UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Medical image segmentation using shape prior information and deep neural networks

Author:
Bohdan PETRYSHAK

Supervisor:
Dr. Jan KYBIC

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2019

Declaration of Authorship

I, Bohdan PETRYSHAK, declare that this thesis titled, "Medical image segmentation using shape prior information and deep neural networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"“What all of us have to do is to make sure we are using AI in a way that is for the benefit of humanity, not to the detriment of humanity.”"

Tim Cook

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Medical image segmentation using shape prior information and deep neural networks

by Bohdan PETRYSHAK

Abstract

Semantic image segmentation is the task of classifying each pixel of an image into a corresponding category of what is being represented. It is an essential step towards automating image analysis process. However, the low-quality signal, high level of noise, variety of objects appearance, little amount of labeled data are the critical obstacles which stand on the way of achieving the perfect segmentation results. Incorporating the shape prior knowledge has proven significant improvement of the segmentation results. In this work, we extend the existing method of incorporating shape priors within the FCN segmentation framework to a multiclass semantic segmentation. We demonstrate the superiority of our extension in five different datasets and show that it capable of making the segmentation results more accurate and plausible in multiclass problems. ...

Acknowledgements

First of all, I would like to thank my supervisor Jan Kybic(Czech technical university in Prague) who supervised my work under the thesis and provided tons of valuable research ideas, scientific papers and possible future ideas for a current project. I would like also to thank Czech technical university for provided scholarship and ability to work in the CMP lab. I am also grateful to my Mom, Father, brother and my girlfriend for solid support and constant motivation to finish the work. Thank you Oles Dobosevych and Oleg Farenyuk for your patience and answering for my stupid questions. Finally, I want to thank Ukrainian Catholic University and Faculty of Applied Science for unforgettable Bachelor's Program in Computer Science and giving me a solid knowledge background to be able to save the world.

Contents

List of Figures

List of Tables

List of Abbreviations

DL	**D**eep **L**earning
CNN	**C**onvolutional **N**eural **N**etwork
FCN	**F**ully **C**convolutional **N**network
US	**U**ltra**s**ound
SoTa	**S**tate **o**f **T**he **A**rt
AG	**A**ttention **G**ate
CRFs	**C**onditional **R**andom **F**ields
RRCU	**R**ecurrent **R**esidual **C**onvolutional **U**ints
GPU	**G**raphics **P**rocessing **U**nit

Chapter 1

Introduction

Semantic segmentation is the task of predicting the category of individual pixels in the image which has been one of the key problems in the field of image understanding and computer vision for a long time. It has a vast range of applications such as autonomous driving [22, 20, 15](detecting road signs, pedestrians and other road users), land use and land cover classification [6, 58], image search engines [65], medical field [5, 3, 21](detecting and localizing the surgical instruments, describing the brain tumors, identifying organs in different image modalities). This problem has been tackled by a combination of machine learning and computer vision, approaches in the past [16, 40, 4]. Despite their popularity and success, deep learning era changed main trends. Many of the problems in computer vision - semantic segmentation among them - have been solved with convolutional neural networks (CNNs) [55, 39, 9].

Segmenting the images with low-quality and low signal to noise ratio remains problematic even for powerful classifiers like CNNs. It has been shown, that incorporation of shape prior information significantly improves the performance of the segmentation algorithms [48, 8, 47]. However, in segmentation techniques like CNNs, it is a very tricky question of how to incorporate such prior knowledge.

In this work, we would like to introduce the extended to a multi-class method [53] of integrating a prior shape knowledge into the training process of CNN as a regularization loss term.

1.1 Task Definition

We want to concentrate this thesis in the three main applications(see in more details in chapter 3):

1. Segmentation of B-mode ultrasound(US) images of the carotid artery.
2. Segmentation of histology of the carotid artery.
3. Segmentation of the laser trace in ablative imprints.

1.2 The proposed method

Taking into account all mentioned before we propose the following:

- Exploit novel CNNs architectures for solving the problem of semantic segmentation.
- Utilize the Fully Convolutional Network(FCN) autoencoder-decoder like architecture to extract the hierarchical representation of shapes in the training set and introduce the extracted shape prior information about the target objects in terms of the loss function.

1.3 Contributions

- To our knowledge, this is the first usage shape prior [53] approach for the multiclass task.
- We explored some additional and more effective strategies of pretraining the FCN network for extracting shape information.
- We investigated the most beneficial augmentation strategies for the US images of the carotid artery.
- We extended the idea [53] and implemented in Pytorch framework and experimentally evaluated its performance on our datasets while optimizing the parameter values and architectural details which were not explained sufficiently in the original article.

1.4 Structure of the thesis

Chapter 2. Literature review

This chapter contains a detailed overview of the research around image semantic segmentation, incorporating prior shape information for segmentation algorithms, optimized segmentation neural networks.

Chapter 3. Applications

In this chapter, we describe our datasets which we used for benchmarking our segmentation methods.

Chapter 4. Methodology

We describe our methods which we used for incorporating shape priors.

Chapter 5. Experiments and Results

We described experimental settings and the results of our experiments.

Chapter 6. Conclusions

We summarize the scope of the work in this thesis and achieved results. The possible future ways of the thesis highlighted.

titlesec

Chapter 2

Literature review

Before the arrival of deep networks, the most effective methods primarily relied on hand-crafted features classifying pixels independently. Typically patch of pixels is fed into a classifier, e.g. SVM [67], Random Forest [57] or Boosting [62] for predicting the category of the central pixel. After the breakthrough by Krizhevsky and Hinton [36] larger and deeper networks have been trained. CNNs showed remarkable results in the classification tasks (i.e., each image has a corresponding label or a couple of labels).

2.1 Recent progress in semantic segmentation

Patch-wise approaches

First trials to utilize neural networks in semantic segmentation were in an old fashion manner, described above. Hand-crafted features and classifiers were substituted by CNN, which extracted features from the patch of pixels by itself and made a classification. One of the most successful papers on this topic was published by Ciresan et al. [13]. He trained a network in a sliding-window manner to predict a class for each pixel by providing a region of pixels(patch) around that pixel as an input. This approach offers two main benefits. Firstly, we can train the network from very few annotated examples as thousands of patches can be extracted per image. Secondly, our network has a high localization accuracy. Thus, the strategy of Ciresan won the EM segmentation challenge at ISBI 2012 by a large margin.

Despite high predictive power, this approach has strong limitations. First of all, this strategy is quite slow because the network must make inference for each patch independently. The network also does lots of redundant computations because of patch overlapping. This method also puts restrictions on receptive fields, as they bounded on the size of the patch.

Fully convolutional network(FCN)

Currently, the most SoTA algorithms for semantic segmentation stem from a common predecessor: the Fully Convolutional Network(FCN) by Lang et al. [39]. The idea was to take advantage of existing CCNs as a powerful hierarchical feature extractor. They adapted and extended the well-known classification models - Alex Net [36], VGG(16-layer net) [60] and GoogLeNet [63] - into fully convolutional ones by replacing last fully connected layers by convolutional units to produce spatial maps instead of classification scores(see figure 2.1). The maps are upsampled by two different techniques - using bilinear interpolation or fractionally strided convolutions(so-called deconvolution [71]) to produce final dense prediction pixel-wise. This method is the first work trained FCN end-to-end for pixel-wise prediction and from supervised pretraining. Comparing with the approaches described in previous paragraph

2.1 both train and inference are performed whole-image-at-a-time with inputs of arbitrary size. It reduced computational redundancy and noticeably decreased training and inference time. This algorithm outperformed previous methods in segmentation accuracy on standart benchmark datasets like PASCAL VOC, NYUDv2, SIFT Flow, and PASCAL-Context. For all those reasons and other meaningful contributions, the FCN served the cornerstone for the most modifications of deep learning algorithms in semantic segmentation in next years.

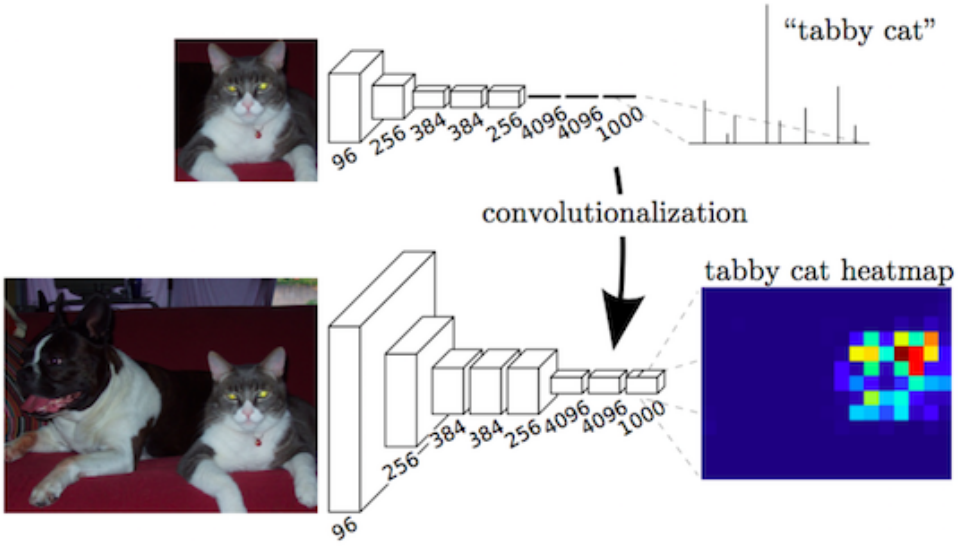

FIGURE 2.1: Transformation of the fully connected layers into convolution layers. *Source:* [39]

Regardless of the power and flexibility of the FCN it still has a range of significant drawbacks. The first problem is signal downsampling. It relates to the decreasing features resolution caused by the repeated combination of max-pooling and downsampling('striding') performed at every layer. The main question here is how to upsample compressed features to the size of the input image and preserve fine-grained details. In the one hand, fine-grained or local information is crucial to achieve good pixel accuracy. On the other hand, it is also important to integrate the information from the global context for solving the local ambiguities. The simple CNNs(which have been crafted for classification) suffer from this balance. Pooling layers, which also allow the network to gain some degree of spatial invariance and lower the computation load, discard the global context.

In the next subsections, we will survey SoTA ideas, which have been presented in recent years to cope with the described challenges.

Encoder-decoder architectures. U-Net family

In the FCN the output is obtained by a high ratio(32x, 16x, and 8x) upsampling which might cause coarse final segmentation output. The compelling idea, which refers to this problem has been utilized in a large number of papers proposed by Hyeonwoo Noh et al. [46]. Instead of high ratio upsampling they suggested to use the entire deconvolution network, which consists of deconvolution, unpooling and rectified linear unit(ReLU) layers (illustrated in figure 2.2).

The deconvolution and unpooling play different roles for constructing the final prediction. Unpooling grabs sample specific structures by tracing the original locations with strong activations back to the image space. Respectively, it effectively reconstructs the detailed structure of the object in finer resolutions. At the same time, trained parameters of the deconvolution tend to catch class-specific shapes.

