Detecting Sarcomere Structures in Cardiomyocytes

Experiments on biomedical images using deep learning algorithms

Michael Wheeler

Supervised by Dr. Lu Cao

A thesis presented for the degree of
Bachelor of Computer Science

Universiteit Leiden

Leiden Institute of Advanced Computer Science
Leiden University
The Netherlands
8-2021

Detecting Sarcomere Structures in Cardiomyocytes

Experimenting on biomedical images using deep learning algorithms

Michael Wheeler

Abstract

Chemotherapy induced hart problems are undesirable when treating cancer patients. Yet cardiotoxicity is a serious adverse effect of many conventional anti cancer drugs. In this study we explore cardiotoxicity through observation of sarcomere structures in cardiomyocytes. We notice these adverse effects through malformations or a reduced number of sarcomeres. We use a high-throughput microscope to gather the input images. Our aim is to develop a method for delineating sarcomere structures using a deep learning algorithm. Usually, sarcomere structures are annotated using a set of imaging filters on microscopic images of cardiomyocytes. This kind of analysis reaches an 80% accuracy. Our approach involves experimenting with a set of deep learning algorithms that might be capable of more accurate results. We find that U-Net is very good at this task and and can reach to a comparable performance as rule-based method. In the future we want to compare our results against a small set of manually delineated sarcomeres as ground truth.

Contents

Chapter 1

Introduction

Over the recent years computer vision deep learning algorithms have become an important research field. Researchers' use of deep learning algorithms to solve biomedical imaging analysis has especially become a point of focus. With many real world and research applications this area has become increasingly more meaningful[13]. The problem that this study tries to solve is that of sarcomere segmentation. A sarcomere is the functional unit of striated muscle. Sarcomeres are the most basic building block of muscle and is what allows all voluntary movement. We are specifically interested in the sarcomere structures within a heart muscle cell or cardiomyocyte.

This research is important because it can help with detecting problems in certain anti-cancer drugs. The situation now is that some anti-cancer drugs are cardiotoxic. Not a lot is known about how damaging this cardiotoxicity can be in the long term. So researchers have decided that this is a problem that needs to be undertaken. To study the cardiotoxicity of anti-cancer drugs they inject the drugs into a heart cell and analyze the effect it has on the number and size of the sarcomeres. To effectively analyze the effect they will need a time sequence of the sarcomere structures using images. Then of the images they gather a binary mask which is made of these images using a specifically developed rule-based algorithm. They can then analyze these images for missing or shrinking or deforming sarcomeres and judge the cardiotoxicity of the drug that has been administered. The problem they face is that their current method of creating the binary mask is only 80% accurate. We believe that we can find a deep learning algorithm that can perform better than this baseline.

The amount of medical data and images produced by current medical devices is enormous therefore medical professionals can no longer keep up by traditional methods of having experts looking through every image and annotating regions of interest. For example decease prediction on MRI images [12], experts are assisted by new deep learning models in order to save time and be more accurate. For now medical experts are just as accurate when compared to deep learning techniques according to this quantitative study [21] their conclusion was that the diagnostic performance of deep learning models is equivalent to that of health-care professionals.

Automatic sarcomere instance segmentation is important because it is a very time intensive task and can only be done by an expert or using a rule based approach but with less accuracy. Also in our research we are dealing with high throughput imaging which means that we have more data than is possible to annotate by hand. This also means that any deep learning algorithm must have a high frame per second

detection rate otherwise it will not be of much use.

We want to analyse different deep learning algorithms in order to establish which algorithm will give us the best results in regard to sarcomere image segmentation or sarcomere instance segmentation. Image segmentation means that for each pixel our model will determine whether its part of a sarcomere or part of the background. In general image segmentation will classify each pixel to its correct class but we only have two classes so we get a binary image. Instance segmentation is more refined in such a way that it will not only find which pixels belong to the sarcomere class but it will identify each distinct object in an image. Since these algorithms have shown promising results for similar projects we expect that we will find an algorithm that can give us better results than our current rule based algorithm.

The current pipeline is outlined in figure 1.1, first we create a preprocessed image using a set of image preprocessing filters. Then from this preprocessed image we create a binary mask. This mask is used as the ground truth for our deep learning training model. Then we pass the ground truth as training data to our models. In this article we will discuss four different deep learning techniques. Two of these techniques are focused on instance segmentation and the other two are focused on semantic segmentation. Our first instance segmentation algorithm is called Mask RCNN. This model has seen widespread use in the biomedical imaging field. For example in 2018 data science bowl on nuclei segmentation[7]. The Mask RCNN code is open source and available on GitHub [2]. We compare Mask RCNN to another instance segmentation algorithm YOLOv4. YOLOv4 is a good choice for this research because it can detect instances fast and with precision which is important for us because an image can have more than 1500 instances and we have a lot of images we need to detect.

We are also experimenting with two semantic segmentation algorithms. Semantic segmentation is the process of labeling each pixel in an image to its corresponding class. The first is called U-Net and has seen success in for example the 2018 data science bowl. U-Net is a lightweight image segmentation algorithm designed to do fast semantic segmentation. Our second model is called SegNet. SegNet has an encoder-decoder structure for predicting masks. In this study we have focused on models that have low inference time. The difference between these models is that U-Net creates a feature map of the input image while SegNet uses its encoder-decoder network to make inferences about the image. U-Net's advantage is that is has been used in biomedical imaging before and is specifically made for this type of data. U-Net's model is bigger in size than SegNet and also takes longer to train.

