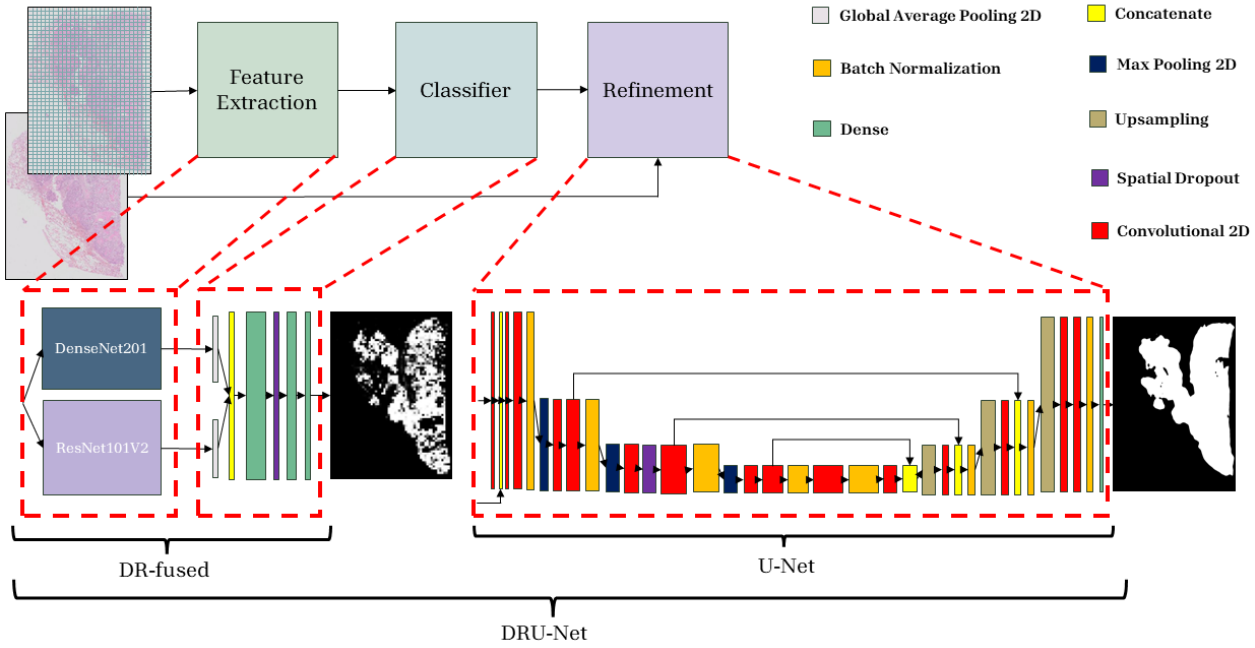


Figure 1. Illustration of the proposed DRU-Net model. The patched image is fed into the classifier part. The output of the classifier is combined with a down-sampled WSI as an input for the refinement head.

2.4.2. Refinement Network

The heatmap is generated from applying the PWC across the WSIs. The resultant heatmap is then resized and concatenated with a down-sampled version of the WSI (1120 × 1120 pixels). The fused inputs are then fed to a refinement network, similar to H2G-Net [23]. Using a refinement network allows for adjusting the initial patch-wise predictions based on global WSI-level information.

The proposed refinement network is a simple, lightweight U-Net architecture, specifically tailored to process two image inputs (Figure 1). In this model, the two inputs (down-sampled RGB WSI and the heatmap) are concatenated into a 4-channel image and then processed through multiple convolutional layers with ReLU activation functions, batch normalization, spatial dropout, skip connections, max pooling, and up-sampling layers (with nearest-neighbor interpolation). The network ends with a softmax activation function.

2.4.3. Data Augmentation

To improve model robustness, data augmentation is commonly performed. Data augmentation generates artificial copies of the training data through a predefined algorithm. This allows the training data to better cover the expected data variation. Data augmentation was integrated into the training data generation process, with the following methods applied randomly: vertical and horizontal flipping, rotations (multiples of 90°), multiplicative contrast adjustment, hue and brightness variations, and the proposed multi-lens distortion augmentation. During the many-shot learning using PSA, we extracted patches by cropping a random 224 × 224-pixel section from each image. Each image appeared only once per epoch, where an epoch is defined as one iteration of all the training data.

2.4.4. Multi-Lens Distortion Augmentation

A novel data augmentation method, multi-lens distortion, was developed to simulate several local random lens distortions. This technique aims to allow the model to recognize the important features of the images under a wider range of cell/tissue shapes.

The algorithm uses a predefined number of lenses. For each lens, a random position within the image is selected. Then, a random distortion radius and strength value are used to apply the barrel and/or pincushion distortion effect at the selected positions (Algorithm 1). An example of this augmentation is shown in Figure 2. The optimal radius range and lens count were established empirically through an iterative series of experiments, with each configuration assessed qualitatively to identify the most compelling results. From a histopathology point of view, too strong augmentations produce morphologically invalid images, which degrade performance. Thus, it is necessary to specifically tune these parameters for the targeted applications, especially in healthcare.

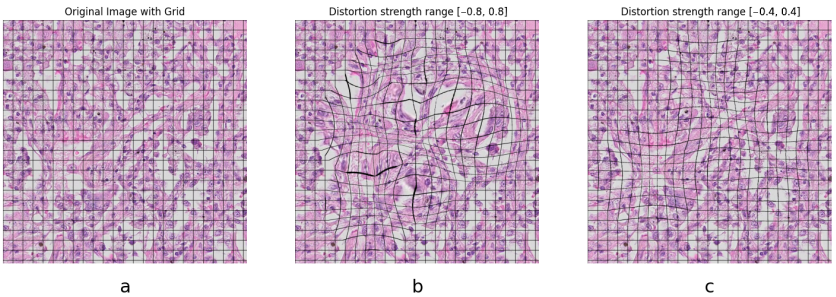


Figure 2. Sample effect of the novel augmentation on a patch with overlaid grids to illustrate the effect. (**a**) Original image showing epithelial cells. (**b**) Augmented image with parameters set too high, cell size variation and deformation are visible. (**c**) Augmented image with a medium setting of the parameters.

Algorithm 1 Multi-Lens Distortion (implementation-level pseudocode)

Require: $img \in \mathbb{R}^{H\times W\times C}$, N ▷ number of lenses, $(r_{\min}, r_{\max})$, $(s_{\min}, s_{\max})$
Ensure: $out \in \mathbb{R}^{H\times W\times C}$
$out \leftarrow img$ ▷ deep copy
$(yidx, xidx) \leftarrow \text{meshgrid}(0{:}H{-}1,\ 0{:}W{-}1)$
for $i \leftarrow 1$ **to** N **do**
  $cx \leftarrow \text{randInt}(0, W{-}1)$
  $cy \leftarrow \text{randInt}(0, H{-}1)$
  $R \leftarrow \text{randInt}(r_{\min},\ r_{\max})$
  $S \leftarrow \text{randFloat}(s_{\min},\ s_{\max})$
  for all (y, x) **in** $\{0{:}H{-}1\} \times \{0{:}W{-}1\}$ **do**
    $dx \leftarrow x - cx\ ;\ dy \leftarrow y - cy$
    $r \leftarrow \sqrt{dx^2 + dy^2}$
    if $r < R$ **then**
      $\hat{r} \leftarrow r/R$ ▷ normalised distance
      $sf \leftarrow 1 - \hat{r}$ ▷ scaling factor
      $scale \leftarrow 1 - S \cdot sf$
      $x_{new} \leftarrow cx + dx \cdot scale$
      $y_{new} \leftarrow cy + dy \cdot scale$
      $x_{new} \leftarrow \text{clamp}(x_{new},\ 0,\ W{-}1)$
      $y_{new} \leftarrow \text{clamp}(y_{new},\ 0,\ H{-}1)$
      $out[y, x] \leftarrow img[y_{new},\ x_{new}]$
    end if
  end for
end for
return out

2.4.5. Model Training

The PWC network was fine-tuned to adapt to the specific task by freezing the initial layers. The following training parameters were included: optimizer: Adamax with a learning rate of 1×10^{-4}; loss function: categorical crossentropy; metrics: F_1-score; batch size: dynamically determined based on the training generator configuration; epochs: up to 200 with early stopping based on validation loss to prevent overfitting.

The refinement network training involved the following: optimizer: Adam with a learning rate of 1×10^{-4}; loss function: Dice loss function, optimized for segmentation tasks; metrics: Thresholded Dice score; batch size: 2; epochs: up to 300 with early stopping based on validation loss to prevent overfitting; training environment: utilization of GPU and memory growth settings to optimize hardware usage.

In the WTA method, the same set of slides was used for both PWC and segmentation models' training. From the 97 slides, 77 slides were randomly chosen and divided into training and validation sets in a 2:1 ratio, with 51 and 26 slides, respectively, while 20 slides (including those with TMA holes) were used for testing.

WSIs in the dataset from the HULC cohort were divided into tiles (patches) and each tile was fed into the neural network along with the non-tumor/tumor label based on the provided annotation. To create the annotation labels for patches, non-tumor and tumor tiles were assigned the values 0 and 1, respectively. We first used a threshold on color gradients to separate the tissue from the background glass. Any tile that did not include more than 25% tissue was disregarded, meaning that all the input tiles contained less than 75% background glass. Also, a minimum of 5% of the tumor area was required for a tile to be classified as tumor, and for the non-tumor regions, only tiles with no tumor were assigned. Tiles containing less than 5% tumor area were excluded.

Using the annotated WSI regions with PSA in the NLCB dataset, 40 areas were assigned to the tumor class (labeled as 1) and 50 areas to the non-tumor class (labeled as 0).

The selected areas led to the generation of patches in subsequent steps. Specifically, out of 50 areas categorized as non-tumor, 40 clearly lacked tumor characteristics, and 10 showed features slightly above the initial threshold, as shown in Supplementary Figure S2. This threshold was established through model training before intentionally creating an imbalance in the dataset. The imbalance was introduced after unsuccessful attempts to enhance model generalizability through various methods, including weighted loss functions, focal loss, threshold adjustment, and sampling strategies.

2.4.6. Post-Processing

After the segmentation results were received, two post-processing steps were performed. First, small fragments were removed by converting images into grayscale and then to binary format to identify and eliminate fragments smaller than a fixed threshold. The threshold was set to the smallest annotated segmentation area in the ground truth. In the second step, an edge smoothing algorithm was applied to enhance image quality. This improvement was achieved through mathematical techniques known as morphological operations, which are commonly used in digital image processing to modify the geometrical structure of images. Specifically, we used a process called morphological opening, which involves an erosion operation followed by a dilation. This sequence helps reduce jagged edges and smooths the boundaries of objects within the image. The operations were performed using a kernel size of 7×7. Additionally, a median blur with a kernel size of 11×11 was applied to further smooth the edges. It is important to note that these morphological operations refer to image processing techniques. They are purely computational methods used to process the digital images and should not be confused with the morphological study of biological tissues.

2.5. Implementation

Implementation was conducted in Python 3.8.10. TensorFlow (v2.13.1) was used for model architecture implementation and training [49]. These additional libraries were used for the experiments: pyFAST, OpenCV, NumPy, Pillow, SciPy, scikit-learn, and Matplotlib [50–57]. Trained models were converted to the ONNX format using the tf2onnx library [58]. Converted models were then integrated into FastPathology for deployment [59]. FastPathology is an open-source, user-friendly software developed for deep learning-based digital pathology that offers tools for processing and visualizing WSIs. The source code used to conduct the experiments is made openly available at. https://github.com/AICAN-Research/DRU-Net (accessed on 29 April 2024).

2.6. Experiments

To compare the proposed model (DRU-Net) with other models, the following experiments were carried out: modifications of the previously introduced H2G-Net model on both datasets, DRU-Net with the backbone trained on the HULC cohort and NLCB, and applying the few-shot and many-shot learning techniques along with clustering (Table 2) [23].

H2G-Net could be tested as is, and be fine-tuned with five different modifications [23]. First, H2G-Net was tested without any modification, fine-tuning, or additional training, to see whether a model trained for breast cancer tumor delineation can also work for lung cancer. Second, the PWC of the H2G-Net was fine-tuned on annotated WSIs from the HULC cohort, and the original U-Net of H2G-Net was applied on top of the PWC results. Third, the whole model (PWC and U-Net) was fine-tuned on the training data. Then, the same three methods were tested, but with the PWC trained on NLCB instead of the HULC cohort.

Table 2. Methods and experiments carried out with various models on the same 20 WSIs of the test set from the HULC cohort. Abbreviations: PWC: patch-wise classifier; HULC: Haukeland University Lung Cancer; NLCB: Norwegian Lung Cancer Biobank; FSC: few-shot (with a pre-trained MobileNetV2 [60] model) + clustering; MSC: many-shot (with a pre-trained MobileNetV2 [60] model) + clustering.

	Models	Modifications	Training Dataset (s)
(I)	H2G-Net	—	—
(II)	H2G-Net	Fine-tuned PWC	HULC Cohort
(III)	H2G-Net	Fine-tuned U-Net	HULC Cohort
(IV)	H2G-Net	Fine-tuned PWC and original U-Net	HULC Cohort
(V)	DRU-Net	—	HULC Cohort
(VI)	H2G-Net	Fine-tuned PWC	NLCB
(VII)	H2G-Net	Fine-tuned PWC and U-Net	PWC trained on NLCB, U-Net trained on HULC Cohort
(VIII)	FSC	—	NLCB
(IX)	MSC	—	NLCB
(X)	DRU-Net	—	PWC trained on NLCB, U-Net trained on HULC Cohort

An ablation study was performed to evaluate the effect of the proposed multi-lens distortion augmentation. A pre-trained DenseNet121 was tested on four open datasets: MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100 [44–46]. Experiments were repeated with and without this augmentation on the mentioned open datasets by randomly selecting 10% of the training data and the results were compared using Wilcoxon test. Both control and test groups included other augmentation techniques, such as color adjustments, flipping, rotation, brightness, and contrast augmentations. The effect of this augmentation on the training time was measured using the integrated TensorFlow functions by comparing the time with and without the augmentation and the results were averaged on WSIs and compared between the two [49].

We also investigated the effect of removing the top-most skip connection of the U-Net refinement model and we calculated the average Hausdorff distances (HDs) for two sets of final segmentation predictions in comparison to a ground truth set. This was conducted to quantify the effect of removing that skip connection, which was implemented to reduce the small fragments around the segmentation perimeter.

2.7. Model Evaluation

2.7.1. Quantitative Model Assessment

To quantitatively validate the patch-wise classification performance, precision, recall, and F_1-score were used [61]. The validation of the final segmentation on WSI-level was performed using DSC and HD [62].

2.7.2. Qualitative Model Assessment

The qualitative assessment of the segmentation results was conducted by two pathologists using the scoring system described in Table 3. Qualitative assessment was conducted on the same 20 WSIs of the test set from the HULC cohort.

2.7.3. Saliency Maps

To survey the model's decision-making process and the areas of patches that were most relevant for predicting the tumor class, we employed a method known as gradient-based saliency maps [63–66]. This approach operates by computing the gradient of the output class (the class for which we want to understand model sensitivity) with respect to the input image. These gradients indicate the sensitivity of the output to each pixel in the input image. By highlighting the pixels with the highest gradients, we can visualize the areas

that most strongly influenced the model's classification decision. We used six different patches selected from six different WSIs from the HULC cohort to analyze the saliency maps. Patches were chosen to represent true positive, false positive, and false negative predictions. Patches with true positive predictions were selected to include various histological features and cell types in each patch to better assess the model's decision process.

Table 3. Qualitative evaluation scoring system.

0	1	2	3	4	5
No tumor tissue in image or segmentation, or image not suitable for analysis	Completely wrong segmentation of tumor, tumor tissue not segmented	A large part of the tumor is not segmented	Most of the tumor is correctly segmented, but some false positive or false negative areas	Most of the tumor is correctly segmented, only sparse false positive or false negative areas	The whole or almost the whole tumor correctly segmented

2.7.4. Computation of FLOPs and Parameters

To quantitatively assess the computational complexity and model size, we calculated the number of floating-point operations (FLOPs) and the total number of trainable parameters for all evaluated models, including DR-Fused and several standard architectures. For each model, FLOPs were estimated by converting the model into a frozen computational graph using TensorFlow's `convert_variables_to_constants_v2` function, followed by profiling with `tf.compat.v1.profiler`. The FLOPs represent the total number of arithmetic operations required for a single forward pass of an input image sized $224 \times 224 \times 3$. Parameter counts were obtained directly via the `count_params` method provided by TensorFlow. All FLOPs and parameter values were reported in millions (M) for clarity. MobileNetV2 was designated as the baseline model. Relative changes in FLOPs and parameters (ΔFLOPs and ΔParams) were computed for each model compared to MobileNetV2, using the following formulas:

$$\Delta\text{FLOPs}\,(\%) = \frac{\text{FLOPs}_{\text{model}} - \text{FLOPs}_{\text{baseline}}}{\text{FLOPs}_{\text{baseline}}} \times 100 \quad (1)$$

$$\Delta\text{Params}\,(\%) = \frac{\text{Params}_{\text{model}} - \text{Params}_{\text{baseline}}}{\text{Params}_{\text{baseline}}} \times 100 \quad (2)$$

3. Results

The highest DSC on average on the 20 WSIs of the test set from HULC cohort was achieved by DRU-Net, followed by the H2G-Net with fine-tuned PWC on the HULC cohort (Figure 3). Similar differences in DSC were observed for the models without the refinement networks (Figure 4).

Proposed multi-lens distortion augmentation applied to various datasets resulted in increased F_1-score overall, this change was statistically significant when applied to our dataset from the NLCB (Table 4). Applying this augmentation technique increased training time by an average of 8%. DSC and patch-wise accuracy increased when multi-lens distortion augmentation was used with a magnitude strength in the range $[-0.4, 0.4]$, but higher magnitudes caused a decrease in performance (Figure 5).

