Addressing Domain Shift in CNN-based Image Segmentation: From Improving Robustness to Unsupervised and Active Learning based Domain Adaptation

KUMULATIVE DISSERTATION

zur Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

RASHA SHEIKH

aus Amman

Bonn, 2024

Angefertigt mit Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachter/Betreuer: Prof. Dr. Thomas Schultz
Gutachter: PD Dr. Volker Steinhage

Tag der Promotion: 20.02.2025
Erscheinungsjahr: 2025

Abstract

RASHA SHEIKH

Addressing Domain Shift in CNN-based Image Segmentation: From Improving Robustness to Unsupervised and Active Learning based Domain Adaptation

Pixel-wise labeling of images is a common task in applications nowadays as it allows humans and machines to make sense of the content in an image and informs their decisions about subsequent steps to be taken. Examples of such applications include scene understanding for autonomous driving, weeding in agricultural fields, and interventional therapies for brain tumors. It is however challenging to build machine learning models that perform well on data exhibiting different characteristics to what they have seen during training, or put differently, that they are robust to domain shift.

We address this problem using different approaches. A starting point is to increase the robustness of a segmentation model by encouraging it to focus more on the shape content of images rather than textural features. We accomplish this by using TV augmentation to smooth images while emphasizing object boundaries. We compare our work to other augmentation techniques and show the benefit of our approach. Another aspect we look into is design decisions employed by a popular open-source framework that is widely used for segmentation of medical images. Concretely, we investigate the effect of optimizers and the number of training epochs on domain generalization, and show through our experiments how the performance on new domains can be improved.

If labeled images from the target domain can be acquired, then these can be used to adapt models to the unseen domains. We devise smart sampling strategies to select which data samples should be annotated for efficient active learning. We first generate pseudo-labels and choose samples based on the loss and gradients of the network. Finally, if only unlabeled images are available, we use self-supervision to leverage this data and adapt the model to the target domain. We add a second branch to the trained model and drive the optimization of the model by comparing two sets of segmentation maps on target data. Our use of probabilistic maps and limited supervision reduces the risk of propagating incorrect signals throughout the network while allowing the model to adapt itself to target data.

Acknowledgements

I would like to thank my advisor Prof. Thomas Schultz for all the great guidance and insightful feedback he provided throughout my PhD journey. I would also like to extend my thanks to Prof. Volker Steinhage for taking the time to review the dissertation and give me invaluable advice. I am also grateful to Prof. Bennewitz and Prof. Fluck for agreeing to be part of the defense committee. My colleagues at the Medical Image Analysis group as well as those from other research groups, along with my friends have made this journey even more enjoyable and I thank them for all the conversations and time spent together. Of course my greatest thanks go out to my family who have been a constant source of encouragement, love, and support. This work is dedicated to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Semantic segmentation is a computer vision task where every pixel in an image is classified as belonging to one of a number of classes. This is a useful application and a desired goal in various fields. Vehicles with autonomous capabilities, for instance, use it to discern different structures in the surrounding scene. In the medical field as another example, segmenting images into semantic regions provides assistance for medical practitioners in diagnostic and interventional procedures.

If labeled training data is available, then deep learning models can be trained to produce segmentation maps with great accuracy. Deploying these models and evaluating them on unseen data is however not guaranteed to perform just as well. Domain shift is the problem that arises when there is a difference in the distributions of the so called source data, i.e data that the model was trained and validated on, and target data, i.e. data that will be used at inference time. Consequently, models that perform well on the training/validation set of one domain might perform poorly on data from another domain. This is a common issue in practice, as medical data for example, might be coming from different sites and generated with different scanners.

The deep neural networks used for semantic segmentation learn features that consecutively build on each other to extract the rich information needed to solve the task. Although several efforts have been made to understand the inner working of these deep models, they are still in some aspects similar to a black box where the details of how the output depends on the input is not fully interpretable. This makes it particularly challenging to understand which characteristics of the source domain the model is overfitting on and how the model should be changed to overcome that.

The generalizability issue of deep learning models is exasperated by the fact that in many fields, annotated data is expensive to generate. In the medical imaging field, for instance, expert knowledge is needed to label medical datasets. Given the limited capacity of medical professionals, it is highly desirable that trained models are able to generalize to new data without the need to acquire new labeled data from the target domain, and if the latter is possible or even necessary, then it is preferable if the amount of annotations requested is limited.

1.2 Contributions

We approach this issue of domain shift for semantic neural networks from two angles. The first one involves making the models more robust in the first place. In Chapter 2 (Sheikh and Schultz, 2020), we augment the training data with images that preserve semantic features and discard the high-frequency information that might

not be shared between the different domains. This encourages the model to focus on those features most relevant for the segmentation task, which consequently makes it more generalizable to other data. In Chapter 3 (Sheikh et al., 2022), we argue that the strategy of early-stopping, when combined with an adaptive optimizer, reduces the risk of overfitting on the training images. This in turn improves the segmentation performance on target data without hurting that of the source data.

The second aspect is domain adaptation. This is the task of taking a model that was trained on one domain and adapting it to another domain. In Chapter 4 (Sheikh et al., 2020), we devise strategies to pick which samples from the target data are to be labeled, in order to fine-tune the model using them. Our gradient and log-based approach picks samples that are diverse enough to allow the model to learn about the target data without having to label the entire dataset. In Chapter 5 (Sheikh and Schultz, 2022), we address the case of adapting the model without having any labeled samples from the target domain; an approach known as unsupervised domain adaptation. We use pseudo-labels from the trained model to refine its early features. This improvement is then propagated through the network resulting in better final segmentations.

1.3 Publications

The research outcome of the dissertation will be presented in the following chapters:

Chapter 2: Rasha Sheikh and Thomas Schultz (2020). "Feature Preserving Smoothing Provides Simple and Effective Data Augmentation for Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part I*. ed. by Anne L. Martel et al. Vol. 12261. Lecture Notes in Computer Science. Springer, pp. 116–126. DOI: 10.1007/978-3-030-59710-8_12. URL: https://doi.org/10.1007/978-3-030-59710-8_12 (Sheikh and Schultz, 2020).

Chapter 3: Rasha Sheikh et al. (2022). "Adaptive Optimization with Fewer Epochs Improves Across-Scanner Generalization of U-Net Based Medical Image Segmentation". In: *Domain Adaptation and Representation Transfer - 4th MICCAI Workshop, DART 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings*. Ed. by Konstantinos Kamnitsas et al. Vol. 13542. Lecture Notes in Computer Science. Springer, pp. 119–128. DOI: 10.1007/978-3-031-16852-9_12. URL: https://doi.org/10.1007/978-3-031-16852-9_12 (Sheikh et al., 2022).

Chapter 4: Rasha Sheikh et al. (2020). "Gradient and Log-based Active Learning for Semantic Segmentation of Crop and Weed for Agricultural Robots". In: *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*. IEEE, pp. 1350–1356. DOI: 10.1109/ICRA40945.2020.9196722. URL: https://doi.org/10.1109/ICRA40945.2020.9196722 (Sheikh et al., 2020).

Chapter 5: Rasha Sheikh and Thomas Schultz (2022). "Unsupervised Domain Adaptation for Medical Image Segmentation via Self-Training of Early Features". In: *International Conference on Medical Imaging with Deep Learning, MIDL 2022, 6-8 July 2022, Zurich, Switzerland*. Ed. by Ender Konukoglu et al. Vol. 172. Proceedings of Machine Learning Research. PMLR, pp. 1096–1107. URL: https://proceedings.mlr.press/v172/sheikh22a.html (Sheikh and Schultz, 2022).

1.4 Outline

In the following sections of this chapter we will review past and recent related work and describe how our own work fits into the respective aspect of the research problem we investigated. This will be followed by the actual publications in Chapters 2, 3, 4, 5. We conclude in Chapter 6 with a summary and outlook.

1.5 Preliminary - Segmentation Network

Networks used for semantic segmentation often follow an encoder-decoder architecture. The encoder part consists of several convolutional blocks arranged in multiple resolution levels. The first convolutional block uses the image as its input feature, and subsequent convolutional blocks in each level learn features using the output of the previous convolutions. These are followed by a non-linearity operation such as ReLU, and are then downsampled before inputting them into the following resolution level. As we go deeper into the encoder, the number of feature maps also progressively increases to capture more of the variations in the input. The decoder part of the network then uses convolutions in multiple levels again to gradually reduce the number of these feature maps and increase the resolution to finally have a segmentation map at the end matching the size of the input image.

Several enhancements to this basic architecture have been proposed. U-Net (Ronneberger et al., 2015) introduces skip connections between the corresponding levels of the encoder and decoder. The goal of these connections is to reintroduce the finer details that are lost due to the downsampling operations. The original U-Net architecture concatenated features from the encoder with those from the decoder and followed that with convolutions that learn from both sets of features. Subsequent works (Isensee et al., 2021) have also experimented with adding those features instead of concatenating them in order to reduce the number of parameters. The model in Chapter 4 does not use skip connections. It rather stores the max-pooling indices used in the encoder and passes them to the decoder. Concatenating features is used in Chapters 2 and 3, whereas in Chapter 5 they are added.

In Chapter 3, we use the nnUNet (Isensee et al., 2021) as our segmentation framework. As the name suggests (No-New-UNet), this framework uses the U-Net architecture for its model but instead of having a fixed structure, the framework adapts the model to properties of the dataset it is trained on. If the images are quite large for instance, it is possible to downsample them a greater number of times in comparison to a dataset with smaller images. The framework would therefore in this case favor instantiating deeper models in order to learn richer features.

Another variation to the original U-Net implementation that is sometimes used and is in fact employed by nnUnet is deep supervision. This is where segmentation maps are learned at the end of each resolution level instead of only at the final one. Having these segmentation maps contribute to the overall loss aids in learning consistent segmentations at different levels since multiple training signals are propagated back. We do not use deep supervision in our models except for the work in Chapter 3 where the nnUNet framework is used.

1.6 Augmentations

One common method to improve the generalization of models is to augment the training data with different transformations. A survey of data augmentation techniques can be found in (Chlap et al., 2021). We summarize next some of the common techniques and recent work in this field. Traditional augmentations include geometric transformations such as scaling, flipping, rotation, and shear. Operations that change the intensities of images are often used. Examples include Gamma correction, histogram equalization, noise injection with Gaussian, uniform, or salt and pepper noise. Filtering the image to sharpen it or blur it can be helpful as well.

More sophisticated augmentation methods include elastic deformations with random displacement fields or deforming images by registering them to other scans. Generative Adversarial Networks (GANs) (Goodfellow et al., 2020) have also been exploited to synthesize new images that look similar to the original training images (Zhu et al., 2017; Isola et al., 2017). In the case of MR images, there have been works (Shaw et al., 2019), that introduce alterations to the k-space of images, which when reconstructed would show effects similar to motion artifacts that MR images can suffer from. Augmenting the training data with these images would therefore increase the robustness of the model against this particular type of artifact. Recently, diffusion models (Ho et al., 2020) have been used to generate synthetic images that are similar to the input distribution. These images can be conditioned on certain properties of the dataset generating realistic-looking images (Pinaya et al., 2022).

Nalepa et al., 2019 surveyed the type of augmentations used by the submissions to the BraTS 2018 challenge and found that the most common ones in the top performing submissions were simple affine transformations such as flipping, rotation and scaling.

Another form of augmentation preserves the original image intensities and geometry and instead occludes parts of it (DeVries and Taylor, 2017). The intuition here is that the network will then rely on global information available in the image to discern useful features. This bears similarities to dropout, differently however, it operates on the input to the network rather than internal feature maps of the network. Pixels in the occluded regions could share the same value or as in (Zhong et al., 2020) be sampled from the entire range of intensities. This introduces an additional source of noise that the network should learn to discard and focus instead on other key features of the image.

Zhang et al., 2018a argue that augmentations based on geometric transformations are dataset-dependent, in the sense that prior knowledge is needed to determine whether a certain augmentation of the image should produce the same label. They came up therefore with a new type of augmentation that is not dependent on this knowledge. In their work, linear combinations of a pair of images along with the same linear combination of the target labels are added to the training data. With this augmentation, their models were able to generalize better and additionally better cope with noisy labels. Eaton-Rosen et al., 2018 verify that this approach not only works for natural images but is also helpful for the task of tumor segmentation.

Once a set of augmentations is found to be useful, a natural extension is to compose them. An image could for instance be first rotated then have its intensity modified. This increases the variability of data and improves the robustness of the model. In (Hendrycks et al., 2020), several such compositions are performed, then linearly combined. In addition to the primary loss that uses the ground-truth, a consistency loss encourages the network to produce similar embeddings for the original image and its augmentation.

The best augmentations can also be learned instead of being fixed. Cubuk et al., 2019 use Reinforcement Learning to train a controller RNN that predicts policies of different augmentations which are then used to train a network. The validation accuracy of that trained network acts subsequently as the reward that guides the training of the network. Although this exploration of the augmentations search space could boost the performance, training so many models is expensive and therefore a significant drawback of this approach.

Cubuk et al., 2020 found that a similarly good performance can be obtained with a reduced search space. The two hyperparameters needed are the number of composite transformations applied to each image and their magnitude. Both of which could be optimized with a grid search using a validation set. They argue that these parameters are also intuitively related to the strength of regularization induced by the augmentation. A smaller model for example might require weaker regularization so one could reduce the number of composite transformations applied to an image and/or reduce how large their deviation is from the original image.

To preserve object boundaries, Hammoudi et al., 2022 exploit the structure obtained with superpixels. Once these are generated, a few can be randomly selected to be replaced with their mean values or dropped altogether in a similar fashion to cutout (DeVries and Taylor, 2017) with pixels. People have also experimented with mixing images, where the pixel values of dropped superpixels are replaced with those from a different image (Hammoudi et al., 2022; Franchi et al., 2021).

In our work, we augment the data with images that have been generated using Total Variation (Rudin et al., 1992). This type of filtering has the appealing quality that it smoothes images without corrupting edges. This is particularly useful for the task of semantic segmentation since edges often delineate different classes.

Recently, Wang et al., 2020 have argued that models are driven to learn both the low-frequency components, which humans correlate with class semantics, and the high-frequency components. They found that this latter part makes the model less robust to data that has been perturbed. This observation supports our approach where we discard textural information with TV augmentation and focus on the semantics.

The motivation behind using TV-augmentation in our work was to encourage the model to focus on shape rather than texture. There have also been works that aim to increase the impact of shape using techniques other than augmentations. Takikawa et al., 2019, for example, add a branch to their model that focuses on learning the shape of objects in an image. Attention maps are constructed from the main branch and applied to the second branch at multiple layers, resulting in shape features that are then fused back with the main segmentation maps. In addition to the ground-truth semantic segmentation masks, the loss includes terms that use binary edge masks, driving the model to learn semantic classes that explicitly exploit shape information. Zotti et al., 2018, on the other hand, make use of shape priors. Label maps are first aligned then the the pixel-wise frequencies of each class are computed. These frequencies are turned into probabilities and act as shape priors, which are concatenated with feature maps from the network and further processed. The contours of the classes are also added as an additional component to the loss. Another work that explicitly integrates shape priors is (Ambellan et al., 2019). The task in their work is to segment knee and bone cartilage using MR images. A combination of 2D and 3D CNNs are trained to learn the desired segmentation. A Statistical Shape Model (SSM) is applied to the output as a post-processing step in order to increase the robustness of segmentations in the presence of image artifacts or low contrast, and produce label masks that are anatomically valid.

1.7 Early Stopping and Optimizers

Arpit et al., 2017 investigate how the learning process of deep neural networks evolves over time. They argue that the complexity of what the network learns is reflected in the number of decision regions that the input space is divided into. This can be approximated by looking at the number of samples with predictions different from one or more other data points in their neighborhood. They track this measure over the training time of the network and observe that this number gradually increases along with the number of epochs until it reaches a plateau. They conclude that this indicates the model first learns simple patterns common between the samples then starts learning more complex decision boundaries fitting even noisy data.

Wang et al., 2020 argue that models tend to first learn low-frequency components of images, and these components are what usually allows the model to generalize to unseen data. Then as the model is trained for more and more epochs, the high-frequency components are learned. These latter components reflect textural information specific to the training images or even noise that should not be learned.

The insights of the previous two works support our idea of not training the model for a large number of epochs if we want to generalize to other domains, since otherwise the model might start learning features that are highly tailored to the source domain and do not carry over to a different domain. With early-stopping on the other hand, the model is more likely to have learned features that are still semantically relevant but with a weaker dependence on the peculiarities of the training data.

Loshchilov and Hutter, 2019 investigated the influence of combining L2 regularization with different optimizers when training deep learning models. Adding the L2 norm of the weights to the loss helps regularize the model and prevent overfitting. In the popular open-source frameworks, weight decay, where weights are exponentially decayed over time, has been used as a proxy for L2 regularization. Loshchilov and Hutter, 2019 show however that weight decay and L2 regularization do not necessarily lead to similar formulations. In the case of SGD, they only differ by a factor which is the learning rate. But in the case of adaptive optimizers, the two methods are not equivalent since gradients are rescaled according to their first and second moments which leads to a tight coupling with weight decay. The authors argue that because of this discrepancy, the desired effect of L2 regularization in combatting overfitting is weakened when adaptive optimizers from online frameworks have been used. In other words, the perceived higher performance of SGD in some experiments is not attributed to the use of the non-adaptive optimizer, rather to the more effective regularization taking place in the online implementations that combine weight decay with the respective optimizer.

Wang et al., 2020 convert images X into the frequency domain then reconstruct them back into two sets. The first one is generated using the low frequency components X_l, and the second one using the high frequency components X_h. A model is then trained on the original images and evaluated on X_l and X_h. A second set of experiments trains models on only the low-frequency or only the high-frequency reconstructions. They show that low-frequency information exhibits better generalizability, and that models which have captured more of the high-frequency information during training is more susceptible to adversarial attacks and noisy labels. They repeat these experiments with different variations in the training pipeline including optimizers such as SGD, ADAM, and others. They find that the high frequency information is more prevalent in the model trained with SGD compared to others. This

further supports our finding that training with adaptive methods and for a shorter number of epochs is beneficial for domain generalization.

1.8 Active Learning

Deep segmentation models often require to be trained on large training sets. This allows them to better capture the distribution of data and the variety of features that share the same semantic class. Annotating a large number of pixels is tedious however, and sometimes even requires expert knowledge that is not readily available, as is the case with medical images. Reducing this effort is thus desirable and active learning is one way to address that.

The aim in active learning is to *actively* select a smaller subset of the data to be annotated such that when the model is trained on this subset, it shows a similar performance to that had it been trained on the whole dataset. A typical process would start with a small labeled subset of the data, train the model on it, then run several rounds where certain criteria are used to filter the remaining unlabeled data, ask for those to be annotated, retrain the model and repeat those last steps until the model performance is satisfactory or the training budget has been consumed.

The question then becomes, how would one choose these samples in an intelligent way rather than just randomly sampling some of the unlabeled data in each round? Ideally these samples should be representative of the training set, diverse enough to capture the underlying distribution, and show different characteristics to what the model has already seen.

Surveys of active learning methods can be found in (Ren et al., 2021; Budd et al., 2021). We summarize next some of the common techniques. The first category of traditional active learning queries favors those samples with high prediction uncertainty. In the case of segmentation, this could simply be those samples where the average maximum probability is low. Another popular approach is to select those images with the highest average entropy. If the number of classes is larger than two, one could also choose those where the difference between the most probable class and the second most probable class is small, which would indicate that the model is not able to highly discriminate between the correct class and the other ones for the particular image.

Uncertainty can also be estimated through bootstrapping (Yang et al., 2017), where data is sampled from the training set with replacement to create multiple subsets. These are then used to train multiple models and the discrepancy between these models serves as an uncertainty measure to select samples for annotation. Monte-Carlo dropout is another popular method for this criteria. Models are trained with dropout, then at inference time rather than disabling it, dropout is used to generate multiple variants of the model. With each forward pass, a different instance of prediction is obtained for each input. The entropy of the average predictions is then used to determine which samples to annotate. Compared to training an ensemble of models, MC dropout is preferred because it is computationally less expensive. Beluch et al., 2018 argue however that ensembling might still be a better option because of the reduced capacity of the MC model.

In addition to using MC dropout, Gal et al., 2017 also take into account the mutual information between the predictions and the model weights. Having multiple models obtained through MC dropout, they compute the difference between the

entropy of the average prediction and the average of each individual prediction entropy. The goal is to pick samples where the model is not only uncertain about but also this uncertainty drops when some of the weights are changed.

Another type of queries focuses on how representative the unlabeled samples are. Sener and Savarese, 2017b use clustering to group the data, then iteratively pick for annotation those samples with the largest distance to their nearest cluster center. Yang et al., 2017 compute the cosine similarity between each pair of unlabeled samples and choose those with the highest average similarity to all the others.

Naturally a combination of these methods can also be used. Smailagic et al., 2018 for example, exploit the richer features that CNNs are able to learn. Starting with a small annotated subset, a network is trained on those samples. The unlabeled data is passed through the network and those with highest entropy proceed to the second stage of selection. The embedded features extracted from the network are then examined to find the next candidate for annotation. This is done by computing the pairwise distance between the samples features and choosing the one with the highest average distance.

The methods defined so far rely on pre-defined heuristics for what kind of information the model might benefit form: uncertainty, diversity, representativeness, etc. A different paradigm frames data selection as a learning problem, e.g. using reinforcement learning. In (Woodward and Finn, 2017), samples are presented to the network and the model has to decide whether to produce a prediction or to request for more annotated data. As is typical in reinforcement training, the model would get back feedback in the form of negative rewards (penalties) for annotation requests and for wrong predictions and positive rewards for correct predictions.

Our work introduces several strategies to select samples for annotation. They are based on choosing samples that might have the biggest impact on the model parameters. These strategies rely on having first pseudo-labels which we generate by clustering the RGB values of images. Looking at a single image, a human annotator then selects those clusters that show vegetation. This results in foreground-background segmentation that we use as pseudo-labels for the subsequent step. Our first strategy of selection uses the loss between the model prediction and the pseudo-groundtruth as a measure of how well the model is performing on the unlabeled samples. Those with the highest loss are candidates for annotation. Rather than simply picking those at the top, however, we use a log-scale to encourage the diversity of the selected samples. The second strategy computes the weights gradients of the model using the loss described above. Those samples with the largest norm of the gradients are chosen for annotation, again using a log-scale approach. The last strategy also uses the computed gradients but relies on a different method to encourage the diversity of samples. The sample with the highest gradient norm is first selected. Then the gradient of each unlabeled sample is projected onto that of the first one. This is then subtracted from the original gradient resulting in a residual gradient that reflects how much the weights would change after taking into account the already selected samples.

We next present recent works for active learning and mention common characteristics with our work if they exist.

These works build on traditional ideas presented at the beginning of the section and extend them further. For the task of cell-counting, Wang and Yin, 2021 train several models and compute the uncertainty among them and the uncertainty among different epochs to rank samples for annotation. To ensure that the samples are diverse, their features in the latent space are classified into a predefined number of clusters. The samples are further filtered by ensuring that they have a high cosine

similarity with the unlabeled samples in their own cluster and a low cosine similarity with already labeled samples.

Active learning can also be leveraged for annotating 3D medical images. In (Wu et al., 2022), a single slice from each 3D image is annotated, then the labels are propagated to the rest of the volume. To pick the most representative 2D slice, the cosine similarity measure is computed between each slice and all the other slices and the one with the largest score is picked for annotation.

Similarly to our work, Nath et al., 2022 first generate pseudo-labels for unlabeled data. The authors threshold the images and refine the output by running it through the connected components analysis and using the resulting labels. A model is subsequently trained on the images and their pseudo-labels. Dropout is then used at inference to generate an uncertainty measure for each volume. Those with high uncertainty measure are then picked to be annotated next.

Bai et al., 2022 leverage the information obtained through Grad-CAM to create pseudo-labels. These are thresholded at different levels and used to train several models, with the discrepancy between the models used as a guidance to select samples for the next annotation iteration.

