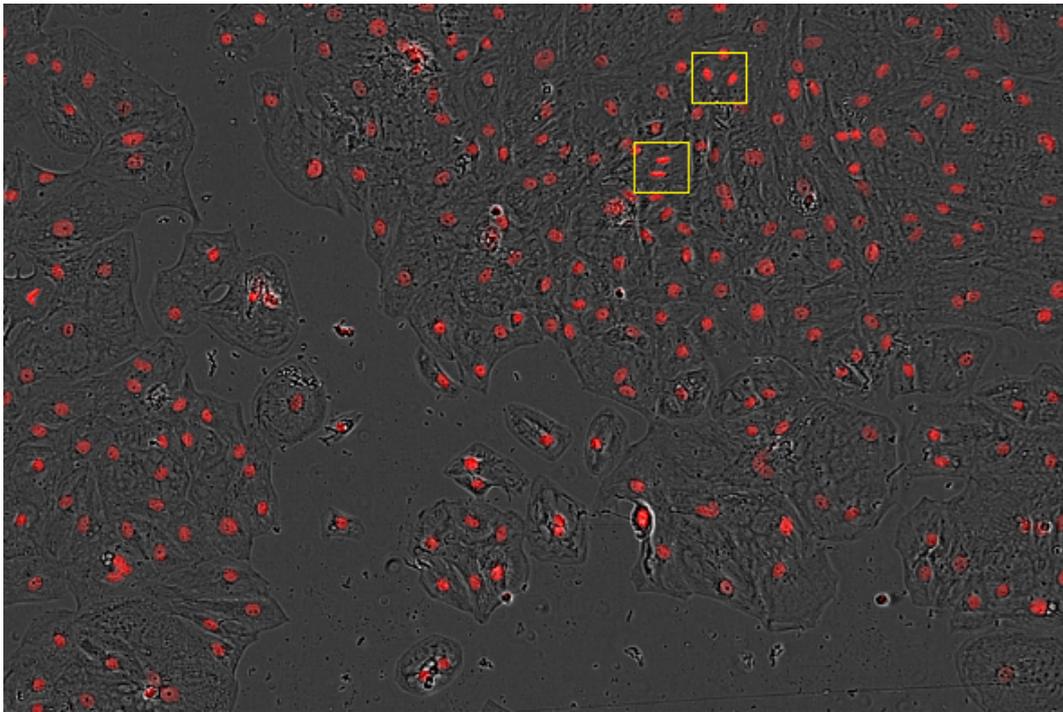




CHALMERS
UNIVERSITY OF TECHNOLOGY



Detection of Cardiomyocyte Proliferation using Convolutional Neural Networks

Master's thesis in Biomedical Engineering

Lina Olandersson and Susanna Larsson

MASTER'S THESIS 2019

Detection of Cardiomyocyte Proliferation using Convolutional Neural Networks

Lina Olandersson
Susanna Larsson



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Biomedical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Detection of Cardiomyocyte Proliferation using Convolutional Neural Networks
LINA OLANDERSSON AND SUSANNA LARSSON

© LINA OLANDERSSON AND SUSANNA LARSSON, 2019.

Supervisor: Dr. Johan Karlsson, AstraZeneca

Examiner: Dr. Fredrik Kahl, Department of Electrical Engineering

Master's Thesis 2019:NN

Department of Electrical Engineering

Division of Biomedical Engineering

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

Cover: Bright field microscope image of cardiomyocytes with an overlaying fluorescent channel. Two cell divisions are marked by the yellow squares.

Typeset in L^AT_EX

Gothenburg, Sweden 2019

Abstract

Cardiovascular diseases are one of the major health issues of today. A myocardial infarction can cause heart muscle cells to die due to oxygen deficiency. The heart cells have no capability of regeneration, which means that the dead cells will not be replaced by new ones. The pharmaceutical company AstraZeneca is searching for a compound that can induce cell divisions of heart cells. The aim of this thesis is to investigate a method of measuring cell division rate. Time sequences of 2D microscope images are analyzed to detect cell divisions using a convolutional neural network. First, all sequences were processed by image registration and normalization. Then, a training set was generated by manual annotation. Data augmentation was used to increase the number of training examples of cell divisions, due to that cell divisions are rare. Mitosis takes several hours and is hard to detect on a single image. To make use of the temporal information, three images from the time sequence were used as input on multiple channels. The convolutional neural network learned to distinguish between two classes, cell division and background, after training on patches of size 60x60 pixels and three time steps. A sliding window was used to classify a full time sequence.

The model learned to detect cell division in a fluorescent channel with a recall of 83% and in a bright field channel with a recall of 48%. A problem with the model detecting false positives was encountered. It was to some extent solved by active learning and using ensembles of models. To be able to measure the true cell division rate accurately, the number of false positives should be lower than the number of true positives. The developed model cannot be applied to count cell divisions to a high enough accuracy. Despite that, the thesis shows that a convolutional neural network can detect cell divisions in bright field images, which are difficult to detect by a human eye. Furthermore, the use of ensembles and active learning to improve the results showed promising result and could be further developed to create a model that can be used in pharmaceutical research.

Keywords: cardiomyocyte, proliferation, CNN, deep learning, bright field, fluorescent, active learning, ensemble.

Acknowledgements

We would like to express our gratitude to our supervisor Dr. Johan Karlsson (Principal Data Scientist at AstraZeneca). Thank you for providing us with guidance and support during the project. Thank you, Dr. Erik Müllers (Associate Principal Scientist at AstraZeneca), for helping us growing cells for images and for answering all biological questions. We would also like to thank our examiner Dr. Fredrik Kahl at Chalmers. Thank you for your feedback and valuable suggestions when selecting a method for the project. Thank you, Johan Fredin Haslum, PhD student at AstraZeneca, for valuable ideas, support and discussions during the first half of the project. Thank you, Anna Kollberg and Rebecca Jonasson, master students at Chalmers University of Technology, for helpful feedback on our report. Finally, we would like to thank Daniel Fridolfsson, our opponent, for your feedback.

Lina Olandersson and Susanna Larsson, Gothenburg, June 2019

Contents

List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Objective	2
1.3 Aim	2
1.4 Limitations	2
1.5 Contribution to science	3
1.6 Thesis outline	3
2 Theory	5
2.1 Cardiomyocytes	5
2.1.1 Eukaryotic cell cycle	5
2.1.2 Cardiomyocyte cell cycle	6
2.1.3 Induce cardiomyocyte proliferation	7
2.1.4 Nuclear staining	8
2.2 State-of-the-art: Measuring proliferation	8
2.3 The principle of a neural network	9
2.3.1 Activation functions	10
2.3.2 Loss functions	11
2.3.3 Optimization methods	12
2.3.4 Evaluation of the model	13
2.3.5 Transfer learning and pre-trained networks	14
2.3.6 Convolutional Neural Networks	14
3 Method	17
3.1 Methods to find cell divisions	17
3.2 Production of image data	18
3.3 Image registration	21
3.4 Image annotation	21
3.5 Extracting training data	22
3.5.1 Image normalization	22
3.5.2 Cell division patches	23
3.5.3 Data augmentation	23
3.6 Convolutional neural network architecture	25

3.6.1	Network architecture with 3 channels	25
3.6.2	Network architecture with 5 channels	25
3.7	Detection of cardiomyocyte proliferation	27
3.7.1	CNN and sliding window	27
3.7.2	Convolutional object detection	28
3.7.3	Ensemble learning	28
3.8	Active learning	28
3.9	Specification of training parameters	29
3.9.1	Fluorescent images	29
3.9.2	Bright field images	30
4	Results and Evaluation	33
4.1	Annotated cell divisions	33
4.1.1	Side effects of nuclear staining	33
4.1.2	Cell growth factor FGF2	33
4.2	Fluorescent images	35
4.2.1	Detection of cardiomyocyte proliferation	35
4.3	Bright field images	40
4.3.1	Detection of cardiomyocyte proliferation	40
4.3.2	Ensemble learning	42
4.3.3	Comparison between ensemble and single model	44
4.4	Image set C	45
5	Analysis and Discussion	47
5.1	Network architecture and training	47
5.1.1	VGG16	47
5.1.2	Stabilization of time sequences	47
5.1.3	Filtering of images	49
5.1.4	Training with different images	50
5.1.5	More training examples	51
5.2	Measure of proliferation rate	51
5.2.1	Density	52
5.2.2	Border cases	52
5.3	False positives	52
5.3.1	Active learning	55
5.4	Ensemble learning	55
5.5	Limitations of the model	56
5.5.1	Finding dividing cells close to each other	56
5.5.2	Endoreduplication and acytocinetic mitosis	56
6	Conclusion	59

List of Figures

2.1	The eukaryotic cell cycle with four phases.	6
2.2	The five phases of mitosis, during which a mother cell is split into two daughter cells. The mitotic spindle, showed in green, takes form during prophase and resolves during telophase. The sister chromatids, showed in blue, split during anaphase and are enclosed by a nuclear envelope in telophase.	6
2.3	Variants of the adult cardiomyocyte cell cycle. A mother cell can double the number of chromosomes by endoreduplication (top) or double the number of nucleus by acytokinetic mitosis (bottom).	7
2.4	The architecture of a deep neural network with two hidden layers V_j and V_l . The inputs are denoted x_k and the outputs O_i . The connections between the layers have weights w	10
2.5	The ReLU activation function. The y-axis defines the mapping of the sum of the weighted inputs.	11
2.6	The squared error loss function, which describes the difference between target and output value. The global minimum indicated by the red mark corresponds to the desired coinciding of target and output. The black arrows explains the direction of a satisfactory updating with gradient descent.	12
2.7	The principle of a convolutional layer. The filter, visualized as the red square, slides over the input image. At each step, a convolution operation outputs a single value which is stored in a feature map.	15
3.1	Fluorescent microscope image with 10X magnification of cardiomyocytes with stained cell nuclei.	19
3.2	Bright field microscope image of cardiomyocytes with 10X magnification.	19
3.3	Mitosis of cardiomyocyte viewed by fluorescent images.	22
3.4	Mitosis of cardiomyocyte viewed by bright field images.	22
3.5	Augmentation by shifting the center of a cell division.	24

3.6	Input channels of the 5-channel network. The three middle channels corresponds to the original pre-trained VGG16 network. The initial weights of the added channels, New 1 and New 2, were copied from one of the original channels. The outermost channels were multiplied with a factor smaller than 1 to tone down their impact.	27
4.1	The number of cell divisions during the duration of a time sequence.	35
4.2	Cell division detection using convolutional object detection. A white square represents what is classified as cell division.	36
4.3	Red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C3 from A, slide 8 with threshold 0.5.	37
4.4	White pixels indicate predicted cell divisions when step size 1 is used. Well C3 from A, slide 8 with threshold 0.999.	37
4.5	The red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C7 from A, slide 74 with threshold 0.5.	38
4.6	White pixels indicate predicted cell divisions when step size 1 is used. Well C7 from A, slide 74 with threshold 0.999.	39
4.7	Red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C3 from A, slide number 8 with threshold 0.5.	41
4.8	White pixels indicate predicted cell divisions when step size 1 is used. Well C3 from A, slide number 8 with threshold 0.999.	41
4.9	Recall and precision of the models from network 1 (circles), network 2 (squares) and the two ensembles (red asterisks).	44
5.1	Rotation of an image in the time sequence causes false positive cell divisions.	48
5.2	Most of the false detections in fig. 5.1 are removed.	49
5.3	Variance filtered bright field image.	50
5.4	Example of images from time sequence C4 B, where an edge is classified as a cell division.	53
5.5	Example of images from time sequence C4 from B, where a dust particle is classified as a cell division.	53
5.6	Example of images from time sequence C4 B, where three cells are classified as a cell division.	54

5.7	Example of images from time sequence C3 B, where a group of cells are dying.	54
5.8	Example of a cardiomyocyte that possibly is undergoing acytokinetic mitosis.	57

List of Tables

2.1	Confusion matrix of the possible outcomes when classifying an example.	13
3.1	Specification of the number of cells in the wells for image set A. Cardiomyocytes in A7-A11 and B7-B11 are simulated with FGF2.	20
3.2	Specification of the number of cells in the wells for image set B. Cardiomyocytes in C13-C16 and D13-D16 are simulated with FGF2. Well C9-C12 and D9-D12 are excluded.	20
3.3	Specification of wells used to obtain image set C.	20
3.4	Modified VGG16 network architecture that takes an input of shape 60x60 with 3 channels and outputs two classes.	26
4.1	The number of cell divisions that was annotated inside each well can be found in the table. Divisions 1 are the number of initially annotated divisions and divisions 2 are new cell divisions found by the network that were missed during the manual annotation.	34
4.2	Number of cell divisions found in the test time sequences of fluorescent images. CNN model with accuracy of 98.88%.	39
4.3	Number of cell divisions found in the test time sequences of fluorescent images. CNN model with accuracy of 97.81%.	39
4.4	Number of cell divisions found in the test time sequences of bright field images by model FIRST.	40
4.5	The total number of false positives from two sets of networks evaluated on the same test sequences. Each network set contains four networks.	42
4.6	Results from ensemble evaluation of the test sets. Ensemble 1 has fewer false positives (fp 1) and found fewer cell divisions than ensemble 2.	43
4.7	Number of cell divisions found in the test time sequences of bright field images by ensemble 1.	43
4.8	Number of cell divisions found in the test time sequences of bright field images by ensemble 2.	44

1

Introduction

1.1 Background

Cardiovascular diseases are one of the major health issues of today. Yearly, more than 11 million people are affected by cardiovascular disease in Europe [1]. Today's more advanced healthcare system has led to a decreasing death rate. Patients who survive the acute phase often suffer from aftereffects, such as a decreased heart function. If the heart muscle is damaged, the cardiomyocytes who make up the muscular tissue in the heart have no capability of regeneration. This means that after for example a myocardial infarction, the dead cardiomyocytes will not be replaced by new ones [2]. The affected person has to live with a reduced cardiovascular function, which currently only can be treated with alleviative medication or heart transplantation.

A discovery of a curative medication would improve the quality of life significantly for patients with reduced cardiovascular function. The key target is to induce cardiomyocyte proliferation to enable a full recovery to the same number of heart muscle cells as before the injury. Proliferation is the process of cell division to increase the number of cells. The pharmaceutical company AstraZeneca is investigating different compounds to discover one that will stimulate cell division in adult cardiomyocytes. Recent research has shown promising results, such as induction of cell division by adding cell cycle regulators [3].

To determine how effective a compound is, the number of cell divisions needs to be counted and compared with a verification. Cell divisions can be observed using microscopes, both live and by using it to obtaining a time sequence of images. Both bright field images and fluorescent images can be taken with added DNA staining. Considering that a time sequence consists of a large number of images, it is time consuming to count all cell divisions manually. An automation of detection of cell divisions from the time sequences would therefore be valuable.

Machine learning models are increasing in popularity among data scientists in the industry. Object detection in images is one of many applications. Deep neural networks can be trained to recognize features in an image, such as yoga mats, bananas, tea bags or dividing cells. A cell division is not a single object in one image, but rather an event taking place during several images. Compared to standard object detection in 2D images, one challenge lies in making use of the temporal information.

Models to detect mitosis or track individual cells exist [4][5], but to the best of our knowledge there is no model that counts the number of cell divisions in a time sequence. Also, most models use fluorescent images whilst bright field images are preferable for cardiomyocytes. The reason is that the addition of staining can affect the normal behavior of the cells [6]. A model that can detect and count cell divisions from fluorescent images would be valuable for AstraZeneca in order to determine the efficiency of different proliferation stimuli. A model that can do the same using bright field images would be a major success.

1.2 Objective

The objective of this master's thesis work is to develop a deep learning model that automates the detection of dividing cardiomyocytes in a time sequence of 2D bright field microscope images.

1.3 Aim

In order to reach the objective of the thesis, the following milestones should be reached:

- Stabilize flickering time sequences of microscope images caused by camera movements.
- Develop a method for automatic/semi-automatic annotation of the images.
- Develop a neural network to detect and count the number of cell divisions in fluorescent images.
- Develop a neural network to detect and count the number of cell divisions in bright field images.

1.4 Limitations

The algorithm was developed using microscope images of cardiomyocytes, although there were also images of cardiac fibroblasts and cancer cells available. The images of the other cells are not very different from cardiomyocytes, especially regarding the fluorescent images. There might be a possibility that the network has the ability to generalize to the other cells, but the development was limited to using cardiomyocyte images.

Furthermore, the learning quality of the network is limited by the number of annotated training examples. Due to the time-consuming manual annotation, there was a limited number of examples.

1.5 Contribution to science

The main contribution of the thesis is that cell divisions are detected from bright field images. It shows that a convolutional neural network can be trained to detect cell divisions which are difficult to find for an untrained human eye. Although the model also classifies some non-dividing cells as cell divisions, the detection of cell division is not random.

The thesis also shows that the VGG16 model trained on ImageNet can be used for microscope data of cells. The weights of the model were used as a starting point in both a fluorescent channel and a bright field channel. The possibility to use a model trained on a different image set and then use transfer learning is a powerful tool in the area of machine learning.

1.6 Thesis outline

Chapter 1 - Introduction, explains the context of the project and highlights the importance of the topic. The research objective, limitations and the reports contribution to science are also presented.