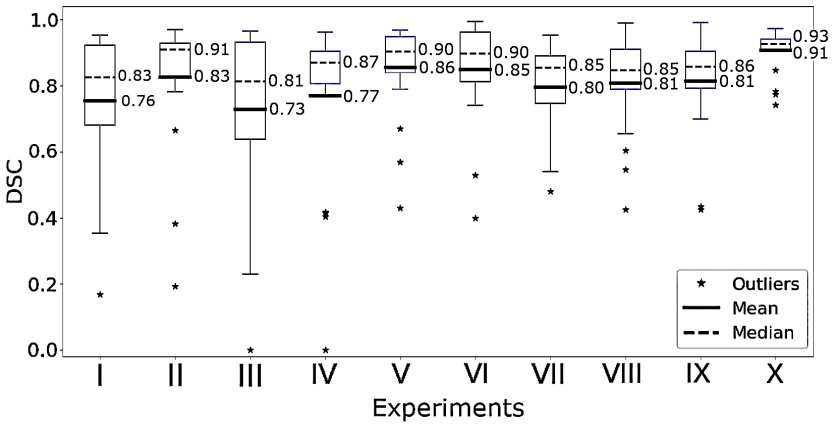


Figure 3. Boxplots of the Dice similarity coefficients (DSCs) of the experiments shown Table 2 on the 20 WSIs of the test set. (I) original H2G-Net, (II) H2G-Net with fine-tuned PWC on HULC cohort, (III) H2G-Net with fine-tuned U-Net on HULC cohort, (IV) H2G-Net with fine-tuned PWC and U-Net on HULC cohort, (V) DRU-Net trained on HULC Cohort, (VI) H2G-Net with fine-tuned PWC on NLCB, (VII) H2G-Net with fine-tuned PWC on NLCB and fine-tuned U-Net on HULC Cohort, (VIII) FSC, (IX) MSC, (X) DRU-Net with PWC trained on NLCB and U-Net trained on HULC Cohort.

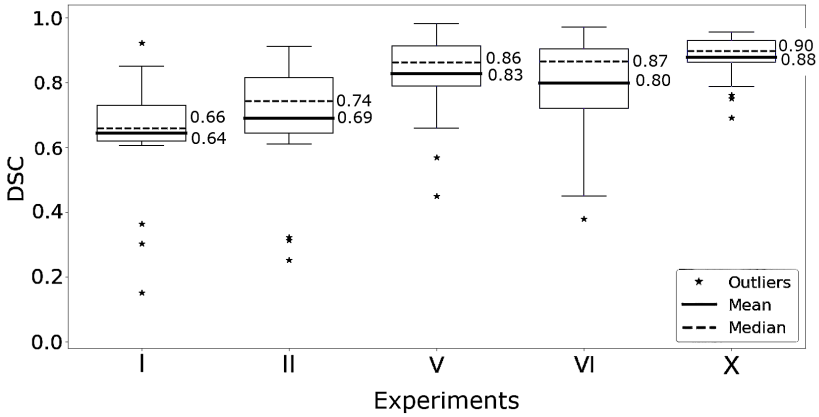


Figure 4. Boxplot of the Dice similarity coefficients (DSCs) of the PWC models in experiments listed in Table 2 without the refinement network, only the patch-wise classifier is used to produce these results. (I) original H2G-Net, (II) H2G-Net with fine-tuned PWC on HULC cohort, (V) DR-Fused trained on HULC Cohort, (VI) H2G-Net with fine-tuned PWC on NLCB, (X) DR-Fused trained on NLCB and U-Net trained on HULC Cohort.

Table 4. The impact of the multi-lens distortion augmentation technique using different architectures on different datasets, randomly selecting 10% of the training data. Pairwise tests were performed using Wilcoxon signed-rank tests. The augmentation design with the highest F_1-scores row-wise are highlighted in bold.

		F_1-Score		
Model	**Dataset**	**W/O Aug**	**W/ Aug**	***p*-Value**
DenseNet121	MNIST	0.9893	**0.9894**	0.2311
DenseNet121	Fashion-MNIST	0.9043	**0.9208**	<0.001
DenseNet121	CIFAR-10	0.8086	**0.8235**	<0.001
DenseNet121	CIFAR-100	0.5199	**0.5581**	0.0502
H2G-Net	NLCB	0.8299	**0.8341**	0.0701
DRU-Net	NLCB	0.8868	**0.9025**	0.0241

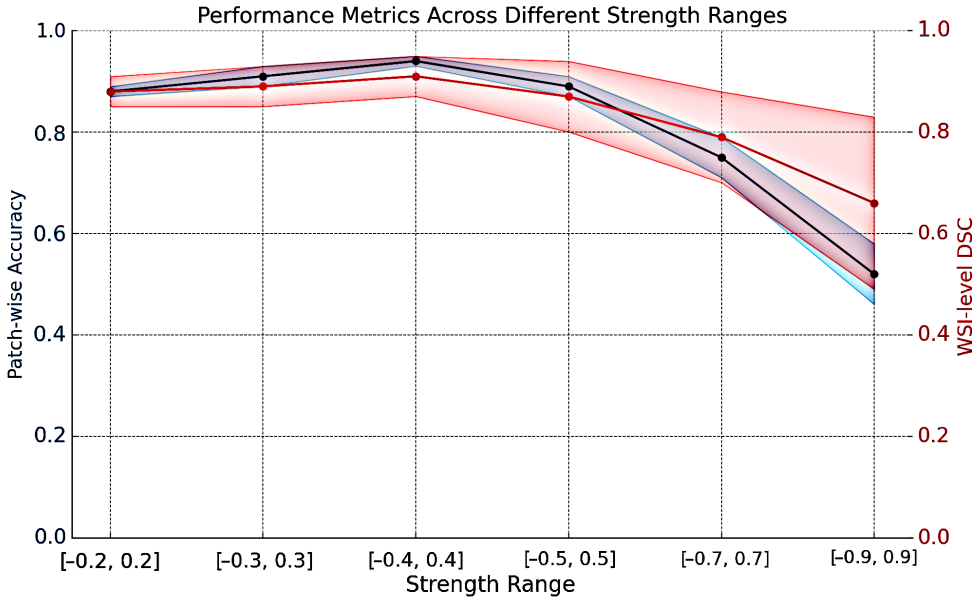


Figure 5. The impact of the multi-lens distortion augmentation technique using the DRU-Net model. DSC: Dice similarity coefficient. The highlighted regions indicate the variance, and the mean values are shown on the curve.

The original H2G-Net resulted in an average of 0.76 DSC (Figure 3) and 0.66 intersection over union (IOU) scores. On average, 25% of the non-tumor regions around the true tumor outlines were falsely labeled as tumor. When the PWC component of the model was used without refinement, the predictions resulted in 0.64 DSC and 0.61 IOU, showing that the refinement improved the predictions significantly.

A fine-tuned PWC trained and validated on 77 WSIs from the HULC cohort, with the direct implementation of the pre-trained U-Net from H2G-Net, was tested on 20 WSIs from the HULC cohort and resulted in an average of 0.83 DSC (median 0.91) (Figure 3) and an average of 0.74 IOU scores. Scores were reduced to an average of 0.77 DSC (median of 0.87) and an average of 0.69 IOU when both the U-Net and the PWC were fine-tuned.

The proposed model (DRU-Net) tested on the same 20 WSIs resulted in an average of 0.91 DSC (median 0.93) and 0.81 IOU. Also, removing the top skip connection in our U-Net model (DRU-Net) resulted in an average reduction in HD by 4.8%. Figure 6 shows a comparison of the results from various models. Table 5 summarizes various backbones' performance in the patch-wise classifier component of the model.

Table 5. Comparison of different backbone architectures for patch-wise classification of lung cancer tissue using the many-shot method. The best-performing architecture per metric is highlighted in bold. Abbreviations: DR: fusion of DenseNet201 (D) and ResNet101V2 (R).

Architecture	F_1-Score	Precision	Recall
VGG19 [67]	0.87	0.86	0.87
ResNet101V2 [37]	0.89	0.89	0.89
MobileNetV2 [60]	0.86	0.86	0.86
EfficientNetV2 [68]	0.89	0.89	0.89
InceptionV3 [69]	0.90	0.89	0.91
DenseNet201 [36]	0.91	0.91	0.91
Proposed DR-Fused	**0.94**	**0.94**	**0.93**

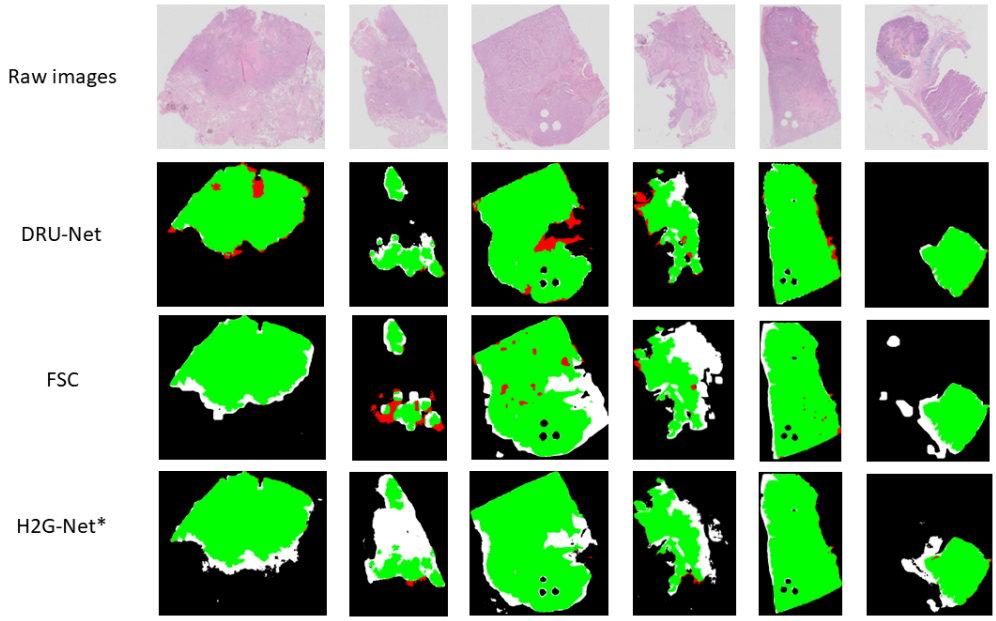


Figure 6. Sample results of three tested networks. First row: original whole slide images (WSIs), second row: DRU-Net, third row: FSC (Few-shot learning + clustering), fourth row: H2G-Net with fine-tuned patch-wise classifier and original U-Net. Green pixels indicate true positives, White pixels indicate false positives and red pixels indicate false negatives. * Indicates that this is not the original H2G-Net, but a modified version.

In addition to the classification performance, we evaluated the computational complexity of each backbone in terms of FLOPs (floating point operations) and number of parameters, as summarized in Table 6. While the proposed DR-Fused backbone exhibits higher computational cost compared to lightweight models such as MobileNetV2 [60], it remains significantly more efficient than very large networks like VGG19 [67] and ResNet101V2 [37]. Importantly, the DR-Fused model achieves substantial improvements in classification performance (Table 5), with an F_1-score of 0.94 compared to 0.86 for MobileNetV2 and 0.91 for DenseNet201 [36].

Table 6. Computational complexity comparison between different backbone architectures. Metrics are reported as total FLOPs and number of parameters. The percentage increase relative to MobileNetV2 is also reported.

Architecture	FLOPs (M)	Params (M)	ΔFLOPs (%)	ΔParams (%)
DR-Fused	11,105.27	13.18	1712.42	483.02
VGG19 [67]	39,276.93	139.58	6310.14	6074.55
ResNet101V2 [37]	14,430.04	42.63	2255.04	1785.86
MobileNetV2 [60]	612.73	2.26	0.00	0.00
EfficientNetV2 [68]	1455.32	5.92	137.51	161.97
InceptionV3 [69]	5693.36	21.81	829.18	864.67
DenseNet201 [36]	8631.68	18.33	1308.72	710.68

We compared the performance of several models on processing a set of 20 WSIs, with the average dimensions being approximately 108,640 pixels in width and 129,835 pixels in height. H2G-Net and its fine-tuned versions were the fastest models during inference (62 s). Although the many-shot and few-shot models had faster training, they exhibited slower runtimes, with MSC taking the longest at 167 s and DRU-Net at 152 s.

The results of the saliency map analysis in six patches are shown in Figure 7. False-positive areas in the saliency maps were partly explained by areas with reactive pneumocytes, macrophages, and reactive pneumocyte hyperplasia.

The qualitative assessment resulted in an average score of 3.95 out of 5. In nine of the cases assessed, there were sparse areas in the periphery of the tumor that the model misclassified.

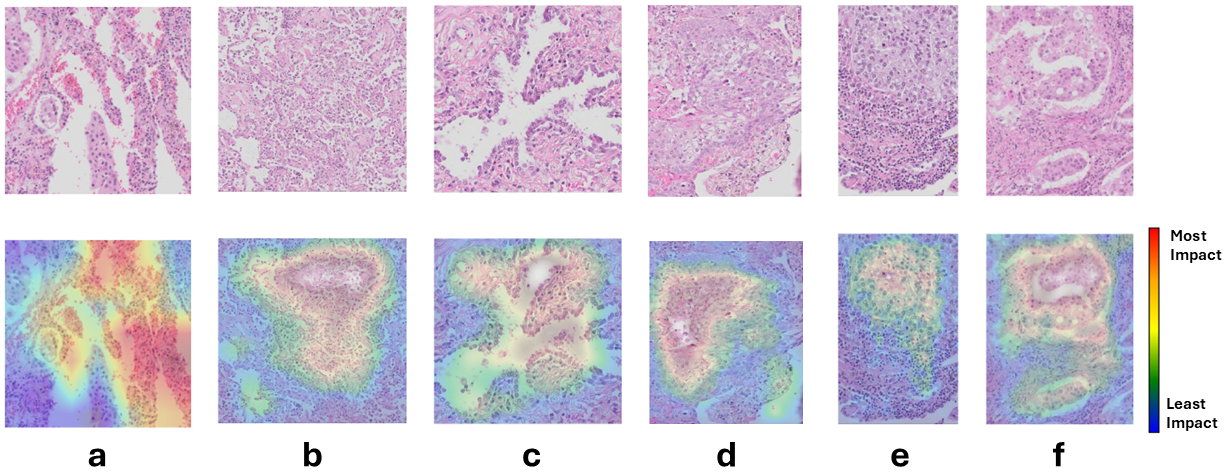


Figure 7. Sample patches (**top** row) and their overlaid saliency maps (**bottom** row); only the patches were given to the PWC model. Note that the saliency map does not indicate malignancy; instead, it shows how different regions of the image influence the classification decision. The colors on the map range from blue, indicating the least influence, to red, which indicates the most influence. (**a**) shows a false negative where it misses the tumor, (**b**,**c**) show false positive tumor detection. (**d**–**f**) show true positive tumor detection. (**a**) shows three small sheets of atypical epithelial tumor cells, of which only one is highlighted in red. The remaining tissue comprises widened alveolar septae with inflammatory cells, pigmented macrophages, reactive pneumocytes and red blood cells. (**b**) includes reactive pneumocytes and macrophages. (**c**) shows reactive pneumocyte hyperplasia. (**d**) presents a solid tumor with enlarged nuclei, where the majority of the model's focus lies; the peripheral parts of the patch contain alveolar tissue. (**e**) highlights a solid tumor (mostly in yellow) alongside inflammatory cells (primarily in blue). (**f**) shows a solid tumor with areas of necrosis (mainly highlighted in yellow and red) as well as fibrous tissues with inflammatory cells, predominantly marked in blue and some green.

4. Discussion

In this paper, we introduce a novel deep learning-based model to segment the outline of NSCLCs. We have incorporated a patch-wise classifier, synergistically integrating truncated DenseNet201 [36] and ResNet101V2 [37] architectures, enhanced by a segmentation refinement U-Net model. The proposed composite PWC model demonstrated superior performance over other tested backbones. Due to our relatively small dataset and considering the desired memory and speed efficiency, CNNs were preferred in this study. Using transformer-based models would have required more extensive datasets and computational resources [70,71].

This study also resulted in a novel dataset comprising annotated NSCLCs and marked regions of interest in WSIs from NSCLCs, covering various tissue types. Our results indicate that the PSA approach yielded more effective training outcomes for the patch-wise classifier than the WTA techniques, both with and without class balancing via tissue clustering. Using the WTA approach, annotations were extremely time-consuming for expert pathologists (including review and correction). However, the PSA method significantly reduced this time by an order of magnitude.

Our study demonstrated that the implementation of the multi-lens distortion augmentation technique enhanced classification outcomes across diverse datasets with limited volume of training data. However, the effect of this augmentation could vary depending on the data themselves. We investigated the effect of the augmentation's strength range

on the patch-wise classification accuracy and refinement network's DSC on WSI-level, concluding that the degree of augmentation is pivotal for its impact on the training process. Excessively strong distortion of images could obstruct the model's ability to learn relevant patterns, as shown in the impact of the multi-lens distortion augmentation with various strength ranges (Figure 5). It is important to note that the effective range is dependent on the dataset, and the same values may not necessarily yield similar improvements across different datasets.

The non-linear warping introduced by the multi-lens distortion mimics the subtle spatial deformations, slight micro-stretches of the tissue, and local distortions. By applying controlled, spatially varying warps at different scales, our augmentation reproduces these effects. This generates realistic variations in cell and tissue morphology. This not only strengthens model robustness to scanner-induced artifacts, but also promotes generalization across varying magnification levels, shapes, stretches, and similar sample-preparation conditions.

Instead of stain normalization techniques, we used an augmentation-based approach to produce more robustness, maintain important staining details, reduce computational complexity, and safeguard essential characteristics from unintended modification. The dataset already had consistent staining, which eliminated the need for traditional stain normalization [72,73]. To mimic the wide range of HE staining protocols seen across laboratories, we applied the mentioned randomized adjustments in brightness, contrast, and hue during training. By exposing the model to these controlled, biologically plausible variations in color balance and intensity, we effectively simulate batch-to-batch and site-to-site staining differences.

