PROJECT REPORT

Semantic Segmentation of Wounds

Cay Rahn
6255648

Course: KEN4244 Deep Learning for Image & Video Processing
Academic Year: 2023/24

December 19, 2023

1 Introduction

1.1 Motivation

Many people are affected by chronic wounds that need to be monitored to detect infections and ensure proper healing of the wound [12]. Other wounds, e.g., surgical wounds need to be monitored as well. Such monitoring might be necessary over long time span [12]. Monitoring by visual assesment yields the risk of missing changes and therefore wound segmentation based on images is needed. Without an algorithm to automise this process, a manual segmentation by experts is needed. However, manual segmentation is not only very expensive and time consuming but experts also differ in the provided segmentation.

An automisation is complex due to various reasons. On one side, the characteristics of wounds itself hold challenges: Wounds have a complex structure and contain different types of tissues with different colours and textures [1]. This means, there are not only borders between wound and healthy tissue but also in the wound itself. There exist also many different types of wounds, e.g. diabetic foot ulcers, surgical wounds, decubitus and many more that have different characteristcs. On the other side, the images itself are very heterogenous. Lightning conditions, the distance to the camera, the camera angle and event the camera itself change the images in a way that they have an impact on the resulting segmentation. Creating a controlled environment sounds like the desired solution but is not feasible in a clinical setting. Ideally, it should be possible to take pictures with a smartphone without overly complicated instructions for the person taking the picture. A clinical employee without further experience as photographer should be able to take pictures that are then segmented correcty.

1.2 Research Questions

A recent publication by Oota et al. claims to have improved the state of the art in the field of Wound Segmentation. Such claims always need to be supported by further research. This project aims to investigate and reimplement the proposed method. Furthermore, the method is set into context with state-of-the-art methods for semantic segmentation in general and for wounds specifically.

- can the results be reproduced?
- what influence does the input image size have? Can we rescale the images and are able to transfer what is learned
- how robust is the model/architectures to transformations/distortions on the input
- XAI

2 Datasets

Unfortunately, few datasets on chronic wounds are publicly available [20]. They often feature only a specific type of chronic wound, mainly diabetic or pressure ulcer. An example of such a specialised dataset is the data from the Diabetic Foot Ulcer Challenge 2022 [12]. However, it is only available after application and, therefore, is inappropriate for this project due to its limited timescope. Another data set featuring foot ulcer wounds is publicly available in the Foot Ulcer Segmentation Challenge 2021 [23]. It consists of 1010 images, which are augmented to build a data set with a training set of 3645 images and a test set of 405 images. Due to the nature of the challenge, labels for the test set are unavailable.

WSNet data set The data set mainly used in the scope of this project is the WSNet data set featuring eight different wound types: venous ulcer, trauma wound, diabetic ulcer, surgical wound, arterial ulcer, cellulitis, pressure ulcer and a not further specified group of other wounds [20, 19]. In total, it consists of 2686 images and their corresponding masks. With this, it consists of more individual images and wound types than the publicly available data set mentioned above. Unfortunately, the wound classification itself is not available. Furthermore, Oota et al. mention another separate data set for pre-training with wound type classification, which is also not publicly available.

3 State of the Art

3.1 Semantic Segmentation

The segmentation of wounds belongs to the class of semantic segmentation problems, where a pixel-wise classification is performed. In the case of wound segmentation, there are two classes: foreground, which is the wound, and background. Deep Learning methods have become dominant in the last few years because they have become more accessible. Fully Convolutional Neural Networks (fCNN) as a starting point in research had the drawback of resulting in a low output resolution, and multiple techniques were invented to increase the output resolution [15]. This results in an encoder-decoder architecture as the base for the networks, inspired by auto-encoders [4], where the encoder subsamples and the decoder upsamples [18]. In such architectures, the encoder generates context information, information in the feature space, while the decoder maps this information into the spatial context.

Pre-training for such models requires a vast amount of data. The typical object classification data set for such pre-training is the ImageNet data set [2].

This project uses four segmentation models:: U-Net, LinkNet, FPN and PSPNet. All are improved architectures about a basic fCNN. Each architecture is described in detail to understand the challenges and approaches to localising information in space.

U-Net U-Net is a convolutional network developed for biomedical image segmentation based on an encoder-decoder architecture. Encoder and decoder are called contracting and expansive paths in the original paper, describing their function. They are also described as context and spatial paths [17]. Both the encoder and decoder consist of different steps to encode and decode the image on different spatial levels. The encoder is a classical CNN; each step consists of two convolutions and a max pooling operation for downsampling. The decoder step upsamples the feature map followed by a convolution. The result is then concatenated with the corresponding feature map from the encoder path, and convolution is applied again. In the final layer, 1x1 convolution maps the feature vector to the desired number of classes. This architecture is visualised in Figure 1. [21]