Chapter 2 - Theory, explains the biology behind the project, state of the art in the field and the principle behind a neural network.

Chapter 3 - Method, the implemented solution is described. Starting with how the images were taken, then followed by processing and annotation of the images. Lastly, the neural network and how it was trained is explained.

Chapter 4 - Results and Evaluation, starts with introducing the results from fluorescent images and then describes the results of the model trained on bright field images.

Chapter 5 - Analysis and Discussion, explains possible causes of the outcome of the project and future possibilities.

Chapter 6 - Conclusion, highlights the outcome of the project in connection with the objective and research aim.

2

Theory

This chapter starts with a description of cardiomyocytes and the cell cycle. Then, different ways of measuring proliferation are presented. At the end, the principle of a neural network is explained.

2.1 Cardiomyocytes

The human heart is responsible for blood circulatory system. The heart wall surrounds the chambers and is divided into three separate layers [7]. The outermost layer is the epicardium, which is a protective sac that holds the heart in place. The middle layer is the myocardium which is the muscular layer. The muscle cells in the myocardium are called cardiomyocytes. Cardiomyocytes are approximately 100 μm long and 25 μm wide. Most of the cells contain one nucleus, but approximately one fourth contain two, three or four nuclei [8]. The cells are surrounded by connective tissue that is produced by cardiac fibroblast [9]. Coronary vessels supply the heart muscle with blood. It is the muscular layer of the heart that contracts to make blood flow through the body. Between myocardium and the chambers of the heart is the last layer, endocardium [7]. It is a thin layer that covers the inner surface of the heart chambers and is continuous with the endothelium of large blood vessels.

2.1.1 Eukaryotic cell cycle

A normal eukaryotic cell cycle consists of four phases, G_1 , S, G_2 and M which are illustrated in fig. 2.1 [7]. During G_1 , the cell leaves the resting phase and is growing in size to get ready for cell division. The synthesis of DNA takes place during the S phase. All chromosomes are replicated to create a pair of sister chromatids. This is followed by a second phase of cell growth, G_2 , where the cell prepares for mitosis. The last phase of the cell cycle is M, mitosis, where chromatids are separated and the mother cell is split into two daughter cells.

Mitosis consists of prophase, prometaphase, metaphase, anaphase and telophase which are illustrated in fig. 2.2 [7]. During prophase, DNA is folded into dense chromosomes where the two sister chromatids are joined in the middle. The mitotic spindle that will separate the two sister chromatids is also formed during prophase, and it is located at the poles of the cell. In prometaphase, the nuclear envelope is resolved and microtubules from the mitotic spider connects to the sister chromatids.

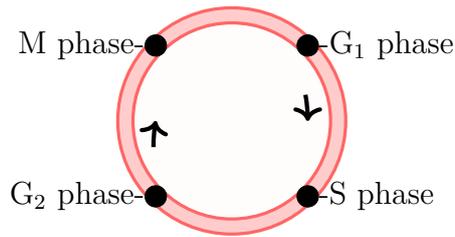


Figure 2.1: The eukaryotic cell cycle with four phases.

Metaphase is a lineup of all chromatids in an equatorial plane inside the cell. It is followed by anaphase, where the sister chromatids are split and pulled towards the poles of the cell. The final part of mitosis is telophase. Two nuclear envelopes are formed around each set of chromatids. The chromatids unfold and the mitotic spindle disassembles. The mother cell now has two nuclei and cytokinesis takes place, where the cell cytoplasm is split into two daughter cells.

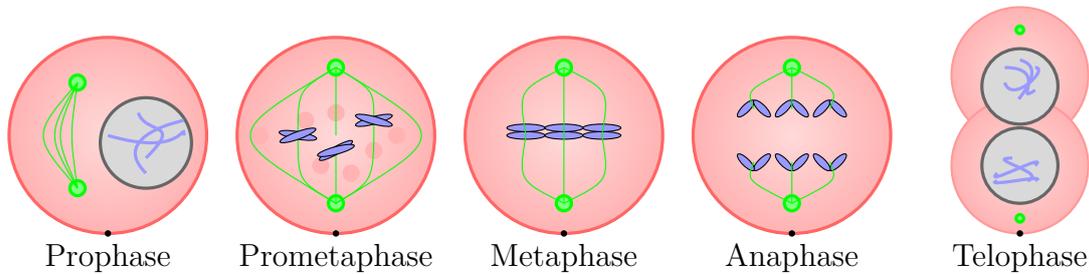


Figure 2.2: The five phases of mitosis, during which a mother cell is split into two daughter cells. The mitotic spindle, showed in green, takes form during prophase and resolves during telophase. The sister chromatids, showed in blue, split during anaphase and are enclosed by a nuclear envelope in telophase.

2.1.2 Cardiomyocyte cell cycle

Adult cardiomyocytes cannot enter the cell cycle to regenerate the heart tissue after an injury. Regeneration means that the heart reconstructs to the same number of cells and structure as before the injury. A study by Hesse et al. shows that the regeneration capability of cardiomyocytes depends on the developmental stage [2]. The developmental stages that differs in regeneration capability is embryo, neonatal and adult. Embryonic mammalian cardiomyocytes can regenerate after ablation (surgical removal) of cells [10]. Remaining cells enter the cell cycle for an increased proliferation rate to restore the cell number and structure of the heart. When followed-up in adulthood, the heart showed no dysfunction caused by the ablation.

Some of the regenerative capability is still active for neonatal heart tissue. Cardiomyocyte proliferation due to resection of the left ventricle has been observed in mice [11]. For mice up to seven days postnatal the heart was able to restore to normal function and anatomy.

The regenerative capacity is lost for adult cardiomyocytes as they enter a post-mitotic state. When injured, a collagenous scar tissue will replace the dead heart muscle and the number of cardiomyocytes will be permanently decreased. What happens next is a hypertrophic process where existing cardiomyocytes will increase in bulk but not in number. This will increase the contractile capacity of the heart, but it will not retain its old shape and structure [2].

Two cell cycle variants take place for adult cardiomyocytes: endoreduplication and acytokinetic mitosis, see fig. 2.3. A cell cycle variant means that the cell is not proceeding through the phases of mitosis as normal. During endoreduplication, the cell replicates the genome in S phase but is not entering the mitosis phase [12]. The result is a polyploid cell, which means that it has more than a double set of chromosomes. Acytokinetic mitosis results in two or more nuclei in the same cell [2]. That is due to the lack of cytokinesis at the end of the cell cycle.

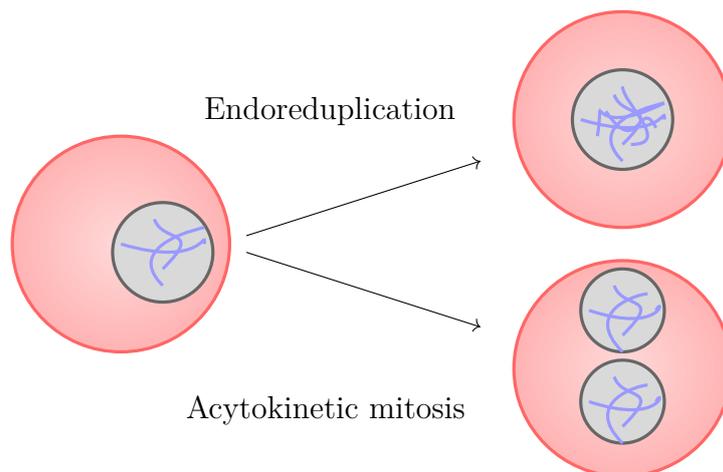


Figure 2.3: Variants of the adult cardiomyocyte cell cycle. A mother cell can double the number of chromosomes by endoreduplication (top) or double the number of nucleus by acytokinetic mitosis (bottom).

2.1.3 Induce cardiomyocyte proliferation

The idea of heart regeneration has driven scientists to explore different ways of inducing cardiomyocyte proliferation. Recently, Mohamed et al. published a study in *Cell* where a cocktail of cell cycle regulators was used to enable cardiomyocyte proliferation [13]. A combination of four regulators (CDK1, CCNB, CDK4 and CCND) was successful in culture dishes and also tested *in vivo* using a mouse model. The proliferation rate increased, and the infarct size was reduced when used after a myocardial infarction. What remains to be answered is if this strategy will

show the same results for human hearts, as there are some differences regarding the nature of the cardiomyocytes [14].

FGF2 is a basic fibroblast growth factor expressed in several tissues, the heart amongst others [15]. Rosenblatt-Velin et al. suggested in 2005 that FGF2 could be used to induce differentiation of resident cardiac precursors [16].

2.1.4 Nuclear staining

When studying microscope images of cells, staining is frequently used to make important parts of the cell brighter than the rest. There are different ways to stain the cells and they are suitable for different tasks. To observe mitosis, a nuclear stain that binds to DNA is convenient. The normal behavior of the cell should not be disturbed; hence the staining has to be nontoxic. It enables time-laps microscope imaging of mitosis.

SiR-Hoechst is a live cell stain for DNA and it is used for the cardiomyocytes in this project. It is a far-red fluorescent staining that binds to chromosomes, known to be minimally toxic when used at the recommended concentration of 1 μM [6]. Recent findings by Sen et al. suggest that the concentration limit below which the cells are not affected is lower than previously reported [17]. Concentrations as low as 0.25 μM impairs cell cycle progression. An arrest in G₂-phase leads to a decrease in cells that enters mitosis over time. Therefore, it is desirable not to use any staining when observing live cells. When staining is necessary, it should be used with care.

2.2 State-of-the-art: Measuring proliferation

Cell proliferation can be measured by proliferation assays, by cell counting or computer algorithms. Proliferation assays is a direct way of targeting cells that are dividing. Counting of cells is an indirect way of measuring proliferation, since the number of cells can decrease by cell death as well as by less cells dividing. Computer algorithms can make use of images to find dividing cells. The different techniques are described below.

Proliferation assays are chemical compounds that targets a certain phase of cell division [18]. Nucleosid-analog incorporation assays target the synthesis of new DNA during S-phase. Nucleosides with chemical or radioactive labels are added and incorporated to the newly synthesized DNA. The radioactivity can be measured and compared with a control group. Cell cycle associated proteins assays targets proteins expressed when a cell enters the cell cycle. Proliferation can also be targeted by staining of cytoplasm.

A computational framework to annotate the different phases of mitosis called CellCognition is presented by Held et al. [4]. First, all cells in an image are segmented using a local threshold and correction by watershed. Then, cells are

classified into the different phases of mitosis using a support vector machine algorithm. At the end, hidden Markov modeling is used to incorporate the temporal information. This method has been used for fluorescent images and requires annotation of all cells in the image to create a training set.

The Cell Tracking Challenge is an initiative to develop cell segmenting and tracking algorithms [5]. For segmenting cells in a microscope image, methods such as thresholding, energy minimization, and machine learning are used. The algorithms that use machine learning models perform best in most of the cases when segmenting an image. To track a cell from one image to another, the most successful method is tracking by detection. A probabilistic framework such as distance to nearest neighbor is used to track cells between images of segmented cells. This method also shows the best result for tracking cells that are dividing [5].

U-net is a deep learning software for cell counting, detection and morphometry [19]. It can be trained and adapted to a new set of data using only a few annotated examples. It has been tried on both fluorescent images and bright field images with success.

The research about classifying cardiomyocyte cell divisions from bright field images is still in its infancy. There is some research about single cell classification in bright field images, but not regarding cell divisions [20] [21]. A recent project focused on classification of bright field images of C2C12 cells depending on the days from when differentiation was induced, but it was not measuring the number of cell divisions [22]. Another project aims to score the proliferation rate from breast cancer images to classify the severity of a tumor [23][24][25]. What all projects have in common is that they are using machine learning for classification.

2.3 The principle of a neural network

A neural network consists of neurons, which can be seen as a model of the biological neurons in the brain [26]. The artificial neurons are structured in layers with connections to the previous layer. Each connection has an individual weight, which describes the contribution from each output from the previous layer to the input of the neuron in the current layer. The structure can be seen in fig. 2.4.

When training a network, the weights are updated according to examples with known outputs [26]. A brief introduction to the subject can be found in the following sections.

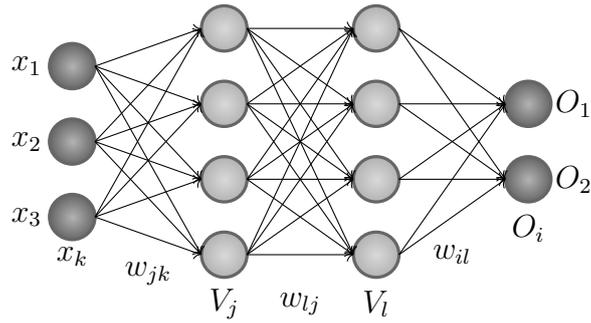


Figure 2.4: The architecture of a deep neural network with two hidden layers V_j and V_l . The inputs are denoted x_k and the outputs O_i . The connections between the layers have weights w .

2.3.1 Activation functions

To define an output to each neuron, an activation function is needed [26]. The weighted inputs to each neuron are summed and then transformed by the activation function. The most suitable choice of transformation depends on the characteristics of the problem.

One of the most used activation functions is Rectified linear unit [26], more commonly known as ReLU, which is defined by:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0. \end{cases} \quad (2.1)$$

As can be seen in eq. 2.1, it is an activation function which is zero for all negative values and linear otherwise. This results in a mathematically simple operation, which speeds up the computation time. The power required for the computations is also greatly reduced because negative values does not result in an activation. For randomly initiated weights, this corresponds to a time reduction of 50%. Furthermore, this sparsity returns a network that is more likely to reduce overfitting. This is due to that a larger amount of connections corresponds to more detailed features. Keeping only the more general features can yield a better performance on images with a structure that is quite similar to the training examples but not on detail level [27].

The possible downside of using ReLU is caused by the plane line corresponding to zero output in fig. 2.5 for negative values [28]. Due to this, the gradient can tend towards zero which means that the gradient-dependent updating of the neuron stops. One way to avoid this problem would be to use a very slight slope instead of a horizontal line for the negative axis.

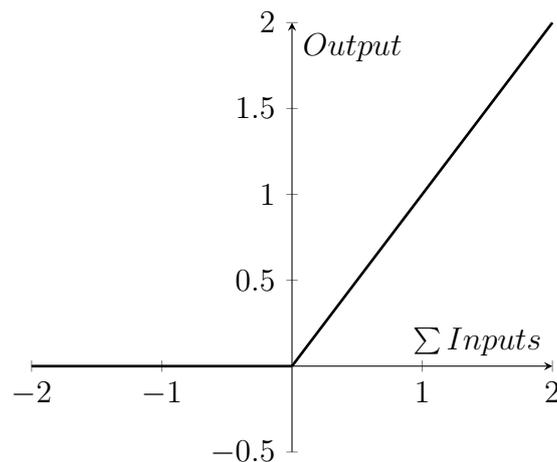


Figure 2.5: The ReLU activation function. The y-axis defines the mapping of the sum of the weighted inputs.

2.3.2 Loss functions

To evaluate the output of the network, a loss function is introduced [26]. Its purpose is to measure the error between the target value and the actual output, to ensure that the updating of the weights strives towards coinciding target and output.

One of the most simple loss functions is the square loss,

$$H_{SL} = \frac{1}{2}(t - O)^2. \quad (2.2)$$

Here, t is the target value and O is the actual output. The desired optimization corresponds to finding the global minimum of the loss function H .

There are other loss functions as well. One of the most commonly used within convolutional neural networks (CNNs) is the logarithmic cross-entropy loss [29],

$$H_{CE} = -\log(p). \quad (2.3)$$

Here p is the model's estimated probability of the corresponding class, which tends towards 1 when H_{CE} is minimized.

Several variations of the cross-entropy loss exist. If there is a large imbalance between the classes, it can result in the network classifying all the examples as the larger class. This is an issue that needs to be addressed since the network does not perform a satisfactory classification even if the score is high. The balanced cross-entropy loss function avoids this by adding a weight factor depending on class [30].

2.3.3 Optimization methods

Optimization methods are used to update the weights in the network [26]. The output of each neuron is a function of the sum of the weighted inputs. The optimization of the loss function is therefore done by updating the weights step by step starting at the output side of the network. It can be done by stochastic gradient descent (SGD), where a small increment

$$\delta w_{mn} = -\eta \frac{\partial H}{\partial w_{mn}} \quad (2.4)$$

is added to an arbitrary weight w_{mn} . The increment in eq. (2.4) depends on the derivative of the loss function H and is weighted by the learning rate η .

The increment corresponds to taking small steps in the direction of the steepest descent of H and hopefully end up in the global minimum. This is visualized in fig. 2.6 with the squared error loss function. The choice of η should be small enough to not get stuck in fluctuations around the minimum or cause divergence, but large enough to reach the minimum within a reasonable time.

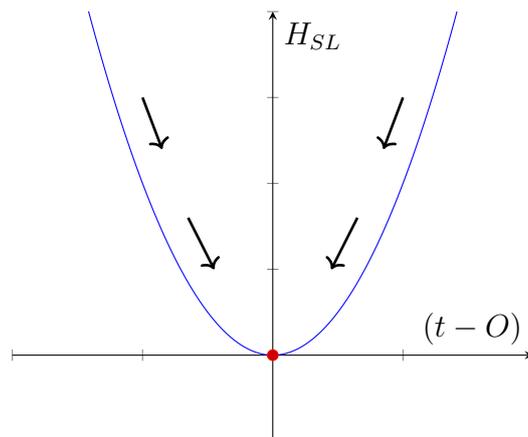


Figure 2.6: The squared error loss function, which describes the difference between target and output value. The global minimum indicated by the red mark corresponds to the desired coinciding of target and output. The black arrows explain the direction of a satisfactory updating with gradient descent.