One of our strategies uses weight gradients. Dai et al., 2020 also use gradients but those of the input space instead. A VAE is first trained on the dataset to learn a latent space. A separate segmentation model is next trained for a few epochs using a small subset of labeled data. The gradients of the loss with respect to the input image are then back propagated and the images are perturbed using these gradients. These new images are passed through the VAE and their encodings are used to search for unlabeled images with similar encodings. These are then suggested for annotation.

Parvaneh et al., 2022 select labels for annotation by investigating how consistent their labels are when their CNN features are interpolated with those of two labeled samples. If the model produces different predictions, they join a pool of candidate samples that will be later filtered by clustering them into diverse samples. They show that these samples often lie at the borders between different classes, so choosing them for annotation helps improve the discriminative power of the model.

After deciding on which images to annotate, some works (Siddiqui et al., 2020; Cai et al., 2021) suggest using superpixels for annotation rather than pixels to further reduce the annotation work.

Finally, one should consider how the model will be trained given the new annotated samples. Options include retraining from scratch, freezing some layers and fine-tuning on the new samples. The optimal choice will depend on how computationally expensive it is to retrain the model, and whether the original labeled data is still available, which might not be the case for scenarios where models are trained on private hospital data for example, and the model has to be adapted to data acquired at a different site.

Munjal et al., 2022 argue that rather than relying on the initial hyperparameters optimized on the small labeled set and keeping them fixed, one should optimize them at every active learning round. As the labeled set grows larger and larger, hyperparameters such as the learning rate and weight decay should be tuned since the distribution of training data has most likely changed as more and more data is annotated and added to the set.

1.9 Unsupervised Domain Adaptation

Adversarial learning is often used in domain adaptation, e.g. through CycleGAN (Zhu et al., 2017), to transfer the style of source images to target images or the other way around. This helps reduce the difference in the distribution of input images from different domains, thereby enabling the model to produce better predictions when switching from one domain to another. This technique is used by Sun et al., 2022, who employ several steps to adapt the model to target data. Source input images are first transformed using an adversarial loss to have a similar appearance to that of the target images. An encoder is then trained to extract features which are then split into domain-invariant and domain-dependent features. A classifier is trained on the domain-specific features to distinguish between the two domains. The domain-invariant features are propagated into a segmentation module that produces pixel-wise predictions. At inference-time, the cosine similarity measure is applied to adjacent slices to ensure consistent predictions.

A similar approach is followed by Lee et al., 2023, where they use an auxiliary task to help the model adapt to unseen data. The model is trained on source data to produce segmentation maps via a segmentation loss and also to generate images similar to the input via an adversarial loss. The network components involved in the generation part are then fine-tuned on the target data. The aim is to separate semantic information relevant to the segmentation task and is common between different domains, from the characteristics that are specific to a domain and contribute to the gap in the model's performance on source and target data.

Koch et al., 2022 transfer the style of images without using adversarial training. Singular Value Decomposition (SVD) is used to decompose source and target images. The first k components from the source data are then combined with the trailing ones from the target images, and the resulting reconstructions are used for training. The argument is that the semantic content is contained in the leading components whereas the remaining ones can be treated as noise that cause the performance to degrade when applying a trained model on a different domain. By mixing both sets of components and reconstructing images, they qualitatively show that the style of target images has been transferred to source images, which they can then use for domain adaptation.

Multiple works incorporated a classifier in the pipeline in order to train the model to learn the domain directly. Lin et al., 2023 use the classifier's features to generate domain-specific prompts which are fused with the segmentation encoder features before propagating them to the decoder. The goal is to learn and explicitly incorporate information that distinguishes one domain from another. The authors use both source and target data along with their labels to train the model, but the method could also be adapted to work in the unsupervised setting by relying on a discriminator to distinguish domain-specific features.

Feng et al., 2022 argue that domain adaptation should also take into account the distribution shift between classes in different domains. They add to the loss regularization terms for classes within the source domain, within the target domain, and across the two domains. Representative features are created for each class and the distance between those of the source and target domains is minimized. These prototype features are also used to encourage the source domain to produce similar class features for each sample by penalizing the difference between them and the representative features. On the target domain, a consistency loss is added to push the model to produce similar class predictions for the original images and their augmented variants. An adversarial term is also added to the loss to encourage the

network to align the distributions of edges and segmentations.

Contrastive loss has also been used for unsupervised domain adaptation. Gomariz et al., 2022 train their model on the source data with a segmentation loss, and on the source and target with a contrastive loss. To use contrastive learning, bottleneck features of the U-Net model are projected into a vector. The loss then tries to minimize the distance between pairs of positive and negative examples. Positive examples are generated through sampling from adjacent slices in the 3D volume and then applying augmentation transformation to them. Negative examples are those that arise from different domains. The authors show that this method not only improves the adaptation performance on unlabeled target data but it also positively influences the performance on the labeled source data.

A student-teacher learning paradigm is used by Xu et al., 2023 to segment unlabeled fetal images based on atlas annotations. Since the appearance can vary between the two domains, they employ several steps to align the different features. Using Fourier transform and its inverse, low-frequency components of the source and target data are swapped. The student model is then optimized to produce similar segmentations. The weights of the teacher model are adapted using Exponential Moving Average (EMA) of the student's weights. A consistency loss encourages both models to produce similar outputs for the the source and target data. The authors also use a form of mixup augmentation where blocks of two images are combined. The same procedure is applied to their teacher predictions which are then used as pseudo-labels for the student model.

Hu et al., 2022b combine several of the previously mentioned techniques. A classifier is trained to learn the domain of source and target images in addition to a new domain constructed from augmented source images, where the low-frequency components are swapped with those from other source or target images. The domain predictor is then used to generate features specific to a domain, and a decoder is added to produce segmentation maps.

The use of pseudo-labels for domain adaptation has also been used in (Ghamsarian et al., 2023). Self-training with pseudo-labels generated from predictions on target data suffers however from the risk that the pseudo-labels are incorrect which would negatively influence what the network learns. The authors propose applying intensity transformations to target images and comparing afterwards the model's predictions using the set of original and transformed images. Predictions with high confidence values are then used as pseudo-labels to train the network on target data along with labeled source data. The motivation being that those pixels, where the model shows robustness to transformations, are more likely to have correct predictions which would mitigate the risk of training with incorrect labels.

In our work, we also make use of pseudo-labels for self-training to adapt the model to a new domain. We add an early segmentation head just before the first down-sampling operation. The model now produces two segmentation maps, the usual one at the end of the network and a rough one at the beginning. The segmentations at the end act as pseudo-labels for the early predictions, and we adapt the early feature maps of the network using the loss between the two predictions. As the updated activations are propagated through the network, this also produces better final segmentations which again can be used for refinement. Since we do not binarize the pseudo-labels and instead keep their probabilistic values to guide the early rough segmentations, in addition to only refining a subset of weights, we manage to better adapt to the target data while reducing the risk of using wrong predictions for adaptation.

Chapter 2

Feature Preserving Smoothing Provides Simple and Effective Data Augmentation for Medical Image Segmentation

2.1 Summary

Augmenting training data with transformed images is a simple way to increase the diversity of images that the model will see during training, thereby allowing it to better generalize to unseen data. We augment our data with Total Variation (TV) smoothing which discards noisy information while preserving shapes in images. Geirhos et al., 2019 ran a set of experiments where they mixed texture and shape information from different images. An example would be an image of a cat with the texture changed to be that of an elephant through style transfer. They found that CNNs trained on ImageNet favour texture whereas humans favor shape, therefore the constructed image mentioned previously would still be classified as a cat by humans. This inspired us to use an augmentation that focuses on shape information, so that models are encouraged to output the same class even in the presence of domain shift.

Denoising images with Total Variation (TV) creates images with piece-wise constant regions. It smoothes images while preserving edges, which is a useful property for segmentation tasks as classes can still be delineated while noisy information is reduced. We apply this smoothing to our training set and use the smoothed images along with the original ones to train our model.

Our network follows the U-Net architecture with encoder and decoder blocks and skip connections in-between, and we train it using the dice loss. We evaluate our method on the Spinal Cord Grey Matter (SCGM) dataset (Prados et al., 2017), Brain Tumor Segmentation Challenge (BraTS) dataset (Menze et al., 2014; Bakas et al., 2017; Bakas et al., 2018), and the White Matter Hyperintensity Challenge (WMH) dataset (Kuijf et al., 2019). All of these datasets contain images collected from different sites and exhibit domain shift when training on one site and evaluating on another.

The SCGM dataset consists of MR images of the Spinal Cord and the task is to segment the Grey Matter in these images. The data is collected from four sites. We train our model on one site and evaluate on the test sets of all sites. Since this dataset was annotated by four annotators, we conduct two sets of experiments. In the first one, the model is trained on annotations from the first rater, and in the second one, the model is trained on annotations from all raters. In the second case, each image

along with three other copies of it are presented to the network, each time with a different annotation mask.

We compare TV smoothing to other filters such as the guided filter, the bilateral filter, and the Gaussian filter, as well as the common augmentations for medical datasets such as flipping, rotation, and elastic deformation. A side effect of JPEG compression is that it smoothes images so we also compare it to our method. Our results show that TV augmentation leads to the best performance in most of the different settings we tested it.

We also evaluated the effect of varying the strength of TV smoothing, and found that the positive improvement in performance is robust to this parameter. An additional benefit is gained when these are combined so that the model is presented with images smoothed at different strengths.

The second dataset that we evaluate our model on is the BraTS 2019 dataset. We split the data depending on its origin into four groups and test the effect of TV augmentation in the presence and absence of domain shift. We segment the whole tumor using FLAIR as the input modality. We observe an improvement in performance in three out of four groups when the training data is augmented with TV smoothing.

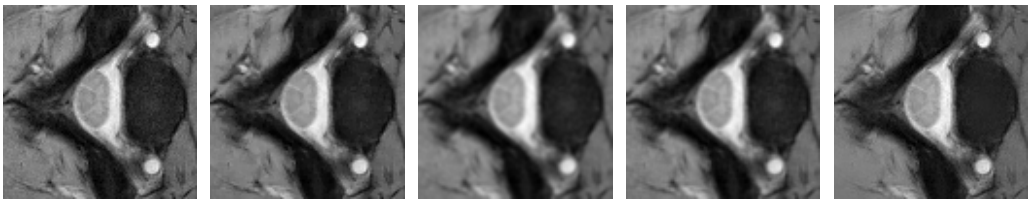
The third dataset used for evaluation is the WMH dataset from the corresponding challenge. After training the model with TV augmented data, we evaluate it on held-out test sets that we generated when splitting the data before training, and we also upload our model to the official challenge page so that it is evaluated on the official test set. In both cases, our augmentation improves the performance.

In the context of this dissertation, the work presented in this chapter suggests an inexpensive method to increase the robustness of deep learning models for the task of semantic segmentation. Models are trained with different transformations of the same data tend to generalize better since the augmentations encourage the model to focus on the semantic content of images and the relationship between that and the ground-truth classes. Feature-preserving smoothing such as Total Variation denoising discards information which does not contribute to the segmentation task, and might even negatively affect the performance when models are evaluated on different domains than what they were trained on. Other traditional methods such as Gaussian blurring also smooth images but they run the risk of smoothing out edges, thereby losing boundaries of classes. TV smoothing on the other hand, creates piecewise constant regions and preserves edges, which makes it a suitable candidate to increase the robustness of segmentation models against domain shift.

The content of this chapter is published as:

Rasha Sheikh and Thomas Schultz (2020). "Feature Preserving Smoothing Provides Simple and Effective Data Augmentation for Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part I*. ed. by Anne L. Martel et al. Vol. 12261. Lecture Notes in Computer Science. Springer, pp. 116–126. DOI: 10.1007/978-3-030-59710-8_12. URL: https://doi.org/10.1007/978-3-030-59710-8_12.

Contribution of the thesis author: Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization.



(A) (B) (C) (D) (E)

FIGURE 2.1: Sample image (A) and the results of smoothing with Total Variation (B), Gaussian (C), bilateral (D), and guided filters (E).

2.2 Abstract

CNNs represent the current state of the art for image classification, as well as for image segmentation. Recent work suggests that CNNs for image classification suffer from a bias towards texture, and that reducing it can increase the network's accuracy. We hypothesize that CNNs for medical image segmentation might suffer from a similar bias. We propose to reduce it by augmenting the training data with feature preserving smoothing, which reduces noise and high-frequency textural features, while preserving semantically meaningful boundaries. Experiments on multiple medical image segmentation tasks confirm that, especially when limited training data is available or a domain shift is involved, feature preserving smoothing can indeed serve as a simple and effective augmentation technique.

2.3 Introduction

Image segmentation is a key problem in medical image analysis. Convolutional neural networks (CNNs) often achieve high accuracy, but only if trained on a sufficient amount of annotated data. In medical applications, the number of images that are available for a given task is often limited, and the time of experts who can provide reliable labels is often scarce and expensive. Therefore, data augmentation is widely used to increase the ability to generalize from limited training data, and it is often indispensable for achieving state-of-the-art results.

Augmentation generates artificial virtual training samples by applying certain transformations to the original training images. Many of these transformations reflect variations that are expected in test images. Examples are geometric transformations such as image flipping, rotations, translations, elastic deformations (Ronneberger et al., 2015), or cropping, but also intensity or color space transformations, which can be used to simulate changes in illumination or acquisition device characteristics (Shorten and Khoshgoftaar, 2019; Billot et al., 2020). More complex augmentation has been performed via data-driven generative models that either generate images for augmentation directly (Bowles et al., 2018; Sandfort et al., 2019), or generate spatial and appearance transformations (Chaitanya et al., 2019; Zhao et al., 2019). However, implementing and training these approaches requires a relatively high effort.

In this work, we demonstrate that feature preserving smoothing provides a novel, simple, and effective data augmentation approach for CNN-based semantic segmentation. In particular, we demonstrate the benefit of augmenting the original training images with copies that have been processed with total variation (TV) based denoising (Rudin et al., 1992). As shown in Figure 2.1 (b), TV regularization creates

piecewise constant images, in which high frequency noise and textural features are removed, but sharp outlines of larger regions are preserved.

Unlike the above-mentioned augmentation techniques, ours does not attempt to generate realistic training samples. Rather, it is inspired by the use of neural style transfer for augmentation. Neural style transfer combines two images with the goal of preserving the semantic contents of one, but the style of the other (Gatys et al., 2016). It is often used for artistic purposes, but has also been found to be effective as an augmentation technique when training CNNs for image classification (Jackson et al., 2019). To explain this, Geirhos et al., 2019 perform experiments for which they generated images with conflicting shape and texture cues. When interpreting such images, ImageNet-trained CNNs were found to favor texture, while humans favor shape. Since shape is more robust than texture against many image distortions, this might be one factor that allows human vision to generalize better than CNNs.

Our work is based on the hypothesis that CNNs for image segmentation suffer from a similar bias towards texture, and will generalize better when increasing the relative impact of shape. Augmenting with TV regularized images should contribute to this, since it preserves shapes, but smoothes out high frequency textural features. In a similar spirit, Zhang et al., 2019 use superpixelization for augmentation, and Ma et al., 2019 use lossy image compression. Both argue that the respective transformations discard information that is less relevant to human perception, and thus might prevent the CNN from relying on features that are irrelevant for human interpretation. However, Ma et al. only evaluate JPEG augmentation in one specialized application, the segmentation of sheep in natural images. As part of our experiments, we verify that their idea, which has a similar motivation as ours, carries over to medical image segmentation.

2.4 Materials and Methods

2.4.1 Selection of Datasets

We selected datasets that allow us to test whether feature preserving smoothing would be an effective data augmentation technique when dealing with limited training data in medical image segmentation. Moreover, we hypothesized that this augmentation might reduce the drop in segmentation accuracy that is frequently associated with domain shifts, such as changes in scanners. If differences in noise levels or textural appearance contribute to those problems, increasing a network's robustness towards them by augmenting with smoothed data should lead to better generalization.

As the primary dataset for our experiments, we selected the Spinal Cord Grey Matter Segmentation Challenge (Prados et al., 2017), because it includes images that differ with respect to scanners and measurement protocols, and provides detailed information about those differences. To verify that our results carry over to other medical image segmentation tasks, we additionally present experiments on the well-known Brain Tumor Segmentation Challenge (Menze et al., 2014; Bakas et al., 2017; Bakas et al., 2018), as well as the White Matter Hyperintensity Segmentation Challenge (Kuijf et al., 2019).

2.4.2 CNN Architecture

We selected the U-Net architecture (Ronneberger et al., 2015) for our experiments, since it is widely used for medical image segmentation. In particular, our model

is based on a variant by Perone et al., 2019, who train it with the Adam optimizer, dropout for regularization, and the dice loss

$$\text{dice} = \frac{2\sum pg}{\sum p^2 + \sum g^2}, \tag{2.1}$$

where p is the predicted probability map and g is the ground-truth mask. Our model is trained to learn 2D segmentation masks from 2D slices. The initial learning rate is 0.001, the dropout rate is 0.5, betas for the Adam optimizer are 0.9 and 0.999, and we train the model for 50, 30, and 100 epochs for the SCGM, BraTS, and WMH datasets respectively. The number of epochs is chosen using cross-validation, the others are the default settings of the framework we use.

2.4.3 Feature-Preserving Smoothing

We mainly focus on Total Variation based denoising (Rudin et al., 1992), which we consider to be a natural match for augmentation in image segmentation problems due to its piecewise constant, segmentation-like output. The TV regularized version of an n-dimensional image $f : D \subset \mathbb{R}^n \to \mathbb{R}$, with smoothing parameter α, can be defined as the function $u : D \to \mathbb{R}$ that minimizes

$$E(u; \alpha, f) := \int_D \left(\frac{1}{2} (u - f)^2 + \alpha \|\nabla u\| \right) dV, \tag{2.2}$$

where the integration is performed over the n-dimensional image domain D. Numerically, we find u by introducing an artificial time parameter $t \in [0, \infty)$, setting $u(\mathbf{x}, t = 0) = f$, and evolving it under the Total Variation flow (Andreu et al., 2001)

$$\frac{\partial u}{\partial t} = \text{div} \left(\frac{\nabla_{\mathbf{x}} u}{\|\nabla_{\mathbf{x}} u\|} \right), \tag{2.3}$$

using an additive operator splitting scheme (Weickert et al., 1998). The resulting image $u(\mathbf{x}, t)$ at time t approximates a TV regularized version of f with parameter $\alpha = t$. We apply TV smoothing to all images in the training set, and train on the union of both sets, randomly shuffling all images before each epoch. We did not observe a clear difference between this basic strategy and a stratified sampling which ensured that half of the images in each batch were TV smoothed.

For comparison, we also consider two alternative feature preserving filters, the bilateral filter (Aurich and Weule, 1995) and the guided filter (He et al., 2010), as well as standard, non feature preserving Gaussian smoothing. We hypothesized that feature preserving filters other than TV regularization might also be effective for augmentation, even though maybe not as much, since they do not create a segmentation-like output to the same extent as TV. We expected that Gaussian smoothing would not be helpful, because it does not preserve sharp edges, and therefore does not provide clear shape cues to the network. Moreover, it can be expressed using a simple convolution which, if useful, could easily be learned by the CNN itself.

2.5 Experiments

2.5.1 Spinal Cord Grey Matter

This challenge dataset consists of spinal cord MR images of healthy subjects acquired from four different sites with different scanners and acquisition parameters. The

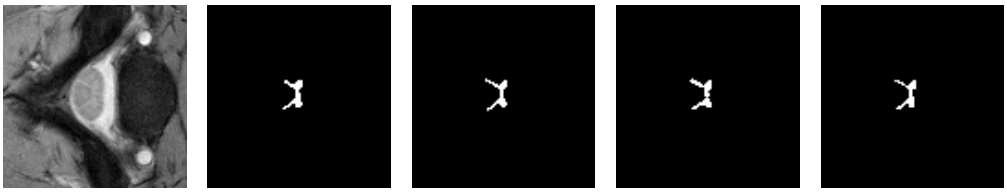


FIGURE 2.2: Spinal cord image and segmentation masks from Site 2.

TABLE 2.1: Dice scores for the model trained on Site 2 using annotations from the first rater, and evaluated by comparing the predictions to annotations made by the same rater (top rows) and annotations made by all raters (bottom rows).

	Original	TV	Flip	Rotate	Elastic	Gauss	Bilateral	Guided	JPEG
Site 1	0.5930	**0.7332**	0.5148	0.4842	0.6484	0.4670	0.5530	0.6050	0.5674
Site 2	0.8337	**0.8610**	0.8254	0.8289	0.8258	0.8341	0.8356	0.8262	0.8567
Site 3	0.5950	0.6466	0.5952	0.6260	0.6260	0.6397	0.6323	0.6207	**0.6605**
Site 4	0.7978	**0.8395**	0.7896	0.8011	0.7960	0.8021	0.8006	0.7909	0.8233
Site 1	0.5695	**0.7044**	0.4934	0.4599	0.6344	0.4533	0.5392	0.5714	0.5366
Site 2	0.8185	**0.8483**	0.8129	0.8158	0.8218	0.8209	0.8245	0.8104	0.8435
Site 3	0.6707	0.7475	0.6664	0.7012	0.7109	0.7182	0.7090	0.6960	**0.7582**
Site 4	0.7776	**0.8184**	0.7751	0.7798	0.7843	0.7813	0.7840	0.7708	0.8046

task of the challenge is to segment the grey matter in those images. The publicly available training data includes MR images of 10 subjects per site for a total of 40 subjects. Each MR image was annotated by four raters. A sample image from site 2 and its four masks are shown in Figure 2.2.

Setup The data was split into 80% training and 20% test sets. To evaluate the effect of TV augmentation under domain shift, we train on data from only one site (Montreal) and report results on the test sets from all sites. Since the four sites have different slice resolutions (0.5 mm, 0.5 mm, 0.25 mm, 0.3 mm), we resample images to the highest resolution. We also standardize intensities.

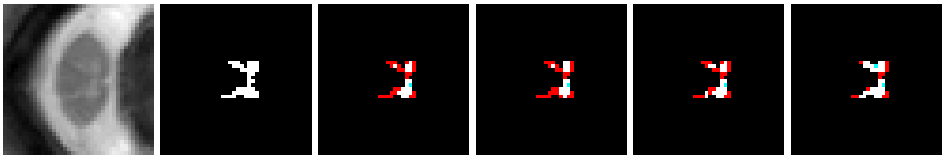
We consider it unusual that each image in this dataset has annotations from four raters. In many other cases, only a single annotation would be available per training image, due to the high cost of creating annotations. We expected that training with multiple annotations per image would provide an additional regularization. To investigate how it interacts with data augmentation, we conducted two sets of experiments. In the first, we train using annotations from the first rater only. In the second, we use annotations from all raters. To facilitate a direct comparison between both, we evaluate each model twice, first by comparing its predictions to annotations from rater one, second by using those from all raters.

Other Augmentation Techniques We compare TV augmentation to random flipping, rotation with a random angle between [-10,10] degrees, and elastic deformations. We also compare against augmentation with Gaussian, bilateral, and guided filters, as well as JPEG compression (Ma et al., 2019). We chose filter parameters visually, so that they result in a smoothed image while not distorting the gray matter shape. Examples are shown in Figure 2.1.

Results Table 2.1 shows the average dice score of each site's test subjects when training using annotations from the first rater. The top rows compare the predictions to annotations made by the same rater, the bottom rows show the average dice across

TABLE 2.2: Dice scores for the model trained on Site 2 using annotations from all raters, and evaluated by comparing the predictions to annotations made by the first rater (top rows) and annotations from all raters (bottom rows).

	Original	TV	Flip	Rotate	Elastic	Gauss	Bilateral	Guided	JPEG
Site 1	0.7936	**0.8254**	0.8227	0.7875	0.7823	0.8238	0.8084	0.8057	0.8236
Site 2	0.8710	**0.8819**	0.8771	0.8729	0.8453	0.8758	0.8786	0.8756	0.8674
Site 3	0.6507	0.6511	0.6582	0.6632	0.6501	0.6620	**0.6680**	0.6546	0.6585
Site 4	0.8480	0.8572	0.8535	0.8486	0.8487	0.8583	0.8544	0.8562	**0.8610**
Site 1	0.7827	**0.8303**	0.8083	0.7617	0.7754	0.8096	0.8000	0.7912	0.8048
Site 2	0.8786	**0.8956**	0.8869	0.8750	0.8554	0.8866	0.8846	0.8796	0.8790
Site 3	0.7672	0.7651	0.7734	0.7768	0.7617	0.7763	**0.7790**	0.7689	0.7761
Site 4	0.8430	**0.8562**	0.8510	0.8424	0.8445	0.8549	0.8497	0.8494	0.8561



(A) (B) (C) (D) (E) (F)

FIGURE 2.3: Sample image from Site 1 (A), its ground truth (B), the prediction of a model trained with no augmentation (C), or with bilateral (D), jpeg (E), and TV augmentation (F), respectively. White, red, and blue colors indicate TP, FN, and FP, respectively. Quantitative performance is shown in Table 2.1.

all four raters.

