National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Enhancement of the Domain Generalization of Vision Transformers through Advanced Data Augmentation Techniques

Diploma Thesis

of

EVANGELOS G. FROUDAKIS

Supervisor: Athanasios Voulodimos
Assistant Professor, ECE NTUA

Athens, July 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

Enhancement of the Domain Generalization of Vision Transformers through Advanced Data Augmentation Techniques

Diploma Thesis

of

EVANGELOS G. FROUDAKIS

Supervisor: Athanasios Voulodimos
Assistant Professor, ECE NTUA

Approved by the examination committee on 9th July 2024.

(Signature)	*(Signature)*	*(Signature)*
Athanasios Voulodimos	Georgios Stamou	Andreas-Georgios Stafylopatis
Assistant Professor, ECE NTUA	Professor, ECE NTUA	Professor, ECE NTUA

Athens, July 2024

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....
Evangelos G. Froudakis

Graduate of Electrical
and Computer
Engineering, National
Technical University
of Athens

9th July 2024

Περίληψη

Η γενίκευση πεδίου (Domain Generalization) αποτελεί μια σημαντική πρόκληση στον τομέα της μηχανικής μάθησης, όπου η απόδοση των μοντέλων βαθιάς μάθησης μπορεί να υποβαθμιστεί σημαντικά λόγω μετατοπίσεων στην κατανομή των δεδομένων, που προκαλούνται από διαφορές στις συνθήκες εκπαίδευσης και εφαρμογής. Τα μοντέλα βαθιάς μάθησης συνήθως υποθέτουν ότι τα δεδομένα εκπαίδευσης και τα δεδομένα που χρησιμοποιούνται κατά την εφαρμογή προέρχονται από την ίδια κατανομή. Ωστόσο, στην πραγματικότητα, αυτή η υπόθεση σπάνια ισχύει. Οι διαφορές στην κατανομή των δεδομένων, γνωστές ως μεταβάσεις πεδίου, μπορούν να προκληθούν από διάφορους παράγοντες, όπως αλλαγές στο περιβάλλον, στον εξοπλισμό ή στις διαδικασίες συλλογής δεδομένων. Αυτές οι μεταβάσεις μπορούν να οδηγήσουν σε σημαντική μείωση της απόδοσης των μοντέλων όταν εφαρμόζονται σε δεδομένα που δεν έχουν δει κατά την εκπαίδευση.

Η γενίκευση πεδίου είναι ζωτικής σημασίας για την ανάπτυξη ανθεκτικών συστημάτων μηχανικής μάθησης που μπορούν να εφαρμοστούν σε διάφορα πραγματικά σενάρια. Οι προκλήσεις που σχετίζονται με τη γενίκευση πεδίου εμφανίζονται σε πολλούς τομείς, όπως η αυτόνομη οδήγηση, όπου τα μοντέλα πρέπει να είναι ανθεκτικά σε μεταβαλλόμενες καιρικές συνθήκες και συνθήκες φωτισμού, και στην αναγνώριση εικόνας, όπου οι μεταβολές στο περιβάλλον ή τη γωνία λήψης μπορούν να επηρεάσουν την απόδοση του μοντέλου. Η αντιμετώπιση αυτών των προκλήσεων απαιτεί καινοτόμες τεχνικές που μπορούν να βελτιώσουν την ικανότητα των μοντέλων να γενικεύουν από την εκπαίδευση, σε πραγματικές εφαρμογές.

Η γενίκευση πεδίου είναι επίσης κρίσιμη για εφαρμογές σε τομείς όπου τα δεδομένα εκπαίδευσης μπορεί να είναι περιορισμένα ή να μην αντιπροσωπεύουν πλήρως την ποικιλία των καταστάσεων που θα αντιμετωπίσει το μοντέλο κατά την εφαρμογή. Για παράδειγμα, στη μετάφραση, τα μοντέλα πρέπει να αντιμετωπίζουν διαφορετικά στιλ γραφής και φράσεις που δεν εμφανίζονται στα δεδομένα εκπαίδευσης. Στην ιατρική απεικόνιση, όπου η συλλογή δεδομένων μπορεί να είναι δαπανηρή ή να υπόκειται σε νομικούς περιορισμούς, η ικανότητα ενός μοντέλου να γενικεύει από περιορισμένα δεδομένα εκπαίδευσης σε νέα δεδομένα είναι ιδιαίτερα σημαντική.

Η ανάγκη για βελτίωση της γενίκευσης των μοντέλων σε διάφορους τομείς είναι σημαντική, και η σύγχρονη επιστήμη έχει βρει τρόπους ώστε τα συστήματα τεχνητής νοημοσύνης να γενικεύουν σε άγνωστα δεδομένα χωρίς να τα έχουν δει κατά την εκπαίδευση. Η επαύξηση δεδομένων είναι μία από αυτές τις μεθόδους που έχει οδηγήσει σε σημαντικές βελτιώσεις στις ικανότητες γενίκευσης των μοντέλων βαθιάς μάθησης. Η επαύξηση δεδομένων στοχεύει στην παραγωγή νέων δειγμάτων εκπαίδευσης μέσω τροποποίησης των υπαρχόντων δεδομένων με διάφορους τρόπους, όπως μέσω περιστροφών, μεταφράσεων και ρυθμίσεων χρώματος, δημιουργώντας παραλλαγές που μπορεί να συναντήσει το μοντέλο σε πραγματικά σενάρια.

Η παρούσα διατριβή διερευνά την ενίσχυση της γενίκευσης πεδίου στην κατάτμηση εικόνων ιατρικής απεικόνισης μέσω προηγμένων τεχνικών επαύξησης δεδομένων. Αυτή η διπλωματική εστιάζει στην αξιολόγηση του αντίκτυπου της επαύξησης δεδομένων τόσο σε επίπεδο εισόδου όσο και σε επίπεδο χαρακτηριστικών χρησιμοποιώντας μεθόδους όπως η επαύξηση βασισμένη σε στυλ. Ενσωματώνοντας αυτές τις στρατηγικές επαύξησης, στόχος είναι να βελτιωθεί η ικανότητα των μοντέλων να χειρίζονται άγνωστες παραλλαγές στις ιατρικές εικόνες, αυξάνοντας έτσι την ανθεκτικότητα και την αξιοπιστία τους σε πραγματικές εφαρμογές.

Στις πειραματικές διαδικασίες, χρησιμοποιήθηκε ένα μοντέλο vision transformer, του οποίου η επίδοση βελτιώθηκε με σύνολα δεδομένων που επαυξήθηκαν μέσω συνδυασμού επαυξήσεων βασισμένων σε στυλ όπως το MaxStyle και άλλων μεθόδων επαύξησης εισόδου, όπως το AugMix. Αυτές οι τεχνικές ενισχύουν την ποικιλία των δεδομένων εκπαίδευσης, επιτρέποντας στο μοντέλο να μάθει ανθεκτικά χαρακτηριστικά που είναι λιγότερο ευαίσθητα σε διάφορους τύπους μετατοπίσεων δεδομένων. Η αξιολόγηση πραγματοποιήθηκε σε σύνολα δεδομένων MRI προστάτη ως εντός πεδίου δεδομένα και έξι επιπλέον σύνολα δεδομένων ως εκτός πεδίου δεδομένα.

Τα αποτελέσματα έδειξαν ότι τα μοντέλα που εκπαιδεύτηκαν με επαυξημένα δεδομένα παρουσίασαν σημαντικά βελτιωμένη ανθεκτικότητα και γενίκευση σε δείγματα εκτός πεδίου. Ο συνδυασμός επαυξήσεων βασισμένων σε στυλ και άλλων μεθόδων επαύξησης οδήγησε σε αξιοσημείωτη αύξηση της γενικευσιμότητας. Αυτό υποδηλώνει ότι ο συνδυασμός και η ενσωμάτωση σύνθετων τεχνικών επαύξησης δεδομένων μπορεί να ενισχύσει σημαντικά την ανθεκτικότητα των μοντέλων κατάτμησης εικόνων ιατρικής απεικόνισης, καθιστώντας τα πιο αξιόπιστα για κλινικές εφαρμογές.

Συγκεκριμένα, η μέθοδος MaxStyle που χρησιμοποιήθηκε για την επαύξηση χαρακτηριστικών, ενσωματώνει επαύξηση βασισμένη στο στυλ, με ανταγωνιστική εκμάθηση, για τη δημιουργία επαυξημένων εικόνων που δυσκολεύουν το δίκτυο κατάτμησης, βελτιώνοντας έτσι την ανθεκτικότητα του μοντέλου έναντι άγνωστων αλλοιώσεων [1]. Επιπλέον, η μέθοδος AugMix που χρησιμοποιήθηκε για την επαύξηση εισόδου περιλαμβάνει τον συνδυασμό απλών λειτουργιών επαύξησης και έναν μηχανισμό συνέπειας απωλειών [2]. Αυτή η προσέγγιση εξασφαλίζει ότι το μοντέλο μαθαίνει χαρακτηριστικά που είναι ανθεκτικά σε διάφορους τύπους διαφθοράς. Ο συνδυασμός αυτής της μεθόδου με τη χρήση ενός vision transformer όπως το ΣεγΦορμερ, ο οποίος έχει σχεδιαστεί για να επωφελείται από τις μακρινές εξαρτήσεις και τα συμφραζόμενα, συνέβαλε στην επίτευξη βέλτιστων αποτελεσμάτων [3].

Η μελέτη αυτή επισημένει τη σημασία των προηγμένων μεθόδων επαύξησης δεδομένων στην εκπαίδευση ανθεκτικών και γενικεύσιμων μοντέλων κατάτμησης εικόνων ιατρικής απεικόνισης. Η ενσωμάτωση επαύξησης βασισμένης σε στυλ στη διαδικασία εκπαίδευσης έδειξε σημαντικές βελτιώσεις στην απόδοση των μοντέλων έναντι μεταβάσεων πεδίου, ανοίγοντας τον δρόμο για πιο αξιόπιστες και ακριβείς λύσεις ιατρικής απεικόνισης.

Λέξεις Κλειδιά

Νευρωνικά Δίκτυα, Βαθιά Μάθηση, Κατάτμηση Εικόνας, Γενίκευση Πεδίου, Μεταφορά Στυλ, Επαύξηση Δεδομένων

Abstract

Domain generalization is a critical challenge in medical imaging, where the performance of deep learning models can significantly degrade due to domain shifts—variations in data distribution caused by differences in imaging protocols, equipment, or patient populations. This issue is particularly problematic in medical image segmentation, where models trained on a specific dataset often fail to generalize to new, unseen data, limiting their practical applicability in clinical settings. Addressing this problem requires innovative techniques to enhance model robustness and generalizability.

This thesis investigates the enhancement of domain generalization in medical image segmentation through advanced data augmentation techniques. This study focuses on evaluating the impact of data augmentation at both the input and feature levels using methods such as style-based augmentation. By incorporating these augmentation strategies, the goal is to improve the ability of models to handle unseen variations in medical images, thereby increasing their robustness and reliability in real-world applications.

In the experiments, a vision transformer model was fine-tuned with datasets augmented through a combination of style-based and other input-level augmentation methods. These techniques enhance the diversity of training data, allowing the model to learn robust features that are less sensitive to various types of data shifts. The evaluation was conducted on prostate MRI datasets as the in-domain data and six additional datasets as the out-of-distribution domains.

The results demonstrated that models trained with augmented data exhibited significantly improved robustness and generalization to OOD samples. The combination of style-based and other augmentation methods led to a notable increase in generalizability. This suggests that integrating complex data augmentation techniques can significantly enhance the robustness of medical image segmentation models, making them more reliable for clinical applications.

Keywords

Neural Networks, Deep Learning, Image Segmentation, Domain Generalization, Style Transfer, Data Augmentation

to my parents

Acknowledgements

I would first like to thank Professor Athanasios Voulodimos, Professor Giorgos Stamou, and Professor Paraskevi Tzouveli for supervising this thesis and for giving me the opportunity to carry it out in the Artificial Intelligence and Learning Systems Laboratory. I also extend my special thanks to Nikolaos Spanos for his guidance and the excellent collaboration we had. I would like to thank my parents for their guidance and moral support throughout all these years. Finally, I would like to thank my friends, as I would not be where I am today without their support.

Athens, July 2024

Evangelos G. Froudakis

Contents

Αβστρασϛτ 1

Abstract 3

Acknowledgements 7

1 Introduction 17
- 1.1 Objective of the Thesis . . . 18
- 1.2 Thesis Organization . . . 18

I Theoretical Part 19

2 Theoretical Background 21
- 2.1 Deep Neural Networks . . . 21
  - 2.1.1 Neural Networks . . . 21
  - 2.1.2 Convolutional Neural Networks . . . 22
  - 2.1.3 Fully Convolutional Neural Networks . . . 25
  - 2.1.4 Encoder Decoder Models . . . 27
  - 2.1.5 Vision Transformers . . . 28
- 2.2 Image segmentation . . . 30
  - 2.2.1 Image Segmentation Types . . . 30
- 2.3 Style Transfer . . . 30
- 2.4 Out of distribution Problem . . . 32
- 2.5 Domain Generalization . . . 34

3 Related work on Domain Generalization on images 37
- 3.1 Ralated work . . . 37
  - 3.1.1 Adversarial Training . . . 37
  - 3.1.2 Style Transfer . . . 38
  - 3.1.3 Data Augmentation . . . 39
  - 3.1.4 Vision Transformers for Medical Imaging . . . 39

II Practical Part 41

4 Datasets 43
- 4.1 Prostate datasets . . . 43

4.1.1 NCI-ISBI 2013 . . . 43
4.1.2 Initiative for Collaborative Computer Vision Benchmarking . . . 43
4.1.3 Prostate MR Image Segmentation 2012 . . . 44
4.1.4 Medical Decathlon . . . 44
4.2 Data preprocessing . . . 44
4.2.1 The N3 Algorithm . . . 44
4.2.2 The N4 Algorithm . . . 45
4.2.3 Rescaling . . . 47
4.2.4 Resizing . . . 47
4.2.5 Photometric and Geometric Transformations . . . 47

5 Implementation **49**
5.1 Data Augmentation . . . 49
5.1.1 Input level augmentation methods . . . 49
5.1.2 Feature level augmentation methods . . . 51
5.2 Experiment Set-up . . . 53

6 Presentation and Analysis of the Results **55**
6.1 Avaluation Metrics . . . 55
6.2 Experiment Results . . . 55

III Epilogue **57**

7 Epilogue **59**
7.1 Conclusions . . . 59
7.2 Future Work . . . 59

Appendices **61**

A Experiments **63**
A.1 Experiment and Resault Presentation . . . 63
A.1.1 Test Setup Analysis . . . 63
A.1.2 Test Hyper-parameters Analysis . . . 66
A.1.3 Test Resaults Pesentation . . . 68

Bibliography **79**

List of Abbreviations **81**

List of Figures

2.1 An example of CNN architecture[4] 22

2.2 An example of a convolution operation[4] 23

2.3 An example of a Pooling Layer operation[4] 24

2.4 An example of deep Convolutional Neural Networks' architectures is shown. At the bottom, the architecture of the VGG-19 model is depicted [5], in the middle is presented a plain CNN with a 32-parameter layer, and at the top, the architecture of the ResNet model is illustrated [6]. 25

2.5 An example of the basic architecture of a Fully Connected Network (FCN) [7] 26

2.6 The original architecture of U-Net [8]. Each blue rectangle represents a multichannel feature map. The white rectangles represent copied feature maps. Finally, the arrows show the different kinds of operations. 26

2.7 An example of the basic architecture of an Encoder Decoder Network [9] . 28

2.8 The architecture of ViT from the original paper [10] is shown. On the left, an overview of the architecture and its different layers is depicted. On the right, there is a close-up of the Transformer Encoder part and its different layers. 29

2.9 Examples of Image Segmentations [11]. Specifically, the Semantic Segmentation of the input images are depicted 31

2.10 Examples of Instance Segmentations from the original paper that proposed Mask R-CNN [12]. 32

2.11 In this Figure from the original paper by Gatys et al. there are images with the content of a photograph of Neckarfront in Tubingen, Germany **A** recreated influenced by the style of five well known paintings [13]. The paintings used that provided the style are shown in the bottom left corner of each generated image and are: **B** The Shipwreck of the Minotaur by J.M.W. Turner, 1805, **C** The Starry Night by Vincent van Gogh, 1889, **D** Der Schrei by Edvard Munch, 1893, **E** Femme nue assise by Pablo Picasso, 1910, and **E** Composition VII by Wassily Kandinsky, 1913. 33

3.1 An example of the generation of an adversarial example from the original paper [14]. The authors add to the original image x an imperceptible vector whose elements are equal to the sign of the elements of the gradient of the cost function. This way the classification of the input image changes to be faulty 38

3.2 In this 2-D t-SNE visualization of style statistics from the original paper that proposed MixStyle [15] it is shown that the four different domains (Cartoon, Sketch, Art Painting, and Photo) are clearly separated. 38
3.3 In this figure, two examples of augmented images are shown. The left image is created using Mixup [16], while the right is created using AugMix [2]. The fisrt one is a combination of two different images, while the second is a combination of deviations of one image. 40

4.1 This figure from the original paper visualises the differences between the algorithms N3 and N4 [17]. The first column shows the original MRIs of the postmortem hippocampuses from three different persons. The second and third columns show the corrected samples outputted from the N3MNI algorithm and the detected bias filter respectively. The fourth and fifth columns show the same elements as the second and third columns but outputted from the N4ITK algorithm. 46