The RGSB-UNet model features a unique hybrid design that combines residual ghost blocks with switchable normalization and a bottleneck transformer [11]. This design focuses on extracting refined features through its complex structure. However, our study found that simpler and more synergistic architectures can also effectively extract reliable features.

The MAMC-Net model improves tumor boundary detection by using a conditional random field layer [21], whereas the DRU-Net model enhances segmentation by fine-tuning a U-Net on a down-sampled image. While both methods achieved good results, our approach—using a U-Net on down-sampled images—proved faster and highly efficient. Notably, our model using the PSA approach achieved comparable results despite using a much smaller dataset.

Transformer-based models like Swin-UNet and InternImage have demonstrated impressive performance in medical image segmentation tasks due to their ability to capture global contextual information through self-attention mechanisms [22,74]. However, transformer architectures typically have higher model complexity due to extensive self-attention operations and large parameter counts, which can result in increased computational demands compared to traditional CNNs [75]. In contrast, our proposed CNN-based DRU-Net maintains competitive segmentation performance with relatively lower computational requirements, potentially making it more suitable for deployment in resource-constrained clinical environments.

Similar to H2G-Net, our proposed model, DRU-Net, also utilizes a cascaded design with two stages of PWC and refinement, and has achieved comparable results [23]. Although H2G-Net uses a lightweight PWC and a relatively heavier U-Net for refinement, our architecture—DRU-Net—demonstrated better performance when using a heavier feature extractor (PWC) combined with a lightweight U-Net. This architectural choice is particularly beneficial in scenarios with limited training data. In such cases, placing the model's capacity earlier in the pipeline allows it to capture more discriminative and generalizable features during the initial extraction stage, while a simpler refinement network, like a lightweight U-Net, helps to avoid overfitting during the later stages. This balance ensures

that the network focuses on learning robust features without excessive parameter overhead in the refinement phase. Pedersen et al. introduced a balancing technique to ensure equitable representation of available categories [23]. This helps minimize bias toward specific tissue types or tumor characteristics.

In this study, we also encountered some challenges due to the significant class imbalance between the patches derived from the WTA approach. Addressing the resultant low precision, a comprehensive strategy was implemented to improve model accuracy. Key interventions included resampling techniques, both under- and over-sampling, as well as the incorporation of focal loss, which specifically helps to address class imbalance by modulating the loss function to focus on harder-to-classify examples [76]. Furthermore, we explored the clustering of similar tissue types before sampling, the use of a weighted loss function, and adjustments to the decision threshold.

In the training phase of the many-shot model using PSA-derived samples, we deliberately introduced a controlled imbalance to optimize threshold settings and enhance performance. Experiments suggested that the deliberately-induced imbalance may offer improved performance compared to methods such as resampling, under-/over-sampling, focal loss, clustered tissue sampling, weighted loss functions, and threshold tuning [76]. However, this approach poses a risk of bias, requiring careful calibration and ongoing monitoring to prevent skewed results. The DRU-Net model's performance was validated externally, trained on the NLCB dataset and tested on 20 slides from the HULC cohort.

The decrease in performance after fine-tuning the U-Net layers of the H2G-Net may be due to the relatively small number of annotated WSIs available in our study. Conversely, the DRU-Net network's superior performance under similar conditions suggests the efficacy of the DR-Fused network, accompanied by a relatively lightweight U-Net architecture in data-scarce scenarios.

The relatively low performance of the original H2G-Net on NSCLCs with no fine-tuning can be explained by different tissue morphology, growth pattern, and stromal invasions, which can mislead the model during inference [42,77–83].

To analyze the effect of the proposed U-Net refinement network, we compared Figures 3 and 4. Our results indicate that refining the PWC heatmap with the suggested refinement network improved the performance of the evaluated models. However, the main strengths and weaknesses of the models compared to each other directly stem from the PWC models and the training methods used. Additionally, combining the two processes seems to improve and reduce the variance in the segmentation DSC values, indicating that the refinement models have learned to understand overall patterns and connections, leading to a better segmentation.

The difference observed in the average DSCs between the PWC models indicates that the models trained using PSA outperformed the WTA approach under limited data conditions. This was likely due to the inadequate separability of the feature distributions between tumor and non-tumor. In the WTA approach, the method involved annotating entire tumor regions, which often included patches where the feature distributions of tumor and non-tumor tissues overlapped significantly. This overlap reduced the separability and weakened the discriminatory power of the classification models trained using this approach. Consequently, the distinction between tumor and non-tumor features in these patches became less pronounced, leading to potential misclassifications.

The PSA method adopted a more selective approach by targeting patches for annotation based on their discriminative morphology. By focusing on patches where tumor and non-tumor features were clearly distinguishable, PSA enhanced the model's ability to accurately classify these features. This selective annotation process effectively increased the inter-class variance while reducing the intra-class variance, thus significantly improving

the overall performance of the classification models in distinguishing between tumor and non-tumor tissues under conditions of limited data. In the WTA approach, the mentioned inseparable feature distribution affected the loss function negatively, resulting in lower accuracy. This was most likely rooted in the fact that the tumor regions also include other cell types than the invasive epithelial cells. By using histopathological knowledge for selecting areas with the most relevant features in PSA, the variation in the features between the two classes could be increased.

It should, nonetheless, be noted that in our case, the PSA and WTA methods were applied to different datasets. Therefore, the observed performance differences do not constitute a statistical comparison, and no definitive claims can be made about the superiority of one approach over the other.

Our study indicates that employing few-shot learning in conjunction with a clustering approach can achieve accuracy levels comparable to methods reliant on extensive datasets, potentially mitigating the need for large-scale data collection. The few-shot learning approach can be beneficial when there is a high degree of similarity within each class of tissue types and a clear distinction between the classes in the feature space [84].

One of the novel techniques presented here was utilizing an evolutionary optimization technique to determine the optimal number of clusters (classes) to minimize intra-cluster variance and maximize inter-cluster variance prior to few-shot training. This method optimally configures clusters to reflect the most coherent and meaningful class structures, which is crucial when the available training data are scarce. By focusing on minimizing intra-cluster variance and minimizing inter-cluster similarity, the approach enhances the model's ability to generalize from limited examples, a critical aspect in few-shot scenarios where the risk of overfitting is high. Evolutionary algorithms also offer adaptability and flexibility. This enables the model to effectively handle varying data types and distributions. This pre-training optimization led to more efficient training and improved model performance by grouping patches into different classes.

The qualitative assessment of our results suggests that the DRU-Net model shows limitations in accurately delineating the tumor periphery. This challenge was particularly evident in regions with fibrosis, reactive tissue, or inflammation, where the model tends to produce false-positive and false-negative segmentations. This limitation is most likely due to the limited size of the training data; with a larger dataset containing more examples of these complex regions correctly annotated, the model's performance in these areas might be significantly improved.

A key limitation of our study is the modest size of our dataset of 97 WSIs from the HULC cohort. Generating pixel-perfect tumor outlines on WSIs is an extremely labor-intensive process and time-consuming for an expert pathologist (including review and correction), even when using semi-automated contouring tools. Under these resource constraints, expanding beyond 97 expertly whole tumor annotated slides was simply not feasible within the project timeline. We chose to create this new dataset rather than relying on existing publicly available annotated datasets because most of them focus exclusively on neoplastic cells at the pixel level, often excluding the surrounding stroma and other intermixed cell types present within the tumor region. Additionally, comparable datasets that adopt a whole-tumor region approach typically lack the resolution and accuracy required to precisely capture tumor borders and small, scattered tumor cell clusters.

Despite the limited dataset size, we observed a consistent alignment between training and validation loss curves along with a stable performance on the external test set. This suggests that the model's performance is not merely a result of overfitting but a genuine generalization to the tested unseen data.

In the future, we suggest reducing the model size using advanced attention-focusing mechanisms and a multi-scale patch-wise classifier to better incorporate information at different scales. Employing anomaly detection algorithms might help identify reactive tissue outliers that contribute to false-positive classifications.

Although HE is the standard coloring method for the assessment of histopathology slides, the stain can vary from laboratory to laboratory. Hence, testing on non-Norwegian cohorts and from laboratories with different staining techniques can be beneficial. We searched extensively for open-access lung tumor-segmentation datasets that include tumor outlines demarcated according to the same protocol we employ, but did not identify any that match our annotation style or resolution. As a result, quantitative evaluation of segmentation generalizability beyond the NLCB and HULC cohorts remains challenging. For future work, we suggest addressing this gap with a proper dataset with multi-institutional WSI cohorts capturing a range of scanners, staining protocols, and patient demographics. After establishing the generalizability, the model should be set up for clinical validation.

Additionally, Mask R-CNN architectures are highly effective in distinguishing complex patterns that can be used for better tumor border delineation. Implementing Bayesian neural networks can potentially improve the prediction of tumor boundaries while quantifying the uncertainty of predictions. To more effectively incorporate global WSI context, methods such as Markov or conditional random fields could be integrated along with PWC or transformer architectures. Using this approach will ensure that segmented areas are not only based on local pixel values. To further improve the differentiation between the two classes, we suggest Neuro-Fuzzy Systems, maintaining the learning capabilities of neural networks while applying the reasoning capabilities of fuzzy logic. To overcome the challenge of limited data, we suggest using unsupervised domain adaptation algorithms to leverage annotated data from other histopathology source domains.

5. Conclusions

In conclusion, we have introduced DRU-Net for non-small cell lung cancer tumor delineation in WSIs. Our new model, which synergistically integrates truncated DenseNet201 and ResNet101V2 with a U-Net-based refinement stage, demonstrated high performance in NSCLCs over various tested methods. Our patch-wise classifier achieves superior performance through an advanced multi-lens distortion augmentation technique and an optimized PSA strategy.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/jimaging11050166/s1. Reference [85] is cited in the supplementary materials. PDF S1: Supplementary Information for Segmentation of Non-Small Cell Lung carcinomas: Introducing DRU-Net and Multi-Lens Distortion. This file contains the following: Methods that were tested to address the class imbalance in the data; Details on the alternative methods that were used for comparison against the proposed method including H2G-Net, few-shot learning, and clustering techniques; Explaining the feature distribution challenges and the deliberately induced data imbalance. The source code is made openly available https://github.com/AICAN-Research/DRU-Net (accessed on 29 April 2024).

Author Contributions: Conceptualization, S.O. and H.S.; Data curation, M.V., V.G.D., S.G.F.W., M.D.H., M.P.R., L.A.A. and H.S.; Formal analysis, S.O., A.P. and E.S.; Funding acquisition, L.A.A. and H.S.; Investigation, S.O., M.V., V.G.D., S.G.F.W., M.D.H. and G.K.; Methodology, S.O., A.P., E.S., T.L. and G.K.; Project administration, H.S.; Resources, H.S.; Software, S.O., A.P. and E.S.; Supervision, H.S.; Validation, S.O. and M.V.; Visualization, S.O., A.P., E.S. and M.D.H.; Writing—original draft, S.O.; Writing—review & editing, S.O., M.V., A.P., V.G.D., M.H., S.G.F.W., M.D.H., T.L., M.P.R., L.A.A., G.K. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results received funding from The Liaison Committee for Education, Research, and Innovation in Central Norway (identifiers 2021/928 and 2022/787). The work was also supported by grants from the Research Council of Norway through its Centres of Excellence funding scheme, project number 223250 (to L.A.A.).

Institutional Review Board Statement: This study was conducted in accordance with the Declaration of Helsinki and approved by the Regional Committee for Medical and Health Sciences Research Ethics (REK) Norway (identifier 257624, date of approval 21 June 2021), the institutional Personal Protection Officer and local Data Access Committee at the Norwegian University of Science and Technology and St. Olavs hospital, Trondheim University Hospital (identifier 2021/1374, date of approval 27 May 2022).

Informed Consent Statement: Written informed consent was obtained from all subjects involved in this study who were recruited from the Norwegian Lung Cancer Biobank (NLCB cohort). For subjects recruited from the University of Bergen (HULC cohort), the Regional Committee for Medical and Health Sciences Research Ethics (REK) Norway granted ethical approval to waive patient consent, since many patients were already deceased and obtaining consent only from surviving subjects would lead to study bias.

Data Availability Statement: The datasets generated and/or analysed during the current study are not publicly available due to the sensitive nature of personal medical data from patients who may still be alive, but might be available from Associate Professor Hanne Sorger upon request, on a mutual collaborative basis.

Acknowledgments: We extend our gratitude to Borgny Ytterhus for her contributions to this project.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rami-Porta, R. Future perspectives on the TNM staging for lung cancer. *Cancers* **2021**, *13*, 1940. [CrossRef] [PubMed]
2. Lim, C.; Tsao, M.; Le, L.; Shepherd, F.; Feld, R.; Burkes, R.; Liu, G.; Kamel-Reid, S.; Hwang, D.; Tanguay, J.; et al. Biomarker testing and time to treatment decision in patients with advanced nonsmall-cell lung cancer. *Ann. Oncol.* **2015**, *26*, 1415–1421. [CrossRef] [PubMed]
3. Woodard, G.A.; Jones, K.D.; Jablons, D.M. Lung cancer staging and prognosis. In *Lung Cancer Treatment and Research*; Springer: Cham, Switzerland, 2016; pp. 47–75.
4. Hanna, M.G.; Reuter, V.E.; Samboy, J.; England, C.; Corsale, L.; Fine, S.W.; Agaram, N.P.; Stamelos, E.; Yagi, Y.; Hameed, M.; et al. Implementation of digital pathology offers clinical and operational increase in efficiency and cost savings. *Arch. Pathol. Lab. Med.* **2019**, *143*, 1545–1555. [CrossRef] [PubMed]
5. Bera, K.; Schalper, K.A.; Rimm, D.L.; Velcheti, V.; Madabhushi, A. Artificial intelligence in digital pathology—New tools for diagnosis and precision oncology. *Nat. Rev. Clin. Oncol.* **2019**, *16*, 703–715. [CrossRef]
6. Sakamoto, T.; Furukawa, T.; Lami, K.; Pham, H.H.N.; Uegami, W.; Kuroda, K.; Kawai, M.; Sakanashi, H.; Cooper, L.A.D.; Bychkov, A.; et al. A narrative review of digital pathology and artificial intelligence: Focusing on lung cancer. *Transl. Lung Cancer Res.* **2020**, *9*, 2255–2276. [CrossRef]
7. Niazi, M.K.K.; Parwani, A.V.; Gurcan, M.N. Digital pathology and artificial intelligence. *Lancet Oncol.* **2019**, *20*, e253–e261. [CrossRef]
8. Kurc, T.; Bakas, S.; Ren, X.; Bagari, A.; Momeni, A.; Huang, Y.; Zhang, L.; Kumar, A.; Thibault, M.; Qi, Q.; et al. Segmentation and classification in digital pathology for glioma research: Challenges and deep learning approaches. *Front. Neurosci.* **2020**, *14*, 27. [CrossRef]
9. Ho, D.J.; Yarlagadda, D.V.; D'Alfonso, T.M.; Hanna, M.G.; Grabenstetter, A.; Ntiamoah, P.; Brogi, E.; Tan, L.K.; Fuchs, T.J. Deep multi-magnification networks for multi-class breast cancer image segmentation. *Comput. Med. Imaging Graph.* **2021**, *88*, 101866. [CrossRef]
10. Qaiser, T.; Tsang, Y.W.; Taniyama, D.; Sakamoto, N.; Nakane, K.; Epstein, D.; Rajpoot, N. Fast and accurate tumor segmentation of histology images using persistent homology and deep convolutional features. *Med. Image Anal.* **2019**, *55*, 1–14. [CrossRef]
11. Zhao, T.; Fu, C.; Tie, M.; Sham, C.W.; Ma, H. RGSB-UNet: Hybrid Deep Learning Framework for Tumour Segmentation in Digital Pathology Images. *Bioengineering* **2023**, *10*, 957. [CrossRef]