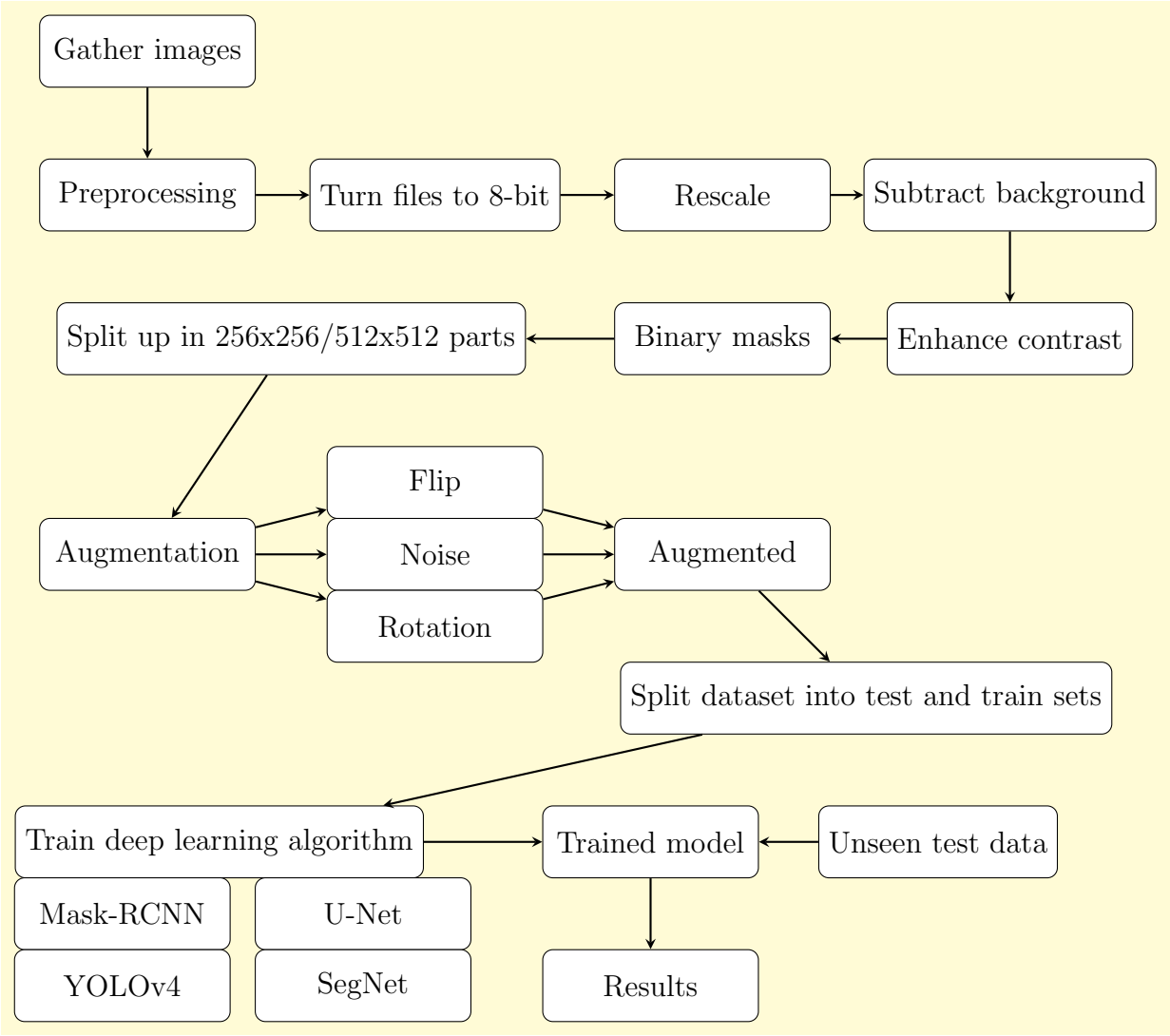

Figure 1.1: Pipeline of creating image data and training the models

Chapter 2

Related Work

2.1 Related Work

In our related work section we will be discussing different articles that have done experiments with deep learning techniques. These articles discuss segmentation of biomedical images or the identification of different objects in biomedical images. Many researchers have started experimenting with deep learning techniques for classification and segmentation of biomedical images. The development in these fields make it attractive since automating these processes can help the medical community save time and energy. Now that these techniques are more well-known and have produced quality results. In a similar study to ours[18] researcher compares U-Net and SegNet[5] performance of segmentation of breast ultrasonography images. The goal of this research was to support radiologists in this time consuming process. Also the medical technique is quite invasive therefore any way to reduce these procedures is favorably looked upon. The researchers trained their models on an image database of 2054 images that were provided by the National Cancer Institute. They found that U-Net obtained a dice coefficient of 86.3% and SegNet obtained a dice coefficient of 81.1%. They concluded that the networks' accuracy is good enough to be used in a real environment and U-Net is slightly favoured.

There have been experiments with combining these models as well. In the researchers combined Mask-RCNN [10] and U-Net [16] in an effort to segment nuclei[19]. The predictions from U-Net and Mask-RCNN were used as input for the ensemble model. Their output consisted of overlapping masks between U-Net and Mask-RCNN. The mask with the highest Intersection over Union (IoU) value is used. Non overlapping masks were added to the output segmentation if their IoU was above the threshold. Using this method the researchers got a mean average precision of 52,3% which was higher than the models' separate mAP but by no more than 1%. In the end they concluded that while the ensemble model had similar results, the models made different errors and combining them did increase its predictive power. A study done in 2020 found that an enhanced light weight U-Net model [11] can be used to accurately perform cell nuclei segmentation. The researchers changed the model in two ways that increased the performance and accuracy of U-Net. Their model when tested on the 2018 Kaggle Science Bowl contest outperformed other methods by 1% to 3%.

A different type of deep learning model is YOLOv4[5]. In this research the researchers tried to find melanoma on images of the skin[4]. Melanoma is a skin cancer caused by radiation from the sun. The researchers preprocessed the image to remove noise for example hairs. Afterwards the images are fed into the YOLOv4 algorithm which produced significant results. The data set to evaluate on was ISIC2018 and ISIC2016, the research reached an average dice score of 1 and Jaccard coefficient of 0.989. A recurrent theme is using the predictions of one deep learning algorithm as guidance for another method as we can see in [6]. In this study a three-stage skin lesion segmentation was proposed. In the first stage Mask-RCNN was used to detect the pixels belonging to the skin lesion from the input image. Then they made a CNN for accurately being able to predict the high-level features of the input image. In the last stage they used the geodesic method to extract the boundary of this new CNN. Their method performed better than other comparable technologies on well-known data sets such as ISBI 2016, ISBI 2017, and ISBI 2018 which were compiled by the International Skin Imaging Collaboration (ISIC) they also used PH2 and DermQuest.

Chapter 3

Data and Preprocessing

3.1 Data set

The proposed segmentation method is evaluated on our own custom data set. The data set was prepared by Dr. Lu Cao. The data set consists of 120 images of 1328x1048 in the .tif format. Each image contains around 1500 sarcomeres. Sarcomere size can range from 1.6 μm to 2.2 μm [8]. The size of the sarcomeres in the images ranges from 5 pixels to 15 pixels. The microscopic images were gathered using a high-throughput automated EVOS FL Auto 2 Thermo Fisher microscope. The monolayer cell culture was scanned by acquiring 55 images per well every 24 hours for 5 days. The cells were maintained at 37 °C and 5% CO_2 in the incubator. The human pluripotent stem cell-derived cardiomyocytes (hPSC-CMs) were treated with dimethylsulfoxide (DMSO 4.23 mM) as control or with 1 μm of the anti cancer drug Doxorubicin 10-12 days after seeding for 4 days.

3.2 Preprocessing

Before we use the data set to start training our models we need to prepare the images in a way that it will be easier for the model to learn its features. To do this we apply some data altering techniques that will make the sarcomeres stand out and make the image resemble our target binary mask. We use the technique developed by [8] which is used for creating the binary mask of our microscopic images. This technique reaches an accuracy of detected sarcomeres of 80%. We will now explain the steps taken in a precise matter.

