USING FASTER RCNN FOR CELL IMAGE DETECTION

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

ARI GJERAZI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY, 2021

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"Using Faster RCNN for cell image detection"** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____

Assist. Prof. Dr. Arban Uka
Head of Department

Date: July, 16, 2021

Examining Committee Members:

Dr. Julian Hoxha (ComputerEngineering) _____

Assist. Prof. Dr. Arban Uka (Computer Engineering) _____

Dr. Shkelqim Hajrulla (Computer Engineering) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Ari Gjerazi

Signature: _____

ABSTRACT

USING FASTER RCNN FOR CELL IMAGE DETECTION

Gjerazi, Ari

M.Sc., Department of Computer Engineering

Supervisor: Dr. Arban Uka

There is an ever-growing need for automated detection (and classification) of microscopic images containing cellular samples. To this end, the focus of this work is to provide a method of performing this detection: through the implementation of a Faster RCNN model. The .tiff images for training and testing are fed to the network, with various hyperparameter adjustments between runs, and the anchors/bounding boxes are calculated by FRCNN. Four separate output loss functions are calculated and then unified for a final metric. The network utilized an underlying VGG-16 architecture, and a RPN (Region Proposal Network) which is responsible for the aforementioned bounding boxes. The architecture is run on keras (tensorflow backend).

Keywords: *cell images, detection, classification, Faster-RCNN, bounding box, Region Proposal Network.*

ABSTRAKT

PERDORIMI I FASTER-RCNN PER DETEKTIMIN E QELIZAVE

Gjerazi, Ari

Master Shkencor, DepartamentiiInxhinierisëKompjuterike

Udhëheqësi: Dr. Arban Uka

Ekziston nje nevoje gjithmone e me ne rritje, per detektim te automatizuar te imazheve mikroskopike te cilat konsistojne ne mostra qelizore. Per kete qellim, fokusi i kesaj pune eshte te ofroje nje menyre per te kryer kete lloj detektimi: permes implementimit te nje modeli Faster-RCNN. Imazhet .tiff per trajnim dhe testim, i jepen rrjetit neural, me optimizime te hiperparametrave midis testimeve, dhe spirancat/kutite kufizuese llogariten nga FRCNN-ja. Kater funksione te ndryshme ne lidhje me outputet llogariten dhe me pas bashkohen per nje metrike te fundit. Rrejta neurale perdor nje arkitekture VGG-16 nen pjeset e tjera, dhe nje RPN (Rrjet Propozimesh Regjionale) i cili eshte pergjegjes per kutite kufizuese te permendura me pare. Arkitektura perdor keras (tensorflow backend).

Fjalëtkyçe: *imazheqelizash, detektim, klasifikim, Faster-RCNN, kutikufizuese, RPN*

This thesis, I dedicate to my family, for their appreciation and support, to all of my professors who have guided and advised me during my time as a student, and to my friends and colleagues, who have shared in the same experience.

ACKNOWLEDGEMENTS

I would like to, first and foremost, thank my thesis supervisor, Assist. Prof. Dr. Arban Uka. Not only has his guidance been invaluable, but he has also offered me advice on the best scientific material to peruse for references, as well as instructed me carefully and to great effect, on how to properly assemble the numerous pieces and put together my thesis, for which I am once again, most grateful.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

As we move further towards the development of artificial intelligence and its varied subfields, we entrust more and more data to such artificial networks, which allow us to swiftly analyze, classify or detect information (a particularly rare aspect for automation), from images that are fed to the network in question [1]. In fact, it is the convolutional neural network that is most suitable for treating these problems (image analysis). Building off this point, it is understandable that a body of work which takes into account cell detection, would quickly turn to CNNs for answers.

One such alternative is found in the FRCNN model, which is the focus of this thesis. This is not a work without challenges: images can have various innate problems, such as improperly displaying cells, overlapping nuclei (which are the cusp of our detection algorithm as we try to focus them: this is based on the notion that each cell has a single nucleus, and thus a cell count would correspond to a nucleus count), bad quality of the images in question and even challenges with the network itself, when it comes to optimizing and running the network model as well, or even fitting it to one's own needs. [2]Thus, the approach to take is two-pronged, both on the front of image retrieval and processing and also on the neural network model that is to be utilized here.

.

1.2 Objectives

This project focuses on training a neural network model, to derive a total cell count out of what is present in an image, having been trained with such images prior. Following its training phase, it will be possible for the network to determine the cell count on its own. Before reaching this point, the project also aims to optimize image

pre-processing as well as select an appropriate (or optimal with the given circumstances) network model.

1.3 Organization of the thesis

The thesis is separated into several subsections. The first section goes over the introduction to the thesis, outlining a brief intro, objectives and this very paragraph on thesis organization. The second section details the literature review, going from macro and microscopic imaging techniques, to the intricacies of neural networks, and finally to the specific model to be used (FRCNN). The third chapter details the methodology: the exact approach utilized and further in the fourth chapter, results are derived from the work done. Lastly, a discussion of the results derived in section four, is displayed in the fifth section.

CHAPTER 2

LITERATURE REVIEW

2.1 Techniques used for Image Gathering

This section will outline various technologies used to acquire medical images for the purposes of this thesis. Some of these technologies are based around taking photographs of specific organs, while others focus on the microscopic imaging aspect, and this thesis will eventually veer towards that for its research aspects.

2.1.1. MRI

Various imaging methods can be applied to retrieve data from a specific part of the body. For brain imaging we have two primary methods: MRI and CT scans. MRI stands for Magnetic Resonance Imaging which involves the use of magnetic fields to retrieve images of human anatomy. [3]

The way this works involves hydrogen nuclei (known to consist of a single proton) sending out signals, which are collected and processed in terms of a density mapping of these nuclei in a certain region of the body. It must be noted that these hydrogen nuclei also interact with other atoms, and in turn separate the hydrogen respounces in specific compounds.

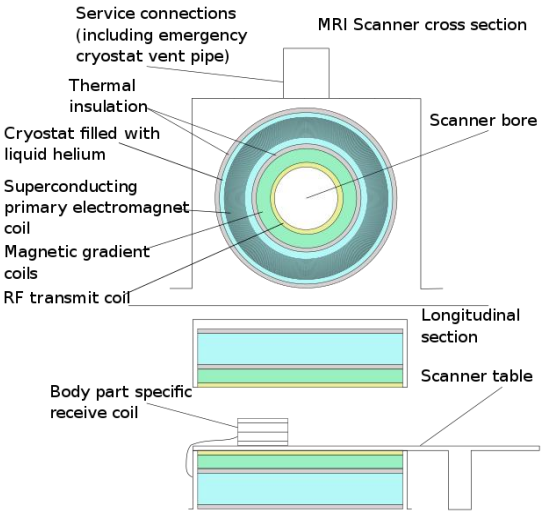

Figure 1.MRI Diagram

The above figure schematizes the inner workings of an MRI scanner. There are certain settings that a MRI image can be derives, T1 and T2 weighted, which involves the display below:

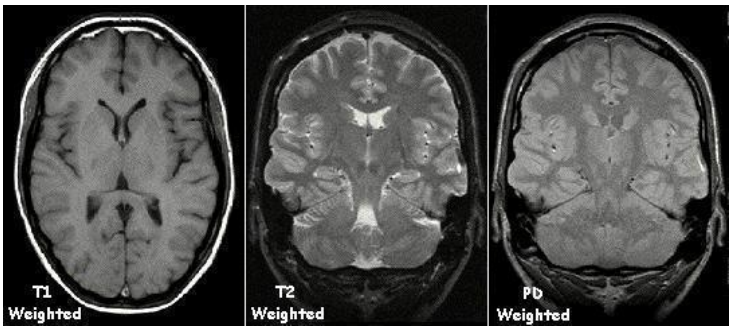

Figure 2.T1 and T2 Weighted Cells

By focusing various cell types and liquids, we can obtain a different view of various parts of human anatomy, as in the above example, the human brain. The selection of these weights is likely up to the specific application, and it is possible that several different weights may be needed. [1]

One of the MRI-using studies mentioned in the papers collected for reference, lists the following methods: “Imaging studies were conducted on 1.5- or 3-Tesla MRI. The multicenter nature of the study and the various clinical presentations did

not allow standardization of sequences. The most frequently sequences performed were:

- 3D T1-weighted spin-echo with and without contrast-enhanced imaging
- Diffusion-weighted imaging (DWI)
- Gradient Echo T2 or Susceptibility-weighted imaging
- 2D or 3D FLAIR postcontrast
- 3D TOF MR angiography of the circle of Willis

There are a number of conditions that MRI is known to be able to discover multiple forms of cancer. Among them are various brain cancers, rectal cancers, prostate cancers, soft tissue tumors and more. Other conditions that can be detected are joint diseases, myocardial ischemia, cardiomyopathies, other vascular conditions, demyelination, dementia, cerebrovascular disease, Alzheimer's and epilepsy.

2.1.2 CT scan

CT stands for computer tomography, known more commonly as a CAT scan, Computed Axial Tomography. CT scan makes use of a multitude of X-ray images, taken at different angles and then followed by a computation of a combined image, a tomography (cross-section) of the body. One of the functions of this technology is of course, the ability to perform a tomography of the head. In fact, it is the fusion of our two goals: X-Ray imaging and brain imaging. While skull injuries and similar cases do not particularly interest us, brain imagery is something we would like to observe and analyze:

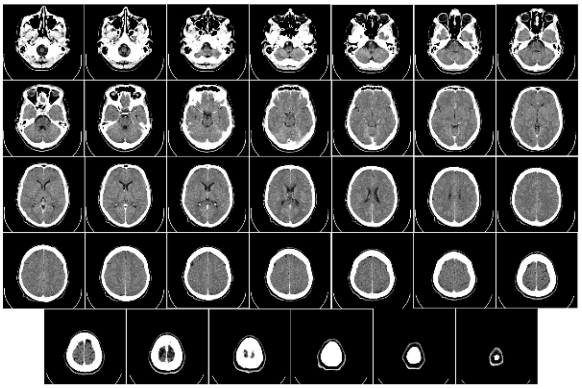

Figure 3.Various CT head images

As we can observe in this picture, there are a multitude of various images we can derive from the head alone.Now, there is a certain comparison to be made between the two imaging methods. MRI is typically preferred over CT Scans, for a variety of reasons:

- MRI will prove itself to be better at deriving tissue contrast than CT scans.
- It produces fewer artifacts when viewing brainstem
- It is superior when it comes to pituitary imaging.
- It is somewhat less effective at detecting early cerebritis
- In a case of concussion, MRI is best avoided, unless symptoms grow more severe.
- MRI will prove superior to CT scanning when it comes to headaches, as well as various other conditions, such as neoplasm or vascular disease presence.
- Typically, traumatic brain injuries will involve a preference for CT.
- CT scanning will prove better in the evaluation of intracerebral calcifications.