12. Viswanathan, V.S.; Toro, P.; Corredor, G.; Mukhopadhyay, S.; Madabhushi, A. The state of the art for artificial intelligence in lung digital pathology. *J. Pathol.* **2022**, *257*, 413–429. [CrossRef] [PubMed]
13. Wang, S.; Yang, D.M.; Rong, R.; Zhan, X.; Fujimoto, J.; Liu, H.; Minna, J.; Wistuba, I.I.; Xie, Y.; Xiao, G. Artificial intelligence in lung cancer pathology image analysis. *Cancers* **2019**, *11*, 1673. [CrossRef] [PubMed]
14. Davri, A.; Birbas, E.; Kanavos, T.; Ntritsos, G.; Giannakeas, N.; Tzallas, A.T.; Batistatou, A. Deep Learning for Lung Cancer Diagnosis, Prognosis and Prediction Using Histological and Cytological Images: A Systematic Review. *Cancers* **2023**, *15*, 3981. [CrossRef]
15. Cheng, J.; Huang, K.; Xu, J. Computational pathology for precision diagnosis, treatment, and prognosis of cancer. *Front. Med.* **2023**, *10*, 1209666. [CrossRef]
16. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 12179–12188.
17. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA,11–15 June 2023; pp. 14408–14419.
18. Park, N.; Kim, S. How do vision transformers work? *arXiv* **2022**, arXiv:2202.06709.
19. Kassani, S.H.; Kassani, P.H.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Deep transfer learning based model for colorectal cancer histopathology segmentation: A comparative study of deep pre-trained models. *Int. J. Med. Inform.* **2022**, *159*, 104669. [CrossRef]
20. Lin, H.; Chen, H.; Dou, Q.; Wang, L.; Qin, J.; Heng, P.A. ScanNet: A Fast and Dense Scanning Framework for Metastastic Breast Cancer Detection from Whole-Slide Image. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 539–546. [CrossRef]
21. Zeng, L.; Tang, H.; Wang, W.; Xie, M.; Ai, Z.; Chen, L.; Wu, Y. MAMC-Net: An effective deep learning framework for whole-slide image tumor segmentation. *Multimed. Tools Appl.* **2023**, *82*, 39349–39369. [CrossRef]
22. Wang, L.; Pan, L.; Wang, H.; Liu, M.; Feng, Z.; Rong, P.; Chen, Z.; Peng, S. DHUnet: Dual-branch hierarchical global–local fusion network for whole slide image segmentation. *Biomed. Signal Process. Control* **2023**, *85*, 104976. [CrossRef]
23. Pedersen, A.; Smistad, E.; Rise, T.V.; Dale, V.G.; Pettersen, H.S.; Nordmo, T.A.S.; Bouget, D.; Reinertsen, I.; Valla, M. H2G-Net: A multi-resolution refinement approach for segmentation of breast cancer region in gigapixel histopathological images. *Front. Med.* **2022**, *9*, 971873. [CrossRef]
24. Albusayli, R.; Graham, D.; Pathmanathan, N.; Shaban, M.; Minhas, F.; Armes, J.E.; Rajpoot, N.M. Simple non-iterative clustering and CNNs for coarse segmentation of breast cancer whole-slide images. In Proceedings of the Medical Imaging 2021: Digital Pathology, Online, 15–20 February 2021; Volume 11603, pp. 100–108.
25. Chelebian, E.; Avenel, C.; Ciompi, F.; Wählby, C. DEPICTER: Deep representation clustering for histology annotation. *Comput. Biol. Med.* **2024**, *170*, 108026. [CrossRef]
26. Yan, J.; Chen, H.; Li, X.; Yao, J. Deep contrastive learning based tissue clustering for annotation-free histopathology image analysis. *Comput. Med. Imaging Graph.* **2022**, *97*, 102053. [CrossRef] [PubMed]
27. Deuschel, J.; Firmbach, D.; Geppert, C.I.; Eckstein, M.; Hartmann, A.; Bruns, V.; Kuritcyn, P.; Dexl, J.; Hartmann, D.; Perrin, D.; et al. Multi-prototype few-shot learning in histopathology. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 620–628.
28. Shakeri, F.; Boudiaf, M.; Mohammadi, S.; Sheth, I.; Havaei, M.; Ayed, I.B.; Kahou, S.E. FHIST: A benchmark for few-shot classification of histological images. *arXiv* **2022**, arXiv:2206.00092.
29. Titoriya, A.K.; Singh, M.P. Few-Shot Learning on Histopathology Image Classification. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 251–256.
30. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
31. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
32. Krikid, F.; Rositi, H.; Vacavant, A. State-of-the-Art Deep Learning Methods for Microscopic Image Segmentation: Applications to Cells, Nuclei, and Tissues. *J. Imaging* **2024**, *10*, 311. [CrossRef] [PubMed]
33. Greeley, C.; Holder, L.; Nilsson, E.E.; Skinner, M.K. Scalable deep learning artificial intelligence histopathology slide analysis and validation. *Sci. Rep.* **2024**, *14*, 26748. [CrossRef]

34. Deng, R.; Cui, C.; Liu, Q.; Yao, T.; Remedios, L.W.; Bao, S.; Landman, B.A.; Wheless, L.E.; Coburn, L.A.; Wilson, K.T.; et al. Segment anything model (sam) for digital pathology: Assess zero-shot segmentation on whole slide imaging. In Proceedings of the IS&T International Symposium on Electronic Imaging, San Francisco, CA, USA, 2–6 February 2025; Volume 37, p. COIMG–132.
35. Ma, J.; He, Y.; Li, F.; Han, L.; You, C.; Wang, B. Segment anything in medical images. *Nat. Commun.* **2024**, *15*, 654. [CrossRef]
36. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Hatlen, P. Lung Cancer—Influence of Comorbidity on Incidence and Survival: The Nord-Trøndelag Health Study. Ph.D. Thesis, Norges Teknisk-Naturvitenskapelige Universitet, Det Medisinske Fakultet, Institutt for Sirkulasjon og Bildediagnostikk, Trondheim, Norway, 2014.
39. Ramnefjell, M.; Aamelfot, C.; Helgeland, L.; Akslen, L.A. Vascular invasion is an adverse prognostic factor in resected non–small-cell lung cancer. *Apmis* **2017**, *125*, 197–206. [CrossRef]
40. Hatlen, P.; Grønberg, B.H.; Langhammer, A.; Carlsen, S.M.; Amundsen, T. Prolonged survival in patients with lung cancer with diabetes mellitus. *J. Thorac. Oncol.* **2011**, *6*, 1810–1817. [CrossRef]
41. Yoh Watanabe, M. TNM classification for lung cancer. *Ann. Thorac. Cardiovasc. Surg.* **2003**, *9*, 343–350.
42. Travis, W. The 2015 WHO classification of lung tumors. *Der Pathol.* **2014**, *35*, 188. [CrossRef]
43. Valla, M.; Vatten, L.J.; Engstrøm, M.J.; Haugen, O.A.; Akslen, L.A.; Bjørngaard, J.H.; Hagen, A.I.; Ytterhus, B.; Bofin, A.M.; Opdahl, S. Molecular subtypes of breast cancer: Long-term incidence trends and prognostic differences. *Cancer Epidemiol. Biomark. Prev.* **2016**, *25*, 1625–1634. [CrossRef]
44. Deng, L. The MNIST Database of Handwritten Digit Images for Machine Learning Research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
45. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
46. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, University of Tront, Toronto, ON, Canada, 2009.
47. Bankhead, P.; Loughrey, M.B.; Fernández, J.A.; Dombrowski, Y.; McArt, D.G.; Dunne, P.D.; McQuaid, S.; Gray, R.T.; Murray, L.J.; Coleman, H.G.; et al. QuPath: Open source software for digital pathology image analysis. *Sci. Rep.* **2017**, *7*, 16878. [CrossRef]
48. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
49. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: https://www.tensorflow.org (accessed on 10 November 2023)
50. Smistad, E.; Bozorgi, M.; Lindseth, F. FAST: Framework for heterogeneous medical image computing and visualization. *Int. J. Comput. Assist. Radiol. Surg.* **2015**, *10*, 1811–1822. [CrossRef]
51. Smistad, E.; Østvik, A.; Pedersen, A. High performance neural network inference, streaming, and visualization of medical images using FAST. *IEEE Access* **2019**, *7*, 136310–136321. [CrossRef]
52. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *120*, 122–125.
53. Clark, A. Pillow (PIL Fork) Documentation. 2015. Available online: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf (accessed on 10 November 2023).
54. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
55. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
56. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
57. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
58. ONNX. Convert TensorFlow, Keras, Tensorflow.js and Tflite Models to ONNX. 2024. Available online: https://github.com/onnx/tensorflow-onnx (accessed on 10 November 2023).
59. Pedersen, A.; Valla, M.; Bofin, A.M.; De Frutos, J.P.; Reinertsen, I.; Smistad, E. FastPathology: An open-source platform for deep learning-based research and decision support in digital pathology. *IEEE Access* **2021**, *9*, 58216–58229. [CrossRef]
60. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

61. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Advances in Information Retrieval, Proceedings of the 27th European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
62. Kim, H.; Monroe, J.I.; Lo, S.; Yao, M.; Harari, P.M.; Machtay, M.; Sohn, J.W. Quantitative evaluation of image segmentation incorporating medical consideration functions. *Med. Phys.* **2015**, *42*, 3013–3023. [CrossRef] [PubMed]
63. Patro, B.N.; Lunayach, M.; Patel, S.; Namboodiri, V.P. U-CAM: Visual Explanation using Uncertainty based Class Activation Maps. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7444–7453.
64. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
65. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
66. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3319–3328.
67. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
68. Tan, M.; Le, Q. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021; pp. 10096–10106.
69. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
70. Gai, L.; Xing, M.; Chen, W.; Zhang, Y.; Qiao, X. Comparing CNN-based and transformer-based models for identifying lung cancer: Which is more effective? *Multimed. Tools Appl.* **2024**, *83*, 59253–59269. [CrossRef]
71. Sangeetha, S.; Mathivanan, S.K.; Muthukumaran, V.; Cho, J.; Easwaramoorthy, S.V. An Empirical Analysis of Transformer-Based and Convolutional Neural Network Approaches for Early Detection and Diagnosis of Cancer Using Multimodal Imaging and Genomic Data. *IEEE Access* **2025**, *13*, 6120–6145. [CrossRef]
72. Lakshmanan, B.; Anand, S.; Jenitha, T. Stain removal through color normalization of haematoxylin and eosin images: A review. *Proc. J. Phys. Conf. Ser.* **2019**, *1362*, 012108. [CrossRef]
73. Tellez, D.; Litjens, G.; Bándi, P.; Bulten, W.; Bokhorst, J.M.; Ciompi, F.; Van Der Laak, J. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.* **2019**, *58*, 101544. [CrossRef]
74. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision—ECCV 2022 Workshops, Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022*; Springer: Cham, Switzerland, 2022; pp. 205–218.
75. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
76. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
77. Menon, A.; Singh, P.; Vinod, P.; Jawahar, C. Exploring Histological Similarities Across Cancers from a Deep Learning Perspective. *Front. Oncol.* **2022**, *12*, 842759. [CrossRef]
78. Kashima, J.; Kitadai, R.; Okuma, Y. Molecular and Morphological Profiling of Lung Cancer: A Foundation for "Next-Generation" Pathologists and Oncologists. *Cancers* **2019**, *11*, 599. [CrossRef] [PubMed]
79. Petersen, I. The morphological and molecular diagnosis of lung cancer. *Dtsch. Ärztebl. Int.* **2011**, *108*, 525–531. [CrossRef] [PubMed]
80. Inamura, K. Lung cancer: Understanding its molecular pathology and the 2015 WHO classification. *Front. Oncol.* **2017**, *7*, 193. [CrossRef] [PubMed]
81. Zhao, S.; Chen, D.P.; Fu, T.; Yang, J.C.; Ma, D.; Zhu, X.Z.; Wang, X.X.; Jiao, Y.P.; Jin, X.; Xiao, Y.; et al. Single-cell morphological and topological atlas reveals the ecosystem diversity of human breast cancer. *Nat. Commun.* **2023**, *14*, 6796. [CrossRef]
82. Binder, A.; Bockmayr, M.; Hägele, M.; Wienert, S.; Heim, D.; Hellweg, K.; Ishii, M.; Stenzinger, A.; Hocke, A.; Denkert, C.; et al. Morphological and molecular breast cancer profiling through explainable machine learning. *Nat. Mach. Intell.* **2021**, *3*, 355–366. [CrossRef]
83. Tan, P.H.; Ellis, I.; Allison, K.; Brogi, E.; Fox, S.B.; Lakhani, S.; Lazar, A.J.; Morris, E.A.; Sahin, A.; Salgado, R.; et al. The 2019 WHO classification of tumours of the breast. *Histopathology* **2020**, *77*, 181–185. [CrossRef]

84. Qi, Y.; Sun, H.; Liu, N.; Zhou, H. A Task-Aware Dual Similarity Network for Fine-Grained Few-Shot Learning. In *PRICAI 2022: Trends in Artificial Intelligence, Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Shanghai, China, 10–13 November* 2022; Springer: Cham, Switzerland, 2022; pp. 606–618.
85. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. *Slic Superpixels*; Ecole Polytechnique F´edrale de Lausanne (EPFL): Lausanne, Switzerland, 2010.

Journal of *Imaging*

MDPI

Article

"ShapeNet": A Shape Regression Convolutional Neural Network Ensemble Applied to the Segmentation of the Left Ventricle in Echocardiography

Eduardo Galicia Gómez [1], Fabián Torres-Robles [2], Jorge Perez-Gonzalez [3] and Fernando Arámbula Cosío [3,*]

[1] Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico; gagoed@comunidad.unam.mx
[2] Laboratorio de Física Médica, Instituto de Física, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico; ftrobles@fisica.unam.mx
[3] Unidad Académica del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas en Yucatán, Universidad Nacional Autónoma de México, Merida 97357, Mexico; jorge.perez@iimas.unam.mx
* Correspondence: fernando.arambula@iimas.unam.mx; Tel.: +52-999-399-0901 (ext. 7800)

Abstract: Left ventricle (LV) segmentation is crucial for cardiac diagnosis but remains challenging in echocardiography. We present ShapeNet, a fully automatic method combining a convolutional neural network (CNN) ensemble with an improved active shape model (ASM). ShapeNet predicts optimal pose (rotation, translation, and scale) and shape parameters, which are refined using the improved ASM. The ASM optimizes an objective function constructed from gray-level profiles concatenated into a single contour appearance vector. The model was trained on 4800 augmented CAMUS images and tested on both CAMUS and EchoNet databases. It achieved a Dice coefficient of 0.87 and a Hausdorff Distance (HD) of 4.08 pixels on CAMUS, and a Dice coefficient of 0.81 with an HD of 10.21 pixels on EchoNet, demonstrating robust performance across datasets. These results highlight the improved accuracy in HD compared to previous semantic and shape-based segmentation methods by generating statistically valid LV contours from ultrasound images.

Keywords: active shape models; convolutional neural networks; echocardiography; left ventricle segmentation; shape constraints; ultrasound segmentation

Academic Editors: Ester Bonmati Coll and Barbara Villarini

Received: 11 April 2025
Revised: 2 May 2025
Accepted: 17 May 2025
Published: 20 May 2025

Citation: Gómez, E.G.; Torres-Robles, F.; Perez-Gonzalez, J.; Arámbula Cosío, F. "ShapeNet": A Shape Regression Convolutional Neural Network Ensemble Applied to the Segmentation of the Left Ventricle in Echocardiography. *J. Imaging* **2025**, *11*, 165. https://doi.org/10.3390/jimaging11050165

1. Introduction

According to the Centers for Disease Control and Prevention (CDC) in the United States [1], one in four individuals with heart failure die each year, with most cases linked to dysfunction in the left ventricle (LV). The LV plays a crucial role in distributing oxygenated blood throughout the body via the aortic valve. Any malfunction in this process can lead to severe complications within the circulatory system and other organs. As reported by Berman et al. [1], heart failure often results from compromised LV function, typically due to structural changes in the ventricular wall or the inability of the LV to fill or eject blood effectively. Patients with cardiac disease frequently experience symptoms such as dyspnea, fatigue, and fluid retention, which can further progress to ischemia, muscle disease, pulmonary congestion, and elevated heart rate. To assess ventricular function, a range of imaging and signal processing methods are currently available, including physical examination, X-rays, electrocardiogram (ECG), magnetic resonance imaging (MRI), and echocardiography (ultrasound). Among these, echocardiography provides valuable insights into both systolic and diastolic LV function, ventricular morphology, and conditions such as aneurysms, along with mitral, tricuspid, aortic, and pulmonary valve function [2].

Due to its non-invasive nature and excellent cost–benefit ratio, echocardiography is widely used in clinical practice for evaluating ventricular function [2].

However, accurately defining the LV contour and shape remains a critical challenge for diagnosing heart failure. Computational methods have emerged to support cardiologists in producing more precise and efficient diagnoses. Currently, deep-learning-based approaches, particularly those employing semantic segmentation like convolutional neural networks (CNNs), have shown promising results. Nevertheless, these methods can produce anatomically inconsistent or noisy LV contours, including implausible segmentations with irregular or disconnected regions that do not correspond to the expected LV morphology. Such artifacts, commonly referred to as *blobs* in medical image analysis, result from pixel-level misclassification errors inherent in semantic segmentation approaches and may significantly compromise clinical reliability. In this work, we address these limitations by presenting a new method for LV segmentation, and the novelty of this work lies in three key contributions. First, we introduce ShapeNet, a specialized ensemble of CNNs that directly predicts both the pose parameters (rotation, translation, and scale) and shape deformation parameters of a statistical shape model, eliminating the artifacts produced by pixel misclassification in semantic segmentation methods (blobs). Second, we develop an improved ASM that optimizes a global objective function based on concatenated gray-level profiles, demonstrating superior capture range and robustness compared to traditional ASM approaches. Third, our fully automatic pipeline uniquely combines these components to generate anatomically plausible contours, without manual initialization—a significant advantage over semi-automatic methods like BEASM [3]. This integrated approach maintains the flexibility of data-driven deep learning, while utilizing the anatomical validity offered by shape models, as evidenced by our consistent performance across both CAMUS and independent EchoNet datasets.

2. Related Work

This section provides an overview of the key methods in LV segmentation, highlighting their challenges and how our proposed method addresses these limitations.

2.1. Deep Learning Approaches

Recent developments and applications in CNNs [4–6] have led to significant progress in the automatic segmentation of organs across various medical imaging modalities. CNNs have become a commonly used method for segmenting the region of interest (ROI), followed by annotation of the boundary. For example, Chen et al. [7] reviewed prominent deep-learning techniques, highlighting that U-Net and its variants are widely used in medical image segmentation tasks, including ensembles of different architectures of CNNs and transformer networks for organ region detection or using mask region CNNs which produce a bounding box and a binary mask of the organ. In [8–10], the U-Net architecture was successfully applied to the classification of pixels corresponding to the LV. Ansari et al. [11] proposed a novel CNN based on a U-Net backbone with PSP in the skip connections. A thorough evaluation was performed to assess the benefits of preprocessing with a contrast limited adaptive histogram equalization (CLAHE), showing improved results for real-time segmentation of ultrasound videos of the liver. Kang et al. [12] reported a new CNN architecture for the segmentation of the LV in transesophageal ultrasound taken during cardiopulmonary resuscitation procedures (CPR). The CNN includes an attention mechanism and a residual feature aggregation module able to accurately segment the LV in the presence of large shadows and atypical deformations. Zhao et al. [13] reported a semi-supervised echocardiography semantic segmentation method, which is able to segment the left ventricle, epicardium, and left atrium on ultrasound images. The method is based

on a boundary attention transformer net and a multi-task semi-supervised model with consistency constraints. This approach enables effective model training with a partially annotated training set. In addition, Shi et al. [14] proposed a hybrid transformer–CNN architecture to enhance segmentation robustness, combining ResNet-50 for spatial feature extraction with a transformer-based encoder–decoder for global context modeling. Their framework integrates two key modules: a Convolutional Block Attention Module (CBAM) to adaptively fuse CNN and transformer features, enhancing focus on anatomically relevant regions, and a Bridge Attention (BA) mechanism to filter non-relevant features, while refining segmentation maps through multi-level feature aggregation. While CNNs have revolutionized segmentation tasks, these methods face several challenges, especially in applications like echocardiography, where the LV boundaries can be imprecise or noisy. Despite their ability to learn spatial features from large datasets, CNNs tend to produce anatomically inconsistent contours when dealing with variations in heart shape, speckle noise, and imaging artifacts.