5.1 In this figure from the original paper, is presented a visualization of the augmentation process of AugMix [2]. Augmentation operators such as *translate_x* and *rotate* and weights such as *m* are randomly sampled. Those randomly selected operators allow to explore the semantically equivalent input space around an image. Mixing these images together produces a new image without veering too far from the original [2]. 51
5.2 In the first to columns (A,B) it is depicted the original use of MixStyle [15] as featur augmentation-based regularization method with a standard encoder-decoder structure (A) and aplide to a regularize a dual-branch network with an auxiliary image decoder. In the third collumn (C) it is depicted the proposed stracture changes to MixStyle by the creators of MaxStyle. They proposed the addition of an auxiliary decoder for the generation of stylized images for feature-to-point space data augmentation. As seen in the examples on the botom of the figure, the proposed model outperformed the first two. [1] 52
5.3 On the left (a) is the architecture of MaxStyle [1]. MaxStyle reconstructs the input with augmented deature styles via style mixing and noise perturbations in the image decoder. In order to find 'harder' style compositions, the authors applied adversarial training. On the right (b) it is shown that MaxStyle generates samples with high corelation to to original but able to fool the network to undersegment. 52
5.4 This figure from the original paper visualises qualitative resaults from MaxStyle compared to other methods [1]. As seen in the figure the MaxStyle not only outperforms all other methods but the resaults are vary accurate, compared to the the graund-truth (GT) 53

List of Tables

6.1 This table presents the results of the initial round of tests. The original dataset from the Medical Decathlon (**G**) [18] was augmented using the **MaxStyle** method [1], thereby increasing the dataset size by 200%, 100%, 50%, and 20%. Subsequently, the pretrained SegFormer model [3] was fine-tuned for up to 10 epochs, selecting the best performing stage. Finally, the fine-tuned model was evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The best performance for each dataset is highlighted in bold. 56

6.2 This table presents the results of the second round of tests. The original dataset from the Medical Decathlon (**G**) [18] was augmented using the **MaxStyle** method [1] followed by the **Tile Mixing** method. The additional samples were generated by tile mixing between the original samples and their corresponding MaxStyle-augmented samples. This approach increased the number of samples in the dataset by 100%, 50%, and 20%. The augmented dataset was then used to fine-tune a SegFormer model for 10 epochs. The fine-tuned model was evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The results indicate an increase in the model's generalization ability. The best performance for each dataset is highlighted in bold. . . . 56

6.3 This table presents the final and most successful round of tests. The same steps as in the second round were followed, but this time the **MaxStyle** method [1] was combined with the **AugMix** method [2]. First, the original dataset was augmented with MaxStyle, increasing the number of samples by 20%. Then, during the fine-tuning of the model, 50%, 20%, and 10% of the input were augmented with the AugMix method. As in the previous tests, the augmented dataset was used to fine-tune a SegFormer model for 10 epochs. The fine-tuned model was then evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The results show an increase in generalizability to out-of-distribution (OOD) samples. The best performance for each dataset is highlighted in bold. 56

A.1 This table presents all the different tests conducted within the scope of this work. The first column shows the number of each test, which is used to refer to each specific test in the following tables and descriptions. The second column provides a coded description of each test. The descriptions focus on the dataset that the model of the test was fine-tuned on and the creation of that dataset. Information about the model, such as hyperparameters, is presented in Table A.2. 66

A.2 This table presents important information about the training of the models for each test. As mentioned in section 6.2, the SegFormer model was used in all experiments [3]. The first column shows the number of each test (for reference, see Table A.1 and section A.1.1). The second column indicates the batch size used during the training of each model. The learning rate and weight decay chosen for each test are depicted in the third and fourth columns, respectively. The fifth column shows the total number of epochs the model was fine-tuned. The Best Epoch column shows the epoch after which the model achieved the best performance on the validation set. The last three columns correspond to the validation loss, the validation F1 score, and the validation Dice score each model achieved after its best epoch. It is interesting to note that the best epoch numbers vary significantly between different tests. 67

A.3 In this table, the loss of the fine-tuned model of each test is depicted. The first column shows the corresponding number of each test (for reference, see Table A.1 and section A.1.1). The second column shows the performance of the model on the IID ground truth dataset, while the rest of the columns show the model's performance on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations. The loss of all models is significantly lower on dataset G, which is expected as they all were trained on an augmented dataset based on dataset G. Another meaningful observation is that in the first tests, the performance of the models was worse than in the following tests, as clearly depicted by the losses in the last couple of columns. This can be attributed to the increase in complexity and sophistication of the augmentation techniques and combinations as the testing process progressed. 68

A.4 This table presents the F1 score achieved by the different fine-tuned models. As in Table A.3, the first column refers to the number of each test (for reference, see Table A.1 and section A.1.1), the second column shows the F1 score of the models on the ground truth dataset, and the final six columns show the F1 score of each model on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations. 69

A.5 This table depicts the Dice coefficient that the models achieved in each of the 20 experiments. This is one of the most important parts of this work. The Dice coefficient was the metric aimed to increase, especially in OOD samples and datasets. Following the format of Tables A.3 and A.4, the number of each test is shown in the first column (for reference, see Table A.1 and section A.1.1), the second column shows the Dice coefficient of the models on the ground truth dataset, and the final six columns show the Dice coefficient of each model on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations. An important observation is that even though tests 14 and 17 were, on average, the best-performing models on OOD datasets, in certain datasets they were outperformed by other models. 70

A.6 This table summarizes all the previous ones by presenting the average of each of the three metrics (Loss, F1 score, and Dice coefficient) that the models achieved on OOD datasets. According to this table and by focusing on the Dice coefficient, the best-performing models were created in tests 14 and 17, achieving a Dice coefficient equal to 0.789. 71

Chapter 1

Introduction

In recent years, artificial intelligence has experienced a notable surge, revolutionizing several aspects of science in many different scientific fields. This surge can be attributed to many factors such as the increased computational power that researchers have access to, and the abundance of data available for training artificial intelligence models. Thanks to these advancements, new systems, known as deep learning systems, have taken a prominent position in artificial intelligence applications.

Deep learning, involving large artificial intelligence models that analyze data with great detail, has fundamentally transformed the field of computer vision in recent years, making tremendous progress in image processing. However, modern models that typically exhibit optimal performance either make certain assumptions during training that do not align with realistic application conditions or are trained on a large volume of data, which is not always easily obtainable.

One of the most fundamental issues in training tools capable of application in various domains is the generalization of models. These systems must be able to withstand any alterations in the data, which they may not have encountered during training. Such alterations may arise from uncontrollable factors, such as weather changes in autonomous driving applications. If such extreme cases are not taken into account during training, they can lead to a significant decrease in performance during application.

Data scarcity is also a serious problem for the implementation of machine learning models. The aforementioned problem is usually addressed by using a large volume of diverse training data. However, for many fields, such as the medical field, collecting data from multiple sources may be costly or even impossible, as there is not yet an abundance of openly available data.

The need for more advanced methods of model generalization is significant, and contemporary science has found ways for artificial intelligence systems to generalize to unknown data without having seen it during training. Style transfer and adversarial learning are two of these methods and have led to significant improvements of the generalization abilities of deep learning models. However, there is a need for further improvement, as performance is still low, and the cost and time of training are quite significant. In such cases, data augmentation methods are employed, which alter the content of the image with the aim of enhancing the robustness of the model.

By utilizing data augmentation methods, one can improve performance with minimal to zero additional costs in terms of time and training resources. However, complex augmentations are avoided in the healthcare sector due to their impact on the semantic content of the images. Nevertheless, numerous studies have demonstrated that introducing significant complexity during training can lead to substantial improvements to the generalization ability of the trained model. Therefore, it is intriguing to explore how such complex augmentations, in combination with modern domain generalization methods, can lead to improved performance without an increase in computational resources.

1.1 Objective of the Thesis

The subject of this thesis is to investigate the utilization of complex data augmentations in medical images, specifically Magnetic Resonance Imaging samples, in combination with modern style transfer and adversarial learning methods, aiming to create a more robust system for unknown data and image alterations. The ultimate goal is to develop a comprehensive, novel augmentation approach that can contribute, in a computationally efficient manner, to the training of valuable tools for the medical domain. Such systems could assist or potentially replace conventional methods in medical image segmentation and serve as advisory tools for health care professionals, thus reducing diagnostic errors and promoting high-quality healthcare.

1.2 Thesis Organization

This work is structured into seven chapters. In Chapter 1 is provided the introduction to the thesis. Chapter 2 presents the theoretical background of the fundamental technologies related to this thesis. Initially, the field of image segmentation is described, followed by the problem faced by deep learning networks regarding generalization to unknown domains, the types of resolution methods, and finally, the basic architectures used for image processing. In Chapter 3, relevant works on the topic are initially described, followed by the specific objectives of this work. In Chapter 4 are presented the datasets used and their specific characteristics as well as the data preprocessing. Chapter 5 analyzes in greater detail on the tools used in the research of this work, while Chapter 6 presents the experimental results. Finally, Chapter 7 provides the contribution of this thesis, as well as potential future extensions.

Part I

Theoretical Part

Chapter 2

Theoretical Background

In this chapter, the theoretical background necessary to understand the work and content of this body of work, as well as the methods and approaches implemented, will be presented and analyzed.

Starting in Section 2.1, the main theory behind Deep Neural Networks (DNNs) will be analyzed and several types of Deep Neural Networks will be presented. Following that, the concept of Image segmentation will be investigated in Section 2.2. The basis of the Domain Generalization problem will be analyzed in Section 2.3. Finally, in Section 2.4, the Style Transfer technique will be analyzed. By the end of this chapter, all the knowledge and information required to understand the work of this Thesis will have been presented and analyzed.

2.1 Deep Neural Networks

An understanding of Neural Networks and Deep Learning Models is required to better understand the following concepts and the work presented. Thus, it is appropriate to present the main theory behind Neural Networks and some information on the main Architectures that are being used for image processing.

2.1.1 Neural Networks

The Term Artificial Neural Networks represents a category of artificial intelligence algorithms which creation was inspired from biological neural networks in the human brain and its information processing mechanisms. They consist of interconnected nodes, connected by interneuron ages, which carry specific weights. The manipulation of those weights allows the network to learn from seen data through the activation functions.

Historical the first mention of Artificial Neural Networks was in 1943 form a neurophysiologist Warren S. and a mathematician McCulloch and Walter Pitts on their paper "The Logical Calculus of the Ideas Immanent in Nervous Activity" [19]. For several years after their introduction, artificial Neural Networks failed to achive significant results. The last decades researchers have made significant advancements on the field of Artificial Neural Networks due to the introduction of new architectures and techniques, such as Convolutional Neural Networks [20],[5] and others that are going to be presented and analyzed in the

following subsections, but also due to the advancements of computer hardware such as better Graphics Processing Units and Tensor Processing Units [21].

2.1.2 Convolutional Neural Networks

Developed to tackle the increasing complexity of data as well as the rising size of datasets, Convolutional Neural Networks (CNNs) serve as the fundamental architecture for image processing [20]. They superseded previous networks that relied on densely connected layers, which became unwieldy due to their excessive connectivity. CNNs are engineered to autonomously discern patterns and features from images and similar data structures through convolutional layers and pooling operations. This hierarchical feature extraction renders CNNs especially adept for tasks encompassing image classification, object detection, and facial recognition [5].

The main workflow of a Convolutional Neural Network [4] consists of:

- The Input Image
- One or more Convolution Leyers (or Kernels)
- One or more Pooling Layers
- One or more Classification Fully Connected Layers
- The Output Layer

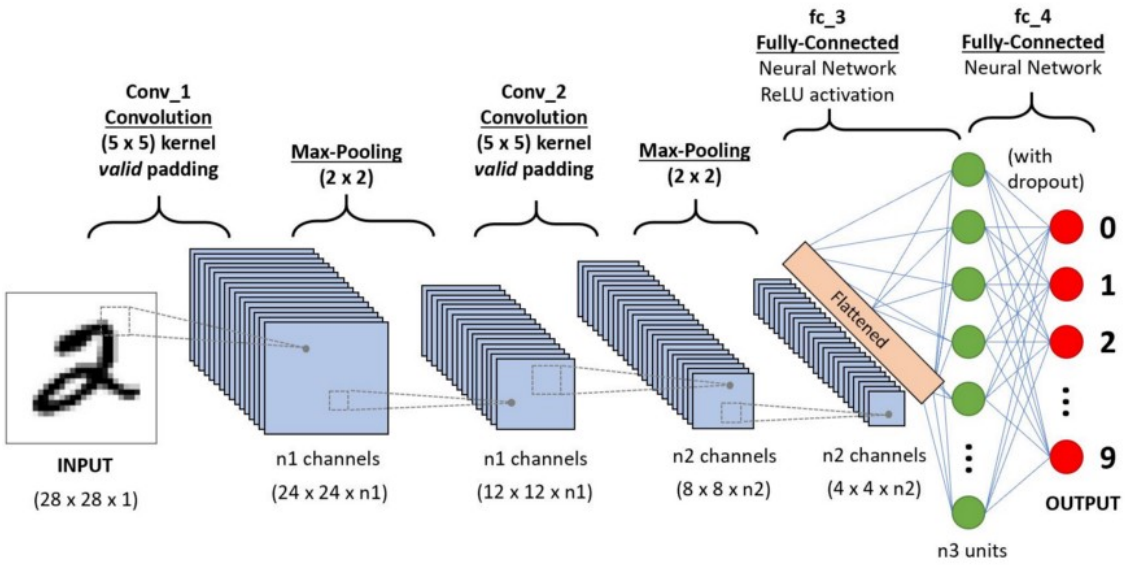

Figure 2.1. *An example of CNN architecture[4]*

The most important and unique parts of a CNN are the Convolution Layers and the Pooling Layers. The functionality and usage of those types of layers is analyzed bellow.

Convolution Layer (or Kernel)

The Convolution Layers of a CNN are based on the following algorithm. Let h,w,c be the height, width and number of channels of the layer's input respectively and K the Convolved Feature a matrix with dimensions $k \times k$. A convolutional operation is performed on every S^{th} $k \times k$ sub-matrix of the input with K (Kernel), where S is the determined Stride. For example if $S = 2$ then the convolution is going to be performed on every other $k \times k$ sub-matrix, decreasing this way the size of the input by a factor of 2, but in a way that as much of the original information is being kept, thanks to the carefully created Kernels. When the number of channels c is greater than 1, as it is in RGB images, the Kernel K should have the same number of depth. Finally, depending on the usage, rows and columns may be added around the input image with specific values (0,1 or the average of their neighbors) with the goal of not shrinking the dimensions of the output. Figure 2.2 depicts the convolution process that has been described above. The Kernel K has dimensions 3×3 and depth 3 as the image input matrix has three layers. After the convolutions take place, the sum of the three results is entered to the Output after the Bias is added. In this case a single row of padding is being used with values 0.

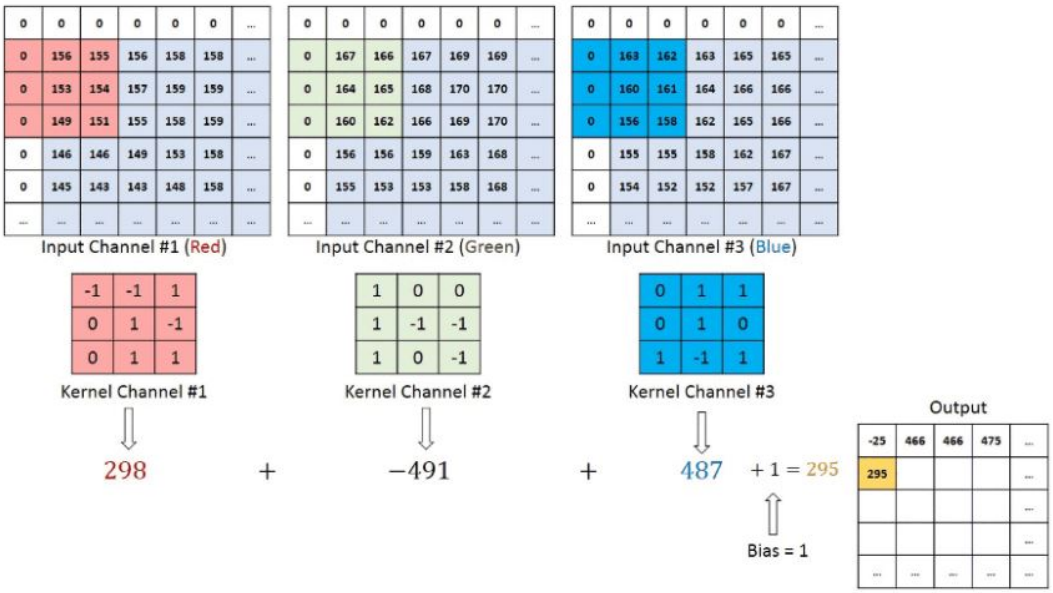

Figure 2.2. *An example of a convolution operation[4]*

The main target of the convolution layer is to obtain high-level features from the input image. High-level features can be lines, edges, blobs but also surfaces and characteristics of images such as tires in cars and windows in buildings. In addition the layer also performs an operation on low-level features such as color and gradient orientation.