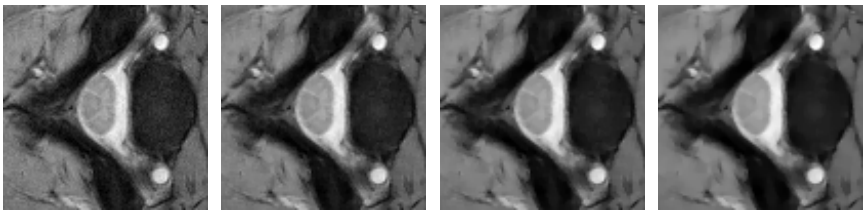
In Table 2.2, we train using annotations from all four raters. At training time, each input image is replicated four times with a different mask for each input. We again evaluate the results by comparing the predictions to the annotations provided by the first rater (top) and annotations from all raters (bottom).

Discussion TV augmentation improved results almost always, by a substantial margin in some of the cases that involved a domain shift. We show qualitative results of a challenging image from Site 1 in Figure 2.3. Most of the time, TV performed better than any other augmentation technique. In the few cases where it did not, the conceptually similar bilateral or JPEG augmentation worked best. As expected, augmentation with Gaussian smoothing did not lead to competitive results. For Site 3, we observed that TV sometimes smoothes out very fine details in the spinal cord structure. This might explain why JPEG augmentation produced slightly higher dice scores than TV on that site.

Traditional augmentation techniques such as flipping, rotation, and elastic deformation did not show a clear benefit in this specific task. Even though it is possible to combine them with TV augmentation, our results suggest that it is unlikely to benefit this particular task. It is thus left for future work.

Comparing Table 2.2 to Table 2.1, we can see the largest benefits when annotations by only one rater are available at training time. This agrees with our intuition that repeated annotations provide a form of regularization. Despite this, we continue seeing a benefit from data augmentation.

Impact of smoothing parameter and multi scale augmentation The effect of the



(A) $\alpha = 0$ (B) $\alpha = 5$ (C) $\alpha = 10$ (D) $\alpha = 15$

FIGURE 2.4: Sample image and the results of different TV smoothing parameters.

TABLE 2.3: Results with varying smoothing parameters suggest that TV augmentation is robust to its choice, and combining multiple scales is a feasible strategy. Dice scores on the left are from the same rater, on the right from all raters.

	$\alpha = 5$	$\alpha = 10$	$\alpha = 15$	Combined
Site 1	0.7332	0.7423	0.7573	**0.8187**
Site 2	0.8610	0.8653	0.8599	**0.8757**
Site 3	0.6466	**0.6581**	0.6579	0.6487
Site 4	0.8395	0.8457	0.8428	**0.8525**
	$\alpha = 5$	$\alpha = 10$	$\alpha = 15$	Combined
Site 1	0.7044	0.7130	0.7305	**0.7848**
Site 2	0.8483	0.8530	0.8536	**0.8749**
Site 3	0.7475	0.7576	0.7574	**0.7633**
Site 4	0.8184	0.8217	0.8215	**0.8378**

TV smoothing parameter α that was discussed in Section 2.4.3 is illustrated in Figure 2.4. Previous experiments used $\alpha = 5$. Table 2.3 studies how sensitive TV augmentation is with respect to this parameter. Again, training was on Site 2 using annotations from the first rater, and evaluation compared the predictions to annotations made by the same rater (left) and annotations made by all raters (right). All scales perform quite well, and combining different ones, as shown in the last column, yields the best dice score in nearly all cases.

2.5.2 BraTS 2019

This challenge data consists of brain MRI scans with high- and low-grade gliomas, collected at different institutions. The publicly available training data includes 3D scans of 335 subjects.

Setup We again aimed for an evaluation that would separately assess the benefit with or without a domain shift. Although there was no explicit mapping of individual subjects to their respective institutions, we could identify four groups based on the challenge data description and the filenames: Brats2013 (30 subjects), CBICA (129 subjects), TCGA (167 subjects), and TMC (9 subjects).

We split each group's data into 80% training and 20% test sets. We train on data from only one group (TMC) and report results on the test sets from all groups. An example image with the ground truth and the smoothed augmentation is shown in Figure 2.5.

As input we use the FLAIR modality and train the model to learn the Whole Tumor label. We follow the preprocessing steps of (Isensee et al., 2017) and standardize intensities to zero mean and unit variance.

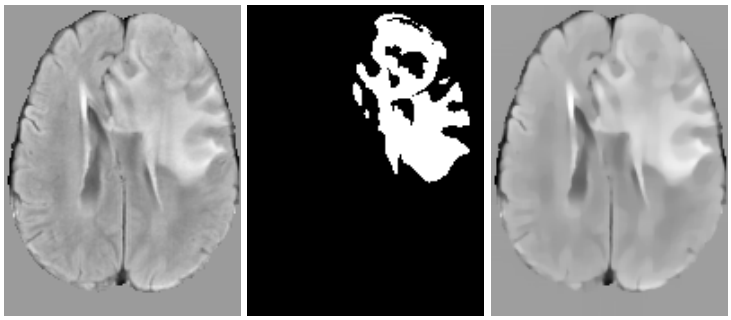


FIGURE 2.5: BraTS (TMC) training image, ground truth, and the TV smoothed image.

TABLE 2.4: Dice Score on held-out test sets of different subsets of BraTS 19.

	Br13	CBICA	TCGA	TMC
Original	0.8088	0.7933	**0.7929**	0.7894
TV Smoothed	**0.8570**	**0.8152**	0.7885	**0.8233**

Results As shown in Table 2.4, TV augmentation increased segmentation accuracy in three out of the four groups. The largest benefit was observed in the group Br13. This involved a domain shift, as the model was only trained on TMC. On the group where the performance slightly decreases (TCGA), we found that the predicted segmentation sometimes misses some of the finer tumor details. Such fine structures might be more difficult to discern after TV smoothing.

2.5.3 White Matter Hyperintensity

The publicly available data from the WMH challenge is acquired with different scanners from 3 institutions. 2D FLAIR and T1 MR images and segmentation masks are provided for 60 subjects, 20 from each institution.

Setup The data is split into 80% training and 20% test sets. We combined the training data from all three sites, and report results on the test sets we held out ourselves, as well as on the official test sets, by submitting to the challenge website. To avoid creating a large number of submissions, we do not investigate the impact of training on one compared to all sites in this case. We follow the preprocessing steps of (Li et al., 2018) and standardize intensities. An example image with the ground truth and TV smoothed augmentation is shown in Figure 2.6.

Results Table 2.5 shows that TV augmentation improved segmentation accuracy on the held-out data in all three sites. Table 2.6 shows a clear improvement also on

FIGURE 2.6: WMH site 1 training image, ground-truth, and the TV smoothed image.

TABLE 2.5: Dice Score on held-out test sets of WMH challenge

	Site 1	Site 2	Site 3
Original	0.6879	0.8348	0.7170
TV Smoothed	**0.7329**	**0.8480**	**0.7685**

TABLE 2.6: Results on the official test set of the challenge. Higher values for DSC, Recall, F1 are better, and lower values for H95, AVD are better.

	DSC	H95	AVD	Recall	F1
Original	0.74	9.05	29.73	0.65	0.64
TV Smoothed	**0.77**	**7.42**	**24.97**	**0.76**	**0.67**

the official test set. The evaluation criteria in this challenge are Dice Score (DSC), Hausdorff distance (H95), Average Volume Difference (AVD), Recall for individual lesions, and F1 score for individual lesions.

Discussion Results on both the held-out test set and the official test set show that TV smoothing improves the segmentation performance, in some cases by a substantial margin. TV-smoothing performs better with respect to all evaluation criteria in the detailed official results. The most pronounced improvement is in the number of lesions that the model detects (recall).

2.6 Conclusion

Our results indicate a clear benefit from using feature preserving smoothing for data augmentation when training CNN-based medical image segmentation on limited data. Advantages were especially pronounced when using TV smoothing, which creates a piecewise constant, segmentation-like output. TV augmentation also helped when a domain shift between training and test data was involved. Consequently, we propose that TV smoothing can be used as a relatively simple and inexpensive data augmentation method for medical image segmentation.

In the future, we hope to better characterize the exact conditions under which the different augmentation techniques that have been proposed for semantic segmentation work well, and when it makes sense to combine them. We expect that factors such as the nature of differences between training and test images will play a role, as well as characteristics of the images (e.g., noise), and the structures that should be segmented (e.g., presence of fine details).

Chapter 3

Adaptive Optimization with Fewer Epochs Improves Across-Scanner Generalization of U-Net based Medical Image Segmentation

3.1 Summary

There are many design choices that go into building strong deep learning models that perform well. We investigate in this work two of those choices: the type of optimizer and how long the model is trained for. The popular nnUNet (Isensee et al., 2021) framework uses the SGD optimizer and trains models for 1000 epochs. We hypothesized that reducing this large number of epochs might be helpful for generalization across domains. Recent work (Savarese et al., 2021) has shown that adaptive optimizers can have a superior performance to non-adaptive ones provided that that their hyperparameters are properly tuned. We therefore also look into how the performance in presence of domain shift changes when adaptive optimizers are used.

To evaluate our approach, we use two datasets. The first one is the Calgary Campinas dataset (Souza et al., 2018), which consists of brain MR images obtained with several scanner types and field strengths, resulting in six domains. The task is to output skull-stripped segmentation masks of the brain. The second dataset we evaluate on is the Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation (M&Ms) dataset (Campello et al., 2021). It contains cardiac MR images obtained with different scanners translating into four domains. The task is to segment the left ventricle cavity, right ventricle cavity, and left ventricle myocardium.

We use the nnUNet (Isensee et al., 2021) as our segmentation framework. The network is trained with a combination of the dice loss and the cross-entropy loss. We evaluate the performance using the Surface Dice score (Nikolov et al., 2018) for the Calgary Campinas dataset and the usual Dice score for the M&Ms dataset.

In addition to SGD and Adam as optimizers, we use AvaGrad (Savarese et al., 2021), which is an adaptive optimizer that decouples the learning rate from the adaptability parameter. This is desirable because it simplifies the hyperparameter search for the learning rate in comparison to the Adam optimizer.

We trained models with the different optimizers and for a varying number of epochs, and found that on the Calgary Campinas dataset and on the M&Ms dataset, models generalize better to target domains when combining adaptive optimizers with training for a smaller number of epochs compared to the default settings of nnUNet which uses SGD and trains for 1000 epochs.

When investigating the average weight update for each layer of the network, we found that they are comparable for AvaGrad whereas there is a larger variance in case of SGD. This might be a reason why training for a smaller number of epochs with SGD does not produce the same improvement as with AvaGrad since those layers that learn more slowly will be negatively affected by early stopping.

Since augmentations reduce overfitting and help models generalize, we ran an ablation study where we disable augmentations to see what effect that will have on models trained for a small number of epochs. The results we obtained show that a larger drop in performance occurs in case of SGD compared to AvaGrad when training with no augmentations and with early stopping.

We also ran experiments with the SpotTUnet (Zakazov et al., 2021), which defines a policy that automatically decides which layers will stay fixed and which ones will be fine-tuned with data from the target domain. We first trained a base model using the framework defined in their work and observe a similar pattern where AvaGrad performs better than SGD. We then fine-tuned on the target domain, once with a small number of slices, and then with a larger number of slices, similar to the approach Zakazov et al., 2021 followed in their work, and found that the policies regarding which layers are fine-tuned differ with the change of the optimizer.

The work presented in this chapter is related to the robustness aspect of segmentation models in the presence of domain shift, that we address in this dissertation. It is highly desirable that models are trained to be generalizable in the first place and not overfit on the source domain. We show that the default settings of the popular nnUNet framework regarding the optimizer and number of epochs might not be optimal if domain shift is present, and argue instead for the use of adaptive optimizers in combination with early stopping.

The content of this chapter is published as:

Rasha Sheikh et al. (2022). "Adaptive Optimization with Fewer Epochs Improves Across-Scanner Generalization of U-Net Based Medical Image Segmentation". In: *Domain Adaptation and Representation Transfer - 4th MICCAI Workshop, DART 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings*. Ed. by Konstantinos Kamnitsas et al. Vol. 13542. Lecture Notes in Computer Science. Springer, pp. 119–128. DOI: 10.1007/978-3-031-16852-9_12. URL: https://doi.org/10.1007/978-3-031-16852-9_12.

Contribution of the thesis author: Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization.

3.2 Abstract

The U-Net architecture is widely used for medical image segmentation. However, accuracy has been observed to drop, sometimes dramatically, when U-Nets are trained on images that have been acquired with a specific scanner, and are applied to images from another scanner. This indicates an overfitting to image characteristics that are irrelevant to the semantic contents, and is usually mitigated with data augmentation. We argue that early stopping additionally improves across-scanner generalization, while greatly reducing training times. For this, we first observe that the widely used stochastic gradient descent (SGD) trains different U-Net layers at different speeds, and demonstrate that this problem is reduced by switching to AvaGrad, a recently

proposed adaptive optimizer. On two different datasets, this allows us to match accuracies from nnUNets with default settings, 1000 epochs of SGD, by training for only 50 epochs with AvaGrad, and to exceed their results in the across-scanner setting. This benefit is specific to combining adaptive optimization and early stopping, since it can be matched neither by SGD with a low number of epochs, nor by Avagrad with many epochs. Finally, we demonstrate that the choice of optimizer can have important implications for domain adaptation. In particular, the SpotTUnet, which was recently proposed to automatically select layers for fine-tuning, arrives at very different policies depending on the optimizer.

3.3 Introduction

The U-Net architecture (Ronneberger et al., 2015) has achieved state-of-the-art results for many different medical image segmentation tasks. However, its ability to generalize to images that have been acquired with a different scanner than the images it was trained on is often limited. In practice, generalization depends on many factors, including the network's architecture, how the input data is pre-processed, and the learning scheme. Our work investigates two of these factors, the type of optimizer and the stopping time. We suggest that widely used settings for them are not optimal with respect to across-scanner generalization, and that this has important implications for domain adaptation techniques.

The nnU-Net framework (Isensee et al., 2021) automatically configures itself for different segmentation tasks (Kavur et al., 2021; Heller et al., 2021). Some of the design choices in the framework are adapted to the dataset that the model is trained on, whereas others were found to work well in different challenges and are fixed. These include having SGD as the optimizer with a learning rate of 0.01, and training for a long time, namely, 1000 epochs. Our work is based on the intuition that stopping earlier, which is widely used as a way of regularizing deep networks, should reduce overfitting and therefore improve across-scanner generalization. However, we found that SGD with few epochs does not sufficiently train all U-Net layers, due to a very different effective speed at which different layers are trained. To mitigate this problem, we investigate adaptive optimization.

Adam and SGD are two of the most commonly used methods in the optimization of a network's parameters during training. For vision tasks with convolutional networks (Chen et al., 2018; Isensee et al., 2021), non-adaptive optimizers, such as SGD, are frequently used with the belief that they generalize better to unseen data (Wilson et al., 2017). However, recent work (Choi et al., 2019; Savarese et al., 2021) has shown that adaptive methods can actually outperform non-adaptive ones if they are properly tuned. AvaGrad (Savarese et al., 2021) is a recently published adaptive optimizer. It is similar in principle to Adam, where running averages of the gradients and their squared values are used to update the network's weights. However, it is different in that it decouples the effective learning rate and the adaptability parameter ϵ. If ϵ is large enough, the only parameter left to tune is the learning rate, which makes the cost of the hyperparameter search similar to that with SGD.

In our experiments, we replace SGD with AvaGrad, fix the learning rate and ϵ, and train for a relatively short number of epochs. We test the performance of the trained models on two datasets that exhibit a domain shift between source and target domains, and observe an improvement in performance on both of them. We also show that the choice of optimizer has a great influence on the fine-tuning policies learned by Zakazov et al., 2021.

3.4 Materials and Methods

3.4.1 Datasets

We selected two public datasets for our experiments, the Calgary-Campinas-359 (CC-359) dataset (Souza et al., 2018) and the Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Image Segmentation (M&Ms) dataset (Campello et al., 2021). They both provide MR images that have been acquired with different scanners, which we can use as the separate domains in our experiments. The **CC-359** dataset consists of 359 3D brain MR images collected from sites with scanners that differ in their type and field strength, resulting in 6 domains: GE 15, GE 3, Philips 15, Philips 3, Siemens 15, and Siemens 3. We use GE 3 as our source domain and the others as the target domains. The groundtruth for this dataset corresponds to skull-stripping segmentation masks. The **M&Ms** dataset consists of cardiac MR images from 345 subjects. These were collected from sites with different scanners: Siemens (A), Philips (B), GE (C), and Canon (D). We use domain B as our source domain and the others as the target domains. The segmented regions in this dataset are the left ventricle cavity (LV), the right ventricle cavity (RV), and the left ventricle myocardium (MYO).

3.4.2 Segmentation Framework

We use the well-known and established nnUNet (Isensee et al., 2021) as our segmentation pipeline. Its backbone is based on the U-Net (Ronneberger et al., 2015) architecture with an encoder and decoder-like structure and skip connections in between. The depth of the network. i.e. number of down/upsampling operations is adaptive and depends on the dataset median image size. As a preprocessing step, the images are normalized to follow a standard distribution. Training data is augmented with various transformations including rotation, scaling, Gaussian blurring, and Gamma augmentation among others. The network is trained with the SGD optimizer with a poly learning rate scheduler, and it uses both the dice and cross-entropy as the loss function. In our experiments we focus on the 2D variant of nnUNet. To evaluate the performance, we use the Dice score for the M&Ms dataset, $\text{Dice} = 2(\sum \hat{y}y)/(\sum \hat{y} + \sum y)$, where $\hat{y}$ and y are the predicted and true labels respectively. For the CC-359 dataset, we follow (Shirokikh et al., 2020) and use the surface Dice score (Nikolov et al., 2018) instead. This score computes how much of the predicted and ground-truth surfaces overlap within a given distance tolerance. Since the segmentation task for this dataset is skull-stripping, this measure is deemed more informative because of the larger focus on the brain contour.

3.4.3 AvaGrad Optimizer

Similar to the Adam optimizer, AvaGrad is an adaptive method where the effective learning rate for each parameter (e.g. network weights) is adapted according to the running averages of the corresponding gradients. Differently however, AvaGrad decouples the adaptability parameter ϵ from the learning rate. This is achieved by normalizing the learning rate vectors before using them to update the parameters. In our experiments, for both datasets we use the values 10 for the learning rate and 0.1 for ϵ. These were found using a separate validation set from domain B of the M&Ms dataset. We show below the relevant equations from (Savarese et al., 2021), where w_t and g_t denote the parameter to be updated and its gradient, m_t and v_t denote the running averages of the gradient and the gradient squared, d is the dimension of η_t, and finally α_t and ϵ are the learning rate and the adaptability parameter.

TABLE 3.1: Surface Dice performance on the Calgary-Campinas dataset. Largest means in each row are bold, statistical significance between S-1000 and the other columns is indicated with an asterisk.

Domain	S-1000	S-25	S-50	A-25	A-50	AG-25	AG-50	AG-1000
GE15	0.9068	0.8345*	0.8782*	0.8970*	0.9076	0.9152*	**0.9214***	0.8181*
Philips 15	0.9407	0.9102*	0.9330*	0.9421	0.9472*	**0.9484***	0.9480*	0.8009*
Philips 3	0.8685	0.7252*	0.7773*	0.8570	0.8345*	0.8848*	**0.9032***	0.7135*
Siemens 15	0.9176	0.8409*	0.8841*	0.9206	0.9219*	0.9324*	**0.9344***	0.7734*
Siemens 3	0.8191	0.8449*	0.8579*	0.9037*	0.9029*	**0.9061***	0.8932*	0.8398
GE 3 Test	0.9645	0.9594	0.9629	0.9622	0.9646	**0.9650**	0.9641	0.9324*

$$m_t = \beta_{1,t} m_{t-1} + (1-\beta_{1,t}) g_t, \qquad \eta_t = \frac{1}{\sqrt{v_{t-1}+\epsilon}},$$
$$w_{t+1} = w_t - \alpha_t \frac{\eta_t}{\|\eta_t/\sqrt{d}\|_2} \odot m_t, \qquad v_t = \beta_{2,t} v_{t-1} + (1-\beta_{2,t}) g_t^2$$

3.4.4 SpotTUNet

If we have annotated data from target domains, we can fine-tune the source-domain model on each target. Zakazov et al., 2021 proposed the SpotTUnet, where an additional ResNet-34 is employed to automatically choose which layers should be fine-tuned. For that, we extract the input features, which get reduced to a 2×32 dimensional output. Each of the 64 output logits is passed through a Gumbel-Softmax (Jang et al., 2016) which would then be mapped to a probability for a UNet layer to be chosen for fine-tuning or to remain frozen. The original SpotTune also proposed a regularization constraint for the global policy aimed at consolidating the global fine-tuning policy. The SpotTUNet loss with added penalty term is defined as $\mathcal{L} = \mathcal{L}_{segm} + \lambda \sum_{l=1}^{64}(1 - I_l(x))$, where $I_l(x)$ is the binary indicator for the l-th frozen layer based on the image input x. We evaluate the SpotTune performance for 7 and 800 target domain slices available for fine-tuning.

3.5 Experiments

3.5.1 Baseline Performance and Early-Stopping

We show in Tables 3.1 and 3.2 the baseline performance with nnUNet in the first column. We refer to this experiment as S-1000 since the default configuration of nnUnet uses the SGD optimizer and trains for 1000 epochs. We next show the performance of the adaptive optimizers, Adam (A) and AvaGrad (AG), with fewer epochs. For comparison, we also train with SGD for the same number of epochs, and with AvaGrad for 1000 epochs. No labels from the target domains were used. On the CC dataset, AvaGrad with few epochs achieves the highest mean scores on all domains. According to a Bonferroni corrected Wilcoxon signed-rank test, all differences between S-1000 and AG-50, which we propose as an improved default, are statistically significant. This improvement is achieved at greatly reduced computational cost, due to a much smaller number of epochs.

Similarly, on the M&Ms dataset (Table 3.2), we either observe a benefit or a comparable performance from using AvaGrad with few epochs. Qualitative results from both datasets can be found in Figure 3.1. The Supplementary Material also includes violin plots that show the distribution of results across the different subjects on both datasets.

TABLE 3.2: Volumetric Dice performance on the M&Ms dataset. Largest means in each row are bold, statistical significance between S-1000 and the other columns is indicated with an asterisk.