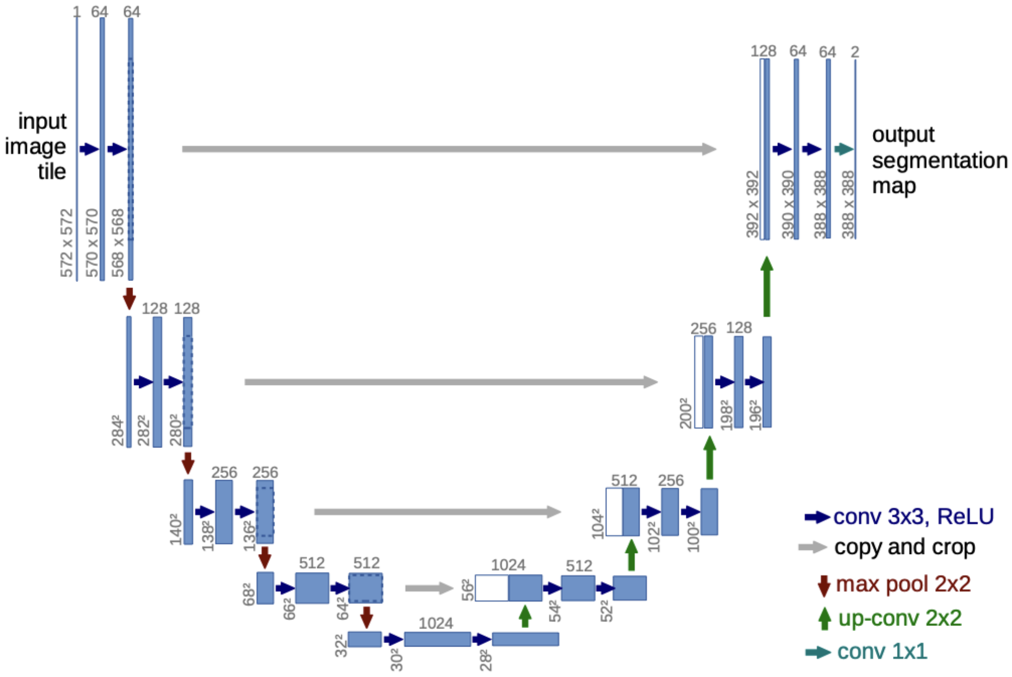


Figure 1: U-Net architecture for 32x32 pixels in the lowes resolution. Blue boxes are feature maps with the number of feature-channels on top of the boxes and the size shown on the left size. Operations are indicated by the arrows. The skip connections are a concatenation. The figure originally create by Ronneberger et al. [21].

The skip connections, connecting the different levels of encoder and decoder prevent a loss of information and extract the features at different resolutions to retrieve spatial information. By doing this, it is one of the first architectures to improve the classical fCNN for semantic segmentation [15]. While U-Net provides spatial localisation of features, its ability to generalise to multi-scale information is limited [18].

One restriction is that the input size must be chosen to apply all 2x2 max-pooling operations in the encoder to an even x and y size.

LinkNet Like U-Net, LinkNet consists of an encoder block for downsampling and a decoder block for upsampling. The downsampling is not done by max pooling as in the U-Net architecture but by using a stride of 2 in a convolutional layer. The initial encoder block also differs from the following blocks as it uses a larger kernel and max pooling. The decoder blocks upsample by a factor of 2 in each block. The final block differs again from the previous blocks. The main difference to the U-Net architecture is how the skip connections are used: Similarly to the U-Net, there are skip connections between the corresponding steps of the encoder and decoder, but the feature map from the encoder is not concatenated but added to decoder data. The LinkNet architecture is visualised in Figure 2. [4]

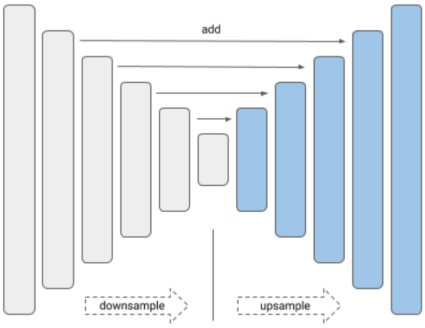


Figure 2: A visualisation of the LinkNet architecture originally provided by Iakubovskii [11].

The implementation used in this project has four skip connections instead of the original three [11]. Similarly to the U-Net, the input size is restricted, so that every upsampling operation can be applied to an even x and y size.

LinkNet has been shown to achieve better results than U-Net under similar conditions [8].

FPN The Feature Pyramid Network (FPN) architecture creates feature maps of various sizes in multiple layers [18]. Like the other architectures, it consists of an encoder and a decoder, called bottom-up and top-down pathways here[14]. Similarly to U-Net, feature maps at different scales with a scaling step of 2 are created in the encoder [14]. In the decoder, the feature maps are upsampled and combined with the encoder information of the same level. Similarly to LinkNet, addition is used in the skip connections, but a 1x1 convolution is applied. By doing this, so-called feature pyramids are built, containing features at different resolutions. Kirillov et al. proposed using these feature pyramids to obtain a segmentation by merging the feature maps using addition or concatenation [13, 11]. The architecture is visualised in Figure 3.