The above weighted methods for brain MRI are also listed below:

- T1 weighted images will show cerebrospinal fluid as dark. Useful for the visualization of otherwise normal anatomy.

- T2 weighted images will show the cerebrospinal fluid as white, but fat in a darker color. This includes white brain matter. Pathology is best viewed with T2 weighted images.
- Diffusion Weighted Images: contrast generation via the diffusion caused by water molecules.
- Proton Density (PD) images: Cerebrospinal fluid which is dense in protons, will appear brighter. Grey matter will also appear brighter than white matter.
- FLAIR (Fluid Attenuation Inversion Recovery): Demyelination is best detected using this method.

Beyond this, it is likely that most brain images will involve MRI scanning, however if the need arises, both methods can be used. One particular function of CT scans is in covid detection. CT scans There are indeed a number of other techniques used for neuroimaging (another term for brain imagery). They will be briefly described below for the sake of reference, although their use may not be as prominent in the context of this research.

2.1.3 Diffuse Optical Imaging (DOI)

The acronym DOI refers to this technique. It is a method that employs Near-Infrared Spectroscopy (NIRS. More on it further). The methods used here are fluorescence-based. The specific terms used for 3D and 2D images (volumetric in the case of three-dimensional imaging) are respectively diffuse optical tomography and diffuse optical topography. This is mostly a semantic issue. The basis for Diffuse Optical Imaging techniques is the observation of changes of hemoglobin, both oxygenated and deoxygenated. If needed, it may take a measure of redox states (of cytochromes). There are other terms for DOI techniques, the above tomography, as well as near-infrared optical tomography (which was referenced earlier, and features as a separate technology) as well as FDOT (Fluorescent Diffuse Optical Tomography), although the exact term varies based on the type of usage.

Depending on these measurements however, it is possible the aforementioned techniques may be folded in with functional Near Infrared Spectroscopy (fNIRS) discussed further below.

2.1.4 Event Related Optical Signal (EROS)

EROS makes use of infrared light, transmitted through the optical fibers in order to measure alterations in the optical properties of various regions in the cerebral cortex. The basic principle involves the detection of infrared light scattering, which is of course associated with neural activity.

There is a difference between EROS as well as DOI and NIRS techniques: DOI and NIRS will make use of the optical absorption property of hemoglobin, which necessitates a measurement based on the cerebral blood flow, EROS instead focuses on the scatter property of the neurons themselves.While this does make EROS a good provider of precise information and also good spatial and temporal resolution: being able to pinpoint activity down to millimeters or milliseconds, it still proves difficult to obtain images from more than a few centimeters of depth. EROS has a few practical advantages: it is inexpensive, non-invasive and it also works well together with other techniques, such as fNIRS or fMRI.

2.1.5 Functional Magnetic Resonance Imaging (fMRI)

A method of measuring brain activity by making use of the change in blood flow (and its subsequent detection). The basis of fMRI is the coupling of blood flow and neuron activity: the neuron activity signifies use of that region of the brain, which further leads to blood flowing into that same region.

A key concept related to the working of fMRI is BOLD (Blood-Oxygen Level Dependent) contrast: a technique that involves mapping neural activity through the relation of hemodynamic response and energy used by individual brain cells.

One reason why it is viewed as particularly effective or desirable to use, relates to the lack of invasive procedures and especially the lack of introduction to ionizing radiation (a common concern with CT scanning). Beyond this, fMRI is attractive to use in research and is further complementary with EEG and NIRS. It also attracts a lot of research itself, into various optimizations related to spatial and temporal resolution.

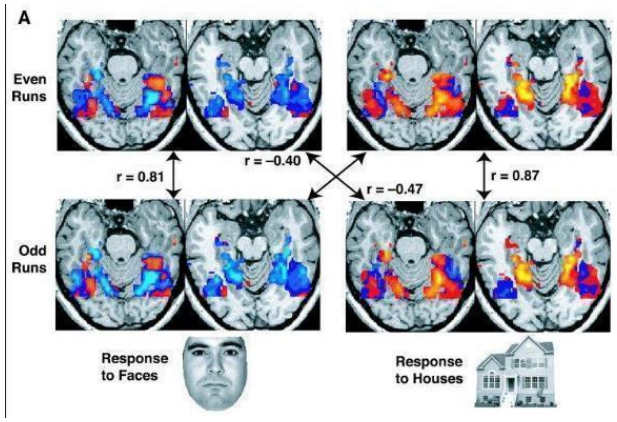

Figure 4. fmRI Images

These fMRI images are from a study showing parts of the brain lighting up on seeing houses and other parts on seeing faces. The 'r' values are correlations, with higher positive or negative values indicating a stronger relationship (i.e., a better match).

2.1.6 Magnetoencephalography (MEG)

A functional neuroimaging technique, MEG uses the magnetic fields which are produced through the electric currents naturally occurring in the brain. There are certain technologies involved in MEG, such as SQUID (superconducting quantum interference devices) forming an array are considered a popular choice. Another variant is the SERF (Spin Exchange Relaxation-Free) magnetometer which is still under investigation for future use.

One of MEG's primary uses is research into cognitive and perceptual functions of the brain. Other functions involve simple measurements of brain activity, or searching for specific abnormalities, which finds more application in a medical setting.

2.1.7 Near Infrared Spectroscopy (NIRS)

NIRS makes use of the near infrared region of the electromagnetic spectrum (780~2500 nm). The theoretical foundation for NIRS is that molecular overtone and combination bands seen in the near-IR are typically very broad, leading to complex

spectra. Assigning specific features to specific chemical components however, does prove difficult.

Some of the main techniques for extracting the desired chemical information involve multivariate calibration. Once more, the same concept of brain activity being associated with increased blood flow (the neural activity - blood flow coupling). As mentioned above, it serves as a key component for the functionality of other technologies.[1]

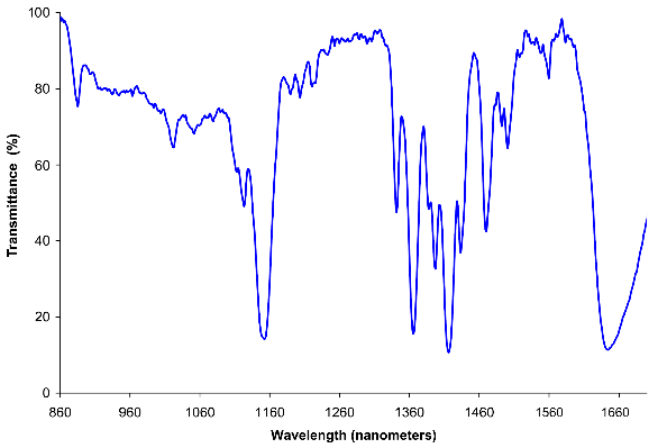

Figure 5. Transmittance and wavelength relation

2.1.8 Positron emission tomography (PET)

“PET is a functional imaging technique that uses radioactive substances known as radiotracers to visualize and measure changes in metabolic processes, and in other physiological activities including blood flow, regional chemical composition, and absorption. Different tracers are used for various imaging purposes, depending on the target process within the body**.”**

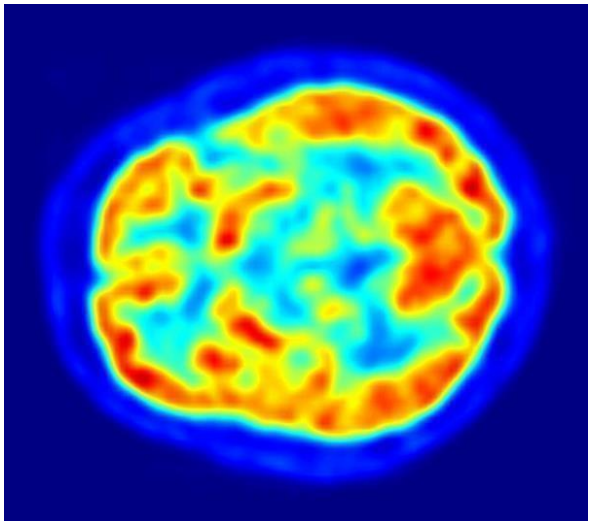

Figure 6. PET Scan

2.2 Optical Microscopy

Beyond the typical medical imaging technologies, we also have to consider imaging for microscopic organisms, such as cells. The focus of this work is indeed based on cellular observation, and therefore microscopic imaging is particularly important.

Lighting in optical microscopy also differs: transparent objects are usually lit from underneath, whereas for solids there is a choice between brightfield and dark field imaging, the two of which will be examined below. Further still, there are several different types of microscopy to be covered here also.In general, optical microscopy involves the usage of lenses and light of the visible spectrum, to produce magnified imaging of a specific sample. It includes placing the object on top of a stage, and then viewed directly under magnification from the microscope.

One of the more significant aspects that are taken into account are the different illumination techniques available.

