UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Marten Türk

Improving Semantic Segmentation of Microscopy Images Using Rotation Equivariant Convolutional Networks

Bachelor's Thesis (9 EAP)

Supervisor: Dmytro Fishman, PhD

Tartu 2022

Improving Semantic Segmentation of Microscopy Images Using Rotation Equivariant Convolutional Networks

Abstract:

The segmentation of the cell nuclei is one of the first steps in medical image analysis workflow. Organisations conducting experiments with image analysis are mainly pharmaceutical companies and biomedicine laboratories, which need to process vast amounts of data and quantify it. The goal of these experiments could be to produce new drugs or diagnose diseases. Due to advancements in deep learning, these processes of nuclei segmentation have been automated, and the level of accuracy is relatively high. However, new methods for improving the accuracy of the models are constantly proposed. One of these proposals uses rotation equivariant convolutional neural networks based on group theory. These networks can produce invariant predictions regardless of the rotation of the input object.

This bachelor's thesis shows that rotation equivariant convolutional neural networks improve the semantic segmentation of nuclei and increase the generalisation capabilities of a model trained on fluorescent images. Additionally, the work gives an overview of failed attempts with brightfield images, outlines the already existing rotation equivariant models on the internet and describes their implementation complexity.

Keywords:
Computer science, biomedicine, machine learning, convolutional neural networks, segmentation, image analysis.

CERCS: P170

Mikroskoobipiltide semantilise segmentatsiooni parendamine pöörde-ekvivariantsete konvolutsionaalsete närvivõrkude abil

Lühikokkuvõte:

Rakutuuma segmenteerimine on üks esimesi samme mikroskoobi piltide uurimise tööprotsessis. Peamisteks uuringute läbiviijateks on eelkõige ravimifirmad ning meditsiinilaborid, mille üheks ülesandeks on analüüsida suurtes kogustes rakupilte ning neid kvantifitseerida. Uuringute eesmärkideks võib olla uute ravimite välja töötamine või haiguste diagnoosimine. Tänu süvaõppele on suudetud antud protsessid automatiseerida ja nende tulemuste täpsus on viidud võrdlemisi kõrgele tasemele. Vaatamata headele tulemustele pakutakse jätkuvalt välja meetodeid, kuidas masinõppe algoritme veel täpsemaks muuta. Üheks selliseks meetodiks on rühma teoorial põhinevad pöörde-ekvivariantsed (ingl. *rotation equivariant*) konvolutsionaalsed närvivõrgud, mis suudavad objekti pöördenurgast olenemata väljastada võrdväärseid ennustusi.

Bakalaureusetöö eesmärgiks on näidata, et pöörde-ekvivariantsed konvolutsionaalsed närvivõrgud parandavad semantilise segmenteerimise tulemusi ning suurendavad masinõppe algoritmi üldistusvõimet fluorestsentspiltidel. Samuti annab töö ülevaate ebaõnnestunud eksperimentidest ereväljapiltidel ja hetkel leiduvatest pöörde-ekvivariantsetest mudelitest ning nende implementeerimise keerukusest.

Võtmesõnad:
Arvutiteadus, biomeditsiin, masinõpe, konvolutsionaalsed närvivõrgud, segmenteerimine, pildianalüüs.

CERCS: P170

Table of contents

1. Terms and notations

Convolutional neural network (CNN) is a subclass of artificial neural networks that utilises convolutional layers. CNNs are most commonly used for analysing visual data.

Group theory is a branch of mathematics and one of the critical components in mathematical algebra that deals with the study of symmetries.

Group equivariance relies on symmetries of a single group, where each group action leaves the object unchanged.

Group equivariant network is a network that utilises group equivariance to be completely invariant when group actions are performed on the data.

Cell nuclei are the core of a cell containing DNA and managing the cell's metabolism, growth and overall functioning.

Brightfield images are one of the most general microscopy images, where light is shone through the sample, and the picture is taken with a brightfield microscope (Zhang et al., 2020).

Fluorescent images are microscopy images where objects of interest start to illuminate under ultraviolet light. This happens due to fluorescent molecules, which have been attached to the DNA of the cells (Zhang et al., 2020).

Semantic segmentation is a segmentation type, where all pixels are classified as different object classes on an image. All instances of the same category are assigned the same labels.