PSPNet The central part of the Pyramid Scene Parsing Network (PSPNet) is the pyramid pooling module, visualised in part c of Figure 4, which extracts context information at different scales. A feature map extracted with a pre-trained backbone is pooled at different pyramid scales. This means that global pooling and sub-regions on different locations are used to extract features from a global to a more fine-grained scale. Each of those scales is passed through a 1x1 convolution and afterwards upsampled to the size of the original feature map. All feature maps, including the original feature map from the backbone, are concatenated and used to extract the final prediction. By this, features on different scales are combined. [24, 9]

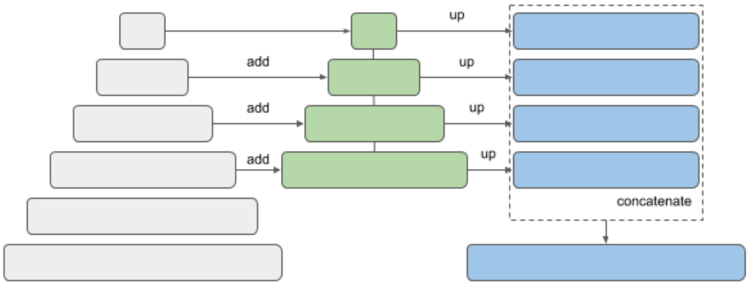


Figure 3: A visualisation of the FPN architecture originally provided by Iakubovskii [11]. Note, that the feature maps can combined either by concatenation or addition.

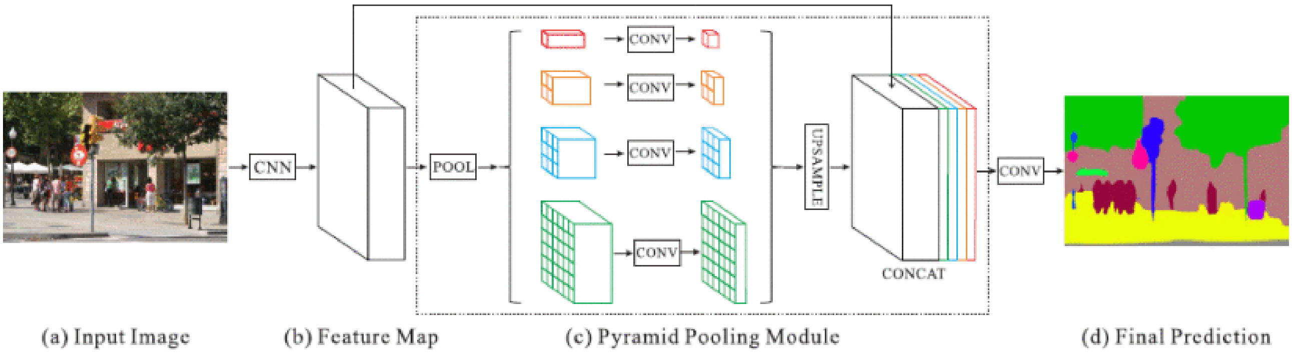


Figure 4: Visualization of the PSPNet-architecture. Originally created by Zhao et al. [24].

3.1.1 Optimisation and Evaluation

Several methods exist to evaluate how good a predicted segmentation is. Since semantic segmentation performs a pixel-wise classification, resulting in a segmentation mask, classical metrics such as accuracy and precision are available. Two performance metrics commonly used in semantic segmentation in medical imaging are the Dice Coefficient and the Intersection over Union (IoU) score. They indicate the segmentation quality better than pixel-wise accuracy [7].

IoU-Score The IoU-Score (Intersection over Union), also known as the Jaccard index J, describes the ratio between the intersection of the ground truth mask y and the predicted mask $\tilde{y}$ and the union of the predicted and the ground truth mask. This compares the similarity of the two masks [5].

$$\begin{aligned} \mathrm{IoU}(y, \tilde{y} :&= \frac{\text{Area of overlap}}{\text{Area of union}} \qquad (1) \\ &= \frac{|y \cap \tilde{y}|}{|y \cup \tilde{y}|} \qquad (2) \end{aligned}$$

Dice Coefficient The Dice coefficient is the F1 score calculated for the image masks. Regarding intersection and union, it calculates the ratio between two times the overlap between ground truth y and predicted mask $\tilde{y}$ and the total area.

$$\begin{aligned}\text{Dice}(y, \tilde{y}) :&= 2 \cdot \frac{\text{Area of overlap}}{\text{Total area}} && (3)\\ &= 2 \cdot \frac{|y \cap \tilde{y}|}{|y| + |\tilde{y}|} && (4)\end{aligned}$$

To gain more insight into the type of errors the model makes, the rate of false positives and false negatives can be reported and then used to differentiate Type I and Type II errors [12].

Loss function Although the Dice Coefficient and IOU-Score are the most commonly used evaluation metrics, pixel-wise cross-entropy is often used as a loss function [7]. Since it is shown that there is no direct link between pixel-wise cross-entropy, either weighted or not, and Dice Coefficient and IoU-Score, this choice does not make sense because it does not optimise towards the goal.

Differentiable approximations for Dice Coefficient and IoU-Score exist that can be used to perform more goal-orientated optimisation. Both metrics and loss functions are shown to be linked to each other [7].

3.1.2 Data Augmentation

Data Augmentation is a valuable technique to make trained models more robust and accurate. This is especially true if available data is limited, as the data set size can be increased or the data set can be made more diverse.

For images, there exist several different possible augmentations. First, there are different positional augmentations, including cropping, flipping, rotating and resizing the image. Another class of augmentations is colour augmentation, which changes the image's brightness, contrast or saturation. Other augmentations include blurring and dropouts.

Not every augmentation is appropriate for every application. Rotating images of standing animals by 180 degrees, for example, would not make sense, while the rotation of wound images is appropriate. Therefore, augmentations must be chosen carefully depending on the application.

3.2 Wound Segmentation

As already discussed in the motivation of this project, wound segmentation is a complex problem due to wound characteristics such as different tissues and, therefore, edges inside of a wound itself on one side and technical reasons such as, e.g. varying lightning, distance to the wound and different angles.