Pooling Layer

Pooling layers works similar to Convolutional layers but do not implement convolution of the input matrix with a kernel. A Pooling Layer replaces a sub-matrix of the input with

a specific number. That number depends on the type of Pooling Layer, which there are two:

- **Max Pooling:** It returns the maximum value of the elements in each sub-matrix
- **Average Pooling:** It returns the average value of the elements in each sub-matrix

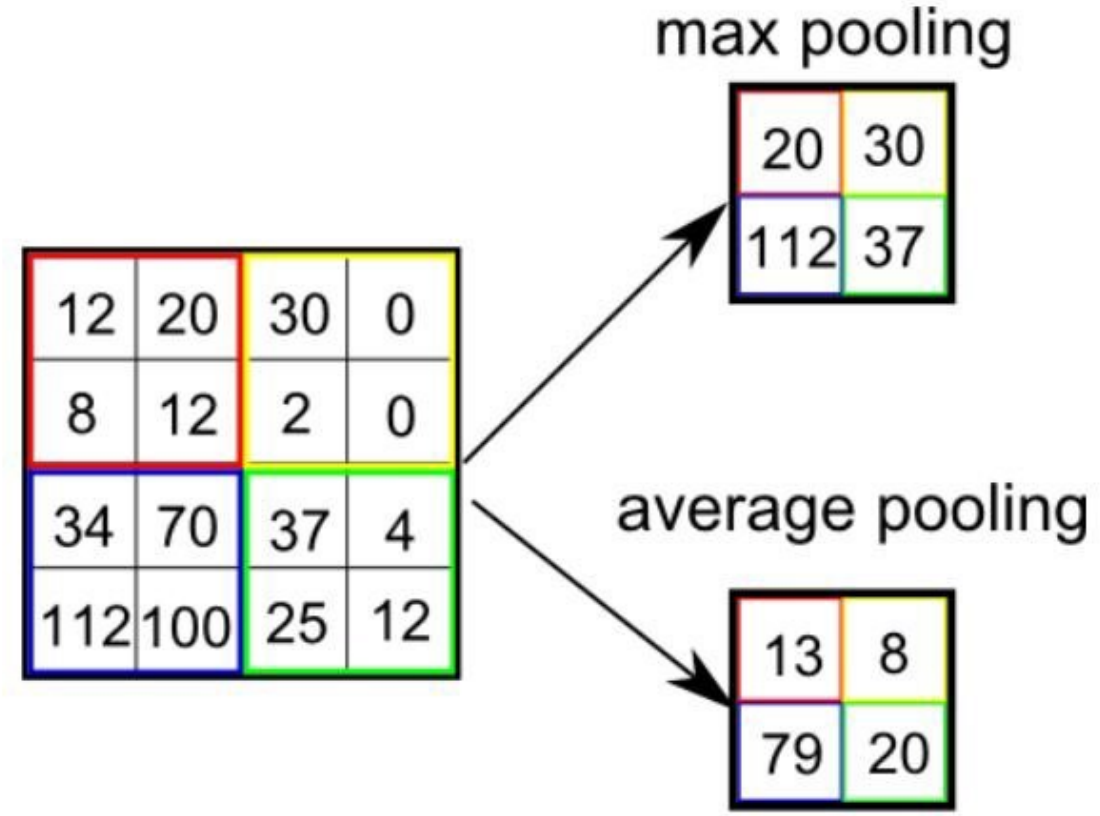

Figure 2.3. *An example of a Pooling Layer operation[4]*

Usually, the input of a Pooling Layer is the output of a Convolutional Layer, and the main functionality of a Pooling Layer is to decrease the size of its input. This way the computational and spatial complexity of the following layers gets reduced. Additionally, Pooling proves advantageous in extracting dominant features that remain invariant to both rotation and position, underscoring the necessity for effective process maintenance. Max Pooling is also used as a noise-suppresing operation, as it discards noisy activations.

Modern avances of Convolutional Neural Network

During the last years Convolutional Neural Networks have been advanced both in terms of their architectural designs and overall performance capabilities. The availability of larger and more diverse datasets has played a pivotal role in training more robust and generalized models. Additionally, techniques like batch normilisation, data augmentation, and implementations of regulirization methods have contributed to the improvment of the learning process, and the models generalisation and robustnes. Morover, the evolution

of network architectures towards deeper structures has allowed CNNs to capture more complex patterns and features from input data. The introduction of deep pre-trained models, including popular architectures like VGG [5], Inception [22], and ResNet [6], has revolutionized the field by enabling researchers and practitioners to leverage transfer learning methodologies, by finetuning models trained on one task to perform effectivly on some ralated tasks.

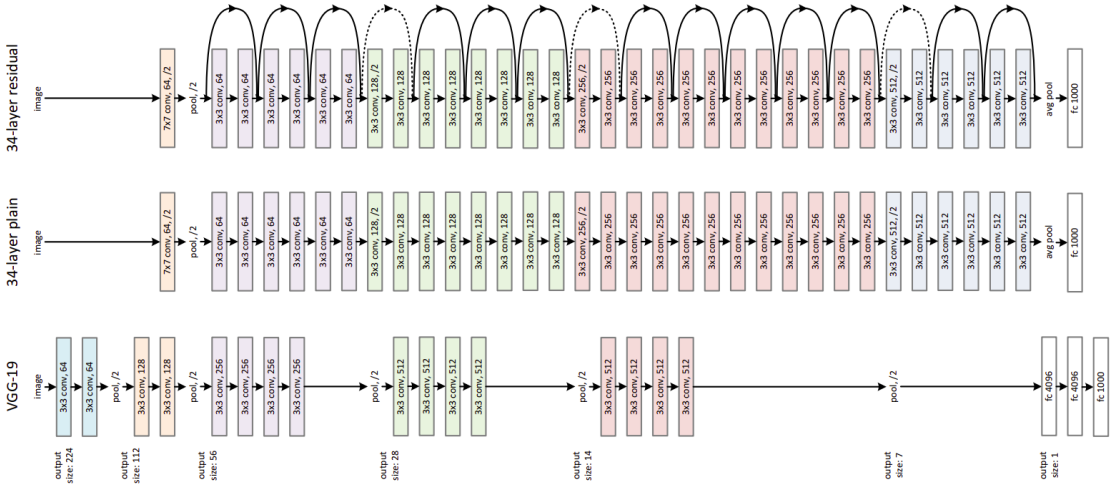

Figure 2.4. *An example of deep Convolutional Neural Networks' architectures is shown. At the bottom, the architecture of the VGG-19 model is depicted [5], in the middle is presented a plain CNN with a 32-parameter layer, and at the top, the architecture of the ResNet model is illustrated [6].*

2.1.3 Fully Convolutional Neural Networks

As mentioned in Section 2.1, one of the main functionalities of Neural Networks regarding image processing is image segmentation. Fully Convolutional Networks (FCN) are created specifically for this task on pixel level. Their target is the classification of each pixel of the input image to one of predetermined classes [7]. The main deference between Convolutional Neural Networks and Fully Connected Neural Networks is that Fully Connected Neural Networks replace the fully connected layers of Convolutional Neural Networks with Convolutional Layers. This is done so that the final output of the Network has the same dimensions as the input and the spatial information is preserved throughout the Network.

Moreover, through the strategic implementation of skip connections, an approach in which feature maps originating from the terminal layers of the model are refined and fused with those from preceding layers, a Fully Convolutional Network can effectively combine rich semantic context, predominantly extracted from deep layers, with nuanced appearance cues meticulously captured from shallow layers. This harmonious integration empowers the model to generate highly accurate and intricately detailed segmentations.

A remarkable implementation of Fully Convolutional Neural Networks for image segmentation is U-Net, proposed by O. Ronneberger et al. [8]. It employed shrinking as well as

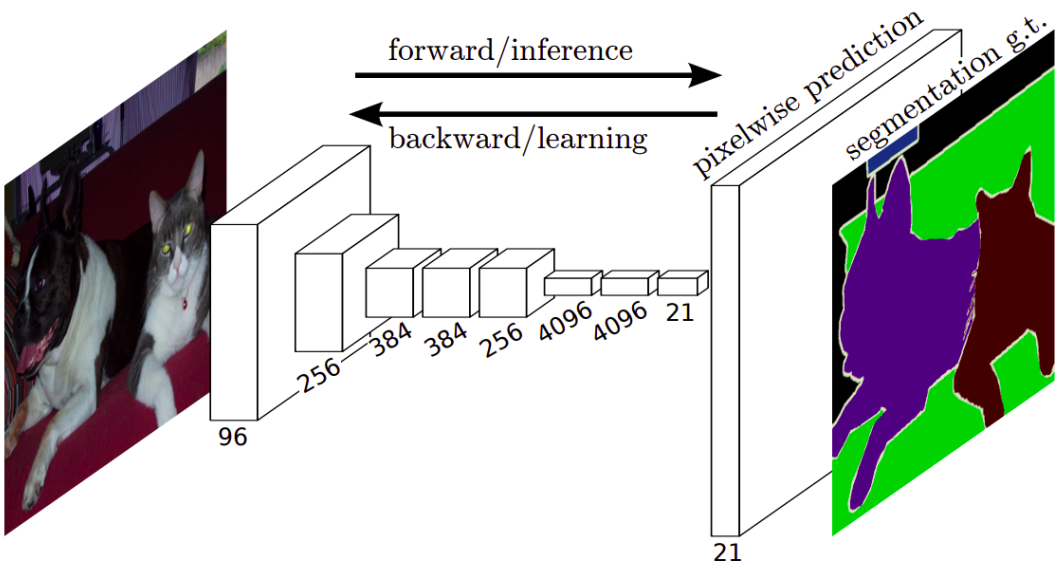

Figure 2.5. *An example of the basic architecture of a Fully Connected Network (FCN) [7]*

expanding operations in a U-shaped architecture which is visible in Figure 2.6.

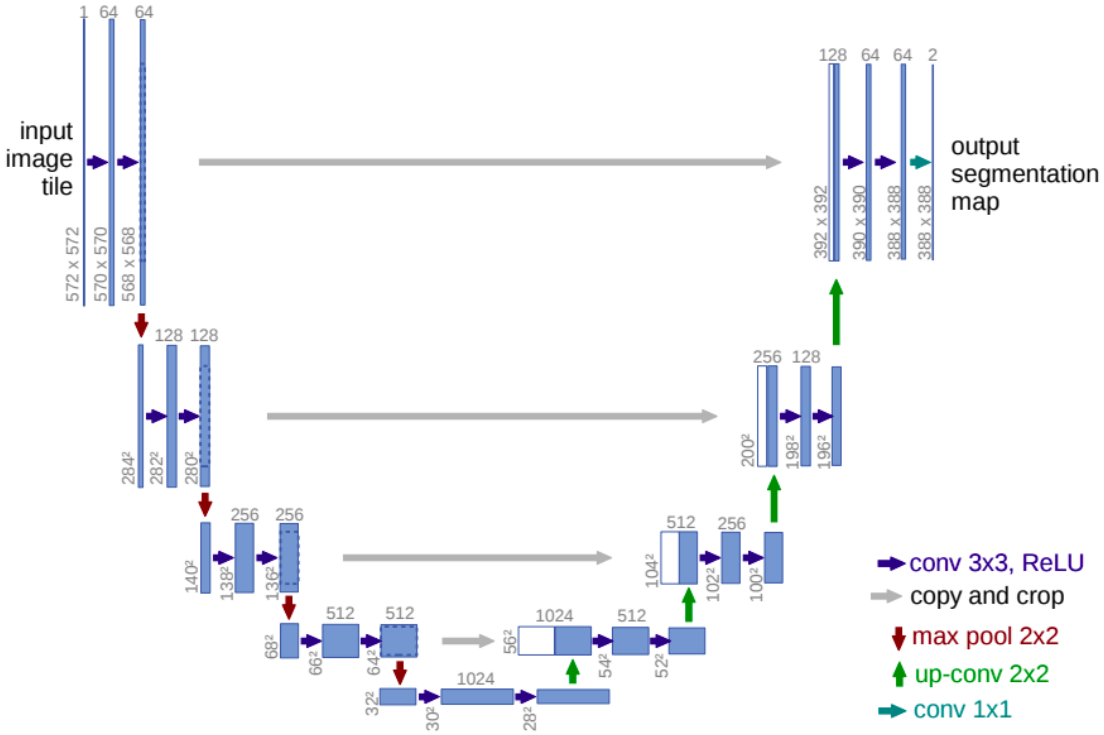

Figure 2.6. *The original architecture of U-Net [8]. Each blue rectangle represents a multichannel feature map. The white rectangles represent copied feature maps. Finally, the arrows show the different kinds of operations.*

The architecture consists of two main parts: a contracting path that captures the context, and a symmetric expanding path that helps with the precise localization of each class. The first part is similar to a traditional Convolutional Network and its purpose is

to extract features using 3×3 unpadded convolutions, each followed by a rectified linear unit (ReLU) and a 2×2 max pooling with stride 2 [8]. In each step of the contracting path, the number of feature channels is doubled. The second part, consists of several upsampling of the feature maps each followed by a 2×2 convolution which halved the feature channels, a concatenation with the corresponding feature map from the first part, and two 3×3 convolutions, each followed by a rectified linear unit (ReLU) [8]. Finally, a 1×1 convolution is implemented to map each of the feature vectors to the number of predetermined classes.

U-Net and other implementations of Fully Convolutional Networks have been used extensively in the medical field on tasks like skin cancer segmentation [23], iris segmentation [24], brain tumor segmentation [25], and Instance-aware Semantic segmentation [26].

Fully Convolutional Networks revolutionized how medical images are analyzed, making it easier to accurately and automatically identify organs and abnormalities across different types of scans like MRIs, CT scans, and X-rays [27]. Using FCNs, doctors and healthcare professionals can speed up their work and improve diagnoses by creating detailed maps of body structures and pinpointing areas of concern.

However, while Fully Convolutional Networks are widely used and effective, they do have some limitations. One key issue is that they can't process images in real-time, which can be a problem for tasks needing immediate results. Also, FCNs face difficulties in taking in all the necessary information from an image, particularly the broader context that helps with accurate segmentation. Additionally, they are difficult to implement on three-dimensional images, limiting their usefulness in certain situations.

To overcome these challenges, researchers have been working on improving Fully Convolutional Networks. By refining their architecture and methods, they aim to make FCNs faster, better at understanding context, and more adaptable to different types of medical images. These efforts are driving innovation in the field, leading to new ways to analyze medical images more effectively.

2.1.4 Encoder Decoder Models

Encoder-Decoder neural networks represent a fundamental element within modern deep learning, significantly impacting various domains through their adeptness in seamlessly converting data across diverse representations [28],[29]. The initial phase of this architecture, the encoder, analyzes the input, whether it be an image, text, or sequence, compressing it into a condensed form commonly referred to as the latent space or embedding. Through intricate layers of processing, the encoder distills vital features while abstracting noise and irrelevant details, thereby constructing a representation that encapsulates the essence of the input.

In the landscape of image analysis, the encoder constituent of the architecture receives inputs of high dimensionality, such as images, and meticulously processes them, gradually molding them into a lower-dimensional embodiment known as a latent space. This transformation, or encoding, transpires through a cascade of convolutional and pooling operations, progressively distilling and consolidating salient features from the input domain.

These condensed representations serve as encoding of essential insights about the input while simultaneusly sieving out unneeded intricacies, thus facilitating the extraction of pivotal features from convoluted datasets.

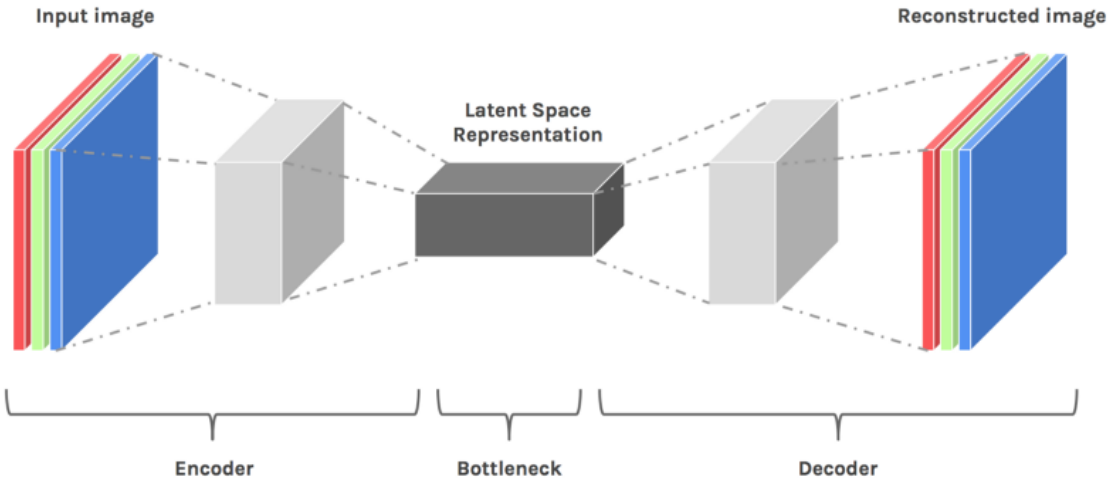

Figure 2.7. *An example of the basic architecture of an Encoder Decoder Network [9]*

Conversely, the decoder counterpart adeptly assimilates the diminished information from the encoder and harnesses it to either recreate the original input or the apropeiate outputs imbued with the desired attributes [28],[29]. In scenarios revolving around images, the decoder typically comprises an assembly of convolutional layers that incrementally amplify the latent representation, meticulously reinstating it to its pristine dimensions. This regenerated output fulfills multifarious objectives, spanning from image rejuvenation and chromatic enhancement to style emulation and resolution augmentation. In alternative applications, such as linguistic translation within the realm of natural language processing, the decoder utilizes the lower-dimensional information to engender textual sequences or alternate manifestations of data.

Encoder-Decoder architectures furnish an adaptive framework for information compression and exploitation, endowing a vast array of tasks with the capability for efficient feature extraction and reconstitution.

2.1.5 Vision Transformers

A Transformer in machine learning is a deep learning model that uses the mechanisms of self-attention. Self-attention allows the model to weigh the importance of different parts of the input when processing them, enabling it to capture dependencies between parts in a more flexible manner Convolutional Neural Networks.

Vision Transformers architectures consist of several steps. Firstly the input image is devidet into a grid of fixed-size patches which then are flattened into sequences of vectors. Each patch vector is associated with an embedding vector. Additionally, a learnable embedding vector is associated with a special token called the "class" token, which represents the entire image. Afterwords, ositional encodings are added to the token embeddings to provide information about the spatial arrangement of the patches.

