PROJECT REPORT

Segmentation of Chronic Wounds

Cay Rahn
6255648

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

December 16, 2023

1 Introduction

1.1 Motivation

- many people affected by chronic wounds that need to be monitored
- Why is automatic Wound Segmentation so important? And why is it a complex problem?
- Manual segmentation by experts expensive and very time consuming
- experts differ in their segmentation
- different types of wounds have different characteristics
- changing lighting conditions, distance to camera, camera angle, different cameras have impact on result
- controlled environment not feasible in clinical setting
- ideally, we want to be able to take pictures with a smartphone without overly complicated instructions for the person taking the picture
- experience as photographer should not be required, clinical proffessionals should be able to take pictures that are then segmented correctly

1.2 Research Questions

A recent publication by Oota et al. claims to have improved the state of the art in Wound Segmentation. Such claims always needs to be supported by further research. This project aims to investigate and reimplement the proposed method. This includes comparing it to 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 Dataset

2.1 Available Datasets

- not many medical datasets on chronic wounds publicly available [17]
- often focus on specific type of chronic wounds - often diabetic or pressure ulcer
- Example for such a dataset: data from Diabetic Foot Ulcer Challenge 2022 [10] $\rightarrow$ unfortunately only available after application, therefore not appropriate for this project with limited timescope
- other dataset of Foot ulcer wounds is available as part of the Foot Ulcer Segmentation Challenge 2021 [20]

2.2 WSNET Dataset

- used dataset consists of 2686 wound images with their corresponding masks introduced by Oota et al. [17].
- 8 different wound types represented in dataset: venous ulcer, trauma wound, diabetic ulcer, surgical wound, arterial ulcer, cellulitis, pressure ulcer and a not further specified group of other wounds
- unfortunately, the wound classification is not available

3 State of the Art

3.1 Semantic Segmentation

The segmentation of wounds belongs to the class of semantic segmantation 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 became dominant in the last years because they became 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 [13]. This results in an encoder-decoder architecture as base for the networks, inspired by auto-encoders [4], where the encoder subsamples and the decoder upsamples [15]. 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 huge amount of data. A typical data set for such pre-training is the ImageNet object classification data set [2].

In this project, four different segmentation models are used: U-Net, Linknet, FPN and PSPNet. All are improved architectures about a basic fCNN. Each architecture is described in detail to understand challenges and approaches of localizing 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 path in the original paper, describing their function. They are also described as context path and spatial path [14]. Both, 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 encdoer path and convolution is applied again. In the final layer, 1x1 convolution is used to map the feature vector to the desired number of classes. This architecture is visualised in figure 1. [18]