Errors between t and O are propagated backwards, which means that the weights of the final layer are updated first. Then, the resulting weights are used to update the second last layer with the same principle. The derivatives with respect to each weight are calculated with the chain rule.

Since the SGD method has shown to be effective, there exist several refined methods with SGD as a starting point [31]. A particular successful method commonly used within deep neural networks is Adaptive moment estimation, known as Adam [32]. It combines the advantages of several other methods such as SGD and RMSprop. One of its main highlights is the use of moment to give a guidance to the direction

of the gradient and thereby yield a faster convergence. In this context, the n -th moment M_n of a variable is the random variable to the power of n ,

$$M_n = E[x^n], \quad (2.5)$$

where x is the random variable. To obtain an effective learning of the network, Adam applies individual learning rates. These are derived using estimations of first and second moment. As can be seen from eq (2.5), the first moment is the mean value and the second is variance without subtracted mean.

2.3.4 Evaluation of the model

It is important to have a good measurement to evaluate the trained model when it is tried on a test set [33]. To make a clear visualization, the possible classification results for an example is structured into a confusion matrix, which can be seen in tab. 2.1. In the table, the abbreviation tp stands for true positive, fp for false positive, fn for false negative and tn for true negative.

Class	Recognized as Positive	Recognized as Negative
Positive	tp	fn
Negative	fp	tn

Table 2.1: Confusion matrix of the possible outcomes when classifying an example.

One of the most used measurements is accuracy [33]:

$$\text{accuracy} = \frac{tp + tn}{tp + fp + fn + tn} = \frac{\text{Correctly classified examples}}{\text{Total number of examples}}. \quad (2.6)$$

If there is a large imbalance between the classes, accuracy can result in a good score for a network that classifies all the examples as the larger class. Since imbalance between the classes characterizes many data sets, it is preferable that the evaluation method can assess this. To define such a measurement, *precision* and *recall* are introduced for one class at a time in eq. (2.7) and (2.8). For the positive class, the equations are:

$$\text{precision} = \frac{tp}{tp + fp}, \quad (2.7)$$

$$\text{recall} = \frac{tp}{tp + fn}. \quad (2.8)$$

From this, the measurement F1 can be defined by:

$$F1 = (\beta^2 + 1) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}. \quad (2.9)$$

As can be seen from the F1 score, β can be adjusted if a larger weight on precision or recall is desired. When $\beta = 1$, precision and recall have an equal contribution.

2.3.5 Transfer learning and pre-trained networks

Concisely, transfer learning is a method which uses training on a different set of images to improve the training on the actual data set [34]. The requirement on the pre-learning image set is that it should have comparable features with the actual data set, but not on a detail level. The weights obtained from these iterations can be used to initialize the weights used to train on the actual data set. It is desired to use the pre-learning for the low-level features in the images, which are mostly represented by the first layers in the network. Therefore, the pre-learning can be used to either lock the weights of the first layers or give them a slower learning rate compared to the last layers.

There are existing networks available to use as a starting point for learning, for example Resnet50 and InceptionV3. They have both shown to be useful to transfer learning into biological data [34]. There are also several other models trained on different set of images that can be used. One example is VGG that has been trained on the ImageNet database [35].

2.3.6 Convolutional Neural Networks

Different structures can be applied when constructing a neural network. For image segmentation and classification, CNNs have proven to be useful due to their ability to recognize both large- and small-scale features [35].

CNNs contain convolutional layers [26]. For each layer, a 2-dimensional filter of a pre-defined size is applied. This filter slides over the image and at each step a convolution operation is performed. The output value resulting from each step is stored in a matrix called a feature map. This can be seen in fig. 2.7.

After a convolutional layer, it is common to apply a pooling layer [26]. This operation simplifies the result from the convolutional layer by combining several neighboring outputs into one single number. A common type of pooling layer is max-pooling, where the neuron returns the maximum value from the neighboring outputs in a feature map.

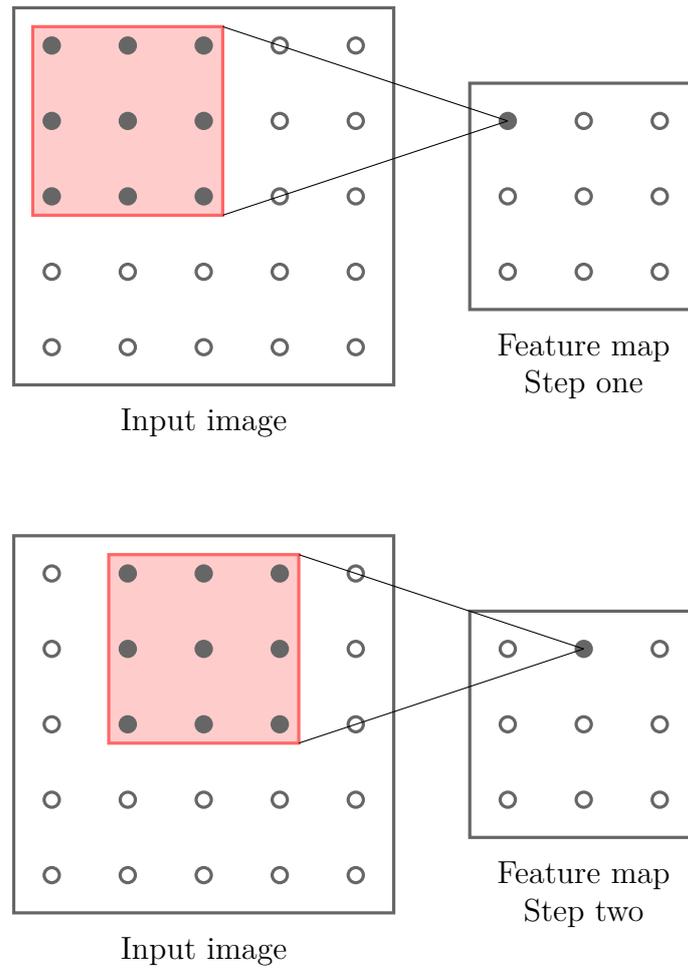


Figure 2.7: The principle of a convolutional layer. The filter, visualized as the red square, slides over the input image. At each step, a convolution operation outputs a single value which is stored in a feature map.

A CNN can also contain fully connected layers, where each neuron in a layer has a connection to all the neurons in the previous layer [26]. This type of layer is commonly used as the output.

3

Method

This chapter describes the procedure of the project. First, a suitable method to solve the problem was selected. Then, the project started with preparation of image data. Cardiomyocyte cells were cultivated to produce images using a microscope. The image sequences were then processed using image registration and then annotated. In the end, a neural network was trained to detect and count the number of cell divisions.

Detecting cardiomyocyte proliferation was made in two steps. First, training data of cell division and background were extracted as small patches from the images in the time sequences. They were used to train a convolutional neural network for classification. Then, the trained network was applied on a full time sequence to detect all cell divisions in the sequence.

3.1 Methods to find cell divisions

Several methods for finding cell divisions in a time sequence were considered. The main objective was to find a method where the time information could be used, because it was thought to have the potential to improve the solution.

A possible solution was to implement an algorithm similar to CellCognition mentioned in section 2.2. It would require segmentation of each cell using for example a watershed algorithm and annotation of the different phases of mitosis. Due to the movement of cardiomyocytes inside the well, cell tracking during mitosis would also be required. This model was rejected due to the time-consuming annotation work it would require.

Cell divisions could also be found by tracking individual cells and recording when a single cell is split into two cells. Because tracking of cells was not required to reach the aim of the project and was regarded as more complex than just counting divisions, this method was rejected.

Deep learning algorithms can be used to detect objects in images. One method is to view the time sequence as a 3D cube where 3D convolutions can be applied to find divisions. But training a network with 3D convolution is more time consuming than 2D convolution, therefore a solution with 2D was preferred. The time information from the sequence is not used when using traditional 2D object detection. A

possible approach to incorporate the information is to use several channels in the convolutional network. Several images from the time sequence can then be fed to the network on multiple channels. It is similar to the way a neural network works for color images where the different color channels are fed to a network using multiple channels.

To detect an object in the image, regions of a larger image are fed to a convolutional neural network for classification. There are different techniques to propose regions of interest to be classified that can be used to save time. The microscopic images of cardiomyocytes are very uniform which could be a problem when proposing regions. Instead of using region proposal, a sliding window can be used to classify all patches of an image.

In the end, training a neural network feeding the temporal information on multiple channels and a sliding window for object detection was considered suitable to solve the problem. Different pre-trained networks were considered to be use as a starting point. ResNet50, previously shown to improve result in biological data studies, was quantitatively tested and rejected as the result on a small training set were unsatisfactory. The VGG16 network showed a more promising result and was used throughout the project.

To train a neural network, there was a need of a large amount of image data. How this data was produced is described in the following chapter.

3.2 Production of image data

The time sequences of microscope images used in the project were cultivated by the cell lab at AstraZeneca. First, a culture of iPS CM2 cardiomyocytes was placed on a microplate to grow for a week. Microplates with 96¹ and 384² wells were used. After a week, the cell nuclei were stained using SiR-Hoechst to make them fluorescent. An automated microscope³ was used to obtain both fluorescent images, fig. 3.1, and the corresponding bright field images of the cells in fig. 3.2. A magnification of 10X was considered suitable due to the tradeoff between details and cost. A higher magnification means more details but also requires more memory and time. Images of the cells were taken with an interval of fifteen minutes for approximately three days.

¹Corning® CellBIND® 96-well Flat Clear Bottom Black Polystyrene Microplates, with Lid, Sterile

²Corning® CellBIND® 384-well Flat Clear Bottom Black Polystyrene Microplates, Low Flange, with Lid, Sterile

³IncuCyte® S3 Live-Cell Analysis System



Figure 3.1: Fluorescent microscope image with 10X magnification of cardiomyocytes with stained cell nuclei.

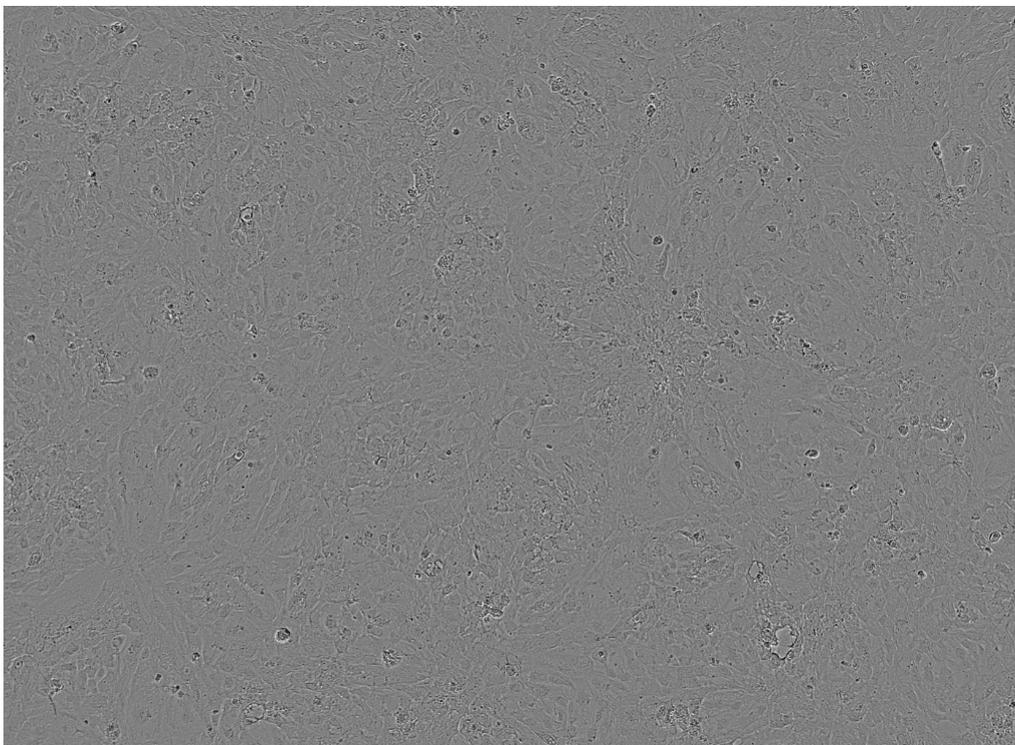


Figure 3.2: Bright field microscope image of cardiomyocytes with 10X magnification.

3. Method

Three sets of image data were produced, called A, B and C. Image set A was obtained from a microplate with 96 wells and image set B from a microplate with 384 wells, both were cultivated in January 2019. Image set C with an unknown number of wells was cultivated in June 2018. The wells on a microplate were filled with different concentrations of cells, from 500 to 20 000 cells per well. Some of the wells were also filled with a proliferation stimulus and some were used as a control. In that way, a comparison of how different cell densities and stimuli affects proliferation was possible.

Specifications of cell densities for the image set A and B can be seen in tab. 3.1 and 3.2 respectively. Those cardiomyocytes were stained with SiR-Hoechst in a concentration of 1 μ M and the growth factor FGF2 was added to half of the wells with a concentration of 20 ng/ml. The cardiomyocytes simulated with FGF2 are marked with a red color in the tables. Unfortunately, there is no specifications of the number of cells for image set C in tab. 3.3.

	2	3	4	5	6	7	8	9	10	11
A	20K	10K	5K	2.5K	1.25K	20K	10K	5K	2.5K	1.25K
B	20K	10K	5K	2.5K	1.25K	20K	10K	5K	2.5K	1.25K

Table 3.1: Specification of the number of cells in the wells for image set A. Cardiomyocytes in A7-A11 and B7-B11 are simulated with FGF2.

	3	4	5	6	7	8	...	13	14	15	16
C	5K	5K	3.75K	3.75K	2.5K	2.5K	...	5K	5K	3.75K	3.75K
D	5K	5K	3.75K	3.75K	2.5K	2.5K	...	5K	5K	3.75K	3.75K

Table 3.2: Specification of the number of cells in the wells for image set B. Cardiomyocytes in C13-C16 and D13-D16 are simulated with FGF2. Well C9-C12 and D9-D12 are excluded.

	2	3	4
B			
C			
D			
E			
F			

Table 3.3: Specification of wells used to obtain image set C.

3.3 Image registration

The time sequences of images had to be stabilized in the meaning that a single cell should stay at approximately the same place in all images. This was accomplished by using image registration. SIFT features were extracted from all images in the time sequence. The features from two images were then matched and a transform between them was found. A rigid transformation with translation and rotation was selected. All images in the time sequence were then aligned to a reference image, when possible the first image in the time sequence was selected.

The rigid transformation was selected due to that the movements of the images was caused by the camera position. The camera is moving from well to well when taking images. The position is not always exactly the same when it returns to the same well again to take the next image in the time sequence. That can cause small offsets from one image to the next in a sequence.

The software Fiji was used to perform image registration on all time sequences. It has several plugins to apply image registration. The plugin *Register Virtual Stack Slices* computes the registration based on the principles explained above [36], and it was used in this project.

The bright field time sequences were stabilized using the same transformations as for their corresponding fluorescent sequence. In that way, annotation of cell divisions in the fluorescent images could be transferred to the same position in the corresponding bright field images.

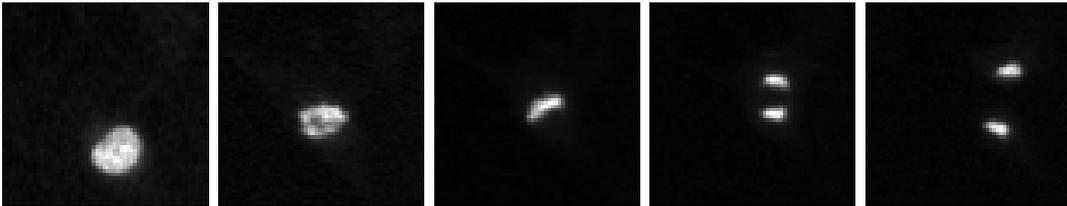
3.4 Image annotation

To create a useful data set for training a network, the time sequences had to be annotated. Cell divisions were found and marked manually in the fluorescent time sequences. Mitosis of a cell is in progress during several images and the nucleus looks different during different phases of mitosis. To always annotate the same phase of mitosis, a phase that was easy to identify was selected. From fig. 3.3 it can be seen that from $t=15$ minutes to $t=45$ minutes there is a significant difference in the nucleus appearance, corresponding to mitosis phases prophase, metaphase and anaphase. Metaphase was the easiest one to distinguish and chosen as the phase to annotate, fig. 3.3c.