	Class1 (LV)							
Domain	S-1000	S-25	S-50	A-25	A-50	AG-25	AG-50	AG-1000
A	0.6065	0.6957	0.6663	0.7131	0.7506*	0.7524*	**0.7845***	0.7355*
C	0.8700	0.8660	0.8693	0.8784	0.8758	**0.8862***	0.8754	0.8734
D	0.8805	0.8833	0.8829	0.8767	0.8745	**0.8919**	0.8866	0.8912
B Test	0.8850	0.8977	0.8932	0.9019	0.9024	0.8988	**0.9063**	0.8937
	Class2 (MYO)							
Domain	S-1000	S-25	S-50	A-25	A-50	AG-25	AG-50	AG-1000
A	0.5335	0.5697	0.5468	0.5785	0.6261	0.6200*	**0.6644***	0.6400*
C	0.8064	0.7994	0.8015	0.8091	0.8010	**0.8170**	0.8143	0.8143
D	0.8127	0.8031	0.8084	0.8126	0.8121	0.8112	0.8134	**0.8163**
B Test	0.8463	0.8461	0.8468	0.8569	0.8561	0.8537	**0.8625**	0.8510
	Class3 (RV)							
Domain	S-1000	S-25	S-50	A-25	A-50	AG-25	AG-50	AG-1000
A	0.5000	0.5578	0.5058	0.5821	0.6392*	0.6187	**0.6973***	0.6582*
C	0.8093	0.8196	0.8179	0.8045	0.8241	0.8311	**0.8451**	0.8305
D	0.8028	0.7682*	0.7645*	0.7766	0.8007	0.8027	0.8041	**0.8454**
B Test	0.8635	0.8535	0.8534	0.8535	0.8570	0.8619	**0.8714**	0.8627

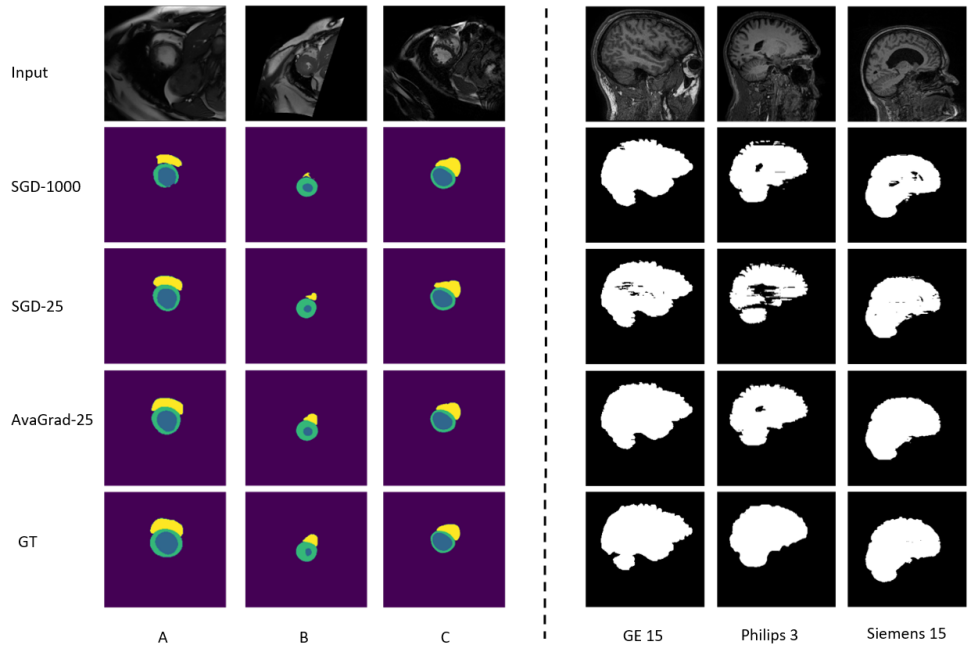


FIGURE 3.1: Qualitative results on the M&Ms dataset on the left (yellow: RV, blue: LV, green: MYO), and on the Calgary-Campinas dataset on the right.

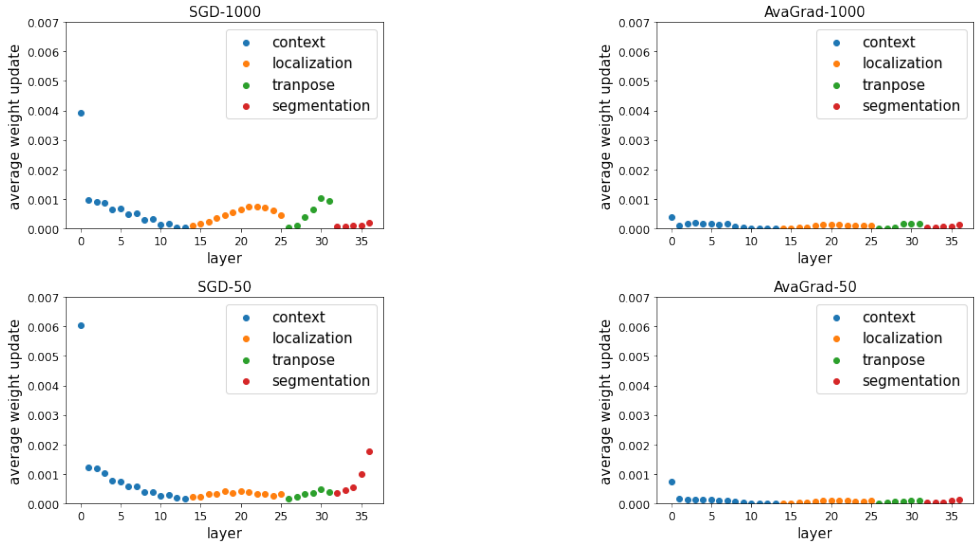


FIGURE 3.2: Average weight update for SGD on the left and AvaGrad on the right.

We note that, throughout this work, we use the term *early stopping* to simply denote training for much fewer epochs than the nnUNet default. Introducing a mechanism that fine-tunes the number of epochs for optimal generalization across multiple target domains seems difficult, given that results in Tables 3.1 and 3.2 indicate that the perfect stopping time depends on the specific target domain. Moreover, in practice, the target domain might not even be known while training the original model.

3.5.2 Training Speed of U-Net Layers

We plot in Figure 3.2 the average magnitude of weight updates per layer for the SGD and AvaGrad optimizers, grouped by the type of convolution defined by the nnUNet. *context* denotes those convolutions in the encoder part of the model, whereas *transpose* refers to the transposed convolutions in the decoder part of the model. *localization* denotes the convolutions that follow the concatenation of feature maps, and finally *segmentation* refers to the convolutions that produce the segmentation maps at different levels of the network. We observe that particularly in the case of SGD, the U-Net layers train at different speeds. This could explain why reducing the number of epochs when training with SGD often works less well, since it might not sufficiently train the low-resolution layers, which learn more slowly. When using the AvaGrad optimizer, the layers are trained at rather comparable speed. We believe that this allows early stopping to reduce overfitting to the source domain, while still mitigating the risk of underfitting individual layers.

3.5.3 No Augmentation

The nnUNet augments the training data with transformations that simulate plausible differences between scanners. We investigate the relative benefits from augmentation and early stopping through an ablation study that deactivates those augmentations. Table 3.3 summarizes these results for the Calgary-Campinas dataset. The corresponding table for the M&Ms dataset can be found in the Supplementary Material. Compared to Tables 3.1 and 3.2, we observe that across-scanner generalization

TABLE 3.3: Surface dice performance on the CC dataset when training with no augmentations

Domain	S-1000-noAug	S-25-noAug	A-25-noAug	AG-25-noAug
GE15	0.66380	0.76537	**0.83760**	0.81029
Philips 15	0.85827	0.87022	0.89176	**0.91129**
Philips 3	0.69653	0.62460	0.65254	**0.80440**
Siemens 15	0.80643	0.80820	0.85617	**0.88390**
Siemens 3	0.63098	**0.70149**	0.57215	0.63266
GE 3 Test	0.92995	0.94423	0.94547	0.95422

suffers drastically from this when training for 1000 epochs. This effect is much reduced when training with AvaGrad and early stopping. Best results are obtained when combining both types of regularization.

3.5.4 SpotTUNet

We evaluate the differences between SGD and AvaGrad under several SpotTUNet settings. For direct comparison, we retain the same architecture and experimental setup as provided by Zakazov et al., 2021[1], where the baseline model was trained for 60 epochs and the spottuned model for 100 epochs. Figure 3.3 shows that using the surface dice as metric and the SGD as an optimizer, we can spottune the Target Domains (TDs) individually to increase the average target domain performance. Changing the optimizer to AvaGrad immediately achieves an increase of 28.1% surface dice across the TDs. Like with SGD, spottuning with AvaGrad further improves the surface dice. We plot the corresponding SpotTUNet policy visualizations in Figures 3.5a and 3.5b. For SGD, they agree with the finding of Zakazov et al., 2021 in that early encoder layers get fine-tuned. This pattern changes with AvaGrad, where fine-tuning focuses on the lowest-resolution layers.

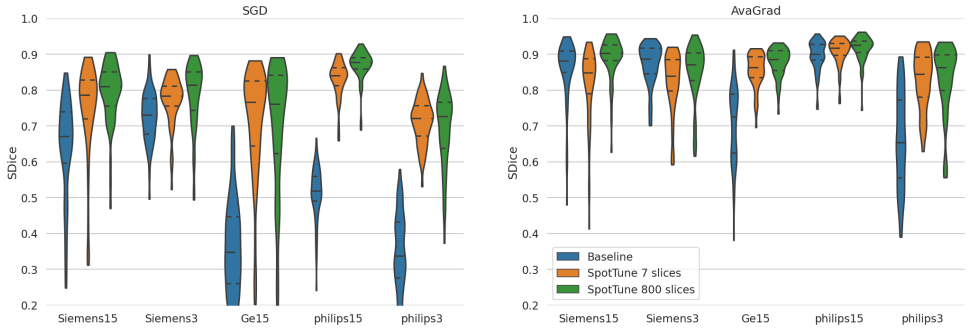


FIGURE 3.3: SpotTUNet performance on the Calgary-Campinas dataset.

To spottune on the M&Ms dataset, we use B as the source domain and fine-tune the other domains as targets. Again, changing to AvaGrad increases the baseline performance on the TDs. However, due to the comparatively smaller number of annotated training slices and stronger inhomogeneity between the domains, the models overfit and degenerate for both SGD and AvaGrad using only 7 slices (Figure 3.4). That is further highlighted by the policy visualization in Figure 3.5c as both SGD and AvaGrad (7 slices) fine-tune layers across the whole model. Refinement with few

[1]https://github.com/neuro-ml/domain_shift_anatomy

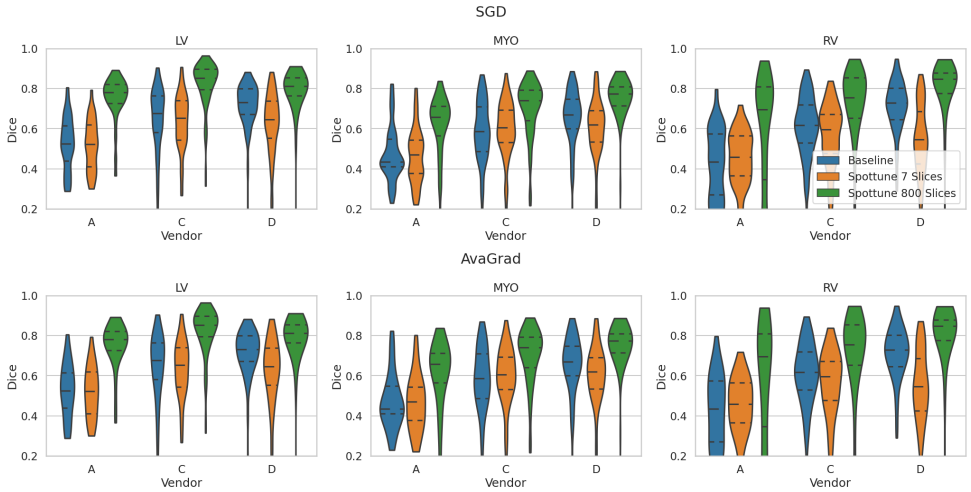


FIGURE 3.4: SpotTUNet performance on the M&Ms dataset.

slices introduces a risk of overfitting. We believe that, with the current Spottuning method, this sometimes outweighs the benefit of introducing domain-specific information. On the other hand, when providing 800 annotated slices for spottuning, we achieve higher average dice scores across all TDs, while AvaGrad still outperforms SGD. The corresponding Tables for both datasets can be found in the Supplementary Material.

3.6 Conclusion

In this work, we argue that a widely used training strategy for U-Nets, SGD with a large number of epochs, is not optimal with respect to generalization to other domains. In particular, we demonstrate that the use of adaptive optimizers together with early stopping improves generalization across scanners on two different datasets. An additional advantage of early stopping is computational efficiency. This is especially beneficial in cases which involve frequent re-training, such as ensembling or automated configuration. Finally, we show that the choice of optimizer influences the policies that are used to fine-tune models when annotated target data is available.

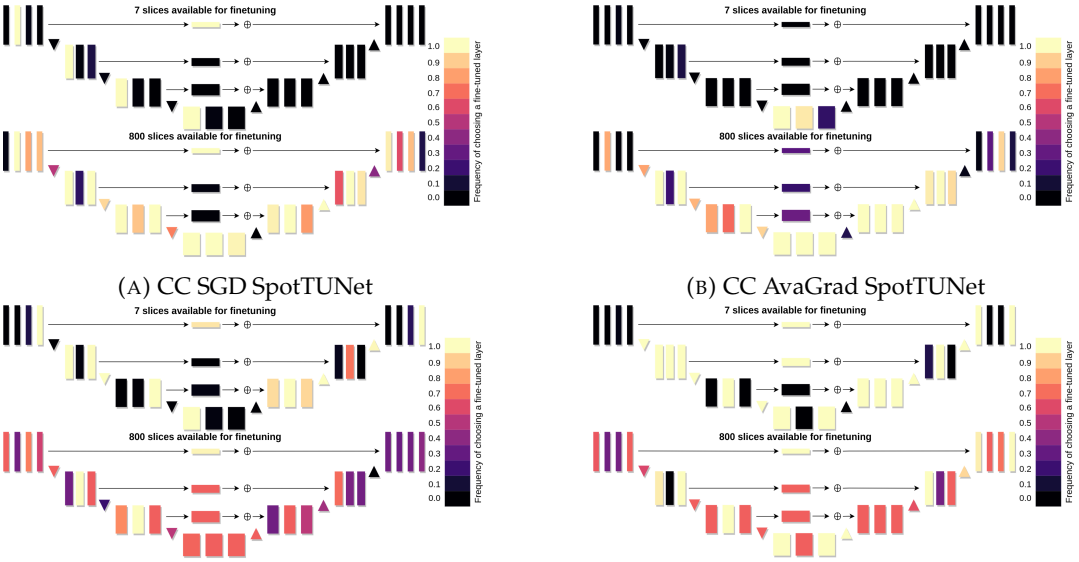
3.7 Supplementary Material

3.7.1 No Augmentation

We show in Table 3.4 the effect of adding no augmentations when training nnUNet models on the M&Ms dataset for a small number of epochs with different optimizers.

3.7.2 Performance with Early Stopping

We show in Figures 3.6 and 3.7 the distribution of the dice and surface dice results for different models trained on each dataset.



(A) CC SGD SpotTUNet

(B) CC AvaGrad SpotTUNet

(C) M&Ms SGD SpotTUNet

(D) M&Ms AvaGrad SpotTUNet

FIGURE 3.5: Visualization of the learned SpotTUnet policy for the CC and M&Ms dataset.

TABLE 3.4: Dice performance on the M&Ms dataset when training with no augmentations

	Class1 (LV)				Class2 (MYO)			
Domain	S-1000	S-25	A-25	AG-25	S-1000	S-25	A-25	AG-25
A	0.57821	0.63325	**0.67412**	0.65944	0.52726	0.41326	**0.55698**	0.50166
C	0.81005	0.79042	0.81621	**0.82020**	**0.75339**	0.69092	0.73445	0.74761
D	0.86613	0.84345	0.85194	**0.87133**	**0.79136**	0.71048	0.77167	0.78518
B Test	0.88666	0.86221	0.87382	0.87754	0.83076	0.79325	0.81404	0.82278

	Class3 (RV)			
Domain	S-1000	S-25	A-25	AG-25
A	0.55301	0.43433	0.50285	**0.58287**
C	**0.71715**	0.64833	0.66497	0.70351
D	**0.78822**	0.65803	0.69090	0.78279
B Test	0.82888	0.80085	0.81276	0.84376

3.7.3 SpotTUNet

We show in Tables 3.5 and 3.6 the quantitative performance of the SpotTUNet models.

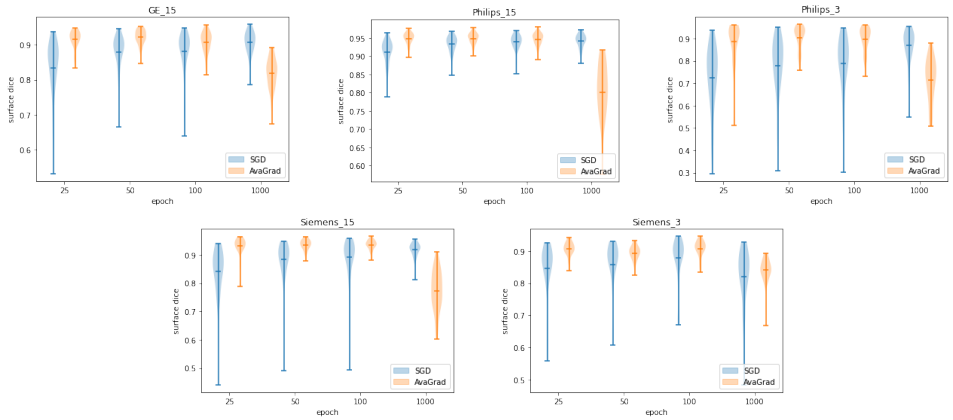


FIGURE 3.6: Performance of SGD and AvaGrad at different epochs on the Calgary-Campinas dataset.

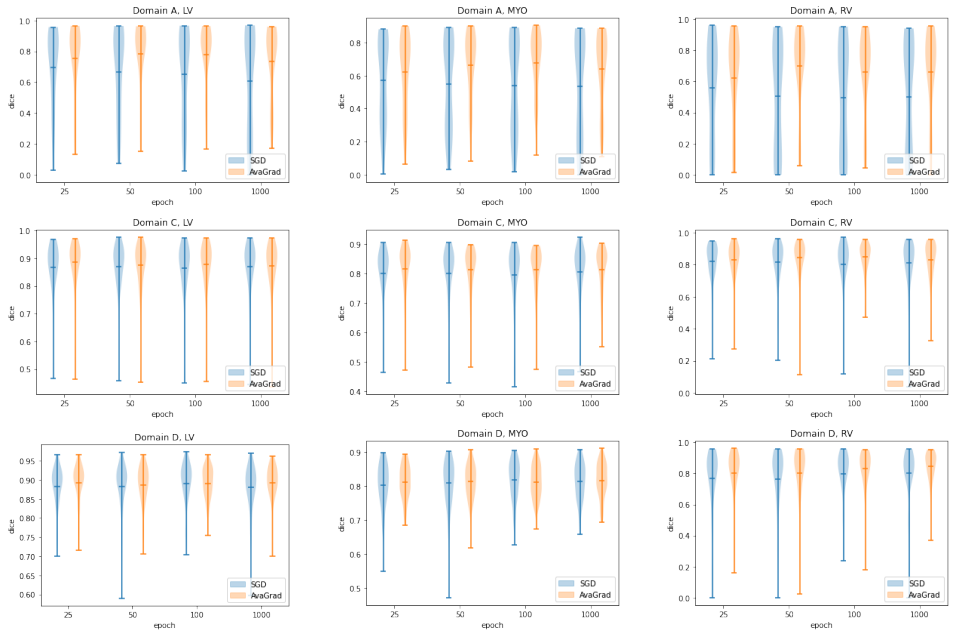


FIGURE 3.7: Performance of SGD and AvaGrad at different epochs on the M&Ms dataset.

TABLE 3.5: Comparing SGD and AvaGrad on CC using SpotTUNet

GE3 SDice SpotTUNet	**TD slices**	**Sms1.5**	**Sms3**	**GE1.5**	**Phi1.5**	**Phi3**	**Avg.**	**SDice-Δ**
SGD Baseline	-	0.640	0.726	0.358	0.513	0.352	**0.517**	-
SGD	7	0.750	0.771	0.712	0.830	0.711	**0.754**	+0.237
SpotTune	800	0.794	0.789	0.700	0.870	0.689	**0.768**	+0.250
AvaGrad Baseline	-	0.858	0.873	0.706	0.896	0.661	**0.798**	-
AvaGrad	7	0.816	0.825	0.856	0.907	0.828	**0.846**	+0.048
Spottune	800	0.890	0.855	0.878	0.916	0.830	**0.874**	+0.075

TABLE 3.6: Comparing SGD and AvaGrad on M&Ms using SpotTUNet

SGD SpotTUNet - Dice									
	Class1 (LV)	Class2 (MYO)	Class3 (RV)	Class1 (LV)	Class2 (MYO)	Class3 (RV)	Class1 (LV)	Class2 (MYO)	Class3 (RV)
Domain	SGD-50-Base			SpotTUnet-7 Slices			SpotTUnet-800 Slices		
A	0.527	0.472	0.422	0.518	0.465	0.458	**0.717**	**0.600**	**0.569**
C	0.611	0.552	0.556	0.635	0.599	0.560	**0.823**	**0.704**	**0.693**
D	0.655	0.596	0.621	0.627	0.609	0.548	**0.770**	**0.735**	**0.775**
AvaGrad SpotTUNet - Dice									
	Class1 (LV)	Class2 (MYO)	Class3 (RV)	Class1 (LV)	Class2 (MYO)	Class3 (RV)	Class1 (LV)	Class2 (MYO)	Class3 (RV)
Domain	AvaGrad-50-Base			SpotTUnet-7 Slices			SpotTUnet-800 Slices		
A	0.658	0.589	0.635	0.531	0.501	0.492	**0.754**	**0.662**	**0.625**
C	0.718	0.676	0.710	0.659	0.622	0.661	**0.835**	**0.729**	**0.721**
D	0.746	0.704	0.746	0.558	0.702	0.272	**0.765**	**0.739**	**0.783**

Chapter 4

Gradient and Log-based Active Learning for Semantic Segmentation of Crop and Weed for Agricultural Robots

4.1 Summary

Deep learning models that were trained for the task of semantic segmentation might perform well on data similar to what they were trained on but not generalize as well to other data with different characteristics. Our target application in this work is agricultural robots that should distinguish between crop and weed in order to eliminate weed plants. Image data captured from different agricultural fields will exhibit variations in their appearance due to location-dependent soil and weather conditions. The statistics of the semantic classes might also be different. These are some of the reasons that make it difficult for trained models to perform similarly on source and target data. A straightforward solution would be to annotate data from new domains and retrain the model on them. Annotating data is however tedious, therefore our goal in this work was to reduce the amount of data needed for annotation by selecting those that the model might most benefit from.

To pick samples for annotation, we compare the model's predictions to pseudo-groundtruth labels. To generate the latter, we run k-means on the RGB images to obtain clusters that only require weak supervision to be transformed into foreground-background segmentation. We found that this simple clustering is already enough to get reasonable segmentation that we can use for active learning. We then devised three methods to choose which samples should be annotated. The first computes the contribution of each sample to the network loss. Those with the highest loss indicate a large discrepancy between their pseudo-groundtruth and the model's prediction. Choosing only the samples with the highest loss, however, runs the risk of picking samples that might be too similar to each other. We therefore sort the losses of samples in a descending order then select samples on a logarithmic scale. This encourages diversity of the samples while still favoring those that the model is struggling with.

Our second and third methods compute the gradients of the network and makes use of them in selecting samples. The intuition behind that is that samples with a large norm of gradients might have a larger influence on the network's weights. In our second method, once gradients are computed, their norm is sorted in a descending order, and again the samples are selected on a logarithmic scale. We use a different technique to encourage diversity in our third method. The first sample

selected is the one with the largest gradient norm. The gradient of each candidate sample is projected onto that of the already selected samples, then subtracted from the original gradient. The sample with the highest residual gradient norm is chosen next. This encourages the selection of samples that are different from what has already been picked for annotation.

The neural network model we use follows an encoder-decoder architecture with residual blocks and ReLU as the non-linearity activation. To train and evaluate our model, we used imaging data acquired by agricultural robots in three fields: Bonn and Stuttgart in Germany and Zurich in Switzerland. The datasets exhibit different crop and weed statistics and also have different appearances. We train our model on the Bonn dataset then refine it on the two other datasets (separately) by picking 10 samples for annotation in each round and subsequently refining the model using these samples. The focal loss was used to train the network. To evaluate the performance, we use two measures. The first one is pixel-wise mean Intersection over Union (mIoU). The second one is driven by the particular application here which is weeding, therefore the object-wise accuracy of objects larger than 50 pixels is computed. In comparison to the baselines, we obtain favorable results, especially when only a small numbers of samples can be annotated.