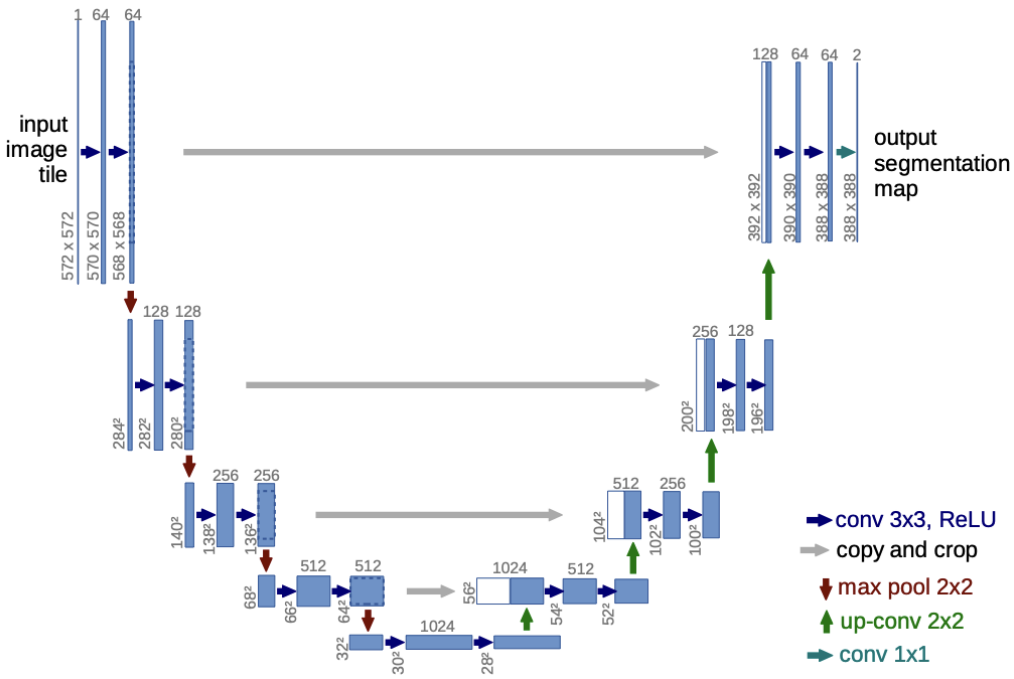

Figure 1: U-Net architecture for 32x32 pixels in the lowes resolution. Blue boxes are feature maps with the number of feature-channels on top of the boxes and the size shown on the left size. Operations are indicated by the arrows. The skip connections are a concatenation. The figure originally create by Ronneberger et al. [18].

The skip connections, connecting the different levels of encoder and decoder prevent a loss of information and extracting the features at different resolutions to retrieve spatial information. By doing this, it is one of the first architectures improving the classical fCNN for semantic segmantation [13]. While U-Net provides spatial localization of features, its ability to generalize to multi-scale information is limited [15].

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

Linknet Similar to U-Net, Linknet contains of an encoder block for downsampling and a decoder block for upsampling. The downsampling is not done by max pooling as it is in the U-Net architecture but by using a stride of 2 in a convolutional layer. Also does the inital encoder block differ from the following blocks as it uses a larger kernel and uses 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 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]

The implementation used in this project has four skip connections instead of the original three [9]. Similarly to the U-Net, the input size is restricted such that every upsampling operations need to be

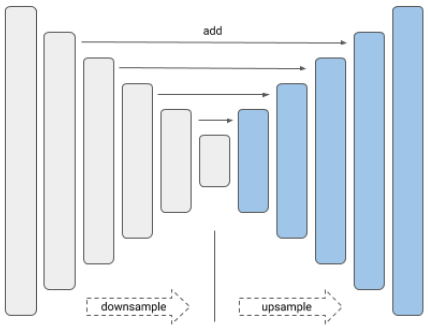

Figure 2: A visualisation of the LinkNet architecture originally provided by Iakubovskii [9].

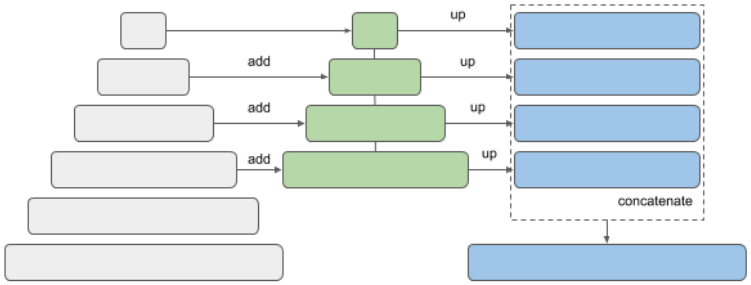

Figure 3: A visualisation of the FPN architecture originally provided by Iakubovskii [9]. Note, that the feature maps can combined either by concatenation or addition.

applied to an even x and y size.

Linknet has been shown to achieve better results than U-Net under similar conditions [7].

FPN The Feature Pyramid Network (FPN) architecture creates feature maps of various sizes in multiple layers [15]. Similar to the other architectures in consists of an encoder and a decoder, called bottom-up and top-down pathway here[12]. Similarly to U-Net, feature maps at different scales with a scaling step of 2 are created in the encoder [12]. 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 an 1x1 convolution is applied. By doing this so-called feature pyramids are build, containing features at different resolution. Kirillov et al. proposed a method to use these feature pyramids to obtain a segmentation by merging the feature maps using addition or concatenation [11, 9]. The architecture is visualised in figure 3.