We execute the following operations using software called ImageJ. ImageJ is an

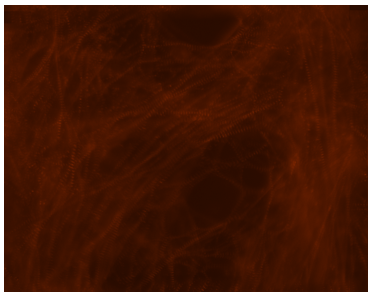

Figure 3.1: Original image

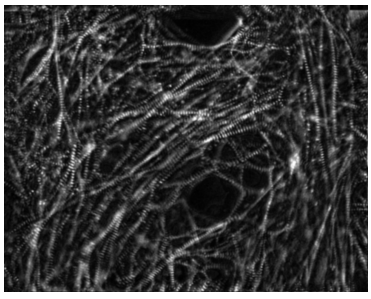

Figure 3.2: Preprocessed

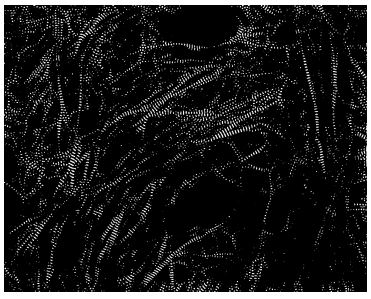

Figure 3.3: Binary mask

open source scientific multidimensional image processing tool[14]. First, we turn the image into an 8-bit image, this means that there is less variation within the image colour wise. Then we redefine the spatial scale of the image so measurement results will be calibrated in units of pixels. Second, we perform subtract background operation. This technique tries to correct uneven illuminated background by using a rolling ball algorithm [15]. The final operation we execute is an enhance contrast process. This operation recalculates the pixel values of the image so the range is equal to the maximum range. The range in this instance is 0-255 since we're dealing with 8-bit images.

3.3 Data augmentation

Data augmentation is a established technique to solve class imbalance or in instances where there is not a lot of data. In our experiments we noticed that the results given by the segmentation algorithms were over fitting to some extent. So we began exploring data augmentation techniques.

In [20], a review was given of multiple techniques and there effectiveness. We settled on using three simple data augmentation algorithms in order to enhance our data. It is important that the ground truth images also have this technique applied to them. The first technique is horizontal flip. This technique is quite straightforward. We gather an image and then flip it along the y-axis. The second technique is rotation. We rotate the image by a random degree. Our third and final technique is adding random noise to the image. An original image is distorted by random pixel values put in the image. We do not apply this technique to the ground truth of course. In figure below you can see the original image and an application of the augmentation. We expect that our augmentations have a positive effect on the effectiveness of our models.

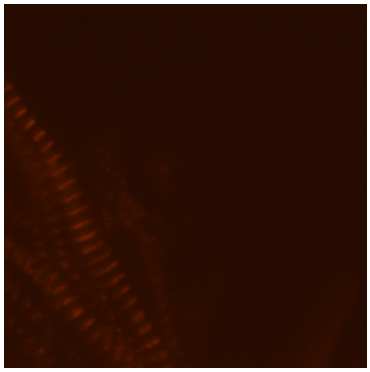

Figure 3.4: Original image

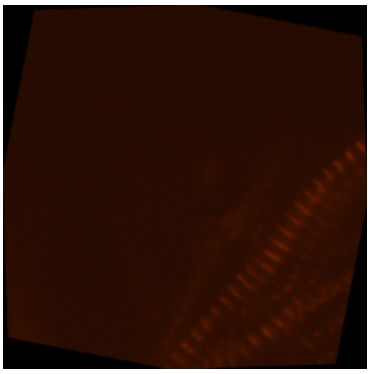

Figure 3.5: Flipped and rotated

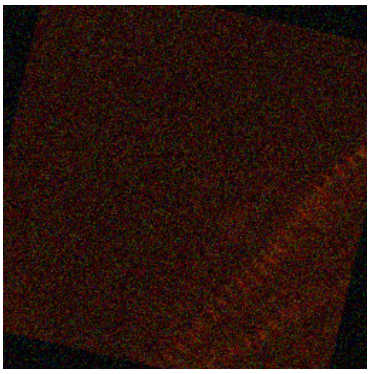

Figure 3.6: Rotated and added noise

Chapter 4

Methods

4.1 Instance Segmentation

Instance segmentation [3] is the process of identifying separate objects at pixel level within an image. This task is one of the hardest problems to solve in the computer vision field. There are four different types of classification that can be done on an image as shown in 4.1.

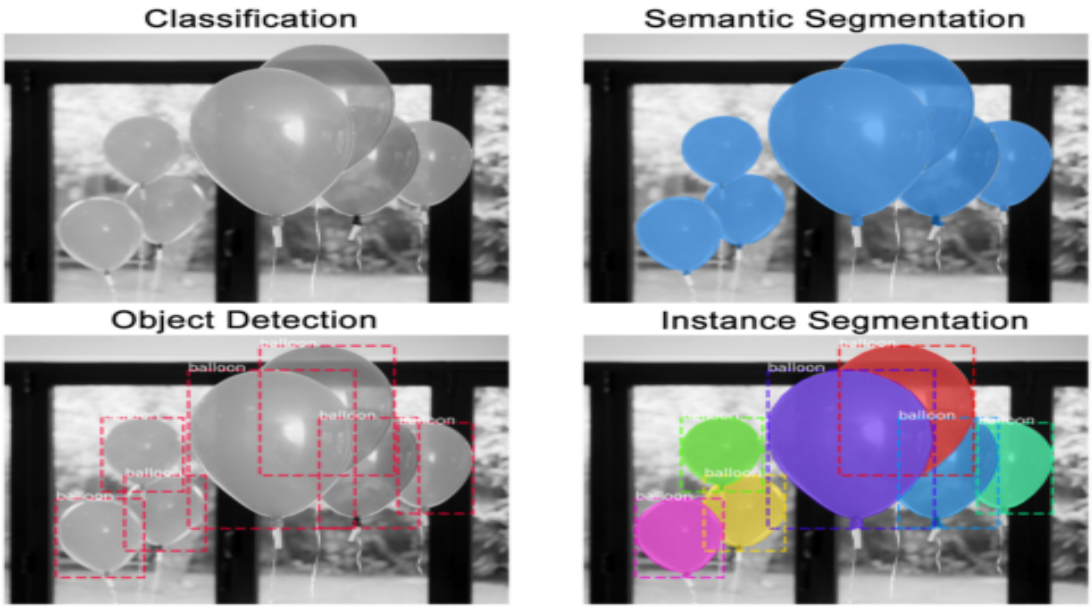

Figure 4.1: Different types of image classification

The first image concerns classification. An image classification algorithm can determine whether there is a certain object in the image. In our example a classification algorithm would try to predict whether there is a balloon in the image. The second type of classification is semantic segmentation. An algorithm that tries to solve this problem determines for each individual pixel to which class it belongs. The third image is an example of object detection. This algorithm can identify a bounding box around the objects its detecting within an image. The last image deals with instance segmentation. This technique can detect multiple instances of an object within an image and classify pixels belonging to that specific object.