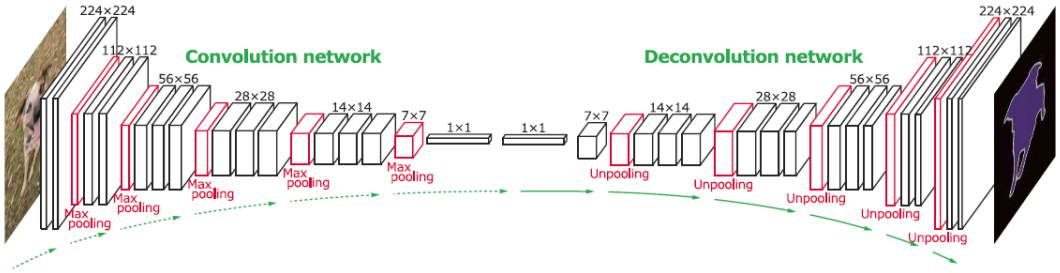

FIGURE 2.2: Overall architecture of the Deconvolution Network. *Source:* [46]

O. Ronneberger et al. [55] have extended the idea of FCN and Deconvolutional network for biological microscopy images. They created the end-to-end architecture called U-Net (illustrated in the figure 2.3). It consists of two parts: a contracting path to capture the context and a symmetric expanding path that enables precise localization. Moreover, they have added skip connections in the network to combine high-level feature map representations with more specific and dense ones at the top of the network. The number of parameters is relatively slow, and it can be trained in a small labeled dataset(with appropriate data augmentation). For example, the authors used a publicly available dataset with 30 images for training during their experiments. The U-net architecture was so successful that it became a generic deep-learning solution for frequently occurring quantification tasks such as cell segmentation, morphology estimation and texture types delineation in biomedical image data [18].

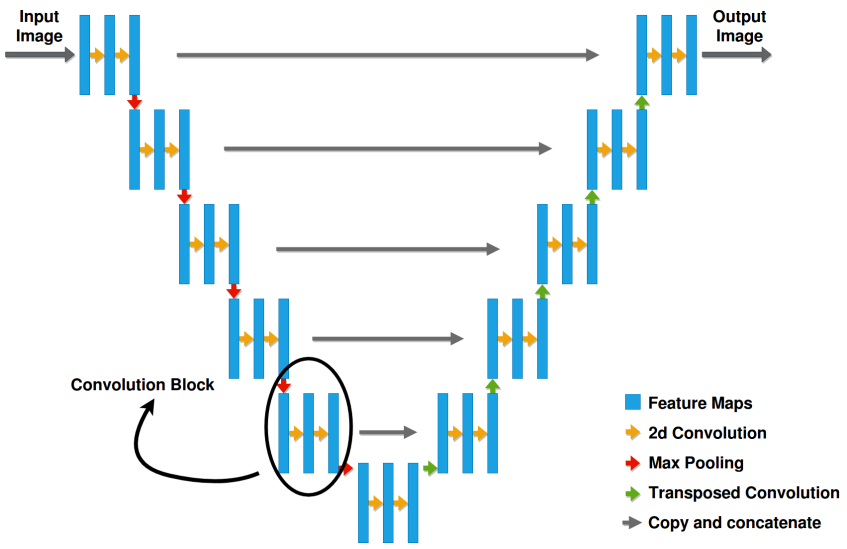

FIGURE 2.3: U-Net architecture

Drozdzal et al. [17] substituted the classical stacked convolution blocks(see figure 2.3) in favor of residual blocks, which introduced short skip connections(inside a block) alongside with long skip connections (corresponding feature maps between the encoder and decoder parts) found in the standart U-Net architecture. As the authors reported, it caused faster convergence and allowed to train deeper models.

Jegou et al. came up with an idea to use dense blocks still preserving the U-Net architecture [31]. They used the characteristics of DenseNet, i.e. carrying low-level features from previous layers alongside higher level features from more recent layers. This property contributes to highly efficient feature reuse.

As we can notice, the modifications of encoder-decoder U-Net like architecture were tightly connected with the progress in classification networks, which researches

usually use as the main backbone(encoder part) for the whole segmentation framework. We also would like to mention the following modifications(the most productive ones) in a nutshell. RU-Net(Residual) and R2U-Net(Recurrent Residual) were proposed by Md Zahangir Alom et al. [2]. Authors propose to merge the power of U-Net and recurrent and residual networks. Another worth mentioning work is written by Ozan Oktay et al. [49]. The central idea is to use novel attention gate(AG), which automatically learns to focus on a target object's morphology and structure. It helps to ignore inappropriate regions in the input images and emphasize the regions which are useful for the specific task. TernausNetV1 and TernausNetV2 [28, 29] used pre-trained on ImageNet encoder(VGG16 and ResNet variations). The pre-trained encoder shows better performance than trained from scratch especially on the small datasets.

Context aggregation methods

As we pointed out in subsection 2.1, FCN hardly segments objects at multiple scales and poorly incorporate the global information. Many approaches were taken to make CNNs aware of that global context information: postprocessing step with Conditional Random Fields(CRFs), dilated convolutions, different pyramids strategies, multiscale and multilevel feature fusion.

CRFs

One possible and common approach to refine the output of the model and enforce its capacity to capture fine-grained details is to use CRFs as a post-processing step [11]. CRFs enable the combination of low-level information - such as pixels interaction with the final output of the segmentation model, i.e. per pixel class scores. It helps to catch long-range dependencies, which CNNs usually miss. Among successful usages of CRFs are the DeepLab works [9, 10]. By using the fully connected pairwise CRF, they overcome the loss of information due to the spatial invariance of CNNs. Another significant work applying CRFs with FCNs is the CRF-RNN by Zheng et al. [75]. The authors reformulated dense CRFs via a recurrent neural network so, that they made it possible to fully integrate the CRF with FCN and the train the whole network in end-to-end.

Dilated convolution

Dilated convolution(so-called, atrous or hole convolution) is the generalization of Kronecker-factored convolutional layers. The dilation rate K controls the upsampling factor. Stacking K-dilated convolution increase the receptive field exponentially meantime the number of parameters grows linearly. The essential work which used dilated convolution was proposed by Yu and Koltun [70]. They have proposed a module for aggregation the multi-scale contextual information systematically with the help of dilated convolutions. The model is based on the dilated convolutions which exponentially increase the receptive field without the loss of resolution or coverage. The aforementioned DeepLab models apply this strategy in their works as well. Dilated convolution has gridding artifacts. Noteworthy works, which tackled with this problem are smoothing dilated convolutions [68] and hybrid dilated convolution(HDC) by Panqu Wang and Pengfei Chen [66].

Pyramid and multi-scale methods

Another possible way to cope with global context integration is multi-scale prediction and pyramid strategies.

The Full-Resolution Residual Networks(FRRN) [52] is the bright example of multi-scale processing technique. The algorithms consist of two separated streams: the residual stream and pooling stream. The first one processes the features in the full resolution. The second one processes and downsamples features via pooling operation. It enables the combination of high-level and low-level semantic information. The authors did not train these two streams disconnected. After each pooling operation, FRRN does the feature combination, to combine the information from two streams. The framework was trained end-to-end. The main drawback is that the preprocessing in full resolution is highly computationally expensive. The RefineNet [38] showed that it is not necessary to process features in the original resolution. When we pass through the feature extraction network, we naturally get the multi-scale feature maps after each downsampling. The authors feed the image at different scales to the RefineBlocks independently in the bottom-up fashion. After each upsampling, they merge the features from the current block with the previous. Thus multi-scale information is obtained and incorporated during the training.

Atrous Spatial PyramidPooling (ASPP) [9] allows to segment objects robustly at multiple scales. It uses dilated convolutions with different dilation rates in parallel. This pyramid strategy effectively helps to capture context at multiple scales. PSPNet [74] is a known strategy of pyramid approach as well. They use a very similar approach as ASPP, but the max pooling operation is used instead of dilated convolutions for context aggregation.

2.2 Real-time semantic segmentation

As we saw in the previous section, most of the works focused on improving the accuracy and robustness of the segmentation without taking much consideration of efficiency. Recent efforts for building light networks can be roughly classified into two groups. The first group proposes to use some light classification network architectures, i.e. ShuffleNet [72, 41], ResNet18 [23], MobileNet [26] as an encoder part and optimized strategies for upsampling [59]. The second one proposes to design the optimized network from scratch. ENet [51] was a very notable example of such an approach. ICNet [73] introduce cascade feature fusion unit to produce the segmentation faster and better quality. Most SOTa efficient networks use depth-wise separable convolutions [26]. This approach factor the convolution into depth-wise and point-wise convolution steps to reduce the complexity. Another common approach is group convolution [37, 72, 26], where input channels and convolutional kernels decomposed into groups. Each group convolved independently. ESPNet and ESPNetv2 [43, 44] employ these tricks to improve both accuracy and efficiency. BiSeNet [69] presented the Spatial Path and Context Path. The first flow preserves spatial information and generates high-resolution features. The second one quickly downsamples the information to increase the receptive field. Also, the researches come up with a new Feature Fusion Module to combine the features effectively.

2.3 Previous works in utilizing shape prior information in segmentation models

Energy-based approaches

In classical energy-based approaches, the shape information can mainly be modeled in three ways: template-based(geometrical), physical and statistical [47]. In some cases, we know the shape of the target object beforehand(e.g. circle, square, human-like shape). The first method suggests to descripe some distance function between the predicted object and the prior shape and penalize any deviation from the given

shape[56, 12]. In lots of real-world problems, intra-class objects vary in their forms a lot. For instance, in medical images, the shapes of organs change from one patient to another or even over time. The assumption about constant shape may be inappropriate. A common way to capture the variation of shapes of one class is to use the probability model and represent the object explicitly(e.g., point cloud), implicitly, boundary-based [14]. The last group of methods suggests modeling the physical qualities of the object and utilizing this prior information in the segmentation framework [32].

Encoding prior knowledge into deep learning algorithms

Fei Chen et al. [8] proposed to incorporate shape prior information in two stages. The first stage uses deep Boltzmann machine to learn the hierarchical structure of shapes in the training set. In the second state, the learned global and local shape variations utilized in an energetic form to data-driven variational methods for making the final predictions. In addition to the training data Mohammad Tofighi et al. [64] used a set of canonical shapes obtained via domain experts. They utilize the expected behavior via regularization term, which penalizes false positives results, which are not inside the boundary of the prior shape object. Zahra Mirikharaji et al. [45] came up with a new loss term for incorporating shape priors into an end-to-end FCN. They penalized all non-star shape segments in the output of FCN as in their task all the training examples have the property of star shape. In comparison with energy-based approaches, this method does require expensive optimization steps at inference time nor user input information about the object center. However, it is a highly task-specific solution.

In [48, 53], authors propose novel cost functions in which they try to minimize distance not only to the ground truth but also to the learned shape. They used convolutional autoencoders for projection the input image into a shape space. These approaches propose to utilize the prior shape information fully automatically, without user interaction and additional computational costs during the inference time.

Chapter 3

Applications

3.1 Syntetic data

Semantic segmentation task is often constrained by the availability of the labeled data. To estimate our algorithms on enough representative and high-quality data, we generated big-sized synthetic datasets.

3.1.1 Binary segmentation dataset

Each image in our synthetic data contains one of the four basic corrupted randomly located shapes(circle, rectangle, triangle, and star). There are two categories: background and shape. We used the following algorithm to produce each image (see on the Fig.3.1):

1. Generate random-sized shape, chosen from the set of four possible shapes.
2. Add some salt-and-pepper noise.
3. Use erosion with round kernel element to corrupt the shapes more intensively and naturally.
4. Apply transpose, shift and rotate augmentation.
5. Add Gaussian noise and median blur.

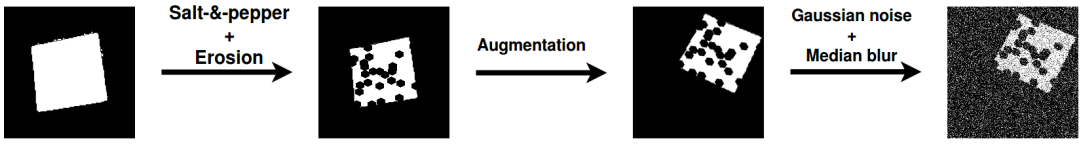

FIGURE 3.1: The corruption pipeline

We generated three different versions of the dataset: with a low level of noise(low level of salt-&pepper noise, without Gaussian noise and median blur), medium(medium level of salt-&pepper noise, low gaussian noise and median blur), high(high level of saltpepper and Gaussian noise and median blur). The examples from each dataset illustrated in Fig.3.2. Each dataset contains 1200 images.