This allows the model to understand the relative positions of the different patches within the image. The token embeddings, along with the positional encodings, are passed through a series of transformer encoder layers. Each layer consists of self-attention mechanisms that allows the model to capture dependencies between different patches and feed-forward neural networks that enable nonlinear transformations. The final layers of Vision Transformers vary depending on the task of the model. For clasification models the final layer can be a Multy Layer Presseptron Head chich takes the final vector representation of the input image and outputs the propability of each class [10]. For segmentation oriented models the final part of the model is a Decoder simila to Encoder-Decoder architectures that outputs a matrix with the propabilities of each class for each pixel of the original input image [3].

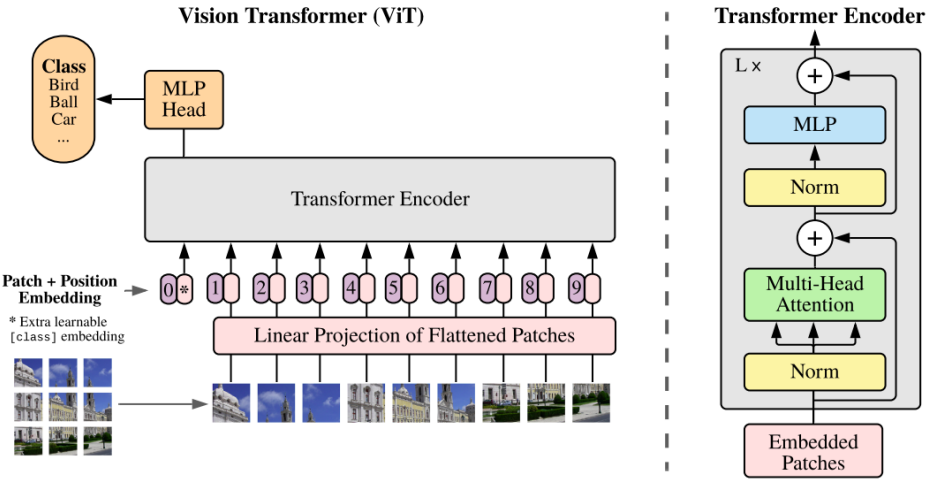

Figure 2.8. *The architecture of ViT from the original paper [10] is shown. On the left, an overview of the architecture and its different layers is depicted. On the right, there is a close-up of the Transformer Encoder part and its different layers.*

Self-attention based architecturs and in particular Transformers architectures originaly started as an architecture for Natuarl Language Processing (NLP) tasks and quickly bekame the de-facto standard approach [30]. Their computational efficiency and scalability made the training of unpresidently deep modeles posible. Inspired my the succes Transformers had in Natuarl Language Processing, works like [31], [32] and [33] tried to combine self-attention with traditional Convolutional Neural Network archytectures, but without succeading to overperform the best model at the time [6].

ViT was the first body of work that experimented with applayinf standar Transformer arcitectures directly to images with as litle modifications as posible [10]. When trained on large datasets, ViT overcomed its lack of inductive biases (which are inherit to Convolutional Neural Networks) achived exelent resaultes, beatin the then state of the art ResNet and opening a new field of Transformer architecturs on Image Processing.

2.2 Image segmentation

Image segmentation is one of the primary applications of deep learning systems in computer vision, where an image is divided into different regions of interest at the pixel level [34], [35]. This process is crucial for simplifying the representation of an image, making it easier to analyze and interpret.

Image segmentation plays a central role in a wide range of applications. In autonomous vehicles, image segmentation is essential for navigation on surfaces and pedestrian detection [36],[37]. Additionally, it is used in security and monitoring applications involving satellite images, helping to detect and track objects and changes in the environment [38]. In medical image analysis, for instance, it is used to identify tumors and measure tissue volumes in radiographs and other medical imaging techniques [39].

2.2.1 Image Segmentation Types

Semantic Segmentation

Image Segmentation can be analyzed in two basic categories based on the target of the segmenting process. The first category is Semantic segmentation. Semantic segmentation is a computer vision task that involves assigning a class label to each pixel in an image, categorizing the image into different semantic classes like "road," "tree," or "building." This task can be approached through supervised learning, which requires a labeled dataset for training, or unsupervised learning, which doesn't need labeled data and uses various methods to learn labels.

The evolution of semantic segmentation models began with fully convolutional networks (FCNs), which were adapted from image classification models. Subsequent advancements include models like DeepLab, FastFCN, DeepLabV3, and newer transformer-based models, each introducing improvements in accuracy and efficiency [40], [41], [42]. These models are essential for applications in autonomous driving, medical imaging, and robotics.

Instance Segmentation

Instance segmentation is an image segmentation task that identifies and segments individual objects in an image, assigning a unique label to each object and delineating their boundaries. This approach relies on the appearance or context of objects to perform segmentation. The Mask R-CNN architecture is a common model used for instance segmentation [12]. This technique is increasingly applied in various fields, impacting daily life.

2.3 Style Transfer

Each artist has a unique way of expression and thus, a unique style of art. Some like to draw oil paintings with heavy strokes on canvas and others prefer sketching with a pencil on a piece of paper. If you gave each one of them the task of drawing something

Figure 2.9. *Examples of Image Segmentations [11]. Specifically, the Semantic Segmentation of the input images are depicted*

specific, e.g. a dog, a lot of unique but in some ways similar pictures of dogs. But for an artist to draw a Picasso sketch dog in the style of Van Goh's Starry Night style, it requires studying both artworks to find the unique characteristics that compose the dog sketch and redraw them using the unique techniques that would make it appear as a Starry Night style artwork.

Style transfer is a modern technique in computer vision and image processing that enables the alteration of an image's artistic style while maintaining its original content. This idea gained significant attention with the development of neural style transfer algorithms,

Figure 2.10. *Examples of Instance Segmentations from the original paper that proposed Mask R-CNN [12].*

which utilize the capabilities of deep neural networks to produce visually striking and artistic images. The use of Convolutional Neural Networks (CNN) for the reproduction of images in the style of paintings was first proposed by Gatys et al. [13] in 2015. They showed that a CNN can capture the *content* information of a photograph, the *style* information of an artwork as a summary of feature statistics, and finally combine them to reproduce a new image. Their work was the ground stone for the, now well documented, field of *Natural Style Transfer*.

The fundamental concept of style transfer involves separating and recombining the content and style of two distinct images to generate a new and unique composition. The process typically involves two key components: the content image and the style image. The content image holds the objects and elements to be preserved in the final output, while the style image contains the stylistic features to be applied to the content. Neural networks are essential in this process, as they analyze the content image to extract its features and the style image to capture its stylistic elements. These extracted features are then used to create a new image that merges the content from the content image with the stylistic attributes of the style image [43]. By employing images from various domains, style transfer can generate images from different fields, thereby enriching the dataset and enhancing the model's ability to generalize from a single dataset. The key quantities involved are the mean and variance of the feature maps, as these metrics store the information about an image's style within the layers of a neural network [15], [1].

2.4 Out of distribution Problem

Deep learning models typically presume that the training data and the data used during deployment share the same distribution. This assumption implies that the mean

Figure 2.11. *In this Figure from the original paper by Gatys et al. there are images with the content of a photograph of Neckarfront in Tubingen, Germany* ***A*** *recreated influenced by the style of five well known paintings [13]. The paintings used that provided the style are shown in the bottom left corner of each generated image and are:* ***B*** *The Shipwreck of the Minotaur by J.M.W. Turner, 1805,* ***C*** *The Starry Night by Vincent van Gogh, 1889,* ***D*** *Der Schrei by Edvard Munch, 1893,* ***E*** *Femme nue assise by Pablo Picasso, 1910, and* ***E*** *Composition VII by Wassily Kandinsky, 1913.*

and standard deviation of the image features are consistent across both datasets /citeOOD1,/citeOOD2,/ci However, this is often not the case in practice, as various external factors, such as the method of image acquisition and environmental conditions, introduce variations in the image domain, known as *domain shifts* [44].

To illustrate, consider a model where the inputs and outputs are connected via the joint probability distribution function $P(X, Y)$, with X representing inputs and Y

representing outputs. During the training and validation phases, the inputs usually come from the same distribution, which means there are no changes in the joint function. However, when applying the model in a real-world context, the data used for training, validation, and testing comes from the conditional distribution $P(X, Y|Z \in U)$. Here, Z is a random variable that might not be observable and is not independent of Y and X, while U is a subset of Z [45].

In the medical field, for example, this issue is evident when MRI scans from different machines show significant discrepancies, even if they represent the same content. These discrepancies are due to the noise and artifacts introduced by medical imaging devices, as well as variations in the magnetic field. Such differences can lead to notable variations during the training of an AI model, despite thorough preprocessing [46],[47].

As a result, the performance of deep learning models can significantly decline in the presence of substantial domain shifts, preventing their practical and widespread application [48],[49],[50]. One potential solution is to use images from a variety of domains to make the model robust against such deviations. However, this approach is often impractical due to the lack of sufficient and diverse data available [46].

To overcome this challenge, the field of deep learning has developed alternative approaches that do not rely on extensive datasets for training. These methods, known as domain generalization techniques, aim to enhance the model's ability to generalize effectively across different domains.

2.5 Domain Generalization

To mitigate the issues caused by domain shifts that was analyzed in Section 2.4 and the lack of diverse data, deep learning models often employ domain generalization techniques. Domain generalization addresses the machine learning challenge of training a model that can generalize to unseen domains by using only labeled data from a set of initial domains during the training phase [51]. The objective is to develop a representation that remains consistent across various domains, capturing their shared underlying structure while being resilient to domain-specific variations. This approach is especially useful in contexts where the target domain is unknown or inaccessible during training, such as in medical diagnosis or autonomous driving.

The concept of domain generalization has its roots in the early 2000s, when Blanchard et al. introduced it as a distinct machine learning problem [52]. The initial motivation for domain generalization came from a medical application called automatic gating in flow cytometry data. The goal was to create algorithms that could automate the classification of cells in blood samples based on various properties, such as distinguishing between lymphocytes and non-lymphocytes. This technology is critical for improving the efficiency and accuracy of patient health diagnostics, as manual classification is labor-intensive and requires specialized knowledge. Unlike domain adaptation or transfer learning, domain generalization tackles scenarios where target data are not available during model training.

The literature extensively explores domain generalization, with numerous methods developed for different applications. These methods are generally classified into four

categories: feature-based methods, model-based methods, metric-based methods, and meta-learning-based methods. Feature-based methods focus on learning a domain-invariant feature representation by adding regularization terms to the loss function or employing domain separation networks. Model-based methods are designed to train models that are robust against domain shifts. Metric-based methods aim to establish a metric space that is unaffected by domain shifts. Lastly, meta-learning-based methods aim to train a meta-learner capable of quickly adapting to new domains with minimal labeled data.

These approaches collectively enhance the capability of deep learning models to perform reliably in real-world scenarios, despite the challenges posed by domain shifts and limited data variety.

Chapter 3

Related work on Domain Generalization on images

In this chapter, a presentation and analysis of some state-of-the-art methods used in Domain Generalization on Images are included, which have an effect on this work.

3.1 Ralated work

3.1.1 Adversarial Training

Adversarial training is a technique in machine learning aimed at improving the robustness of deep learning models. Inspired by adversarial examples—inputs designed to mislead a neural network—this approach involves exposing the model to these challenging examples during training. This exposure forces the model to learn resilience against such perturbations. The concept stems from the realization that even a slight, imperceptible noise added to images can cause deep learning models to produce incorrect outputs, despite no visible change to the human eye [53].

The primary function of adversarial training in enhancing robustness is its capacity to improve model performance when encountering adversarial inputs or unfamiliar data distributions. By integrating adversarial examples into the training data, the model learns to identify and adapt to subtle changes and variations in input data, thereby reducing its vulnerability to manipulation and exploitation. This approach strengthens the model's defense against adversarial attacks and simultaneously enhances its generalization performance. Adversarial training is applied in various fields, such as computer vision, natural language processing, and reinforcement learning.

Primarily, adversarial training has been applied within Generative Adversarial Networks (GANs), where it addresses the issue of distribution minimization through a minimax game involving two players [14]. This involves training a discriminator to differentiate between real and generated fake images, while simultaneously encouraging the generator to fool the discriminator. This method, used to enhance model robustness, is widely adopted for domain generalization [54],[55], [56],[57]. In modern techniques, adversarial learning can also be performed without a discriminator, relying instead on the "competition" between two distinct errors during training [1].

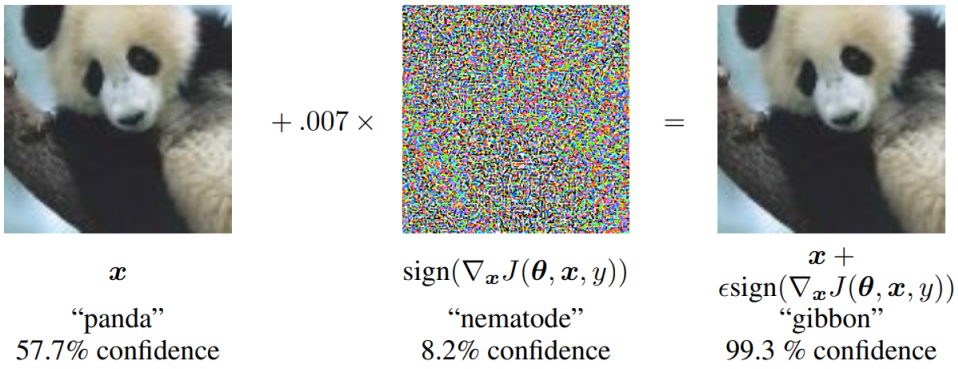

Figure 3.1. *An example of the generation of an adversarial example from the original paper [14]. The authors add to the original image x an imperceptible vector whose elements are equal to the sign of the elements of the gradient of the cost function. This way the classification of the input image changes to be faulty*

3.1.2 Style Transfer

As analyzed in section 2.3, style transfer is a technique in computer vision that involves creating a new image by merging the content of one image with the style of another. The aim of style transfer is to produce an image that preserves the content of the original image while applying the visual style of a different one [58].

Recent advancements in style transfer have led to its widespread use as a method for domain generalization [59],[60]. Various generalization techniques like [61], [62], [63] utilize pre-existing style transfer models, like AdaIN [60], or develop networks that learn from examples to perform data augmentation with specific styles [1],[59]. Furthermore, external styles are used to enhance the diversity of training data [64]. This technique is quite beneficial, as it enables the creation of images representing different domains, resulting in improved model generalization even with limited data.

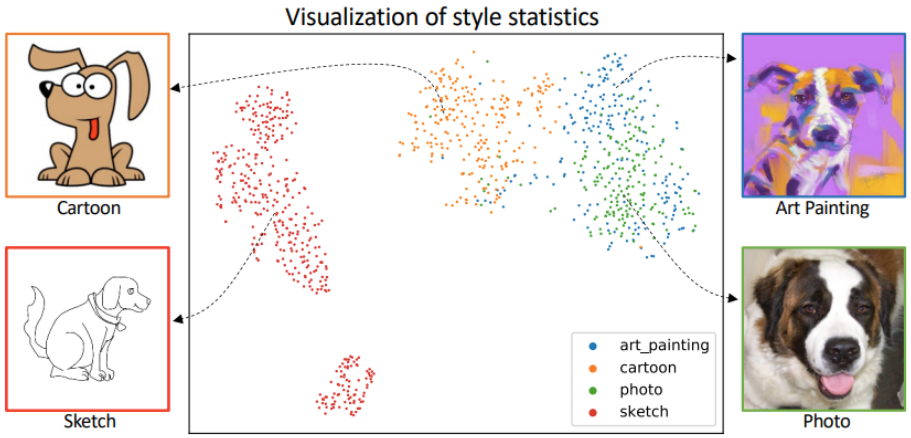

Figure 3.2. *In this 2-D t-SNE visualization of style statistics from the original paper that proposed MixStyle [15] it is shown that the four different domains (Cartoon, Sketch, Art Painting, and Photo) are clearly separated.*

3.1.3 Data Augmentation

Data augmentation involves the creation of new samples by altering the data in a way that makes them different from the original but still retains the information they contain [65]. Feature-level augmentation has emerged as a groundbreaking technique specifically designed to address the challenges associated with domain generalization [15], [16]. Within the framework of domain generalization, feature-based augmentation operates on the premise that CNN feature statistics encapsulate domain-related information. This method seeks to improve the robustness and adaptability of models to unseen domains.

Two prominent examples of this approach are MixStyle and Mixup. MixStyle enhances style augmentation by blending CNN feature statistics from instances across different domains. MixStyle's process enables models to adapt more effectively to new, unseen domains by integrating a variety of visual styles and characteristics [15]. On the other hand, Mixup performs augmentation at the pixel level and extends its influence to the feature space, which allows for the seamless merging of instances from different domains at the feature level [16]. This blending technique broadens the training distribution by incorporating prior knowledge that the linear interpolation of feature vectors should correspond to the linear interpolation of their associated targets.

The integration of feature-based augmentation strategies such as MixStyle and Mixup has significantly advanced the field of domain generalization. These methods equip models with the capability to perform reliably across a diverse range of domains without necessitating extensive training on specific domains. By adopting these innovative approaches, researchers have enhanced the ability of models to generalize effectively, thereby increasing their robustness and overall performance in various applications.

AugMix is another method of image augmentation proposed by Hendrycks et al. [2]. It enhances model robustness and improves uncertainty estimates, easily integrating into existing training pipelines. It uses simple augmentation operations, stochastically sampled and layered, to create diverse augmented images. Furthermore, it employs a consistency loss to ensure the model maintains stable predictions across these augmentations. This approach results in improved model performance, robustness, and reliability.