2.2.1 Brightfield Microscopy

Brightfield Microscopy is comparatively straightforward. The image produced will have a white background and the sample within it will be black. The process involves the emission of bright light from underneath the sample, to reach the microscope and then be magnified and visible. Technically, it is one of the simplest variations possible for illumination. The light pathing is also largely unaltered from the typical setup of an optical microscope. A light source, the condenser and objective lenses as well as the device (such as a camera) with which to view the sample. [2]

It shares a few advantages such as being fairly simple to set up and produce images with, as well as having fairly good visibility for most kinds of living cells, useful in the field of medical imaging. On the other hand, it also comes with certain disadvantages, such as contrast being quite low, as well as magnification having a cap of sorts, which means that more precise imaging is sadly, beyond the scope of this technology. Other issues include problems with resolution being low, as well as the fact that it does not properly (or at all) display transparent samples, based on the straightforward method of illumination behind it (although this can be fixed to some extent, with artificial coloring of the samples). [1]

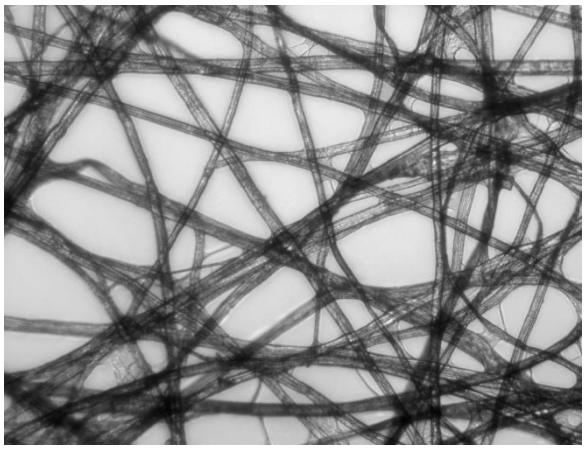

Figure 7. Brightfield microscopy on paper fiber

2.2.2 Darkfield Microscopy

Dark-field Microscopy is also applicable to electron microscopy. The base concept behind it is the exclusion of the unscattered light beam from the image, producing a different type of image, which should look somewhat like an inversion of the brightfield alternative.Components that are a part of a darkfield microscope include a special disc or patch which prevents some of the light from going through, and producing a ring-shaped perimeter of light around the image, the condenser lens as usual for optical microscopy, and then the filtering between the scattered and unscattered light occurs, following the entrance of the light through the sample.The ultimate result is that only scattered light makes it into the image itself.

Dark Field Microscopy is also very functional when it comes to live samples, such as the ones involved in medical imaging cases. One advantage it shares with brightfield microscopy is the simplicity behind it's setup. Certain artifacts are entirely eliminated, although as will be demonstrated below, this does imply certain other disadvantages. Being the reverse (process) of brightfield microscopy, it acts as a sort of high-pass alternative, regarding the structure behind the image. Note, that a dark field image is not exactly a negative of the brightfield image, mostly in regards to the formation of shadows.

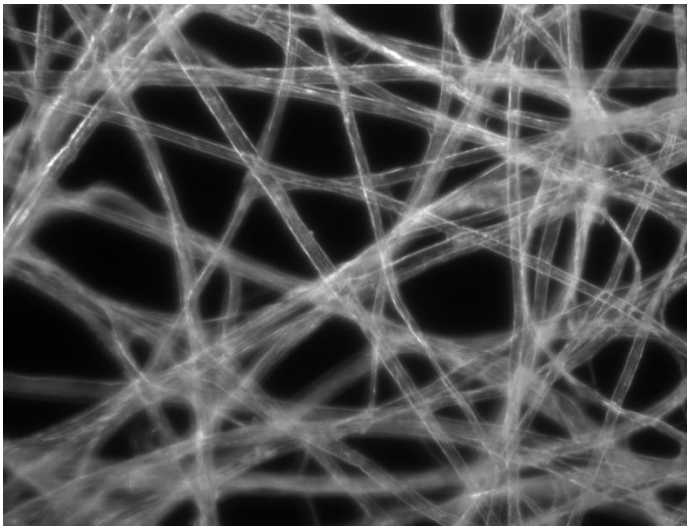

Figure 8. Dark field microscopy on paper fiber

2.2.3 Cross-polarized light microscopy

Cross-polarized light microscopy is another option to work with. The basic idea is, as the name suggests, illumination of the sample using any variation of polarized light. Other variations of this technique exist, which are, however, more complex. [1]

It does not have tremendous application in the medical field, being more common in mineralogy. The nature of the sample being viewed is also of great importance, the index of refraction being of a specific range and kind, which is affected by polarization (a property known as birefringence).

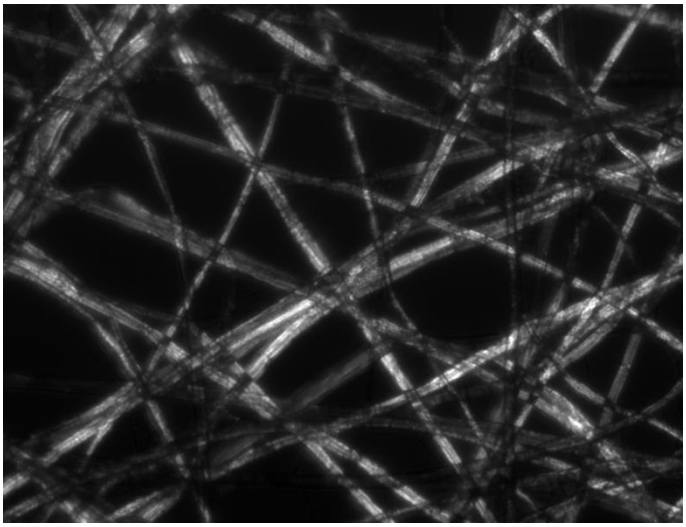

Figure 9. Cross polarized light microscopy on paper fiber

2.2.4 Phase-contrast microscopy

Phase-Contrast Microscopyis the fourth main method to be considered here. It tends to operate under the principle of converting phase shifts regarding the specimen into brightness changes. Normally, operators do not perceive phase shifts (as they are invisible), however this method highlights their contrast by displaying them as brightness changes instead.This working principle proceeds with the separation of the background illumination and the light scattered by the sample, and then proceeding with image manipulation (done differently for each).

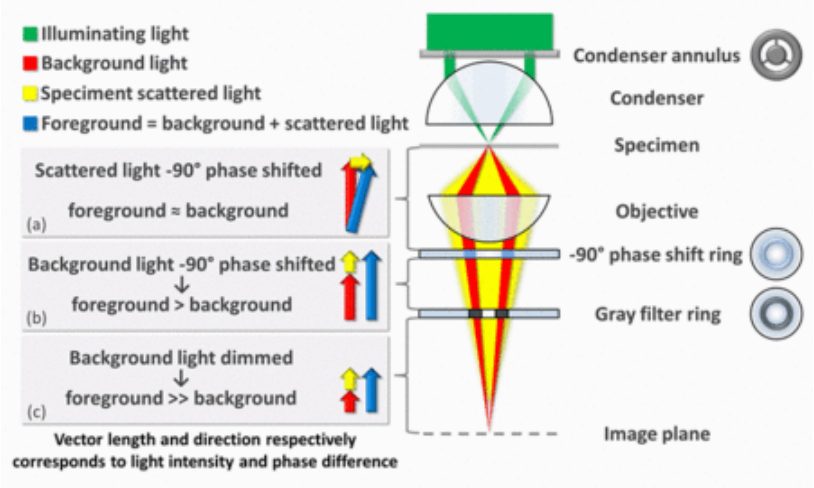

Figure 10. **Phase contrast microscopy workings**

One of the most significant applications of phase-contrast microscopy is related to biology and medical imaging. It is, in certain aspects, superior to bright-field microscopy, being able to reveal certain image aspects that do not normally appear in the former technique. This technique has made possible many discoveries in the field of cell observation, including the observation of cell division. It is an accepted method that does not make use of fluorescence, yet is able to observe various cellular formations nonetheless.

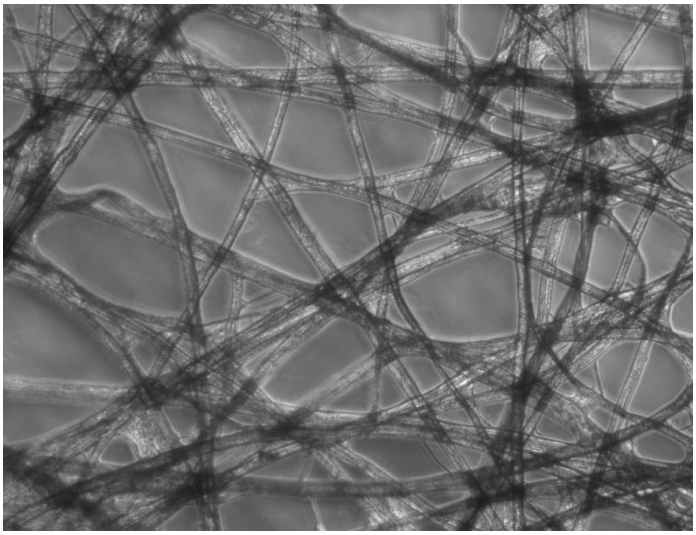

Figure 11. Phase Contrast Microscopy on Paper Fiber

2.3 Learning Models

The idea of learning when applied to a neural network (or any machine learning-capable model) involves the constant improvement over a number of iterations that eventually lead to an optimized output, towards either regression or classification (although the exact desired output is to be adjusted to one's needs). [**Error! Reference source not found.**]

To this end, there are a number of parameters and paradigms that we must consider. First and foremost, the main 'components' of learning. [2]

2.3.1 Introduction to Machine Learning

One of the primary questions to ask when approaching any particular method for research, is that of purpose. Why is machine learning useful? Why are these models implemented? There is a combination of large datasets, and an extensive study regarding the interconnection between human perception and image analysis, and the same analytical concept carried out by machine intelligence. [**Error! Reference source not found.**]

There exists a wealth of learning models that have been developed and continue to be developed for the express purpose of conducting fast, precise and efficient image analysis. The paragraphs below will look over the various features of such models and how they interact.