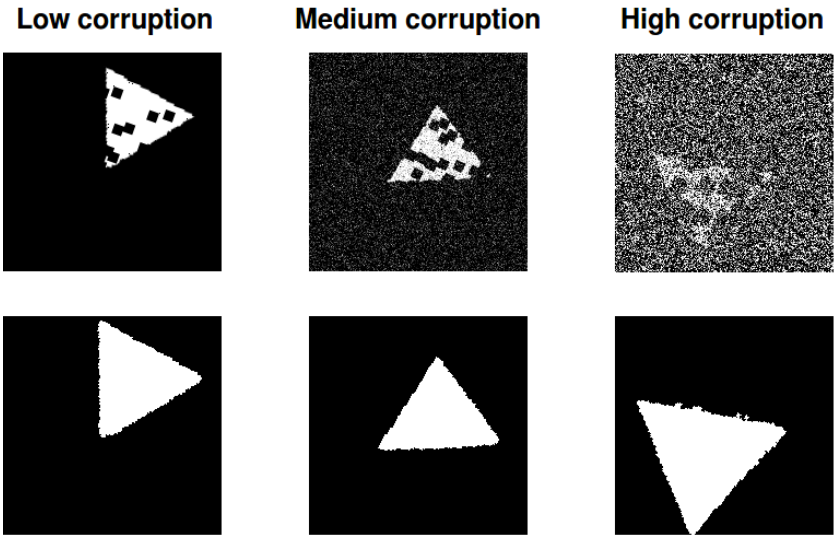

FIGURE 3.2: Datasets with different variations of noise intensity. In each column you can see the input image and corresponding ground truth mask.

3.1.2 Multiclass segmentation dataset

The segmentation task of the human carotid artery can be described with two categories: lumen(space, through which blood flows inside the artery) and an artery wall which surrounds the lumen. We simulated the location dependency of the classes in problem, described in 3.2. We generated two ellipses of random size, one inside another. The inner oval is the imitation of the lumen, and the outer one is the artificial wall of the vessel. We also added some noise(Gaussian noise, median blur) and corruption(as in second and third steps in 3.1.1) to the generated objects. To increase the task complexity, we used textures samples from the Brodatz collection [1]. We took 56 unique textures(22 for outer oval, 22 for inner oval and 12 for the background). Accordingly, each image contains three unique textures picked randomly from the predefined subsets. We also applied augmentation techniques(shift, scale, rotate and elastic transform) to introduce more variety of shapes. In Figure 3.3 has shown the whole pipeline of image generation.

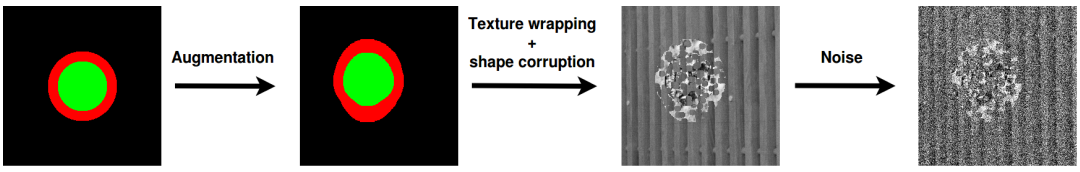

FIGURE 3.3: Visualization of the sample generation

We created two versions of the dataset: with a low level of noise(low level of object corruption, without blur and Gaussian noise) and high level of noise(high level of object corruption and with a high level of Gaussian noise and median blur), as illustrated in Fig.3.4

3.2 Carotid artery segmentation

Stroke is one of the leading cause of illness, mortality and long-term disability around the world [1]. Atherosclerosis is the most common cause of ischemic stroke. It occurs due to the build-up of plaque on the inside of the arteries which cause narrowing

[1] Avaliable in the digital form, in http://sipi.usc.edu/database

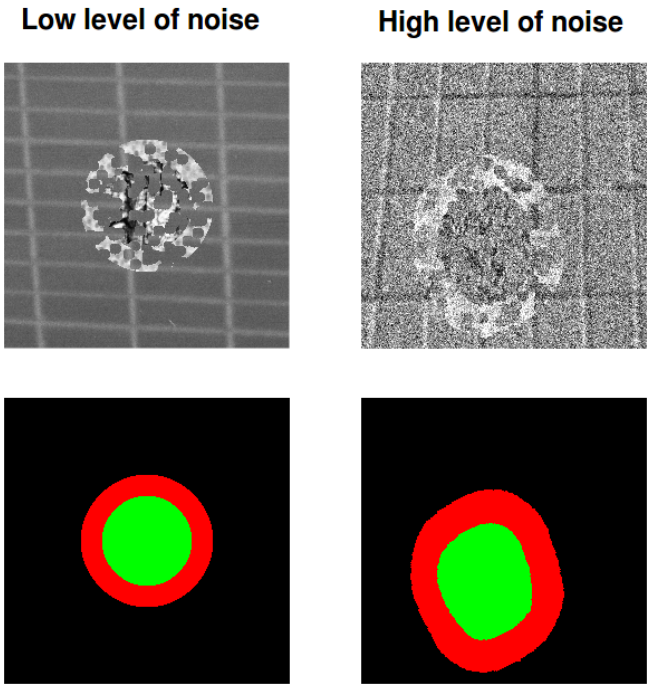

FIGURE 3.4: Datasets with different variations of noise intensity. In each column you can see the input image and corresponding ground truth mask

of the lumen of the artery(stenosis). Over time, stenosis advance and may increase the risk of thrombosis, or/and artery-to-artery embolism in the brain resulting in the stroke in many patients. The results of the studies [19] showed that the higher risk of ischemic stroke associated with the special types of the atherosclerotic plaque in the carotid artery. Recently, many studies with different diagnostic methods have been performed to identify the high risk of atherosclerotic plaques in the carotids [61, 7, 50]. Duplex sonography is one of the most available of these methods. The evaluation of plaque characteristics is mainly only visual, which time demanding and dependent on the sonographer experience. We tried to apply our segmentation model, trained with shape priors to make the process of localization and description of the carotid artery fully-automated and more precise than manual routine.

We obtained the images of B-mode ultrasound of in vitro examination of the carotid artery and digitized histological slices of carotid artery from more than one hundred and fifty patients. The data previously acquired from the Internal Grant Agency of the Ministry of Health of the Czech Republic and a multicenter prospective study – Neurological Ultrasound Study.

3.2.1 Ultrasound of the carotid artery

We labeled 35 images with four classes: lumen(space, where the blood flows), the wall of the artery, artifacts and the background. Our manual segmentations were checked and approved by sonography experts. We used the ImageJ [2] image analysis software for producing the ground truth masks to the ultrasound images. The program provides the ability to write own plugins and macros to adopt for if for personal needs. We wrote the plugin, which supports the multiclass labeling of the images. The raw pictures have contained lots of the metadata, depicted on them. These regions are useless for our algorithms. We cropped the regions only with the relevant information for our algorithms and used zero padding for fitting the size

[2]See the official documentation https://imagej.net/ImageJ

of the images for our algorithms(see the Fig. 3.5). We did not use any resizing techniques as we wanted to keep the original aspect ratio. The examples from the dataset can be found in Fig. 3.7.

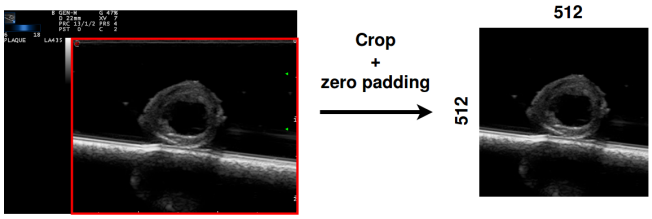

FIGURE 3.5: Preprocessing of the ultrasound input images.

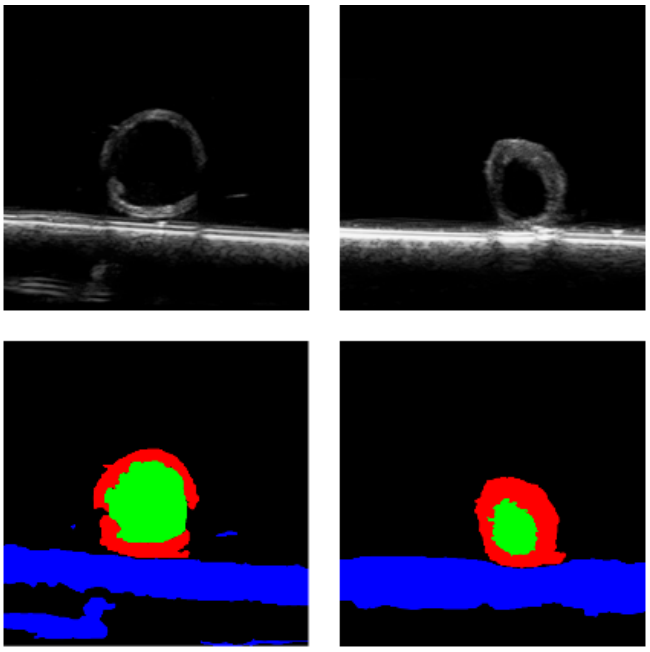

FIGURE 3.6: Ultrasound dataset examples. The first row is the input images and the second one is the corresponding ground truth. Each ground truth contain 4 classes: green is the lumen of the artery, red is the wall of the artery, black - is the background and blue - is the artifacts.

3.2.2 Histology of the carotid artery

The clinician experts marked 166 images of histology of the carotid artery. There are two types of histology images in our data with different stainings: light and dark(see in Fig. 3.7). Each slice of the carotid artery is represented with two types of staining. Correspondingly, we have 83 images of each type. The dark type of images looks more feasible for semantic segmentation task as it contains much more color, texture and topology details about different classes. This dataset contains 12 classes. The color encoding is depicted in Fig.3.8.

The quality of the annotations was very poor. The ground truth examples were provided in the format of unclosed contours, which is unsuitable for semantic segmentation. We developed an application in OpenCV[3] for completing and filling the contours. There also were lots of pieces of the tissue in the labeled images, which were not marked by any class. We trained a binary classifier for identifying tissue.

[3]See the official documentation https://opencv.org/

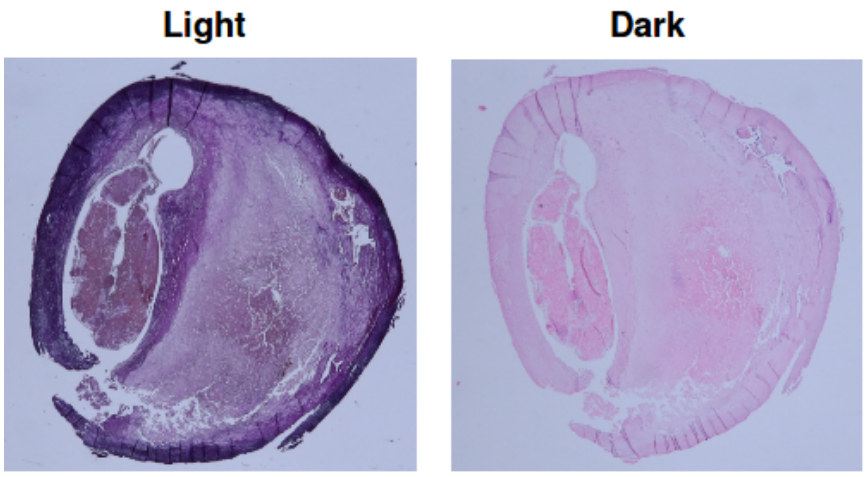

FIGURE 3.7: Two different versions of the histology staining.