Before Deep Learning became easily accessible and popular, methods based on features describing colour and textures, region-growing with optimal thresholding algorithms, and classical machine learning models were used to perform segmentation [22]. Convolutional Neural Networks replaced manually extracted features with autonomously learned ones [22]. Some methods included pre-processing steps to remove the background by, e.g., user interaction indicating the background, using a standardized background when taking the image, or using manual feature engineering to detect the background more efficiently and make the wound segmentation task easier. Such non-automatic steps limit the use of the segmentation algorithms because they either require more resources in the image-taking process or the segmentation process or are specifically tailored for specific lighting conditions or camera settings.

The Diabetic Foot Ulcer Challenge 2022 used FCN, U-Net and SegNet with different backbones and categorical cross-entropy loss as the baseline for their challenge, indicating those methods reflect the current state of the art that needs to be improved [12]. Generally, such classic models are commonly used and extended with minor adaptions. Two methods stood out in the performed literature review, including more sophisticated adaptions.

Scebba et al. proposed a method consisting of two steps: An object detection step that produces bounding boxes containing the wounds and a second step that performs segmentation on those areas. Segmentation is performed using classical architectures for semantic segmentation, as described in section 3.1. The loss function used was a pixel-wise weighted binary cross-entropy loss. Weighting was calculated based on the number of wound pixels and background pixels of each training set fold.

Oota et al. claim they set a new state of the art for wound segmentation while providing a data set together with their work [20]. The latter made them a suitable method for further investigation in this project's scope. Their approach is described in more detail in the following section.

3.2.1 WSNet

The framework proposed by Oota et al. uses the previously described segmentation architectures: U-Net, LinkNet, PSPNet, and FPN. Experiments with different backbones were performed in their work. However, in this project's scope, MobileNet [10] is mainly used since it is the smallest one, which allows faster training, which is needed in the limited time of the project.

ImageNet pre-trained weights are used. WSNet also describes pre-training specific to wounds, called Wound-Domain Adaptive Pre-training. During this pre-training, wound images are classified into five different ulcer types.

Oota et al. experimented with data augmentation on the training data and the corresponding masks, including optical distortion, horizontal flip, random rotation, blur, and more, to make their models more robust.

Global-Local Architecture The network architecture proposed by Oota et al. is called Global-Local architecture. It consists of two segmentation models, a global model and a local model, combining the result for the final segmentation. The global model is a standard segmentation model of one of the four architectures described in section 3.1. In the global model, the image (size 192x192x3) is split into 16 non-overlapping patches, resulting in a size of 48x48x3 per patch. The patches are then stacked, resulting in a size of 48x48x(3x16). The patches are the input to 16 local models in parallel, with shared weights between the local models. The output is eventually combined to obtain a full-sized mask. This mask and the output of the global model are concatenated to a mask of size 192x192x2, and a final convolution of size 1x1 results in the predicted mask. This architecture is motivated by the need to combine global signals from the entire image and local signals from smaller patches for more details. Only capturing local signals might cause an incomplete segmentation for large wounds. [20]

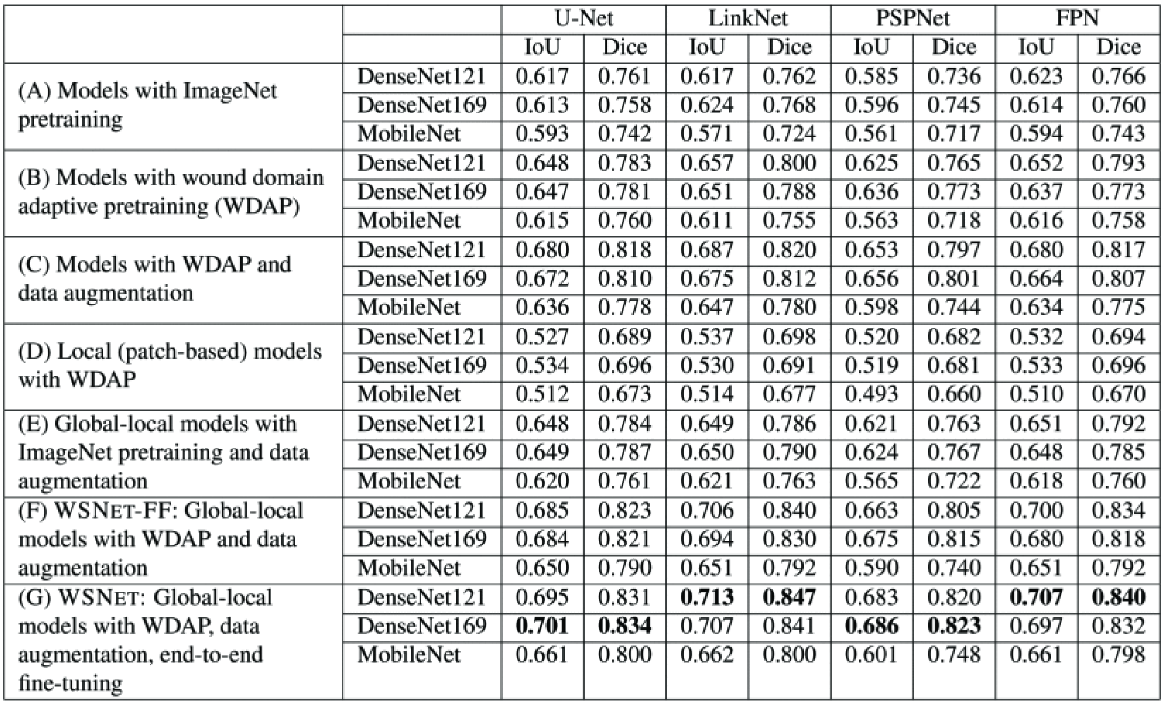
Although the combination of global and local signals sounds reasonable initially, it is interesting that this approach is combined with segmentation models that already contain different context sizes and localisation in these context sizes, as described in section 3.1. An explicitly chosen patch size implies some property of the wound images that makes this size particularly important for local information. Oota et al. stated in their paper that they tested different patch sizes and chose 48 because it led to the best results [20], supporting the theory that this patch size yields more information than others.