A semi-automatic annotation solution was developed in MATLAB. The principle was to proceed through all images in the time sequence. When metaphase of a cell division was visually discovered, it was possible to click on it which registered and saved the coordinates and the image number. Since the mitotic event is in progress during several images, the event could later be denoted as a cube in the 2D-area space and 1D-time space. The size of the cube depends on the pixel width of a cell

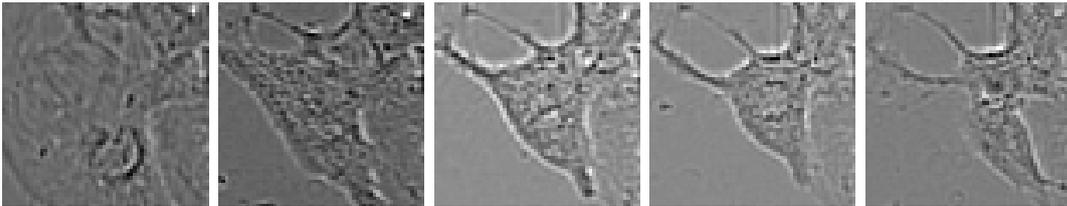
division and the duration of mitosis.

Annotations from the fluorescent images were copied and used for the bright field images. An example of a cell division in bright field can be seen in fig. 3.4. It is the same cell division as in image 3.3. While most of the cell divisions in the fluorescent channel look almost the same, cell divisions in the bright field channel differ a lot. All annotated cell divisions were then used to create a training set for a neural network.



(a) $t=0$ min. (b) $t=15$ min. (c) $t=30$ min. (d) $t=45$ min. (e) $t=60$ min.

Figure 3.3: Mitosis of cardiomyocyte viewed by fluorescent images.



(a) $t=0$ min. (b) $t=15$ min. (c) $t=30$ min. (d) $t=45$ min. (e) $t=60$ min.

Figure 3.4: Mitosis of cardiomyocyte viewed by bright field images.

3.5 Extracting training data

Based on the annotation, smaller patches like the one in fig. 3.3 were cut out from the large images, fig. 3.1, in the time sequences. Two classes of patches were extracted; cell division and background. They were then used to train a convolutional neural network for classification. The procedure of extracting cell division patches and background patches is described below.

3.5.1 Image normalization

Nuclear staining weakens over time which affects the contrast and brightness of the microscope image. All images in all time sequences were therefore normalized by

$$I_N = \frac{I - \mu_S}{\sigma_S}, \quad (3.1)$$

where I is one image in the sequence, I_N the normalized image, μ_S the mean over one time sequence and σ_S the standard deviation of one time sequence. Normalization

by image, defined in eq. (3.2), could also be applied, where the normalized image is calculated by subtracting the mean of the image and dividing by the standard deviation of the same image:

$$I_N = \frac{I - \mu_I}{\sigma_I}. \quad (3.2)$$

Here μ_I denotes the mean of image I and σ_I the variance.

3.5.2 Cell division patches

Small patches from the time sequences containing cell divisions and background were extracted for training. Different sizes in the spatial domain of the patches were tried, as well as including different numbers of images in the temporal domain. By manually examining the size of a cell division, a size of 60x60 pixels was considered suitable. However, the size of the square could be set to a smaller size and still include the whole process. Therefore, training examples with a size of 32x32 pixels were also evaluated. Early results suggested that training with 60x60 pixels was favorable, and the smaller patches were therefore rejected.

Regarding the length of the time sequence, different numbers were tried. The time duration of mitosis of a cardiomyocyte is individual for each cell, some divide fast while others are slower. By looking at a large amount of cell divisions, it was considered that most cells proceeded thorough mitosis in approximately 10 images, corresponding to 2.5 hours. During the late phase of mitosis, from anaphase, the two daughter cells separated. Including telophase would therefore require larger patches which would include more irrelevant information. The conclusion was that 5 images was suitable to capture a cell division. The largest difference in appearance of the cell was from the image right before metaphase to the image right after metaphase. Therefore, training a network using only 3 images was also tried.

Training examples of the cell division class were extracted based on the coordinates from annotation. The background patches were randomly selected from the time sequences. The total number of cell divisions in all time sequences were very few compared to background, which makes the problem imbalanced. To make the two classes more equal in number, data augmentation was performed.

3.5.3 Data augmentation

When training a convolutional neural network, access to a lot of training examples is essential. Due to that the number of cell divisions was greatly exceeded by background, each example of a cell division was valuable. All cell divisions that were found, when annotating the training set, were therefore augmented to increase the number of examples.

Augmentation was performed by translation, rotation and flipping of each cell division patch. An annotated cell division was denoted by the coordinates in the

image corresponding to the center of the cell division. When extracting cell division patches, this center was moved in x and y to perform translation of the patches. The result is patches of cell divisions where the cell division is centered or has a little offset to the middle of the patch, fig. 3.5. For patches of size 60x60, a shift by 10 pixels in x and y was used. This increased the number of training examples by a factor of 9.

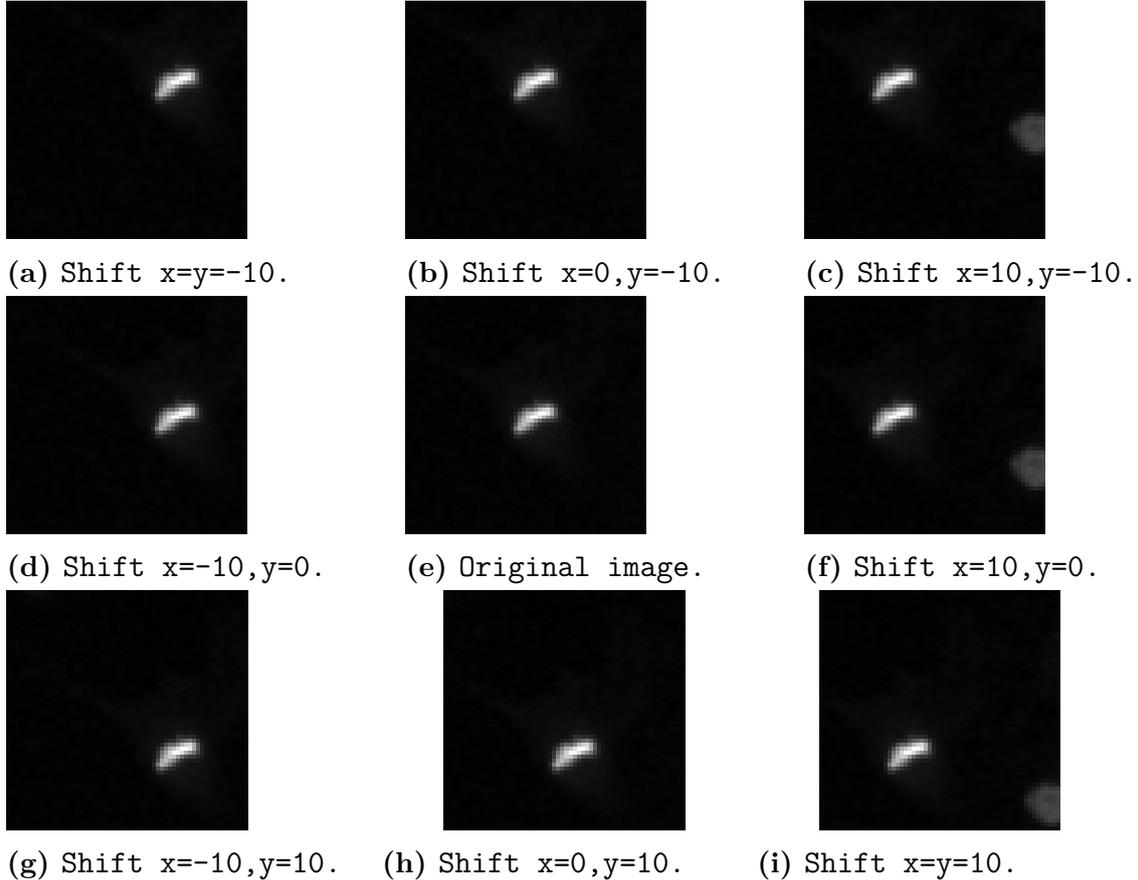


Figure 3.5: Augmentation by shifting the center of a cell division.

After translation to extract patches each patch was rotated 90, 180 and 270 degrees to increase the training set by a factor of 4. The translated patches were also flipped both horizontally and vertically. In total, a single cell division gave rise to 54 training examples.

To increase the training set further, more augmentation was performed. First, the time sequence normalization, eq (3.1), was changed to normalization by image, eq (3.2). Then, extra translation was added with a shift of 10 and 5 pixels in x and y. All cell divisions were rotated and flipped as before. In total, one cell division gave 150 additional training examples.

The small image patches of class cell division and class background were then used to train a neural network for classification. The architecture of the network will be described below.

3.6 Convolutional neural network architecture

The architecture of the convolutional neural network utilized the pre-trained VGG16 model as a starting point [35]. The VGG16 model is trained on ImageNet and takes a 3-channel image of size 224x224 as input. It consists of five blocks, each containing several convolutional layers followed by a max pooling layer. The top layers of the network consist of flattening and fully connected layers. This network was modified to be able to input the training patches of cell division and background. Two variants of the network were developed, one with 3 channels and one with 5 channels.

3.6.1 Network architecture with 3 channels

As the VGG16 network takes a larger image than the size of a cell division as input, the network had to be modified to take smaller input images. VGG16 is also designed to classify into ten different classes and not two, which had to be changed.

Reshaping the input size was done by loading VGG16 with the parameter input size set to the desired shape of (rows, cols, channels). The weights of the convolutional layers were initiated as the weights obtained from training on ImageNet. The top layers of the network, consisting of fully connected layers, were removed to be able to change the output to two classes.

To convert the network output into a classification of cell division or background, the top of the network was replaced by new layers. A global averaging pooling layer was added after block 5. It takes an input of (batch size, rows, cols, channels) and outputs (batch size, channels). Next, a fully connected layer with output shape 256 was added. At the top of the network, another fully connected layer was added with output shape 2, namely the two classes cell division or background. The network architecture can be seen in tab. 3.4. Note that the fully connected layers are denoted by dense in the table.

3.6.2 Network architecture with 5 channels

To include more phases of the cell division, it was considered a requirement to evaluate a model with examples containing more than three time steps. To perform this, the input shape of the original VGG16 model had to be reshaped with not only a smaller input size as in tab. 3.4, but also more channels.

The construction of a Keras network with more than 3 channels is alone no major difficulty. However, the weights of the network is of high importance since it has been shown that a pre-trained network most often have a better performance than a network with randomly initialized weights [37]. Because of this, it was desired to use the weights from the 3-channel VGG16 network trained on ImageNet.

Block	Layer type	Output shape
Input	Input layer	(None, 60, 60, 3)
Block 1	Conv2D	(None, 60, 60, 64)
	Conv2D	(None, 60, 60, 64)
	MaxPooling2D	(None, 30, 30, 64)
Block 2	Conv2D	(None, 30, 30, 128)
	Conv2D	(None, 30, 30, 128)
	MaxPooling2D	(None, 15, 15, 128)
Block 3	Conv2D	(None, 15, 15, 256)
	Conv2D	(None, 15, 15, 256)
	Conv2D	(None, 15, 15, 256)
	MaxPooling2D	(None, 7, 7, 256)
Block 4	Conv2D	(None, 7, 7, 512)
	Conv2D	(None, 7, 7, 512)
	Conv2D	(None, 7, 7, 512)
	MaxPooling2D	(None, 3, 3, 512)
Block 5	Conv2D	(None, 3, 3, 512)
	Conv2D	(None, 3, 3, 512)
	Conv2D	(None, 3, 3, 512)
	MaxPooling2D	(None, 1, 1, 512)
Output	Global averaging pooling	(None, 512)
	Dense	(None, 256)
	Dense	(None, 2)

Table 3.4: Modified VGG16 network architecture that takes an input of shape 60x60 with 3 channels and outputs two classes.

The reconstruction of the input was performed by writing a similar network as VGG16, but with five input channels. The three middle channels were given the weights from the original pre-trained VGG16 network. The weights of the two outermost input channels were initialized as the weights from one of the original three channels. This correspond to that one extra time step in each direction could be included. This is visualized in fig. 3.6.

If the important information is surrounded by too much irrelevant information, the network can fail with detecting the most significant features in the example. It can happen if the square marking the cell division is too big or if an excessive number of channels is included. Since it is possible that the information from the extra channels takes attention away from the most important features in the time sequence, it can be advantageous to reduce the impact from the additional channels but still keeping them. This was performed by multiplying the initial weights of the added channels by a factor smaller than 1.

The same layers as in tab. 3.4 were added to the output. After training a network to classify the smaller patches as cell division or background, a transformation to classify an entire time sequence was needed.

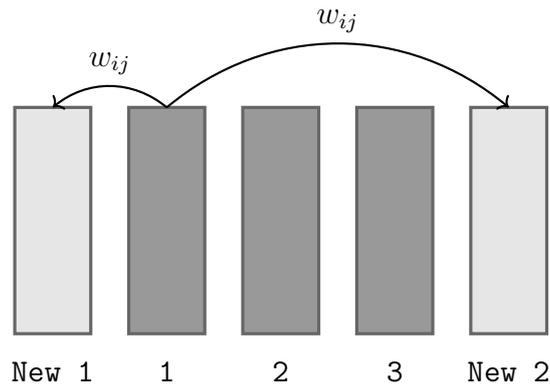


Figure 3.6: Input channels of the 5-channel network. The three middle channels corresponds to the original pre-trained VGG16 network. The initial weights of the added channels, New 1 and New 2, were copied from one of the original channels. The outermost channels were multiplied with a factor smaller than 1 to tone down their impact.

3.7 Detection of cardiomyocyte proliferation

The convolutional neural network described in section 3.6 classifies patches as cell division or background. The size of the patches is 60x60 pixels and three or five time steps long. To be able to detect cell divisions in a time sequence of size 1200x1450 pixels and with 270 time steps, some additional computation is needed. When an entire time sequence has been classified, the total number of cell divisions found can be extracted and counted.

Three different approaches were tested and compared to count the number of cell divisions in a time sequence. The first used sliding window on each image in the time sequence to evaluate if a patch contained a division or not. For the second method, the neural network was transformed to only contain convolutional layers, thus making it possible to input images of multiple size. The last method used ensemble learning to combine the evaluation of multiple sliding window models.

3.7.1 CNN and sliding window

The CNN trained to classify patches as cell division or background was used to classify images in a sliding window manner. The window had a shape of 60x60 pixels in the spatial domain and three or five images in the temporal domain, corresponding to the input shape of the CNN. Starting from the first image in the time sequence, the window extracted a patch and moved a chosen step to extract a new patch from the same image. The sliding window was applied on the entire image to extract patches. All patches were then classified by the CNN as cell division or background. This procedure was repeated for all images in the time sequence.

The step size of the sliding window could be changed to achieve the best result. Moving the window with a step that is half of the width of a patch, 30 pixels, was considered suitable. In that way, all cell divisions would be present in at least one patch. If the step size is the same as the width of a patch, there is a problem that a cell division could be split into two neighboring patches.

Also, a step size of 1 was tested. As the number of patches increases with a smaller step size, it will take more computational time to classify a time sequence. With a step size of 1, the window will be centered over every pixel in the sequence. The classification will then be per pixel and all cell divisions will be in the middle of some window.

The result from the CNN was a probability to belong to each of the two classes. A threshold was therefore set to classify the output to cell division or background. This threshold was changed to find a value that gave a high number of true positive divisions and a low number of false positive divisions.

3.7.2 Convolutional object detection

The CNN trained on patches of cell division and background was translated to a network where the fully connected layers were replaced by convolutional layers [38]. Weights and biases from the two dense layers at the end were reshaped and copied into two new convolutional 2D layers. An input layer was also added to the beginning of the network to allow multiple sized images with 3 channels as input. The network architecture can be found in appendix ???. This network can take an entire image from the time sequence as input and classify its pixels as cell division or background. The size of the output will however be reduced by a factor of 2^5 due to the five max pooling layers.

3.7.3 Ensemble learning

Ensemble learning can be used to combine the results from several models. The idea is that most of the models will find the true cell divisions, while having different false positives. Ensemble learning was implemented by training models with different initializers. If a patch in the time sequence was classified as a cell division by most of the models, it was considered as a true positive and counted as a cell division. Respectively, if a patch was considered as true by only a few of the models, it was considered as a false positive and discarded.

3.8 Active learning

Early results suggested that the number of false positives greatly exceeded the number of false negatives. Active learning, where more training examples of

the background class were extracted and then used to re-train the network, was therefore applied.

A large proportion of the false positives were located around the edges of the images. Patches from edges were therefore extracted and used to re-train the network. Also, evaluating the network on wells with different cell densities showed that the network found more false positives in wells containing more cells. Active learning was used in the form that more background patches were randomly selected from wells with more cells. The network was re-trained to increase the performance.

The false examples used for training were randomly selected from the time sequences. To guarantee that the most difficult false examples also were used for training, they were extracted manually. A network was trained with randomly selected false patches, edge patches and extra patches from wells with high density. Then, the result was analyzed and all false positives extracted to be used for re-training. As the network found more cell divisions than the annotated divisions, all patches marked as true were viewed manually to sort out the false positives.

3.9 Specification of training parameters

During the development of a neural network, it is important to split the available data into training, validation and test. A suitable split was considered to be by time sequence. Two time sequences from each of image set A and B was set aside for testing, well C3 and C7 from A and well D3 and C13 from B. Specifications of the image sets can be found in tab. 3.1 and 3.2.