We used an interactive image segmentation tool ilastik[4], where it is possible to mark images and retrain classifiers interactively. We added two more classes called "Unknown" and "Unknown inside the lumen" to the ground truth labels. By "Unknown" we marked all tissue pieces which did not belong to any class before but was identified by our binary classifier. By "Unknown inside the lumen" we marked all tissue pieces, which belonged to the lumen or were not marked by any class. The illustration of the final data samples is in the Fig. 3.9

The size of the input images varies a lot. We padded all images to the size of the image with maximum size with mirror padding and resized all images to 1024x1024 to be able to fit the GPU memory.

FIGURE 3.8: Two different versions of the histology staining.

3.3 Laser beam characterization

3.3.1 Problem description

Free-electron lasers (FEL) belong to the fourth generation of light sources which can produce short-wavelength radiation with outstanding parameters such as spectacular brightness, great transverse coherence and very short (femtosecond) pulse duration. Radiation generated at large FEL facilities can be easily used for creation of exotic states of matter which are often at the edge of our knowledge and help us to understand processes occurring in inertial confinement fusion (ICF) or core of large gaseous planets. It has been shown that an accurate description of the beam parameters is of great importance when comparing experimental data with theoretical calculations.

Up to now, there are only a few methods which can be used for spatial characterization of the intensity profile of the focused FEL beam. One of them is a method

[4]See the official site https://www.ilastik.org/

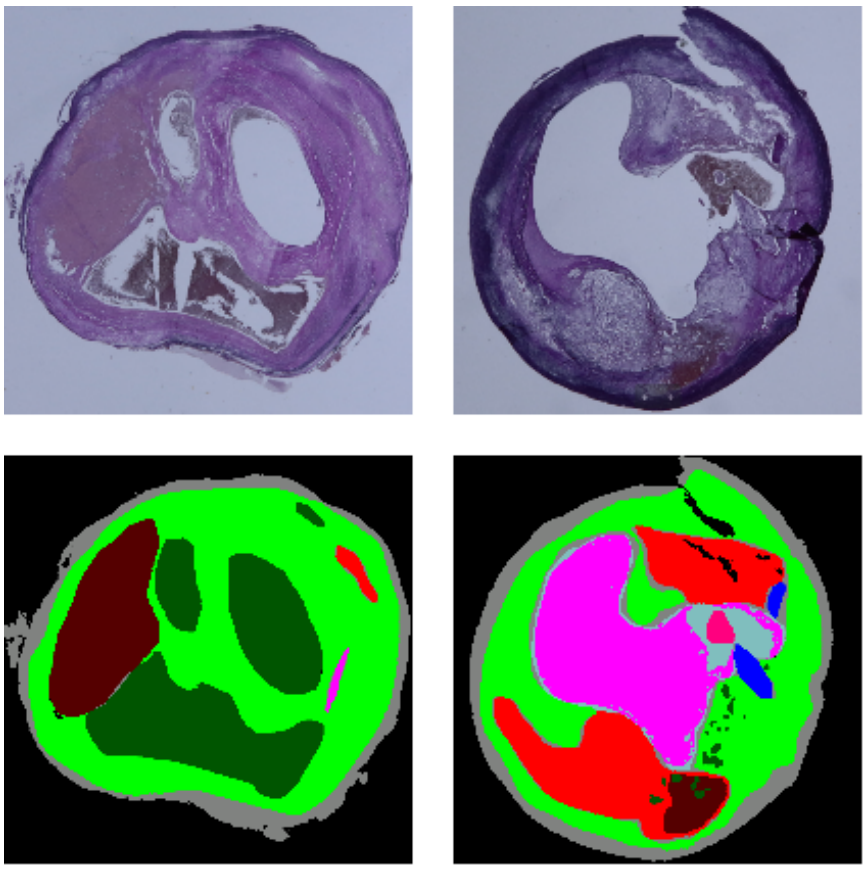

FIGURE 3.9: Examples of the histology images(first row) and the corresponding ground truth(second row).

of ablative imprints based on damage of material which is placed directly into the beam focus. This method can be used for estimation of the best focus position by making several imprints at different locations along the beam propagation axis and measuring dimensions of the imprints.

Another use of this method is a so-called fluence scan when there is no change of the sample position, but fluence of the beam is varied across several attenuation levels. Among the other methods, fluence scan is used for determination of the beam spot size which is characterized by a parameter called effective area. Usually, tens or hundreds of imprints have to be done for precise beam characterization and area of each spot has to be measured by hand. To speed-up the beam characterization procedure, automatization of imprinting and imprint measurements is the main objective to be solved. Our main task is to localize the laser trace in the imprint and calculate its area and the perimeter. The algorithm should provide a reasonable segmentation of the laser trace while maintaining a low computational complexity as it will be used on simple desktop computers.

3.3.2 Data description

Our photos were taken by Nomarski DIC microscope and show an ablative imprint to poly(methyl methacrylate) (PMMA) taken at FLASH2 (DESY, Hamburg) at photon wavelength 13.5 nm. The dataset consists of 105 images annotated by domain experts as shown in Fig.3.10. Pictures were gathered from the six separate experiments. We took four experiments(78 images) for training, one experiment(10 images) for validation and 1 experiment(17 images) for testing.

FIGURE 3.10: Example of training data. The first row is the input images, the second one is the corresponding ground truth.

3.3.3 Data preprocessing

Input images are too big(6000x4000). It is highly computationally expensive. We did two main steps to prepare data for training(see Fig.3.11):

1. Central crop(2048x2048 part of the image). We investigated all the images, which we got. It seems, that all important information(at least for this type of images and this type of dataset) located inside this crop. If this is not going to be true in the future data, we will need to add code to find the trace in the image, and if there are more than one, we shall need some means of identifying the right one.

2. Resize to 1024x1024. Downscaling of the central crops doesn't cause any noticeable deteriorations of results of our algorithms but allow efficiently store and process them.

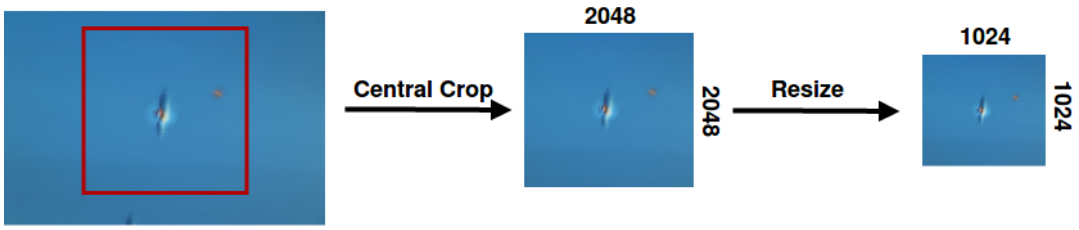

FIGURE 3.11: Example of samples from laser trace data. The first column is the input images, the second one is the corresponding ground truth.

titlesec

Chapter 4

Methodology

As we discussed in chapter 2, there is a wide range of challenges in semantic segmentation. Our goal is to solve the problem of integration of the global context information via shape priors of the training set. We extend the idea, presented by H. Ravishankar et al. [53] to the multi-class problem and investigate different segmentation architectures with an extended method.

4.1 Incorporating prior shape information into a segmentation network

4.1.1 Standart pixel-wise training losses for FCNs

Given the set of N training images and the corresponding ground truth segmentations, $\{X^{(i)}, Y^{(i)}; i = 1, 2, 3..., N\}$ the DNNs learn the transformation of the input image $X^{(i)}$ to the probability map of the same size as input that assigns a category to each pixel. Learning the weights W of the neural network is performed by minimizing the distance between the $X^{(i)}$ and $Y^{(i)}$ pairs in the training set which described by some loss function. The most commonly used loss function is a pixel-wise cross-entropy(see formula 4.1). This loss considers each pixel individually, comparing the class predictions to the target vector and then sums over all pixels.

$$L_{ce}(\mathbf{y}_{true}, \mathbf{y}_{pred}) = -\sum_{classes} y_{true} \log(y_{pred}) \tag{4.1}$$

where y_{true} - ground truth, y_{pred} - prediction of the network. Another common choice is the Dice loss(described in 4.2). It based on the Dice coefficient, which computes a measure of overlap between two samples. The metric varies from 0 to 1, where 1 is the perfect overlapping, and 0 is the empty intersection of the two objects.

$$L_{dl}(\mathbf{y}_{true}, \mathbf{y}_{pred}) = 1 - \frac{2 \sum\limits_{\text{classes}} y_{true} y_{pred}}{\sum\limits_{\text{classes}} y_{true}^2 + \sum\limits_{\text{classes}} y_{pred}^2} \tag{4.2}$$

4.1.2 Integrating shape priors into the loss function

The main drawback of the previous approach is optimizing the distance between input and target objects only pixel-wise without any global context and prior information. We expand the idea, introduced by H. Ravishankar et al. [53] and propose a multi-class loss, which encourages the output segmentation mask to be close not only to the ground truth but also to the learned shapes of the classes. There is a wide range of choices of how to represent the shape priors [47]. We used the convolutional autoencoder [35] in the role of regularization model like in [53] which is trained in end-to-end with the segmentation network. This option proposes fully-automated

extraction of information about shapes from the training set without any hardcoded formulas, user interaction and pre or postprocessing techniques.

We have a set of ground truth examples $\{Y^{(i)}; i = 1, 2, 3..., N\}$ which define some valid shape space Z. Assume that it is possible to learn some p-dimensional shape projection encoder E and a decoder D. Encoder E should be able to map any random shape S to a valid representation on Z in order to make end-to-end training with segmentation network possible. Accordingly, the composition with the decoder D, i.e. $(D \circ E)[S]$ is the projection of S onto a valid shape space Z. The $D \circ E$ can be interpreted as a convolutional de-noising autoencoder withing a loss function. We modify the traditional loss in the following way:

$$L(Y^{(i)}, \widehat{Y}^{(i)}) = (\widehat{Y}^{(i)} - (D \circ E)[\widehat{Y}^{(i)}])^2 + \lambda_1 (E[Y^{(i)}] - E[\widehat{Y}^{(i)}])^2 + \lambda_2 (- \sum_{\text{classes}} Y^{(i)} \log(\widehat{Y}^{(i)})) \tag{4.3}$$

where $Y^{(i)}$ is ground truth mask, $\widehat{Y}^{(i)} = h_w[I^{(i)}]$ is output of the segmentation network and $I^{(i)}$ is the corresponding input image.

The first term pushes the predicted shape from the segmentation network $\widehat{Y}^{(i)}$ to be close to the shape space Z by minimizing the projection error(see Fig. 4.1). The second term minimizes the distance between the encoded representations of the target object and predicted the output of the segmentation network. The last term keeps the diversity of the ground truth shape from the learned shape space Z. In the standard implementations of most common FCNs architectures since the loss function based on Euclidean distance or KL divergence, the network has to make a complex mapping from the input image to the high dimensional shape. Consequently, there is a need for enough representative training dataset to be able to learn the objects morphology, shapes and contextual dependencies. In the proposed method of training, the distance between predicted shape $\widehat{Y}^{(i)}$ and ground truth $Y^{(i)}$ encoded by shape network cause the higher complexity of the segmentation model. Meanwhile, this approach increases the number of the parameters of the segmentation model only during the training time as we discard the regularization autoencoder in the test phase and use for prediction only the segmentation model trained with prior shape information.