2.3.2 Learning Rate

Learning Rate is a value which determines the exact quantity of correction incurred between each step of training a model. The main measures of learning rate tend to be speed and accuracy. A higher learning rate will lead to a quicker convergence and the final output is derived sooned. On the other hand, a smaller learning rate will consume more time and likely more computational resources, but

with it comes a more accurate final prediction (as the threshold for correctness is shrunk closer towards the expected result). There are many ways through which optimization occurs.Some optimizations tend to focus on one aspect or the other: either quickening the rate of error minimization or otherwise attempting to make faster predictions also more reliable. A middle-ground is found with adaptive learning rate: increasing and decreasing as necessary. A further refinement involves weighing the collective of previous learning (gradient) and the most recent change: assigning priority towards one or the other. This concept is also known as momentum. [**Error! Reference source not found.**] [[2]

2.3.3 Error Function

The error function is an important part of training a neural network. Also known as a loss function or cost function. The purpose of such a function is mapping the inputs presented to a numerical weight. From the perspective of the one running the model, the error function must be minimized.The exact nature of the loss function varies based on the problem: it generally follows the specifications of the problem (what is required to be optimized).Typical choices for loss are squared and absolute loss functions. As the loss function is preferred to be continuous (as we need data on any point) and differentiable (for maximization/minimization), absolute loss has one minor issue, of not being differentiable at the 0 point. [**Error! Reference source not found.**]

On the other hand, particularly large values of loss in the square function will likely influence the total sum more than may be necessarily indicative of the true situation regarding loss. [2]

2.3.4 Backpropagation

Another important concept is that of backpropagation. Backpropagation assists in the fitting (aligning the classification the model makes, to best classify test data). Backpropagation functions by computing a gradient for the error function (taking weights into account) for a single pair of input and output.

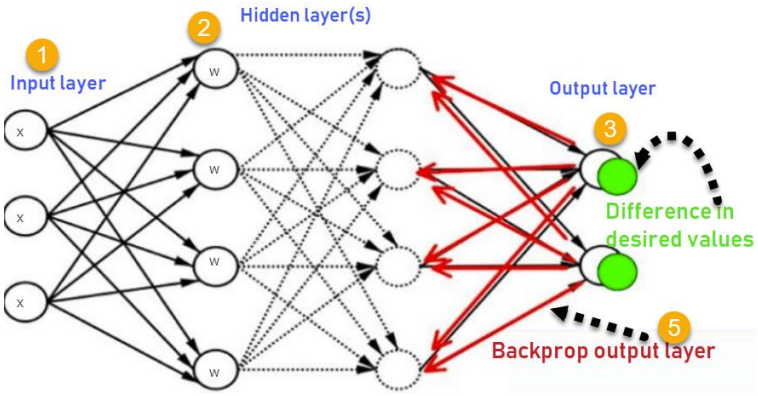

Figure 12. Back Propagation Layer

There are several components to a backpropagation algorithm. They are denoted below:

- Input, expressed as a vector
- Output (a series of probabilities in the case of classification)
- The error function
- The number of layers
- The weights between the current and previous layer, as well as between specific nodes in each layer
- The activation function at the present layer. [2]

If we are to consider a feed-forward NN, then we can view backpropagation through the lens of matrix multiplication. In this situation, backpropagation derives the derivative of the error function, proceeding opposite to the layer progression (so left to right), the gradient calculated in the process also known as backwards propagated error.[10][2]

2.3.5 Supervised Learning Paradigm

One of the main learning paradigms is the one known as supervised learning. It consists of having an existing input with the associated correct outputs, which allows the model to learn over time through the observation of its own outputs and the correct ones. There are two main purposes to Supervised Learning.

Classification is a process which separates the input data according to certain categories. To this end, the output is a probability which assigns the input a certain likelihood to being within a certain class. There are several important classification algorithms, from linear classifiers, to SVMs (Support Vector Machines), KNNs (K-Nearest Neighbors) and even random forest classifiers.

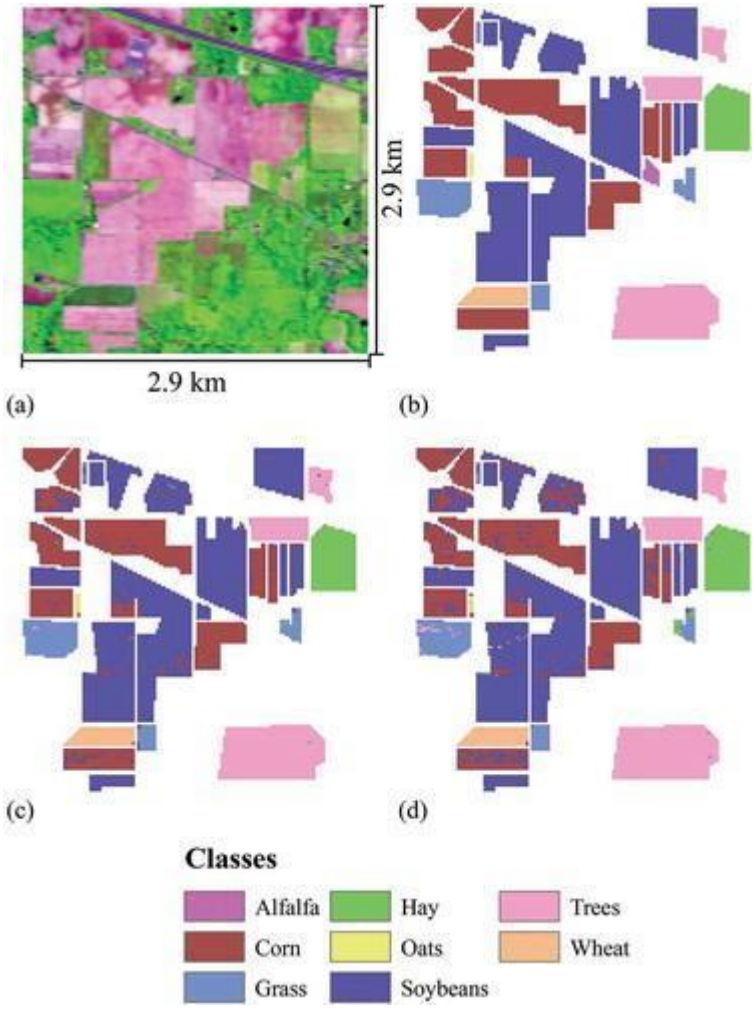

Figure 13. Classification

Regression on the other hand is a process which focuses on determining the relationship between variables. Rejection is used to make future predictions regarding a certain output given the trends displayed by prior inputs. There are different kinds of regressors: linear, logistical and polynomial regression.

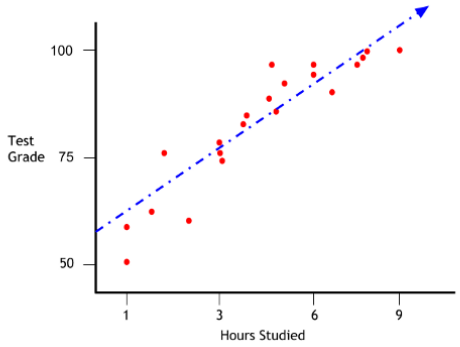

Figure 14. Regression

2.3.6 Unsupervised Learning Paradigm

The unsupervised learning paradigm consists of feeding the network a set of inputs, some output, a function of the input data as well as the error function. The core concept associated is that the machine should be able to determine a connection between inputs and outputs, clustering the data along a certain model. [7]

2.4 Convolutional Neural Networks

One of the methods employed involves machine learning, specifically the use of neural networks to detect the presence of cellular matter in an image and further still to be able to properly account for the number of cells visible. [3]

2.4.1 Convolutional Neural Networks Basics

Convolutional Neural Networks are a form of Neural Network that is often applied to image classification. Conceptually, it is a model that attempts to align the perception of images by the machine, to that of humans. This can be witnessed in the general nature of CNNs, the very name (neural) referring to the structure of the human brain.Indeed, Neural Networks in general attempt to simulate the connections (and pattern) of neurons in the human brain. As a basis, it takes an image input and by weighting the importance of certain aspects and mapping out spatial/temporal

relationships between said aspects as well, it provides detection or classification functionality. [4]

Normally, we have to consider the alternative that is, simpler Neural Networks (feed forward for example). It must be noted however that such networks would turn a nxn array into a n^2x1 column vector and analyze the data as is. This would compromise a lot of the features of an image that are used for classification, such as the positioning of various elements for example, that would likely be entirely lost for more complex images (with the associated pixel values). A feed forward NN would fail to properly detect the necessary features or dependencies of an image, which a CNN can. [5]

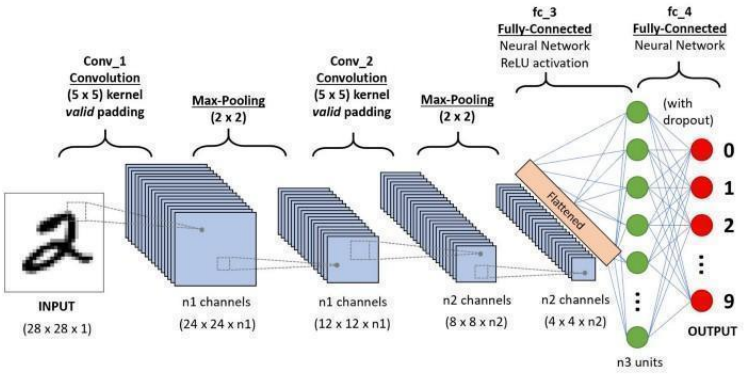

Figure 15. Classification

2.4.2 Convolutional Layer

A CNN will receive an input image, of a certain height x width set of dimensions and the input channels as well (keep in mind an image also has to have a consideration of the three primary colors as separate channels: each image within its own channel is of course, in grayscale). For example a square RGB image would be of the parameters n x n x 3 where n corresponds to the height/weight of the image in pixels. [3]

This input is convolved with a feature map. A feature map is also of the same types of parameters, height x width x channels. The name convolutional layer comes

thus, from the interaction of this feature map (also called an activation map) and the base image itself, as the later is convolved with the former. [3]

The map ‘slides’ over the input image, the corresponding (positionally) pixel values undergoing matrix multiplication, and the final output being the convolved image (across multiple channels, depending on the image itself).This is also called a Kernel or filter. Other than the specific height and width parameters of both the image and the filter, there are other parameters to take into consideration as well, for example stride, padding or even dilation. Stride represents the amount of pixels the kernel moves after each multiplication (1 at default, although it can be different). Padding is another such parameter. It can have an effect on the dimensionality of the input image (and its changes following convolution). One consideration would be a type of padding where only one row and column is added, taking a n x n (without channels considered) image to (n+1) x (n+1). When convolved with a m x m matrix, it will produce an n x n image in turn. Note that in say, a n x n x m convolution, m corresponds to the number of kernels the image is convolved with. One of the primary functions of this process is extraction of significant features (edges come to mind). This is mostly noticeable within the first layer, however it is notable that CNNs are *not* limited to a single convolutional layer, and further convolution can expose a number of other, more complex features as well as the base ones. [3]

2.4.3 Pooling Layer

The purpose of the Pooling Layer is to reduce dimensionality of the inputs produced at the prior layer. The reason for this is that collecting these inputs into lower dimensional ones, leads to less computational power requirements, making the aforementioned computation proceed faster and with less resource consumption as well.Certain features are also invariant (in regards to rotation or even positioning) and thus, will prove to be retained even following the pooling process. Pooling will thus reduce the number of neurons, from the n total of a cluster, to one per cluster following the process. To this end, there are two main modes through which the Pooling Layer functions: maximum pooling and average pooling.