As mentioned in Chapter 1, the goal of this dissertation is to tackle domain shift from two angles. The work presented in this chapter falls under one of those where models trained on one domain are adapted to another domain that exhibits different characteristics. With no adaptation, the model does not generalize as well to target data due to domain shift. Given a small budget for annotating data, we show how we can leverage information deduced from the model and unlabeled target data, to pick samples for annotation and refine the model. Our methods for sample selection enable us to select samples which the model is performing poorly on or might have the largest impact on the weights. This allows us to introduce a new domain to the model to be refined on, so existing models can be reused and the cost for annotating new data is reduced.

The content of this chapter is published as:

Contribution of the thesis author: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization.

4.2 Abstract

Annotated datasets are essential for supervised learning. However, annotating large datasets is a tedious and time-intensive task. This paper addresses active learning in the context of semantic segmentation with the goal of reducing the human labeling effort. Our application is agricultural robotics and we focus on the task of distinguishing between crop and weed plants from image data. A key challenge in this application is the transfer of an existing semantic segmentation CNN to a new

field, in which growth stage, weeds, soil, and weather conditions differ. We propose a novel approach that, given a trained model on one field together with rough foreground segmentation, refines the network on a substantially different field providing an effective method of selecting samples to annotate for supporting the transfer. We evaluated our approach on two challenging datasets from the agricultural robotics domain and show that we achieve a higher accuracy with a smaller number of samples compared to random sampling as well as entropy based sampling, which consequently reduces the required human labeling effort.

4.3 Introduction

The ability to interpret the scene in front of a robot is key for intelligent behavior in several applications. For example, precision farming robots need to know which type of plant they perceive or autonomous cars need to know which object in their surroundings is a car, a pedestrian, or a cyclist. These classification or semantic segmentation tasks are typically tackled using convolutional neural networks (CNNs) operating on image data. In order to perform well, neural networks need to be trained with appropriately annotated datasets.

The performance of most supervised learning approaches and especially deep learning systems is related to the quality and quantity of training data. Annotated training data, however, has a high cost as often a larger number of labeled training data is required. In this work, we focus on optimizing the training set generation for semantic segmentation of image data obtained from a mobile robot. Semantic segmentation refers to the task of computing a pixel-wise labeling of the images. More concretely, we address the agricultural robotics application in which robots should perform automated weed control. For the semantic segmentation, this means that we need to compute the semantic label "crop", "weed", or "misc" for each pixel in the image. This task is particularly challenging as the field conditions often change substantially between years, regions, weather, and soil conditions as can be seen in Figure 4.1.

One solution to adapt and refine existing semantic segmentation systems to new field conditions is through additional labeled data from the new field. As these new annotations need to be executed at the end-users site, one is interested in keeping this effort as low as possible. Given annotated data on one agricultural field and a CNN that was trained on it, we address the problem of transferring this knowledge to new fields with minimum effort. Datasets from different fields reveal different crop and weed statistics. They often differ by soil type, weather condition, or various small objects that can be found on the ground, such as stones, dried vegetation, or marks from agricultural machines, i.e., patterns that are neither crop nor weed. Additionally, the robot can acquire images of plants at a certain growth stage in one field, while the growth state on the target field is different. Lastly, artifacts such as contrast changes can be found in the camera images captured from the various locations. As illustrated by (Lottes et al., 2018a; Lottes and Stachniss, 2017), these conditions make it difficult to simply reuse a previously trained network from one field and infer the labels on another.

The contribution of this work is to introduce and compare three active learning strategies that intelligently pick images taken under new conditions to re-train an existing network: The first one picks samples based on a log-space ranking of their loss with respect to pseudo labels. The second and third approaches select training samples that are expected to have a maximum effect on the network weights.

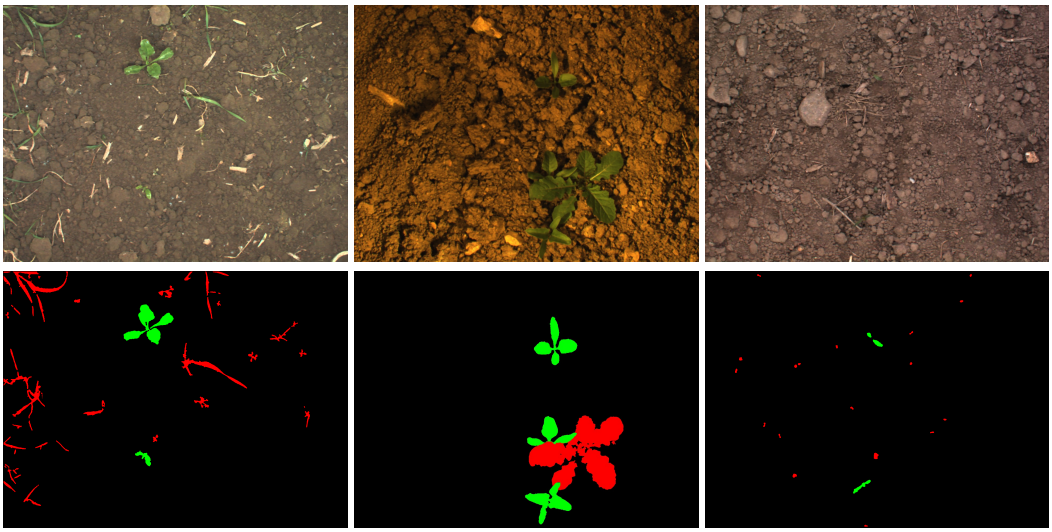


FIGURE 4.1: Sample images from the Bonn, Stuttgart, and Zurich sugar beet datasets in the first, second, and third column, respectively. The first row shows the RGB images and the second row shows their annotations (green denotes crop while red denotes weed). As can be seen, the appearance differs substantially.

Even though similar ideas have been explored for active learning in other application contexts, it is non trivial to apply them for semantic segmentation. An important technical novelty in our work is to exploit a pseudo ground truth, which we obtain with very weakly supervised segmentation. Our approach selects samples in batches, each time refining the network, then computing a new ranking of the unlabeled data. The best samples are then selected and the network is re-trained. To compute the real gradients, corresponding ground truth data is needed. Thus, in our approach, we approximate the ground truth as the result of unsupervised segmentation to estimate the gradient. We evaluated our framework using three distinctive sugar beet datasets (Chebrolu et al., 2017) that have different characteristics. Our results indicate that our method produces a higher accuracy on the datasets with a fewer number of samples compared to random sampling for annotation as well as entropy based sampling.

4.4 Related Work

Several works focusing on the elimination or reduction of herbicide use, through the incorporation of autonomous ground robots in crop fields, have been introduced to the community in the last years (Duckett et al., 2018; Liebisch et al., 2016; McCool et al., 2018). A key component of each of these unmanned platforms is a core perception system that has the ability to accurately distinguish crops from weeds in order to effectively and selectively apply the desired individual treatment (Lottes et al., 2018b; McCool et al., 2017; Milioto et al., 2017; Milioto et al., 2018; Sa et al., 2018). These systems allow autonomous robots to perform actuation in the fields without human supervision, treating each plant individually. All of the works referenced, however, are based on supervised learning approaches which take large amounts of pixel-accurate hand-labeled images for training. Accordingly, one of the main bottlenecks of these visual processing pipelines is the amount of expensive labeled training data required to deploy them in real agricultural fields, which often limits

their applicability. In order to tackle this data starvation problem, we propose an active learning based solution.

Numerous works on general active learning have been presented in the community (Settles, 2009; Guyon et al., 2011; Holub et al., 2008; Yoo and Kweon, 2019). The most common measures for selecting samples are based on the uncertainty of the network (Zhou et al., 2017; Yang et al., 2017; Gal et al., 2017; Wang et al., 2017) and diversity (Zhou et al., 2017; Dutt Jain and Grauman, 2016; Käding et al., 2016). (Sener and Savarese, 2017a) assert based on the experiments they performed that uncertainty based approaches are not effective for active learning with CNNs. They hypothesize that this is not due to the inaccurate estimate of uncertainty by the network, rather to the ineffectiveness of uncertainty based approaches to cover the space of image features. The Expected Model Output Change Principle (EMOC) developed by (Freytag et al., 2014) tries to avoid selecting samples that are redundant and (Käding et al., 2016) follow this approach with deep neural networks. This principle measures how a model would perform with and without the candidate sample. Given that the labels are unknown, a marginalization over the possible labels is needed. Uncertainty estimation for active learning can be performed using Monte-Carlo dropout as in (Gal et al., 2017) or with an ensemble of deep networks. (Beluch et al., 2018) compare both of these approaches on different datasets. They found that an ensemble of deep classifiers has a superior performance even with a smaller number of models. They conclude that Monte-Carlo dropout approaches suffer from a lower diversity and a smaller model capacity.

Weakly supervised segmentation is an active research topic (Wei et al., 2018; Acuna et al., 2018; Tang et al., 2018; Kwak et al., 2017). In the context of self-learning, (Zhang et al., 2018b) use labels obtained with K-means graph cuts as ground truth for their network. The predictions produced by the model are then used as the target labels for the next iteration of the process.

The works mentioned previously and the current state-of-the-art methods for active learning including (Gal et al., 2017; Beluch et al., 2018; Sener and Savarese, 2017b; Yoo and Kweon, 2019) are either more suitable for tasks other than pixel-wise semantic segmentation of images with CNNs and/or are memory and computationally expensive. Differently, we experiment with approaches that directly measure how annotated samples can affect the gradients. We use labels obtained with very weak supervision as pseudo ground truth and compute the gradients w.r.t the weights. We then refine a pre-trained network with the newly annotated samples in an iterative manner. Our intuition for using gradients is driven by the observation that the greater the mismatch is between the predicted segmentation and the ground truth, the larger the change is to the weights. This is in contrast to most of the approaches mentioned earlier that rely on the confidence of the network which may not be the best indication of the best samples to choose for annotation, as the network output might actually be correct although the network is uncertain about it.

Previous work, such as the Expected Gradient Length (EGL) (Huang et al., 2016; Settles et al., 2008), has explored how changes in model parameters can be exploited for sample selection. However, it computes the expectation of the gradient norm over all possible annotations, which would be prohibitively expensive for pixel-wise semantic segmentation of images. We instead compute gradients from rough foreground/background segmentation. (Du et al., 2018) use gradient similarity to determine when an auxiliary task is helpful for transfer learning to the main task and when it can be hurtful. Although in our work, the weakly supervised setting can be seen as an auxiliary task, we only use the gradients computed there as a guidance to

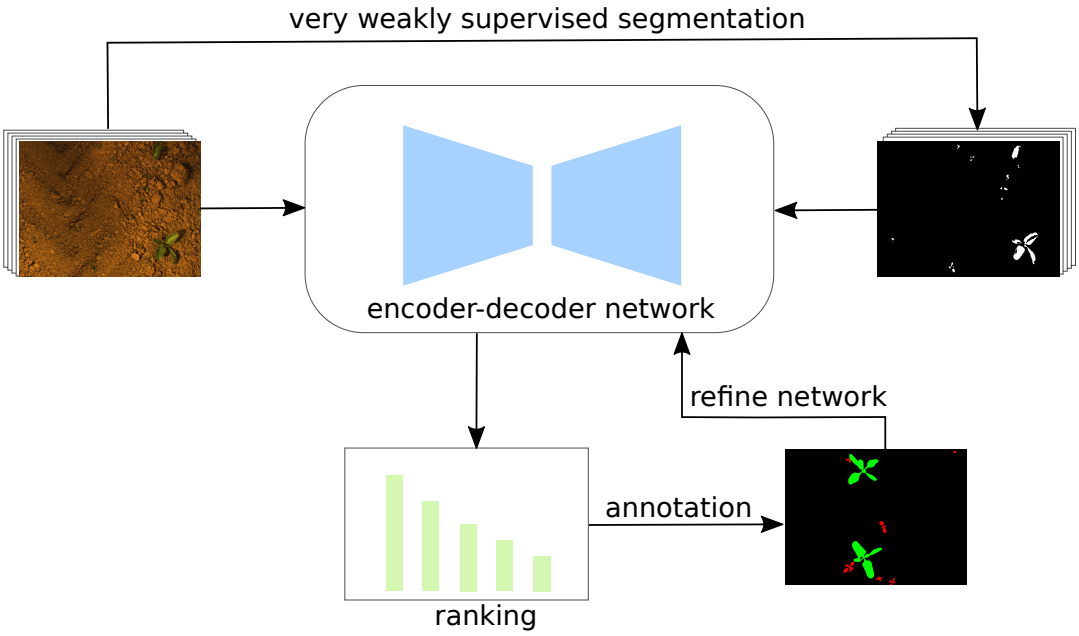


FIGURE 4.2: Overview of our approach. The key idea is that we first perform a very weakly supervised segmentation to obtain pseudo ground truth. Given the labels and different ranking measures obtained from the network, we rank the unlabeled samples and pick them accordingly for annotation. Those samples are then used to refine the entire network.

choose samples for annotations. These gradients are not used to measure similarity with those of the main task nor are the parameters of the main task updated with those gradients.

4.5 Our Approach to Effective Sample Selection

Figure 4.2 shows an overview of our framework. The key idea of our approach is to perform a very weakly supervised segmentation to obtain pseudo ground truth. Given the labels and different measures produced by the network, we rank the unlabeled samples and pick them accordingly for annotation. These are then used to refine the entire network.

Our CNN for semantic segmentation relies on Bonnet (Milioto and Stachniss, 2019). The used network is based on SegNet (Badrinarayanan et al., 2017) and ENet (Paszke et al., 2016). It has an encoder-decoder structure with a total of 25 [5x5] convolutional layers. It uses batch normalization, residual connections, ReLU as the non-linearity layer, and the focal loss function (Lin et al., 2017). As input to our network, we only use the standard RGB channels of a camera.

In order to perform the semantic segmentation in sugar beet field for agricultural robotics tasks, we train our model on the Bonn sugar beet dataset (Chebrolu et al., 2017). We then refine the trained model on other datasets by incrementally selecting batches of samples. The datasets differ in their crop/weed statistics and the images acquired with the cameras also differ in their illumination. Therefore, simply running the trained model to segment the vegetation in other fields does not work.

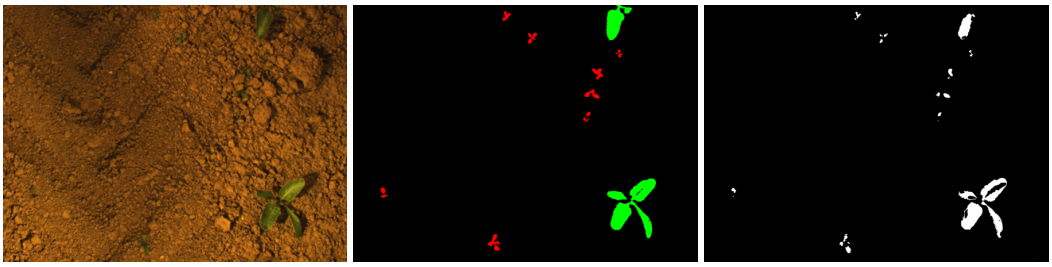


FIGURE 4.3: Very weakly supervised segmentation used as pseudo ground truth by our approach. Left: Input image. Middle: Ground truth semantic segmentation; Right: Foreground segmentation of vegetation provided by k-means clustering. Note that only such a rough segmentation as pseudo ground truth is enough for our approach.

We compare three different approaches to sample selection for active learning. Our main technical contribution is the generation of a pseudo ground truth (Section 4.5.2) and its use for loss-based (Section 4.5.3), as well as two gradient-based approaches for sample selection (Section 4.5.4 and Section 4.5.5).

4.5.1 Setup

We evaluate our different approaches by first training a network on the Bonn sugar beet dataset then refining it on the Stuttgart and Zurich datasets separately. To refine the network we pick unlabeled samples in batches of 10 using one of the methods described in this section. Once the samples are annotated, they are given to the network. We repeat this process iteratively, each time refining the network on all of the newly annotated samples.

4.5.2 Generation and Use of Pseudo Ground Truth

Our three main methods make use of "pseudo ground truth" foreground-background segmentation masks, which we obtain by clustering the values of the RGB channels. An example is shown in Figure 4.3. We run k-means to determine 20 cluster representatives from 10 randomly selected images. After viewing a single image that contains all 20 clusters, a human annotator chooses which clusters represent vegetation. In our experiments, it was enough to select two clusters. Therefore, the human annotation effort that is required to obtain the pseudo ground truth amounts to a few seconds for a complete new dataset. In accordance with previously used terminology (Zhang et al., 2018b), we refer to this as very weak supervision. Figure 4.3 shows an image, its ground truth and the foreground segmentation (pseudo ground truth) provided by clustering. It is an important finding from our experiments that a rough and easy to compute segmentation is sufficient for the purpose of selecting images for annotation. This makes our proposed gradient-based approach feasible in practice.

In order to compute a loss from the network output, which includes three classes, and the pseudo ground truth, which merely includes two, one might combine crop and weed into a single foreground class, or treat the foreground class as a specific type of vegetation (i.e., crop or weed). We tried all three options and found that treating the foreground from the pseudo ground truth as crop empirically produced the best result. We emphasize that the pseudo ground truth is only used to select

training samples that should be annotated; the network weights are updated based on manual annotations of the selected samples, which include all three classes.

In our agricultural application, the number of true classes (3) is not much higher than the number of classes (2) in our pseudo-ground truth. Naturally, in a different semantic segmentation task, the number of classes could be higher and might require generating a pseudo-ground truth with a larger number of classes. Our method here uses a simple clustering mechanism but other unsupervised or weakly supervised methods can also be used to generate pseudo-labels with a higher number of classes that can be later used to compute the gradients for sample selection.

4.5.3 Sample Selection Using Loss

The loss of the network is an indication of the segmentation error. Given that training neural networks with backpropagation is driven by the loss, it also provides a useful cue as to which samples the network will most benefit from. We compute the focal loss (Lin et al., 2017) based on the pseudo ground truth.

We found that training only on the images with the highest loss values did not generalize well. This could indicate that they are not representative enough of the overall dataset. Therefore, we instead employ a scheme that samples images with a diverse range of loss values, but prefers those with higher losses. To this end, we sort the images by their loss in a descending order, and then select them uniformly on a logarithmic scale. Specifically, we compute index i of the n-th sample as:

$$i = \lfloor |P|^{n/(|S|-1)} \rfloor - 1, \qquad n \in \{0, 1, \ldots, |S| - 1\} \tag{4.1}$$

where $|S|$ is the number of samples to be selected and $|P|$ is the size of the images pool. Since the samples are sorted, this approach would more heavily select those that have higher loss values while not completely discarding images that the network is performing well on.

4.5.4 Sample Selection Using Norm of Gradients

For this approach and the following one, we pick those samples for annotation that might have the largest impact on the network weights. The norm of the network gradients is a measure that is indicative of which samples will affect the weights more than others. Although the loss and norm of gradients are correlated, there are instances where the loss could be high for certain samples, yet the gradient is locally small. This depends on the loss function and the state of the current network parameters.

As in the previous approach, we use labels from very weakly supervised segmentation as pseudo ground truth. We run the network on the training images for one epoch (to maintain computational efficiency) and compute the gradients. Again we note that this step is only used to compute the gradients but the network weights remain unchanged. Once we have the gradients, we compute the L_2 norm of those in the last two layers of the network (the classifier layer and the one immediately before it):

$$n_g(\mathbf{x}) = \left\| \nabla_{w_f} \mathcal{L}(\mathbf{x}) \right\|, \tag{4.2}$$

where $\mathbf{x}$ is the image and w are the weights of the final two layers. The images are sorted based on this measure in a descending order and again we pick samples on a log-space scale afterwards as explained earlier.

4.5.5 Sample Selection Using Gradient Projection

The log-space in the previous approaches was used to ensure there is enough diversity among the samples so that the network does not overfit on them and can generalize to unseen data. Here we use a different method that relies on the space spanned by the gradients where we project onto the orthogonal complement of the gradients of the selected samples. For every picked sample, we project the gradients of all remaining samples onto the selected sample gradient. We then subtract the projected gradient from the original gradients. The residual we are left with indicates which samples have the strongest remaining effect on the weights after accounting for the already selected samples. This can be formulated as:

$$n_p(\mathbf{x}) = \left\| \mathbf{g_x} - \sum_{i=1}^{S} \frac{\langle \mathbf{g_i}, \mathbf{g_x} \rangle}{\langle \mathbf{g_i}, \mathbf{g_i} \rangle} \mathbf{g_i} \right\|, \tag{4.3}$$

where $\mathbf{x}$ is the image, $\mathbf{g_i}$ is the gradient of the ith sample out of S previously selected samples, and $\mathbf{g_x}$ is the gradient of the current sample. We select samples one by one, each time sorting them according to this measure and choosing the one with the highest norm of the residual. To pick the first sample, we choose that with the highest norm of the gradient.

4.6 Experimental Evaluation

In this section, we demonstrate the effectiveness of the approaches we designed for active learning and evaluate the performance of the different sample selection methods on different datasets, and compare them to random and entropy based approaches.

4.6.1 Datasets

The datasets we used were acquired with a Bosch Deepfield Robotics UGV. The robot was developed to assist in several agricultural applications, including mechanical weed control and selective herbicide spraying (Chebrolu et al., 2017). It is equipped with multiple sensors such as cameras, GPS trackers, and 3D laser sensors. For our experiments we use the RGB data provided by the JAI AD-130GE camera.

The data was captured in three different fields: Bonn and Stuttgart in Germany, and Zurich in Switzerland. The datasets have weed and crop plants at different stages of growth. Figure 4.1 shows sample images from the different datasets. The images vary in their illumination, soil type, and class statistics, hence the need for transfer learning. The images have been annotated into three classes: weed, crop, and soil/misc. Table 4.1 shows the number of images in each dataset and the ratio of foreground pixels. It can be clearly seen that there is a high imbalance of classes in the data. We follow the approach of (Milioto et al., 2018) and split the new dataset into three sets: 40% for training, 10% for validation, and 50% for testing. The samples are picked from the training set. All experiments were conducted on four Nvidia Titan X GPUs.

4.6.2 Re-Training Performance

The experiments in this section are designed to show how the proposed sample selection strategies impact the performance of the network in the new environment.

TABLE 4.1: Datasets Statistics of Crop and Weed Plants

	Bonn	Stuttgart	Zurich
Images	8230	2584	2577
Crop pixels	2.0%	1.5%	0.4%
Weed pixels	0.3%	0.7%	0.1%

TABLE 4.2: IoU without any refinement (lower bound) and IoU when training on the whole dataset (upper bound).

	No Refinement	Fully supervised
Stuttgart	0.3429	0.7989
Zurich	0.3595	0.7024

For quantifying the performance, we use the mean Intersection over Union (mIoU) as the performance measure. To provide the lower and upper bounds for the methods, we list in Table 4.2 the mIoU for each dataset when running the model without any refinement as well as when training on all of the samples.

Figures 4.4 and 4.5 show the performance on the Stuttgart and Zurich datasets when selecting samples for annotation with different methods. As baselines we include random sampling, and selecting samples that have the highest entropy (Chakraborty et al., 2015; Zhou et al., 2017):

$$H(\mathbf{x}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c} p(c \mid x_i)\log p(c \mid x_i), \tag{4.4}$$

where x_i is pixel i in image $\mathbf{x}$, c is the class and N is the number of pixels in the image.

A few observations can be made from the figures: the effect of the sampling method is stronger when only a few images are selected. As the model is trained on more and more samples, the accuracy plateaus as expected and the variation between the different methods decreases. It can be noted however that random sampling has a lower performance even with a greater number of images.

The overall performance on the Stuttgart dataset is better than that on the Zurich dataset. This can be attributed to the different class statistics of the two datasets. As can be seen in Table 4.1, the Stuttgart dataset has a larger percentage of crop and weed pixels compared to the Zurich dataset. This allows the model to better distinguish between the different classes. This observation is also supported by the fully supervised performance shown in Table 4.2 where a higher IoU can be obtained on the Stuttgart dataset.

When training the model with only a handful of images, 10 or 20 images, the methods that take into account the impact of the samples on the weights lead to better generalization to the rest of the unseen data. In particular, ranking samples by projecting out gradients results in higher mIoU on both datasets. With 10 samples, which would amount to roughly 1% of the training dataset size, we can achieve 90% of the fully supervised performance (Table 4.2) on the Stuttgart dataset, compared to 76% with random selection. On the Zurich dataset, we can achieve 77% of the fully supervised performance compared to 63% with random selection.