PSPNet

- long for Pyramid Scene Parsing Network
- feature map extracted with pretrained backbone
- pyramid pooling to get context information
- pyramid pooling: fusion of features under four different pyramid scales (global pooling and sub-regions for different locations), 1x1 convolution to maintain weight of global feature after each pyramid, upsampling of output to get same size as original feature map
- "different levels of features concatenated as final pyramid pooling feature"
- final prediction by convolution layer which input isoriginal feature map concatenated with pyramid pooling output
- motivation: pyramid pooling provides levels of information, more helpful than global pooling

3.1.1 Evaluation

There exist several methods 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 that are commonly used in semantic segmentation in medical imaging are the Dice Coefficient and the Intersetcion over Union (IoU) score. They indicate the segmentation quality better than pixel-wise accuracy [6].

IOUScore 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. By this it compares the similarity of the two masks [5].

$$\text{IoU}(y, \tilde{y} := \frac{\text{Area of overlap}}{\text{Area of union}} \tag{1}$$

$$= \frac{|y \cap \tilde{y}|}{|y \cup \tilde{y}|} \tag{2}$$

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

$$\text{Dice}(y, \tilde{y}) := 2 \cdot \frac{\text{Area of overlap}}{\text{Total area}} \tag{3}$$

$$= 2 \cdot \frac{|y \cap \tilde{y}|}{|y| + |\tilde{y}|} \tag{4}$$

To gain more insight into the type of the errors the model makes, the rate of false positives and false negatives can be used do differentiate Type I and Type II errors [10].

3.1.2 Loss function

- loss function often uses pixel-wise (weighted) cross-entropy loss even though differentiable approximations of the two metrics exist [6]

3.1.3 Data Augmentation

- augmentation useful especially if available data is limited
- makes it more robust and accurate
- augmentations can be divided in two categories: position augmentation and color augmentation
- what augmentations are appropriate for the application

3.2 Wound Segmentation

- one type: diabetic foot ulcers → are monitored to ensure healing process is optimal and there is no infection, normally long time span [10]

- wounds have complex structure containing different types of tissue with different colour and texture → different regions with borders in between [1]
- heterogeneous wound images

- before deep networks: features describing color and texture, algorithms such as region growing and optimal thresholding or classical machine learning models, e.g. Support Vector Machines [19]
- Convolutional Neural Networks then used, manually extracted features replaced by the ones the CNN learns autonomously [19]
- some methods include pre-processing steps to remove background (User interaction, manual feature engineering to detect background pixels, standardizing background in advance before taking feature) → not automatic
- Diabetic Foot Ulcer Challenge 2022 used FCN, U-Net and SegNet with different backbones as baseline for their challenge (categorical cross-entropy loss) [10]
- generally often classical models used with minor adaptions
- following standing out
- Scebba et al. proposed two step method: object detector that produces bounding boxes containing the wounds and then segmentation on those areas (u-net, convNet, DeepLapV3 with ResNet-101 backbone and FCN with VGG16 backbone, pixel-wise weighted binary cross entropy loss, weighting term was computed as the ratio between the total number of wound bed and background pixels of each training set fold)

- Oota et al. claim they set a new state of the art, their method will described in more detail in the following

3.2.1 WSNET

- based on the four before described segmentation architectures: U-Net, LinkNet, PSPNet and FPN
- experimented with different backbones, in the scope of this project MobileNet [8] is used since it is the smallest one and allows faster training
- all backbones with ImageNet pre-trained weights
- they performed Wound-Domain Adaptive Pretraining by classifying the wound images in 5 ulcer types
- data augmentation on the training data and corresponding masks, not on test data
- augmentation consists of horizontal flip, random rotation, optical distortion, grid distortion, blur, random brightness contrast, and transpose

Global-Local Architecture

- motivation: obtain global signals from entire image and local signals from smaller patches for details
- only local might cause incomplete segmentation for large wounds
- local architecture: split image in 16 non-overlapping patches (48x48x3), stacking results in 48x48x(3x16) volume
- parallel 16 local models with shared weights
- combined to full-size mask at end
- stack output of global and local model to output of size (192x192x2)
- 1x1 convolution to get final mask