Maximum pooling will look over a m x m region of the n x n array of inputs and select the maximum input out of the region. Average pooling will instead, select the average of the values in the region. Max pooling also has the benefit of acting as a mechanism for noise suppression. While Average Pooling can perform the same function through dimensionality reduction, it does not perform as well since the ultimate result is likely to be affected by noise (through the process of deriving the average value).Thus, it is preferable to use maximum pooling over average pooling for most intents and purposes.

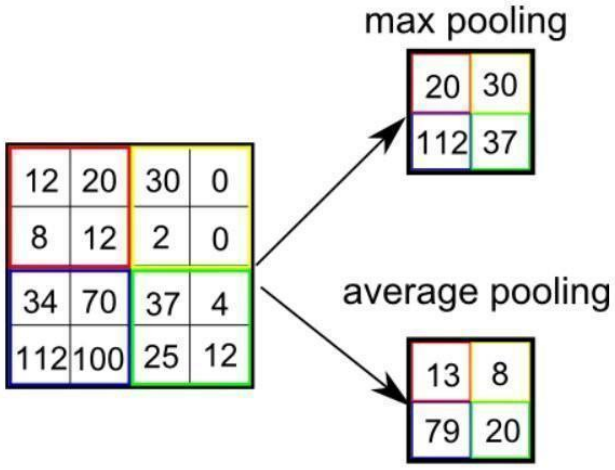

Figure 16. Max and average pooling

2.4.4 Fully Connected Layer

The presence of a FC Layer adds certain benefits. One of them is the ability to connect any neuron in any one layer to any neuron in any other layer (hence the name). It is at this phase that the above mentioned flattening/linearization can be performed safely, converting the n x n input to a n^2 x 1.In this case what is produced is little but a classical MLP (Multi-Layer Perceptron), an Artiicial Neural Network that follows the above precepts of linearizing input data and producing a classified output. Normally, in broad strokes a MLP contains three layers, an input, output and a hidden layer in between, the number of which is variable.

Following the feed-forward step, we apply another one: backpropagation. Backpropagation as defined above, enables the use of gradient methods for the training of multi-layer neural networks. It does so by deriving (computing) a gradient from the model's loss function, while taking into accounts the respective weights.

This allows for the weights to be continuously recalculated and updated to a more accurate value that will eventually also lead to a higher accuracy.

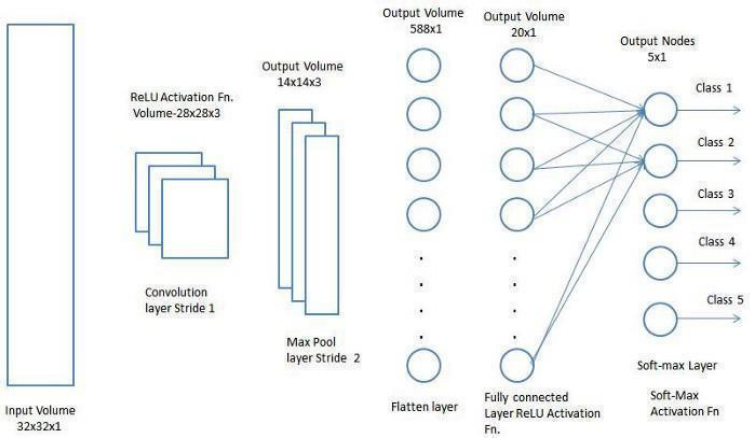

Figure 17. Fully Connected Layer

2.5 FRCNN

FRCNN is said, by its creators (Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun) to depend on region proposal algorithms to hypothesize object locations. [13]

According to their work, the region proposal has been exposed as a bottleneck of research through advancements in various neural network technologies, including the related Fast R-CNN. Faster-RCNN is greatly improved by the introduction of the Region Proposal Network (RPN). It shares the convolutional features (of the full image) with the detection network. The benefit presented is a highly reduced region proposal cost. A Region Proposal Network operates as a proper convolutional neural network, both predicting object bounds and scores at every individual position. [4]

The training process for RPN is end-to-end. Its main function is to generate high-quality RPs, which are then processed by FRCNN for purposes of detection.The above mentioned sharing of features leads to the merging of the Region Proposal Network and Fast R-CNN which is done through the so-called ‘attention’ mechanisms. The RPN component guides the entirety of the unified network in its search. RPN and Fast R-CNN are merged into a single network by sharing their

convolutional features---using the recently popular terminology of neural networks with 'attention' mechanisms, the RPN component leads the unified network where to look. For the very deep VGG-16 model, this detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks. [5]

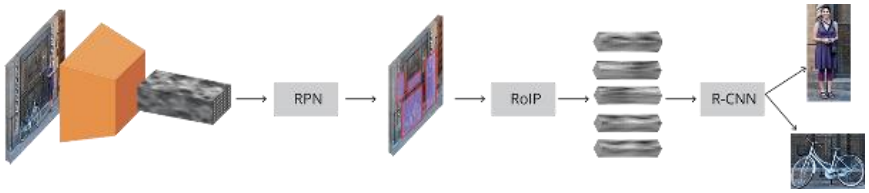

Figure 18. FRCNN schematic

- From the base image, we want to obtain: i) A list of bounding boxes, ii) A label assigned to each bounding box, iii) A probability for each label and bounding box.

The representation of input images is through height x width x depth tensors. These are then passed through a pre-trained CNN (Convolutional Neural Network), into an intermediate layer, and then a convolutional feature map. This is utilized as a feature extractor for the next step.Another part of the above architecture is the one labelled RPN. RPN stands for Region Proposal Network. Its use is in locating a number of predetermined regions which could contain objects. One of the key issues faced with this part, is the creation of variable length lists of bounding boxes, a problem solved by the implementation of anchors.

Anchors are fixed bounding boxes that are placed throughout the image with different sizes and ratios that are going to be used for reference when first predicting object locations. [1][10]

In order to choose the set of anchors we usually define a set of sizes (e.g. 64px, 128px, 256px) and a set of ratios between width and height of boxes (e.g. 0.5, 1, 1.5) and use all the possible combinations of sizes and ratios.

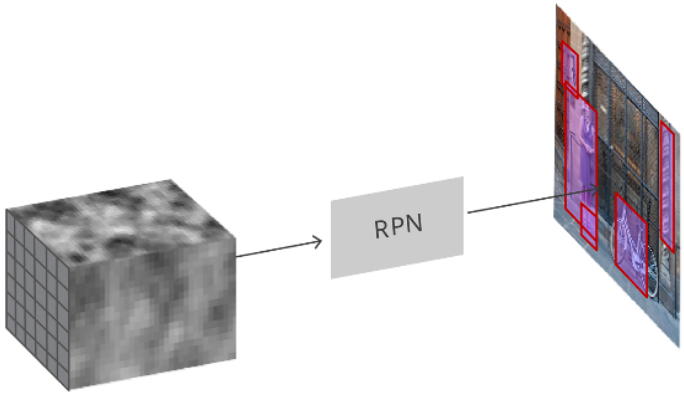

Figure 19. RPN schematic

The RPN retrieves all anchors and proceeds to output a set of proposals (preferably good ones in regards to the objects in question). To this end, each anchor (reference box) has two separate outputs.The first output concerns itself with the probability that the anchor refers to an object. It is simply a score that rates how likely this anchor is to represent an object. The separation here is between objects and background: this first output only concerns itself with the aforementioned separation, rather than classifying objects according to some criterion. This score also serves another role: it can filter bad predictions made on the second phase of the FRCNN training process. The second output involves the bounding box regression. It adjusts the reference boxes to fit the prediction better.

As mentioned before, a convolutional feature map will be returned by the base network, and that helps set up the Region Proposal Network in turn, in a fully convolutional manner. The first step involves a convolutional layer. It has 512 channels and a kernel size of 3x3. Following this, there are two parallel convolution layers with a 1x2 channel, with the channel number varying based on the number of

reference boxes/anchors per point. RPN itself contains two different types of predictions: binary classification and bounding box regression adjustment. Training involves taking all the anchors and separating them into two categories. The overlapping (with a ground-truth object) ones with an Intersection over Union bigger than 0.5 are considered as part of the foreground, while the ones that are not part of this category, or have an IoU of less than 0.1, are classified as background.

Then, we randomly sample those anchors to form a mini batch of size 256 — trying to maintain a balanced ratio between foreground and background anchors.The RPN uses all the anchors selected for the mini batch to calculate the classification loss using **binary cross entropy**. Then, it uses only those minibatch anchors marked as foreground to calculate the regression loss. When calculating the regression targets, we utilize the foreground reference and then the closest ground-truth object to calculate a correct Δ. This Δ is needed to perform a transformation on the anchor, into a full object. Instead of using L1 or L2 loss for calculating regression error, another suggestion is to use smooth L1 loss.