4.2 Mask-RCNN

Mask RCNN is the first instance segmentation method we experimented on. Mask RCNN is a convolutional neural network (CNN) and one of the most modern techniques in the realm of image segmentation. This algorithm detects objects in an image and is capable of generating superb quality segmentation masks for each instance. Mask RCNN is capable of separating multiple classes of objects in an image but in our use case we have only one object namely the sarcomeres but in general Mask RCNN is capable of detecting multiple objects.

The way Mask RCNN works is that it uses a two stage approach. First, it identifies regions of interest within the image of a certain size. This means that there might be an object located in this region. Further, it predicts the class of that object within the region of interest. The bounding box gets refined and a pixel level mask gets generated based on the object and the first stage region of interest. Both stages are connected to the backbone structure. The backbone extracts features from the image, for our research we have chosen the backbone ResNet which was promoted as being the most accurate.

Mask RCNN uses a feature called anchor boxes. Anchor boxes are a set of predefined bounding boxes. These anchor boxes are used to quickly identify certain objects in an image. For our image the average size of a sarcomere in pixels can range from 4x4 to 10x10 pixels. This means that looking for bigger objects would be a waste of time and computing power. You can define anchor boxes by scale an aspect ratio depending on the type of objects you want to detect. This is just one example of the many hyperparameters that Mask RCNN offers. Unlike other similar models like YOLOv4, Mask RCNN offers a great deal of freedom when it comes to hyperparameter tuning. In turn, to do this correctly you need to be an expert in the field of artificial intelligence. For this project Mask RCNN is relevant since it has a relatively fast training time. Also, it is an open source project that offers examples of other users' code which makes it trivial to implement on our own custom data set. The low training time is especially useful when we are experimenting with hyperparameter tuning. Hyperparameter tuning is the task of configuring the model's option in such a way that the paramaters are refined to our specific data set. Unlike our second instance segmentation model Mask RCNN offers a great deal of freedom.

4.3 YOLOv4

YOLOv4 is a continuation of YOLOv3 but done by different authors. Yolo stands for: "You only look once" because YOLOv4 uses a one stage approach as opposed to

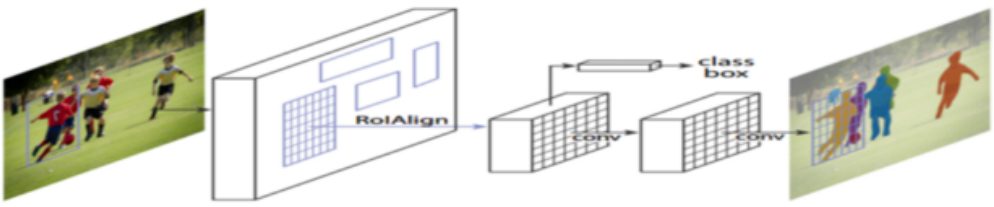

Figure 4.2: Mask RCNN framework for instance segmentation

Mask RCNN's two stage approach. The premise of one stage detection is that you only look at the image once. YOLOv4 goes through the image in a sliding window and classifies the objects within that windows as seen in figure 4.3.

There were two big changes made from YOLOv3 to YOLOv4. First, the model can be trained on a single GPU. This was relevant for this project since we did not have access to a kernel that could do the computation. The second change was that a specific set of techniques is applied to the training data in order to make the model more robust. In the paper these techniques are referred to as "Bag-of-freebies" and "Bag-of-specials". Bag-of-freebies refers to the data augmentation techniques that are applied to the training data. The data augmentation techniques are based on photometric distortions for example adding noise to the image or geometric distortions which entail flipping or rotating the image. Bag of specials refer to post-processing methods that add some inference time. Inference time is the time it takes the model to detect all instances of the of every class in an object. So with these bag-of-special techniques inference time goes up slightly but accuracy improves significantly.

We have chosen to use YOLOv4 because this project has been used in different fields in other projects.

4.4 Semantic Segmentation

The goal of semantic segmentation is to classify each pixel in an image to the correct class. This is also referred to as dense prediction. Dense prediction means that every individual pixel is getting classified using the algorithm's learned rules. Another example of a dense prediction algorithm is YOLOv4. If there are only two classes like in our case, the sarcomeres and the background, then the output of the prediction will be a binary mask.

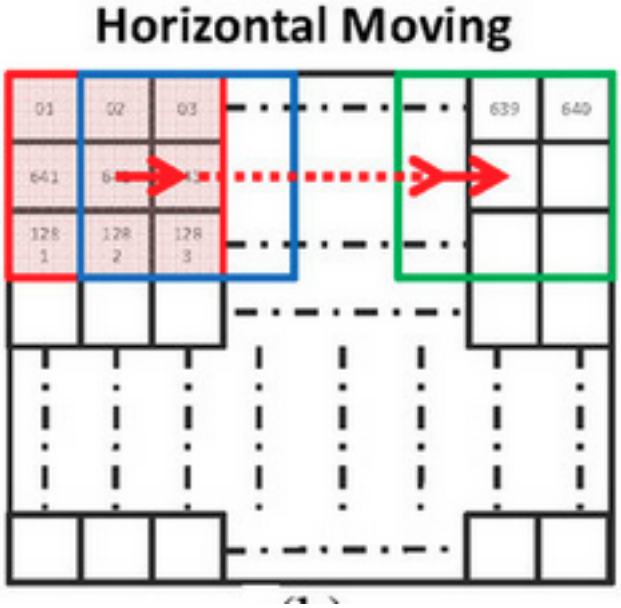

Figure 4.3: Sliding window schematic

4.5 U-Net

U-Net is our first semantic segmentation algorithm. The main idea behind any convolutional neural network is to learn the feature mappings of an image and find the most expressive features that characterize an object. Semantic segmentation is different from normal classification since we need to convert the original feature mapping of the image to a new feature mapping of an image. This problem is a lot harder and the main problem that U-Net tries to solve. The developers of U-Net then figured that when we convert the image into a vector we are learning the feature mapping. So when it is time to classify the pixels we already learned the features of the image so we can reuse this knowledge. This feature helps the image retain its structural integrity and reduce distortion.

We choose to experiment using U-Net because of its model which is intuitive and has seen very good results on well-known data sets. Also the model is quite fast to train and inference speed is also very high. We do not have to provide a lot of data to get these results which is another benefit. Overall we expect U-Net to perform very well on our data set.

4.6 SegNet

SegNet is our second semantic segmentation algorithm. We will use this model to compare it to the performance of U-Net. We have chosen this model for its simplicity to implement and also because it is a more generalized image segmentation algorithm. This architecture makes use of an encoder-decoder structure. That means that we have a general encoder network followed by a decoder network. The encoding takes place within a pre-trained classification network. In our case this will be the VGG network. The task of the decoder is then to predict output segmentation using the encoders encoding.