The following two sections describes the training parameters used. Two networks were developed, one for fluorescent images and one for bright field images.

3.9.1 Fluorescent images

Training of the neural network to classify fluorescent images was made with 52 110 examples of cell divisions and 161 607 examples of background. From these examples, 10 % was randomly selected as the validation set. In detail, the training and validation set consisted of the following images:

- 52 110 examples of the class cell division.
- 28 248 examples of the class cell division with more augmentation.
- 102 000 randomly selected examples of the background class from A and B.
- 1054 background class examples of edges.
- 30 009 randomly selected examples of background from C.
- 28 544 false positive background examples from first training.
- 7 296 false positive examples from second training.

The hyperparameters of the neural network are important for the model performance. Some different parameter settings were tried on a small set of data and the parameters presented next were selected. The model was trained in two steps. First, the new top layers of the VGG16 model were trained. A ReLU-layer was used as the activation function for the dense 256 layer. A sigmoid function was used for the dense 2 layer for classification. As an optimizer, Adam with a learning rate of 10^{-3} was used. The loss was computed as binary cross-entropy.

For the second round of training, the last convolutional layer of block 5 in tab. 3.4 was trained together with the top layers. The optimizer used was SGD with learning rate 10^{-4} . Again, binary cross-entropy was used as loss.

3.9.2 Bright field images

The training parameters for bright field images differed partly from fluorescent images. First, more examples were used from both classes. That was possible due to annotation of well D4, E4 and F2 from image set C, that were not yet annotated when developing the network for fluorescent images. All cell divisions were also augmented using more translation and a different normalization to increase the number of positive examples. The number of false examples was increased by randomly extracting background patches and by extracting false positive examples. The false positives were found by training a model, classifying all training time sequences and then extracting the false positives. A second model was trained and the false positives were once again extracted to train a third model. The total number of training examples for the last model was:

- 64 314 examples of the class cell division.
- 178 500 examples of the class cell division with more augmentation.
- 150 000 randomly selected examples of the background class.
- 47 796 false positive background examples from first training.
- 3 883 false positive background examples from second training.

The model was trained in the same way as for the fluorescent images. A first round of training of the new top layers of VGG16 using the Adam optimizer with learning rate 10^{-3} . A ReLU-layer was used as the activation function for the dense 256 layer. A sigmoid function was used for the dense 2 layer for classification. The loss was computed by binary cross-entropy. For the new top layers of VGG16, different initializers were used to obtain slightly different models [39]. The five weight initializers used were:

- Model trained on part of the bright field images.
- Glorot uniform.
- Random uniform.
- Truncated normal.
- Lecun uniform.

The first model was initialized with the weights of a model that was previously trained on part of the bright field examples. For example when false positives had been extracted from a model, the same model weights was used to train a new model using the false positives.

For the second round of training, the last convolutional layer of block 5 was trained together with the top layers. The model was trained using SGD optimizer with a learning rate of 10^{-4} . Binary cross entropy was used as loss function.

4

Results and Evaluation

This chapter presents the results of the project. First, an overview of the annotation of cell divisions is shown. Then, the result of the CNN is divided into two parts. The first part covers the fluorescent channel and the second part covers the bright field channel. Ensemble learning was evaluated on the bright field images only.

4.1 Annotated cell divisions

The number of manually counted cell divisions in each well can be found in tab. 4.1. As can be seen, there was a large difference between the number of cell divisions from the image sets A and B compared with C. There was also a difference depending on the cell density. Wells with fewer cells contained fewer cell divisions. It was partly due to that cardiomyocytes tend to stick to the edges of a well while the image was taken in the middle. In a well with fewer cells, it resulted in that only a small fraction of the total number of cells in the well was captured on the image.

4.1.1 Side effects of nuclear staining

The image sequences A and B showed indications of toxicity. As can be seen in fig. 4.1, the number of cell divisions decreases with time. A possible cause is the nuclear staining SiR-Hoechst, which was used in a concentration of $1\mu\text{M}$. The nuclear staining can cause cells to arrest in the cell cycle and fewer cells to enter mitosis over time [17].

4.1.2 Cell growth factor FGF2

The cell growth factor FGF2 described in section 2.1.3 was added to half of the wells from image set A and B. Out of the 147 cell divisions from A and B in tab. 4.1, 90% was from cardiomyocytes in the control group. Only 10% of the cell divisions were found in wells with FGF2.

Image set	Well	Cells in well	Divisions 1	Divisions 2
A	B2	20 000	24	10
A	B7	20 000	1	0
A	B8	10 000	1	0
A	B9	5000	1	0
A	C2	20 000	11	9
A	C3	10 000	4	4
A	C4	5000	5	11
A	C6	1250	2	1
A	C7	20 000	2	1
B	C3	5000	10	0
B	C4	5000	9	2
B	C13	5000	2	0
B	C14	5000	1	2
B	D3	5000	9	2
B	D4	5000	15	2
B	D8	2500	1	1
B	D13	5000	2	0
B	D14	5000	1	1
C	B2	–	192	–
C	B3	–	188	–
C	B4	–	169	–
C	C2	–	89	–
C	C3	–	125	–
C	C4	–	134	–
C	D2	–	189	–
C	D3	–	233	–
C	D4	–	130	–
C	E2	–	25	–
C	E3	–	47	–
C	E4	–	37	–
C	F2	–	21	–

Table 4.1: The number of cell divisions that was annotated inside each well can be found in the table. Divisions 1 are the number of initially annotated divisions and divisions 2 are new cell divisions found by the network that were missed during the manual annotation.

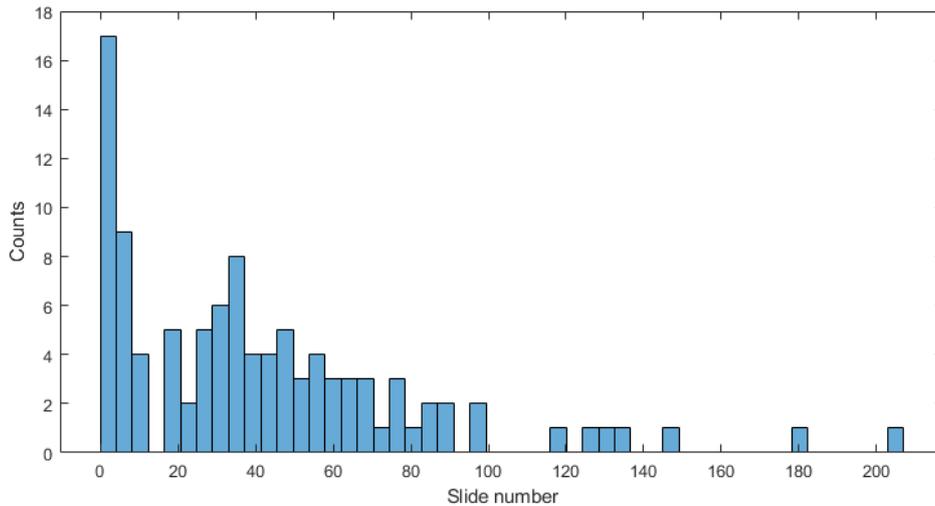


Figure 4.1: The number of cell divisions during the duration of a time sequence.

4.2 Fluorescent images

Two CNN models were trained to classify 60x60 patches as cell division or background, the 3-channel network described in section 3.6.1 and the 5-channel network described in section 3.6.2. They were initially compared by evaluating the accuracy of a validation set. The 3-channel model had a better performance than the 5-channel model with an accuracy of 99.37%, compared with 98.09%. Therefore, the 3-channel model was selected to develop further to maximize the performance.

Active learning was used to improve the 3-channel model. False positives from a trained model were extracted twice and used to train a new model. From the first iteration of including false positives, the model achieved an accuracy of 98.88%. For the second iteration with more false positives, the accuracy was 97.81%. For the second round, more true positives were also added by extra augmentation. Details of the training examples used can be found in section 3.9.1.

Even though the accuracy of the validation set decreased, the performance of the model when detecting cardiomyocyte proliferation in a full time sequence increased. Result from detecting cardiomyocyte proliferation will be presented below.

4.2.1 Detection of cardiomyocyte proliferation

The trained models were used to detect and count cardiomyocyte proliferation in a time sequence. Two things were considered when evaluating their performance; the success in finding the annotated cell divisions and the number of false cell divisions found. For the 5-channel model, the number of false positives was considerably

4. Results and Evaluation

higher than for the 3-channel model. The 3-channel model was therefore used in further tests.

Two different approaches were used to detect and count cell divisions. The *CNN and sliding window* method and the *convolutional object detection* method described in section 3.7. For both approaches, the model with an accuracy of 98.88% was used to evaluate the results.

Translating the trained 3-channel model to an object detection model to find cell divisions was not successful. Even though the computational speed increased a lot, the number of false positives were high. Figure 4.2 shows predicted cell divisions for a single image in a time sequence with object detection. The true number of divisions in this image is 1, and the model found 38. The convolutional object detection method was therefore abandoned.

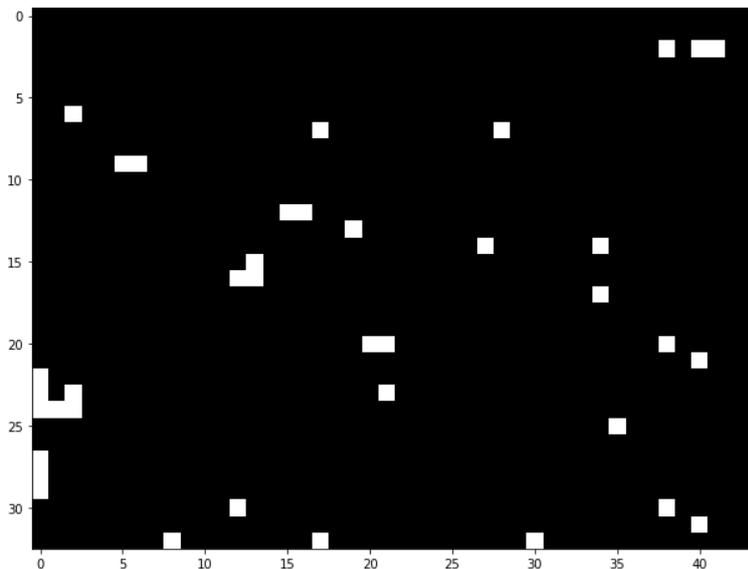


Figure 4.2: Cell division detection using convolutional object detection. A white square represents what is classified as cell division.

Counting cell divisions by evaluating patch by patch using a sliding window gave the most promising results. To increase the speed of the calculation, a larger step size of the sliding window can be used. Therefore, the performance was evaluated both with a step size of 1 and a step size of 30. Figure 4.3 shows the predicted cell divisions on a single image using a step size of 30 and threshold of 0.5. As can be seen in the figure, the model correctly classifies the true cell division as a cell division, but also classifies two background patches as divisions. Figure 4.4 shows the same image predicted with a step size of 1 and threshold of 0.999. A smaller step size enables a higher threshold since each pixel will be classified. The two white dots represent pixels classified as cell divisions. For this image, predicting

with a step size of 1 and 30 gave the same result.

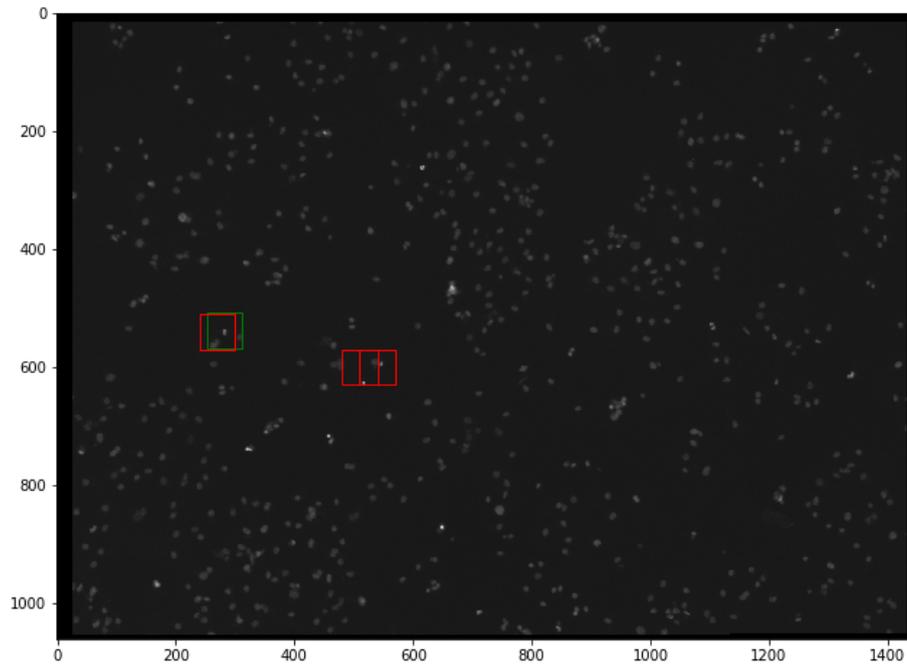


Figure 4.3: Red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C3 from A, slide 8 with threshold 0.5.

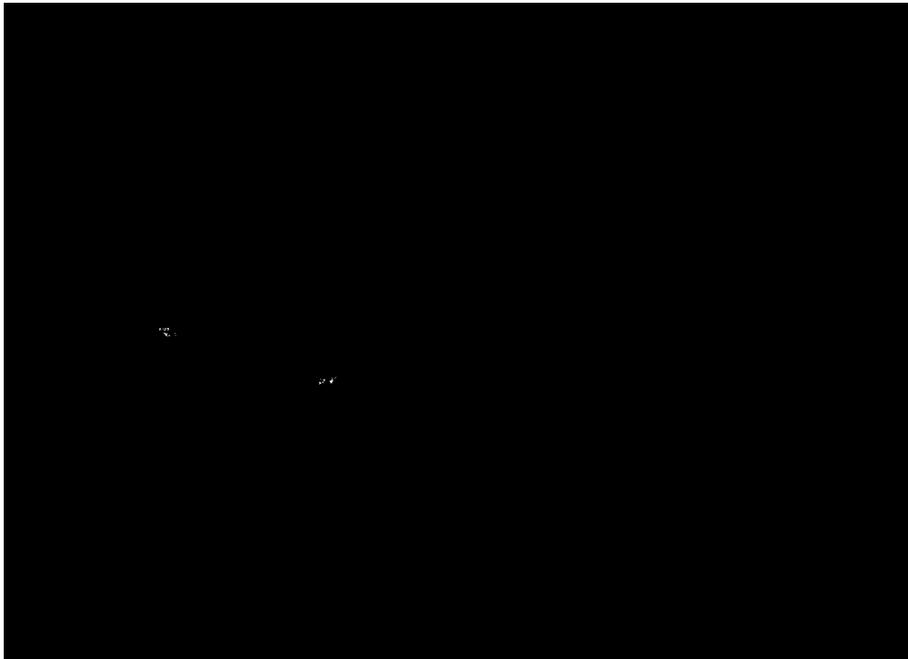


Figure 4.4: White pixels indicate predicted cell divisions when step size 1 is used. Well C3 from A, slide 8 with threshold 0.999.

Testing the model on an image containing more cells showed a difference between a step size of 1 and 30. Figure 4.5 shows the predicted cell divisions using a step size of 30 and in fig. 4.6 the same image with a step size of 1 can be seen. For the smaller step size, the true cell division in the bottom of the image is clearly visible.

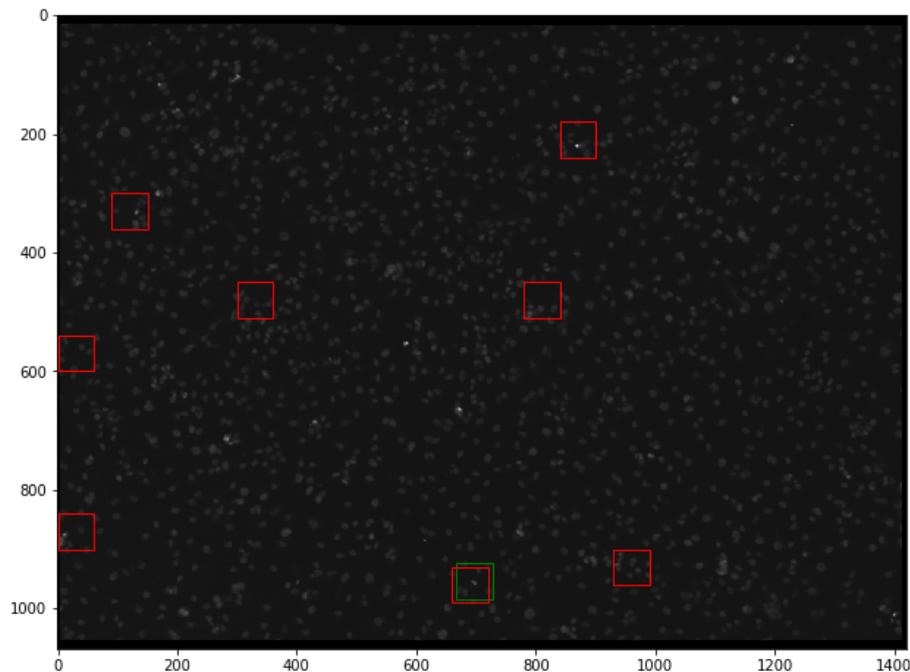


Figure 4.5: The red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C7 from A, slide 74 with threshold 0.5.