Smooth L1 is a less restrictive L1, which upon finding a small enough L1 error, defined by a set σ, the consideration is 'almost correct' and the diminishing of loss happens at a faster rate. Originally, faster R-CNN was trained with a multi step approach. Parts were trained independently and the trained weights were merged, followed by a full final training. After this, it was found that end-to-end joint training led to superior results. Putting a complete model together provides 4 different losses. Two of them correspond to the Region Proposal Network and two to the R-CNN. The trainable layers exist in both the RPN and R-CNN and there is also the base network which can be fine tuned (trained) or not.

Whether we want to train the base network or not depends on the nature of the objects we want to study and the computing power we have available. If the objects being detected are similar to the contents of the original base network training dataset, therein't really a need to train, other than to improve performance to the finer ends.

However, the base network training can be expensive, both in terms of time elapsed and hardware resources, needing to fit all the complete gradients.

The four different losses come together through a weighted sum. The reason for this is that we may want to give certain weight to, for example, classification losses over regression losses, or R-CNN losses over RPN losses. Apart from these regular losses, there are the regularization losses. They can be defined in both the RPN and R-CNN. L2 regularization is used for some layers and depending on the base network at hand or whether it is trained, there may be regularization.

Training uses Stochastic Gradient Descent (SGD) with momentum. The momentum value is set to 0.9. A Faster R-CNN can be trained with any other optimizer, without running into significant problems. The learning rate starts at 0.001 and then eventually goes down to 0.0001 after 50.000 steps. This hyperparameter is one of the most significant ones. The evaluation done uses mAP (Mean Average Precision) at a specific Intersection-over-Union threshold (for example mAP at 0.5). This metric comes from information retrieval. It is commonly used to calculate error in problems that involve ranking or object detection.

These types of metrics are a more difficult subject to discuss in full, however the conclusion is that mAP invokes a penalty when missing a box that should have been detected, or when detecting something not existing and detecting the same thing repeatedly. [9]

CHAPTER 3

METHODOLOGY

The images on which we use FRCNN have to be processed in various means, through the use of certain techniques. Below, the procedure for labelling the image with the cells contained within is shown.[6][7]

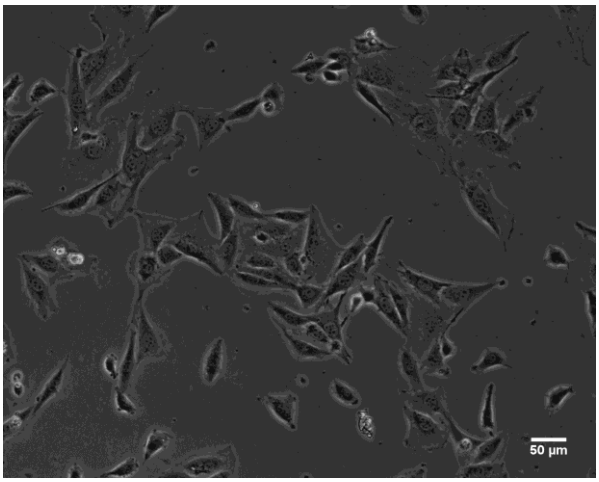

Figure 20. Base Cell Image

This is the original image that we are going to use for purposes of cell detection. In this state, it offers us little in the way of making our work easier. Annotation (labelling) is needed. We can use various services that enable annotation, such as Apeer, as is the case with our work here. [5][7]The aim is to retrieve all of these nuclei and to enable the machine learning model used to be trained to the best accuracy possible. [18]

The result of our annotation is a mask that does not look quite clear or even useful for detection. The varying quality of the labelled nuclei and the uneven distribution of color means that it will not be very effective. A way to deal with this involves the use of image editing software, although programs can also be utilized to retrieve a proper image.

By either selecting the color (if distribution is uniform) and converting it to white (or any other desired color) or alternatively, setting a threshold directly above pure black (000000) we can retrieve a fully black and white mask, as displayed below. [4][18]

The steps to set up this model are simple. Our primary configuration involves importing a number of machine learning libraries (given that the code is fundamentally Python-based), and then configuring the anchor box properties.

We follow this with the annotation parsing (from the related files) and then the definition of the pooling layer. We proceed with the vgg16 model definition, the RPN layer, classifier and then the calculation of the Intersection over Union. The next few steps look over RPN calculation, image augmentation and generating ground truth anchors. Loss function is defined, and then training begins properly. [15]

```
from __future__ import division
from __future__ import print_function
from __future__ import absolute_import
import random
import pprint
import sys
import time
import numpy as np
from optparse import OptionParser
import pickle
import math
import cv2
import copy
from matplotlib import pyplot as plt
import tensorflow as tf
import pandas as pd
import os
import tensorflow.keras as keras

from sklearn.metrics import average_precision_score
```

Figure 21 Library Loading

```python
def nn_base(input_tensor=None, trainable=False):


    input_shape = (None, None, 3)

    if input_tensor is None:
        img_input = Input(shape=input_shape)
    else:
        if not K.is_keras_tensor(input_tensor):
            img_input = Input(tensor=input_tensor, shape=input_shape)
        else:
            img_input = input_tensor

    bn_axis = 3

    # Block 1
    x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv1')(img_input)
    x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv2')(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool')(x)

    # Block 2
    x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv1')(x)
    x = Conv2D(128, (3, 3), activation='relu', padding='same', name='block2_conv2')(x)
    x = MaxPooling2D((2, 2), strides=(2, 2), name='block2_pool')(x)
```

Figure 22. VGG implementation

The above structure defines the neural network base. Note that there are five of the above blocks, and the number of kernels increases by a factor of two, until the fourth block (512), and doesn't change for the fifth. [21]

```python
def rpn_layer(base_layers, num_anchors):
    """Create a rpn layer
        Step1: Pass through the feature map from base layer to a 3x3 512 channels convolutional layer
                Keep the padding 'same' to preserve the feature map's size
        Step2: Pass the step1 to two (1,1) convolutional layer to replace the fully connected layer
                classification layer: num_anchors (9 in here) channels for 0, 1 sigmoid activation output
                regression layer: num_anchors*4 (36 in here) channels for computing the regression of bboxes with linear
    Args:
        base_layers: vgg in here
        num_anchors: 9 in here

    Returns:
        [x_class, x_regr, base_layers]
        x_class: classification for whether it's an object
        x_regr: bboxes regression
        base_layers: vgg in here
    """
    x = Conv2D(512, (3, 3), padding='same', activation='relu', kernel_initializer='normal', name='rpn_conv1')(base_layers

    x_class = Conv2D(num_anchors, (1, 1), activation='sigmoid', kernel_initializer='uniform', name='rpn_out_class')(x)
    x_regr = Conv2D(num_anchors * 4, (1, 1), activation='linear', kernel_initializer='zero', name='rpn_out_regress')(x)

    return [x_class, x_regr, base_layers]
```

Figure 23. RPN implementation

Here, the RPN layer can be observed. As the comments suggest, it begins by going over the feature map of the base layer, to the convolutional layer (3x3) of 512 channels. [15]

Following this step, we have two (1,1) conv. layers that replace the fully connected layer. Beyond the structuring of the RPN layer, there is also the classifier layer, which has the following properties:

```python
def classifier_layer(base_layers, input_rois, num_rois, nb_classes = 4):
    """Create a classifier layer

    Args:
        base_layers: vgg
        input_rois: `(1,num_rois,4)` list of rois, with ordering (x,y,w,h)
        num_rois: number of rois to be processed in one time (4 in here)

    Returns:
        list(out_class, out_regr)
        out_class: classifier layer output
        out_regr: regression layer output
    """

    input_shape = (num_rois,7,7,512)

    pooling_regions = 7

    # out_roi_pool.shape = (1, num_rois, channels, pool_size, pool_size)
    # num_rois (4) 7x7 roi pooling
    out_roi_pool = RoiPoolingConv(pooling_regions, num_rois)([base_layers, input_rois])

    # Flatten the convlutional layer and connected to 2 FC and 2 dropout
    out = TimeDistributed(Flatten(name='flatten'))(out_roi_pool)
    out = TimeDistributed(Dense(4096, activation='relu', name='fc1'))(out)
    out = TimeDistributed(Dropout(0.5))(out)
    out = TimeDistributed(Dense(4096, activation='relu', name='fc2'))(out)
```

Figure 24. Classifier Layer

The base layer here is vgg. It takes as input the ROI list and number, and will return the output of the regression and classifier layers. [15]

```python
def union(au, bu, area_intersection):
  area_a = (au[2] - au[0]) * (au[3] - au[1])
  area_b = (bu[2] - bu[0]) * (bu[3] - bu[1])
  area_union = area_a + area_b - area_intersection
  return area_union


def intersection(ai, bi):
  x = max(ai[0], bi[0])
  y = max(ai[1], bi[1])
  w = min(ai[2], bi[2]) - x
  h = min(ai[3], bi[3]) - y
  if w < 0 or h < 0:
    return 0
  return w*h


def iou(a, b):
  # a and b should be (x1,y1,x2,y2)

  if a[0] >= a[2] or a[1] >= a[3] or b[0] >= b[2] or b[1] >= b[3]:
    return 0.0

  area_i = intersection(a, b)
  area_u = union(a, b, area_i)

  return float(area_i) / float(area_u + 1e-6)
```

Figure 25. IoU Calculation

In this fragment of the model we look at the union and intersection definitions. In the IoU module we see how the intersection is calculated and the output is fed to the union, before performing the necessary division for IoU. [9]

```
downscale = float(C.rpn_stride)
anchor_sizes = C.anchor_box_scales   # 128, 256, 512
anchor_ratios = C.anchor_box_ratios  # 1:1, 1:2*sqrt(2), 2*sqrt(2):1
num_anchors = len(anchor_sizes) * len(anchor_ratios) # 3x3=9

# calculate the output map size based on the network architecture
(output_width, output_height) = img_length_calc_function(resized_width, resized_height)
```

Figure 26.RPN Calculation

The RPN calculation takes in certain arguments: the configuration, augmented image data, height and width of the original image, the same parameters for the resized image as well as the function which calculates the finalized image's feature map. [10]

In turn, the rpn calculation module will return: the rpn class, whether the boxes are valid or not and similarly whether they overlap (as boolean/binary values) as well as the bounding box coordinates for the rpn regression.