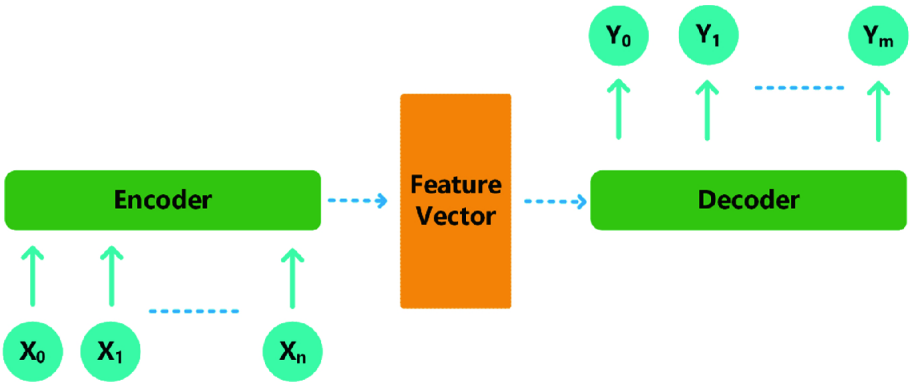

Figure 4.4: Encoder-decoder structure

How this works specifically is as follows. The encoder compresses the input image into feature vector representation. This feature vector essentially holds all the information that is used to for predicting the output. The decoder takes this vector and tries to predict the output. The model is trained using the differences between the output and the given ground truths.

Chapter 5

Results

5.1 Results

We will now present the results from our different methods. In this section we will discuss our hardware environment that the experiments executed on. After that we will explain the evaluation techniques that we apply to judge the effectiveness of our models. In the final part of this chapter we will try to explain the outcomes of our models and how we could make advancements in the future. In table 5.2 we present the settings for our experiments for the semantic segmentation algorithms. In 5.1 we present the settings for our experiments for the instance segmentation algorithms.

5.2 environment settings

Most of the models, preprocessing and evaluation code is written in Python 3.8.0 with TensorFlow 2.4.1 and Keras 2.4.0 libraries for deep learning. YOLOv4 is build from its C++ source. The testing platforms specifications are Linux-Ubuntu 20.04 (64-bit) with an Intel Core i5-9600K at 3.7Ghz, 16 GB of RAM and a Nvidia GeForce RTX 2060 Super with 8 GB of dedicated memory. We used the latest Nvidia drivers version 471.68. Requirements for TensorFlow and Keras are Cuda and CUdnn which are libraries written for API use of the graphics card driver. On this machine we have installed Cuda 11.1 and CUdnn 8.1.

Table 5.1: Overview of the settings for our experiments

Model	**Augmentation**	**Preprocessing**	**Epochs**	**Batch Size**	**256x256**
Mask-RCNN	No	No	50	2	Yes
	Yes	Yes	50	2	Yes
YOLOv4	No	No	8	2	Yes
	Yes	Yes	8	2	Yes

Table 5.2: Overview of the settings for our experiments

Model	Augmentation	Preprocessing	Epochs	Batch Size	256x256	512x512
U-Net	No	No	60	2	Yes	Yes
	Yes	No	60	2	Yes	Yes
	No	Yes	60	2	Yes	Yes
	Yes	Yes	60	2	Yes	Yes
SegNet	No	No	60	2	Yes	Yes
	Yes	No	60	2	Yes	Yes
	No	Yes	60	2	Yes	Yes
	Yes	Yes	60	2	Yes	Yes

5.3 Measurement techniques

Evaluation plays a pivotal role in determining whether a model performs well enough to be considered for real applications. We have chosen to evaluate our model using five different statistical analysis methods. All of the following evaluation techniques have been chosen because they are what is commonly used when measuring the effect of a model in the computer vision field. We are missing a true accuracy report this is because after enough epochs the accuracy according to the model was in most cases close to one. This is probably due to amount of background pixels it correctly predicted but not a very informative measure in any other aspect.

Our first evaluation method is Intersection over Union (IoU). This method divides the area of overlap of both segmentation masks by the area of union of both segmentation masks. We get a high IoU score if our segmentation mask lines up correctly with our ground truth. The max value of IoU is 1, this means that our predicted mask and ground truth are the same. In our next formula's we will call area of union by true positive (TP) and area of overlap by true positive plus false positive plus true negative (TP + FP + TN).

$$\text{Intersection over Union} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{5.1}$$

Our next equation is called Dice Coefficient. the calculation for Dice and IoU might look very similar but they serve a different purpose [1]. The problem and difference between these two metrics is shown when we take the average score over a set of predictions. Subsequently, the difference between the two formula's appears when assessing how much worse model A is than model B for any experiment.

$$\text{Dice Coefficient} = \frac{\text{TP} + \text{TP}}{\text{TP} + \text{TP} + \text{FP} + \text{TN}} \tag{5.2}$$

According to [17] Precision, Recall, and F-Score are biased and don't consider the level of chance. They argue that if there is some class imbalance within a data set that these measures don't accurately inform us of the models' performance. Once we have presented the results we will discuss their findings further and see if they are correct. We assert that a predicted sarcomere is a true positive if it has at least 50% overlap with another sarcomere in the ground truth. A false positive is when

a predicted sarcomere does not have 50% overlap with a sarcomere in the ground truth. A false negative is when there is no corresponding sarcomere in the predicted image with more than 50% overlap in the predicted binary mask.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.4}$$

$$\text{F-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

5.4 Instance segmentation

Our instance segmentation experiments did not go as we had expected. The results are lackluster and do not warrant any further research. We will explain why we are not further studying instance segmentation in the discussion section. Image 5.1 shows the predicted bounding boxes made by the YOLOv4 model. In the image we notice two important elements. The first is that the bounding box does not follow the shape of the sarcomere. The bounding box is a rectangular shape and does not track the edges of the sarcomeres. The second element of poor performance is in the amount of sarcomeres it predicts. YOLOv4 had 50% mean accurate precision. So from the test set we gave it YOLOv4 only accurately detected half of the sarcomeres in the test set. This fact together with the inaccurate bounding boxes makes YOLOv4 not viable for our research.

The result of Mask RCNN is pictured in 5.2. Mask RCNN performs very poor on our data set. The sarcomeres it predicts are very small that is why you do not see a bounding box. Mask RCNN has a number of paramaters but it is not clear how to tune these values for our data set. If we had better knowledge of the model it might have performed better. But it is evident that this model can not accurately predict sarcomeres with the data we have fed into it and with the paramaters we have set.

5.5 Semantic segmentation

Our experiments on images of size 256x256 can be found in table 5.3. Our best results were found using U-Net's algorithm combined with our preprocessing techniques and augmentation. The results of U-Net without augmentation and preprocessing is virtually the same as with these two changes. We can infer from this that preprocessing is needed to extract a binary mask from a rule based algorithm but preprocessing is not needed when we're feeding the data into an artificial intelligence model. U-Net reached a maximum IoU score of 0.67. This is not very high when we factor in that the rule based method we're comparing against reaches an accuracy of 80%. In reality, the accuracy will presumably be lower.