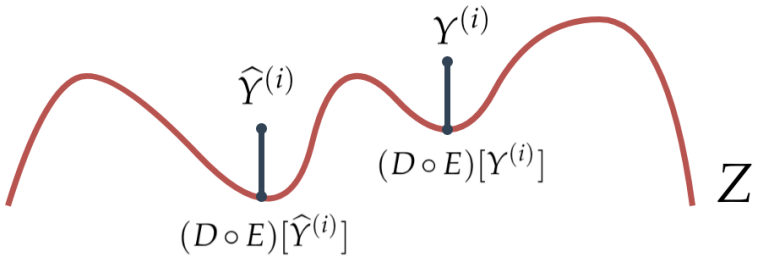

FIGURE 4.1: Projection on the shape space Z

4.2 Architectures

In this section, we explain which network architectures we utilized to implement our shape prior loss formulation in 4.3. We build a framework which consists of two networks, tightly connected and trained end-to-end. The first network performs the segmentation of the input image. The second was pretrained beforehand end-to-end training. It penalizes the first one from the deviation of the valid shape space. The whole process of training without pretraining of shape network has been shown in Fig.4.2. See more about pretraining of the shape network in 4.3.1

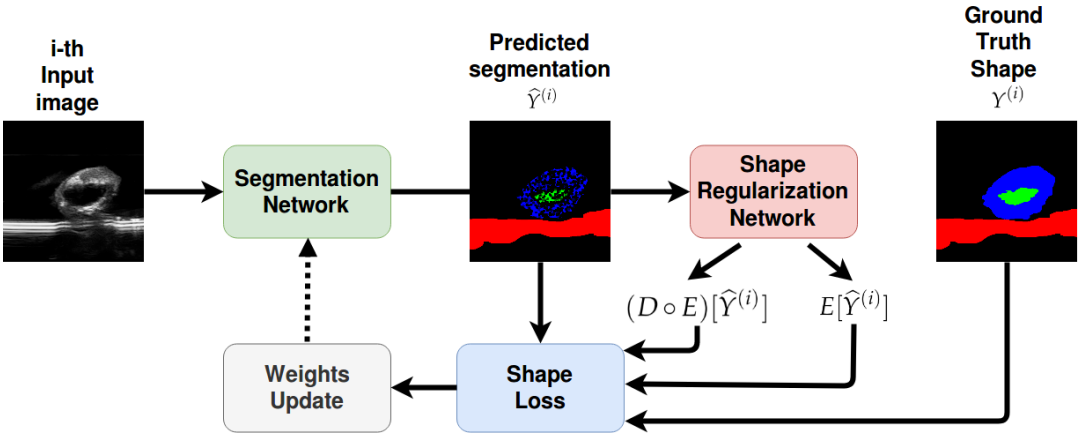

FIGURE 4.2: The pipeline of incorporating shape priors into the segmentation network

4.2.1 Shape regularization network

The main task of this network is to penalize the poor quality predictions of the segmentation network which lie far from the learned shapes from the training data. As a base for the shape network architecture, we took U-Net [55] like the authors in [53]. It contains encoder and decoder parts, which map the defective shapes into a latent representation with the help of stacks of convolutions and nonlinearities. The encoder should bring the compressed latent space representation, which is invariant to the defected incompleted input shapes, from which the decoder will be able to reconstruct the complete shape. One noticeable modification from the U-Net architecture is removing skip connections as illustrated in Fig. 4.3 in order to smooth the projection of the shape and pay more attention to the global structures rather than individual pixels.

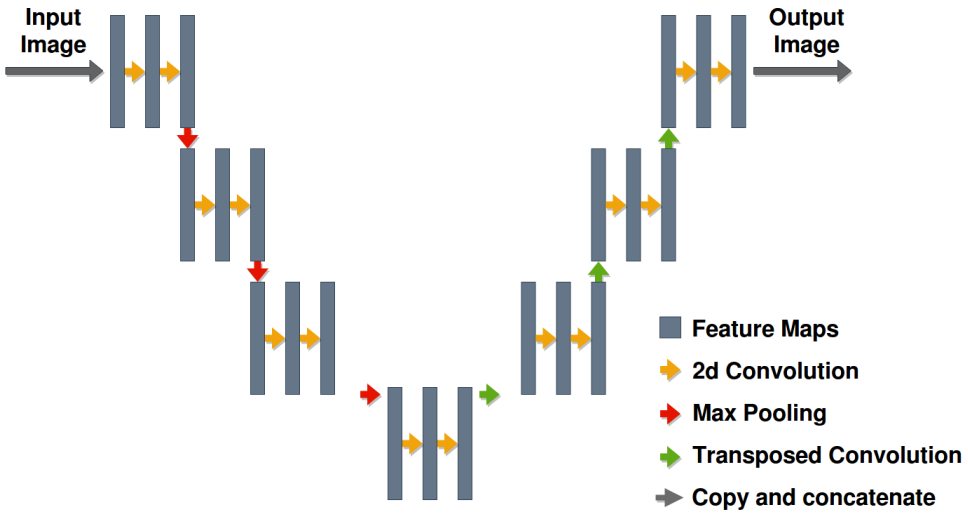

FIGURE 4.3: Shape regularization network architecture

4.2.2 Segmentation networks

Depending on the task, we employed different architectures in place of the segmentation module.

U-Net

U-Net is the most popular architecture in the biomedical image segmentation. We took it as a baseline for our experiments. See the 2.1 for more details. The only modification of the original implementation was adding the Batch Normalization[30] after each convolution layer.

Attention R2U-Net

It is the combination of the two recent advanced works. We merged the attention gate presented by Ozan Oktay[49] and recurrent residual convolution units(RRCU)[2]. The general structure of the network is very similar to the U-Net. It consists of 3 main segments: encoder, corresponding decoder part, and long skip connections.

The encoder consists of convolutional blocks. The various amount of possible structures of the convolutional building blocks has been proposed[23, 25, 27]. In this architecture, instead of the traditional convolution block, the Recurrent and Residual ideas were combined(see d in Fig.4.4). The usage of residual and recurrent blocks do not increase the numbers of parameters of the network. Nevertheless, it significantly affects training and test performance. It has been shown through empirical observations with a set of experiments in [2].

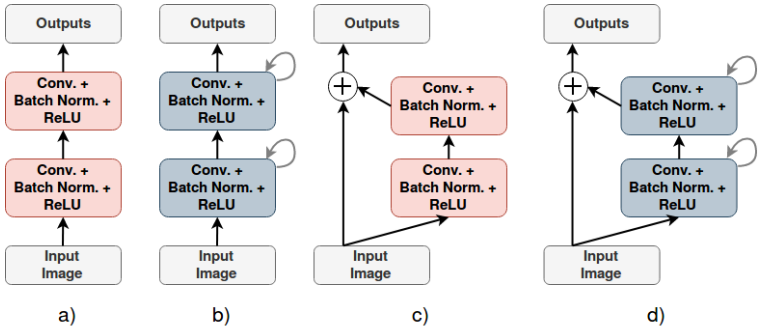

FIGURE 4.4: Different convolution blocks a) standard convolution block(used in original U-Net), b) Recurrent block, c) Residual block and d) combination of Recurrent and Residual blocks

The decoder convolution block consists of the attention gate[49], which gradually suppress the feature responses in inappropriate background areas and preserve only that regions of activations which are relevant to the specific task, upsampling of features(bilinear interpolation), followed by convolution 3x3 convolution, Batch Normalization and ReLU. The attention gates increase accuracy by reducing false positive errors, especially for small objects. The general modified architecture illustrates in Fig.4.5

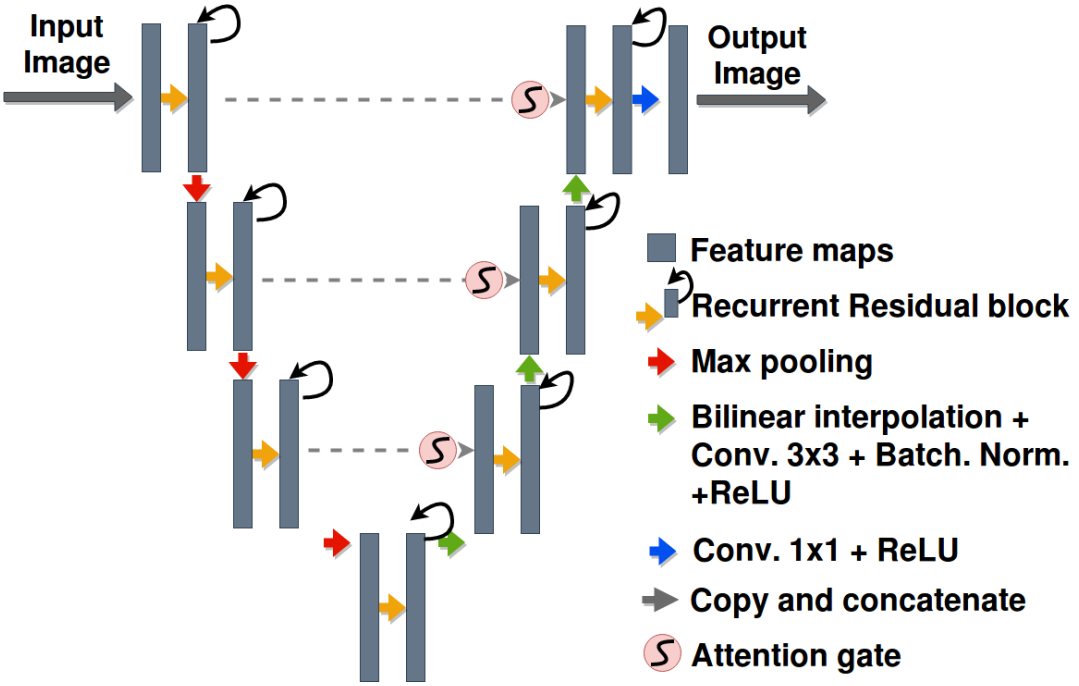

FIGURE 4.5: The Attention R2U-Net architecture

ESPNetv2

We exploited the power efficient, a lightweight network for semantic segmentation. We took the vanilla implementation of the architecture, presented in [44]. We changed only the way of optimizing the network weights. The authors used outputs from the fourth and second levels of architecture as inputs into two separate losses. The final loss value was the sum of both losses. The features only from the fourth level were utilized for the final prediction during test and validation phases. We used the sum of features from the fourth and the second level as an input only to one loss. The same combination of features was used for test and validation. More details about the results will be in the next chapter 5.

4.3 Implementation details

4.3.1 Shape network pretraining

For shape regularization network(SR) in order to use it in end-to-end training with segmentation network(SN), we need to pretrain it beforehand in corrupted shapes as input and ground truth shapes as output. As a corrupted shape input we used the intermediate predictions at differed epochs before the convergence of the SN. The process of sampling the dataset from intermediate predictions depends on the dataset. For separate training of SN we used loss, described above(see 4.1). For SR we used the following the first two terms of 4.3 without lambda coefficients:

$$L(Y^{(i)}, \widehat{Y}^{(i)}) = (Y^{(i)} - (D \circ E)[\widehat{Y}^{(i)}])^2 + (E[Y^{(i)}] - E[\widehat{Y}^{(i)}])^2 \tag{4.4}$$

4.3.2 Shape framework implementation

In this section, we want to clarify a few important aspects of the implementation of the whole shape framework. Firstly, before training the whole framework, the SN should be pretrained. Secondly, after training of the whole framework, we use only the SN(network, which was trained with shape priors) for the final prediction

on the test set. Thirdly, while training the framework, the weights of the SR should be updated as well. We did not investigate different lambda coefficients in the loss. We used 0.5 for both of them like in [53]. The last point is the SN and SR should be connected with which other, i.e. the input to the SR should be output from the SN with all history of computation, not as a constant image array.

Chapter 5

Experiments and results

In this chapter, we perform comprehensive experiments on four datasets to show the effectiveness of our proposed approaches.

5.1 Experimental set-up

5.1.1 Training details

We used Python, more specifically Python3 programing language to conduct our experiments. The network models were implemented using Pytorch with CUDA 8.0 and cuDNN back-ends. All losses, described in 4 were optimized using Adam optimizer[33]. It has lots of different parameters including learning rate, β_1 and β_2(control the decay of first and second moments), epsilon(for improving numerical stability). However, in our work, we used settings which authors mentioned in original paper [33]. Following [24], the weights of our models were initialized with He Normal method. We trained our models for 200 epochs. The reason behind that is that after 200 epochs our models were not showing any improvements.