A summary of the 3-channel model’s performance using sliding window with step size 30 for detection can be seen in tab. 4.2. The table contains an evaluation of the four test sequences, together with the number of annotated divisions, true positives (tp) and false positives (fp). A difference can be seen connected to the cell densities. The number of false positives is higher for C7 A and C3 A, which have a density of 20 000 respectively 10 000 cells/well. The wells D3 B and C13 B both have a density of 5000 cells/well. Most of the cell divisions are found, which also gives a recall close to one. A number of false positives that is higher than the number of true positives will however result in a low precision. The accuracy of the model is still high due to that almost 400 000 patches are classified in a full sequence.

After an additional round of active learning the performance of detecting cardiomyocyte proliferation increased. As stated above, the model had an accuracy of 97.81% on a validation set. When used to detect proliferation in the test set, the number of false positives was lower than for the model with an accuracy of 98.88%. A summary of the evaluation can be found in tab. 4.3. Here, the additional cell divisions in the last column in tab. 4.1 are also evaluated.

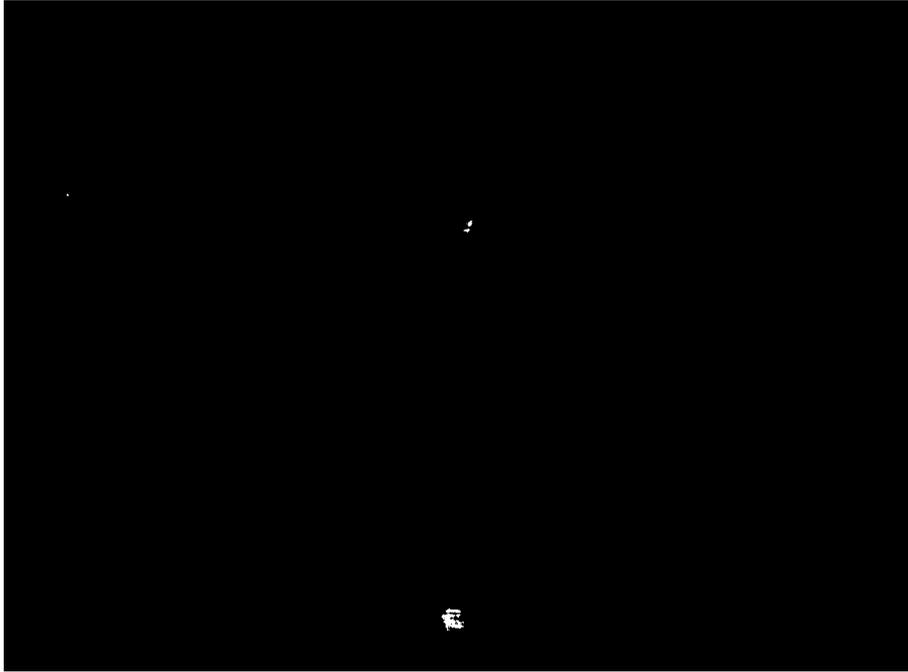


Figure 4.6: White pixels indicate predicted cell divisions when step size 1 is used. Well C7 from A, slide 74 with threshold 0.999.

Well	Image set	Divisions	tp	fp	Accuracy	Precision	Recall
C3	A	4	4	201	0.9995	0.0195	1
C7	A	2	2	772	0.9981	0.0026	1
D3	B	9	6	88	0.9998	0.0714	0.67
C13	B	2	1	42	0.9999	0.0227	0.5

Table 4.2: Number of cell divisions found in the test time sequences of fluorescent images. CNN model with accuracy of 98.88%.

Well	Image set	Divisions	tp	fp	Accuracy	Precision	Recall
C3	A	8	7	78	0.9998	0.0824	0.875
C7	A	3	3	653	0.9984	0.0046	1
D3	B	10	8	35	0.9999	0.1860	0.8
C13	B	2	1	18	0.9999	0.0526	0.5

Table 4.3: Number of cell divisions found in the test time sequences of fluorescent images. CNN model with accuracy of 97.81%.

4.3 Bright field images

Two models were trained on the same examples, one using 3 channels and one with 5 channels. The 3-channel network achieved an accuracy of 97.75% on the validation set, compared with 96.88% for the 5-channel model. Since the 3-channel model had a higher accuracy, it was used during the evaluation of detecting proliferation.

4.3.1 Detection of cardiomyocyte proliferation

Three channel CNN models were used to detect and count cardiomyocyte proliferation in a full time sequence. Two different approaches were used, *CNN and sliding window* and *Ensemble learning* described in section 3.7. The *convolutional object detection* method was not tested due to the poor result in the fluorescent channel.

The performance of the model regarding counting cell divisions was first evaluated using a sliding window on the test time sequences. A model trained on 64 314 examples of cell division and 150 000 examples of background was evaluated to begin with. The model will from now on be referred to as model FIRST. The results on the test set evaluated with a step size of 30 can be seen in tab. 4.4. The accuracy is close to one due to the high number of patches evaluated in a time sequence. The recall is lower than for the fluorescent images in tab. 4.2 which means that fewer of the cell divisions are found. A high number of false positives were detected by the model, which gives a low precision.

Well	Image set	Divisions	tp	fp	Accuracy	Precision	Recall
C3	A	4	4	523	0.9987	0.0076	1
C7	A	2	1	4504	0.9887	0.0002	0.5
D3	B	9	4	331	0.9992	0.0119	0.44
C13	B	2	1	936	0.9977	0.0011	0.5

Table 4.4: Number of cell divisions found in the test time sequences of bright field images by model FIRST.

The model was evaluated both with a step size of 1 and 30. An example image from well C3 from image set A can be seen in fig. 4.7. There is one cell division in the image, indicated with a green square. The true cell division was successfully found, but a false cell division was also detected by the red square to the right of the true division. The false detection disappeared when a step size of 1 was used. A smaller step size enables a higher threshold since each pixel will be classified. For fig. 4.8, the white pixels indicate cell divisions found with a threshold of 0.999.

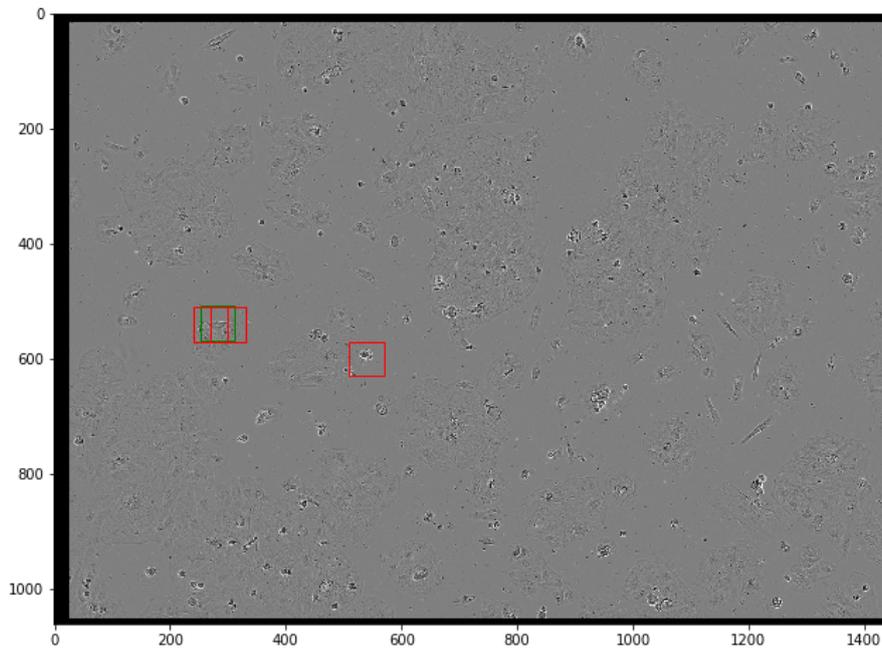


Figure 4.7: Red squares indicate predicted cell divisions while the green square is a true cell division, sliding window with step size 30. Well C3 from A, slide number 8 with threshold 0.5.

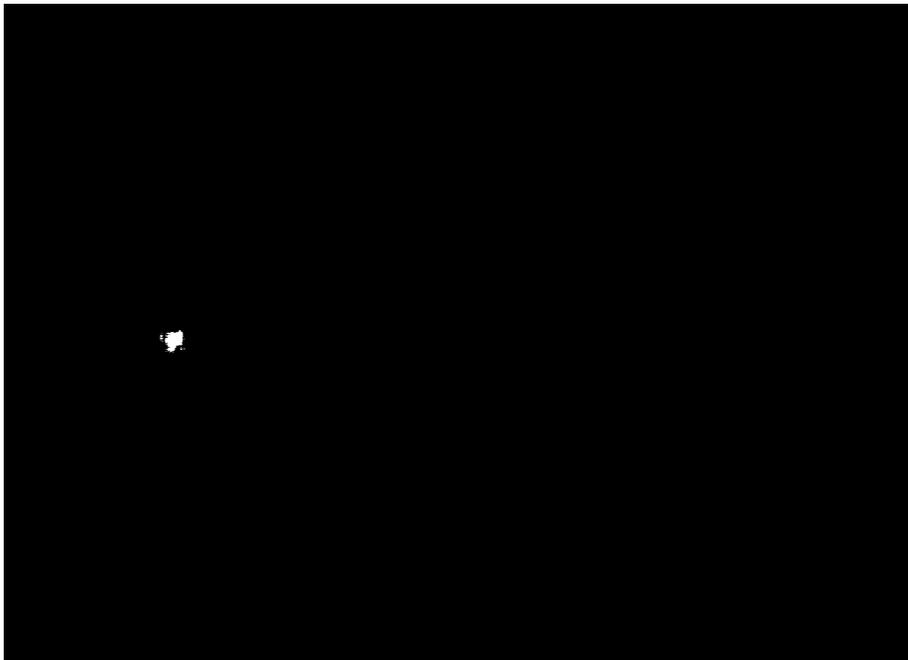


Figure 4.8: White pixels indicate predicted cell divisions when step size 1 is used. Well C3 from A, slide number 8 with threshold 0.999.

4.3.2 Ensemble learning

The second method used to detect and count proliferation was *Ensemble learning*. In total, nine models were trained using different initializations. The trained models are described in the following list. For Network 1, the training consisted of images where the number of negative examples was approximately three times higher than the positive examples. For Network 2, extra augmentation of positive examples was performed to create an almost equal number of positive and negative examples. The trained models are from now on referred to by the number in the list. The initialization type is also stated in the list.

1. Network 1:
 - (a) Weights from model FIRST
 - (b) Random Uniform
 - (c) Glorot Uniform
 - (d) Truncated Normal
 - (e) Lecun Uniform
2. Network 2:
 - (a) Glorot Uniform
 - (b) Weights from model 1 (a)
 - (c) Truncated Normal
 - (d) Lecun Uniform

Ensemble learning was applied on all time sequences using two different ensembles. The selection of models for each ensemble was based on each model’s performance of finding cell divisions and the number of false examples. For details on what cell divisions were found and the number of false positives from each model, see appendix ???. Models from network 2 found more cell divisions but also had significantly more false positives than models from network 1, which can be seen in tab. 4.5. To select some of these models and mix them with the models from network 1 was considered a suitable choice for the first ensemble. From network 2, model 2 (a) and 2 (b) had the least amount of false positives and was therefore used. The models of ensemble 1 was then model 1 (a) - 1 (e) and 2 (a) to 2 (b).

Well	Image set	Network 1	Network 2
C3	A	181	1704
C7	A	3458	17276
C13	B	278	2694
D3	B	125	1772
Total:		4042	23446

Table 4.5: The total number of false positives from two sets of networks evaluated on the same test sequences. Each network set contains four networks.

While ensemble 1 was focused on minimizing the number of false positives, ensemble 2 was weighted towards maximizing the number of found cell divisions. Selecting models for each ensemble, it turned out that there is a tradeoff between found divisions and false positives. Including all models from network 2 increased the number of false positives but also the number of cell divisions found. In ensemble 2, model 1 (b) and 1 (c) was discarded as they were the two models that found fewest cell divisions. It was considered that these models would not contribute significantly in increasing the number of found cell divisions by the ensemble. From network 2, model 2 (d) was added. The models of ensemble 2 was then 1 (a), 1 (d), 1 (e), 2 (a), 2 (b) and 2 (d).

The results from evaluating ensemble 1 and 2 can be seen in tab. 4.6. A patch was considered as a true cell division, if it was classified as a cell division by at least three of the models in the ensemble. The number of false positives detected for each sequence is found in the same table. It can be seen that the number of false positives was greatly reduced, but several true divisions were dismissed as well. A summary of the result containing all training and test wells can be seen in appendix ??.

Well	Found by ensemble 1	Found by ensemble 2	fp 1	fp 2
C3, A	63 %	75 %	39	71
C7, A	33 %	33 %	679	1091
C13, B	50 %	50 %	33	82
D3, B	40 %	40 %	26	47

Table 4.6: Results from ensemble evaluation of the test sets. Ensemble 1 has fewer false positives (fp 1) and found fewer cell divisions than ensemble 2.

A summary of the two ensemble performances on the test set can be seen in tab. 4.7 and 4.8. Compared with the previous reported result in tab. 4.4 and 4.2, the new cell divisions found by the fluorescent model are included. The number of false positives was reduced which increased the accuracy and precision of the ensembles compared to model FIRST. Some cell divisions were missed by the ensemble that were found by model FIRST. The recall is therefore lower in three of the four test wells.

Well	Image set	Divisions	tp	fp	Accuracy	Precision	Recall
C3	A	8	5	39	0.9999	0.1136	0.625
C7	A	3	1	679	0.9983	0.0015	0.33
D3	B	10	4	26	0.9999	0.1333	0.40
C13	B	2	1	33	0.9999	0.0294	0.5

Table 4.7: Number of cell divisions found in the test time sequences of bright field images by ensemble 1.

Well	Image set	Divisions	tp	fp	Accuracy	Precision	Recall
C3	A	8	6	71	0.9998	0.0779	0.75
C7	A	3	1	1091	0.9973	0.0009	0.33
D3	B	10	4	47	0.9999	0.0784	0.40
C13	B	2	1	82	0.9998	0.0120	0.5

Table 4.8: Number of cell divisions found in the test time sequences of bright field images by ensemble 2.

4.3.3 Comparison between ensemble and single model

A comparison was made to qualitatively evaluate if an ensemble has a better performance than a single model. The evaluation was based on the four test wells. To obtain a robust result, the median of the test wells was selected. Since there are four wells, an average of the two median wells was calculated. The median wells were C7 from A and D3 from B. The metrics used in the comparison were precision and recall.

In fig. 4.9, the precision and recall of all single models and the two ensembles can be seen. The models from network 1 are plotted with circles, network 2 with squares and ensemble 1 and 2 with red asterisks. A perfect model would have a precision and recall of 1, corresponding to a marker in the top right corner. The marker closest to the corner is from ensemble 1. The ensemble has the highest precision and the second highest recall. The only model with better recall is model 2 (b), but this model has more false positives which decreases the precision.

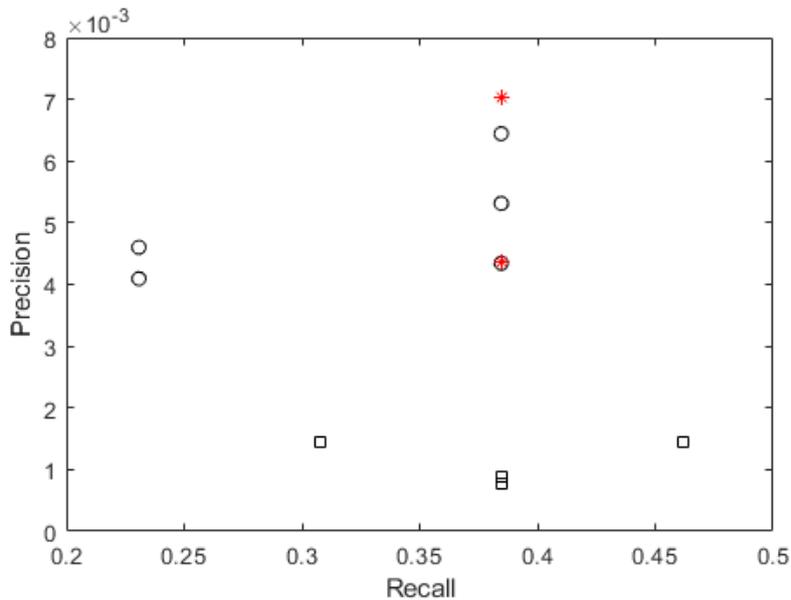


Figure 4.9: Recall and precision of the models from network 1 (circles), network 2 (squares) and the two ensembles (red asterisks).

4.4 Image set C

The time sequences in image set C contained a considerably larger amount of cell divisions. To manually denote if each cell division had been discovered by the network was therefore considered too time consuming. Furthermore, the number of false positives was extremely high. Due to this, image set C was only used to obtain training examples of cell divisions.

5

Analysis and Discussion

In this chapter, the results will be analyzed and discussed. First, possibilities and drawbacks of the method used are described together with suggestions of future development. Then, the model's capability of counting cell divisions is discussed. Possible causes of the high number of false positives are explained next, together with the improvement due to active learning and using ensembles. In the end, some limitations of the developed model are presented.