Reported Results Oota et al. report that wound-specific pre-training improves the resulting segmentation. Furthermore, they find that data augmentation leads to improvements. As expected, the local-only models perform significantly worse than using the segmentation model as intended in a global way. Furthermore, they find that combining global and local models leads to an improved performance compared to the global models. This indicates that the chosen patch size yields valuable information for the segmentation. The best results are therefore achieved with wound-specific pre-training, data augmentation and the invented Global-Local architecture.

Reporting such results always leads to the question of whether they are reproducible. This is especially the case with the Global-Local architecture with a specific patch size, leading to questioning whether the patch size can be generalised to other image sizes and other data sets.

4 WSNet

The following section describes the assessment and evaluation of WSNet. This starts with an implementation of the tested models and with checking for deviations between described methods and available code. Following that, several experiments are performed to assess the algorithm in more detail and to check its robustness.



		U-Net		LinkNet		PSPNet		FPN	
		IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
(A) Models with ImageNet pretraining	DenseNet121	0.617	0.761	0.617	0.762	0.585	0.736	0.623	0.766
	DenseNet169	0.613	0.758	0.624	0.768	0.596	0.745	0.614	0.760
	MobileNet	0.593	0.742	0.571	0.724	0.561	0.717	0.594	0.743
(B) Models with wound domain adaptive pretraining (WDAP)	DenseNet121	0.648	0.783	0.657	0.800	0.625	0.765	0.652	0.793
	DenseNet169	0.647	0.781	0.651	0.788	0.636	0.773	0.637	0.773
	MobileNet	0.615	0.760	0.611	0.755	0.563	0.718	0.616	0.758
(C) Models with WDAP and data augmentation	DenseNet121	0.680	0.818	0.687	0.820	0.653	0.797	0.680	0.817
	DenseNet169	0.672	0.810	0.675	0.812	0.656	0.801	0.664	0.807
	MobileNet	0.636	0.778	0.647	0.780	0.598	0.744	0.634	0.775
(D) Local (patch-based) models with WDAP	DenseNet121	0.527	0.689	0.537	0.698	0.520	0.682	0.532	0.694
	DenseNet169	0.534	0.696	0.530	0.691	0.519	0.681	0.533	0.696
	MobileNet	0.512	0.673	0.514	0.677	0.493	0.660	0.510	0.670
(E) Global-local models with ImageNet pretraining and data augmentation	DenseNet121	0.648	0.784	0.649	0.786	0.621	0.763	0.651	0.792
	DenseNet169	0.649	0.787	0.650	0.790	0.624	0.767	0.648	0.785
	MobileNet	0.620	0.761	0.621	0.763	0.565	0.722	0.618	0.760
(F) WSNet-FF: Global-local models with WDAP and data augmentation	DenseNet121	0.685	0.823	0.706	0.840	0.663	0.805	0.700	0.834
	DenseNet169	0.684	0.821	0.694	0.830	0.675	0.815	0.680	0.818
	MobileNet	0.650	0.790	0.651	0.792	0.590	0.740	0.651	0.792
(G) WSNet: Global-local models with WDAP, data augmentation, end-to-end fine-tuning	DenseNet121	0.695	0.831	**0.713**	**0.847**	0.683	0.820	**0.707**	**0.840**
	DenseNet169	**0.701**	**0.834**	0.707	0.841	**0.686**	**0.823**	0.697	0.832
	MobileNet	0.661	0.800	0.662	0.800	0.601	0.748	0.661	0.798

Table 1: Results reported by Oota et al. [20].

4.1 Code availability and reproduction of the results

Although the code for WSNet [20] is stated to be publicly available, a closer inspection of the linked GitHub repository shows that this is only partially the case. A lack of documentation makes using the code hard, especially since it seems to contain multiple errors, making it only suitable as a base for new code.

In this project's scope, the code was used to create runnable models again. Unfortunately, the classes of the wounds are not available, making it impossible to perform pre-training as described in the original paper [20]. In total, there are eight models available: A local model and a combined global-local model for each of the segmentation models: U-Net, PSPNet, FPN, and LinkNet. The Python library used for the segmentation models is `segmentation_models` [11]. The implementation process showed some differences from the described model architecture. In particular, it was claimed that the wound images were split up in parts of 48 px times 48 px. However, three of the four models, all besides PSPNet, only allow input sizes divisible by 32, and the code in GitHub showed a size of 64 px was used. Another difference between the available code and the paper is that it is claimed that augmentation is not performed on the test images, which is not the case.