The next step before training of the data can begin, is resizing the images to fit with the outcome of the model. Beyond this, the model will proceed with the training phase. [5]

CHAPTER 4

RESULTS AND DISCUSSIONS

At this part, the implementation of a FRCNN model is needed. By taking into account the various layers present within FRCNN, the code will need to reflect these features, in particular, the rpn layer. Following this stage, training proceeds, for 32 epochs, at 80 parts on average (adjustments were made to attempt to get a better result). The final results of the training, are charted in the below graphs

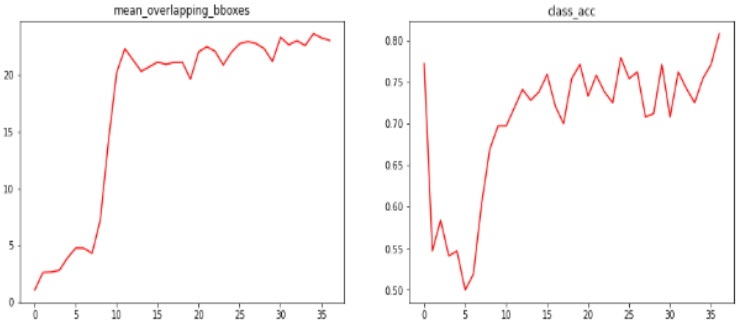

Figure 27. Mean overlapping boundaries and accuracies

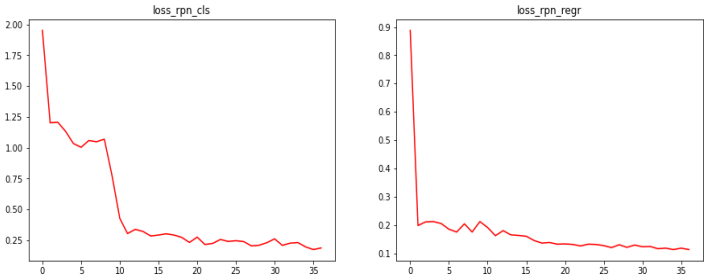

Figure 28. RPN loss

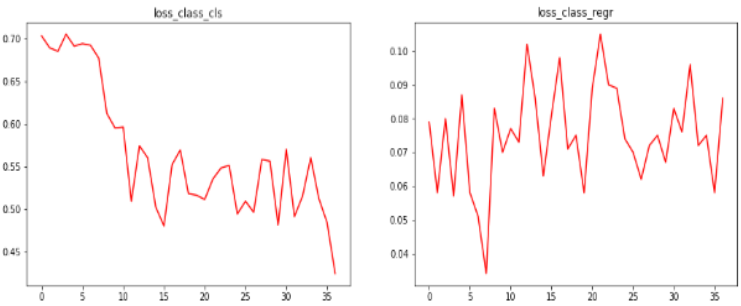

Figure 29. Loss Class

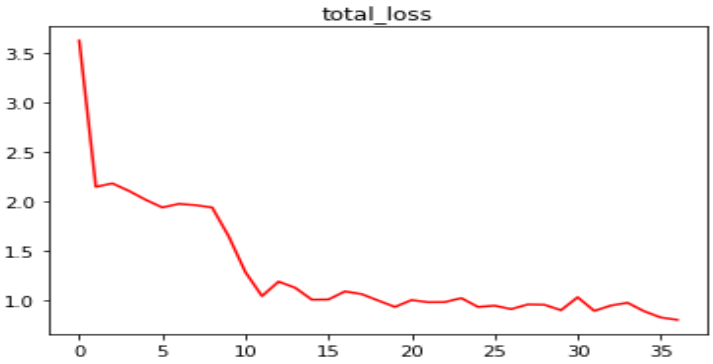

Figure 30. Total loss

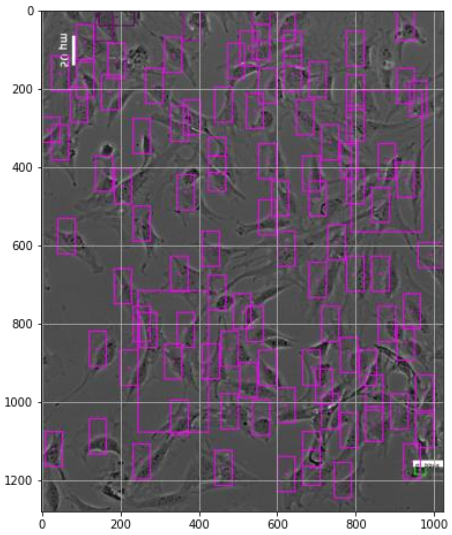

Figure 31. Bounding Boxes with 35 epochs

This does not appear to be a very acceptable (accuracy-wise) outcome. The training proceeds, so that we may obtain a more optimal accuracy read. We proceed to increase the number of epochs by another 70. It must be noted that it is not always training time that influences the final accuracy.

In fact, as has been shown, it is entirely possible that excessive training will lead to suboptimal accuracy. Adjustments will need to be made for a fixed training time/epoch split (which in our case is mostly a matter of multiplication: total time equals the product of epochs multiplied by parts).

We proceed to train it further, to 70 epochs. The outcomes here appear a bit better.

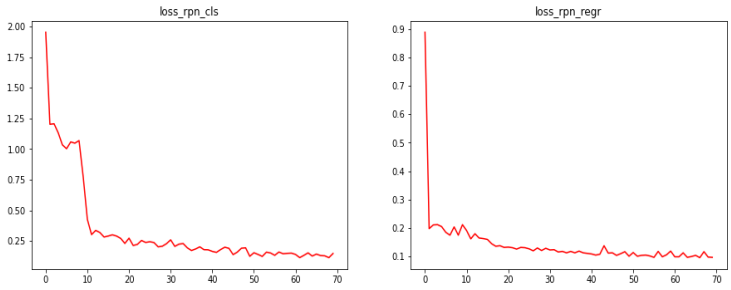

Figure 32. 70 epoch metrics

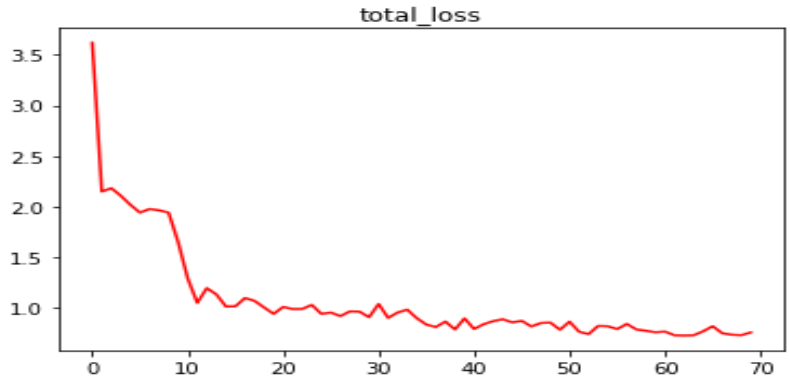

Figure 33. Total Loss at 70 epochs

```
Epoch 70/70
60/60 [==============================] - 1871s 31s/step - rpn_cls: 0.1461 - rpn_regr: 0.0941 - final_cls: 0.3740 - final
Mean number of bounding boxes from RPN overlapping ground truth boxes: 22.033333333333335
Classifier accuracy for bounding boxes from RPN: 0.8041666666666667
Loss RPN classifier: 0.14924117042683066
Loss RPN regression: 0.09701324713726839
Loss Detector classifier: 0.4373878602559368
Loss Detector regression: 0.07166050247809229
Total loss: 0.7553027802981281
Elapsed time: 1871.1091504096985
Training complete, exiting.
```

Figure 34. Final epoch output

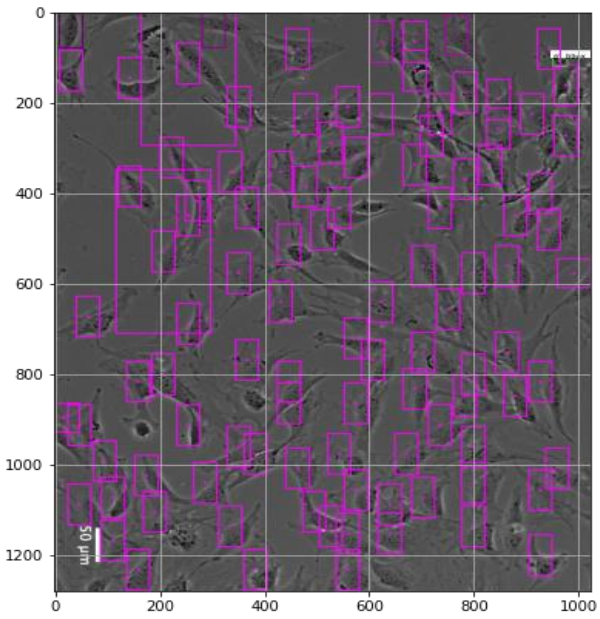

Figure 35. Bounding Boxes

As we can see, the loss is fairly low for each individual classifier. Testing phase commences afterward, taking into account the model definition from the training step (which is loaded into the testing framework) and then we test for individual images. The following losses are retrieved.