3.1.4 Vision Transformers for Medical Imaging

The application of visual transformers in medical imaging, particularly for segmentation, has shown significant advancements and promise. Transformers, originally designed for natural language processing, have been adapted to the medical imaging domain due to their ability to capture long-range dependencies and contextual information effectively. This adaptation has led to the development of various transformer-based architectures that excel in medical image segmentation tasks by addressing the limitations of convolutional neural networks (CNNs) in capturing global relationships within images.

One prominent model in this field is the Vision Transformer (ViT), which has been adapted for medical image segmentation [10]. ViT splits an image into patches and processes each patch as a token, leveraging self-attention mechanisms to capture dependencies across the entire image. This method contrasts with CNNs, which often struggle with

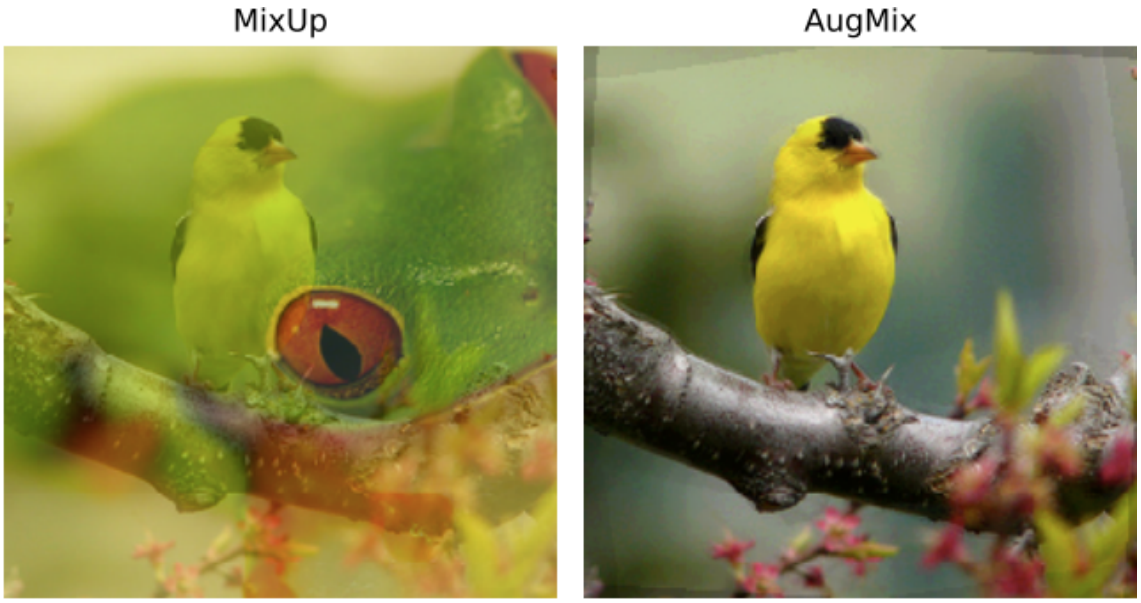

Figure 3.3. *In this figure, two examples of augmented images are shown. The left image is created using Mixup [16], while the right is created using AugMix [2]. The fisrt one is a combination of two different images, while the second is a combination of deviations of one image.*

capturing long-range dependencies due to their limited receptive fields. ViT-based models, such as UNETR and TransBTSV2, have demonstrated superior performance in segmenting complex medical images like MRIs and CT scans by effectively integrating global context into their predictions [66], [67], [68] .

Another significant advancement is the development of hybrid models that combine the strengths of CNNs and transformers. For instance, models like TransAttUNet and Swin UNETR incorporate transformer blocks within the encoder-decoder framework of traditional CNNs [69],[70]. These hybrids utilize CNNs for local feature extraction and transformers for global context integration, resulting in highly detailed and accurate segmentation maps. This approach addresses the issue of information recession commonly seen in pure CNN architectures, ensuring that fine-grained details are preserved and utilized effectively during the segmentation process [71].

The success of transformers in medical imaging is further highlighted by their application across various imaging modalities, including X-rays, MRIs, and CT scans. Models like TransAttUNet have been specifically designed for segmenting organs and lesions from X-ray and CT images, utilizing multi-scale skip connections and guided attention mechanisms to enhance the segmentation accuracy. These innovations are crucial in clinical settings where precise and reliable segmentation can significantly impact diagnosis and treatment planning [72].

Part II

Practical Part

Chapter 4

Datasets

In Chapter 4, the datasets used for this work will be presented and analyzed. Additionally, the methods used for preprocessing the training data of the experiments will be presented, and their usage will be analyzed.

4.1 Prostate datasets

The main target of the proposed pipeline is to segment Magnetic Resonance Imaging (MRI) samples of prostates. The data used from training end evaluating the models came from four different public datasets which in total contained seven different data sources. More specifically, the data where collected by 141 multiparametric studies, which are enhanced versions of MRI aimed at better visualization of the subject.

Most of the data came from T2-weighted studies for the prostate. The T2 technique refers to adjusting the MRI scanner to emphasize the tissue relaxation times [73]. T2 is defined as the time it takes for the activated tissue protons to realign after being activated by radio waves. Tissue with high T2 time appear dark, while tissue with low T2 time appear brighter. These images are extremely useful for identifying soft tissues and organs that retain fluids and are often used in detecting abnormalities.

The datasets used in this wwork are analyzed bellow.

4.1.1 NCI-ISBI 2013

This dataset contains MRI samples from two different sources. The first source was a 1.5T Philips Achieva with endorectal receiver coil stationed at Boston Medical Center. The second source was a 3T Siemens TIM with surface coil stationed at Radboud University Medical Center in Nijman, Netherlands. The dataset was composed by the National Cancer Institute's (NCI) cancer imaging program in collaboration with the International Society for Biomedical Imaging (ISBI), for the 2013 competition [74],[75],[76].

4.1.2 Initiative for Collaborative Computer Vision Benchmarking

Initiative for Collaborative Computer Vision Benchmarking (I2CVB) created this dataset created collecting images from a 3T Siemens [75],[76],[77]. For improved representation of the samples the following methods where used:

- T2-weighted MRI (T2-W)
- Dynamic contrast-enhanced MRI (DCE)
- Diffusion-weighted MRI (DWI)
- Magnetic resonance spectroscopic imaging (MRSI)

4.1.3 Prostate MR Image Segmentation 2012

PROMISE12 contains samples from three different medical centers using different capturing methods. This dataset was created for the PROMISE12 competition for medical image segmentation for prostate [78],[75],[76].

4.1.4 Medical Decathlon

This dataset was created for the Medical Decathlon Dataset competition and contains 10 different datasets for different parts of the human body. The data was collected from multiparametric studies. This dataset was used for the training of the models as the ground truth domain and for the creation of the augmented data [18].

4.2 Data preprocessing

Image preprocessing is a fundamental step in the preparation of data for neural network training. This process encompasses a range of techniques designed to enhance the quality and uniformity of raw inputs, which typically suffer from noise, artifacts, and variability. By implementing these preprocessing methods, the resulting dataset becomes more conducive to effective and efficient neural network learning, thereby improving the overall performance and reliability of the model.

Medical images, particularly MRI images, necessitate extensive preprocessing due to their high noise levels and inherently problematic raw form for computational processing. These images are usually examined by professionals who have the expertise to process and interpret their contents without relying on software. For those reasons, it is necessary to take actions for the proper preprocessing of the data as done in [1] and [79].

Beyond the standard image preprocessing steps, which will be detailed later, specific algorithms play a crucial role in medical image processing. Initially, the N3 algorithm will be presented, and subsequently the N4 algorithm, an improved version of the former, which was utilized in this work.

4.2.1 The N3 Algorithm

The Non-parametric Non-uniform intensity Normalization algorithm (N3) is a pivotal method for correcting bias fields in MRI images [80]. These images, obtained using strong magnetic fields (typically 1.5 to 3 Tesla), suffer from non-homogeneous low-frequency noise, commonly referred to as the bias field. This bias field can significantly degrade image quality, making accurate analysis challenging.

The N3 algorithm operates on the premise that the observed image $u(x)$ can be expressed as the product of the true image $v(x)$, a smoothly varying bias field $f(x)$, and an additive Gaussian noise $n(x)$:

$$u(x) = v(x)f(x) + n(x) \tag{4.1}$$

where $u(x)$ is the observed image, $v(x)$ is the true image, $f(x)$ is the bias field, and $n(x)$ is independent Gaussian noise. Assuming that a filter has been applied to remove the white noise from the image:

$$\hat{u}(x) = \hat{v}(x) + \hat{f}(x) \tag{4.2}$$

where

$$\hat{u}(x) = log(u(x)) \tag{4.3}$$

The goal of the N3 algorithm is to iteratively estimate and remove the bias field $f(x)$, thereby restoring the true image $v(x)$ [80].

Retrospectively, the algorithm estimates the bias field using a smoothing function, specifically a B-spline approximation. A "spline" is defined as a function that is piecewise explained by polynomial functions. Splines are used in image smoothing, data interpolation, and the approximation of complex shapes and curve fitting. The connecting points of these polynomial functions are called knots. B-splines form the basis of splines, allowing any spline to be described as a linear combination of multiple B-splines. In practice, splines are a combination of flexible bands controlled by these connecting points.

Using these B-spline functions to approximate the bias field, the image is corrected through the following iterative algorithm:

$$\hat{v}_n = \hat{u} - \hat{f}_e^n = \hat{u} - S\{\hat{u} - E[\hat{u} \mid \hat{u}_{n-1}]\} \tag{4.4}$$

At each iteration, an estimate of the bias field is made using the B-spline approximation of the expected value of the true image, given the estimate from the previous iteration. This estimate is then subtracted from the corrupted image to eventually yield the corrected image. Importantly, the algorithm works solely with the input image and the recursive estimate, requiring no prior knowledge of the field or the image itself [80].

4.2.2 The N4 Algorithm

The N3 algorithm has gained popularity for correcting bias fields due to its proven superiority over other methods and simplicity. However, this popularity led to a stagnation in the development of improved algorithms. To counter this, researchers introduced specific corrections and enhancements to the N3 algorithm, resulting in the N4 algorithm [17]. Significant changes to the recursive process now characterize the N4 algorithm, which operates as follows:

$$\hat{u}_n = \hat{u}_{n-1} - \hat{f}_r^n = \hat{u}_{n-1} - S * \{\hat{u}_{n-1} - E[\hat{u} \parallel \hat{u}_{n-1}]\} \tag{4.5}$$

In experiments, the N4 algorithm demonstrated improved performance with increasing bias field intensity and Gaussian noise [17]. Additionally, reducing the distance between spline functions enhanced its effectiveness. Generally, the N4 algorithm significantly outperforms the N3 algorithm by supporting multi-resolution field approximation and incorporating an improved recursive process. This allows it to calculate noise levels more effectively and address the N3 algorithm's dependence on standard deviation.

One major change in the N4 algorithm is the approximation of the residual noise field, as opposed to the total approximation in the N3 algorithm. Another change involves replacing the distorted image with the approximately corrected one at each step, necessitating the calculation of the residual field as the image undergoes recursive corrections. Furthermore, the updated B-Spline approximator permits smaller distances between control points to handle higher field intensities without risking algorithmic failure. This eliminates the need for an artificial regularization parameter and allows for the definition of a weighted regional mask for iterative segmentation frameworks. Additional advantages include faster execution times due to the parallelization of the B-spline approximation algorithm and a multi-resolution approximation strategy that fits successively higher levels of bias field modulation frequencies hierarchically [17].

Overall, the N4 algorithm offers significant advancements over the N3 algorithm, making it a valuable tool for preprocessing medical images, particularly in correcting bias fields in MRI images.

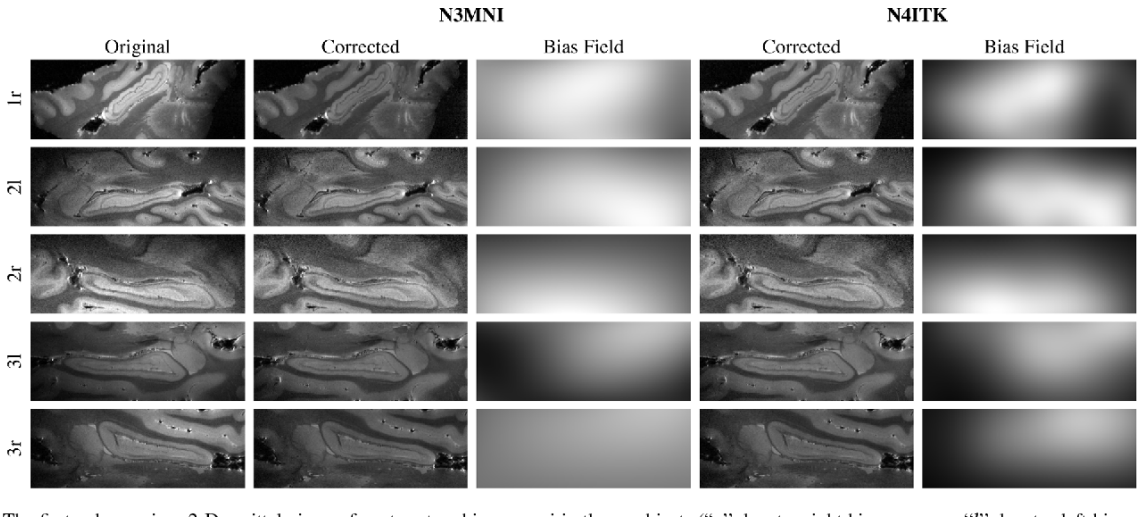

Figure 4.1. *This figure from the original paper visualises the differences between the algorithms N3 and N4 [17]. The first column shows the original MRIs of the postmortem hippocampuses from three different persons. The second and third columns show the corrected samples outputted from the N3MNI algorithm and the detected bias filter respectively. The fourth and fifth columns show the same elements as the second and third columns but outputted from the N4ITK algorithm.*

4.2.3 Rescaling

The image intensities were rescaled according to the following formula:

$$(x - x_2)/(x_{98} - x) \tag{4.6}$$

x_2 and x_{98} represent the 2nd and 98th percentile intensities of each image, respectively, and x denotes the image. This rescaling process helps maintain information and ensures similar intensity levels across images, particularly for MRI images, where intensity holds significant informational value.

4.2.4 Resizing

A standard preprocessing step for images is resizing, particularly prevalent in deep learning applications that handle images. Resizing is essential because it normalizes all images to a consistent size, thereby reducing training costs. Moreover, it has been shown that using lower resolution data can enhance the generalization capability of models. In this work, resizing was executed in three dimensions by altering voxel distances, followed by cropping in two dimensions to achieve a uniform size. For prostate images, the final dimensions were set to $0.627 \times 0.627 \times 3.6\,\mathrm{mm}^3$. The final resolution for the 2-dimentional slices was set to 288×288.

4.2.5 Photometric and Geometric Transformations

To optimize the training process of the networks, photometric filters were applied to the images. These filters restricted certain frequencies while preserving the crucial ones needed for accurate measurements. Moreover, geometric filters were employed to adjust the shape and orientation of the images, ensuring they were formatted correctly for use.

Chapter 5

Implementation

In this chapter are going to presented and analyzed the main methods and techniques used in this body of work.

5.1 Data Augmentation

Data augmentation is a fundamental technique in machine learning and computer vision, particularly for improving the performance and robustness of models [65]. It involves generating new training samples by altering existing data in various ways, such as through rotations, translations, and color adjustments, to create variations that a model might encounter in real-world scenarios. This process is essential for enhancing model generalization, especially when the available dataset is limited or lacks diversity. By simulating a broader range of data conditions, data augmentation helps in mitigating overfitting and enables the model to learn more robust and invariant features. Recent advancements also explore sophisticated methods like adversarial data augmentation, which introduce subtle perturbations to challenge and improve the model's resilience. These techniques are critical for applications in fields like medical imaging, where data is often scarce and expensive to obtain, yet high accuracy and generalization are paramount. In this work, data augmentation methods on the input samples and on the feature levels within model's architecture were used.

5.1.1 Input level augmentation methods

The most prevalent methods of data augmentation involve making simple alterations to the image and its label before they enter the model. The primary aim is to enhance the dataset with variations of the existing images, thereby helping the model extract the correct information for decision-making and increasing its robustness [81],[16],[2]. These techniques range from very simple to quite complex augmentations. Although these methods are straightforward to use and implement, they usually do not result in substantial improvements in model generalization for complex architectures and applications. As a result, more advanced augmentation techniques or feature-level augmentation methods are typically chosen.

Tile Mixing

One of the simpler augmentation methods utilized in this work was augmented data creation through tile mixing. This method involves taking two samples, x_1 and x_2, and splitting them into N rows and M columns, thereby creating $N \times M$ tiles from each sample. Subsequently, the tiles from the two samples x_1 and x_2 were mixed to generate two new samples, $\hat{x_1}$ and $\hat{x_2}$. The mixing process of the tiles involves only the exchange of tiles between the original samples without altering the relative positions of the tiles. This implies that if the input samples x_1 and x_2 are identical images, the output samples $\hat{x_1}$ and $\hat{x_2}$ will also be identical to the input samples.

For the test, $N = M = 3$ was used, thus splitting each input into 9 equal parts. During the augmentation process, the samples were initially divided into three horizontal slices, and then each slice was further divided into three tiles. The number s of tiles to be switched between x_1 and x_2 was uniformly selected between zero and eight. Subsequently, s positions p were selected from zero to nine to determine which tiles were to be switched. A position p corresponded to the tile at $pdiv3$ row and $pmod3$ column.