5.1 Network architecture and training

In section 3.1, different approaches of finding cell divisions from a time sequence were presented. The choice of a convolutional neural network where multiple images were used as input on three channels did not result in a perfect classification. However, cell divisions were found in both the fluorescent channel and the bright field channel. The main problem is the high number of false positives. To be useful in the application of counting cell divisions in a time sequence, the number of true positives should be higher than the number of false positives. We cannot know if some of the other methods might have resulted in fewer false positives.

5.1.1 VGG16

The starting point of all the used networks had the structure of VGG16. It is developed to be used with color images with three channels. The adaption to a 5-channel network seemed in theory to be a good choice to include a larger part of mitosis. Surprisingly, the performance of a 3-channel network was better. This is probably due to that the relevant information got swallowed up by the surroundings. It was tried to add just a fraction of the two outermost channels to still incorporate a larger part of the event but emphasize the three middle channels. This network was still outperformed by the 3-channel network, which suggests that the additional channel fractions were considered as irrelevant noise by the network.

5.1.2 Stabilization of time sequences

A rigid transformation with translation and rotation was used to stabilize all time sequences. The stabilization did not turn out perfect for all images. In some cases,

a rotation of an image in the time sequence was considered needed by the Fiji software, but unnecessary to an observer. The result was that some images were rotated in relation to each other. This rotation triggered the model to classify a lot of cells in the rotating image as cell divisions. An example can be seen in fig. 5.1 for well B7 from A. There are 87 false positive examples for this single image in the time sequence, while the average number of false positives per image for the rest of the time sequence is five.

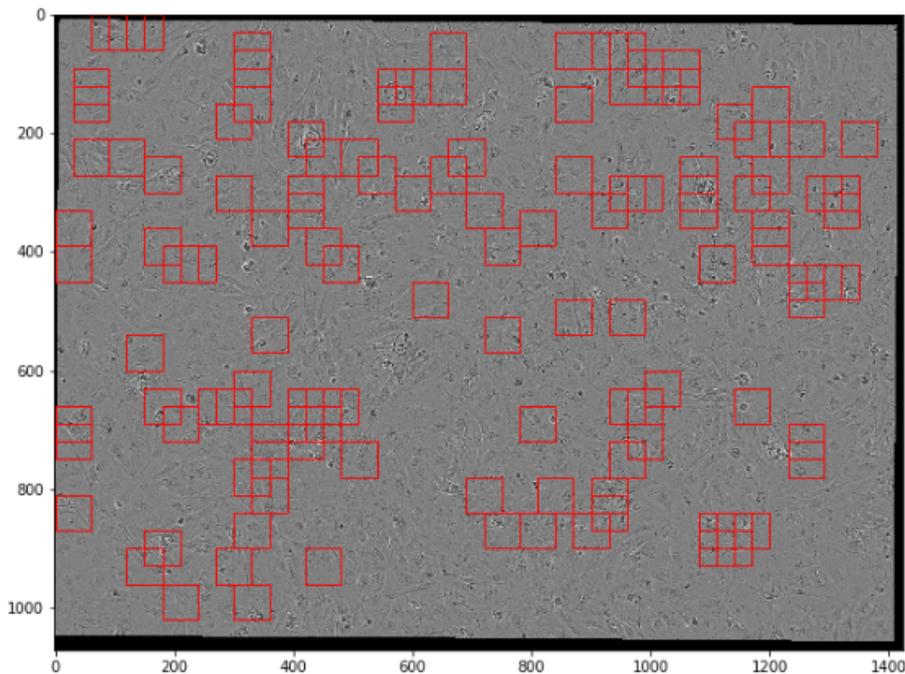


Figure 5.1: Rotation of an image in the time sequence causes false positive cell divisions.

A solution to the problem would be to change the transformation from translation and rotation to only translation. The movement of images in the time sequences occurs due to the camera position above the well. A rotation of the camera or microplate inside the microscope is unlikely and thus a rotation should be redundant.

Another solution that was applied was to extend the training set with the false detections. After extracting the red squares in fig. 5.1 and training a new model, the number of squares reduced to 2. The result can be seen in fig. 5.2. The reason that the first model could not handle the rotation of images was probably because there was few or no training examples with rotation. When the model was trained with examples of rotation the outcome was significantly increased. The result is thus affected by the fact that the false positive examples detected in fig. 5.1 were used to train the model in fig. 5.2. Evaluating the new model on a test well was not possible due to that there was no rotations in the test wells.

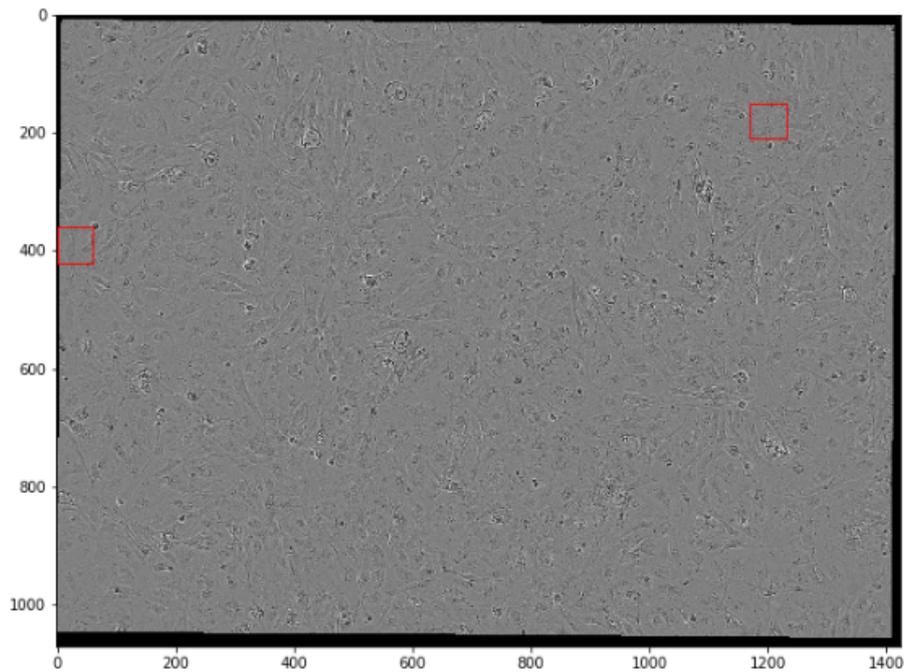


Figure 5.2: Most of the false detections in fig. 5.1 are removed.

5.1.3 Filtering of images

The structure of the bright field images was diffuse, and it was not possible to undoubtedly distinguish a cell division with the human eye. Since the network managed to find cell divisions among the bright field images, it was considered if filtering the images would make the cell divisions more discernible to the eye too. Applying some filtering could thereby yield even better results with the network as well. In the software Fiji, a variance filter was applied. It was desired that this image processing would reveal valuable details not visible in the original images. As can be seen in fig. 5.3, the structure did appear different after applying the variance filter. Unfortunately, the cell divisions could not be easily distinguished for the human eye. The filtered images were not used for the development of any network in this project.

When filtering the bright field images with a variance filter, the structure appeared differently from the original images. Even though we could not distinguish any clear patterns of cell divisions, it is worth investigating if the information is useful to the network. Training a network with the corresponding filtered training sets and compare both the number of found cell divisions and false positives with the original network would be a suitable proceeding.

The method used to train the network used a pre-trained model as a starting point and the first layers were frozen before training with the training set. The motivation to this was that the first layers corresponds to more general features and the last layers recognizes the detailed features. Another approach could have been to not freeze the first layers. Instead, the first layers can be used by the network itself to filter the input image. Exactly how a neural network learns is still to be

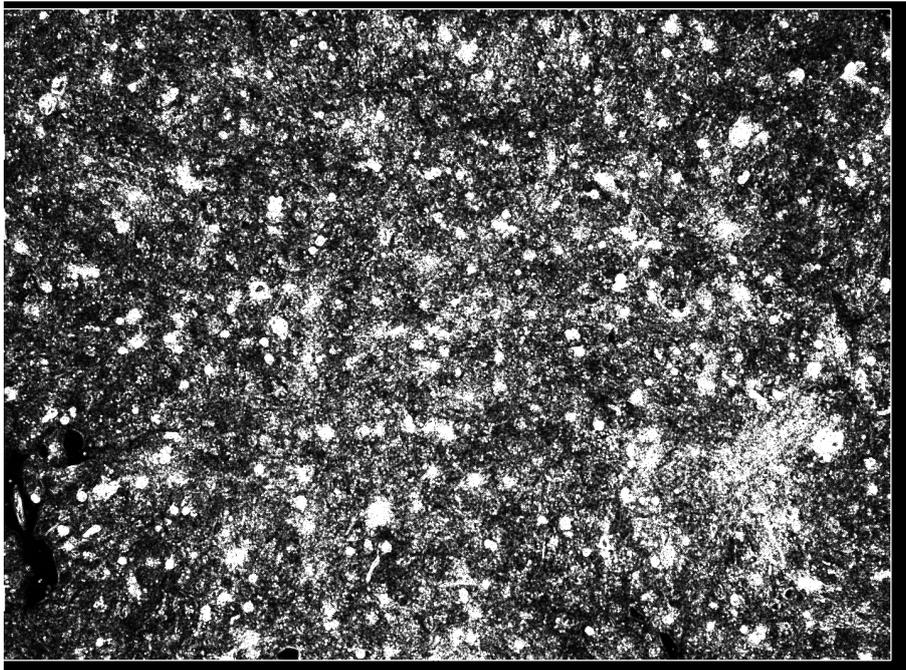


Figure 5.3: Variance filtered bright field image.

examined. Filtering an image to make it clear for the human eye might not mean that a neural network can learn better. To let the network itself be responsible for filtering might therefore return a better model.

5.1.4 Training with different images

One future possibility is to use some other images of cardiomyocytes to train a network. This project was limited to use images from a fluorescent channel and a bright field channel. The greatest success would be to use images without staining to detect proliferation, as staining might affect the normal behavior of a cardiomyocyte. In the fluorescent channel, the nuclear staining SiR-Hoechst was used. There are other types of staining available and different parts of the cell could be stained. One possibility is to use staining of the cytoplasm that would hopefully not affect the cardiomyocyte as much as staining the nucleus. A drawback is that mitosis would probably not be as easy to detect, at least for a human eye.

There are techniques to produce other images than bright field images of non-stained cardiomyocytes. One technique is *Digital Phase Contrast* which is available at AstraZeneca. Images obtained from *Digital Phase Contrast* can be more clear than bright field images. However, it requires more work to produce those images and it will thus be costlier.

5.1.5 More training examples

The access to a lot of training data is essential to train a neural network. In this project, approximately 240 000 examples of cell divisions and 200 000 examples of background were used for the bright field channel. The training examples were extracted from image set A, B and C where a total of 1726 cell divisions were found. The training set of cell divisions thus originated from a small number of annotated divisions which were augmented. It would be of interest to have more examples of cell divisions to use less augmentation.

An analysis of how much the number of training examples affected the model was performed. A network was trained with 100 000 randomly selected examples, half from each class. The same network was then trained with 200 000 examples, half from each class. With fewer training examples, the model found 74% of the cell divisions and 11 049 false positives. With 200 000 training examples, the model found 70 % of the cell divisions and 6 981 false positives. In conclusion, this indicates that it would be worth to produce more image sets of cardiomyocytes for training.

5.2 Measure of proliferation rate

The counting of divisions was done with a sliding window approach where each patch was evaluated. The step size of the sliding window turned out to be of high importance. The cell divisions in the training examples were located both approximately in the middle of the image and with an offset of 10 pixels due to augmentation. If the step size is too large, a dividing cell could be missed if no evaluated patch contained the cell division approximately in the middle of the square. On the other hand, an excessive number of squares is time-consuming to evaluate. The chosen step size of half a square was a relevant compromise which would give the network a possibility to detect all cell divisions.

It was shown that the cell divisions often were marked several times in a neighborhood of time and space. This was used in an attempt to distinguish cell divisions from false positives. A detected cell division was considered true if enough neighboring patches had detected it. Even if the number of false positives was reduced, this did not solve the problem. Several of the false positives were just as strongly marked as the true positives. Furthermore, a considerably amount of true positives was not marked enough times to be registered as a division. This method was therefore rejected.

Decreasing the step size to one pixel enabled a higher threshold for classification. In fig. 4.8, the threshold of class cell division is 0.999. The true cell division gets a score higher than the threshold on several pixels in the image. Furthermore, the false positive detection in fig. 4.7 vanishes due to a score below the threshold. To decrease the step size and increase the threshold therefore seems like a promising approach. Because of time limitations, this method could not be evaluated on a full time sequence to compare with a step size of 30.

5.2.1 Density

The four test wells contained different number of cells. Well C7 from A was the well that contained most cells, 20 000 compared to 10 000 in C3 from A and 5000 in C13 and D3 from B. From tab. 4.2 and 4.4, it can be seen that the number of false positives was considerably higher in well C7 than the rest of the test wells. The performance of the model thereby depends on the cell density of the well. Initially, it was thought that a high cell density was preferable due to that wells with higher density from image set A and B also contained more cell divisions, see tab. 4.1. For image set C, the number of cells in the wells is unknown, but based of the images they look like the 20 000 wells from image set A. When evaluating the network on image set C, the accuracy was lower than for image set A and B. In conclusion, images with lower cell density is preferable if the model is to be developed further.

5.2.2 Border cases

Some cases of cell divisions were not suitable examples. Cell divisions on the edges of the images were considered more complicated due to the presence of the frame and thus not a characterizing example. Also, examples in the beginning and end of a time sequence lack some information of the process. The results were adjusted accordingly afterwards to only include relevant examples. For instance, well B8 from A was removed since the only cell division in this sequence was close to the edge. Well B7 from A was also removed since only the annotated division in this sequence was considered a not fully completed division. For the rest of the time sequences, a total of seven cell divisions were removed.

The algorithm could be developed to include examples on the edges. As for now, the sliding window stops when it cannot be moved a full step to the side without part of it getting outside of the image. That means that a sliding window with step size of 30 could miss the last 29 pixels of the image. An alternative would be to move the sliding window the remaining pixels instead of neglecting the edges.

5.3 False positives

After training, the network did find cell divisions and the detections were not randomly distributed. For the fluorescent images, a large amount of the false positives could be understood. For example, brighter cells which had the appearance of an early cell division, cells that moved quickly or bright dust particles. For the fluorescent images, the reason to the false positives could not be explained simply. This was due to the complex homogeneous structure which did not give any possibilities to distinguish the features of a true cell division manually. However, the trained network managed to find cell divisions even in the bright field channel.

The main drawback of the model is the high number of false positives that is detected. We have identified some patterns of what part of the image that were most often classified wrong. The analysis was done with fluorescent images as the

cells are substantially more distinguishable with the human eye than in the bright field images.

When stabilizing the time sequence, a black frame was added around each image. When the black frame moves from one image to another, a cell close to the edge can be visible in one image and covered by the black frame in the next image. This disappearing and reappearing of cells triggered the network to classify edges as cell divisions. An example could be seen in fig. 5.4. The problem was partly solved by adding examples of edges to the training of the network, but some false positives remained.

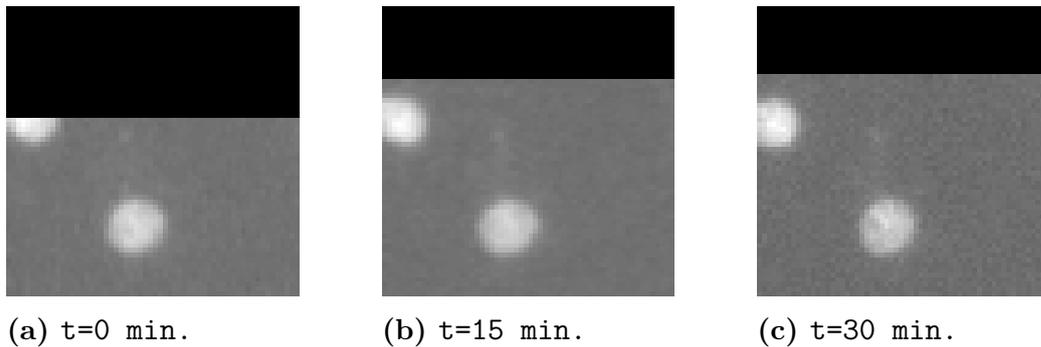


Figure 5.4: Example of images from time sequence C4 B, where an edge is classified as a cell division.

When observing the false positive classifications from the model, it seemed like the model learned to find movements of particles from one image to another. The result was that bright dust particles that were moving a lot in the time sequence were classified as cell divisions. An example could be seen in fig. 5.5 where the dust particle triggers the model to classify it as cell division. After extracting patches of falsely classified dust particles, the model improved but some hits still remained.