2.2. Traditional Machine Learning Methods

Previous machine learning approaches for LV segmentation, including geodesic models [15], level sets [16], and shape-based deformable models [17], offer effective ways to model and deform contours based on anatomical points. Moreover, Statistical Shape Models (SSMs) have been particularly effective in left ventricle segmentation when closely initialized [18]. SSMs provide an effective means to incorporate expert shape knowledge into organ segmentation techniques. These methods rely on predefined shape models and iterative algorithms to find the optimal boundary. However, they often require accurate initialization and are sensitive to the quality of the input data, which makes them less robust when dealing with noisy or low-quality images. Registration-based techniques [19], supervised learning [20], and active appearance models [21] have also been explored for LV segmentation, in combination with the availability of large annotated datasets [22]. These methods integrate image registration with statistical models to align anatomical shapes. While they provide more accurate segmentation in some cases, they are often computationally expensive and require manual intervention for initialization, making them less practical for routine clinical use. Despite their strengths, these approaches have limitations in terms of robustness, scalability, and flexibility, particularly in dealing with complex, real-world datasets like echocardiography images.

2.3. Hybrid Methods

Hybrid methods that combine SSMs, ASMs, and CNNs have gained attention as a way to leverage the strengths of both paradigms. For example, Li et al. [23] introduced a hybrid method where a CNN first detects three key landmarks on the LV: the apex and the starting and end points of the endocardium. These landmarks are used to initialize a deep-snake model [24], which then adjusts the contour using circular convolution. This method demonstrated strong performance on the HMC-QU echocardiography dataset, but its reliance on landmark-based initialization limits its flexibility in more challenging scenarios. Hybrid methods combining region-based CNNs and ASMs have also been explored. Wei-Yen et al. [25] proposed a method where a CNN detects a bounding box around the LV, which is then used to initialize the ASM for final contour refinement. While this approach is promising, it still requires accurate initialization and may struggle with complex deformations. Our approach builds on these hybrid techniques by combining a CNN ensemble (ShapeNet) with a Point Distribution Model (PDM), ensuring fully automatic initialization and the generation of anatomically plausible contours. This integration

enhances robustness by eliminating the need for manual intervention, unlike previous hybrid methods.

2.4. Methods Incorporating Anatomical Constraints

In recent years, methods incorporating anatomical constraints have shown improved performance for LV segmentation. Oktay et al. [26] developed the Anatomically Constrained Neural Network (ACNN), which combines a CNN with an autoencoder trained on ground truth ventricle masks. Anatomical constraints are included during training of the objective function, which is made of a linear combination of cross entropy, shape regularization, and weight decay terms. Shape regularization is implemented with a distance function between the encoded ground truth mask and the encoded prediction of the segmentation CNN. The ACNN architecture was evaluated on both MRI and ultrasound images, demonstrating improved performance over previous approaches.

Gaggion et al. [27] proposed a hybrid CNN architecture that uses graph convolutional networks to impose anatomical constraints on the latent space. This method integrates image-based feature extraction with landmark-based shape modeling, effectively ensuring anatomically valid segmentations. The model is trained with pairs of input images and the corresponding landmark annotations of the organs of interest, the same number of landmarks is used in all the training examples. This approach was validated on chest X-ray images and demonstrated favorable results for anatomical structure segmentation.

Ribeiro et al. [28] presented a fully-automatic hybrid method for LV segmentation in cardiac MRI images, combining deep learning and deformable models. Initially, a ROI containing the LV is extracted using heart movement analysis. A deep learning network (DLN) is then employed to generate an initial segmentation of the LV cavity and myocardium. DLN-based segmentation is subsequently used to estimate exam-specific statistical information about the LV, which helps initialize and constrain a level-set-based deformable model. This deformable model incorporates anatomical constraints to refine the segmentation and generate the final result. In the final step, failed segmentations are detected and corrected using information from adjacent frames. We also reported a modified U-Net in a previous work [29], with a regression layer replacing the final classification layer, enabling the model to predict LV pose and shape parameters directly.

While deep learning methods, traditional shape-based techniques, and hybrid approaches have made significant strides in LV segmentation, each faces limitations related to initialization, anatomical accuracy, and robustness across diverse datasets. Our proposed method addresses these challenges by combining a CNN ensemble with a PDM, providing a fully automated, anatomically statistically consistent, and robust solution for LV segmentation.

3. Materials and Methods

The objective of ShapeNet is the accurate characterization the left ventricle contour through the prediction of optimal pose (rotation, translation, and scale) and shape (deformation vector) parameters for a trained point distribution model (PDM) of the LV in echocardiography. This is achieved by training a CNN with the aforementioned parameters of the PDM that have been accurately adjusted to the left ventricle on each image of the training set. This approach prevents the formation of blobs produced by pixel classification errors in semantic classification, preventing correct characterization of the LV, as illustrated in Figure 1. In contrast, our approach consistently produces statistically valid contours of the left ventricle, which are accurately adjusted by means of the improved ASM proposed in this paper.

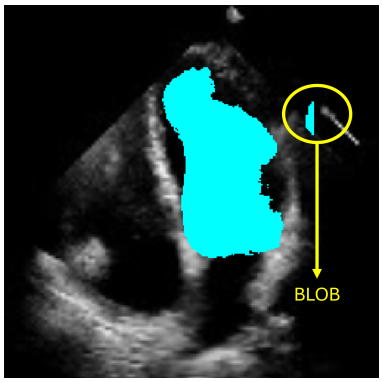


Figure 1. Example of blobs produced by misclassification in semantic segmentation of the left ventricle.

3.1. Point Distribution Model of the Left Ventricle (LV-PDM)

Point distribution models (PDMs) provide a compact representation of a class of shapes; in this case, the shape of the left ventricle. Construction of the PDM was performed as described in [30] by annotating landmark points around the contour of each left ventricle previously marked by an expert. For this LV-PDM, 64 landmarks were selected, which accurately represent the contour of the left ventricle, as shown in Figure 2, using the training data described in Section 4.1. Principal component analysis (PCA) of the normalized landmark training set resulted in five principal modes of variation in the shape of the LV, contained in a principal eigenvector matrix (ϕ), which allows creating new instances of the LV shape $\hat{s}$ using Equation (1). Figure 3 shows some examples of shapes, corresponding to different values of the five weights in vector b.

$$\hat{s} = \bar{s} + \phi b \tag{1}$$

where

$\hat{s}$ = LV shape.
$\bar{s}$ = the mean shape of the training set.
ϕ = principal eigenvector matrix of the training set.
b = vector of deformation parameters of the training set.

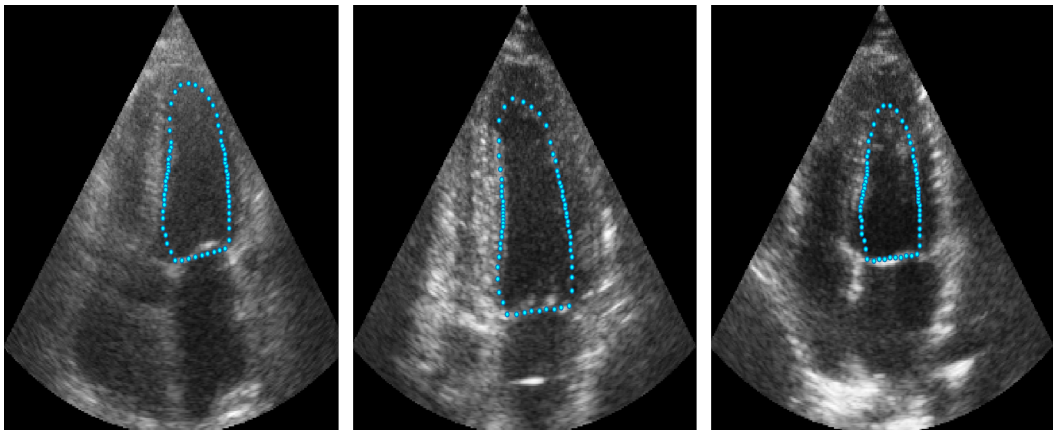


Figure 2. Example of landmarks sampled in LV ultrasound images.

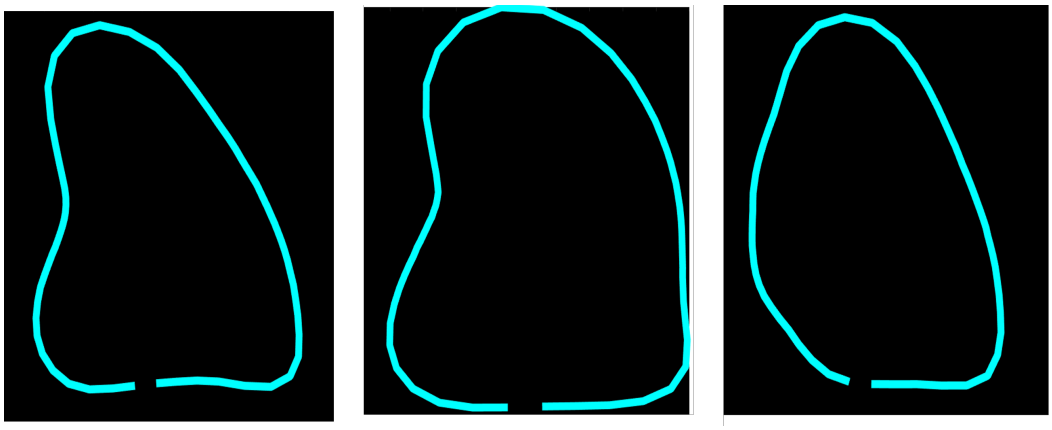


Figure 3. Example of different shapes of the LV using the principal variation modes calculated from the PDM.

The final contour of a ventricle (*R_Shape*) on an echocardiography is defined using Equation (2).

$$R\_Shape = \sigma \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \widehat{s} + \begin{bmatrix} T_x, T_y \end{bmatrix} \quad (2)$$

where

R_Shape = reconstructed LV shape on echocardiography.
$\widehat{s}$ = shape obtained after applying Equation (1).
σ = the scale of the LV.
θ = the rotation of the LV.
Tx = translation in X axis of the LV.
Ty = translation in Y axis of the LV.

ShapeNet was trained to predict all the pose $(\sigma, \theta, T_x, T_y)$ and shape parameters (b), as described below.

3.2. ShapeNet Architecture and Training

The architecture is inspired by encoder–decoder architectures, taking as a basis the encoder part and its power for image feature extraction and dimensionality reduction, hence the origin of the first block. Then, a second block formed by fully connected layers and a regression layer are used to relate the features extracted in the first block with the pose and shape values extracted from the PDM. This architecture is designed to be used and implemented on off-the-shelf computing equipment. The two blocks mentioned above and the full architecture are detailed below.

1. Input layer: This is an image input layer of size 256 × 256 × 1 of the form: (width, height, channels). This is an echocardiographic image of the left ventricle.
2. Convolutional block: This is formed of 2 parts, as shown in Figure 4, the convolution filters are filters of size 3 × 3 with a stride of 1, and the number of filters increases, as shown in Figure 5, starting with 32 filters and doubling the number at each convolution stage. The purpose of these convolutional layers is to extract features from the LV image, which will later be used to link them with the pose and shape features extracted from the previously trained LV-PDM. In addition, there are maxpooling layers in between each convolution stage with a stride of 2; with maxpooling, the dimensionality of the problem is reduced and the most important features of the image are preserved. Furthermore, this allows the training process to be lighter, since it reduces the number of parameters that have to be learned by ShapeNet. An overview of this block is shown in Figure 5.

3. Fully connected block: This consists of a flatten layer followed by a set of fully connected and one dropout layer, to avoid data over-fitting during training. This block links the features extracted in the convolutional block with the pose and shape parameters of the left ventricle and then adjusts the weights to make predictions in the regression layer. Finally, at the end of the fully connected block, a regression layer (Figure 6 pink block) calculates one, pose or shape, parameter: rotation, translation, scale, or b_i for a given input image.

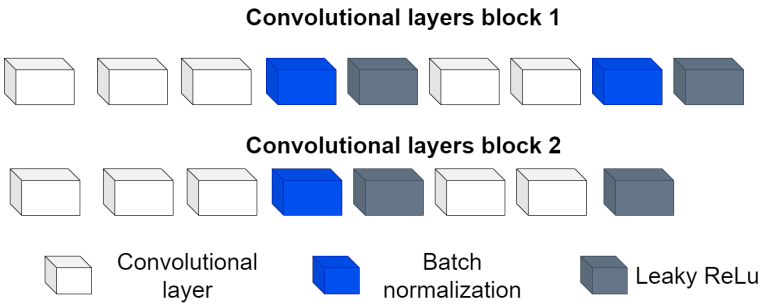


Figure 4. Structure of the convolutional block.

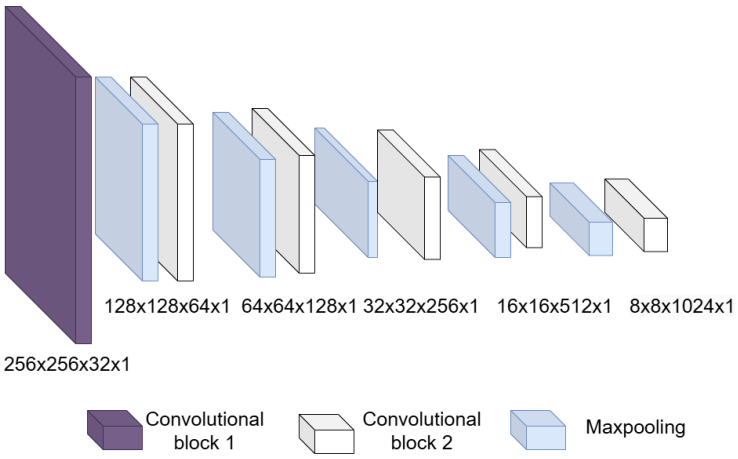


Figure 5. Overview of the convolutional section of ShapeNet.

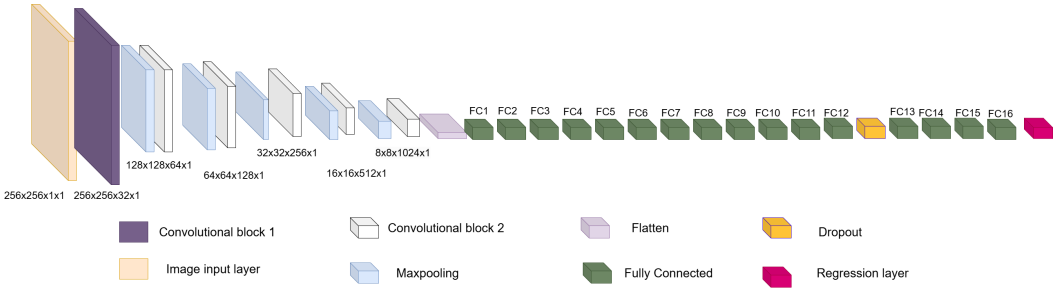


Figure 6. Overview of ShapeNet architecture.

In order to train ShapeNet, the pose parameters rotation (θ), translation (Tx, Ty), and scale (σ), as well as the shape parameters b, were extracted from each LV contour example in the training set as follows:

- Translation (Tx, Ty): The translation was calculated as the mean of the LV contour coordinates on the X axis and Y axis for each example in the training set.
- Rotation (θ): This value was calculated using the binary mask of each example, enclosing it within an ellipse and then calculating the angle between the X axis and the major axis of the ellipse using the second moments of the mask.

- Scale (σ): This was calculated as shown in [30] by normalizing each LV shape to a common scale and minimizing the root-mean-square distance between the corresponding landmarks of the LV i-th training shape and the mean shape ($\bar{s}$) obtained in Section 3.1.
- Shape parameters (b): The deformation parameters were extracted from the expert annotation by solving for the value of b in Equation (1) for each contour in the training set, as shown in Equation (3).

$$b = (\widehat{s}_i - \bar{s}) * \phi' \tag{3}$$

where

$\widehat{s}_i$ = The expert annotation of the i-th example in the training set.
$\bar{s}$ = The mean LV shape of the training set.
ϕ' = The transposed principal eigenvector matrix of the training set.

After extracting the aforementioned parameters of the LV shape, a training vector (V) was constructed with the shape and pose parameters of the left ventricle in the corresponding training image, as shown in Equation (4).

$$V = [I_i, \beta_i] \tag{4}$$

where

I_i = the i-th image of the training set.
β_i = Is the corresponding shape or pose parameter to be learned by the network.

Our approach is based on an ensemble of networks, where each network is trained to optimize one specific parameter of pose and shape of the left ventricle; therefore, we developed dedicated networks for rotation (θ), translation along both the X (T_x) and Y (T_y) axes, scale (σ), and each of the five deformation parameters contained in vector b in Equation (1). Consequently, our method involves the training of nine networks, one for each pose $(\sigma, \theta, T_x, T_y)$ and shape parameter (b).

During the training of each network, a vector V is used as input, depending on which parameter β needs be trained. Consequently, the network adjusts the weights according to a loss function defined as the root mean squared error (RMSE) between the value predicted by the network (β_p) and the parameter β_i contained in the input vector V (see Equation (5)).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |\beta_p - \beta_i|^2}{n}} \tag{5}$$

where:

β_p = Value predicted by the network.
β_i = Input value contained in the V vector.
n = The number of images in the training set.