5.1.2 Evaluation metrics

We used two common evaluation metrics for semantics segmentation.

One of them is mean pixel accuracy. It measures the ratio between the correctly predicted pixels of a specific class to the overall numbers of pixels which belong to that class and then averages over all classes5.1. This measure suitable for datasets with no or little amount of background class but might be problematic for datasets with large background class.

$$Mean\ accuracy = \frac{1}{C}\sum_{i=0}^{C}\frac{n_i}{y_i} \tag{5.1}$$

where C - number of classes, n_i - number of pixels, predicted to belong to class i, y_i - total number of pixels, which belong to class i in ground truth.

The second is the mean Jaccard Index(also called mean intersection over union, mean IoU), which has been widely used to benchmark image segmentation and object localization[42]. It takes into account false positive and false negative cases for each class. It calculates the ratio of the intersection and union of the two sets:

$$Mean\ Intersection\ over\ Union(mean\ IoU) = \frac{1}{C}\sum_{i=0}^{C}\frac{\widehat{Y}^{(i)} \cap Y^{(i)}}{\widehat{Y}^{(i)} \cup Y^{(i)}} \tag{5.2}$$

where $Y^{(i)}$ is the predicted binary mask for class i, $Y^{(i)}$ represents the ground truth binary mask for class i.

5.1.3 Validation techniques

Validation evaluates the general effectiveness of the algorithms on the independent dataset, ensuring the bias and variance trade-off. In our work, we used two validation techniques: hold-out and k-fold cross-validation [34].

Hold-out

Hold-out is when you split your dataset into three subsets: train, validation and test set. The train is used for training. The validation set is for tuning the hyperparameters of the model and test is to see how the model performs on the unseen data. The typical split is using 70% of the data for training, and the rest is for the validation and test. Our division highly depends on the data, which we used in our experiments.

K-fold cross-validation

In k-fold cross-validation, the data randomly split up into "k" folds. We use k-1 folds for model training and the last one for testing. The process repeated until each unique fold has been utilized as a test set(see Fig. 5.1. For instance, in the case of 3-fold cross-validation, the data should be split into three groups. We should train three separate models so that each fold should be in the role of a test set. Then the error rate averages over all test sets and we show the general performance of the model. The advantage of this method is that all data is used for testing and training. It gives a better indication of how well your models perform on unseen data. However, in the case of deep learning, it is very costly to train a model. That's why we used it only in 3.2.1, where the size of the training data was extremely low.

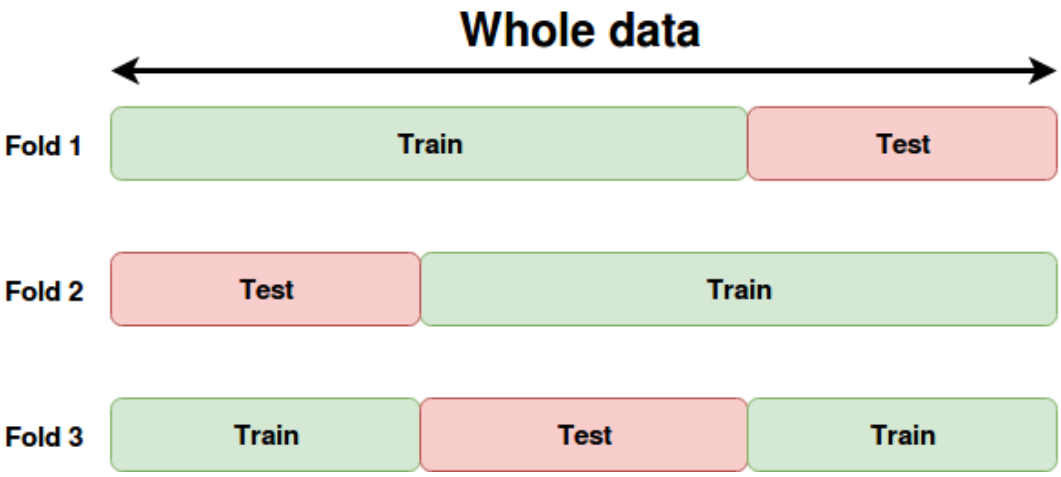

FIGURE 5.1: 3-fold cross-validation

5.1.4 Augmentation

For augmentation, we used Python library called Argumentations. The following techniques have been tried:

1. **Color augmentation**
    - CLAHE — applies Contrast Limited Adaptive Histogram Equalization to the input image.
    - Random Brightness — changes brightness of a random channel or of a combinations of channels

- Random Contrast — changes contrast of a random channel or of a combinations of channels
- Random Gamma — applies gamma contrast adjustment

2. **Non-Rigid Augmentation**
   - Elastic Transform — moves each pixel individually around based on distortion fields
3. **Non destructive transformations**
   - Horizontal Flip
   - Vertical Flip
   - Transpose
   - Random Scale
   - Rotation

5.2 Results

5.2.1 Binary synthetic dataset

As we mentioned in the Introduction chapter 1.3, the authors of the [53] didn't provide enough details about implementation of their idea. We used dataset 3.1.1 to check our hypotheses about possible implementations details and as a sanity check for the performance of the algorithms. We hypothesized that the more noisy and complicated data we have, the better is the performance of the regularized segmentation network over the baseline model(see Table 5.1). As a baseline, we consider the same segmentation architecture, which was trained without the shape regularization. Our experiments proved this hypothesis. Moreover, the baseline outperforms the model, which was trained with shape regularization on the low-level noise dataset. The reason for that is the regularization usually makes the predictions of the model more smooth and less dependent on raw input image structures. The are two possible outputs from the shape regularization framework, which we could measure. The first one is the output of the segmentation network and the second one is the output from the shape regularization network. As we pointed out in the Implementation details4.3, the segmentation network trained with shape regularization should be used for final testing, the outcomes of our experiments also confirm that.

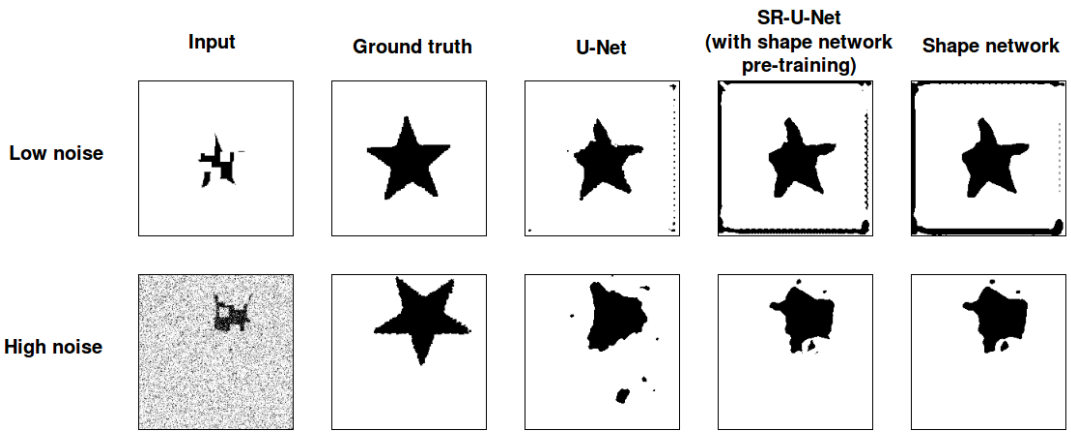

FIGURE 5.2: Qualitative comparison on binary segmentation datset. Note, that in a very low noise dataset the U-Net without regularization performs better.

Models	Low noise		Medium noise		High noise	
	Mean IoU (%)	*Mean Pixel Accuracy (%)*	*Mean IoU (%)*	*Mean Pixel Accuracy (%)*	*Mean IoU (%)*	*Mean Pixel Accuracy (%)*
U-Net	**96.2**	**97.9**	88.5	93.6	77.7	86.0
Shape network (without shape network pre-training)	95.8	97.8	88.2	93.7	78.0	86.9
Shape network (with shape network pre-training)	95.6	97.9	89.5	94.9	78.5	**87.0**
SR-U-Net (without shape network pre-training)	95.9	97.8	88.2	94.8	78.1	86.8
SR-U-Net (with shape network pre-training)	95.9	**97.9**	**89.5**	**94.9**	**78.6**	86.9

TABLE 5.1: Results of our models on the binary segmentation dataset with different levels of noise. SR-U-Net is the shape regularized U-Net.

5.2.2 Multiclass synthetic dataset

This dataset 3.1.2 has been utilized as a validation of our multiclass extension of the work [53]. Our results(see Table 5.2) show that our extension capable of capturing multiclass dependencies and performs much better than a baseline(vanilla U-Net, without shape regularization). Our experiments also demonstrate the superiority of the segmentation network over shape regularization network. It is also worth noting that our modification of the method [53] introduces additional computational cost only during training. The segmentation network after being trained has the same computational complexity as an original model but higher prediction capacity. The fact, that shape regularized segmentation network, trained without pre-trained shape regularization network outperforms that one, which was trained with pre-training shows that the pre-training strategy was not optimal for the dataset. We used a general approach for pretraining, described in Implementation details 4.3. However, there is a wide range of possible solutions for improvement(see Future work 6.2). Note, that the shape regularized segmentation produces much less noise in the output and tend to capture the general class dependency(one class surrounded by another) much better than plain U-Net.

5.2.3 Ultrasound of the carotid artery

The dataset is extremely low, only 35 images. For estimating our models, we used a 3-fold cross-validation technique to be able to test on all images in the dataset. The results in Table 5.3 averaged over all validation datasets(three of them). We observe the negligible difference in the quantitative comparison between the regularized model and the baseline(see Table 5.3). It might be caused by a very limited amount of training data and low quality of annotated examples. On the other hand, we can inspect visually, that shape regularized network prone to produce more accurate predictions in tough cases.

As the size of the dataset is extremely slow, we used the tough augmentation strategies. The performance of trained models on different augmentation strategies was tested on the same test set (see the results in the Fig. 5.4).

Models	Low noise		High noise	
	Mean IoU (%)	*Mean Pixel Accuracy (%)*	*Mean IoU (%)*	*Mean Pixel Accuracy (%)*
U-Net	91.2	96.9	75.8	87.6
SR-U-Net (without shape network pre-training)	95.0	97.8	77.1	**88.2**
SR-U-Net (with shape network pre-training)	95.2	97.9	76.9	87.3
Attention R2U-Net	95.9	98.0	77.3	87.5
Attention SR-R2U-Net (without shape network pre-training)	**96.4**	**98.3**	**77.5**	87.0
Attention SR-R2U-Net (with shape network pre-training)	96.2	98.1	74.5	87.1

TABLE 5.2: Performance of the proposed method on the artificial multiclass dataset. Attention SR-R2U-Net is the shape regularized Residual Recurrent U-Net with attention gates.

5.2.4 Histology of the carotid artery

We used a hold-out validation strategy on this data as well. We took 77 images for training, 5 for validation and 10 for the test set. The experiments show(Table 5.4) that the performance on this task is quite poor, near 50% of mean IoU. We can inspect visually, that the shape regularized network capable of catching one more class - calcification(blue color in the prediction examples 5.6), which is very important for evaluating a level of stenosis in the carotid artery. The low-level performance of the algorithms in this task caused by the bad quality of provided annotated data, a large number of classes and unbalanced amount of pixels per each category. The additional data preprocessing steps(signal smoothing in the ground truth) or some specific training strategies e.g., weakly supervised learning might be helpful in this task.

5.2.5 Laser beam dataset

We hypothesized that shape prior information might be especially beneficial in case of optimized segmentation networks as their predictive power is much weaker because of their smaller receptive fields, pruned decoder structure, and reduced encoder path. The results of our experiments show that the performance of the optimized network with shape regularization(SR-ESPNetv2) is near the performance of the much complex algorithms (U-Net), see Table 5.5. Visually, the segmented by an optimized network with shape prior information objects look more smooth, i.e., it fails to capture little details(see the Fig. 5.7). However, it catches the global shapes of the objects and structure very well achieving high test mean IoU score(near 94%).