Information about the training, validation and test set size is not given in the paper or code. However, the code reveals that the train and validation sets were just the first x % of the dataset, and no randomisation was used to separate the test set as it is usually done. In this project's scope, a split of 70 % training, 15 % validation and 15 % test data is used.

Because the data training for the wound-specific pre-training is not available, the results can only be compared for imagenet pre-training. An important aspect is, that MobileNet has no pretrained weights for images with the size of the patches and the default size of 224x224 px is used instead, which might impact results negatively.

The used loss and activation functions were a Dice-Loss function and sigmoid as activation function.

4.2 Comparison of the achieved performance

The results achieved with the 70:15:15 split of the data and MobileNet as backbone are shown in Table 2. Generally, the results are comparable to the results reported by Oota et al. (shown in Table 1), although the performance achieved in this project is slightly lower. Some deviations are bigger, for example the Global-Local model with U-Net achieved an IoU score of 0.495 while Oota et al. reported a score of 0.620 (Dice score 0.658

vs. 0.763). Other scores are closer, e.g., the Global-Local model with LinkNet that achieved IoE scores of 0.618 vs. 0.621 respectively and a Dices score of 0.763. Such deviations could be based in a different training size.

	U-Net		LinkNet		PSPNet		FPN	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
Local model	0.359	0.523	0.398	0.564	0.373	0.538	0.408	0.574
Global model	0.504	0.668	0.631	0.772	0.458	0.627	0.632	0.772
Global-Local model	0.495	0.658	0.618	0.763	0.476	0.642	0.612	0.758

Table 2: IoU-Scores and Dice Coefficients for the four different models with each Global-Local, Global and Local architecture. The backbone used is mobilenet.

A more important finding is that in the re-implemented models the Global-Local model does not neccessarily lead to an improved performance compared to the Global model. So the additional use of local information in a specific region size dose not necessarily improve the overall results. However, the results reported here are only for MobileNet as backbone where no weights specific to the patch size could be loaded. But even with a slight improved performance it is still arguable that the use of a Global-Local model is not the best choice since it includes some drawbacks: First, it is computationally significantly more complex and second implies that the chosen patch size has a special meaning.

4.3 Experiments with the activation function

The original papers describing the four used model architectures mostly mention ReLU as activation function. However, Oota et al. use Sigmoid in all their experiments. Because no clear findings were find on whether one of the activation functions is more appropriate, an experiment comparing both for all model architectures was performed.

		U-Net		LinkNet		PSPNet		FPN	
	Activation	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
Local model	Sigmoid	0.359	0.523	0.398	0.564	0.373	0.538	0.408	0.574
	ReLU	0.398	0.565	0.396	0.561	0.372	0.536	0.380	0.546
Global model	Sigmoid	0.504	0.668	0.631	0.772	0.458	0.627	0.632	0.772
	ReLU	0.513	0.676	0.509	0.672	0.463	0.631	0.505	0.669
Global-Local model	Sigmoid	0.495	0.658	0.618	0.763	0.476	0.642	0.612	0.758
	ReLU	0.498	0.662	0.588	0.738	0.569	0.724	0.610	0.756

Table 3: IoU-Scores and Dice Coefficients for the four different models with each Global-Local, Global and Local architecture compared for the Sigmoid and ReLU activation function.

The results, reported in Table 3, show that there is no clear trend of one of the functions performaning higher on all model architectures. Therefore, the chosen activation function is excluded as factor for further experiments.

4.4 Combination of different architectures

Since all four model architectures localize signals differently by design, the resulting local and global feature information also differ from each other. Following Oota et al. and assuming the inclusion of local models increases the performance, it might be interesting to combine different architectures in the Global-Local models. This was assesed by training and evaluating all possible combinations of the models in the Global-Local architectures. The results are reported in Table 4.

TODO: interpret results

4.5 Assessing the Robustness

As already discussed, one problem of wound segmentation is the diversity of available wound images. A segmentation should therefore be robust and work for a variety of image types. To asses the robustness of the models, two experiments were performed: Testing the performance on augmented images and testing the performances on another data set without re-training the models.

Global Model	Local Model	IoU	Dice
U-Net	U-Net	0.495	0.658
	LinkNet	0.602	0.749
	PSPNet	0.607	0.753
	FPN	0.613	0.757
LinkNet	LinkNet	0.618	0.763
	U-Net	0.633	0.774
	PSPNet	0.612	0.757
	FPN	0.613	0.758
PSPNet	PSPNet	0.476	0.642
	U-Net	0.554	0.711
	LinkNet	0.576	0.729
	FPN	0.580	0.732
FPN	FPN	0.612	0.758
	U-Net	0.605	0.752
	LinkNet	0.585	0.735
	PSPNet	0.627	0.769

Table 4: The performance of Global-Local models with all possible architecture combinations. MobileNet is used as backbone and sigmoid as activation function

4.5.1 Robustness against image augmentations

Augmentations are, as already discussed, commonly performed to improve the robustness of models. Analogously, augmentations can be used to access the robustness of the resulting model during testing. For the clinical application ouf wound segmentation, this includes various lightning conditions, varying image quality and various image sizes. To assess the models' robustness regarding this conditions, augmentations are performed on the test set. Because a trained model ideally should be able to deal with data acquired from different settings that was not included in training, no specific training for the tested augmentations was done.