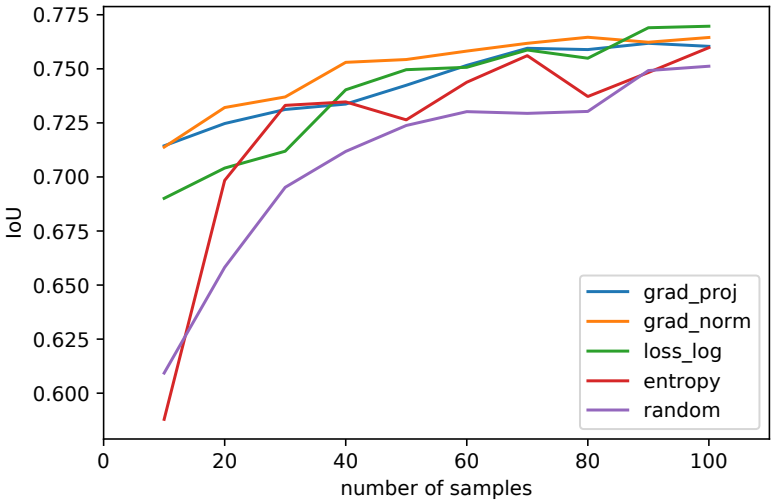


FIGURE 4.4: Pixel-wise mean IoU on the Stuttgart dataset. Running the model without any new annotations yields an IoU of 0.34. Running the model on the whole dataset yields an IoU of 0.79. Gradient-based approaches can reach 90% of the fully supervised performance with 10 samples.

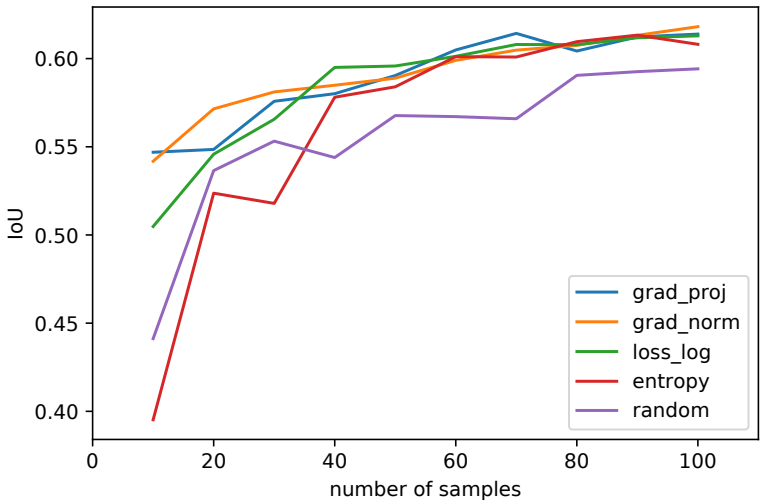


FIGURE 4.5: Pixel-wise mean IoU on the Zurich dataset. Running the model without any new annotations yields an IoU of 0.36. Running the model on the whole dataset yields an IoU of 0.70. Gradient-based approaches can reach 77% of the fully supervised performance with 10 samples.

TABLE 4.3: Object-wise Performance on the Stuttgart and Zurich datasets respectively. Each row shows the performance after selecting 10 samples with the different methods and refining the network. Running the model without any new annotations yields an accuracy of 0.15 on Stuttgart and 0.33 on Zurich.

Samples No.	Random	Entropy	Loss	Gradient Norm	Gradient Proj.
10	0.6920	0.6890	0.7882	0.8040	**0.8196**
20	0.7402	0.8050	0.7769	0.8350	**0.8404**
30	0.8138	0.8300	0.7950	0.8461	**0.8470**
40	0.8254	0.8463	0.8555	**0.8682**	0.8252
50	0.8225	0.8405	0.8523	**0.8599**	0.8278

Samples No.	Random	Entropy	Loss	Gradient Norm	Gradient Proj.
10	0.7552	0.7879	0.7697	**0.8354**	0.8025
20	0.7971	0.8212	0.8189	**0.8768**	0.8170
30	**0.8591**	0.7884	0.8321	0.8553	0.8299
40	0.8575	**0.8711**	0.8610	**0.8711**	0.8479
50	0.8593	0.8688	0.8636	**0.8852**	0.8784

To further quantify the performance of our approach, we use the object-wise metric defined by (Milioto et al., 2018), where the accuracy is measured for objects larger than 50 pixels. Since the target application is weeding with agricultural robotics, this metric is more directly useful than pixel-wise performance. Table 4.3 shows how our approach performs on the Stuttgart and Zurich datasets. Each row shows the mean accuracy when selecting n samples with different methods. For comparison, random and entropy based sampling are shown in the first and second columns respectively.

4.6.3 Comparison to Other Baselines

To gain more insight into what our baselines are, we ran additional experiments with the results shown in Table 4.4.

In the first row, we ran an experiment where we trained the model with the pseudo ground truth first and picked samples randomly afterwards. We found that it performs slightly better than when picking random samples directly (0.64 vs. 0.61) but still worse than our log and gradient based methods (e.g. 0.64 vs. 0.71 for the gradient-norm approach). Although pre-training with the pseudo ground truth allows the network to distinguish foreground vegetation from background, the task at hand is to learn three classes and more importantly distinguish crop from weed. Therefore for all experiments, we refine the model without pre-training on the foreground masks.

In the second row, we run an "oracle" experiment. We compute the difference between the parameters of the model without any refinement and the parameters of the fully supervised model. We then find samples with gradients that align with the parameters difference. This experiment is not intended for sample selection, rather

TABLE 4.4: Additional baselines for training with 10 samples on the Stuttgart dataset. Compare with Figure 4.4.

Method	mIoU
Random-pseudo ground truth	0.6448
Align with parameters difference (oracle)	0.7010

to know if the framework had complete knowledge of how the gradients should look like, would it be able to pick better samples. We found that the oracle performance is similar to our gradient-based approaches after seeing 10 new samples. This implies that the gradient-based approaches are bounded by this performance. Substantially improving upon their performance might require exploiting additional knowledge from the model, possibly with the aid of unsupervised segmentation.

4.6.4 Inspecting t-SNE of Samples Gradients

To further analyze the ranking methods and inspect potential patterns in the different sampling approaches, we plot the t-distributed Stochastic Neighbor Embedding (t-SNE) of the gradients in Figure 4.6. Each circle denotes the 2-D embedding of the gradient of a single image before picking the first 10 samples. Samples selected by each method are shown in different colors. As explained in Section 4.5.4 and 4.5.5, we combined the idea of gradient-based selection with two alternative approaches to achieving diversity in the selected images: picking on a log scale, or projecting out gradients that have been selected previously. In our experiments, both strategies performed well (see Figure 4.4 and Figure 4.5). When inspecting the gradients of the samples selected, we found that the strongest gradients cluster together, near the top left. Additionally, the gradient projection method selects many points at the boundary of the distribution, suggesting that it might be improved further by adding a mechanism to ensure that selected images are representative of a larger subset in the overall dataset.

4.6.5 Performance on Weed and Crop Classes

A more detailed breakdown of the methods performance is shown in Table 4.5. The first table shows the pixel-wise precision and recall on the Stuttgart dataset after selecting the first 10 samples. Both methods, Gradient Norm and Gradient Projection have a high recall and precision of the crop class without degrading those of the weed class. The object-wise performance in the second table further illustrates the effectiveness of these methods. Gradient Norm and Gradient Projection produce high precision and recall for both classes. We observed the same behavior on the Zurich dataset (not included here).

4.7 Conclusion

In this paper, we introduced and compared several active learning approaches that support the adaptation of semantic segmentation networks to new environments. Our approaches effectively select samples from the new environment for user annotation with the goal of maximizing the benefit from a small number of annotated

TABLE 4.5: Precision and recall on the Stuttgart dataset after selecting the first 10 samples. The first table shows the pixel-wise performance and the second table shows the object-wise performance. The highest values along a column are in bold and the lowest in italics.

	Precision		Recall	
	Weed	Crop	Weed	Crop
Random	*0.4095*	*0.7278*	0.4851	0.6946
Entropy	0.4158	0.7334	*0.4786*	*0.5894*
Loss Log	0.5331	0.8025	0.6179	0.8112
Gradient Norm	**0.5970**	0.8259	0.6136	**0.8402**
Gradient Projection	0.5745	**0.8365**	**0.6564**	0.8212

	Precision		Recall	
	Weed	Crop	Weed	Crop
Random	*0.8723*	0.5740	0.6587	*0.6474*
Entropy	0.8851	*0.5238*	*0.6122*	0.7399
Loss Log	0.9005	0.6898	0.7811	0.7351
Gradient Norm	**0.9090**	**0.7390**	0.7970	**0.7536**
Gradient Projection	0.9030	0.7308	**0.8289**	0.7375

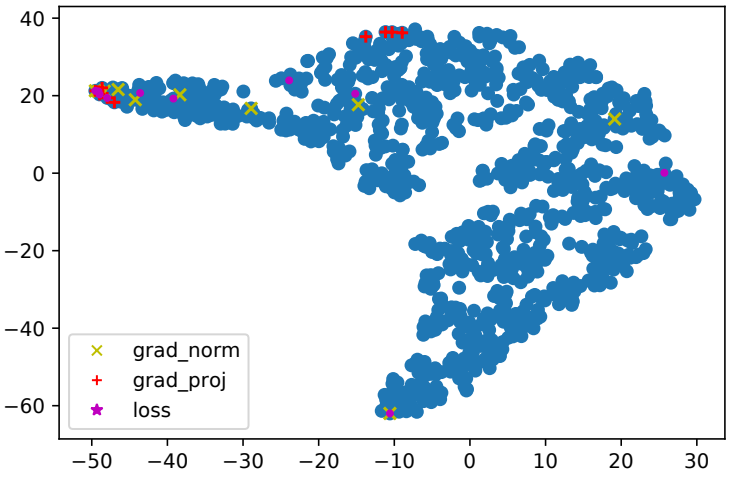


FIGURE 4.6: t-SNE of the images gradients on the Stuttgart dataset. Each point represents the 2-D embedding of the gradient vector. The first 10 samples selected by each method are shown in different colors.

examples. We applied sample selection strategies to the task of crop/weed classification for agricultural robots, as the appearance between agricultural fields often changes substantially such that re-training is needed. We compute pseudo ground truth labels using very weakly supervised segmentation and use those labels to estimate how new, unlabeled samples will affect the weights of the CNN if selected for training. We select the training samples for user annotation based on the estimated effect on the weights and use them to refine the network.

We evaluated the performance gain of our gradient-based and log-based approaches on two agricultural datasets for weed detection. The datasets reveal different characteristics from the dataset on which the network was pretrained. Our results show the effectiveness of our method as it produces higher semantic segmentation accuracies with a small number of training samples, compared to random sampling as well as entropy based sampling. As a result of that, the effort in human annotation is reduced without compromising performance.

Chapter 5

Unsupervised Domain Adaptation for Medical Image Segmentation via Self-Training of Early Features

5.1 Summary

One of the challenges with building and using models in the medical field, is that data acquired at different sites will have different properties, such as slice thickness and in-plane resolution, in addition to showing different intensity levels and possible acquisition artifacts. This makes it challenging to reuse a model trained on data generated at one hospital for example to data generated at another. When no target annotated data is available, Unsupervised Domain Adaptation (UDA) can be used to transfer the knowledge from one domain to another. Our approach to UDA involves the application of self-training with pseudo-labels that we generate from the target data.

Our model follows the U-Net architecture (Ronneberger et al., 2015) which we train using the cross-entropy loss. We add another segmentation head to the network just before the first downsampling operation and train this second head on the source domain with the same cross-entropy loss. The model now produces two segmentation maps, a rough one after a few convolutional layers, and the usual one at the end of the network.

Having trained the model on the source domain, we next adapt it to the target domain without using any target labels. To do so, the final segmentations are used as pseudo-labels for the early segmentations, therefore the early segmentation head acts as a student, and the base U-Net as the teacher. We compute the loss between the two segmentation maps and refine the shared convolutional layers at the beginning of the network. Put differently, the stronger segmentations at the end of the network act as a self-supervising signal. Exploiting that information to refine the early convolutions results in improvement in the final segmentations as well since the refined activations are propagated through the network.

We evaluate our approach on two datasets. The first one is the Calgary-Campinas dataset (Souza et al., 2018) where the task is to segment out the brain using MR images as input. The data was generated using a combination of different scanners and field strengths resulting in six domains. We use one of them as our source domain and adapt to each one of the other five target domains. Similar to (Shirokikh et al., 2020), the Surface Dice score (Nikolov et al., 2018) is used to evaluate the performance on this dataset. Our method for unsupervised adaptation considerably improves the performance which we observe quantitatively and qualitatively. We

repeat the experiment using a different domain as our source domain and observe a similar improvement in performance.

Taking a model refined on target data, we ran an additional experiment to see if the refined model generalizes to unseen data from the same target domain but without refining on them again. The results show that the model indeed generalizes after adaptation. Another set of experiments that we ran looked into whether we gain a benefit from refining not only the early layers but also deeper ones. We found a marginal improvement in three out of five domains. We also experimented with refining only the batch normalization layers (Hu et al., 2021), which resulted in an improvement over the non-adapted model but it was worse than when refining the entire convolutional blocks.

The second dataset we evaluate on is the Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation (M&Ms) dataset (Campello et al., 2021). This is a multi-class problem where the task is to segment the left ventricle cavity, right ventricle cavity, and the left ventricle myocardium in cardiac MR images. The data was generated using four scanner vendors. We choose one as our source domain and adapt to the other three target domains. For quantitative evaluation, we use the volumetric dice score and observe an improvement in performance, which is also reflected qualitatively. Looking at the breakdown of performance on each class individually, the results show a balanced improvement across all classes. Comparing our method to (Zou et al., 2018) who also use pseudo-labels, we find that the samples selected for refinement in their approach are frequently mislabeled which negatively affects the adaptation performance.

The work presented in this chapter is part of the domain adaptation aspect that we address in this dissertation. If we have no annotated data from the target domain or it is costly to acquire, then being able to do domain adaptation in an unsupervised manner is an appealing choice to adapt models from one domain to another. Our use of the entire probabilistic label maps as the supervising signal along with limiting the refinement to early convolutional blocks avoids relying on a small subset of pixels and reduces the risk of overfitting to the pseudolabels.

The content of this chapter is published as:

Rasha Sheikh and Thomas Schultz (2022). "Unsupervised Domain Adaptation for Medical Image Segmentation via Self-Training of Early Features". In: *International Conference on Medical Imaging with Deep Learning, MIDL 2022, 6-8 July 2022, Zurich, Switzerland*. Ed. by Ender Konukoglu et al. Vol. 172. Proceedings of Machine Learning Research. PMLR, pp. 1096–1107. URL: https://proceedings.mlr.press/v172/sheikh22a.html.

Contribution of the thesis author: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft, Visualization.

5.2 Abstract

U-Net models provide a state-of-the-art approach for medical image segmentation, but their accuracy is often reduced when training and test images come from different domains, such as different scanners. Recent work suggests that, when limited supervision is available for domain adaptation, early U-Net layers benefit the most

from a refinement. This motivates our proposed approach for self-supervised refinement, which does not require any manual annotations, but instead refines early layers based on the richer, higher-level information that is derived in later layers of the U-Net. This is achieved by adding a segmentation head for early features, and using the final predictions of the network as pseudo-labels for refinement. This strategy reduces detrimental effects of imperfection in the pseudo-labels, which are unavoidable given the domain shift, by retaining their probabilistic nature and restricting the refinement to early layers. Experiments on two medical image segmentation tasks confirm the effectiveness of this approach, even in a one-shot setting, and compare favorably to a baseline method for unsupervised domain adaptation.
Keywords: Unsupervised Domain Adaptation, Segmentation

5.3 Introduction

Annotating medical images to supervise the training of deep neural networks for segmentation is time-consuming and often requires medical experts. Once trained, these models perform well on similar data from the same site, but the performance often drops on data acquired in a different site with another type of scanner for example. Transfer learning and domain adaptation offer various solutions to this problem by adapting the model to new data and addressing the domain shift between source and target domains. Unsupervised domain adaptation attempts to do so without using any labeled target data.

There are different approaches to unsupervised domain adaptation for the task of semantic segmentation. Several works use an adversarial scheme to learn domain-invariant features (Hoffman et al., 2016) or image-to-image translation as in CycleGAN (Zhu et al., 2017) to adapt the segmentation model (Li et al., 2019). Others perform a layer-wise matching of activations between the domains (Huang et al., 2018), or aim to minimize the entropy of target predictions (Vu et al., 2019) based on the observation that source predictions often have higher confidence values.

Our proposed approach follows a self-supervision strategy. Self-supervision either makes use of auxiliary tasks (Sun et al., 2019) or losses (Hu et al., 2021) that do not require supervision, or it refines the network based on its own predictions on the target domain. An important issue with the latter strategy is to avoid propagating incorrect predictions. This has been approached by filtering predictions so that only the most confident ones are used as a training signal (Zou et al., 2018). We propose to use the predictions differently, inspired by recent work that has observed that the first layers of U-Net models learn more domain-specific features (Shirokikh et al., 2020), and benefit most from a refinement when limited training data is available (Zakazov et al., 2021).

Therefore, we add a segmentation head after the first few convolutional layers and use final predictions of the network as pseudo-labels to refine only those early features. Our main finding is that this leads to a stronger improvement of the final segmentation than a filtered self-training of the whole network. We believe that this reflects a greatly decreased re-enforcement of incorrect predictions, because our pseudo-labels retain their probabilistic nature, and because we limit adaptation to early layers. Our code is publicly available at `https://github.com/ferasha/UDAS`.

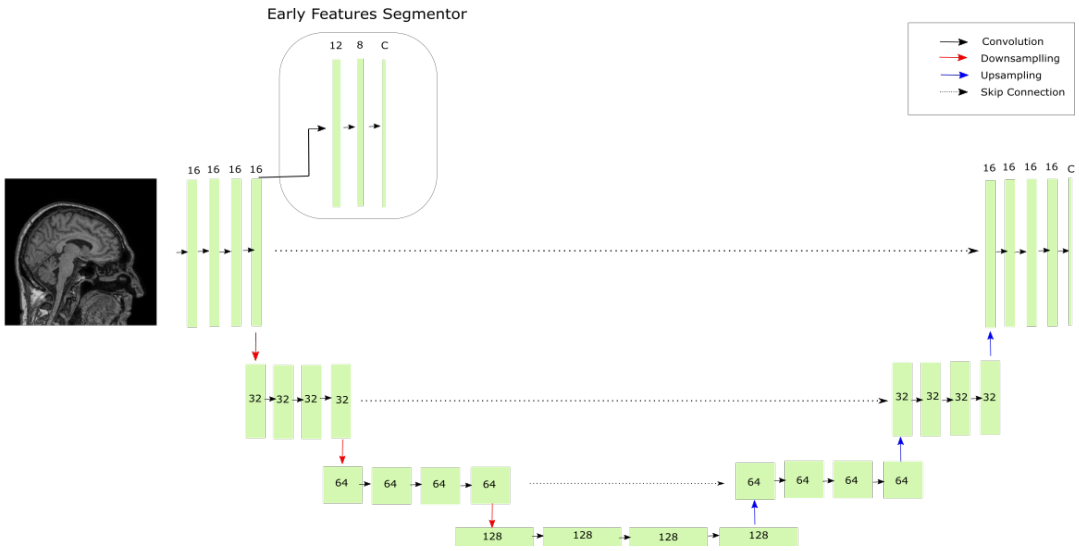


FIGURE 5.1: Architecture of the segmentation model.

5.4 Method

5.4.1 Base Model

We first train a segmentation model on a source domain, and later adapt it to a target domain with unlabeled data. The base model is shown in Figure 5.1. It has a U-Net (Ronneberger et al., 2015) structure identical to the one used by (Shirokikh et al., 2020) where images go through a number of convolutional blocks that learn M feature maps with the same spatial size as the input before proceeding with the encoder-like part of the architecture. It uses 3×3 convolution kernels, ReLU activation functions, and the skip connections are implemented as convolutions followed by a sum operation. The model is trained with the cross-entropy loss,

$$l(x, y)_{source} = -\frac{1}{N_S} \sum_{n=1}^{N_S} \sum_{c=1}^{C} y_{n,c} \log p_c(x_n), \tag{5.1}$$

where N_S is the number of samples in the source images, $p_c(x_n)$ is our model's predicted probability that pixel n in image x belongs to class c, and y_n is a one-hot-encoding vector of the true label for pixel n. We use the Adam optimizer, no augmentation, and 0.001 as the learning rate.

5.4.2 Domain Adaptation Through Self-Training

To prepare adapting the base model to the target domain, we first add another segmentation head to it, just before the first downsampling operation. This block is titled *Early Features Segmentor* in Figure 5.1. During refinement, this head will act as a student which is trained by the output of the overall base model, which acts as a teacher. Because student and teacher share the first few convolutional layers, refining the student on the target domain also benefits the teacher. While initializing the student on the source domain, we freeze all weights in the base model, and train with the same loss as in Equation 5.1, again using the ground-truth segmentation masks.

The model now produces two probabilistic segmentation outputs, a weak one $\tilde{p}_c$ based on early features, and the final segmentation p_c at the end of the network. For domain adaptation, we feed samples from the target domain through both branches of the network, and compute a cross-entropy loss between them,

$$l(x)_{target} = -\frac{1}{N_T} \sum_{n=1}^{N_T} \sum_{c=1}^{C} p_c(x_n) \log \tilde{p}_c(x_n), \tag{5.2}$$

where N_T is the number of samples in the target images.

We now use the stronger segmentation p_c at the end of the network to improve the weaker early segmentation $\tilde{p}_c$. This adapts the early features based on the richer and higher level information that was learned by the rest of the network. In this phase, we only update the weights of the early convolutional blocks that are shared between the two branches, freezing the weights in the early segmentation head itself. Since we minimize Equation 5.2 only with respect to the early segmentation $\tilde{p}_c$, we do not obtain a gradient in the rest of the U-Net, so weights there remain unaffected as well. Despite this, the probabilities p_c that we use as pseudo ground truth also change during the refinement process, since they are affected by the updates in the early layers. Throughout the refinement, we track the Dice score between the early and final segmentations. We stop when the absolute difference between the Dice in the current and the previous epoch drops below 0.005.

5.5 Experiments

5.5.1 Calgary-Campinas Dataset

The Calgary-Campinas dataset (Souza et al., 2018) consists of 359 3D volumes of brain MR images with corresponding skull-stripping segmentation masks. The data is generated using six scanners which differ in the vendor type and the field strength. Those scanners represent the different domains in our experiments.

We train on 40 subjects from GE 3 (i.e. source domain) and test on 10 subjects from each of the other target domains. The only pre-processing is a min-max scaling of each volume. To evaluate the performance, we follow (Shirokikh et al., 2020) and use the surface Dice score (Nikolov et al., 2018), which quantifies the fraction of the predicted and ground truth surfaces that are within a pre-specified distance of each other. This score is deemed more informative than the usual volumetric Dice in this context because it focuses on the structure of interest, i.e., the brain contour, as opposed to the large, but mostly trivial internal volume.

Improvement Over Base Model

Table 5.1 shows the adaptation result on the different domains. Here, the label "base model" refers to the model that was trained on the source domain, without any adaptation. Our approach significantly improves upon the base model. Qualitative results from all targets domains are shown in Figure 5.2.

To further validate our approach, we also treated another domain, Siemens 3, as our source domain and adapt to the other target domains. These results are shown in Appendix 5.7.1. Once again, we observe an improvement in the performance.

To illustrate that the refinement works as described above, Appendix 5.7.2 shows the early and final probabilistic segmentations for an example input, before and after refinement.

TABLE 5.1: Surface Dice scores on the Calgary-Campinas dataset. ST and CBST refer to the self-training and class-balanced self-training proposed by (Zou et al., 2018).