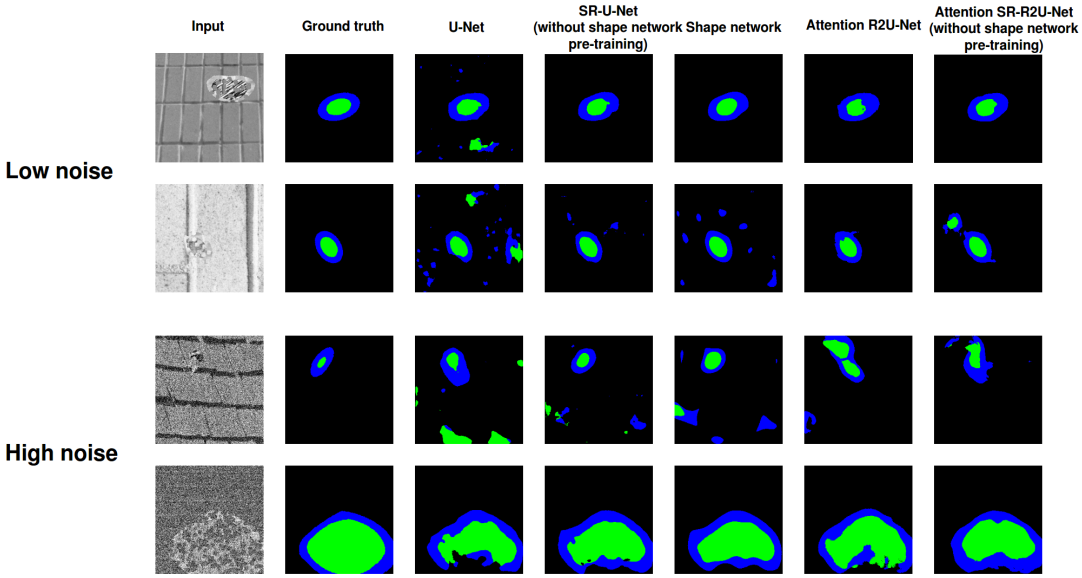

FIGURE 5.3: The visual comparison of our models with and without shape regularization on the multiclass dataset.

5.2.6 Statistical testing

It might be the case, that we can not spot a big quantitative difference in the performance of the model with and without shape regularization. Although, visually, we can inspect a huge influence of the shape regularization on the model performance. We used the nonparametric statistical Wilcoxon Signed-Rank Test [54] on the samples of models skills scores(mean IoU and mean Pixel Accuracy) to confirm that the influence of the shape regularization on our model is significant. The p-values in all datasets vary 0.001+-0.004, statistic values are 16823+-5000. The p-values are interpreted strongly suggesting, that the outputs from the models are from different distributions. Accordingly, the shape regularization term affects the training of the model intensively.

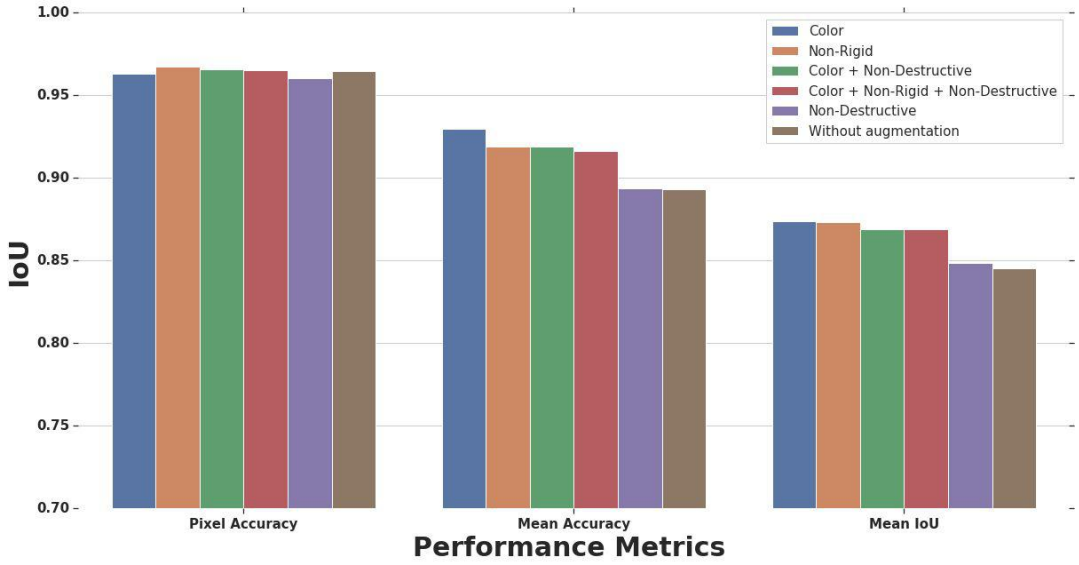

FIGURE 5.4: Comparison of the different augmentation strategies on Ultrasound dataset

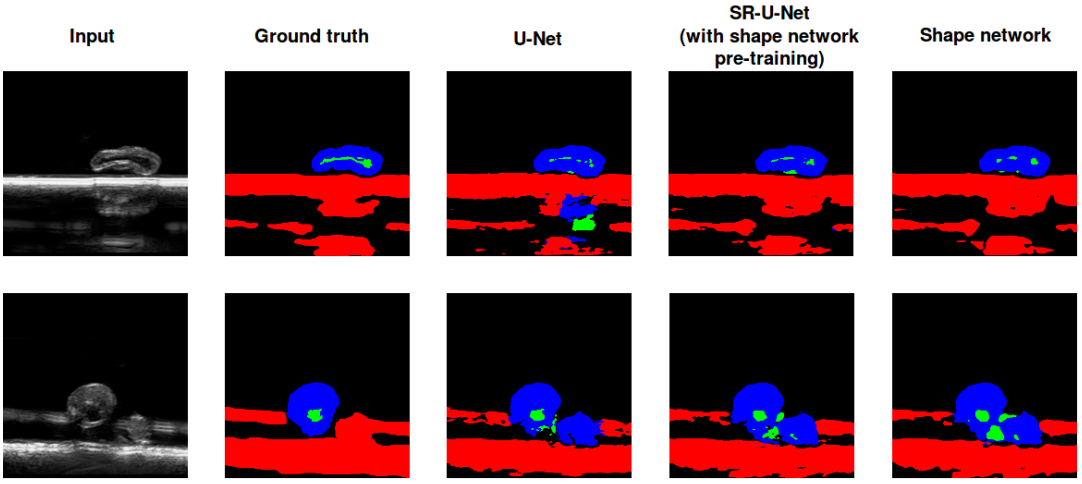

FIGURE 5.5: Prediction of the algorithms on the ultrasound data

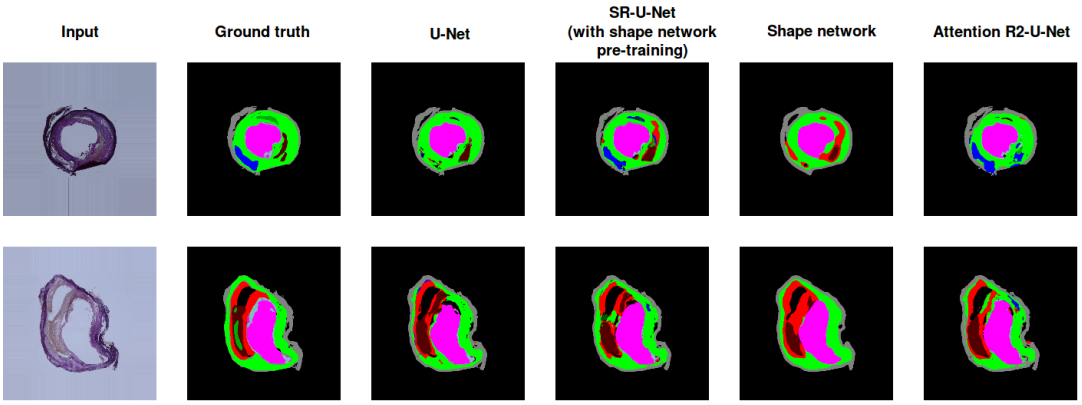

FIGURE 5.6: Prediction of the algorithms on the histology data

Models	Mean IoU (%)	Mean Pixel Accuracy (%)
U-Net	84.7 ± 5.1	91.8 ± 2.3
Shape network (without shape network pre-training)	84.6 ± 4.5	91.8 ± 1.9
Shape network (with shape network pre-training)	84.2 ± 5.2	91.3 ± 2.6
SR-U-Net (without shape network pre-training)	**85.0 ± 4.8**	**91.9 ± 2.0**
SR-U-Net (with shape network pre-training)	84.7 ± 5.2	91.3 ± 2.7

TABLE 5.3: Results on the ultrasound of carotid artery dataset, averaged over all validation test sets.

Models	*Mean IoU (%)*	*Mean Pixel Accuracy (%)*
U-Net	51.0	60.0
Attention R2U-Net	50.9	**62.4**
Shape network (without shape network pre-training)	48.6	57.6
Shape network (with shape network pre-training)	50.0	59.4
SR-U-Net (without shape network pre-training)	51.4	60.5
SR-U-Net (with shape network pre-training)	**51.4**	60.6

TABLE 5.4: Comparison of the methods on the histology data.

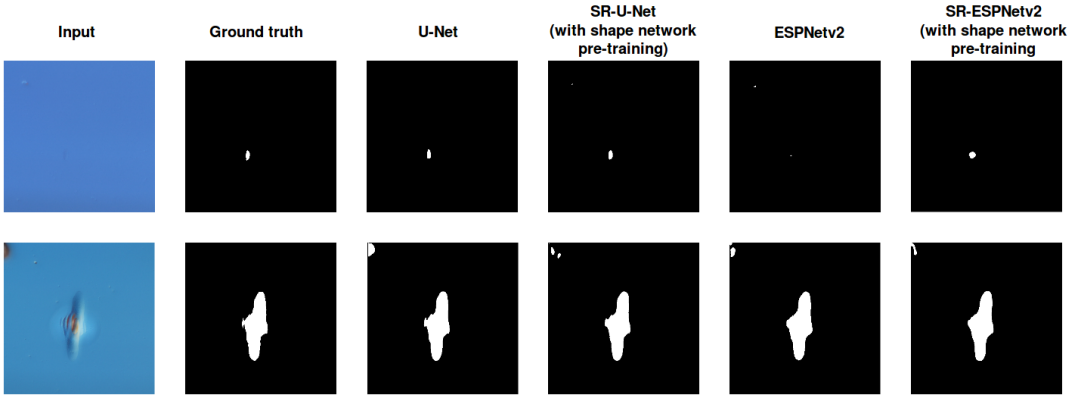

FIGURE 5.7: Prediction of the algorithms on the laser beam dataset.

Models	Mean IoU (%)	Mean Pixel Accuracy (%)	Number of parameters (in millions)	Size of the model (in MB)
U-Net	96.0	97.7	53.09	212
SR-U-Net (without shape network pre-training)	96.0	**98.2**	53.09	212
SR-U-Net (with shape network pre-training)	**96.5**	98.0	53.09	212
ESPNetv2	91.7	93.9	**1.24**	**5.1**
SR-ESPNetv2 (without shape network pre-training)	93.9	97.3	**1.24**	**5.1**
SR-ESPNetv2 (with shape network pre-training)	92.7	95.3	**1.24**	**5.1**

TABLE 5.5: The results of the algorithms in the Laser beam data. ESPNetv2 is the Efficient Spatial Pyramid Network, SR-ESPNetv2 is the shape regularized ESPNetv2.