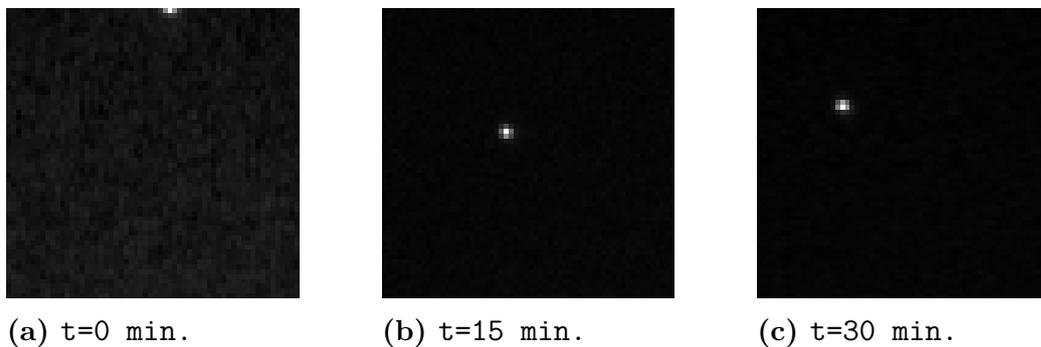


Figure 5.5: Example of images from time sequence C4 from B, where a dust particle is classified as a cell division.

Another pattern that could be seen was patches containing three cells that were false positives. An example can be seen in fig. 5.6. There is no obvious reason why

those examples were classified as cell divisions. It seems like they have little in common with patches of true cell divisions.

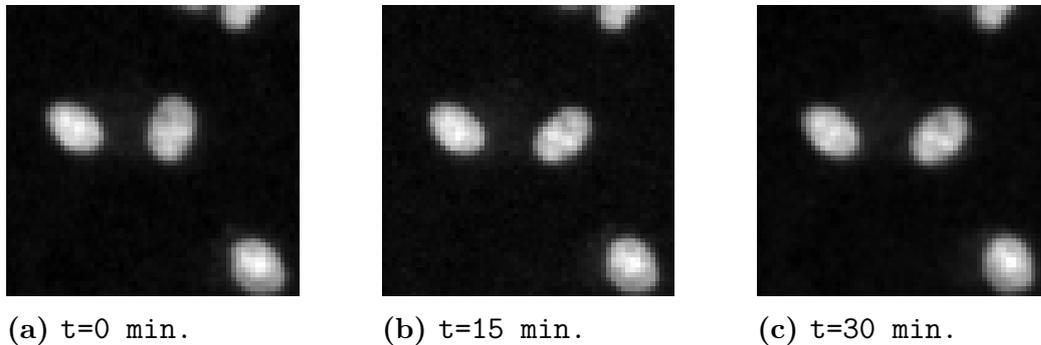


Figure 5.6: Example of images from time sequence C4 B, where three cells are classified as a cell division.

In the image sets A and B, there are several examples of groups of cells that are dying. It starts with that some cells get brighter and then get attracted towards each other, see fig 5.7a. The dying cells end up in a bright blob in fig. 5.7e. This process triggers the model to classify the dying cells as cell division. There are several similarities to a true cell division. The illumination, the movement and that there is more than one bright dot in the patch. Problems with dying cells could hopefully be avoided by adjusting the way the cells are cultured and stained.

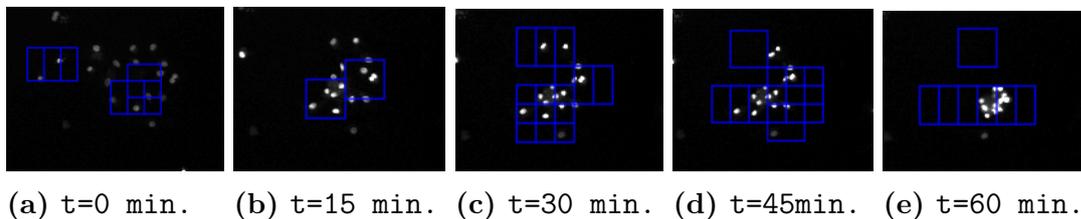


Figure 5.7: Example of images from time sequence C3 B, where a group of cells are dying.

Lastly, as mentioned in section 5.1.2, images rotating with respect to the previous image in the time sequence have a lot of false positives. It is probably once again because of a movement from one image to another.

Regarding the bright field images, patterns from the false positives are much harder to find. The true cell divisions are not as similar to each other as they are in fluorescent images. That makes a comparison between false and true positives difficult. What could be seen, however, is that problems with dust particles disappear. The reason is probably that dust particles are illuminated in the fluorescent channel but not in the bright field channel. Furthermore, some false positives along the edges are present in this channel too. There are also some examples of what probably is a cell that is moving that is classified as a cell division. In conclusion, some patterns

of the false positives might be able to detect. If not by us, the network could be able to see the pattern.

5.3.1 Active learning

The active learning principle with including false positives in the training set gave a significantly better result. The method was especially effective to reduce false positives on the edges of the images. As there was a pattern of what patches were classified as false positives in the training set, it was also likely that the model would do the same error in the test set. The fact that the number of false detections decreased also in the test set after active learning is a proof of this.

Active learning was considered suitable also for bright field images. Even though it was hard to find a pattern amongst the false detections, the network might catch something. It proved a success to use active learning in the bright field channel. From a total of 6294 false positives detected by the model in tab. 4.4 to 1324 false positives detected by model 1 (a) in section 4.3.2. The cost was that 20% of the cell division that was found before active learning were lost after. That is one of the reasons that ensemble learning was applied.

5.4 Ensemble learning

To eliminate the problem with high number of false positives and lost true positives, ensemble learning was applied in a small scale. Two different ensembles consisting of 6 respectively 7 networks were tried.

The difference between the networks was the weight initialization and, for some of the networks, the training examples. It was anticipated that the false positives would be different between the networks and the true examples would be detected by several of the networks. When evaluating the combinations of networks, it was shown that the number of false positives was largely reduced. On the other hand, the ensemble discarded several true divisions since too few of the networks had discovered them. The threshold of number of networks that discovered a detection was set to 3. This encountered the possible risk of two of the networks having very similar performance regarding the false positives. The discarding of true positives was an expected price to pay for the reduction of false positives. To consider the performance excellent and useful in its current state, it was desired that the number of false positives should not exceed the number of true positives. Even though the number of false positives remained high compared to the true positives, the ensemble learning was shown to be effective even in a small scale and the true positives was not discarded in the same pace as the false positives.

When evaluating the two different groups of networks before combining them in an ensemble, tab. 4.5, the number of detected false positives differed substantially. The networks with a training where the number of positive and negative examples were similar produced considerably more false positives. The examples of this training

also included smaller shifts and normalization over each image. It is possible that the total number of examples originated from too few different examples and thereby caused overfitting.

It would be of interest to investigate this on a larger scale. To do so, a higher number of networks should be implemented in the detection system. This would give a more general result not depending on one or two specific networks. However, the networks have to be different to not detect the same false positives. The level of how different they should be to obtain the best result is something that has to be evaluated in practice. The most basic level is our applied approach where the networks have the same structure, but the weights are initialized different before training. If this yields too similar result where the same false positives are detected, a larger difference must be made. This can be done by changing the architecture of the network. Either modify the layers but keeping the overall structure or have a completely different architecture on the network.

The ensemble learning was only applied on bright field images. Since the single network performed better on the fluorescent images, ensemble learning would probably give a good result on those images as well. Due to time limitations and that the main purpose was the bright field images, ensemble learning was not evaluated on fluorescent images.

5.5 Limitations of the model

During the development of the model, some limitations were discovered regarding the detection of proliferation.

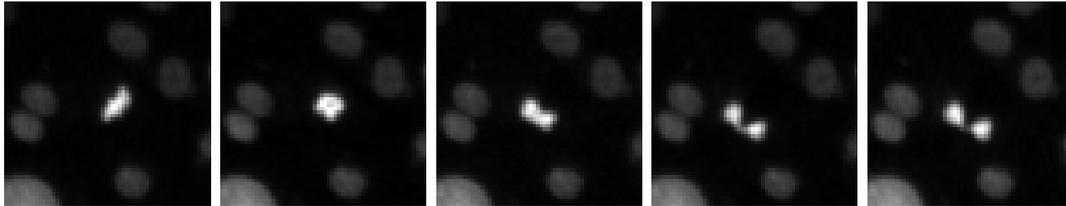
5.5.1 Finding dividing cells close to each other

Two or more cardiomyocytes could divide next to each other at approximately the same time. This is, to the best of our knowledge, not the case for the cell divisions used during the project. If it happens and some kind of ensemble learning or sorting is used, there is a risk that multiple divisions will be counted as one. If the model is used to detect cell division when a compound is added that will make more cells divide this could be a problem which is not investigated here.

5.5.2 Endoreduplication and acytokinetic mitosis

Adult cardiomyocytes can perform incomplete mitosis via endoreduplication and acytokinetic mitosis as described in section 2.1.2. For both variants, the genome is replicated and thus the cell nucleus will appear brighter in the fluorescent images. One question that arises is if the incomplete cell divisions were annotated as cell divisions. To see if a compound makes the cardiomyocytes regenerate, only true cell divisions should be counted.

In the example in fig. 5.8, the dividing cell does not have as distinct phases as in the example of mitosis in fig. 3.3. This example might be of a cardiomyocyte that undergoes acytokinetic mitosis. It cannot be seen from the images if the two sets of chromosomes are completely separated into two new cells or not. There might also be a possibility that the cell is dividing but then arresting and dying in anaphase. To be able to tell the difference between cell division and some other variant of a cell division, more knowledge about the appearance in fluorescent images is needed.



(a) $t=0$ min. (b) $t=15$ min. (c) $t=30$ min. (d) $t=45$ min. (e) $t=60$ min.

Figure 5.8: Example of a cardiomyocyte that possibly is undergoing acytokinetic mitosis.

Annotation of incomplete cell divisions in the fluorescent channel might lead to problems in the bright field channel. If a cell cycle variant does not look at all like a cell division in the bright field channel, some of the training examples could have been wrong. This is a larger problem in the bright field channel since in the fluorescent channel they would look like a true cell division. By excluding examples of cell cycle variants when training the bright field model, the result might improve.

With more knowledge about the difference between different cell divisions, a refined classification could be possible. After classifying images as cell division or not, patches of cell division could be fed to another model for classification into different types of mitosis. That would require a development of a model for cell cycle variant classification.

6

Conclusion

The objective of this thesis was to develop a neural network for detection of cardiomyocyte proliferation in time sequences with fluorescent and bright field images respectively. This was implemented by using the VGG16 network as a starting point. The network was able to distinguish most of the cell divisions, however, a high amount of false positives remained a problem. To reduce this and refine the model, active learning and ensemble learning was implemented. A cell division was considered true if enough networks in the ensemble had marked it. This proved successful in order to reduce the false positives. The drawback of the ensemble method was that some true positives were dismissed. In conclusion, we have shown that neural networks can be an efficient method to discover cell divisions. It is worth to investigate and optimize ensemble learning since it has shown to be a promising method.

Bibliography

- [1] E. Wilkins *et al.*, *European Cardiovascular Disease Statistics 2017*. Belgium: European Heart Network, 2017/02/15.
- [2] M. Hesse, A. Wetz, and B. K. Fleischman, “Heart regeneration and the cardiomyocyte cell cycle,” *Pflugers Arch*, vol. 470(2), 2018/02.
- [3] T. M. Mohamed, Y.-S. Ang, E. Radzinsky, P. Zhou, Y. Huang, A. Elfenbein, A. Foley, S. Magnitsky, and D. Srivastava, “Regulation of cell cycle to stimulate adult cardiomyocyte proliferation and cardiac regeneration,” *Cell*, vol. 173, no. 1, pp. 104 – 116.e12, 2018.
- [4] M. Held *et al.*, “Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging,” *Nature Methods*, vol. 7, 2010/08/08.
- [5] V. Ulman, “An objective comparison of cell-tracking algorithms,” *Nature Methods*, vol. 14, 2017/10/30.
- [6] G. Lukinavičius *et al.*, “Sir-hoechst is a far-red dna stain for live-cell nanoscopy,” *Nature Communications*, vol. 6, 2015/10/01.
- [7] J. G. Betts *et al.*, *Anatomy and Physiology*, vol. 15:2. OpenStax, 2019.
- [8] G. Olivettia *et al.*, “Aging, cardiac hypertrophy and ischemic cardiomyopathy do not affect the proportion of mononucleated and multinucleated myocytes in the human heart,” *Journal of Molecular and Cellular Cardiology*, vol. 28(7), 1996/07.
- [9] M. J. Ivey and M. D. Tallquist, “Defining the cardiac fibroblast,” *Circulation journal : official journal of the Japanese Circulation Society*, vol. 80(11), 2016/10/25.
- [10] A. Sturzu *et al.*, “Fetal mammalian heart generates a robust compensatory response to cell loss,” *Circulation*, vol. 132(2), 2015/07/14.
- [11] E. R. Porrello *et al.*, “Transient regenerative potential of the neonatal mouse heart,” *Science*, vol. 331(6020), 2011/02/05.
- [12] S. van den Heuvel and E. T. Kipreos, “Chapter 9 - c. elegans cell cycle analysis,” in *Caenorhabditis elegans: Cell Biology and Physiology* (J. H. Rothman and A. Singson, eds.), vol. 107 of *Methods in Cell Biology*, pp. 265 – 294, Academic Press, 2012.
- [13] T. M. A. Mohamed *et al.*, “Regulation of cell cycle to stimulate adult cardiomyocyte proliferation and cardiac regeneration,” *Cell*, vol. 173, no. 1-13, pp. 104–116, 2018 Mar 22.
- [14] P. C. Hsieh and T. J. Kamp, “To be young at heart,” *Cell Stem Cell*, vol. 22, no. 4, pp. 475 – 476, 2018.
- [15] NCBI, “Fgf2 fibroblast growth factor 2 [homo sapiens (human)],” 2019. [online] <https://www.ncbi.nlm.nih.gov/gene/2247#gene-expression> 2019/04/18.

- [16] N. Rosenblatt-Velin *et al.*, “Fgf-2 controls the differentiation of resident cardiac precursors into functional cardiomyocytes,” *The Journal of clinical investigation*, 2005/07/01.
- [17] O. Sen, A. T. Saurin, and J. M. G. Higgins, “The live cell dna stain sir-hoechst induces dna damage responses and impairs cell cycle progression,” *Scientific Reports SP*, vol. 7898(1), 2018/05/21.
- [18] G. A. Romar, T. S. Kupper, and S. J. Divito, “Research techniques made simple: Techniques to assess cell proliferation,” *Journal of Investigative Dermatology*, vol. 136(1), 2016/01.
- [19] F. Thorsten *et al.*, “U-net: deep learning for cell counting, detection, and morphometry,” *Nature Methods*, vol. 16, 2019/01/01.
- [20] N. Meng, E. Lam, K. K. M. Tsia, and H. K.-H. So, “Large-scale multi-class image-based cell classification with deep learning,” *IEEE Journal of Biomedical and Health Informatics*, 2018/10/31.
- [21] L. Jean-Baptiste *et al.*, “Identification of individual cells from z-stacks of bright-field microscopy images,” *Scientific Reports*, vol. 8, 2018/07/30.
- [22] H. Niioka, S. Asatani, A. Yoshimura, H. Ohigashi, S. Tagawa, and J. Miyake, “Classification of c2c12 cells at differentiation by convolutional neural network of deep learning using phase contrast images,” *Human Cell*, vol. 31, no. 1, pp. 87–93, 2018/01/01.
- [23] M. Saha, C. Chakraborty, I. Arun, R. Ahmed, and S. Chatterjee, “An advanced deep learning approach for ki-67 stained hotspot detection and proliferation rate scoring for prognostic evaluation of breast cancer,” *Scientific Reports*, vol. 7, 2017/06/12.
- [24] M. Veta *et al.*, “Predicting breast tumor proliferation from whole-slide images: The tupac16 challenge,” *Medical Image Analysis*, vol. 54, pp. 111 – 121, 2019.
- [25] K. Paeng *et al.*, “A unified framework for tumor proliferation score prediction in breast histopathology,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, (Cham), pp. 231–239, Springer International Publishing, 2017.
- [26] B. Mehlig, “Artificial neural networks,” *CoRR*, vol. abs/1901.05639, 2019.
- [27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [28] A. L. Maas, “Rectifier nonlinearities improve neural network acoustic models,” *Computer Science Department, Stanford University*, 2013.
- [29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *The IEEE International Conference on Computer Vision (ICCV)*, 2017/10.
- [30] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” *CoRR*, vol. abs/1708.02002, 2017.
- [31] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.
- [32] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 2014/12.

- [33] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation,” vol. 4304, pp. 1015–1021, 2006/01.
- [34] A. Kensert, P. J. Harrison, and O. Spjuth, “Transfer learning with deep convolutional neural networks for classifying cellular morphological changes,” *bioRxiv*, 2018.
- [35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv 1409.1556*, 09 2014.
- [36] I. Arganda-Carreras, “Register virtual stack slices,” 2018. [online] https://imagej.net/Register_Virtual_Stack_Slices 2019-01-30.
- [37] A. Kensert, P. J. Harrison, and O. Spjuth, “Transfer learning with deep convolutional neural networks for classifying cellular morphological changes,” *SLAS DISCOVERY: Advancing Life Sciences R&D*, vol. 24, no. 4, pp. 466–475, 2019.
- [38] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, 2015/06.
- [39] K. Documentation, “Usage of initializers,” 2019. [online] <https://keras.io/initializers/> 2019/05/21.