	Base Model	**ST**	**CBST**	**Ours**
GE 1.5	0.55487	0.53042	0.55347	**0.75887**
Philips 1.5	0.74974	0.72526	0.77563	**0.84601**
Philips 3	0.65806	0.66237	0.68958	**0.85810**
Siemens 1.5	0.70478	0.69294	0.75003	**0.82457**
Siemens 3	0.88651	**0.89180**	0.88418	0.88740

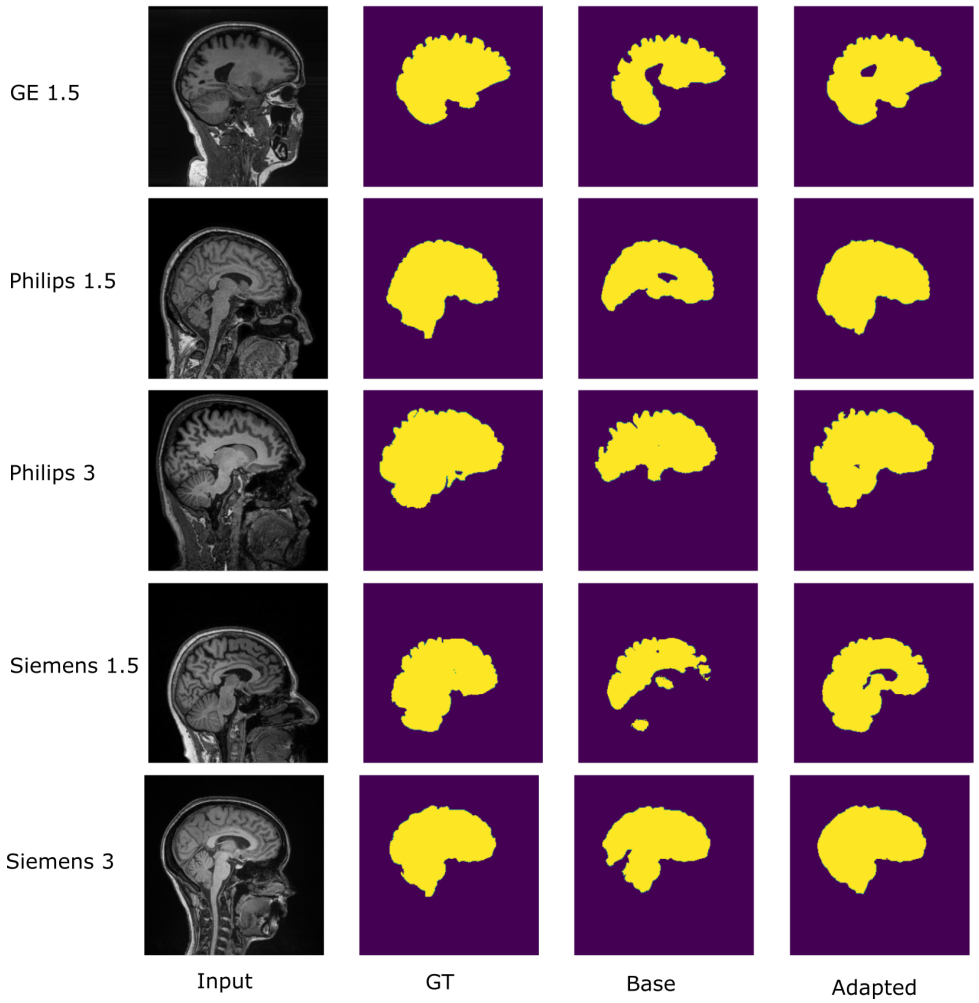


FIGURE 5.2: Qualitative results from the Calgary-Campinas dataset. Columns show the input image, ground-truth, and segmentation using the base and adapted models, respectively. The rows represent the different domains.

TABLE 5.2: Surface Dice when refining and testing on one target subject at a time, averaged over 10 subjects.

	Base Model	One-Shot Refinement
GE 1.5	0.55487	**0.73655**
Philips 1.5	0.74974	**0.84003**
Philips 3	0.65806	**0.85312**
Siemens 1.5	0.70478	**0.82449**
Siemens 3	**0.88651**	0.87900

TABLE 5.3: Surface Dice of the adapted model on a new subset from the same target domain, without additional refinement.

	Base Model	Fixed Adapted Model
GE 1.5	0.46892	**0.72773**
Philips 1.5	0.78818	**0.88231**
Philips 3	0.55264	**0.82302**
Siemens 1.5	0.72790	**0.82344**
Siemens 3	0.88564	**0.90815**

Comparison to Previous Work

We consider the previous work by (Zou et al., 2018) to be most similar to ours, since it also uses a segmentation loss towards pseudo-labels to refine the network weights. Unlike our approach, they filter the predictions of the network to keep those with higher confidence values and use them as labels to refine the whole network. They implement two variations of this idea, with and without class-balanced filtering. These are referred to as Class-Balanced Self-Training (CBST) and Self-Training (ST) in Table 5.1. On our data, we observe moderate improvements with this approach and occasionally a drop in performance. For a more direct comparison, we also tried to restrict the CBST approach to update the same layers as our method. However, results were worse than when refining the entire model.

One-Shot Domain Adaptation

In the one-shot setting, a single dataset should be segmented, and is the only data available from the target domain. Running our refinement in that mode, separately for each subject in our test set, produced results that were almost as accurate as the refinement on 10 volumes, which was reported above. Table 5.2 shows the resulting average surface Dice score.

Generalization to Unseen Data

After using our approach to refine the early layers on 10 subjects from the target domain, we applied the model to further data from the same domain without further refinement. Results on a subset of the target domain that was disjoint from the one used for refinement are shown in Table 5.3. They confirm that the model now generalizes successfully.

Alternative Modes of Refinement

Given the benefit from refining the first few convolutional layers, it is natural to try a similar strategy for refining deeper layers as well. Due to the mismatch in

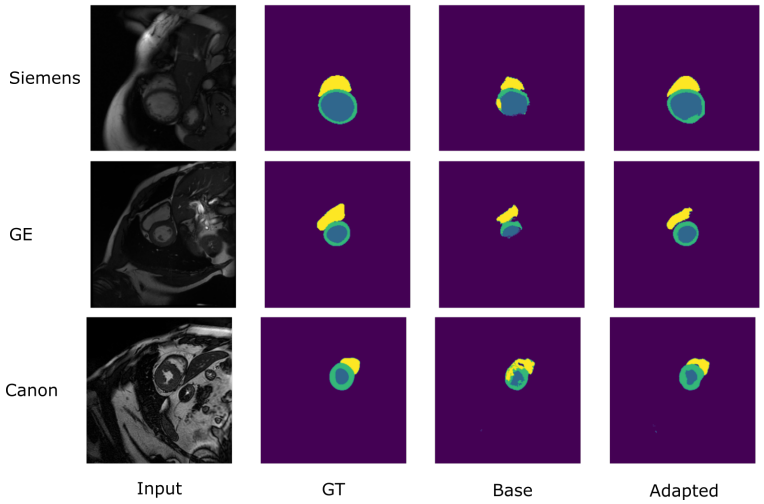


FIGURE 5.3: Qualitative results from the M&Ms dataset. The rows represent the different domains. The columns show the input image, ground-truth, segmentation using the base and adapted model respectively. Yellow: RV, Blue: LV, Green: MYO

image resolution at deeper layers, this requires a resampling of features, predictions, or labels. Even the variant that worked best in our experiments only provided a marginal additional benefit when compared to the simpler refinement of the earliest layers alone.

Moreover, we tried refining only the batch normalization layers, as proposed for domain adaptation by (Hu et al., 2021). However, refining all weights gave better results in our case. Experimental results related to these alternatives are discussed in Appendix 5.7.3.

5.5.2 Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation (M&Ms) Dataset

We also present results on a more challenging multi-class segmentation task. This dataset (Campello et al., 2021) consists of four domains corresponding to images from four scanner vendors. The data from the different sites vary in their in-plane resolution, slice thickness, number of slices, and number of time frames.

The publicly available data includes cardiac MR scans of 345 subjects. The segmented regions are the left ventricle cavity (LV), the right ventricle cavity (RV), and the left ventricle myocardium (MYO). The only pre-processing we apply is min-max scaling.

We train on 75 subjects from the Philips training set (source domain) and test on the official test sets of the other target domains. We chose Philips as our source domain because of the large drop in performance when testing on the other domains (Campello et al., 2021). Exemplary segmentation results are shown in Figure 5.3. They illustrate differences in image appearance across the domains (rows), and the benefit of our refinement (final column) compared to the base model with respect to the ground truth (GT).

To quantify the performance, we use the volumetric Dice score

$$\text{Dice} = \frac{2\sum \hat{y}y}{\sum \hat{y} + \sum y}, \tag{5.3}$$

TABLE 5.4: Volumetric Dice scores on the M&Ms Dataset. ST and CBST again refer to the self-training and class-balanced self-training proposed by (Zou et al., 2018).

	Base Model	**ST**	**CBST**	**Ours**
Siemens	0.61926	0.54632	0.53262	**0.68608**
GE	0.43403	0.37390	0.38717	**0.67550**
Canon	0.65464	0.66477	0.68268	**0.70576**

where $\hat{y}$ and y are the predicted and true labels respectively. Table 5.4 reports the corresponding results on the different domains. We again compare the performance to the self-training approach in (Zou et al., 2018) and also show the baseline performance with no adaptation.

We found that on this more challenging dataset, the selection made by ST and CBST often includes samples that are incorrect despite a high confidence. In our experiments, the accuracy among pixels that were selected for adaptation was sometimes as low as 20%. This explains why the filtering approach was sometimes detrimental on this dataset. In contrast to this, our use of probabilistic pseudo-labels from the full image for early feature refinement still provided a benefit.

Appendix 5.7.4 provides a breakdown of the per-class performance, demonstrating that all classes benefit from the refinement.

5.6 Conclusion

Domain shift frequently occurs in medical imaging when data generation differs between sites, e.g., when different scanners are in use. This can severely impact the performance of segmentation models on test data from a different site. Therefore, models have to be adapted. Unsupervised domain adaptation is an attractive solution since it lifts the need for annotating data from the other domain.

We proposed a novel, simple, and efficient strategy for domain adaptation via self-training and demonstrated clear qualitative and quantitative benefits on segmentation performance on two medical image segmentation tasks. We achieved superior performance compared to the CT and CBST baselines, which can be explained by reducing detrimental effects of propagating incorrect labels by retaining probabilistic pseudo-labels, and restricting the refinement to early layers. Using pseudo-labels and having to refine only a subset of weights also leads to fast training times: Our experiments only required up to five epochs. Compared to unsupervised domain adaptation based on adversarial training, our approach is easier to use because it does not require a careful balancing of the training signals from a generator and discriminator.

5.7 Appendix

5.7.1 Results With Siemens 3 as the Source Domain

5.7.2 Illustration of Early and Final Segmentations

Figure 5.4 compares segmentations from the base model in the second column to the refined model in the third column. As expected given the limited receptive field and complexity of early features, the early segmentations (top row) are much weaker than the ones at the end of the network (bottom row). The top image in the third

TABLE 5.5: Surface Dice when using the Siemens 3T domain of the Calgary-Campinas dataset as the source domain.

	Base Model	**Adapted Model**
GE 1.5	0.60650	**0.74609**
GE 3	0.82829	**0.95561**
Philips 1.5	0.73747	**0.83830**
Philips 3	0.56104	**0.85981**
Siemens 1.5	0.47547	**0.82672**

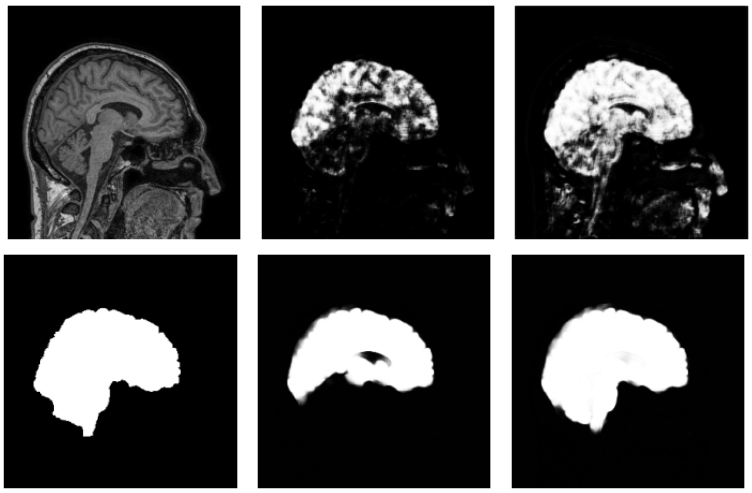


FIGURE 5.4: First column shows the input and ground truth. Second column shows the early and final segmentations using the base model. Third column shows the early and final segmentations using the refined model.

column shows that early feature refinement improved the early segmentation. The bottom image shows the final improvement when propagating the refined features through the remaining network.

5.7.3 Results from Alternative Refinement Strategies

TABLE 5.6: Surface Dice when also refining deeper layers, or only batch normalization weights

	Base Model	**Refine L1**	**Refine L1 then L2**	**Refine L1 and L2**	**Refine L1-BN**
GE 1.5	0.55487	0.75887	0.75994	**0.76352**	0.70093
Philips 1.5	0.74974	0.84601	0.84120	**0.85202**	0.83101
Philips 3	0.65806	0.85810	0.85979	**0.86574**	0.85083
Siemens 1.5	0.70478	**0.82457**	0.82176	0.82338	0.81101
Siemens 3	0.88651	**0.88740**	0.87205	0.87971	0.88291

Table 5.6 compares results from our proposed refinement ("refine L1") to alternative refinement strategies. In "refine L1 then L2", we extended our proposed method with a second refinement, in which we place an additional early segmentation head at the second resolution level, just before the second downsampling. We initialized it in the same way as it is described in Section 5.4.2, and used it to refine the weights at the second resolution level of the encoder, freezing the weights at the first level, which had already been refined previously. We resolved the resolution mismatch by upsampling logits and computing the segmentation losses at the original resolution. In "refine L1 and L2", we only used an early segmentation head at the second level, to update the weights of the first and second layer jointly. Compared to the benefit from refining the initial layers, the additional benefit from refining deeper layers with our method was marginal.

"Refine L1-BN" corresponds to our proposed method, but only refines parameters in the batch normalization blocks of the first layer. It did not perform as well as a full refinement of all weights.

5.7.4 Class-Wise Quantitative Results on M&Ms Dataset

TABLE 5.7: Breakdown of class-specific Dice scores on the M&Ms Dataset

		LV	MYO	RV
Siemens	Base Model	0.71047	0.54531	0.60199
	Ours	**0.77685**	**0.62302**	**0.65838**
GE	Base Model	0.49160	0.38219	0.42831
	Ours	**0.74438**	**0.62778**	**0.65434**
Canon	Base Model	0.71849	0.63117	0.61427
	Ours	**0.77674**	**0.65661**	**0.68394**

Chapter 6

Conclusion

Deep learning models for the task of semantic segmentation have shown remarkable accuracies when trained and evaluated on data coming from the same distribution. In practice however, that is not always the case as data used for inference might show different characteristics to the data that was collected and used to train the models. We explored different aspects of this problem in our research and proposed solutions that improve the robustness of segmentation models and ease the adaptation to new data. In the next section we summarize our work and contributions and conclude with an outlook that takes into account recent advances in deep learning.

6.1 Feature Preserving Smoothing

Augmenting the data pool is often used when training neural networks as it increases the training set size without the need to annotate additional data. Deep learning models benefit from large training sets as that allows the model to better learn semantic relationships despite variations in the input data. This in turn helps the model generalize to unseen data.

Different augmentation techniques continue to be popular. Some of them change the intensity values of images such as changing the brightness of images or adding Gaussian noise to them. Other augmentations are geometric in nature. These include resizing the image, rotating it, or flipping it for example. Generative models have also been used to transfer the style of one set of images to another.

When comparing what CNNs focus on in images compared to humans, it was found that CNNs give larger importance to high-frequency information such as texture whereas humans favor shape. This motivated us to apply Total Variation augmentation to our training images and train our model on the augmented and original set of images. Total Variation denoising creates piecewise constant regions, where high frequency noise is suppressed, while still preserving the edges between regions. This makes it particularly appealing for the task of semantic segmentation as the boundaries between objects are important to distinguish different classes, and we can get rid of the noise that does not contribute to the optimization goal.

We ran experiments on multiple datasets and compared our results quantitavely and qualitatively to other augmentation techniques. We found that using feature preserving smoothing boosts the segmentation performance. Given the simplicity of the proposed approach - no change for example in the model architecture or in the optimization loss - this can be a useful addition to the set of augmentations that are commonly used to train models.

6.1.1 Possible Extensions

It would be worthwhile to investigate the effect of combining feature smoothing augmentation with other types of augmentations, e.g. one could apply TV smoothing then rotate the image. Popular training frameworks now offer augmentation components that can be attached to the training pipeline. These components, such as AugRand or Augmix, offer strategies that take care of randomly selecting the augmentation type, its parameters, and whether it is combined with another transformation. TV smoothing can then be added as another possible choice to pick from these augmentations. As we show in our experiments, we see an improvement in performance with different strengths of smoothing. This value could therefore be a parameter that the augmentation strategy samples from a uniform distribution as it does with other augmentations (e.g. degree of rotation). This would allow for a greater variability in the generated images, as each batch might be augmented with a different smoothing strength.

Domain shift can be caused by different factors. For examples, images acquired with one scanner might be brighter or noisier. A particular site might have more pediatric patients leading to smaller structures in images to be identified. It would be helpful to investigate how to tailor augmentations to the particular domain shift present. Looking into how activations of the network's layers differ on source and target domains can provide useful insights. Test-time augmentation is also a rather inexpensive tool that could allow us to observe the effect of a particular augmentation without having to retrain the network.

6.2 Adaptive Optimization with Fewer Epochs

The nnUNet (Isensee et al., 2021) has become a popular framework for the segmentation of medical images, due to its general applicability and good performance. Some settings of the framework are dynamically changed according to the dataset properties. Others such as the optimizer and number of epochs have default values and are not adapted.

Given the large number of epochs (1000) that the nnUNet trains for, we speculated that in the context of domain generalization, a benefit could arise from lowering that number. Early-stopping is commonly used to prevent overfitting on training data, so the question is whether that also carries over to the case of having different domains such as data from scanners.

We found that when combining early stopping with the default optimizer of the nnUNet (SGD), the different speed at which layers were training meant that some were not trained long enough given the lower number of epochs. We therefore looked into whether adaptive optimizers are more suitable. One of the recently published adaptive optimizers is AvaGrad (Savarese et al., 2021), where the effective learning rate and the adaptability parameter ϵ are decoupled. This is appealing because when ϵ is large enough, the only hyperparameter needed to do a search for is the learning rate.

We ran experiments on two datasets where each one has multiple domains and found that using adaptive optimizers along with early stopping improves the generalization performance on other domains.

6.2.1 Possible Extensions

A natural question that arises from the insights we had with early stopping and adaptive optimizers, is how best to find the epoch at which we should stop training. It is a common practice in machine learning to track the performance of the model on a validation set. Once the improvement there plateaus or even becomes worse in case of overfitting, training stops. To have the model generalize to other domains, we can design a validation set that acts as a proxy for a different domain. The style for example of target images can be transferred to the source images, through adversarial means or other methods, and the performance on this transformed validation set can be a better measure of how the model might perform on target data. If there is no specific target domain that we want to adapt our model to, but we still want to improve the generalization of the model to other domains, we can generate styles from different transformations and have those applied only to the validation set and not the training set, thereby simulating the case of having different domains.

6.3 Gradient and Log-based Active Learning

Deep neural networks usually require large training sets to ensure that they capture rich features and do not overfit. Annotating such datasets is however a tedious task. This is exasperated by the problem of domain shift where data properties could differ from one domain to another. This has the consequence that a trained model cannot simply be applied to data from another domain. One obvious solution would be to again annotate data from the new domain and train or refine the model on them. But doing this for each new domain can be prohibitively expensive.

The task that we tackled was distinguishing between crop and weed plants in agricultural fields. This would enable autonomous robots to navigate agricultural fields that share the same crop and perform automatic weeding. The issue described earlier of not being able to simply reuse trained models shows up here, since images acquired from different fields can exhibit different statistics due to illumination changes, different weather conditions, and image acquisition artifacts.

The goal of active learning is to reduce the annotation effort by labeling only a small subset of the new data and refine the model on it. We create in our work three active learning strategies to select data for annotation. They work in conjunction with a proxy task which is a simple foreground/background segmentation of the images acquired through k-means. This segmentation provides us with pseudo-labels which we can use to rank samples in terms of how useful they might be for the model, and therefore will be annotated.

The first strategy computes the loss of each sample's prediction when comparing it with the pseudo-ground truth. These are then ranked on a log-space scale to encourage diversity between the samples. The second strategy computes the norm of the gradients with respect to the loss. The motivation being that larger norms indicate a larger influence on the weights so the respective samples might be more important. Using this measure, the samples are again sorted based on a log-space scale. The third strategy also computes first the norms of gradients and sorts them in a descending order. Samples are then selected in a sequential manner based on the norm of the residual gradient, which essentially is what remains from the original gradient once the gradient contribution of the selected samples is projected out.

We evaluate our strategies on two agricultural datasets that show different image characteristics. We observe an improvement in segmentation performance even

when only a small number of samples is selected for annotation, and this improvement is higher compared to random or entropy-based sampling. Our method therefore fulfills our goal of reducing the human effort needed for annotation.

6.3.1 Possible Extensions

To reduce the annotation effort we experimented with different strategies for active learning, where a small subset of images are proposed for annotation. To reduce the effort even further, one could look into proposing only a part of an image for annotation, or even individual pixels. The simplest approach would be to randomly sample portions of an image and apply the same strategies that we devised. But this sampling can also be made smarter by looking at which parts/pixels of an image are more salient than others. Adding human input to the process would also be worth investigating. A feedback loop can be generated where human made scribbles can be an input to the selection strategy and that in turn could propose regions for annotation.

6.4 Unsupervised Domain Adaptation via Self-Training

Unsupervised Domain Adaptation (UDA) attempts to solve the issue of a model's lower performance on a domain different from what it was trained on without relying on labeled data from the new domain. There are different approaches to UDA. Some researchers try to match the network responses at different layers between source and target domains, or use adversarial learning to encourage the network to learn domain-invariant features. Others use proxy tasks to create pseudo-labels that can assist in the adaptation.

Our approach to UDA uses self-training to adapt the weights of the network using target predictions. A common technique has been to keep the most-confident target predictions and refine the network using them. This runs however the risk that the network might actually be producing wrong predictions with high confidence. Therefore using these samples will further degrade the performance of the network instead of improving it.

We make use of the fact that U-NET models, similar to other deep learning architectures, successively build richer features going from one layer to the next. We add a second segmentation head to the model just before the first downsampling operation. The model now produces two segmentation maps; a rough one at the beginning of the network and a more refined one at the end of the network. We use the latter as pseudo-groundtruth for the former. In other words, we compute the loss between the two sets of segmentations and refine the weights leading up to the first segmentation head.

Our use of probablistic predictions as guidance for refinement coupled with restricting the update to the early features, reduces the potential negative effect of incorrect predictions that was described earlier. We ran experiments using our method on two datasets and compare to a baseline adaptation method. Our results show that the model is able to better adapt to the new domain and we observe that improvement qualitatively as well.

6.4.1 Possible Extensions

We use self-training in our work to adapt the model to new domains. Our loss takes as input rough and refined predictions from the target domain. It would be interesting to see the effect of complementing that with a consistency loss. A common practice in UDA is to encourage the model to produce similar predictions for different augmentations of the input image. We can use here the feature smoothing transformation described earlier to push the model to output segmentation maps that are identical for the original and smoothed images by adding a penalty to the loss that matches those two sets of predictions.