Segnet has the best results when there is no preprocessing and no augmentation done. Its best IoU score is 0.22. Interestingly is that Segnet performed considerably worse on the prepossessed images. One thing we found through research is that Segnet performs poorly when the objects to find are very small.

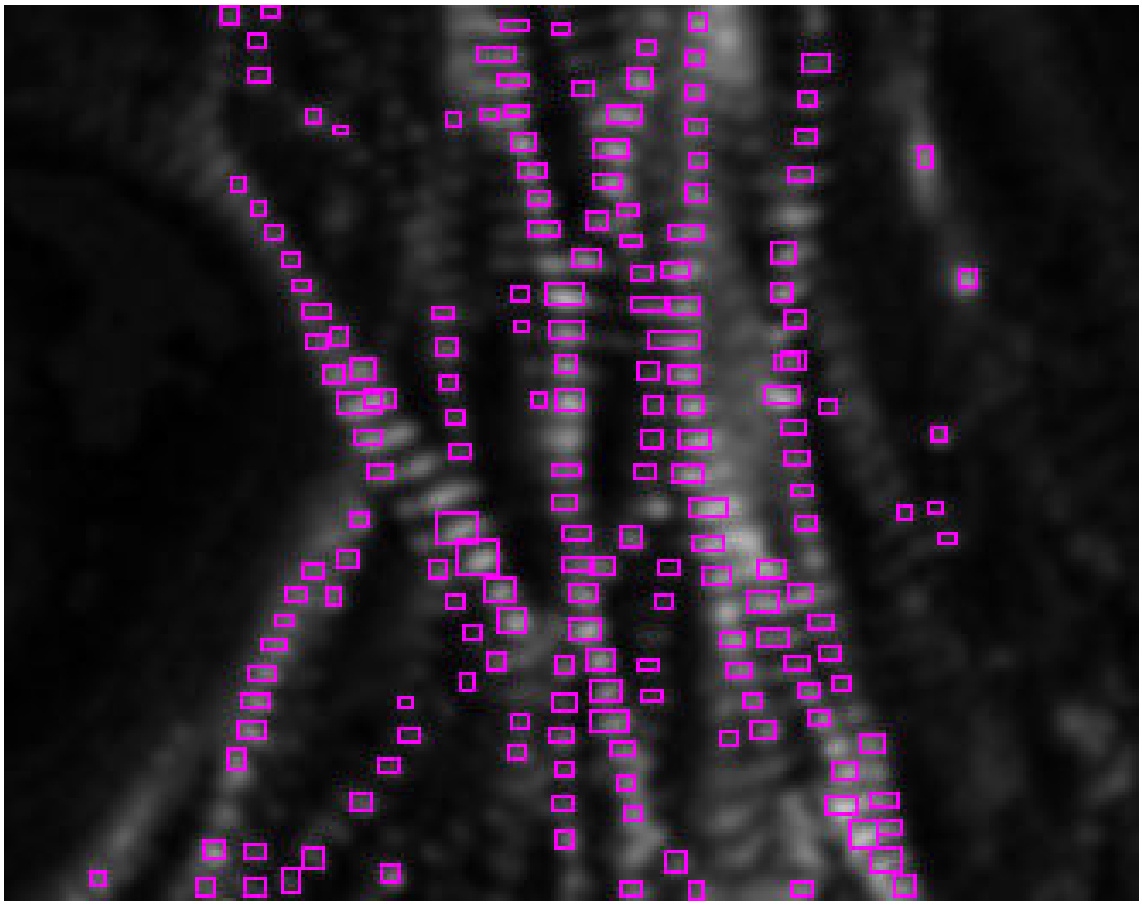

Figure 5.1: Original image

Table 5.3: Results of the semantic segmentation models for 256x256 images

Model	Augmentation	Preprocessing	IoU	Dice	Precision	Recall	F-Score
U-Net	No	No	0.66	0.79	0.97	0.90	**0.93**
	Yes	No	0.54	0.70	**1.0**	0.79	0.88
	No	Yes	0.46	0.57	0.59	0.88	0.64
	Yes	Yes	**0.67**	**0.80**	0.95	**0.90**	0.92
SegNet	No	No	0.22	0.35	0.93	0.57	0.69
	Yes	No	0.13	0.22	0.94	0.43	0.59
	No	Yes	0.0	0.0	1.0	0.0	0.1
	Yes	Yes	0.05	0.09	0.92	0.26	0.40

There is a big difference in the measurement scores between 256x256 images and 512x512 images. Almost every score for both models improve significantly with 512x512 images. U-Net's IoU for preprocessing and augmentation increase from 0.67 to .81. That is a 21% increase. If we can take anything away from this study it is that when feeding images into a deep learning algorithm we should experiment on different image sizes.

In SegNet and U-Net the results show that the precision score is very high in every category. Yet recall is low in SegNet and high in U-Net. This has to do with

Figure 5.2: Original image

the fact that we have two classes and a serious class imbalance. If the model predicts for a particular pixel that it is not part of a sarcomere it has about 90% chance that it is accurate. So we won't see many false positives or false negatives.

Table 5.4: Results of the semantic segmentation models for 512x512 images

Model	**Augmentation**	**Preprocessing**	**IoU**	**Dice**	**Precision**	**Recall**	**F-Score**
U-Net	No	No	0.73	0.84	0.98	0.95	0.96
	Yes	No	0.65	0.79	0.99	0.89	0.94
	No	Yes	0.76	0.86	0.94	0.96	0.95
	Yes	Yes	**0.81**	**0.90**	**0.99**	**0.98**	**0.99**
SegNet	No	No	0.32	0.49	0.97	0.83	0.89
	Yes	No	0.04	0.09	0.95	0.41	0.56
	No	Yes	0.39	0.56	0.98	0.91	0.95
	Yes	Yes	0.0	0.01	1.0	0.22	0.04

In figure 5.3 to 5.6 we see an example of input, output and overlap of U-Net's best performing setup for 512x512 sized images. In 5.6 red is used to denote the ground truth and blue for the predictions made by U-Net. The purple area shows