AugMix

A more complex data augmentation method that was employed on this work is AugMix [2]. AugMix is a data augmentation technique designed to improve the robustness and uncertainty estimates of image classifiers, particularly when dealing with data distribution shifts. The method enhances the diversity of training data through a combination of simple augmentation operations and a consistency loss mechanism. The augmentation process involves applying transformations such as rotations, translations, and other basic image manipulations to create multiple versions of each input image [2]. These transformations are applied stochastically, ensuring a wide variety of augmented images that maintain the semantic content of the original images.

To further improve robustness, AugMix employs a mixing strategy where the augmented images are combined using elementwise convex combinations. This approach generates new images that blend multiple augmentations, thereby increasing the training data's diversity without veering too far from the original data distribution. Additionally, AugMix integrates a Jensen-Shannon Divergence (JSD) consistency loss to enforce that the neural network's predictions remain consistent across different augmented versions of the same image. This consistency loss minimizes the divergence between the probability distributions of the original and augmented images' predictions, ensuring that the model learns features that are robust to various types of corruptions. By combining these techniques, AugMix effectively enhances the model's ability to generalize to unseen data shifts and improves its overall reliability and performance.

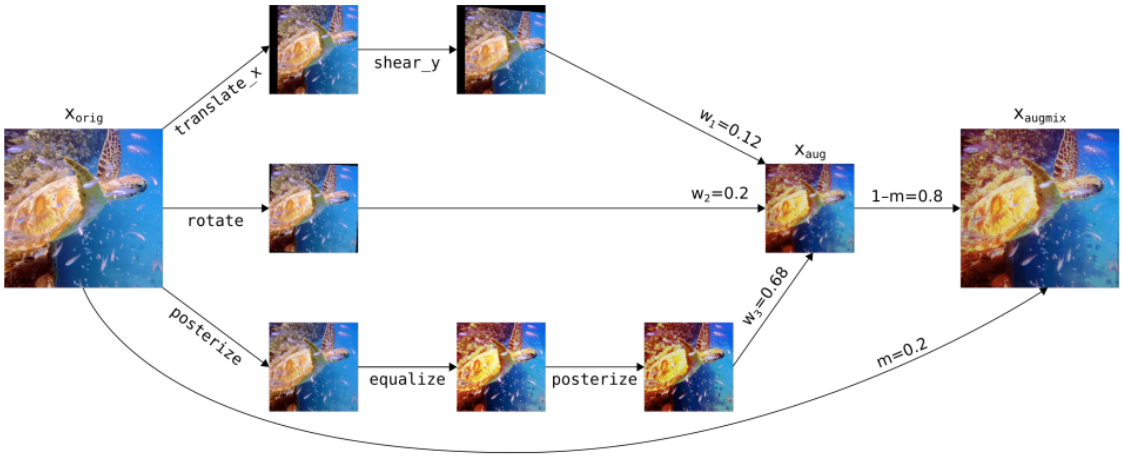

Figure 5.1. *In this figure from the original paper, is presented a visualization of the augmentation process of AugMix [2]. Augmentation operators such as translate_x and rotate and weights such as m are randomly sampled. Those randomly selected operators allow to explore the semantically equivalent input space around an image. Mixing these images together produces a new image without veering too far from the original [2].*

5.1.2 Feature level augmentation methods

MaxStyle

The mehtod that was used for feature level augmentation is MaxStyle, a compicated implematation that invovles style augmentationin the feature level as well as adversarial training [1].

The MaxStyle method represents a significant advancement in data augmentation techniques aimed at enhancing the robustness of convolutional neural networks (CNNs) for medical image segmentation tasks, particularly when facing out-of-domain (OOD) datasets [1]. Unlike conventional methods that often require multi-domain datasets, MaxStyle improves model robustness using only a single-domain dataset. This method was proposed to tackle the challenges posed by domain shifts which can significantly degrade the performance of CNNs when applied to unseen domains.

MaxStyle enhances the standard encoder-decoder architecture by integrating an auxiliary image decoder. This decoder performs self-supervised image reconstruction and style augmentation, which not only augments the training data but also forces the network to learn robust, reconstructive features that contribute to improved OOD performance[1].

The core innovation of MaxStyle lies in its ability to expand the style space of training images through adversarial style augmentation. This is achieved by introducing additional style noise and conducting adversarial training to identify the worst-case style compositions that could potentially degrade model performance. Formally, given a feature map f_i extracted from a certain layer of the CNN, MaxStyle augments f_i by mixing styles and adding noise:

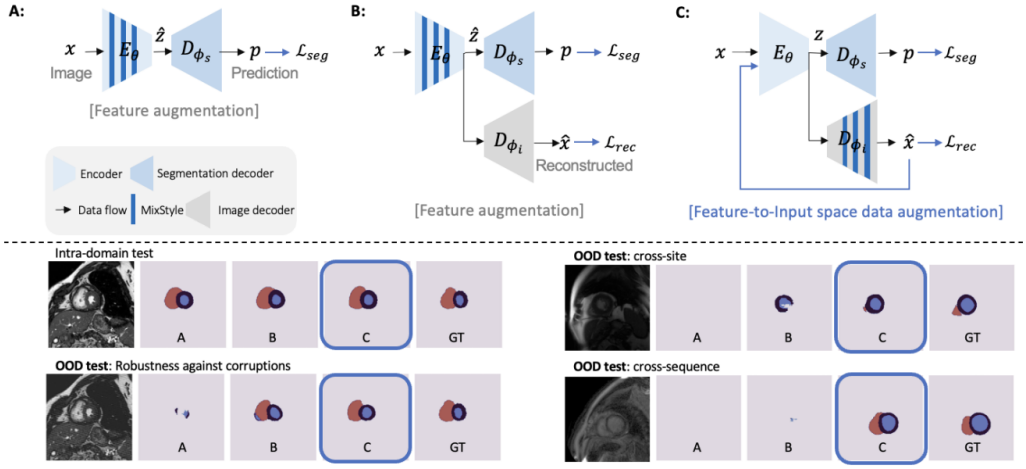

Figure 5.2. *In the first to columns (A,B) it is depicted the original use of MixStyle [15] as featur augmentation-based regularization method with a standard encoder-decoder structure (A) and aplide to a regularize a dual-branch network with an auxiliary image decoder. In the third collumn (C) it is depicted the proposed stracture changes to MixStyle by the creators of MaxStyle. They proposed the addition of an auxiliary decoder for the generation of stylized images for feature-to-point space data augmentation. As seen in the examples on the botom of the figure, the proposed model outperformed the first two. [1]*

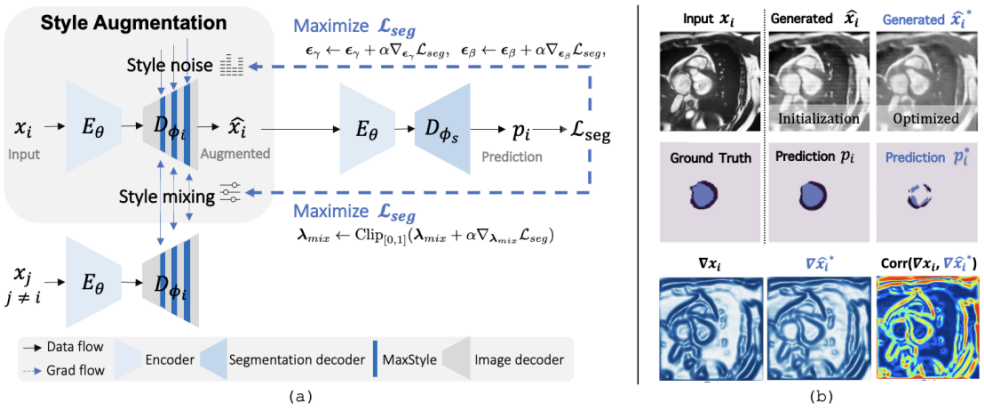

Figure 5.3. *On the left (a) is the architecture of MaxStyle [1]. MaxStyle reconstructs the input with augmented deature styles via style mixing and noise perturbations in the image decoder. In order to find 'harder' style compositions, the authors applied adversarial training. On the right (b) it is shown that MaxStyle generates samples with high corelation to to original but able to fool the network to undersegment.*

$$\text{MaxStyle}(f_i) = (\gamma_{\text{mix}} + \Sigma_\gamma \epsilon_\gamma) \odot \bar{f}_i + (\beta_{\text{mix}} + \Sigma_\beta \epsilon_\beta),$$

where $\bar{f}_i$ represents the normalized feature map, γ_{mix} and β_{mix} are the mixed style statistics, and $\Sigma_\gamma \epsilon_\gamma$ and $\Sigma_\beta \epsilon_\beta$ denote the additional style noise sampled from a re-scaled Gaussian distribution.

The adversarial training process involves optimizing the style noise and mixing coefficients

to maximize the segmentation loss L_{seg}, thereby generating style-augmented images that challenge the segmentation network [1]. This adversarial optimization is formalized as follows:

$$\epsilon_\gamma \leftarrow \epsilon_\gamma + a\nabla_{\epsilon_\gamma} L_{\text{seg}}(\hat{p}, y), \quad \epsilon_\beta \leftarrow \epsilon_\beta + a\nabla_{\epsilon_\beta} L_{\text{seg}}(\hat{p}, y),$$

$$\lambda_{\text{mix}} \leftarrow \text{Clip}[0, 1](\lambda_{\text{mix}} + a\nabla_{\lambda_{\text{mix}}} L_{\text{seg}}(\hat{p}, y)),$$

where a is the step size for the gradient ascent.

Extensive experiments on public cardiac and prostate MR datasets have demonstrated that MaxStyle significantly improves OOD robustness against various unseen corruptions and distribution shifts [1] [79]. Notably, MaxStyle outperformed several competitive methods in both low-data and high-data training regimes, underscoring its efficacy and generalizability.

In conclusion, MaxStyle provides a powerful and efficient solution for enhancing the robustness of medical image segmentation models, making them more reliable for real-world clinical applications. The integration of adversarial style augmentation into the training process allows for a broader exploration of the style space, resulting in more robust feature learning and improved model generalization.

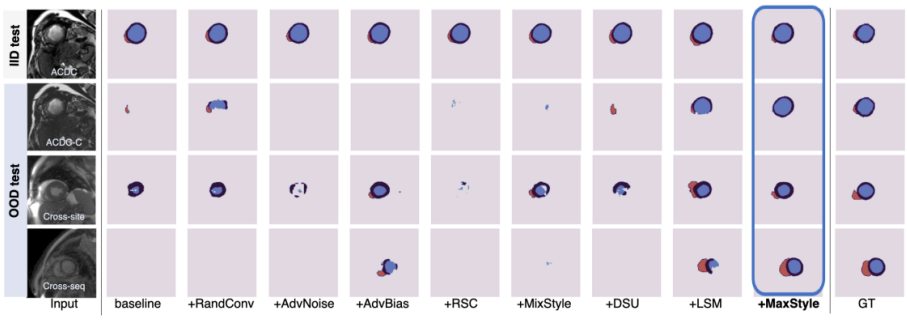

Figure 5.4. *This figure from the original paper visualises qualitative resaults from MaxStyle compared to other methods [1]. As seen in the figure the MaxStyle not only outperforms all other methods but the resaults are vary accurate, compared to the the graund-truth (GT)*

5.2 Experiment Set-up

The objective of this study is to determine the effect of data augmentation on the input and feature layers and how it impacts the generalization ability of image segmentation transformers. Specifically, combinations of the methods described in Section 5.1 will be evaluated on medical images, with a particular focus on prostate MRIs.

THe experimental process consists of two main parts. In the first part, the datasets will be augmented using different combinations of the presented methods. The primary method employed is the one proposed in MaxStyle [1], as all tests, except for the ground truth, include images generated using this method. The second part of the experiment

involves training an image segmentation transformer and testing it on images from unseen domains. During the training phase, the Medical Decathlon dataset [18] will be used as the IID domain. For testing, the other six datasets mentioned in Section 4.1 will be used as unseen domains (OOD).

Chapter 6

Presentation and Analysis of the Results

In this chapter, the results from the experiments will be presented and analyzed. Before that, critical information about the testing process will be mentioned.

6.1 Avaluation Metrics

Dice Coefficient

During the training and testing of the models, the Dice coefficient was chousen as the primary evaluation metric [82], [83]. The Dice coefficient, also known as the Sørensen–Dice coefficient or Sørensen index, is a formula that measures the similarity between two different discrete data points. In the context of images, it quantifies the similarity between two samples at the pixel level. The Dice coefficient of two samples is defined as twice the number of elements common to both sets divided by the sum of the number of elements in each set:

$$\mathrm{DSC} = \frac{2|X \cap Y|}{|X| + |Y|}$$

where $|X|$ and $|Y|$ are the cardinalities of the two sets (in this case, the number of pixels in each set). The segmented images contain only two classes: prostate and background. Therefore, the Dice coefficient was used solely for the prostate class.

6.2 Experiment Results

For the test, the pretrained SegFormer model was used [3]. The pretrained transformer was fine-tuned for 10 epochs, but in most cases, the best resulting model was selected from an epoch before the final one. In all processes, the Adam optimizer was used [84]. The following tables show the results from the tests.

Datasets	**IID**	**OOD**						**Average**
	G	A	B	C	D	E	F	**OOD**
Ground Truth	0.862	**0.811**	**0.842**	0.815	**0.876**	0.684	**0.611**	**0.773**
MixStyle 200%	0.963	0.682	0.688	0.799	0.687	**0.689**	0.599	0.691
MixStyle 100%	0.978	0.750	0.722	0.822	0.750	0.682	0.605	0.722
MixStyle 50%	**0.981**	0.722	0.728	0.817	0.704	0.712	0.608	0.715
MixStyle 20%	0.978	0.730	0.793	**0.837**	0.750	0.683	0.607	0.734

Table 6.1. *This table presents the results of the initial round of tests. The original dataset from the Medical Decathlon (**G**) [18] was augmented using the **MaxStyle** method [1], thereby increasing the dataset size by 200%, 100%, 50%, and 20%. Subsequently, the pretrained SegFormer model [3] was fine-tuned for up to 10 epochs, selecting the best performing stage. Finally, the fine-tuned model was evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The best performance for each dataset is highlighted in bold.*

Datasets	**IID**	**OOD**						**Average**
	G	A	B	C	D	E	F	**OOD**
Ground Truth	0.862	**0.812**	0.842	0.815	**0.876**	0.684	0.611	**0.773**
Tiles 100%	**0.981**	0.667	0.665	0.793	0.712	0.681	0.598	0.686
Tiles 50%	0.977	0.726	0.796	0.844	0.756	**0.716**	0.615	0.742
Tiles 20%	0.979	0.756	**0.875**	**0.869**	0.760	0.687	**0.626**	0.762

Table 6.2. *This table presents the results of the second round of tests. The original dataset from the Medical Decathlon (**G**) [18] was augmented using the **MaxStyle** method [1] followed by the **Tile Mixing** method. The additional samples were generated by tile mixing between the original samples and their corresponding MaxStyle-augmented samples. This approach increased the number of samples in the dataset by 100%, 50%, and 20%. The augmented dataset was then used to fine-tune a SegFormer model for 10 epochs. The fine-tuned model was evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The results indicate an increase in the model's generalization ability. The best performance for each dataset is highlighted in bold.*

Datasets	**IID**	**OOD**						**Average**
	G	A	B	C	D	E	F	**OOD**
Ground Truth	0.862	**0.812**	0.842	0.815	**0.876**	0.684	0.611	0.773
AugMix 50%	0.968	0.781	0.834	0.861	0.823	0.675	0.615	0.765
AugMix 20%	0.967	0.799	0.830	0.837	0.816	0.686	0.606	0.762
AugMix 10%	**0.975**	0.779	**0.868**	**0.876**	0.848	**0.688**	**0.673**	**0.789**

Table 6.3. *This table presents the final and most successful round of tests. The same steps as in the second round were followed, but this time the **MaxStyle** method [1] was combined with the **AugMix** method [2]. First, the original dataset was augmented with MaxStyle, increasing the number of samples by 20%. Then, during the fine-tuning of the model, 50%, 20%, and 10% of the input were augmented with the AugMix method. As in the previous tests, the augmented dataset was used to fine-tune a SegFormer model for 10 epochs. The fine-tuned model was then evaluated on the remaining datasets mentioned in Section 4.1: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK. The results show an increase in generalizability to out-of-distribution (OOD) samples. The best performance for each dataset is highlighted in bold.*

Part III

Epilogue

Chapter 7

Epilogue

7.1 Conclusions

In this thesis, the significant challenge of domain generalization in medical image segmentation was tackled, focusing on the robustness and adaptability of transformers for prostate MRIs. Domain generalization is critical in medical imaging as models often encounter variations in data distribution due to differences in imaging devices, patient demographics, and environmental conditions. This study specifically explored advanced data augmentation techniques based on style transfer to address this issue.

Through extensive experimentation and analysis, it was demonstrated that integrating complex augmentation methods, such as MaxStyle and AugMix, enhances model performance, particularly in out-of-distribution (OOD) scenarios. MaxStyle combines adversarial training with style transfer, effectively broadening the style space explored during training, leading to improved robustness against various unseen corruptions and distribution shifts. Models trained with MaxStyle exhibited superior generalization capabilities compared to those trained with traditional methods.

Furthermore, by incorporating AugMix, additional complexity was introduced during training without compromising the semantic integrity of the medical images. This method provided a diverse yet realistic augmentation strategy that further bolstered the model's ability to handle unseen data variations. The combined use of MaxStyle and AugMix resulted in the highest generalization performance, as evidenced by the evaluation metrics and comparative analysis.

Overall, this work highlights the importance of sophisticated data augmentation techniques in developing robust and reliable medical image segmentation models. These findings are particularly relevant for clinical applications where data variability and scarcity pose significant challenges. By leveraging advanced augmentation strategies, models can be created that not only perform well on known datasets but also maintain high accuracy in real-world, unpredictable environments.