2. Introduction

Cells are the building blocks of all living organisms. The nucleus is at the heart of cells in eukaryotes. The cell nucleus is a round organelle that is an object of interest for researchers in pharmaceutical companies and biomedical labs. Identifying the cell nuclei is one of the standard first steps in the majority of medical image analysis pipelines (Juan C. Caicedo, 2019). Machine learning algorithms mainly do the discovery of nuclei on microscopy images through semantic segmentation. Nuclei segmentation allows to count the number of nuclei before and after a drug has been applied to the sample (Lu et al., 2021). The quantification then helps to assess whether or not the drug has had an effect. Segmentation has other benefits, such as locating an object of interest for further research or performing morphological analysis on the nuclei, which are helpful in disease diagnosis or when monitoring cell development (Dinesh D. Pati & Ms. Sonal G. Deore, 2013).

Although a handful of approaches for segmenting cells have been developed by many in the past (Ali et al., 2021; Fishman et al., 2021; Olaf Ronneberger et al., 2015), there is still room for improvement. The main area for refinement lies in improving the f1-scores on brightfield images, which currently stand between 0.76-0.86, depending on the cell type (Fishman et al., 2021).

One of the ways to improve the current state of the art models is by focusing on the fact that cells do not possess any orientation in which they can be called upright. Unlike a dog, which is assumed to stand on its legs, not its tail, a cell remains equivariant to transformations such as rotation. These properties can be utilised to improve the accuracy of nuclei segmentation. One of the ways to make use of these properties is through rotation equivariant networks.

Rotation equivariant networks have been shown to outperform conventional methods generally used in the segmentation of histopathology, synthetic biomarker and subcellular protein localisation images (B. Chidester et al., 2019; Bastiaan S. Veeling et al., 2018; Chidester et al., 2019). Hence, we reasoned that rotation equivariant convolutional neural networks (Taco S. Cohen & Max Welling, 2016) can be used to leverage the symmetry of nuclei.

To summarise, this thesis mainly focuses on exploring the possibilities of improving semantic segmentation of cell nuclei from fluorescent and brightfield microscopy images through the use of rotation equivariant neural networks.

2.1 Research questions

- Explore the value of equivariant models, namely convolutional neural networks for semantic segmentation of cell nuclei from microscopy images.
- Compare the results of the equivariant networks to existing relevant alternatives.
- Make the results of the analysis available to a community.

3. Background information

This chapter introduces the technologies and their working cycles used in the work. The main topics that will be covered are convolutional neural networks, group equivariance and semantic segmentation. We retain a relatively shallow depth of the discussion presented in this chapter to on one hand support readers in further chapters and on the other hand remain inside the scope of the BSc thesis.

3.1 Convolutional Neural Networks

A convolutional neural network (CNN) is a subclass of artificial neural networks mainly used for image analysis. As the name suggests, CNN's are built using convolutional layers and for every convolutional layer, a matrix with a size predefined by the architecture is used to scan the image. Such matrices are called convolutional filters.

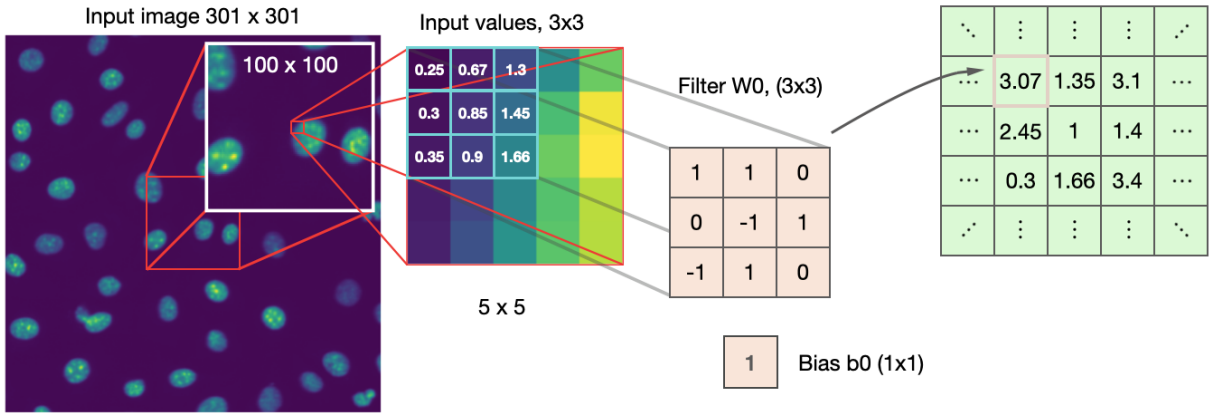

Figure 1. Example of a convolutional operation. The figure is meant to be read from left to right, starting with the 301px x 301px input image. To the right of the input image, there is a 5 x 5 magnification of the input, which contains a 3 x 3 blue matrix with numbers that represent pixel intensities. The figure also displays a convolutional filter W0 with corresponding weights and a bias term b0. The green matrix on the right is the output of the convolutional operation, often referred to as a feature map. The elements in the feature map are calculated by summing up the bias term and the results of element-wise multiplication between the filter and the corresponding place in the original image.