- interesting that they use segmentation models that already use methods to localize method, e.g. FPN already considers different context sizes
- chosen patch size implies some property of the wound images, which size is important for local information
- they stated in their paper that they tested different patch sizes and chose 48 because it lead to the best results

Reported Results

- pretraining on wound images improves results
- data augmentation leads to improvements
- local only models significantly worse than global model
- global only models worse than global-local model

4 WSNET

4.1 Code availability and reproduction of the results

Although the code for WSNET [17] 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 documentations makes it hard to make use of the code, especially since the code seems to contain some errors.

In the scope of this project, the code was used to create runable models again. Unfortunately, the classes of the wounds are not available, which makes it impossible to perform pre-training as it was described in the original paper [17]. In total there are eight models available: A local model and a combined global-local model for each of the segmentation models Unet, PSPNet, FPN, and Linknet. The Python library used for the segmentation models is `segmentation_models` [9]. The implementation processed showed some differences to 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 beside PSPNet, only allow input sizes that are divisable by 32 and the GitHub showed a size of 64 px was used. Another difference between available code and the paper is, that it is claimed that augmentation is not performed on the test images which is not the case in the available code.

Information about the size of training, validation and test set is not given in the paper or code. In the scope of this project, a split of 70 % training, 15 % validation and 15 % test data is used.

- train + validation set were just first x % of dataset, not shuffled as one would normally do it

Because the data training for the wound-specific pre-training is not available, the results can only be compared for imagenet pre-training.

- problem with imagenet pretraining: input size for patches is not available, instead the default size of 224 is used, which might impact results negatively

4.2 Comparison of the achieved performance

	Unet		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 1: IoUe-Scores and Dice Coefficients for the four different models with each Global-Local, Global and Local architecture. The backbone used is mobilenet.

- results are comparable with results reported in paper, slightly lower scores
- e.g. Unet IoU score 0.495 with my code, 0.620 in paper (dice score 0.761 vs 0.658)
- others are closer (linknet 0.618 from me vs 0.621 in paper, dice 0.763 in paper and for me)

		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

Figure 4: Results reported by Oota et al. [17].

- maybe differences in training size
- most important thing: the global-local model does not improve about global model
- TODO: explain here or later??
- the results reported by Oota et al. are shown in figure

5 Results and Evaluation

5.1 Re-implementation and evaluation of WSNET

- contribution: reimplementation of a framework in a better documented way, making the reconstruction of results easier
- documentation of this implementation
- identified discrepances between paper and available code
- comparison of results
- showed that reported "good" architecture does not yield significant performance increasements
- why: models already capture localized information, this is why those are sota segmenatation frameworks, further localization does not really make sense, maybe rather change parameters of used models to improve results
- Which architecture is suited best for wound segmentation? Why and what other alternatives would there be?

5.2 Robustness of wound segmentation

- augmentations commonly performed to improve robustness of models
- can also be used to access robustness of the resulting model
- for clinical application, lightning and size might vary between images
- batch normalization might further increase this problem (TODO: search references)
-

5.3 Explainability of segmentation results

6 Technical Information

6.1 Prior Experience

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

6.2 Code and Data Availability

The code produced in the scope of the project is available on GitHub: `https://github.com/Zianor/DLIV-chronic-wound-segmentation`. Package versions are included to ensure reproducability.

The used data is available on GitHub as well: `https://github.com/subbareddy248/WSNET/` [16, 17]. Availability on a later point of time cannot be guaranteed.

6.3 Libraries

- Tensorflow
- `segmentation_models` [9] providing the implementations for Link-Net, U-Net, PSPNet and FPN
- image augmentations performed with `Albumentations` [3]

6.4 Learning Process

- Getting familiar with tensorflow
- learning about the state of the art in segmantic segmentation and segmentation of wound images and evaluation methods
- more experience in dealing with paper results and how trustworthy they are

6.5 Used Hardware

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

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] 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.

[7] 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.

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

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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