7.2 Future Work

In this work, not only the concept of domain generalization through data augmentation based on style transfer was further explored, but also a fully functional pipeline for testing

the developed theories. This research can be advanced in two main ways.

Firstly, the combination of additional techniques for achieving data augmentation based on style could enhance the generalizability of downstream models. Studies like ours and those by Spanos et al. have only scratched the surface of the numerous possibilities in this field. Additionally, with the rapid advancement in the field of neural networks and the increasing importance of robust applications in real-world scenarios, new methods for data augmentation and domain generalization enhancement are being published daily. As demonstrated in this thesis, the appropriate combination of these methods can significantly improve the capabilities of neural networks.

Another way this work could be advanced is by testing the proposed theory in other computer vision tasks, such as automated driving. As shown, the results of the executed tests are promising. The generalization ability of the backbone model was successfully increased by properly augmenting the single-domain dataset of prostate MRIs. This indicates that the approach of this thesis, with appropriate modifications, could also enhance the generalization ability of models used in various other computer vision tasks.

Appendices

Appendix A

Experiments

In Appendix A, all the different tests conducted within the scope of this work are presented and analyzed. The data are presented in easy-to-understand Tables A.1 - A.6. In Table A.1, the process for all 20 different tests is presented and explained. In Table A.2, the training parameters for each of those tests as well as the results of the models during training are presented. Tables A.3 - A.5 depict the results of each test during the testing phase on out-of-distribution datasets. More specifically, in Table A.3, the loss of the models trained on each test is shown. The loss is a smooth Dice coefficient plus Cross-entropy. This loss was used as it helped the model during training but was ultimately not the final evaluation criterion. In Table A.4, the F1 score achieved by the models trained on OOD data is presented. Table A.5 depicts the Dice coefficient of the models. As mentioned in the main body of this thesis, this was the final evaluation metric. Finally, in Table A.6, the average of the three metrics shown in Tables A.3 - A.5, on all out-of-distribution datasets is presented.

A.1 Experiment and Resault Presentation

A.1.1 Test Setup Analysis

In Table A.1, all 20 different tests executed within the scope of this work are presented. Test 1 consists of training the model on the original data from the Medical Decathlon [18] dataset without any augmentation. This model is used as the ground truth.

Tests 2 - 5

For Tests 2, 5-7, two different augmentations were used, both generated with MaxStyle [1], one altering layers 2, 3, 4, and 5 for one iteration, and the second altering layers 3, 4, and 5 for one iteration. In all of those tests, those two augmentations contributed 50% each to the augmented data. In Test 2, the generated data were 200% the size of the original data (100% each type of augmentation). In Test 3, the generated data were 100% the size of the original data (50% each type of augmentation). In Test 4, the generated data were 50% the size of the original data (25% each type of augmentation). In Test 5, the generated data were 20% the size of the original data (10% each type of augmentation).

Tests 6 - 8

In Test 6, augmented the original dataset using only one of the two methods mentioned above combined with the tile mixing method (Patches). All the dataset used for this test was created using tile mixing, combining original samples with samples generated with MaxStyle, altering layers 2, 3, 4, and 5 for one iteration. This means that no original samples were used during the training of the model. For Test 7, the same process was followed, using patches that combined original samples with samples generated with MaxStyle, altering layers 3, 4, and 5 for one iteration. The dataset for Test 8 was created as before, but both types of samples generated by MaxStyle contributed equally in the augmentation with tile mixing.

Tests 9 - 11

For Tests 9 through 11, the dataset used for training the model was created by adding to the original Medical Decathlon dataset, samples created by tile mixing. The generated samples were created by mixing original data with MaxStyle generated data. Each of the two types of MaxStyle generated data were used to augment 50% of the final augmented data. The training dataset for Test 9 included all the original data from the Medical Decathlon dataset and augmented data generated by tile mixing. The number of samples generated by tile mixing was equal to the number of original samples, thus increasing the size of the original dataset by 100%. Test 10 was performed with the same process, but the generated data increased the original dataset by 50%. Finally, for Test 11, the original dataset's size was increased by 20% using the same tile mixing process as above.

Tests 12 - 15

At this point, the implementation of the AugMix augmentation process into the augmentation pipeline began [2]. For tests 12 through 15, AugMix was implemented with the default settings (severity=3, width=3, depth=-1, alpha=1) [2]. According to the authors of the original paper [2]:

- **severity**: Severity of underlying augmentation operators (between 1 to 10).
- **width**: Width of augmentation chain
- **depth**: Depth of augmentation chain. -1 enables stochastic depth uniformly from [1, 3]
- **alpha**: Probability coefficient for Beta and Dirichlet distributions.

The AugMix operations were implemented on the dataloader part of the pipeline. This implementation ensured that all samples of the dataset had an independent probability of being augmented with AugMix. It is important to note that the samples augmented with AugMix were different in each epoch of the training process. This increased the generalization ability of the model without increasing the necessary training time. In three tests (12, 13, 14), the dataset used, before the AugMix augmentation, consisted

of the original Medical Decathlon dataset plus an additional 20% of its size in samples augmented with MaxStyle using both types of augmentations (altering layers 2, 3, 4, 5 and altering layers 3, 4, 5). In test 12, AugMix-type augmentations were performed on 50% of the data inputted to the model during training. For test number 13, the percentage of samples that were augmented with AugMix before being inputted to the model for training was lowered to 20%. During test 14, that percentage was further lowered to 10%. In test 15, the same process as in the above tests was followed, but the size of the original dataset was increased by only 10% augmented samples by MaxStyle, compared to the 20% increase in tests 12, 13, and 14. The percentage of samples augmented by AugMix during the loading from the data loader was 10%.

Tests 16 - 19

In Tests 16 to 19, the same pipeline as in tests 12 to 15 was maintained, and further experimentation with the hyperparameters of AugMix was conducted. During all four tests, the percentage of input samples augmented with AugMix was set to 10%. For tests 16 and 17, the severity of the augmentations from AugMix was increased from 3 (the default) to 5. In test 16, the percentage of samples created with MaxStyle was 10% of the original dataset, while in test 17, the samples added to the original dataset increased its size by 20%. Test 18 was conducted with the severity of AugMix set to 2, a lower number than the standard 3, which means that the augmentations performed by AugMix were milder. The dataset for test 18 was increased by 20% of its original size by including augmented data generated by MaxStyle. Finally, for test 19, the same dataset as in tests 18 and 17 was used, but the severity of AugMix augmentations was increased to 6.

Test 20

In test 20, the same setup as in test 14 was maintained and experimented with the value that was monitored for the early-stop function, changing it from the F1 score during validation to the loss during validation.

	Traind on
Test 1	Original
Test 2	Augmented 200% (100% each type)
Test 3	Augmented 100% (50% each type)
Test 4	Augmented 50% (25% each type)
Test 5	Augmented 20% (10% each type)
Test 6	Tiles (2345 only tiles)
Test 7	Tiles (345 only tiles)
Test 8	Tiles (50% each type)
Test 9	Og + Tiles (50% each type)
Test 10	Og + 50% Tiles (25% each type)
Test 11	Og + 20% Tiles (10% each type)
Test 12	AugMix 50% (OG + 20% augmented)(default settings)
Test 13	AugMix 20% (OG + 20% augmented)(default settings)
Test 14	AugMix 10% (OG + 20% augmented)(default settings)
Test 15	AugMix 10% (OG + 10% augmented)(default settings)
Test 16	AugMix 10% (OG + 10% augmented)(severity=5)
Test 17	AugMix 10% (OG + 20% augmented)(severity=5)
Test 18	AugMix 10% (OG + 20% augmented)(severity=2)
Test 19	AugMix 10% (OG + 20% augmented)(severity=6)
Test 20	AugMix 10% (OG + 20% augmented)(default settings)(monitor="valid/loss")

Table A.1. *This table presents all the different tests conducted within the scope of this work. The first column shows the number of each test, which is used to refer to each specific test in the following tables and descriptions. The second column provides a coded description of each test. The descriptions focus on the dataset that the model of the test was fine-tuned on and the creation of that dataset. Information about the model, such as hyperparameters, is presented in Table A.2.*

A.1.2 Test Hyper-parameters Analysis

For all 20 tests, the same hyperparameters shown in Table A.2 were used.

- **Batch Size**: Batch size was set to 8.
- **Learning Rate**: Learning rate was kept at the standard 3e – 4.
- **Weight Decay**: Weight decay was kept at the standard 1e – 4.
- **Total Epochs**: The number of total epochs was kept to 10, as it was concluded that in most cases, that was enough for the model to fine-tune properly.

The Best Epoch field corresponds to the epoch in which each model performed best. This field is not the same in every test as it is not a hyperparameter set before the test, but rather an outcome influenced by the learning ability and the overfitting of the model.

Table A.2 also demonstrates the performance of each model in its best epoch. Specifically, it depicts the Validation Loss (V_Loss), the Validation F1 score (V_f1), and the Validation Dice coefficient (V_dice).

An important observation is that the validation Dice score of test 20 is greater than he validation Dice score of test 14. This means that the change of the monitored metric

from validation loss to validation F1 score increased the segmentation ability of the model. This increase was not transferred to the generalization ability of the model as model 20 has a lower average Dice score on OOD datasets than test 14.

	Batch size	**Learning rate**	**Weight decay**	**Total Epochs**	**Best Epoch**	**V_loss**	**V_f1**	**V_dice**
Test 1	8	3e-4	1e-4	10	8	0.123	0.944	0.862
Test 2	8	3e-4	1e-4	10	7	0.168	0.926	0.932
Test 3	8	3e-4	1e-4	10	6	0.141	0.940	0.872
Test 4	8	3e-4	1e-4	10	6	0.138	0.939	0.872
Test 5	8	3e-4	1e-4	10	8	0.140	0.938	0.860
Test 6	8	3e-4	1e-4	10	5	0.132	0.937	0.865
Test 7	8	3e-4	1e-4	10	5	0.119	0.943	0.881
Test 8	8	3e-4	1e-4	10	9	0.139	0.936	0.870
Test 9	8	3e-4	1e-4	10	9	0.163	0.938	0.868
Test 10	8	3e-4	1e-4	10	6	0.128	0.939	0.880
Test 11	8	3e-4	1e-4	10	5	0.142	0.935	0.883
Test 12	8	3e-4	1e-4	10	7	0.144	0.930	0.839
Test 13	8	3e-4	1e-4	10	5	0.141	0.933	0.817
Test 14	8	3e-4	1e-4	10	7	0.138	0.936	0.822
Test 15	8	3e-4	1e-4	10	8	0.142	0.933	0.852
Test 16	8	3e-4	1e-4	10	7	0.133	0.936	0.857
Test 17	8	3e-4	1e-4	10	7	0.146	0.927	0.849
Test 18	8	3e-4	1e-4	10	7	0.152	0.930	0.873
Test 19	8	3e-4	1e-4	10	7	0.137	0.935	0.851
Test 20	8	3e-4	1e-4	10	3	0.118	0.946	0.851

Table A.2. *This table presents important information about the training of the models for each test. As mentioned in section 6.2, the SegFormer model was used in all experiments [3]. The first column shows the number of each test (for reference, see Table A.1 and section A.1.1). The second column indicates the batch size used during the training of each model. The learning rate and weight decay chosen for each test are depicted in the third and fourth columns, respectively. The fifth column shows the total number of epochs the model was fine-tuned. The Best Epoch column shows the epoch after which the model achieved the best performance on the validation set. The last three columns correspond to the validation loss, the validation F1 score, and the validation Dice score each model achieved after its best epoch. It is interesting to note that the best epoch numbers vary significantly between different tests.*

A.1.3 Test Resaults Pesentation

In Tables A.3, A.4, and A.5, the results of the tests are presented. The models were tested on the original Medical Decathlon dataset [18] as the IID and ground truth. Additionally, to test the generalization ability of the models, they were tested on the remaining datasets mentioned in section 4.1 as out-of-distribution datasets: **A**: ISBI, **B**: ISBI_1.5, **C**: I2CVB, **D**: UCL, **E**: BIDMC, and **F**: HK.

Finally, in Table A.6, the average of each metric that each model achieved on the six OOD datasets is presented. As mentioned in section 6.2, the best-performing models were achieved in tests 14 and 17.

Loss	**G**	**A**	**B**	**C**	**D**	**E**	**F**
Test 1	-	0.407	0.538	0.613	0.429	1.142	1.046
Test 2	0.062	0.740	1.164	0.735	0.736	1.437	1.332
Test 3	0.066	0.478	0.933	0.453	0.507	1.081	1.072
Test 4	0.065	0.505	0.942	0.444	0.628	1.057	1.034
Test 5	0.053	0.465	0.639	0.429	0.477	0.987	0.934
Test 6	0.070	0.426	0.826	0.457	0.522	0.994	0.842
Test 7	0.075	0.427	0.714	0.575	0.527	0.962	0.828
Test 8	0.060	0.571	0.789	0.606	0.645	0.918	0.897
Test 9	0.047	0.725	1.107	0.627	0.814	1.252	1.362
Test 10	0.056	0.483	0.673	0.452	0.494	0.963	0.918
Test 11	0.062	0.416	0.523	0.356	0.400	0.982	0.878
Test 12	0.074	0.360	0.444	0.288	0.371	0.698	0.744
Test 13	0.066	0.346	0.536	0.466	0.346	0.847	0.735
Test 14	0.072	0.385	0.440	0.363	0.368	0.749	0.689
Test 15	0.061	0.441	0.610	0.364	0.450	0.976	0.976
Test 16	0.066	0.443	0.601	0.420	0.395	0.795	0.752
Test 17	0.065	0.370	0.462	0.354	0.389	0.721	0.709
Test 18	0.057	0.494	0.817	0.502	0.461	0.996	0.897
Test 19	0.066	0.380	0.449	0.335	0.355	0.806	0.733
Test 20	0.070	0.591	0.705	0.484	0.501	0.829	0.872

Table A.3. *In this table, the loss of the fine-tuned model of each test is depicted. The first column shows the corresponding number of each test (for reference, see Table A.1 and section A.1.1). The second column shows the performance of the model on the IID ground truth dataset, while the rest of the columns show the model's performance on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations. The loss of all models is significantly lower on dataset G, which is expected as they all were trained on an augmented dataset based on dataset G. Another meaningful observation is that in the first tests, the performance of the models was worse than in the following tests, as clearly depicted by the losses in the last couple of columns. This can be attributed to the increase in complexity and sophistication of the augmentation techniques and combinations as the testing process progressed.*

F1 score	G	A	B	C	D	E	F
Test 1	-	0.838	0.721	0.720	0.779	0.513	0.550
Test 2	0.975	0.777	0.603	0.717	0.697	0.516	0.539
Test 3	0.971	0.824	0.611	0.777	0.762	0.526	0.571
Test 4	0.971	0.803	0.561	0.785	0.684	0.511	0.541
Test 5	0.976	0.782	0.638	0.770	0.755	0.527	0.538
Test 6	0.968	0.824	0.577	0.762	0.704	0.512	0.547
Test 7	0.966	0.815	0.608	0.708	0.704	0.520	0.566
Test 8	0.973	0.734	0.539	0.693	0.659	0.511	0.541
Test 9	0.980	0.747	0.571	0.751	0.676	0.545	0.528
Test 10	0.975	0.760	0.599	0.750	0.719	0.514	0.542
Test 11	0.972	0.793	0.698	0.804	0.771	0.519	0.565
Test 12	0.967	0.809	0.725	0.832	0.783	0.589	0.587
Test 13	0.970	0.826	0.681	0.762	0.803	0.563	0.611
Test 14	0.969	0.812	0.737	0.803	0.797	0.601	0.608
Test 15	0.973	0.790	0.630	0.798	0.754	0.497	0.526
Test 16	0.970	0.786	0.675	0.789	0.788	0.569	0.623
Test 17	0.971	0.819	0.716	0.795	0.779	0.565	0.582
Test 18	0.974	0.755	0.552	0.730	0.738	0.510	0.538
Test 19	0.971	0.816	0.743	0.816	0.804	0.560	0.609
Test 20	0.969	0.684	0.571	0.728	0.697	0.508	0.504

Table A.4. *This table presents the F1 score achieved by the different fine-tuned models. As in Table A.3, the first column refers to the number of each test (for reference, see Table A.1 and section A.1.1), the second column shows the F1 score of the models on the ground truth dataset, and the final six columns show the F1 score of each model on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations.*

Dice	G	A	B	C	D	E	F
Test 1	-	0.812	0.842	0.815	0.876	0.684	0.611
Test 2	0.963	0.682	0.688	0.799	0.687	0.689	0.599
Test 3	0.978	0.750	0.722	0.822	0.750	0.682	0.605
Test 4	0.981	0.722	0.728	0.817	0.704	0.712	0.608
Test 5	0.978	0.730	0.793	0.837	0.750	0.683	0.607
Test 6	0.978	0.727	0.753	0.845	0.797	0.693	0.606
Test 7	0.976	0.736	0.740	0.789	0.760	0.690	0.609
Test 8	0.975	0.778	0.809	0.781	0.729	0.701	0.624
Test 9	0.981	0.667	0.665	0.794	0.712	0.681	0.598
Test 10	0.977	0.726	0.796	0.844	0.756	0.716	0.615
Test 11	0.979	0.756	0.875	0.869	0.760	0.687	0.626
Test 12	0.968	0.781	0.834	0.861	0.823	0.675	0.615
Test 13	0.967	0.799	0.831	0.837	0.817	0.686	0.606
Test 14	0.975	0.779	0.868	0.876	0.848	0.688	0.673
Test 15	0.977	0.745	0.793	0.863	0.753	0.690	0.606
Test 16	0.978	0.748	0.822	0.836	0.840	0.703	0.610
Test 17	0.978	0.833	0.879	0.839	0.869	0.694	0.619
Test 18	0.977	0.743	0.736	0.818	0.786	0.688	0.616
Test 19	0.974	0.815	0.860	0.857	0.812	0.660	0.620
Test 20	0.971	0.726	0.778	0.805	0.750	0.662	0.605

Table A.5. *This table depicts the Dice coefficient that the models achieved in each of the 20 experiments. This is one of the most important parts of this work. The Dice coefficient was the metric aimed to increase, especially in OOD samples and datasets. Following the format of Tables A.3 and A.4, the number of each test is shown in the first column (for reference, see Table A.1 and section A.1.1), the second column shows the Dice coefficient of the models on the ground truth dataset, and the final six columns show the Dice coefficient of each model on the six OOD datasets. In test 1, the loss of the model on dataset G is left blank because it is the same as the loss during training, as this specific model (ground truth model) was trained on the original dataset G without any augmentations. An important observation is that even though tests 14 and 17 were, on average, the best-performing models on OOD datasets, in certain datasets they were outperformed by other models.*

Average	Loss	F1	Dice
Test 1	0.696	0.687	0.773
Test 2	1.024	0.642	0.691
Test 3	0.754	0.678	0.722
Test 4	0.768	0.647	0.715
Test 5	0.655	0.668	0.734
Test 6	0.678	0.654	0.737
Test 7	0.672	0.654	0.721
Test 8	0.738	0.613	0.737
Test 9	0.981	0.636	0.686
Test 10	0.664	0.647	0.742
Test 11	0.593	0.692	0.762
Test 12	0.484	0.721	0.765
Test 13	0.546	0.708	0.763
Test 14	0.499	0.726	0.789
Test 15	0.636	0.666	0.742
Test 16	0.568	0.705	0.760
Test 17	0.501	0.709	0.789
Test 18	0.694	0.637	0.731
Test 19	0.510	0.725	0.771
Test 20	0.664	0.615	0.721

Table A.6. *This table summarizes all the previous ones by presenting the average of each of the three metrics (Loss, F1 score, and Dice coefficient) that the models achieved on OOD datasets. According to this table and by focusing on the Dice coefficient, the best-performing models were created in tests 14 and 17, achieving a Dice coefficient equal to 0.789.*

Bibliography

[1] Chen Chen, Zeju Li, Cheng Ouyang, Matt Sinclair, Wenjia Bai και Daniel Rueckert. *MaxStyle: Adversarial Style Composition for Robust Medical Image Segmentation*, 2022.