Chapter 6

Conclusion

6.1 Brief summary

In this work, we considered the problem of semantic image segmentation. We learned the non-linear shape model, which projects arbitrary masks into a shape space. We incorporated the shape prior information into a segmentation network via loss function, which penalizes the deviation of prediction of the segmentation network from the learned shape model, presented in [53]. We further extend it to the multiclass segmentation problem. Our experimental results show that the proposed method constantly advances the previous techniques, which were trained without shape prior in five challenging datasets. We also trained different segmentation architectures with a shape priors to show, that this method can be generalized into any segmentation architecture and still provide some benefits. The prior shape method will be especially profitable in case of improving the performance of optimized networks, which lack of predictive power, but computationally are very efficient.

6.2 Future work

Our work has a wide range of possible directions for future work:

Shape network pretraining

Experiments on our datasets showed that it is often the case, that pretraining strategy of the shape network is not optimal as results without pretraining of shape network sometimes performed better. We assume that adding some noise and augmentation to the intermediate predictions of the segmentation network, on which we trained shape model can help to learn the projector onto a shape space.

3D image segmentation

Our extension might be easily generalized to 3D medical image segmentation in such modalities as Computed Tomography(CT) or Magnetic Resonance Imaging(MRI).

Shape network architecture

We used the same network architecture as in [53] for shape regularization model. We suppose that there are more suitable options for the shape model. Also there an option of adding more networks which will incorporate the shape information. For example, it might be the discriminator, which will penalize the shapes, which doesn't conform to the manifold of the shapes from the training data.

Other types of prior information

We used only the shape prior. There are other types of prior information, which have been widely used in traditional computer vision approaches like topology, moment,

spatial distance, etc. The question is how to incorporate such types of prior into the training process of the CNNs.

Bibliography

[1] M. Ezzati D. Jamison A. Lopez C. Mathers and C. Murray. "Global and regional burden of disease and risk factors, 2001: systematic analysis of population health data". In: *The Lancet* (May 2006).

[2] Md Zahangir Alom et al. *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*. 2018.

[3] Mohamed Babikir Ali, Ruba Ali Hamad, and Mohanned Ahmed. "Optimizing Convolutional Neural Networks for Brain Tumor Segmentation in MRI Images". In: Aug. 2018, pp. 1–5. DOI: 10.1109/ICCCEEE.2018.8515826.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "SURF: Speeded Up Robust Features." In: vol. 110. Jan. 2006, pp. 404–417.

[5] Alexandre de Brebisson and Giovanni Montana. "Deep neural networks for anatomical brain segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2015). DOI: 10.1109/cvprw.2015.7301312. URL: http://dx.doi.org/10.1109/CVPRW.2015.7301312.

[6] L.S. Davis C. Huang and J.R.G. Townshend. "An assessment of support vector machines for land cover classification". In: (2002). DOI: 10.1080/01431160110040323.

[7] É. Therasse P. Robillard M. Giroux F. Arsenault G. Cloutier C. Naim M. Douziech and G.Soulez. "Vulnerable Atherosclerotic Carotid Plaque Evaluation by Ultrasound, Computed Tomography Angiography, and Magnetic Resonance Imaging: An Overview". In: *Canadian Association of Radiologists Journal* (Oct. 2014).

[8] Fei Chen et al. "Deep Learning Shape Priors for Object Segmentation." In: *CVPR*. IEEE Computer Society, 2013.

[9] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018). DOI: 10.1109/tpami.2017.2699184.

[10] Liang-Chieh Chen et al. "Rethinking Atrous Convolution for Semantic Image Segmentation". In: (2017). arXiv: 1706.05587 [cs.CV].

[11] Liang-Chieh Chen et al. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*. 2014. arXiv: 1412.7062 [cs.CV].

[12] Yunmei Chen et al. "Using Prior Shapes in Geometric Active Contours in a Variational Framework." In: *International Journal of Computer Vision* (2002).

[13] Dan C. Ciresan et al. "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images". In: 2012.

[14] T. F. Cootes et al. "Active shape models—their training and application". In: *Comput. Vis. Image Underst.* (1995).

[15] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). DOI: 10.1109/cvpr.2016.350. URL: http://dx.doi.org/10.1109/CVPR.2016.350.

[16] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[17] Michal Drozdzal et al. "The Importance of Skip Connections in Biomedical Image Segmentation". In: *Lecture Notes in Computer Science* (2016).

[18] T. Falk et al. "U-Net – Deep Learning for Cell Counting, Detection, and Morphometry". In: *Nature Methods* (2019).

[19] Khoury JC et al. Flaherty ML Kissela B. "Carotid artery stenosis as a cause of stroke". In: *NCBI* (June 2012).

[20] Andreas Geiger. "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 3354–3361. ISBN: 978-1-4673-1226-4. URL: http://dl.acm.org/citation.cfm?id=2354409.2354978.

[21] Adel Hafiane, Pierre Vieyres, and Alain Delbos. "Deep learning with spatiotemporal consistency for nerve segmentation in ultrasound images". In: (2017). arXiv: 1706.05870 [cs.CV].

[22] Peihan Hao et al. "Semantic Segmentation for Traffic Scene Understanding Based on Mobile Networks". In: *SAE Technical Paper*. SAE International, Aug. 2018. DOI: 10.4271/2018-01-1600. URL: https://doi.org/10.4271/2018-01-1600.

[23] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[24] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015).

[25] Kaiming He et al. "Identity Mappings in Deep Residual Networks". In: *Lecture Notes in Computer Science* (2016).

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

[27] Gao Huang et al. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[28] Vladimir Iglovikov and Alexey Shvets. *TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation*. 2018.

[29] Vladimir Iglovikov et al. "TernausNetV2: Fully Convolutional Network for Instance Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018).

[30] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015.

[31] Simon Jegou et al. "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2017). DOI: 10.1109/cvprw.2017.156.

[32] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. "Snakes: Active contour models". In: *International Journal of Computer Vision* (1988).

[33] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014.

[34] Ron Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection". In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*. 1995. URL: http://ijcai.org/Proceedings/95-2/Papers/016.pdf.

[35] Maximilian Kohlbrenner. "Pre-Training CNNs Using Convolutional Autoencoders". In: 2017.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: 2012.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* 2012.

[38] Guosheng Lin et al. "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: 10.1109/cvpr.2017.549. URL: http://dx.doi.org/10.1109/CVPR.2017.549.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[40] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[41] Ningning Ma et al. "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design". In: *Lecture Notes in Computer Science* (2018), 122–138. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01264-9_8. URL: http://dx.doi.org/10.1007/978-3-030-01264-9_8.

[42] Kevin McGuinness and Noel O'Connor. "A comparative evaluation of interactive segmentation algorithms". In: *Pattern Recognition* (Feb. 2010).

[43] Sachin Mehta et al. "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation". In: *Lecture Notes in Computer Science* (2018), 561–580. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01249-6_34. URL: http://dx.doi.org/10.1007/978-3-030-01249-6_34.

[44] Sachin Mehta et al. *ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network*. 2018.

[45] Zahra Mirikharaji and Ghassan Hamarneh. "Star Shape Prior in Fully Convolutional Networks for Skin Lesion Segmentation." In: *CoRR* (2018). URL: http://dblp.uni-trier.de/db/journals/corr/corr1806.html#abs-1806-08437.

[46] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation." In: *CoRR* (2015).

[47] Masoud S. Nosrati and Ghassan Hamarneh. *Incorporating prior knowledge in medical image segmentation: a survey*. 2016. arXiv: 1607.01092 [cs.CV].

[48] Ozan Oktay et al. "Anatomically Constrained Neural Networks (ACNN): Application to Cardiac Image Enhancement and Segmentation". In: *CoRR* (2017). URL: http://arxiv.org/abs/1705.08302.

[49] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. arXiv: 1804.03999 [cs.CV].

[50] R. Nicoll G. Bajraktari P. Wester P. Ibrahimi F. Jashari and M. Henein. "Coronary and carotid atherosclerosis: How useful is the imaging?" In: *NCBI* (Feb. 2013).

[51] Adam Paszke et al. *ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation*. 2016. arXiv: 1606.02147 [cs.CV].

[52] Tobias Pohlen et al. "Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: 10.1109/cvpr.2017.353. URL: http://dx.doi.org/10.1109/CVPR.2017.353.

[53] Hariharan Ravishankar et al. "Learning and Incorporating Shape Models for Semantic Segmentation". In: Sept. 2017. DOI: 10.1007/978-3-319-66182-7_24.

[54] Denise Rey and Markus Neuhäuser. "Wilcoxon-Signed-Rank Test". In: (2011). Ed. by Miodrag Lovric, pp. 1658–1659. URL: https://doi.org/10.1007/978-3-642-04898-2_616.

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (2015), 234–241. ISSN: 1611-3349. DOI: 10.1007/978-3-319-24574-4_28. URL: http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[56] Mikaël Rousson and Nikos Paragios. "Shape Priors for Level Set Representations." In: *ECCV (2)*. 2002.

[57] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. "Object Class Segmentation using Random Forests." In: *BMVC*. British Machine Vision Association, 2008.

[58] Selim Seferbekov et al. "Feature Pyramid Network for Multi-class Land Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018). DOI: 10.1109/cvprw.2018.00051. URL: http://dx.doi.org/10.1109/CVPRW.2018.00051.

[59] Mennatullah Siam et al. "A Comparative Study of Real-Time Semantic Segmentation for Autonomous Driving". In: June 2018. DOI: 10.1109/CVPRW.2018.00101.

[60] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[61] Zhang YM. Song ZZ. "Contrast-enhanced ultrasound imaging of the vasa vasorum of carotid artery plaque." In: *NCBI* (July 2015).

[62] Paul Sturgess et al. "Combining Appearance and Structure from Motion Features for Road Scene Understanding." In: *BMVC*. 2009.

[63] Christian Szegedy et al. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). DOI: 10.1109/cvpr.2015.7298594.

[64] Mohammad Tofighi et al. "Deep Networks with Shape Priors for Nucleus Detection". In: (2018).

[65] Ji Wan et al. "Deep Learning for Content-Based Image Retrieval: A Comprehensive Study". In: *Proceedings of the 22Nd ACM International Conference on Multimedia*. MM '14. Orlando, Florida, USA: ACM, 2014, pp. 157–166. ISBN: 978-1-4503-3063-3. DOI: 10.1145/2647868.2654948. URL: http://doi.acm.org/10.1145/2647868.2654948.

[66] Panqu Wang et al. "Understanding Convolution for Semantic Segmentation". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2018).

[67] Xiangyang Wang, Ting Wang, and Juan Bu. "Color image segmentation using pixel wise support vector machine classification." In: *Pattern Recognition* (2011). URL: http://dblp.uni-trier.de/db/journals/pr/pr44.html#WangWB11.

[68] Zhengyang Wang and Shuiwang Ji. "Smoothed Dilated Convolutions for Improved Dense Prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining - KDD '18* (2018).

[69] Changqian Yu et al. "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation". In: *Lecture Notes in Computer Science* (2018), 334–349. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01261-8_20. URL: http://dx.doi.org/10.1007/978-3-030-01261-8_20.

[70] Fisher Yu and Vladlen Koltun. *Multi-Scale Context Aggregation by Dilated Convolutions*. 2015. arXiv: 1511.07122 [cs.CV].

[71] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. "Adaptive deconvolutional networks for mid and high level feature learning". In: *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*. 2011. DOI: 10.1109/ICCV.2011.6126474.

[72] Xiangyu Zhang et al. "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018).

[73] Hengshuang Zhao et al. "ICNet for Real-Time Semantic Segmentation on High-Resolution Images". In: *Lecture Notes in Computer Science* (2018), 418–434.

[74] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: 10.1109/cvpr.2017.660. URL: http://dx.doi.org/10.1109/CVPR.2017.660.

[75] Shuai Zheng et al. "Conditional Random Fields as Recurrent Neural Networks". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015).