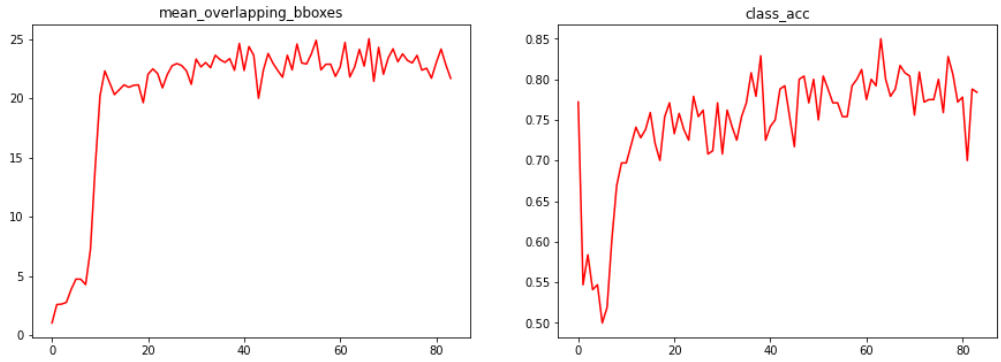

Figure 36.Box overlap and class accuracy

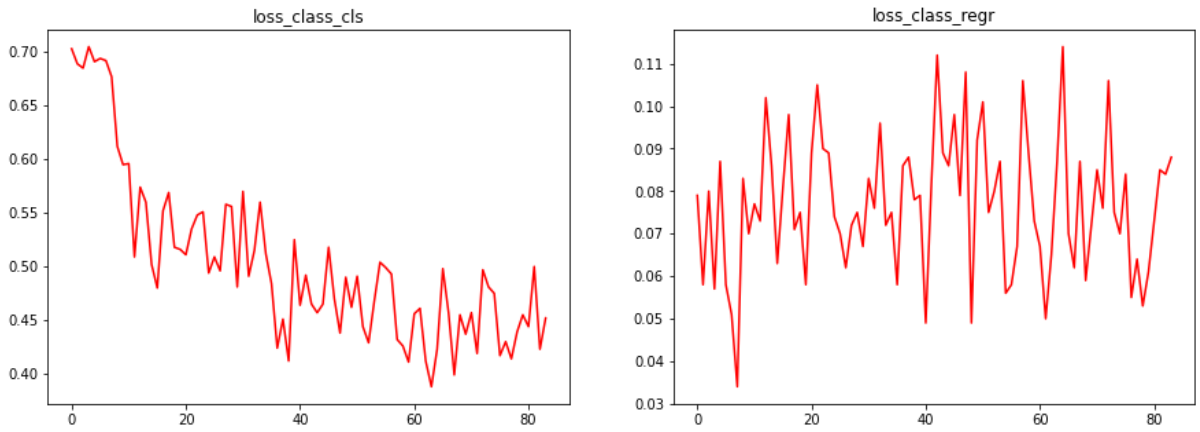

Figure 37. Train Loss Metrics

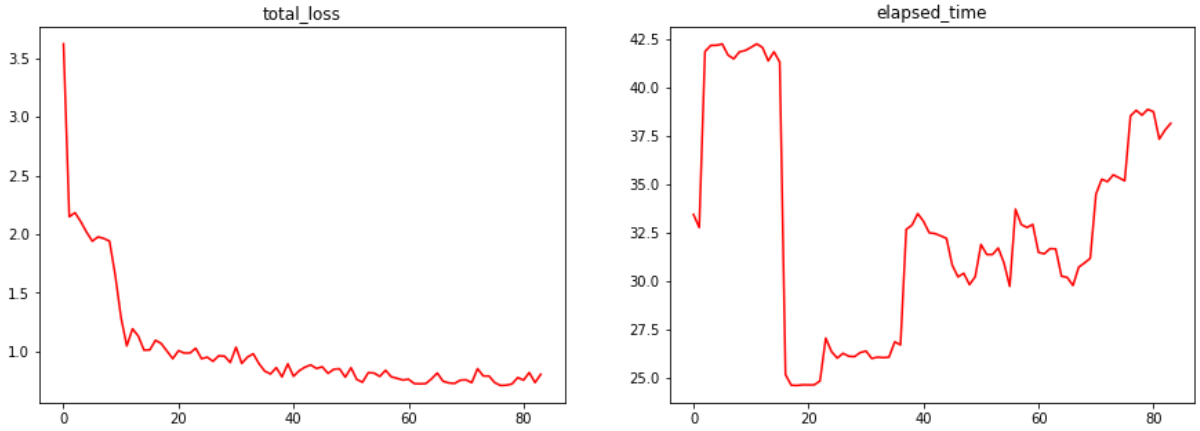

Figure 38. Final train metric

CHAPTER 5

CONCLUSIONS

5.1 Conclusions

As work on this thesis began, multiple approaches were considered. Originally the subject of the detection algorithm were not microscopic images, and originally different considerations were made for using LeNet and the UNet architecture as well, the latter leaving a mark through the presence of annotations that

were ultimately unused. Eventually, Faster-RCNN was chosen as the model for detection.

The total number of images utilized is 84. The number of cells to be counted in each image, is variable, and tends to vary, but averages out in between 100~200.

The final averaged precision with the increased number of epochs hovers around 80%, for the total number of cells found across all images. The improvement is displayed through the bounding boxes: note that they attempt to focus on the nucleus. A key concept to work with is the fact that each cell has a single nucleus and thus, what is actually being counted are cell nuclei: easier to detect due to their circular membrane and their inner, darker coloration (depicted as several smaller black dots).

Among the challenges experienced were the above-mentioned difficulties with the images, where there is some lack of clarity, that can prove difficult even if one is to manually annotate the cells. Overlapping nuclei, poor contrast and even the presence of mitosis (marked by a brightly glowing area), as the cells split in two, can prove to make work difficult, for human operators and obviously enough for the machine as well.

Beyond this, there are the standard challenges of implementing a neural network. Proper training is needed to ensure that the testing phase proceeds smoothly. Furthermore, adjustment of the input dataset to better fit with the present model is also needed, not to mention adjustment of many parameters to ensure optimal training: this is often time consuming and rarely set-in-stone, so it requires some amount of trial and error.

5.2 Recommendations for future research

Future work could be directed at improving the average precision to better values. Alternatively, adjustments in the model could also be introduced, that make it more fit for medical imaging (in fact, this was one of the considerations/upsides UNet had). Another direction to be possibly taken, is that of augmenting the dataset in various ways, either by introducing greater contrast, or using different imaging technology (if/where possible), that makes cell detection easier. Ultimately, the

contribution of this work lies in providing a reasonably efficient and capable tool for detecting large numbers of cells within a set of images: a task that automatization has yet to widely grasp.

REFERENCES

[1] F. Zernike, "Phase contrast, a new method for the microscopic observation of transparent objects part II," *Physica*, pp. 974--986, 1942.

[2] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, 1992.

[3] "tryolabs," 18 01 2018. [Online]. Available: https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/. [Accessed 3 Aug 2021].

[4] C. Alippi, S. Disabato and M. Roveri, "Moving convolutional neural networks to embedded systems: the alexnet and VGG-16 case," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2018.

[5] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, D. Nardi and others, The description logic handbook: Theory, implementation and applications, Cambridge university press, 2003.

[6] R. D. Barnes and J. Holliday, "The morphological identity of maternal cells in newborn mice," *Blood*, p. 1970, 480--490.

[7] P. K. Jain, S. Gupta, A. Bhavsar, A. Nigam and N. Sharma, "Localization of common carotid artery transverse section in B-mode ultrasound images using faster RCNN: a deep learning approac," *Medical \& biological engineering \& computing*, pp. 471--482, 2020.

[8] S. Jinnai, N. Yamazaki, Y. Hirano, Y. Sugawara, Y. Ohe and R. Hamamoto, "The development of a skin cancer classification system for pigmented skin lesions using deep learning," *Biomolecules*, p. 1123, 2020.

[9] T. Mahmood, M. Arsalan, M. Owais, M. B. Lee and K. R. Park, "Artificial

intelligence-based mitosis detection in breast cancer histopathology images using faster R-CNN and deep CNNs," *Journal of clinical medicine,* p. 749, 2020.

[10] A. E. Maxwell, T. A. Warner and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *International Journal of Remote Sensing,* pp. 2784--2817, 2018.

[11] T. T. D. Nguyen, B.-N. Vo, B.-T. Vo, D. Y. Kim and Y. S. Choi, "Tracking Cells and their Lineages via Labeled Random Finite Sets," *arXiv preprint arXiv:2104.10964,* 2021.

[12] R. E. Putra, H. Tjandrasa, N. Suciati and A. Y. Wicaksono, "on-Proliferative Diabetic Retinopathy Classification Based on Hard Exudates Using Combination of FRCNN, Morphology, and ANFIS," in *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 2020.

[13] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* pp. 91--99, 2015.

[14] S. Tu, J. Pang, H. Liu, N. Zhuang, Y. Chen, C. Zheng, H. Wan and Y. Xue, "Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images," *Precision Agriculture,* pp. 1072--1091, 2020.

[15] A. Uka, A. N. Halili, X. Polisi, A. O. Topal, G. Imeraj and N. E. Vrana, "Basis of image analysis for evaluating cell biomaterial interaction using brightfield microscopy," *Cells Tissues Organs,* pp. 77--104, 2021.

[16] A. Uka, A. Tare, X. Polisi and I. Panci, "FASTER R-CNN for cell counting in low contrast microscopic images," in *2020 International Conference on Computing, Networking, Telecommunications \& Engineering Sciences Applications (CoNTESA)*, 2020.

[17] A. Villringer, J. Planck, C. Hock, L. Schleinkofer and U. Dirnagl, "Near infrared spectroscopy (NIRS): a new tool to study hemodynamic changes during activation of brain function in human adults," *Neuroscience letters,* p. 1993, Neuroscience letters.

[18] J. Wang, K. Chen, S. Yang, C. C. Loy and D. Lin, "Region proposal by guided

anchoring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[19] A. Zaritsky, S. Natan, J. Horev, I. Hecht, L. Wolf, E. Ben-Jacob and I. Tsarfaty, "Cell motility dynamics: a novel segmentation algorithm to quantify multi-cellular bright field microscopy images," *PloS one*, p. e27593, 2011.

[20] M. K. Zeybel and Y. S. Akgul, "Identification and Localization of Lumbar Intervertebral Discs on MRI Images with Faster RCNN," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019.

[21] L. Zhang and J. a. Z. B. Shen, "A research on an improved Unet-based concrete crack detection algorithm," *Structural Health Monitoring*, pp. 1864--1879, 2021.

[22] Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, pp. 3212--3232, 2019.

APPENDIX

https://colab.research.google.com/drive/1IoPy8R_C9vcRsOc7t4JXajodrWGFZy_r
https://colab.research.google.com/drive/1l2nP709X4PkkaAqwSEleCwHNQVTSVljW