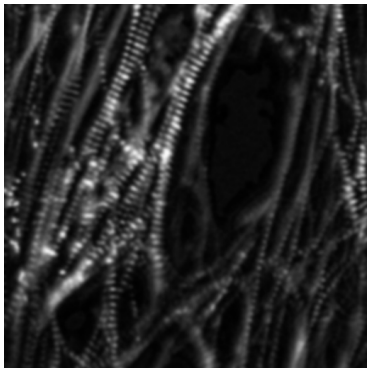

Figure 5.3: Original image

Figure 5.4: Ground truth

Figure 5.5: Predicted mask

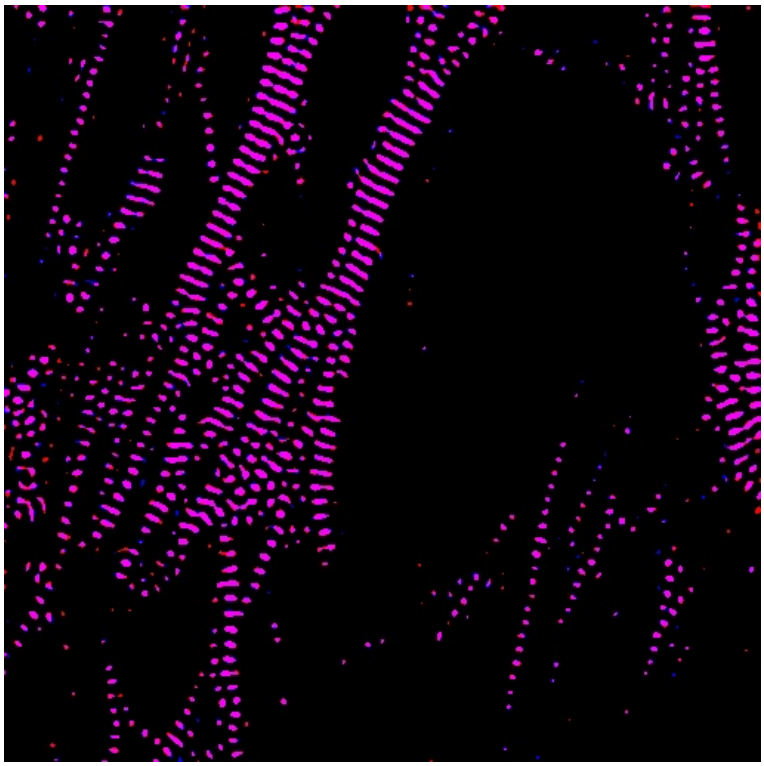

Figure 5.6: Overlap image

the overlap between these two images. It is clear from this image that U-Net is capable of highlighting most of the ground truth's sarcomeres.

In figure 5.7 to 5.10 we show the results of our best performing SegNet experiment on 512x512 sized images. In 5.10 you see the overlap of the ground truth and our predicted mask. The colour red denotes the ground truth and the colour blue indicate our predictions. Their overlap makes purple just like in figure 5.6. What is interesting in SegNet's prediction is that every corner is a hard one. The sarcomeres have a long oval shape yet SegNet uses strict corners as cutoff point for its predictions. This causes the algorithm to miss a lot of the sarcomeres true size and shape when predicting. We see in the overlap that SegNet is quite capable of finding the location of the sarcomeres but not their size and shape.

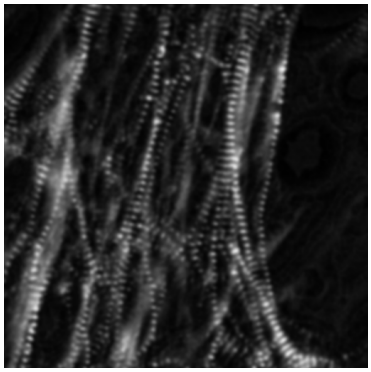

Figure 5.7: Original image

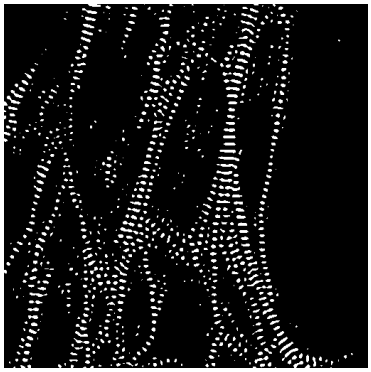

Figure 5.8: Ground truth

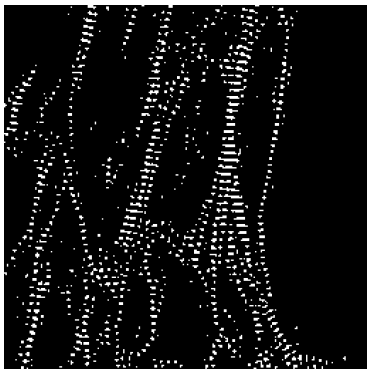

Figure 5.9: Predicted mask

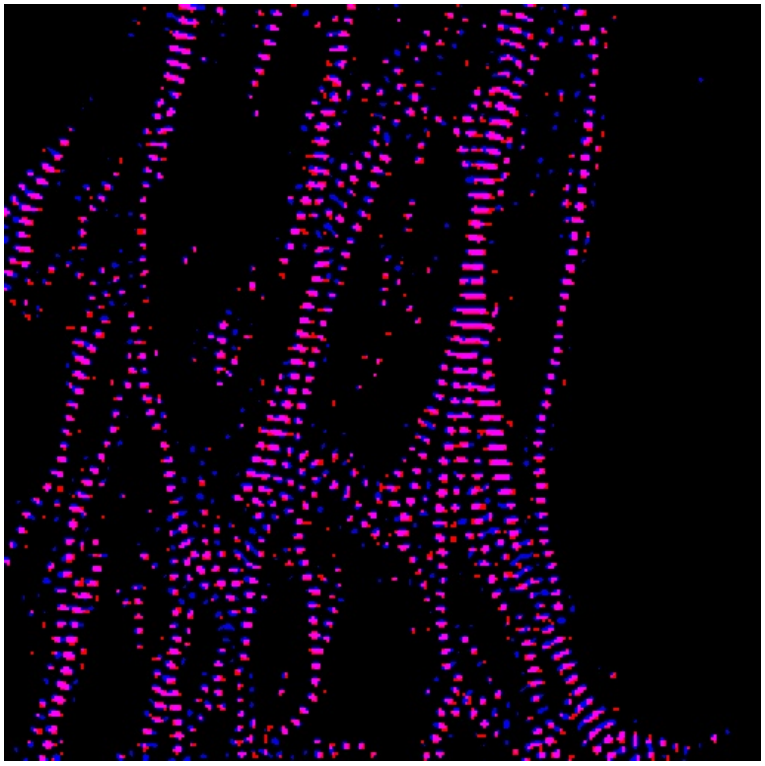

Figure 5.10: Overlap image

5.6 Discussion

We will be discussing some areas of improvement and other aspects of the research that could have an impact on the results.

Regarding the segmentation models there are four elements that we would like to discuss. The first is their performance. The best score that we have achieved is using U-Net with augmentation and preprocessing techniques applied to the data set. Although this is barely better than U-Net with no augmentation or preprocessing. We can infer from this that the feature maps must have been quite similar or that U-Net learned the features of the image with minimal use of the difference in colours. Also the contrast in 256x256 images compared to 512x512 images is quite stark. It seems that the feature maps of 512x512 images could be learned better than the 256x256 images. This might be due to that the original U-Net implementation used

a training set of 512x512 images and was optimized for this purpose.