To augment the training set, tensorflow image functions were used. This makes it possible to use the already created test dataset containing tensors and transform it in a specific way. The implemented and evaluated augmentations are the following:

Embed The test images were resized to be smaller than the original. The resized image was positioned in the center and remaining space was filled with black colour. This tests the model against changes in size of the wound and background changes, which is a very likely case for a real-world application. Additionally, the models can only deal with square images and rescaling and padding non-square images is a likely solution for images in other formats.

Brightness Changes in brightness are likely to happen as well. A brightness change with a delta of 0.1 is used in the experiment but can be adapted to test multiple scenarios.

Saturation This augmentation adapts the saturation of the image, which can happen due to various changes in the camera setting. The saturation is changed by a factor of 2 in this experiment.

Contrast A change in contrast might happen due to different camera and lightning settings. In this augmentation, the saturation is increased by a factor of 2 in this experiment.

In the scope of this project, only one augmentation is done at a time, although the experiment can be extended to include more augmentations and combinations of multiple simultaneous augmentations. The results are reported in Table 5. Generally, changes in brightness influence the performance the least.

- local models generally worst
- embed did not generally impacts global-local model the most, but for fpn, it is significantly worse than when just using a global model

TODO

		U-Net		LinkNet		PSPNet		FPN	
	Augmentation	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
Local model	-	0.359	0.523	0.398	0.564	0.373	0.538	0.408	0.574
	Embed	0.365	0.528	0.378	0.545	0.373	0.534	0.383	0.550
	Brightness	0.341	0.503	0.391	0.557	0.348	0.510	0.417	0.583
	Contrast	0.297	0.454	0.286	0.442	0.270	0.422	0.206	0.338
	Saturation	0.310	0.470	0.284	0.396	0.245	0.390	0.211	0.346
Global model	-	0.504	0.668	0.631	0.772	0.458	0.627	0.632	0.772
	Embed	0.400	0.566	0.438	0.607	0.333	0.497	0.454	0.622
	Brightness	0.500	0.663	0.629	0.770	0.452	0.620	0.625	0.767
	Contrast	0.404	0.573	0.539	0.699	0.334	0.499	0.562	0.718
	Saturation	0.420	0.586	0.475	0.641	0.346	0.511	0.441	0.609
Global-Local model	-	0.495	0.658	0.618	0.763	0.476	0.642	0.612	0.758
	Embed	0.372	0.541	0.451	0.619	0.393	0.559	0.375	0.542
	Brightness	0.495	0.659	0.613	0.759	0.465	0.632	0.604	0.751
	Contrast	0.408	0.577	0.545	0.704	0.414	0.583	0.503	0.666
	Saturation	0.402	0.570	0.491	0.657	0.387	0.555	0.490	0.654

Table 5: IoU-Scores and Dice Coefficients for the four different models with each Global-Local, Global and Local architecture compared for different augmentations on the test images.

4.5.2 Performance on an unseen data set

In a second experiment, the performance is evaluated on an additional, unseen data set. For this purpose, the data set of the Diabetes Foot Ulcer Segmentation Challenge 2021 [23] is used. More details can be found in section 2. The non-augmented images are loaded, resized to 192x192 px and then tested with the already trained models. The performance metrics are displayed in table 6, in comparison with the performance on test data of the original WSNet data set.

		U-Net		LinkNet		PSPNet		FPN	
	Data set	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
Local model	WSNet	0.359	0.523	0.398	0.564	0.373	0.538	0.408	0.574
	DFUC	0.262	0.411	0.203	0.335	0.231	0.372	0.231	0.372
Global model	WSNet	0.504	0.668	0.631	0.772	0.458	0.627	0.632	0.772
	DFUC	0.214	0.350	0.276	0.428	0.155	0.265	0.238	0.380
Global-Local model	WSNet	0.495	0.658	0.618	0.763	0.476	0.642	0.612	0.758
	DFUC	0.201	0.330	0.181	0.304	0.181	0.304	0.215	0.349

Table 6: IoU-Scores and Dice Coefficients for the four different models with each Global-Local, Global and Local architecture compared for the WSNet and the Diabetes Foot Ulcer Segmentation Challenge (DFUC) 2021 data.

- worsens performance by huge amount for every model
- probably training on mix needed but no time in the scope of the project
- means results are not directly transferable to new models

TODO

5 Results and Evaluation

In the following, the results and contribution of this project's work are summarised and further fields of research are named.

5.1 Re-implementation and evaluation of WSNet

The first contribution of this project is the re-implementation of the WSNet framework in a more structured and parametrized manner, such that a reconstruction of results is easier and does not need to make so many assumptions about unknown factors. This includes proper documentation of the code, which was lacking in the original WSNet code. During this, discrepances between the described procedure in the paper and the available code were identified and described.

In a second step, the results were reproduces as far as possible. This showed that the proposed new "state-of-the-art" architecture does not yield significant performance increasments. The global state-of-the-art methods already aim to include localized information and are state-of-the-art segmentation models because of this. Further localisation does create a hidden assumption of the context size of interest in is not neccessarily transferable to different data sets. Interesting further research could include a change of the parameters, e.g., the amount of skip connections of the used models, to improve the results.

5.2 Robustness of wound segmentation

5.3 Explainability of segmentation results

6 Technical Information

6.1 Code and Data Availability

The code produced in the project's scope is available on GitHub: `https://github.com/Zianor/DLIV-semantic-segmentation-of-wounds`. Package versions are included to ensure reproducibility.