Additionally, other important hyperparameters to consider during our network training are the batch size and learning rate. For this paper, a batch size of 64 images was used, which we consider to be a moderate size and manageable by most modern GPUs. Moreover, as mentioned by [31], increasing the batch size does not have a significant impact on the accuracy of the gradient calculation when using the ADAM optimizer (which was employed to train these networks). Regarding the learning rate, a rate of 1×10^{-3} was used, as we consider this to be not too high to destabilize training, and not too low to slow down convergence. This leads us to the convergence criterion, which was early stopping. According to our experiments, most networks concluded with an

average of 50 epochs; beyond this number of epochs, the RMSE value for validation did not change. The validation patience parameter was set to 40 iterations.

After the *ShapeNet* has been trained as described above, the next step involves an improved ASM, as detailed in the next section.

3.3. Improved ASM

Following the automatic initialization of the LV-PDM using ShapeNet, an active shape model was used to improve the segmentation accuracy. This ASM is based on the optimization of an objective function constructed with a set of gray-level profiles sampled around the contour of the left ventricle in an echocardiogram. Gaussian filtering ($\sigma = 0.8$, 3×3 kernel) was applied to reduce speckle noise, while preserving edge information of the ultrasound images, before adjusting the ASM, as described in [32].

ASM Objective Function

Our objective function is based on a set of perpendicular gray level profiles of length l, sampled from each landmark point in an image of the training set, as proposed in [30]. All gray profiles are concatenated into a single vector, referred to as the C vector, with a length of $64 \times l$ for our 64-point LV-PDM, as illustrated in Figure 7.

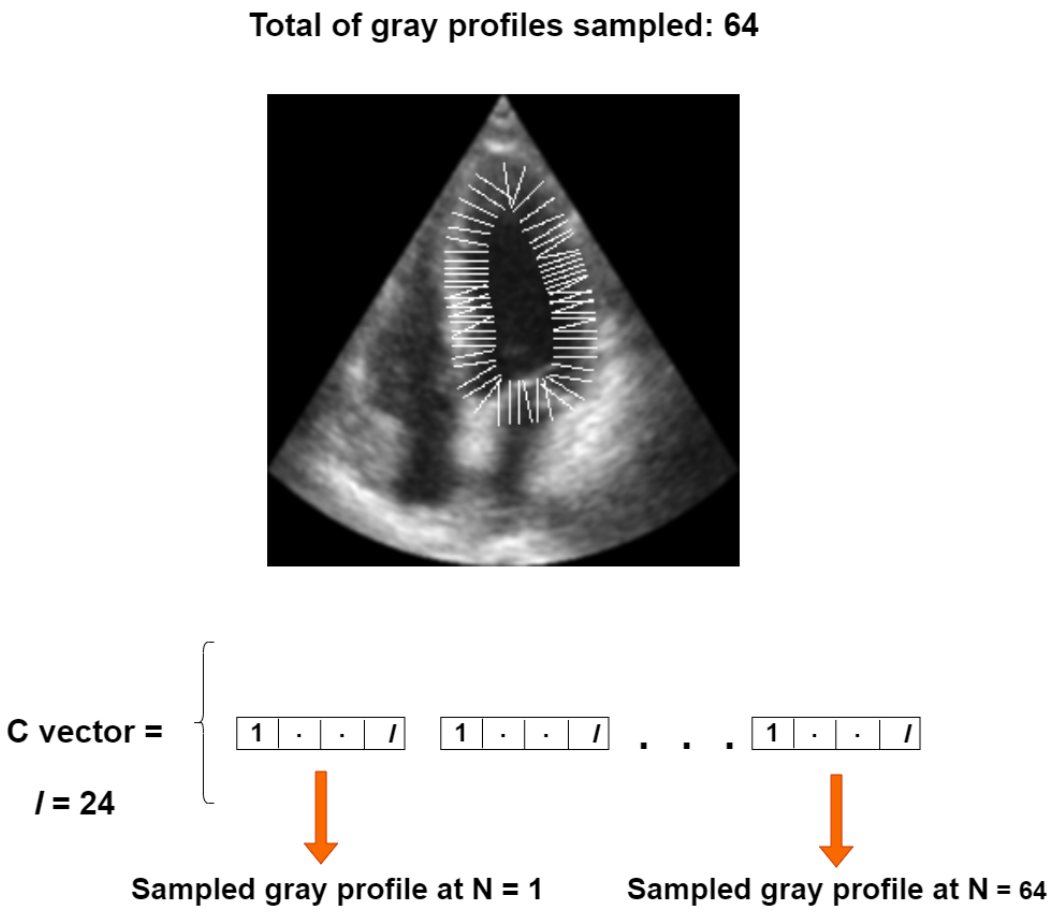


Figure 7. Construction of the C vector for one training instance.

The mean vector of the training set ($\overline{C}$) is given by Equation (6), while the objective function (f_C) represents the RMSE distance between $\overline{C}$ and a newly sampled vector (C_{new}) from a new echocardiogram, as shown in Equation (7). During image segmentation, $f(c)$

is optimized by iteratively alternating the local search of the ASM [30] with optimization using the simplex algorithm, as previously reported in [32].

$$\overline{C} = \frac{\sum_{i=1}^{n} C_i}{n} \tag{6}$$

where

n = number of examples in the training set.
C_i = the i-th C vector sampled on each training image.

$$f_C = \sqrt{\frac{\sum_{i=1}^{l_c} (C_{i_{new}} - \overline{C_i})^2}{l_c}} \tag{7}$$

where

C_{new} = a sampled C vector on a new echocardiogram.
$\overline{C}$ = the mean C vector of the training set.
l_c = The size of the sampled C_{new} vector $(64 \times l)$.

With the improved ASM and ShapeNet training explained, the final step is to perform segmentation of a new image, as described in the following section.

3.4. Left Ventricle Contour Reconstruction and Segmentation

ShapeNet predicts the optimal pose and shape parameters, including rotation (θ), translation (T_x, T_y), and scale (σ), specific to the new image. Using these parameters, the left ventricle contour is reconstructed based on Equations (1) and (2). Once the shape predicted has been reconstructed, this contour acts as an automatic initialization for the improved ASM and will undergo fine-tuning to obtain the final LV contour. Figure 8 depicts the segmentation workflow described.

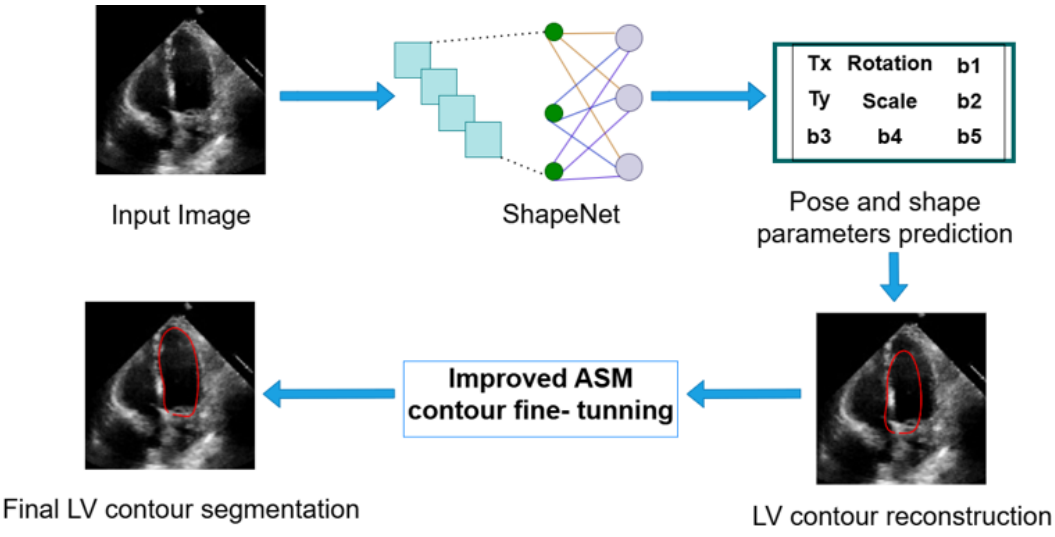


Figure 8. Complete inference pipeline of the ShapeNet+ASM approach.

4. Results

In this section, we present the experiments and results of the ShapeNet and the improved ASM method, which were conducted on a server running on the Ubuntu operating system, with 32 GB of RAM. Additionally, two GPUs were used in parallel: an NVIDIA Tesla K40c and a Tesla T4. Finally, all experiments were implemented in MATLAB R2022b.

4.1. Dataset

The dataset was divided into two parts: training and test. For the training set, we used the CAMUS database [3], comprising a total of 800 images, with 400 corresponding to systole and 400 to diastole end of cycle. Data augmentation was applied to this set of 800 images, including rotation, translation, scaling transformations, and the addition of acoustic shadows, following the work of [33], resulting in a total of 4800 training images. On the other hand, for the test set, 49 systole images and 49 diastole images were reserved from the CAMUS database. Additionally, images extracted by Guzman et al. [34] from the EchoNet Dynamic database [35] were also used for testing; of these images, 207 corresponded to end-systole and 210 to end-diastole. These images were preprocessed following the methodology in [34], which included rigorous frame selection to isolate end-systolic and end-diastolic phases using the dataset's provided timestamps and quality control metrics, automated region-of-interest cropping centered on the left ventricle using landmark detection heuristics, and intensity normalization through min-max scaling of pixel values to [0, 1]. These preprocessed images were then resized to 256 × 256 pixels to match our network ensemble input dimensions.

Figure 9 illustrates an example of the images used for training the ShapeNet. The improved ASM was also trained with the set of 800 images extracted from the CAMUS database before the data augmentation process.

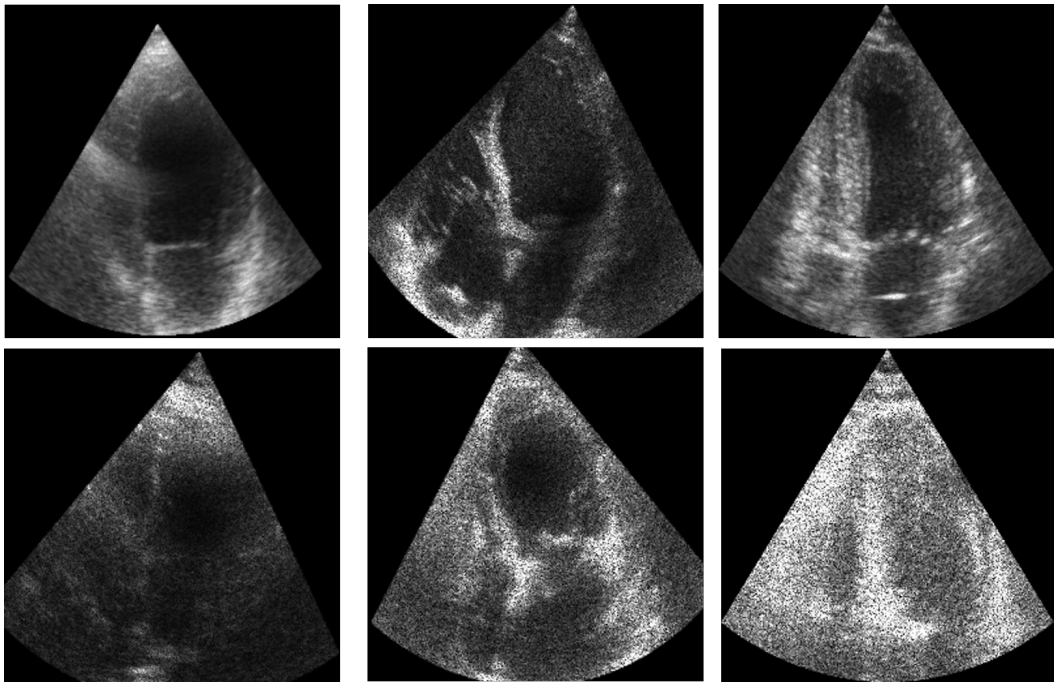


Figure 9. ShapeNet training images example.

Data Augmentation Parameters

To increase the diversity of the training data and improve model generalization, the following augmentation techniques were applied to the original CAMUS dataset, resulting in 4800 training samples:

Geometric Transformations:

- Rotation: Limited to ±15°, to simulate minor probe orientation changes during acquisition.
- Translation: Random shifts of ±10 pixels along both X and Y axes, for natural LV positioning differences across patients.
- Scaling: Random scaling factors between 0.8× and 1.2×, simulating heart size variations.

Ultrasound-Specific Augmentation:

- Acoustic shadows: Simulated shadow artifacts were added through pixelwise multiplication of a spatial Gaussian kernel with selected image regions, following the methodology described in [33].

4.2. Evaluation of the Improved ASM

Capture range tests were performed on the ASM reported here (Section 3.3), and compared to the original ASM reported by Cootes [30]. The mean shape was manually aligned to the left ventricle on an US image, and it was automatically adjusted to the contour of the left ventricle using the original local search of the ASM and our function optimization method. This was repeated for a range of values around the initial manual pose values: $\sigma_0, \theta_0, Tx_0, Ty_0$. The range of values for each pose parameter and the contour segmentation errors for each method are shown in Figure 10 for 30 diastole images.

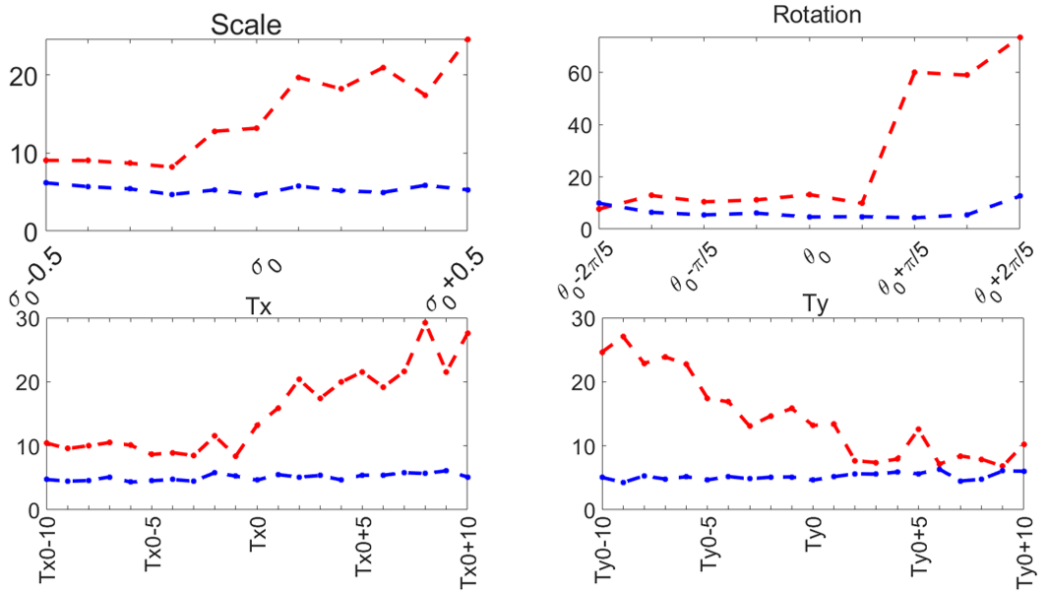


Figure 10. Capture range errors. Haussdorf distance: improved ASM (blue); original ASM [30] (red).

The training parameters for both ASMs were as follows:

- 64 landmarks were used to represent the contour of the LV, as depicted in Figure 2.
- The length of the perpendicular sampled gray profiles for each training example of the PDM was 24 pixels.
- The explained shape variance was 90% for 5 principal components.

These parameters have been demonstrated to accurately represent the LV contour in previous works, as shown in [29,36]. During our capture range experiments, the original ASM reported in [30] failed to converge in several cases, causing run-time errors. Table 1 reports the number of capture range tests conducted and the number of run-time errors that occurred for 30 diastole images from the CAMUS database.

Table 1. Run-time errors for ASM [30] and improved ASM.

	Type of Capture Range Test			
	Rotation	**Tx**	**Ty**	**Scale**
Number of tests performed	270	630	630	330
Percentage of ASM run-time errors	17.8	3.8	2.2	15.5
Percentage of improved ASM run-time errors	0.7	0.0	0.0	0.6

4.3. ShapeNet Contour Prediction Results

In this section, we present the results obtained from the reconstruction of the LV using the parameters predicted by ShapeNet, following the algorithm shown in Figure 8. To evaluate these results, we used the Dice coefficient and Hausdorff distance (HD) expressed in pixels (px), compared against expert annotations. Table 2 shows the Dice and Hausdorff values for the two test datasets: EchoNet and CAMUS. In Figures 11 and 12 are shown six examples of LV reconstruction for systole and diastole, respectively.

Table 2. ShapeNet standalone segmentation errors on CAMUS vs. EchoNet (independent test set).