Furthermore the difference in performance between U-Net and SegNet is significant. It is clear that SegNet is unable to learn the features of our images correctly. An interesting aspect of SegNet is that it also performs better on 512x512 images. We can conclude from this that any further experimentation with SegNet in our current experiment setup is not necessary. Perhaps we could have made some changes in the augmentation or preprocessing process to correct for SegNet's faults. Another thing we would like to try in the future is transfer learning. Transfer learning is initializing some of the networks nodes by a value learned from a different data set. That data set does not have to fall within the domain of our data set. Transfer learning is best used when we do not have much data. We might benefit from transfer learning when doing instance segmentation.

Thirdly, the problem of instance segmentation within our experiments. We can conclude from our experiments that our setup for Mask-RCNN and YOLOv4 does not produce very good results. We can think of two reasons why this might be the case. The first being that we do not have enough training data. Our training data consisted of 120 images that we split up in 256x256 squares. It could be that these models need thousands of images to correctly identify instances of objects within an image. Secondly, it could be the case that we did not properly tune the hyperparameters of Mask-RCNN and YOLOv4.

Another interesting thing to note is that since we have two classes or objects we're trying to detect. Namely the background and the sarcomeres. That means that a binary segmentation and a correct instance segmentation map are functionally the same. This would not hold up if we have overlapping objects in our image but we do not thus we can come to this conclusion. So with that in mind it makes no sense to further do research into instance segmentation models if semantic segmentation models have better results.

Lastly, we present some areas we could have done more research on and would like to improve upon in further research. The first area is image preprocessing. In the ground truth there are some areas of less than 5 pixels in size. We can assume that a sarcomere is never less than 5 pixels in size and should have filtered out these parts of the ground truth. This can also be a part of post processing in the future. Secondly, we can experiment using transfer learning. Transfer learning is a powerful tool where some parts of the neural network have been trained on a different data set for example ImageNet[9]. Implementing this could have an impact on our accuracy.

Furthermore the manner in which we measure our results could be improved upon. Since we compare the models' results against the rule based algorithm we can not be sure of the real evaluation. So in the future we would like to have a small set of manually delineated sarcomeres that we can compare our results against.

Chapter 6

Conclusion

As we find out more about anti-cancer drugs that are cardiotoxic we will need an efficient way of analyzing these drugs such that these problems can be found before clinical trials. To do this we will need a way of measuring the impact of these drugs in cardiomyocytes. Our research aims to find a deep learning model to segment microscopic images of sarcomere structures in order to follow the effect of anti-cancer drugs on sarcomeres. In this paper we present and analyze four deep learning models for instance segmentation and semantic segmentation. For both methods we have chosen two models that have seen much success in past applications and compared them to each other. We found that instance segmentation models could not properly locate enough sarcomeres for them to be effective. On the other hand semantic segmentation models predicted binary masks that were very close to their ground truth. Especially U-Net gave us very good results. In the future we could use a more refined U-Net model with more data and different preprocessing rules and augmentation techniques to get a better result. Furthermore we might try testing our deep learning models against a small set of manually delineated sarcomeres as ground truth.

Bibliography

[1] willem (https://stats.stackexchange.com/users/159052/willem). *F1/Dice-Score vs IoU*. Cross Validated. URL:https://stats.stackexchange.com/q/276144 (version: 2017-11-13). eprint: https://stats.stackexchange.com/q/276144. URL: https://stats.stackexchange.com/q/276144.

[2] Waleed Abdulla. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*. https://github.com/matterport/Mask_RCNN. 2017.

[3] Waleed Abdulla. *Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow*. Dec. 2018. URL: https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46.

[4] Saleh Albahli et al. "Melanoma Lesion Detection and Segmentation Using YOLOv4-DarkNet and Active Contour". eng. In: *IEEE access* 8 (2020), pp. 198403–198414. ISSN: 2169-3536.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *CoRR* abs/1511.00561 (2015). arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561.

[6] Fatemeh Bagheri, Mohammad Jafar Tarokh, and Majid Ziaratban. "Skin lesion segmentation based on mask RCNN , Multi Atrous Full-CNN , and a geodesic method". eng. In: *International journal of imaging systems and technology* (2021). ISSN: 0899-9457.

[7] Juan C. Caicedo et al. "Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl". In: *Nature Methods* 16.12 (Dec. 2019), pp. 1247–1253. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0612-7. URL: https://doi.org/10.1038/s41592-019-0612-7.

[8] Lu Cao et al. "Automated image analysis system for studying cardiotoxicity in human pluripotent stem cell-Derived cardiomyocytes". eng. In: *BMC bioinformatics* 21.1 (2020), pp. 187–187. ISSN: 1471-2105.

[9] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[10] Kaiming He et al. *Mask R-CNN*. cite arxiv:1703.06870Comment: open source; appendix on more results. 2017. URL: http://arxiv.org/abs/1703.06870.

[11] Feixiao Long. "Microscopy cell nuclei segmentation with enhanced U-Net". eng. In: *BMC bioinformatics* 21.1 (2020), pp. 8–8.

[12] Alexander Selvikvåg Lundervold and Arvid Lundervold. "An overview of deep learning in medical imaging focusing on MRI". In: *Zeitschrift für Medizinische Physik* 29.2 (2019). Special Issue: Deep Learning in Medical Physics, pp. 102–127. ISSN: 0939-3889. DOI: https://doi.org/10.1016/j.zemedi.2018.11.002. URL: https://www.sciencedirect.com/science/article/pii/S0939388918301181.

[13] Rangaraj M Rangayyan. *Biomedical image analysis*. CRC press, 2004.

[14] W.S. Rasband. *ImageJ*. 1997. URL: https://imagej.nih.gov/ij/.

[15] *Rolling Ball Background Subtraction*. https://imagej.net/plugins/rolling-ball-background-subtraction. Accessed: 2021-08-15.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[17] Abdel Aziz Taha, Allan Hanbury, and Oscar Jimenez-del-Toro. "A Formal Method For Selecting Evaluation Metrics for Image Segmentation". In: Oct. 2014. DOI: 10.1109/ICIP.2014.7025187.

[18] Pedro Vianna, Ricardo Farias, and Wagner Coelho de Albuquerque Pereira. "U-Net and SegNet performances on lesion segmentation of breast ultrasonography images". eng. In: *Research on biomedical engineering* 37.2 (2021), pp. 171–179. ISSN: 2446-4732.

[19] Aarno Oskar Vuola, Saad Ullah Akram, and Juho Kannala. "Mask-RCNN and U-Net Ensembled for Nuclei Segmentation". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 208–212. DOI: 10.1109/ISBI.2019.8759574.

[20] Qingsong Wen et al. *Time Series Data Augmentation for Deep Learning: A Survey*. 2021. arXiv: 2002.12478 [cs.LG].

[21] Aditya U Kale Xiaoxuan Liu Livia Faes. "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis". In: *The lancet* 1.6 (2019). DOI: https://doi.org/10.1016/S2589-7500(19)30123-2.