The used data is also available on GitHub: `https://github.com/subbareddy248/WSNET/` [19, 20]. Availability at a later point in time cannot be guaranteed.

6.2 Libraries

Several libraries were used in this project. All work is based on the Deep Learning framework TensorFlow with Keras [16, 6]. The implementation of the four used network architectures was provided by the Python library `segmentation_models` [11]. Image augmentations were performed with `Albumentations` [3].

6.3 Used Hardware

All computations are performed on a MacBook Air (24 GB RAM, Apple M2 Chip with an 8-core GPU) or a computer with 16GB RAM and an Nvidia GeForce GTX 1070 Ti as GPU. The package versions for GPU-utilization on MacOS are included in the package versions on GitHub.

6.4 Prior Experience

I have a strong programming background, consisting of a B.Sc. in Computer Science and three years of experience in Web development with Python. Besides the Advanced Concepts of Machine Learning course, I have no experience with Deep Learning.

6.5 Learning Process

During this project I got familiar with TensorFlow and Keras. I learned a lot about the state-of-the-art methods for semantic segmentation and segmentation of wound images in particular and how they can be evaluated. I was not familiar with the local contextualisation of features that is done by using different encoder-decoder architectures and I found it particularly interest combined with the WSNet paper. Originally, I planned on spending more time with explainability of segmentation results but frameworks I found were written mainly for PyTorch. Additionally, the proposed architecture of WSNet sounded better before I knew about the state of the art in segmentation. So I decided to focus on global and local features in semantic segmentation and spend a lot of time researching the background of the model architectures and what their abilities and limitations are.

References

[1] Mohammad Faizal Ahmad Fauzi et al. “Computerized segmentation and measurement of chronic wound images”. In: *Computers in Biology and Medicine* 60 (2015), pp. 74–85. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2015.02.015.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[3] Alexander Buslaev et al. “Albumentations: Fast and Flexible Image Augmentations”. In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125.

[4] Abhishek Chaurasia and Eugenio Culurciello. “LinkNet: Exploiting encoder representations for efficient semantic segmentation”. In: *2017 IEEE Visual Communications and Image Processing (VCIP)*. 2017, pp. 1–4. DOI: 10.1109/VCIP.2017.8305148.

[5] Yeong-Jun Cho. “Weighted Intersection over Union (wIoU): A New Evaluation Metric for Image Segmentation”. In: *ArXiv* abs/2107.09858 (2021).

[6] François Chollet et al. *Keras*. https://keras.io. 2015.

[7] Tom Eelbode et al. “Optimization for Medical Image Segmentation: Theory and Practice When Evaluating With Dice Score or Jaccard Index”. In: *IEEE Transactions on Medical Imaging* 39.11 (2020), pp. 3679–3690. DOI: 10.1109/TMI.2020.3002417.

[8] Yunya Gao et al. “Comparing the robustness of U-Net, LinkNet, and FPN towards label noise for refugee dwelling extraction from satellite imagery”. In: *2022 IEEE Global Humanitarian Technology Conference (GHTC)*. 2022, pp. 88–94. DOI: 10.1109/GHTC55712.2022.9911036.

[9] *How PSPNet works?* URL: https://developers.arcgis.com/python/guide/how-pspnet-works/ (visited on 12/16/2023).

[10] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: 1704.04861 [cs.CV].

[11] Pavel Iakubovskii. *Segmentation Models*. https://github.com/qubvel/segmentation_models. 2019.

[12] Connah Kendrick et al. *Translating Clinical Delineation of Diabetic Foot Ulcers into Machine Interpretable Segmentation*. 2022. arXiv: 2204.11618 [eess.IV].

[13] Alexander Kirillov et al. “Panoptic feature pyramid networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6399–6408.

[14] Tsung-Yi Lin et al. “Feature Pyramid Networks for Object Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[15] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical Image Analysis* 42 (2017), pp. 60–88. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2017.07.005.

[16] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[17] Yujian Mo et al. “Review the state-of-the-art technologies of semantic segmentation based on deep learning”. In: *Neurocomputing* 493 (2022), pp. 626–646. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2022.01.005.

[18] Abderrahim Norelyaqine, Rida Azmi, and Abderrahim Saadane. “Architecture of deep convolutional encoder-decoder networks for building footprint semantic segmentation”. In: *Scientific Programming* 2023, 8552624 (2023). DOI: https://doi.org/10.1155/2023/8552624.

[19] Subba Reddy Oota et al. “HealTech - A System for Predicting Patient Hospitalization Risk and Wound Progression in Old Patients”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2021, pp. 2463–2472.

[20] Subba Reddy Oota et al. "WSNet: Towards an Effective Method for Wound Image Segmentation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2023, pp. 3234–3243.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: (May 2015).

[22] Gaetano Scebba et al. "Detect-and-segment: A deep learning approach to automate wound image segmentation". In: *Informatics in Medicine Unlocked* 29 (2022), p. 100884. ISSN: 2352-9148. DOI: `https://doi.org/10.1016/j.imu.2022.100884`.

[23] Chuanbo Wang et al. *Fully automatic wound segmentation with deep convolutional neural networks*. en. Dec. 2020. DOI: `10.1038/s41598-020-78799-w`.

[24] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6230–6239. DOI: `10.1109/CVPR.2017.660`.