EchoNet			
Number of evaluated images	Cycle	Mean Dice	Mean HD (px)
207	Systole	0.65 ± 0.11	30.24 ± 9.87
210	Diastole	0.62 ± 0.10	37.07 ± 13.0
CAMUS			
Number of evaluated images	Cycle	Mean Dice	Mean HD (px)
49	Systole	0.76 ± 0.10	19.13 ± 7.91
49	Diastole	0.74 ± 0.10	25.86 ± 14.04

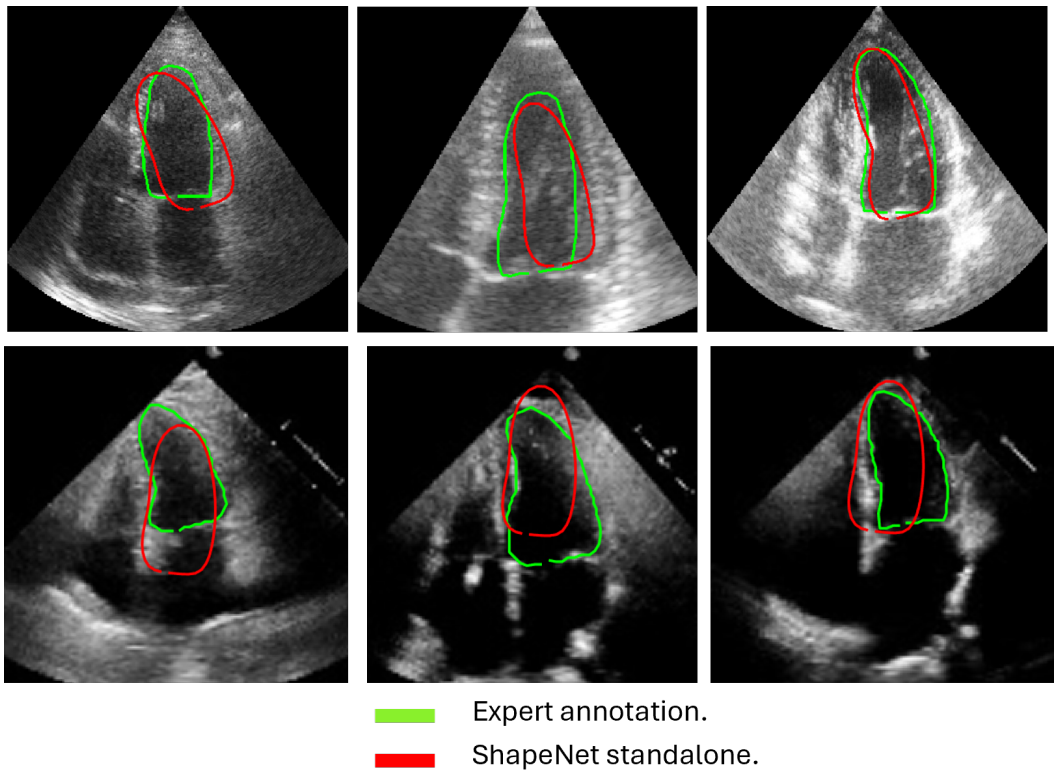


Figure 11. ShapeNet standalone systole segmentation examples for CAMUS (**top**) and EchoNet (**bottom**) images.

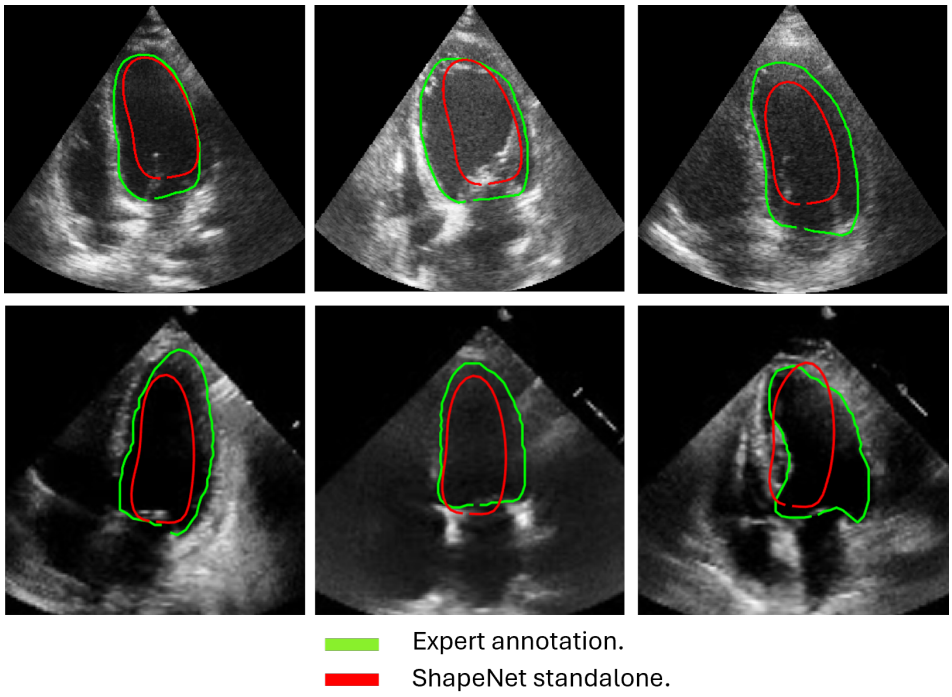


Figure 12. ShapeNet standalone diastole segmentation examples for CAMUS (**top**) and EchoNet (**bottom**) images.

4.4. Improved ASM with ShapeNet Initialization

Following the algorithm depicted in Figure 8, the contour predicted by ShapeNet was used to automatically initialize the ASM described in Section 3.3. The result of combining these two methods can be observed in Figures 13 and 14 for systole and diastole, respectively. In Table 3 is shown the Dice coefficient and Hausdorff distance for each of the test sets (CAMUS and EchoNet), with overall values of 0.83 ± 0.09 for Dice and 7.36 ± 8.02 pixels for Hausdorff distance.

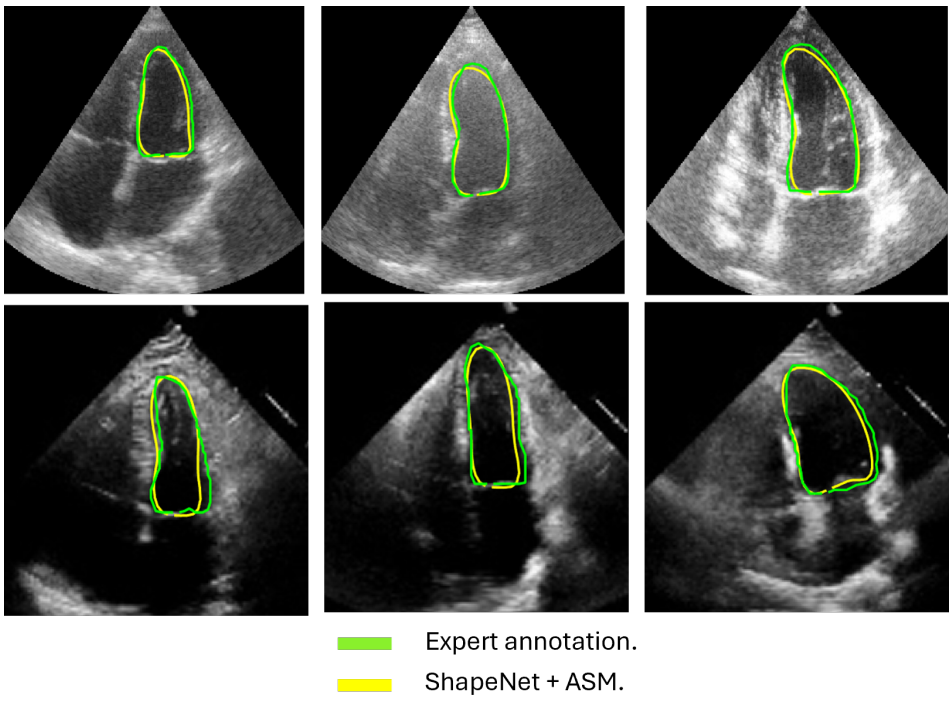


Figure 13. ShapeNet + improved ASM systole segmentation examples for CAMUS (**top**) and EchoNet (**bottom**) images.

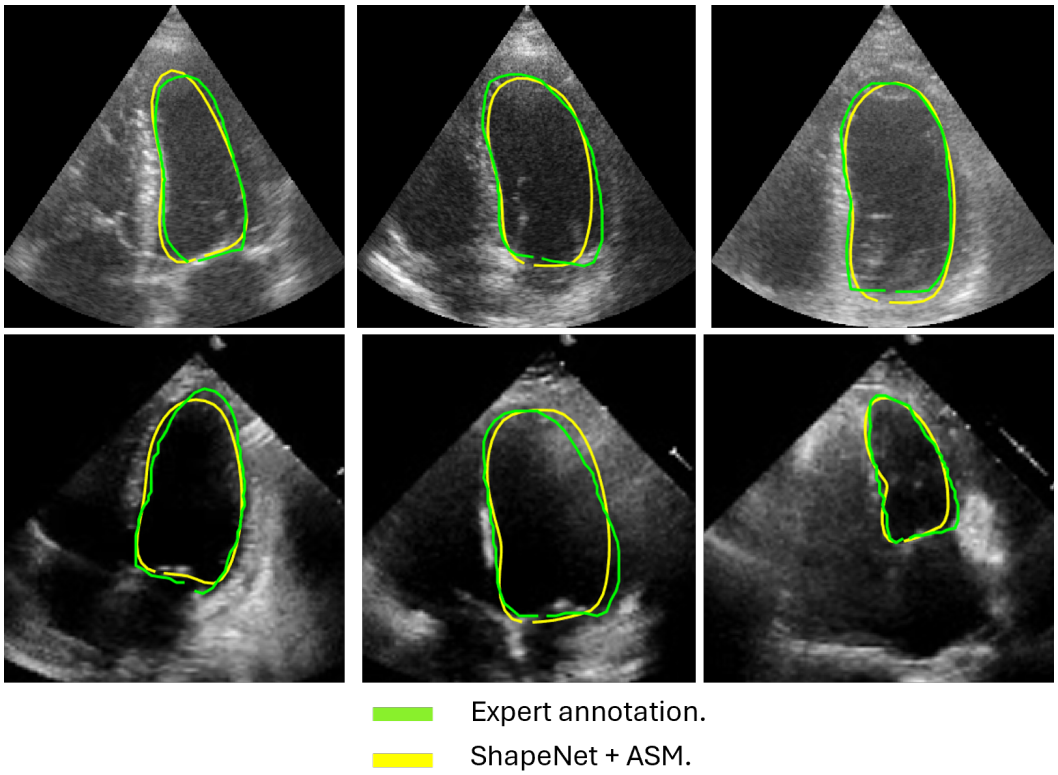


Figure 14. ShapeNet + improved ASM diastole segmentation examples for CAMUS (**top**) and EchoNet (**bottom**) images.

Table 3. ShapeNet + improved ASM segmentation errors on CAMUS vs. EchoNet (independent test set).

EchoNet			
Number of evaluated images	Cycle	Mean Dice	Mean HD (px)
207	Systole	0.81 ± 0.10	9.72 ± 11.66
210	Diastole	0.81 ± 0.13	10.70 ± 13.21
CAMUS			
Number of evaluated images	Cycle	Mean Dice	Mean HD (px)
49	Systole	0.87 ± 0.07	4.26 ± 3.29
49	Diastole	0.87 ± 0.07	4.88 ± 4.65
Overall performance			
Cycle	Overall Dice		Overall HD (px)
Systole	0.84 ± 0.08		6.99 ± 7.47
Diastole	0.84 ± 0.10		7.79 ± 8.93

4.5. ShapeNet + ASM Algorithm vs. Other Methods

This section provides a comparative analysis between the ShapeNet ensemble and two alternative segmentation approaches: (1) shape-based methods (Table 4) and (2) an in-house U-Net (Table 5). In Table 4, we compare the performance of our approach against other shape-based methods reported in [3] and in [26] trained and tested on the same dataset. On the other hand, we developed an in-house U-Net under the same experimental conditions as the ShapeNet. The in-house U-Net was configured with a batch size of 32 images, using the Adam optimizer with a learning rate of 1×10^{-3} for 50 training epochs, and the same dataset as described above was used to train both the U-Net and ShapeNet models. Additionally, the performance of this U-Net was evaluated using CAMUS and the EchoNet dataset, a fully independent dataset not previously seen by either the ShapeNet or the

U-Net model. This setup ensured consistent conditions, for a fair comparison between the U-Net and ShapeNet methods. The performance metrics for each approach are shown in Tables 4 and 5. We also conducted a statistical significance T-test on the Dice and HD values for ShapeNet + ASM and the in-house U-Net using the EchoNet database, as reported in Table 6.

Table 4. Comparison between ShapeNet + improved ASM and other shape-based methods trained and tested with the same dataset.

Method	Database	Systole		Diastole	
		Mean Dice	Mean HD (px)	Mean Dice	Mean HD (px)
ShapeNet + improved ASM	CAMUS	0.875 ± 0.07	4.26 ± 3.29	0.870 ± 0.07	4.88 ± 4.65
BEASM-Fully [3]	CAMUS	0.826 ± 0.09	9.9 ± 5.1	0.879 ± 0.065	9.2 ± 4.9
BEASM- Semi [3]	CAMUS	0.861 ± 0.07	7.7 ± 3.2	0.920 ± 0.03	6.0 ± 2.4
ACNN [26]	CETUS'14	0.873 ± 0.05	7.75 ± 2.65	0.912 ± 0.023	6.96 ± 1.75

Table 5. Performance comparison between ShapeNet + improved ASM and in-house U-Net on CAMUS vs. EchoNet (independent test set).

CAMUS				
Method	**Systole**		**Diastole**	
	Mean Dice	Mean HD (px)	Mean Dice	Mean HD (px)
ShapeNet + improved ASM	0.87 ± 0.07	4.26 ± 3.29	0.87 ± 0.07	4.88 ± 4.65
In-house U-Net	0.90 ± 0.06	9.64 ± 7.36	0.93 ± 0.03	12.44 ± 12.38
EchoNet				
	Mean Dice	Mean HD (px)	Mean Dice	Mean HD (px)
ShapeNet + improved ASM	0.81 ± 0.10	9.72 ± 11.66	0.81 ± 0.13	10.70 ± 13.21
In-house U-Net	0.75 ± 0.09	19.03 ± 8.44	0.78 ± 0.10	24.81 ± 16.19

Table 6. *T*-Test results for ShapeNet + ASM vs. U-Net using EchoNet database.

	Dice Score		HD	
	Systole	Diastole	Systole	Diastole
p value	$p < 1 \times 10^{-08}$	$p < 1^{-02}$	$p < 1 \times 10^{-16}$	$p < 1 \times 10^{-17}$
h value	1.0	1.0	1.0	1.0
Effect size (d)	0.0607	0.0321	9.31	14.132

5. Discussion

This study demonstrated that the proposed ShapeNet + ASM method achieved robust and competitive segmentation performance for left ventricle (LV) contours in echocardiography. Our results, evaluated across the CAMUS and EchoNet datasets, highlighted the method's capability to generate statistically valid and anatomically accurate LV contours. The use of two datasets allowed us to employ the majority of the available images in the CAMUS dataset, to maximize the number of training patterns. We initially reserved a limited number of images to test our algorithm. To further test the accuracy of our proposal we used an independent unseen dataset, EchoNet Dynamic, which comprises a total of 417 images spanning both systole and diastole. As expected, the accuracy of ventricle segmentation was higher for the small test set CAMUS, and slightly lower for the independent test set EchoNet (Tables 2, 3 and 5), which provides more representative values for accuracy than can be expected during clinical use.

The approach implemented in ShapeNet, where the parameters of a statistical shape model of the organ of interest are optimized with a convolutional neural network, provides restrictions that contribute to the explicability of the final segmentation results. All shapes produced were statistically valid organ shapes. As observed in Figures 11 and 12, the results were always smoothed ventricle shapes located closely to the LV in the echocardiography, with a scale and rotation approximate to the expert annotation. However, for deformations, in some cases, the predicted values of the deformation vector b were not as accurate, which was reflected in a higher Hausdorff distance (see Table 2).

Our proposed ASM was used to improve the accuracy of the final segmentation of the LV. This ASM proved to be more accurate than the original ASM reported in [30]. In our capture range tests (see Figure 10), the improved ASM produced smaller mean values for the Hausdorff distance. Additionally, in some cases, when the initialization pose values were far from the ventricle contour, the original ASM failed to converge, causing run-time errors when the LV model grew outside the image. Table 1 shows that, for the improved ASM, the number of run-time errors was exceptionally low compared to the ASM reported in [30]. The improvements in accuracy and robustness of our ASM are most likely due to the use of all the gray level profiles concatenated in a single vector, as well as the objective function, which together provide the means to evaluate the image fitting of a whole ventricle contour, instead of the local adjustment point-by-point performed in the original ASM, and this was reflected in the final segmentation of the LV for the EchoNet dataset, as seen in Table 5 and in Figures 13 and 14.

Table 4 presents a comparison of our results against the methods proposed in [3], specifically the BEASM approaches, the fully automatic BEASM (BEASM-fully) and the semi-automatic BEASM (BEASM-semi), which is manually initialized at three points: two at the LV base and one at the apex, alongside the ACNN method [26], which incorporates shape constraints. ShapeNet + improved ASM, being a fully automatic method, demonstrated competitive performance. Although BEASM-semi and ACNN achieved slightly higher Dice scores, the results in Table 4 indicate that our method achieved the lowest Hausdorff distance for both systole and diastole phases, demonstrating improved characterization of the LV contour over the methods listed in Table 4.

The comparison with our in-house U-Net highlights that our method achieved competitive Dice scores and lower Hausdorff distances under the same training conditions (see Table 5). Specifically, the statistical analysis in Table 6 shows that for Dice scores, our method exhibited statistically significant improvements over the in-house U-Net. In contrast, the Hausdorff distance results reveal large effect sizes of $d = 9.31$ for systole and $d = 14.132$ for diastole, meaning a substantial reduction in spatial errors was achieved by our approach. These large effect sizes support the fact that our method produced a statistically valid LV contour closely aligned with expert annotations, demonstrating the robustness of our approach, even with a limited training set and the challenges of ultrasound imaging, particularly the presence of speckle noise. The added advantage of statistical shape constraints in our method reduced misclassified regions more effectively than traditional semantic classification, as depicted in Figure 15. While our method may not always have reached peak precision, it consistently achieved better segmentation performance on an entirely new dataset (EchoNet), as demonstrated in Table 5.

In Figure 16 are shown histograms of the ShapeNet + improved ASM approach for the Dice coefficient and Hausdorff distance values for systole and diastole for the EchoNet dataset. It can be observed that 75.84% and 77.61% of the cases for systole and diastole, respectively, fell within the range of 0.8 to 1. Similarly, for the Hausdorff distance, the majority of values were concentrated between 0 and 10 pixels, 68.11% and 66.66%, for systole and diastole, respectively. The density concentrated in the previously mentioned

values demonstrate that our algorithm performed well and that the results were consistent, even with a completely new dataset, as was the case with EchoNet, in contrast to the methods shown in Table 4, which were trained and tested on images from the same dataset.