[2] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer και Balaji Lakshminarayanan. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*, 2020.

[3] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José M. Álvarez και Ping Luo. *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers*. *CoRR*, abs/2105.15203, 2021.

[4] *Introduction to Convolutional Neural Networks CNNs*. https://aigents.co/data-science-blog/publication/introduction-to-convolutional-neural-networks-cnns. Access date:8-5-2024.

[5] Andrew Zisserman Karen Simonyan. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *International Conference on Learning Representations*, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep Residual Learning for Image Recognition*. *arXiv preprint arXiv:1512.03385*, 2015.

[7] Trevor Darrell Jonathan Long, Evan Shelhamer. *Fully Convolutional Networks for Semantic Segmentation*. *Computer Vision and Pattern Recognition*, 2014.

[8] Thomas Brox Olaf Ronneberger, Philipp Fischer. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *Computer Vision and Pattern Recognition*, 2015.

[9] https://raw.githubusercontent.com/snatch59/keras-autoencoders/master/assets/autoencoder_latent_space.png. Access date: 3-6-2024.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit και Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. *CoRR*, abs/2010.11929, 2020.

[11] Liang Chieh Chen, George Papandreou, Florian Schroff και Hartwig Adam. *Rethinking Atrous Convolution for Semantic Image Segmentation*, 2017.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár και Ross B. Girshick. *Mask R-CNN*. *CoRR*, abs/1703.06870, 2017.

[13] Leon A. Gatys, Alexander S. Ecker και Matthias Bethge. *A Neural Algorithm of Artistic Style*. *CoRR*, abs/1508.06576, 2015.

[14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville και Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[15] Kaiyang Zhou, Yongxin Yang, Yu Qiao και Tao Xiang. *Domain Generalization with MixStyle*. *CoRR*, abs/2104.02008, 2021.

[16] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin και David Lopez-Paz. *mixup: Beyond Empirical Risk Minimization*. *CoRR*, abs/1710.09412, 2017.

[17] N. Tustison, Brian B. Avants, Philip A. Cook, Yuanjie Zheng, Alexander Egan, Paul Yushkevich και James C. Gee. *N4ITK: Improved N3 Bias Correction*. *IEEE Transactions on Medical Imaging*, 29:1310–1320, 2010.

[18] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bramvan Ginneken, Michel Bilello, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc J. Gollub, Stephan H. Heckers, Henkjan Huisman, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Jennifer S. Golia Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, James A. Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbeláez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee, Yuanfeng Ji, Fucang Jia, Ildoo Kim, Klaus Maier-Hein, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaiifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, Yan Wang, Yingda Xia, Daguang Xu, Zhanwei Xu, Yefeng Zheng, Amber L. Simpson, Lena Maier-Hein και M. Jorge Cardoso. *The Medical Segmentation Decathlon*. *Nature Communications*, 13(1), 2022.

[19] McCulloch S Warren. και Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. *Bulletin of Mathematical Biophysics*, (5):115–133.

[20] Geoffrey E Hinton Alex Krizhevsky, Ilya Sutskever. *ImageNet Classification with Deep Convolutional Neural Networks*. *Advances in Neural Information Processing Systems*, 25, 2012.

[21] Vivienne Sze, Yu Hsin Chen, Tien Ju Yang και Joel S. Emer. *Efficient Processing of Deep Neural Networks: A Tutorial and Survey*. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

[22] Christian Szegedy, Yangqing Jia Wei Liu, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke και Andrew Rabinovich. *Going Deeper with Convolutions*. *arXiv preprint arXiv:1409.4842*, 2014.

[23] Yading Yuan. *Automatic skin lesion segmentation with fully convolutional-deconvolutional networks*. *IEEE Journal of Biomedical and Health Informatics*, 2018.

[24] Chunyang Feng, Yufeng Sun και Xin Li. *iris R-CNN: Accurate Iris Segmentation in Non-cooperative Environment*. *CoRR*, abs/1903.10140, 2019.

[25] Guotai Wang, Wenqi Li, Sébastien Ourselin και Tom Vercauteren. *Automatic Brain Tumor Segmentation using Cascaded Anisotropic Convolutional Neural Networks*. *CoRR*, abs/1709.00382, 2017.

[26] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji και Yichen Wei. *Fully Convolutional Instance-aware Semantic Segmentation*. *CoRR*, abs/1611.07709, 2016.

[27] Dimitrios Kollias Natalia Salpea, Paraskevi Tzouveli. *Medical Image Segmentation: A Review of Modern Architectures*. *Computer Vision - ECCV 2022 Workshops*, 2022.

[28] Diederik P Kingma και Max Welling. *Auto-encoding variational bayes*. *arXiv preprint arXiv:1312.6114*, 2013.

[29] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio και Pierre Antoine Manzagol. *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.* *J. Mach. Learn. Res.*, 11:3371–3408, 2010.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser και Illia Polosukhin. *Attention Is All You Need*. *CoRR*, abs/1706.03762, 2017.

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta και Kaiming He. *Non-Local Neural Networks*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[32] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov και Sergey Zagoruyko. *End-to-End Object Detection with Transformers*. *CoRR*, abs/2005.12872, 2020.

[33] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan L. Yuille και Liang-Chieh Chen. *Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation*. *CoRR*, abs/2003.07853, 2020.

[34] Yuanbo Wang, Unaiza Ahsan, Hanyan Li και Matthew Hagen. *A Comprehensive Review of Modern Object Segmentation Approaches*. *Foundations and Trends® in Computer Graphics and Vision*, 13(2–3):111–283, 2022.

[35] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz και Demetri Terzopoulos. *Image Segmentation Using Deep Learning: A Survey*, 2020.

[36] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, Werner Wiesbeck και Klaus Dietmayer. *Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges*. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2021.

[37] Çağrı Kaymak και Ayşegül Uçar. *A brief survey and an application of semantic image segmentation for autonomous driving*. *Handbook of Deep Learning Applications*, σελίδες 161–200, 2019.

[38] Çağrı Kaymak και Ayşegül Uçar. *A Brief Survey and an Application of Semantic Image Segmentation for Autonomous Driving*, σελίδες 161–200. Springer International Publishing, Cham, 2019.

[39] 2 Ali F. Khalifa, Eman Badr. *Deep Learning for Image Segmentation: A Focus on Medical Imaging*. *Computers, Materials & Continua*, 75(1):1995–2024, 2023.

[40] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy και Alan L. Yuille. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. *CoRR*, abs/1606.00915, 2016.

[41] Liang-Chieh Chen, George Papandreou, Florian Schroff και Hartwig Adam. *Rethinking Atrous Convolution for Semantic Image Segmentation*. *CoRR*, abs/1706.05587, 2017.

[42] Huikai Wu, Junge Zhang, Kaiqi Huang, Kongming Liang και Yizhou Yu. *FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation*. *CoRR*, abs/1903.11816, 2019.

[43] *Neural Style Transfer: Everything You Need to Know*. https://www.v7labs.com/blog/neural-style-transfer. Access date:2-6-2024.

[44] Hidetoshi Shimodaira. *Improving predictive inference under covariate shift by weighting the log-likelihood function*. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[45] *Domain Shift*. https://www.statlect.com/machine-learning/domain-shift. Access date:1-6-2024.

[46] Hao Guan και Mingxia Liu. *Domain Adaptation for Medical Image Analysis: A Survey*. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185, 2022.

[47] Wenjun Yan, Yuanyuan Wang, Shengjia Gu, Lu Huang, Fuhua Yan, Liming Xia και Qian Tao. *The Domain Shift Problem of Medical Image Segmentation and Vendor-Adaptation by Unet-GAN*, 2019.

[48] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt και Vaishaal Shankar. *Do ImageNet Classifiers Generalize to ImageNet?* *CoRR*, abs/1902.10811, 2019.

[49] Jingkang Yang, Kaiyang Zhou, Yixuan Li και Ziwei Liu. *Generalized Out-of-Distribution Detection: A Survey*. *CoRR*, abs/2110.11334, 2021.

[50] Dan Hendrycks και Thomas G. Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*. *CoRR*, abs/1903.12261, 2019.

[51] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang και Chen Change Loy. *Domain Generalization: A Survey*. *CoRR*, abs/2103.02503, 2021.

[52] Gilles Blanchard, Gyemin Lee και Clayton Scott. *Generalizing from Several Related Classification Tasks to a New Unlabeled Sample*. *Advances in Neural Information Processing Systems*J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira και K.Q. Weinberger, επιμελητές, τόμος 24. Curran Associates, Inc., 2011.

[53] Ian J. Goodfellow, Jonathon Shlens και Christian Szegedy. *Explaining and Harnessing Adversarial Examples*, 2015.

[54] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh και Sridha Sridharan. *Correlation-aware Adversarial Domain Adaptation and Generalization*. *CoRR*, abs/1911.12983, 2019.

[55] Zhun Deng, Frances Ding, Cynthia Dwork, Rachel Hong, Giovanni Parmigiani, Prasad Patil και Pragya Sur. *Representation via Representations: Domain Generalization via Adversarially Learned Invariant Representations*. *CoRR*, abs/2006.11478, 2020.

[56] Isabela Albuquerque, João Monteiro, Tiago H. Falk και Ioannis Mitliagkas. *Adversarial target-invariant representation learning for domain generalization*. *CoRR*, abs/1911.00804, 2019.

[57] Rui Shao, Xiangyuan Lan, Jiawei Li και Pong C. Yuen. *Multi-Adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[58] *Style Transfer*. https://paperswithcode.com/task/style-transfer. Access date:3-6-2024.

[59] Lei Li, Veronika A. Zimmer, Wangbin Ding, Fuping Wu, Liqin Huang, Julia A. Schnabel και Xiahai Zhuang. *Random Style Transfer based Domain Generalization Networks Integrating Shape and Spatial Information*. *CoRR*, abs/2008.12205, 2020.

[60] Xun Huang και Serge J. Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. *CoRR*, abs/1703.06868, 2017.

[61] Kaiyang Zhou, Chen Change Loy και Ziwei Liu. *Semi-Supervised Domain Generalization with Stochastic StyleMatch*. *CoRR*, abs/2106.00592, 2021.

[62] Nathan Somavarapu, Chih-Yao Ma και Zsolt Kira. *Frustratingly Simple Domain Generalization via Image Stylization*. *CoRR*, abs/2006.11207, 2020.

[63] Francesco Cappio Borlino, Antonio D'Innocente και Tatiana Tommasi. *Rethinking Domain Generalization Baselines*. *CoRR*, abs/2101.09060, 2021.

[64] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto L. Sangiovanni-Vincentelli, Kurt Keutzer και Boqing Gong. *Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data*. *CoRR*, abs/1909.00889, 2019.

[65] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao και Furao Shen. *Image Data Augmentation for Deep Learning: A Survey*, 2023.

[66] Emerald U. Henry, Onyeka Emebo και Conrad Asotie Omonhinmind. *Vision Transformers in Medical Imaging: A Review*. *ArXiv*, 2022.

[67] Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger Roth και Daguang Xu. *UNETR: Transformers for 3D Medical Image Segmentation*, 2021.

[68] Jiangyun Li, Wenxuan Wang, Chen Chen, Tianxiang Zhang, Sen Zha, Jing Wang και Hong Yu. *TransBTSV2: Towards Better and More Efficient Volumetric Segmentation of Medical Images*, 2022.

[69] B. Chen, Y. Liu, Z. Zhang, G. Lu και D. Zhang. *TransAttUnet: Multi-level Attention-guided U-Net with Transformer for Medical Image Segmentation*. *ArXiv*, 2021.

[70] Yucheng Tang, Dong Yang, Wenqi Li, Holger Roth, Bennett Landman, Daguang Xu, Vishwesh Nath και Ali Hatamizadeh. *Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis*, 2022.

[71] Eunji Jun, Seungwoo Jeong, Da Woon Heo και Heung Il Suk. *Medical Transformer: Universal Brain Encoder for 3D MRI Analysis*, 2021.

[72] Hong Yu Zhou, Jiansen Guo, Yinghao Zhang, Lequan Yu, Liansheng Wang και Yizhou Yu. *nnFormer: Interleaved Transformer for Volumetric Segmentation*, 2022.

[73] *Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics*. htm. Access date:3-6-2024.

[74] *NCI-ICBI 2013 Challenge -Automated Segmentation of Prostate Structures*. https://wiki.cancerimagingarchive.net/display/Public/NCI-ISBI+2013+Challenge+-+Automated+Segmentation+of+Prostate+Structures. Access date:3-6-2024.

[75] Quande Liu, Qi Dou, Lequan Yu και Pheng Ann Heng. *MS-Net: Multi-Site Network for Improving Prostate Segmentation With Heterogeneous MRI Data*. *IEEE Transactions on Medical Imaging*, 39(9):2713–2724, 2020.

[76] Quande Liu, Qi Dou και Pheng-Ann Heng. *Shape-aware Meta-learning for Generalizing Prostate MRI Segmentation to Unseen Domains*. *CoRR*, abs/2007.02035, 2020.

[77] Guillaume Lemaître, Robert Martí, Jordi Freixenet, Joan C. Vilanova, Paul M. Walker και Fabrice Meriaudeau. *Computer-Aided Detection and diagnosis for prostate cancer*

based on mono and multi-parametric MRI: A review. *Computers in Biology and Medicine*, 60:8–31, 2015.

[78] *MICCAI Grand Challenge: Prostate MR Image Segmentation 2012*. https://promise12.grand-challenge.org/. Access date:3-6-2024.

[79] N. Spanos. *Improvement of style transfer methods with data augmentation for domain generalization*. Διπλωματική εργασία, National Technical University of Athens, 2023.

[80] Christian Thode Larsen, J. Eugenio Iglesias και Koen Van Leemput. *N3 Bias Field Correction Explained as a Bayesian Modeling Method*. *Bayesian and grAphical Models for Biomedical Imaging*M. Jorge Cardoso, Ivor Simpson, Tal Arbel, Doina Precup και Annemie Ribbens, επιμελητές, σελίδες 1–12, Cham, 2014. Springer International Publishing.

[81] Humza Naveed, Saeed Anwar, Munawar Hayat, Kashif Javed και Ajmal Mian. *Survey: Image Mixing and Deleting for Data Augmentation*, 2023.

[82] Lee R. Dice. *Measures of the Amount of Ecologic Association Between Species*. *Ecology*, 26(3):297–302, 1945.

[83] Aaron Carass, Snehashis Roy, Adrian Gherman, Jacob C. Reinhold, Andrew Jesson, Tal Arbel, Oskar Maier, Heinz Handels, Mohsen Ghafoorian, Bram Platel, Ariel Birenbaum, Hayit Greenspan, Dzung L. Pham, Ciprian M. Crainiceanu, Peter A. Calabresi, Jerry L. Prince, William R. Gray Roncal, Russell T. Shinohara και Ipek Oguz. *Evaluating White Matter Lesion Segmentations with Refined Sørensen-Dice Analysis*. *Scientific Reports*, 10:8242, 2020.

[84] Diederik P. Kingma και Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2017.

List of Abbreviations

CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
GAN	Generative Adversarial Network
DG	Domain generalization
OOD	Out-of-Domain
IID	Independent and Identically Distribution