6.5 Outlook

Fully convolutional networks such as the U-Net (Ronneberger et al., 2015) have been widely used in the past years for segmentation tasks, and this is the type of models that we used in our work. Due to the popularity however of transformers (Vaswani et al., 2017) for Natural Language Processing (NLP), the same attention mechanism has been lately applied for vision tasks, resulting in models such as the Vision Transformer (ViT) (Dosovitskiy et al., 2021). Recent segmentation networks (Hatamizadeh et al., 2022) combine both convolutional layers and transfomer blocks. The methods we developed in our work such as those for UDA or active learning can be easily adapted to these mixed architectures. The improvement in performance will depend on the extent to which domain shift is impacted by the attention mechanism employed in transformers. In ViT, the attention is computed by comparing small blocks of the image to one another resulting in features that take into account the global context. This might reduce the impact of domain shift if for example an object's segmentation is driven by global dependencies within the image.

There have also been several works recently that aim to build so-called foundation models. These are often trained on large datasets that contain a diverse set of images allowing the model to learn general features that can transfer to other datasets or downstream tasks.

Ji et al., 2023 create a model that is able to segment 143 organs with CT scans as input. They start by training a network on a large dataset, namely the TotalSegmentor dataset (Wasserthal et al., 2023). They assume a setting where there is a need to adapt models to new data without having access to the previous data that it was trained on. To prevent knowledge forgetting, they freeze the encoder and sequentially train additional decoders on different datasets. To limit the size increase of the model, the decoders are pruned after being trained in order to reduce the total number of parameters. A prerequisite of this method is that the first dataset used to train the model needs to be quite extensive in terms of the number of classes and the variations in images within each class. This would enable the creation of a strong encoder with rich features that do not need to be adapted to new datasets, as the adaptation is left to the respective decoder.

Butoi et al., 2023 also train a model on a large dataset. The model is adapted to new data without re-training or fine-tuning. It is instead presented at inference time with a small set of labeled images from the target domain, called a support set, in addition to the input image that should be segmented. Convolutions between the test image and the support set enable the knowledge transfer of the new task to the trained model.

The Segment Anything Model (SAM) (Kirillov et al., 2023) is a foundation model that was trained on 11 million images and one billion masks. It encodes images using transformer-based blocks. Image embeddings are then combined with prompts to produce segmentation masks. These prompts can for example be bounding boxes of the desired object to be segmented or can be points indicating what should be considered foreground or background in an image. It can also operate in a full-segmentation mode where all objects in an image are segmented. Given the large size of the dataset it was trained on, the model can be used to segment natural images acquired in various scenes, alleviating the need to train specific models for different datasets. Several works (Mazurowski et al., 2023; Ma et al., 2024) have looked into whether this model can also be used for medical images, and found that it underperforms when compared to specialized models that were trained for a particular task. This can be explained by the fact that medical images exhibit different characteristics compared to natural images when it comes to their appearance and their semantic content. Ma et al., 2024 fine-tuned SAM on medical images and reported similar or superior performance to task-specific trained models when the model is combined with prompts. Although this is encouraging as it facilitates interactive segmentation, its dependence on bounding box prompts for each 2D image, would still result in a considerable annotation cost.

Foundation models, examples of which were presented earlier, can reduce the effect of domain shift on segmentation models. Since these models are pre-trained on a large number of diverse images, they are less susceptible to overfitting on peculiarities of a single domain. It is still however challenging to train these models. Recent works (Kirillov et al., 2023) use large transformers to build image encoders, which makes them costly in terms of hardware requirements and training time. Since these networks have a large number of parameters, fine-tuning them for specialized tasks can be prohibitive. Possible solutions include fine-tuning only a subset of the layers, e.g. those of the decoder, while freezing the others, or to decompose the weights (Hu et al., 2022a) such that only parts of them need to be replaced for each task.

Bibliography

Acuna, David et al. (2018). "Efficient interactive annotation of segmentation datasets with polygon-rnn++". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 859–868.

Ambellan, Felix et al. (2019). "Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the Osteoarthritis Initiative". In: *Medical image analysis* 52, pp. 109–118.

Andreu, Fuensanta et al. (2001). "Minimizing total variation flow". In: *Differential and integral equations* 14.3, pp. 321–360.

Arpit, Devansh et al. (2017). "A closer look at memorization in deep networks". In: *International conference on machine learning*. PMLR, pp. 233–242.

Aurich, Volker and Jörg Weule (1995). "Non-Linear Gaussian Filters Performing Edge Preserving Diffusion". In: *Mustererkennung*. Ed. by Gerhard Sagerer et al. Informatik Aktuell. Springer, pp. 538–545.

Badrinarayanan, Vijay et al. (2017). "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 39.12, pp. 2481–2495.

Bai, Fan et al. (2022). "Discrepancy-Based Active Learning for Weakly Supervised Bleeding Segmentation in Wireless Capsule Endoscopy Images". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. Springer, pp. 24–34.

Bakas, Spyridon et al. (2017). "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features". In: *Scientific data* 4, p. 170117.

Bakas, Spyridon et al. (2018). *Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge*. Tech. rep. 1811.02629. arXiv.

Beluch, William H. et al. (2018). "The power of ensembles for active learning in image classification". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 9368–9377.

Billot, Benjamin et al. (2020). *A Learning Strategy for Contrast-agnostic MRI Segmentation*. Tech. rep. 2003.01995. arXiv.

Bowles, Christopher et al. (2018). *GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks*. Tech. rep. 1810.10863. arXiv.

Budd, Samuel et al. (2021). "A survey on active learning and human-in-the-loop deep learning for medical image analysis". In: *Medical Image Analysis* 71, p. 102062.

Butoi, Victor Ion et al. (2023). "UniverSeg: Universal Medical Image Segmentation". In: *International Conference on Computer Vision*.

Cai, Lile et al. (2021). "Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10988–10997.

Campello, Víctor M et al. (2021). "Multi-centre, multi-vendor and multi-disease cardiac segmentation: the M&Ms challenge". In: *IEEE Transactions on Medical Imaging* 40.12, pp. 3543–3554.

Chaitanya, Krishna et al. (2019). "Semi-supervised and task-driven data augmentation". In: *Int'l Conf. on Information Processing in Medical Imaging (IPMI)*. Springer, pp. 29–41.

Chakraborty, Shayok et al. (2015). "Active batch selection via convex relaxations with guaranteed solution bounds". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 37.10, pp. 1945–1958.

Chebrolu, Nived et al. (2017). "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields". In: *The Intl. Journal of Robotics Research* 36.10, pp. 1045–1052.

Chen, Liang-Chieh et al. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818.

Chlap, Phillip et al. (2021). "A review of medical image data augmentation techniques for deep learning applications". In: *Journal of Medical Imaging and Radiation Oncology* 65.5, pp. 545–563.

Choi, Dami et al. (2019). "On empirical comparisons of optimizers for deep learning". In: *arXiv preprint arXiv:1910.05446*.

Cubuk, Ekin D et al. (2019). "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123.

Cubuk, Ekin D et al. (2020). "Randaugment: Practical automated data augmentation with a reduced search space". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703.

Dai, Chengliang et al. (2020). "Suggestive annotation of brain tumour images with gradient-guided sampling". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part IV 23*. Springer, pp. 156–165.

DeVries, Terrance and Graham W Taylor (2017). "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552*.

Dosovitskiy, Alexey et al. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=YicbFdNTTy.

Du, Yunshu et al. (2018). "Adapting auxiliary losses using gradient similarity". In: *arXiv preprint arXiv:1812.02224*.

Duckett, Tom et al. (2018). "Agricultural Robotics: The Future of Robotic Agriculture". In: *arXiv preprint* abs/1806.06762. arXiv: 1806.06762. URL: http://arxiv.org/abs/1806.06762.

Dutt Jain, Suyog and Kristen Grauman (2016). "Active image segmentation propagation". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2864–2873.

Eaton-Rosen, Zach et al. (2018). "Improving data augmentation for medical image segmentation". In.

Feng, Wei et al. (2022). "Unsupervised domain adaptive fundus image segmentation with category-level regularization". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 497–506.

Franchi, Gianni et al. (2021). "Robust Semantic Segmentation with Superpixel-Mix". In: *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*. BMVA Press, p. 158. URL: https://www.bmvc2021-virtualconference.com/assets/papers/0509.pdf.

Freytag, Alexander et al. (2014). "Selecting influential examples: Active learning with expected model output changes". In: *European Conf. on Computer Vision*, pp. 562–577.

Gal, Yarin et al. (2017). "Deep bayesian active learning with image data". In: *Proc. of the Intl. Conf. on Machine Learning*, pp. 1183–1192.

Gatys, Leon A. et al. (2016). "Image Style Transfer Using Convolutional Neural Networks". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423.

Geirhos, Robert et al. (2019). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *Int'l Conf. on Learning Representations (ICLR)*.

Ghamsarian, Negin et al. (2023). "Domain Adaptation for Medical Image Segmentation Using Transformation-Invariant Self-training". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 331–341.

Gomariz, Alvaro et al. (2022). "Unsupervised Domain Adaptation with Contrastive Learning for OCT Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 351–361.

Goodfellow, Ian et al. (2020). "Generative adversarial networks". In: *Communications of the ACM* 63.11, pp. 139–144.

Guyon, Isabelle et al. (2011). "Results of the active learning challenge". In: *Proc. of the AISTATS Active Learning and Experimental Design Workshop*, pp. 19–45.

Hammoudi, Karim et al. (2022). "Superpixelgridmasks data augmentation: Application to precision health and other real-world data". In: *Journal of Healthcare Informatics Research* 6.4, pp. 442–460.

Hatamizadeh, Ali et al. (2022). "Unetr: Transformers for 3d medical image segmentation". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 574–584.

He, Kaiming et al. (2010). "Guided Image Filtering". In: *Proc. European Conf. on Computer Vision (ECCV), Part I*. Ed. by Kostas Daniilidis et al. Vol. 6311. Lecture Notes in Computer Science. Springer, pp. 1–14.

Heller, Nicholas et al. (2021). "The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KiTS19 challenge". In: *Medical image analysis* 67, p. 101821.

Hendrycks, Dan et al. (2020). "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty". In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ho, Jonathan et al. (2020). "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Hoffman, Judy et al. (2016). "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation". In: *arXiv preprint arXiv:1612.02649*.

Holub, Alex et al. (2008). "Entropy-based active learning for object recognition". In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 1–8.

Hu, Edward J et al. (2022a). "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

Hu, Minhao et al. (2021). "Fully Test-Time Adaptation for Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 251–260.

Hu, Shishuai et al. (2022b). "Domain specific convolution and high frequency reconstruction based unsupervised domain adaptation for medical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 650–659.

Huang, Haoshuo et al. (2018). "Domain Transfer Through Deep Activation Matching". In: *Proc. European Conference on Computer Vision (ECCV) Part XVI*. Ed. by Vittorio Ferrari et al. Vol. 11220. LNCS. Springer, pp. 611–626.

Huang, Jiaji et al. (2016). "Active learning for speech recognition: the power of gradients". In: *arXiv preprint arXiv:1612.03226*.

Isensee, Fabian et al. (2017). "Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries - Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers*. Ed. by Alessandro Crimi et al. Vol. 10670. Lecture Notes in Computer Science. Springer, pp. 287–297.

Isensee, Fabian et al. (2021). "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature methods* 18.2, pp. 203–211.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Jackson, Philip T. G. et al. (2019). "Style Augmentation: Data Augmentation via Style Randomization". In: *CVPR Deep Vision Workshop*, pp. 83–92.

Jang, Eric et al. (2016). "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144*.

Ji, Zhanghexuan et al. (2023). "Continual segment: Towards a single, unified and non-forgetting continual segmentation model of 143 whole-body organs in ct scans". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21140–21151.

Käding, Christoph et al. (2016). "Active and continuous exploration with deep neural networks and expected model output changes". In: *arXiv preprint arXiv:1612.06129*.

Kavur, A Emre et al. (2021). "CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation". In: *Medical Image Analysis* 69, p. 101950.

Kirillov, Alexander et al. (2023). "Segment anything". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026.

Koch, Valentin et al. (2022). "Noise transfer for unsupervised domain adaptation of retinal OCT images". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 699–708.

Kuijf, Hugo J et al. (2019). "Standardized Assessment of Automatic Segmentation of White Matter Hyperintensities and Results of the WMH Segmentation Challenge". In: *IEEE Trans. on Medical Imaging* 38.11, pp. 2556–2568.

Kwak, Suha et al. (2017). "Weakly supervised semantic segmentation using superpixel pooling network". In: *Thirty-First AAAI Conference on Artificial Intelligence*.

Lee, Kyungsu et al. (2023). "Self-Supervised Domain Adaptive Segmentation of Breast Cancer via Test-Time Fine-Tuning". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 539–550.

Li, Hongwei et al. (2018). "Fully convolutional network ensembles for white matter hyperintensities segmentation in MR images". In: *NeuroImage* 183, pp. 650–665.

Li, Yunsheng et al. (2019). "Bidirectional learning for domain adaptation of semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6936–6945.

Liebisch, Frank et al. (2016). "Flourish – A robotic approach for automation in crop management". In: *In Proc. of the Workshop für Computer-Bildanalyse und unbemannte autonom fliegende Systeme in der Landwirtschaft*.

Lin, Tsung-Yi et al. (2017). "Focal Loss for Dense Object Detection". In: *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pp. 2999–3007.

Lin, Yili et al. (2023). "Multi-Target Domain Adaptation with Prompt Learning for Medical Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 717–727.

Loshchilov, Ilya and Frank Hutter (2019). "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Lottes, Philipp and Cyrill Stachniss (2017). "Semi-Supervised Online Visual Crop and Weed Classification in Precision Farming Exploiting Plant Arrangement". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. URL: http://www.ipb.uni-bonn.de/wp-content/papercite-data/pdf/lottes17iros.pdf.

Lottes, Philipp et al. (2018a). "Fully Convolutional Networks with Sequential Information for Robust Crop and Weed Detection in Precision Farming". In: *IEEE Robotics and Automation Letters (RA-L)* 3 (4), pp. 3097–3104. DOI: 10.1109/LRA.2018.2846289. URL: https://arxiv.org/abs/1806.03412.

Lottes, Philipp et al. (2018b). "Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Ma, Jun et al. (2024). "Segment anything in medical images". In: *Nature Communications* 15.1, p. 654.

Ma, Rui et al. (2019). "Optimizing Data Augmentation for Semantic Segmentation on Small-Scale Dataset". In: *Proc. Int'l Conf. on Control and Computer Vision (ICCCV)*, pp. 77–81.

Mazurowski, Maciej A et al. (2023). "Segment anything model for medical image analysis: an experimental study". In: *Medical Image Analysis* 89, p. 102918.

McCool, Chris et al. (2017). "Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics". In: *IEEE Robotics and Automation Letters (RA-L)*.

McCool, Chris et al. (2018). "Efficacy of Mechanical Weeding Tools: A Study into Alternative Weed Management Strategies Enabled by Robotics". In: *IEEE Robotics and Automation Letters (RA-L)*.

Menze, Bjoern H et al. (2014). "The multimodal brain tumor image segmentation benchmark (BRATS)". In: *IEEE Trans. on Medical Imaging* 34.10, pp. 1993–2024.

Milioto, Andres and Cyrill Stachniss (2019). "Bonnet: An Open-Source Training and Deployment Framework for Semantic Segmentation in Robotics using CNNs". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Milioto, Andres et al. (2017). "Real-time Blob-wise Sugar Beets vs Weeds Classification for Monitoring Fields using Convolutional Neural Networks". In: *Proc. of the Intl. Conf. on Unmanned Aerial Vehicles in Geomatics*. URL: http://www.ipb.uni-bonn.de/pdfs/milioto17uavg.pdf.

— (2018). "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns". In: *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 2229–2235.

Munjal, Prateek et al. (2022). "Towards robust and reproducible active learning using neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 223–232.

Nalepa, Jakub et al. (2019). "Data augmentation for brain-tumor segmentation: a review". In: *Frontiers in computational neuroscience* 13, p. 83.

Nath, Vishwesh et al. (2022). "Warm start active learning with proxy labels and selection via semi-supervised fine-tuning". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. Springer, pp. 297–308.

Nikolov, Stanislav et al. (2018). "Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy". In: *arXiv preprint arXiv:1809.04430*.

Parvaneh, Amin et al. (2022). "Active learning by feature mixing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12237–12246.

Paszke, Adam et al. (2016). "Enet: A deep neural network architecture for real-time semantic segmentation". In: *arXiv preprint arXiv:1606.02147*.

Perone, Christian S et al. (2019). "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling". In: *NeuroImage* 194, pp. 1–11.

Pinaya, Walter H. L. et al. (2022). "Brain Imaging Generation with Latent Diffusion Models". In: *Deep Generative Models - Second MICCAI Workshop, DGM4MICCAI 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings*. Ed. by Anirban Mukhopadhyay et al. Vol. 13609. Lecture Notes in Computer Science. Springer, pp. 117–126.

Prados, Ferran et al. (2017). "Spinal cord grey matter segmentation challenge". In: *NeuroImage* 152, pp. 312–329.

Ren, Pengzhen et al. (2021). "A survey of deep active learning". In: *ACM computing surveys (CSUR)* 54.9, pp. 1–40.

Ronneberger, Olaf et al. (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Rudin, Leonid I et al. (1992). "Nonlinear total variation based noise removal algorithms". In: *Physica D* 60.1, pp. 259–268.

Sa, Inkyu et al. (2018). "WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming". In: *Remote Sensing* 10 (9). DOI: 10.3390/rs10091423. URL: http://www.mdpi.com/2072-4292/10/9/1423/pdf.

Sandfort, Veit et al. (2019). "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks". In: *Scientific Reports* 9.1, p. 16884.

Savarese, Pedro et al. (2021). "Domain-independent dominance of adaptive methods". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16286–16295.

Sener, Ozan and Silvio Savarese (2017a). "A geometric approach to active learning for convolutional neural networks". In: *arXiv preprint arXiv* 1708, p. 1.

— (2017b). "Active learning for convolutional neural networks: A core-set approach". In: *arXiv preprint arXiv:1708.00489*.

Settles, Burr (2009). *Active learning literature survey*. Tech. rep. Univ. of Wisconsin-Madison, Dep. of Computer Sciences.

Settles, Burr et al. (2008). "Multiple-instance active learning". In: *Advances in Neural Information Processing Systems*, pp. 1289–1296.

Shaw, Richard et al. (2019). "MRI k-Space Motion Artefact Augmentation: Model Robustness and Task-Specific Uncertainty". In: *International Conference on Medical Imaging with Deep Learning*. PMLR, pp. 427–436.

Sheikh, Rasha and Thomas Schultz (2020). "Feature Preserving Smoothing Provides Simple and Effective Data Augmentation for Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part I*. Ed. by Anne L. Martel et al. Vol. 12261. Lecture Notes in Computer Science. Springer, pp. 116–126. DOI: 10.1007/978-3-030-59710-8_12. URL: https://doi.org/10.1007/978-3-030-59710-8_12.

— (2022). "Unsupervised Domain Adaptation for Medical Image Segmentation via Self-Training of Early Features". In: *International Conference on Medical Imaging with Deep Learning, MIDL 2022, 6-8 July 2022, Zurich, Switzerland*. Ed. by Ender Konukoglu et al. Vol. 172. Proceedings of Machine Learning Research. PMLR, pp. 1096–1107. URL: https://proceedings.mlr.press/v172/sheikh22a.html.

Sheikh, Rasha et al. (2020). "Gradient and Log-based Active Learning for Semantic Segmentation of Crop and Weed for Agricultural Robots". In: *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*. IEEE, pp. 1350–1356. DOI: 10.1109/ICRA40945.2020.9196722. URL: https://doi.org/10.1109/ICRA40945.2020.9196722.

Sheikh, Rasha et al. (2022). "Adaptive Optimization with Fewer Epochs Improves Across-Scanner Generalization of U-Net Based Medical Image Segmentation". In: *Domain Adaptation and Representation Transfer - 4th MICCAI Workshop, DART 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings*. Ed. by Konstantinos Kamnitsas et al. Vol. 13542. Lecture Notes in Computer Science. Springer, pp. 119–128. DOI: 10.1007/978-3-031-16852-9_12. URL: https://doi.org/10.1007/978-3-031-16852-9_12.

Shirokikh, Boris et al. (2020). "First U-Net Layers Contain More Domain Specific Information Than The Last Ones". In: *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*. Springer, pp. 117–126.

Shorten, Connor and Taghi M. Khoshgoftaar (2019). "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6.1, p. 60.

Siddiqui, Yawar et al. (2020). "Viewal: Active learning with viewpoint entropy for semantic segmentation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9433–9443.

Smailagic, Asim et al. (2018). "MedAL: Accurate and Robust Deep Active Learning for Medical Image Analysis". In: *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*. Ed. by M. Arif Wani et al. IEEE, pp. 481–488. DOI: 10.1109/ICMLA.2018.00078. URL: https://doi.org/10.1109/ICMLA.2018.00078.

Souza, Roberto et al. (2018). "An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement". In: *NeuroImage* 170, pp. 482–494.

Sun, Xiaoyi et al. (2022). "Attention-Enhanced Disentangled Representation Learning for Unsupervised Domain Adaptation in Cardiac Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 745–754.

Sun, Yu et al. (2019). "Unsupervised domain adaptation through self-supervision". In: *arXiv preprint arXiv:1909.11825*.

Takikawa, Towaki et al. (2019). "Gated-scnn: Gated shape cnns for semantic segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5229–5238.

Tang, Meng et al. (2018). "Normalized cut loss for weakly-supervised CNN segmentation". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1818–1827.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Vu, Tuan-Hung et al. (2019). "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2517–2526.

Wang, Haohan et al. (2020). "High-frequency component helps explain the generalization of convolutional neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694.

Wang, Keze et al. (2017). "Cost-effective active learning for deep image classification". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.12, pp. 2591–2600.

Wang, Zuhui and Zhaozheng Yin (2021). "Annotation-efficient cell counting". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*. Springer, pp. 405–414.

Wasserthal, Jakob et al. (2023). "Totalsegmentator: Robust segmentation of 104 anatomic structures in ct images". In: *Radiology: Artificial Intelligence* 5.5.

Wei, Yunchao et al. (2018). "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7268–7277.

Weickert, Joachim et al. (1998). "Efficient and reliable schemes for nonlinear diffusion filtering". In: *IEEE Trans. on Image Processing* 7.3, pp. 398–410.

Wilson, Ashia C et al. (2017). "The marginal value of adaptive gradient methods in machine learning". In: *Advances in neural information processing systems* 30.

Woodward, Mark and Chelsea Finn (2017). "Active one-shot learning". In: *arXiv preprint arXiv:1702.06559*.

Wu, Yixuan et al. (2022). "Self-learning and One-Shot Learning Based Single-Slice Annotation for 3D Medical Image Segmentation". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*. Springer, pp. 244–254.

Xu, Zihang et al. (2023). "ASC: Appearance and Structure Consistency for Unsupervised Domain Adaptation in Fetal Brain MRI Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 325–335.

Yang, Lin et al. (2017). "Suggestive annotation: A deep active learning framework for biomedical image segmentation". In: *Proc. of the Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 399–407.

Yoo, Donggeun and In So Kweon (2019). "Learning Loss for Active Learning". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 93–102.

Zakazov, Ivan et al. (2021). "Anatomy of Domain Shift Impact on U-Net Layers in MRI Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 211–220.

Zhang, Hongyi et al. (2018a). "mixup: Beyond Empirical Risk Minimization". In: *International Conference on Learning Representations*.

Zhang, Ling et al. (2018b). "Self-learning to detect and segment cysts in lung CT images without manual annotation". In: *IEEE Intl. Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1100–1103.

Zhang, Yizhe et al. (2019). "SPDA: Superpixel-based Data Augmentation for Biomedical Image Segmentation". In: *Int'l Conf. on Medical Imaging with Deep Learning (MIDL)*. Vol. 102. Proceedings of Machine Learning Research, pp. 572–587.

Zhao, Amy et al. (2019). "Data augmentation using learned transformations for one-shot medical image segmentation". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 8543–8553.

Zhong, Zhun et al. (2020). "Random erasing data augmentation". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07, pp. 13001–13008.

Zhou, Zongwei et al. (2017). "Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally". In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 7340–7351.

Zhu, Jun-Yan et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Zotti, Clement et al. (2018). "Convolutional neural network with shape prior applied to cardiac MRI segmentation". In: *IEEE journal of biomedical and health informatics* 23.3, pp. 1119–1128.

Zou, Yang et al. (2018). "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 289–305.