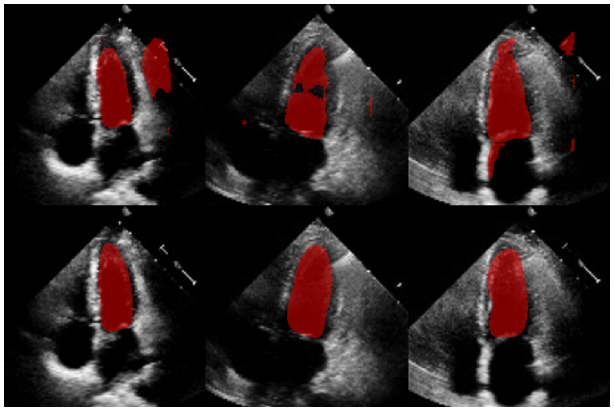


Figure 15. Comparison of segmentation masks between in-house U-Net (**top**) and ShapeNet + improved ASM (**bottom**) on EchoNet database.

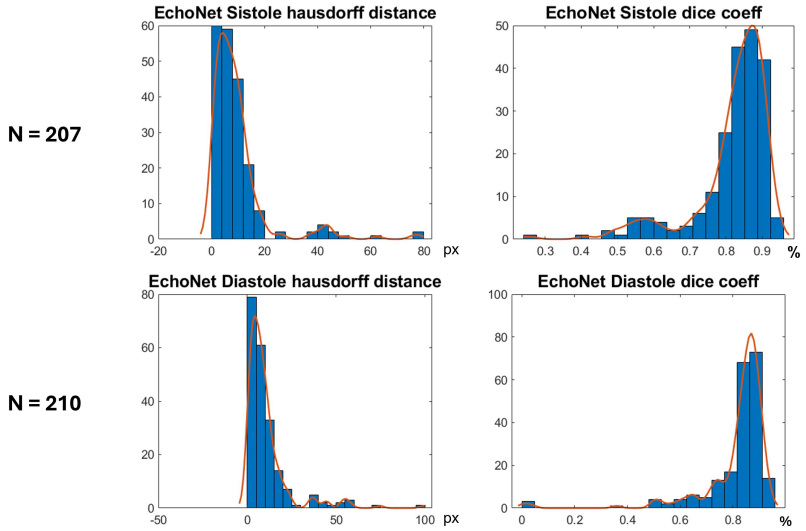


Figure 16. Segmentation errors (Dice score and HD) for EchoNet dataset using ShapeNet + improved ASM approach.

Despite the strengths of the proposed method, some limitations should be acknowledged. The CAMUS dataset, although it was augmented to 4800 images, remains relatively small for deep learning applications, and this may have affected the model's generalization. In addition, the ShapeNet + improved ASM automatic initialization introduces a dependency that may reduce accuracy if there are significant deviations in initial conditions, such as patient positioning or heart orientation. The effectiveness of the improved ASM remains dependent on ShapeNet's initialization quality. While ShapeNet's ensemble architecture and the PDM's shape constraints help mitigate this dependency, extreme imaging artifacts or anatomical anomalies may still lead to suboptimal ASM convergence. This limitation was evidenced in the EchoNet HD variance (10.21 px), where acoustic shadowing occasionally compromised initialization (Figure 12, bottom right). Nevertheless, our experimental results demonstrated that this combined approach remains clinically viable. ShapeNet's parameter regression achieved Dice scores superior to 0.65, even in challenging cases (Table 2), providing sufficiently accurate initialization for ASM refinement. Furthermore, the ASM's global objective function effectively corrected local errors when initialization was imperfect, as demonstrated by its superior performance compared to the original ASM (Figure 10).

On the other hand, the model's reliance on a convolutional neural network ensemble and active shape models demands substantial computational resources for both training and inference. Each network in the ShapeNet ensemble was trained for a maximum of 50 epochs with early stopping (validation patience = 40 iterations). With our hardware configuration (NVIDIA Tesla K40c/T4), the average training time per epoch was approximately 20 min for each specialized CNN (rotation, translation, scale, and shape parameters). For comparison, our in-house U-Net implementation reached early stopping at 38 epochs with longer epoch durations of 41 min on average. Although we tried to make it a lightweight model, this requirement may limit accessibility and real-time application in clinical settings without the availability of modern GPUs. Regarding this issue, we conducted performance tests on a current gaming computer with the following specifications: 24 GB of RAM, an 8th generation Core i7 processor, and an NVIDIA GeForce GTX 1060 with Max Q with 6 GB, achieving offline segmentation of 207 LV images in an average of 18 min (5.27 s per image).

Concerning the morphology of the ventricle, although ShapeNet generally performs well, the segmentation accuracy during the systolic phase may be compromised due to increased left ventricular deformation during contraction, along with greater deformation observed in the diastolic phase, as indicated by higher Hausdorff distances in some cases. Our method showed enhanced robustness with limited training data, offering a statistically grounded alternative to traditional segmentation models.

Future work will involve exploring the incorporation of additional datasets to further enhance the model's inference capabilities. Additionally, we plan to investigate techniques such as model pruning and knowledge distillation to reduce computational demands, while preserving the performance advantages of the ensemble.

6. Conclusions

In this paper, we proposed a new scheme for automatic LV segmentation in echocardiography images. The proposal consists of two stages: ShapeNet, which is an ensemble of CNNs to predict pose and shape parameters; and an improved ASM, which is initialized with the parameters estimated by the ensemble of neural networks, in order to fine-tune the LV contours for improved segmentation. Our study demonstrated that integrating ShapeNet with an improved ASM enhanced LV segmentation accuracy and effectively prevented blob artifacts commonly found in semantic segmentation. Our algorithm was tested on two different datasets, CAMUS and EchoNet, providing an overall Dice coefficient of 0.83 and a Hausdorff distance of 7.36 pixels for both systole and diastole.

A major strength of our approach is its ability to automatically generate statistically valid shapes, offering a new perspective on the utilization of convolutional neural networks (CNNs) in medical imaging. When combined with the improved ASM, which outperformed traditional ASM techniques, our method provided competitive and accurate fitting of the LV contour compared to existing shape-based methods, as evidenced by higher Dice coefficients and lower Hausdorff distances in ultrasound images. Notably, we demonstrated that the ShapeNet + ASM approach was more robust with a limited training set than traditional semantic segmentation methods such as U-Net under the same conditions. Despite these strengths, limitations should be considered, including the need for substantial computational resources, the limited availability of training images, and the complexity of ultrasound images, due to speckle noise and heart morphology variations in both systolic and diastolic phases.

The ShapeNet and improved ASM methodologies presented here offer a promising alternative to semantic segmentation CNNs in medical image analysis. This approach, based on statistically valid shape adjustment, holds potential for broad applications in automated medical image analysis and clinical decision-making.

Author Contributions: Conceptualization, E.G.G., F.T.-R. and F.A.C.; methodology, E.G.G., F.T.-R. and F.A.C.; software, E.G.G.; validation, E.G.G. and J.P.-G.; formal analysis, E.G.G.; investigation, E.G.G., F.T.-R. and J.P.-G.; resources, F.T.-R.; data curation, E.G.G. and F.T.-R.; writing—original draft preparation, E.G.G.; writing—review and editing, F.T.-R., J.P.-G. and F.A.C.; visualization, E.G.G.; supervision, F.T.-R. and F.A.C.; project administration, E.G.G. and F.A.C.; funding acquisition, J.P.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by UNAM under project: UNAM-PAPIIT Program IA100924.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the use of two public image databases.

Informed Consent Statement: Patient consent was waived due to the use of two public image databases.

Data Availability Statement: CAMUS database can be found at https://www.creatis.insa-lyon.fr/Challenge/camus/ (accessed on 10 January 2021). In addition, the EchoNet Dynamic database can be downloaded from https://echonet.github.io/dynamic/ (accessed on 22 June 2023).

Acknowledgments: Eduardo Galicia acknowledges the support of the doctoral fellowship granted by CONAHCYT (CVU 593961). We also extend our acknowledgment to Daniel Colin for his support in the validation process and to Adrian Duran for his technical assistance with the LUCAR server.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LV	Left Ventricle
CNN	Convolutional Neural Network
PDM	Point Distribution Model
ASM	Active Shape Model
PCA	Principal Component Analysis
HD	Hausdorff Distance
SSM	Statistical Shape Model
px	Pixel

References

1. Berman, M.N.; Tupper, C.B.A. *Physiology, Left Ventricular Function*; StatPearls Publishing: Treasure Island, FL, USA, 2022.
2. Conti, C.R. Assessing ventricular function. *Clin. Cardiol.* **2004**, *27*, 1–2. [CrossRef] [PubMed]
3. Leclerc, S.; Smistad, E.; Pedrosa, J.; Østvik, A.; Cervenansky, F.; Espinosa, F.; Espeland, T.; Berg, E.A.R.; Jodoin, P.M.; Grenier, T.; et al. Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography. *IEEE Trans. Med. Imaging* **2019**, *38*, 2198–2210. [CrossRef] [PubMed]
4. Shoaib, M.A.; Chuah, J.H.; Ali, R.; Hasikin, K.; Khalil, A.; Hum, Y.C.; Tee, Y.K.; Dhanalakshmi, S.; Lai, K.W. An Overview of Deep Learning Methods for Left Ventricle Segmentation. *Comput. Intell. Neurosci.* **2023**, *2023*, 4208231. [CrossRef] [PubMed]
5. Savioli, N.; Vieira, M.S.; Lamata, P.; Montana, G. Automated segmentation on the entire cardiac cycle using a deep learning work-flow. *arXiv* **2018**, arXiv:1809.01015.
6. Abdeltawab, H.; Khalifa, F.; Taher, F.; Alghamdi, N.S.; Ghazal, M.; Beache, G.; Mohamed, T.; Keynton, R.; El-Baz, A. A deep learning-based approach for automatic segmentation and quantification of the left ventricle from cardiac cine MR images. *Comput. Med. Imaging Graph.* **2020**, *81*, 101717. [CrossRef]
7. Chen, X.; Wang, X.; Zhang, K.; Fung, K.M.; Thai, T.C.; Moore, K.; Mannel, R.S.; Liu, H.; Zheng, B.; Qiu, Y. Recent advances and clinical applications of deep learning in medical image analysis. *Med. Image Anal.* **2022**, *79*, 102444. [CrossRef]
8. Zou, X.; Wang, Q.; Luo, T. A novel approach for left ventricle segmentation in tagged MRI. *Comput. Electr. Eng.* **2021**, *95*, 107416. [CrossRef]
9. Wech, T.; Ankenbrand, M.; Bley, T.; Heidenreich, J. A data-driven semantic segmentation model for direct cardiac functional analysis based on undersampled radial MR cine series. *Magn. Reson. Med.* **2021**, *87*, 972–983. [CrossRef]

10. Veni, G.; Moradi, M.; Bulu, H.; Narayan, G.; Syeda-Mahmood, T. Echocardiography segmentation based on a shape-guided deformable model driven by a fully convolutional network prior. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 898–902. [CrossRef]
11. Ansari, M.Y.; Yang, Y.; Meher, P.K.; Dakua, S.P. Dense-PSP-UNet: A neural network for fast inference liver ultrasound segmentation. *Comput. Biol. Med.* **2023**, *153*, 106478. [CrossRef]
12. Kang, S.; Kim, S.J.; Ahn, H.G.; Cha, K.C.; Yang, S. Left ventricle segmentation in transesophageal echocardiography images using a deep neural network. *PLoS ONE* **2023**, *18*, e0280485. [CrossRef]
13. Zhao, Y.; Liao, K.; Zheng, Y.; Zhou, X.; Guo, X. Boundary attention with multi-task consistency constraints for semi-supervised 2D echocardiography segmentation. *Comput. Biol. Med.* **2024**, *171*, 108100. [CrossRef] [PubMed]
14. Shi, S.; Alimu, P.; Mahemut, P. The Study of Echocardiography of Left Ventricle Segmentation Combining Transformer and Convolutional Neural Networks. *Int. Heart J.* **2024**, *65*, 889–897. [CrossRef]
15. Paragios, N.; Deriche, R. Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation. *Int. J. Comput. Vis.* **2002**, *46*, 223–247. [CrossRef]
16. Paragios, N. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Trans. Med. Imaging* **2003**, *22*, 773–776. [CrossRef] [PubMed]
17. Nascimento, J.C.; Marques, J.S. Robust Shape Tracking With Multiple Models in Ultrasound Images. *Trans. Img. Proc.* **2008**, *17*, 392–406. [CrossRef] [PubMed]
18. Ali, Y.; Beheshti, S.; Janabi-Sharifi, F. Echocardiogram segmentation using active shape model and mean squared eigenvalue error. *Biomed. Signal Process. Control* **2021**, *69*, 102807. [CrossRef]
19. Zagrodsky, V.; Walimbe, V.; Castro-Pareja, C.; Qin, J.; Song, J.M.; Shekhar, R. Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models. *IEEE Trans. Med. Imaging* **2005**, *24*, 1089–1099. [CrossRef]
20. Georgescu, B.; Zhou, X.; Comaniciu, D.; Gupta, A. Database-guided segmentation of anatomical structures with complex appearance. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 429–436. [CrossRef]
21. Mitchell, S.; Lelieveldt, B.; van der Geest, R.; Bosch, H.; Reiber, J.; Sonka, M. Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images. *IEEE Trans. Med. Imaging* **2001**, *20*, 415–423. [CrossRef]
22. Bernard, O.; Bosch, J.G.; Heyde, B.; Alessandrini, M.; Barbosa, D.; Camarasu-Pop, S.; Cervenansky, F.; Valette, S.; Mirea, O.; Bernier, M.; et al. Standardized Evaluation System for Left Ventricular Segmentation Algorithms in 3D Echocardiography. *IEEE Trans. Med. Imaging* **2016**, *35*, 967–977. [CrossRef]
23. Li, Y.; Lu, W.; Monkam, P.; Zhu, Z.; Wu, W.; Liu, M. LVSnake: Accurate and robust left ventricle contour localization for myocardial infarction detection. *Biomed. Signal Process. Control* **2023**, *85*, 105076. [CrossRef]
24. Peng, S.; Jiang, W.; Pi, H.; Bao, H.; Zhou, X. Deep Snake for Real-Time Instance Segmentation. *arXiv* **2020**, arXiv:2001.01629,
25. Hsu, W.Y. Automatic Left Ventricle Recognition, Segmentation and Tracking in Cardiac Ultrasound Image Sequences. *IEEE Access* **2019**, *7*, 140524–140533. [CrossRef]
26. Oktay, O.; Ferrante, E.; Kamnitsas, K.; Heinrich, M.; Bai, W.; Caballero, J.; Cook, S.; De Marvao, A.; Dawes, T.; O'Regan, D.; et al. Anatomically Constrained Neural Networks (ACNNs): Application to Cardiac Image Enhancement and Segmentation. *IEEE Trans. Med. Imaging* **2018**, *37*, 384–395.
27. Gaggion, N.; Mansilla, L.; Mosquera, C.; Milone, D.H.; Ferrante, E. Improving Anatomical Plausibility in Medical Image Segmentation via Hybrid Graph Neural Networks: Applications to Chest X-Ray Analysis. *IEEE Trans. Med. Imaging* **2023**, *42*, 546–556. [CrossRef]
28. Ribeiro, M.A.O.; Nunes, F.L.S. Left ventricle segmentation combining deep learning and deformable models with anatomical constraints. *J. Biomed. Inform.* **2023**, *142*, 104366. [CrossRef]
29. Galicia-Gómez, E.; Torres-Robles, F.; Pérez, J.; Escalante-Ramírez, B.; Arámbula Cosío, F. A U-Net with Statistical Shape Restrictions Applied to the Segmentation of the Left Ventricle in Echocardiographic Images. *Rev. Mex. Ing. Biomed.* **2024**, *44*, 140–151. [CrossRef]
30. Cootes, T.; Taylor, C.; Cooper, D.; Graham, J. Active Shape Models-Their Training and Application. *Comput. Vis. Image Underst.* **1995**, *61*, 38–59. [CrossRef]
31. Li, S.; Zhao, P.; Zhang, H.; Sun, X.; Wu, H.; Jiao, D.; Wang, W.; Liu, C.; Fang, Z.; Xue, J.; et al. Surge Phenomenon in Optimal Learning Rate and Batch Size Scaling. *arXiv* **2024**, arXiv:2405.14578.
32. Arámbula Cosío, F.; Flores, J.M.; Castañeda, M.P. Use of simplex search in active shape models for improved boundary segmentation. *Pattern Recognit. Lett.* **2010**, *31*, 806–817. [CrossRef]
33. Romero-Pacheco, A.; Perez-Gonzalez, J.; Hevia-Montiel, N. Estimating Echocardiographic Myocardial Strain of Left Ventricle with Deep Learning. In Proceedings of the 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Glasgow, Scotland, UK, 11–15 July 2022; pp. 3891–3894. [CrossRef]

34. Cervantes-Guzmán, A.; McPherson, K.; Olveres, J.; Moreno-García, C.F.; Robles, F.T.; Elyan, E.; Escalante-Ramírez, B. Robust cardiac segmentation corrected with heuristics. *PLoS ONE* **2023**, *18*, e0293560. [CrossRef]
35. Ouyang, D.; He, B.; Ghorbani, A.; Lungren, M.P.; Ashley, E.A.; Liang, D.H.; Zou, J.Y. EchoNet-Dynamic: A Large New Cardiac Motion Video Data Resource for Medical Machine Learning. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
36. Gomez, E.G.; Robles, F.T.; Ramirez, B.E.; Olveres, J.; Cosío, F.A. Full multi resolution active shape model for left ventricle segmentation. In Proceedings of the 17th International Symposium on Medical Information Processing and Analysis, Campinas, Brazil, 17–19 November 2021; Rittner, L., Romero, E., Tavares Costa, E., Lepore, N., Brieva, J., Linguraru, M.G., Eds.; International Society for Optics and Photonics; SPIE: Philadelphia, PA, USA, 2021; Volume 12088, p. 120881C. [CrossRef]