The convolutional filters convolve over the image, meaning they calculate a value for an output matrix by multiplying element-wise values with the input and adding the bias. The resulting matrix is called a feature map, which will then be passed on to the following convolutional filters. After a single value has been calculated for the output matrix, the convolutional filter moves forward in the input row by X number of steps, which the model architecture predefines and calculates the following value for the feature map. If the filter reaches the end of a row, it will go back to the start of the row, but this time starts in the column, which is X steps down. This process is repeated until the entire input matrix has been covered (LeCun et al., 1999).

The models often incorporate other layers that alter the feature map, such as max-pooling or upsampling. A max-pooling layer undergoes an operation by choosing the maximum value inside the feature map on a limited patch. The outcome is a downsampled feature map highlighting the most informative feature in that specific patch. On the other hand, an upsampling layer doubles the dimensions of its input. The layer works by duplicating the rows and columns of its input. The goal of the upsampling layer is to increase resolution and reduce noise in the image.

In addition to changing the inputs to layers, these input matrices also can be standardised to make the model training faster and more stable. This is done by a layer called batch normalisation, which normalises inputs to layers after each mini-batch (Sergey Ioffe & Christian Szegedy, n.d.).

During training, the network uses functions to update the bias and the values inside the filters, called weights, which at first are generated randomly. The purpose of updating the weights is to optimise a training objective, often set by a quantity called loss function. The smaller the value of the loss function the more successful the network is in solving a particular problem, in our case - segmenting nuclei from microscopy images. Adjusting weights is called backpropagation (Rumelhart et al., 1986), and the updating is done based on the loss function value obtained during the previous epoch. In order for the adjustments of weights to match the expected outcome as accurately as possible, the models use an optimizer. The optimizer is the connector

between the loss function and the updating of weights, it is built to update the weights as effectively as possible through trial and error with the information from the loss function. This work relies on using the Adam optimizer (Diederik P. Kingma & Jimmy Ba, 2014).

Convolutional neural networks and other types of deep learning approaches rely heavily on the use of one or more types of functions called activation functions. In CNN's, these functions are mainly used between two convolutional layers to normalise the output of one layer before sending the information to the next. This has been shown to improve the speed of convergence for models while training (Wang et al., 2020).

This work mostly uses the rectified linear unit (ReLU) (Figure 2, left) and the sigmoid (Figure 2, right) activation functions.

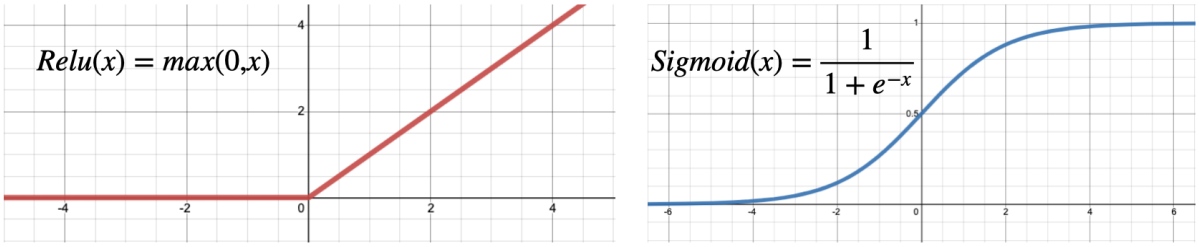

Figure 2. Rectified linear unit (ReLU) and Sigmoid activation functions. The left graph shows the ReLU activation function, transforming negative values (z) into zeroes. The graph on the right shows the Sigmoid activation function - no matter how big or small the input value is, the output value always remains between zero and one.

The sigmoid function transforms all input values to the range between zero and one. The lower the input value, the closer the output is to zero, and the bigger the input value, the closer the output is to one. The ReLU activation function transforms all the inputs to be the maximum between zero and the input value, effectively nullifying all negative input values.

The activation functions are one of the few things that do not need to be altered when converting a regular model into a group equivariant model.

3.2 Group Equivariant neural networks

Group equivariance is generally referred to when talking about models, which predict the same output regardless of the group transformation the input has undergone. One of the groups that a model can be equivariant to, is all 90-degree rotations. Being equivariant to rotations means that if an input image is rotated with multiples of 90-degrees, the output will always be the same. This is achieved through the filters learning all rotations of training images during training. This is not the expected behaviour for regular CNNs because even though multiple feature maps together might produce an equivariant result, a single feature map will not (Taco S. Cohen & Max Welling, 2016). Figure 3 shows the results produced by a non-equivariant and an equivariant model.

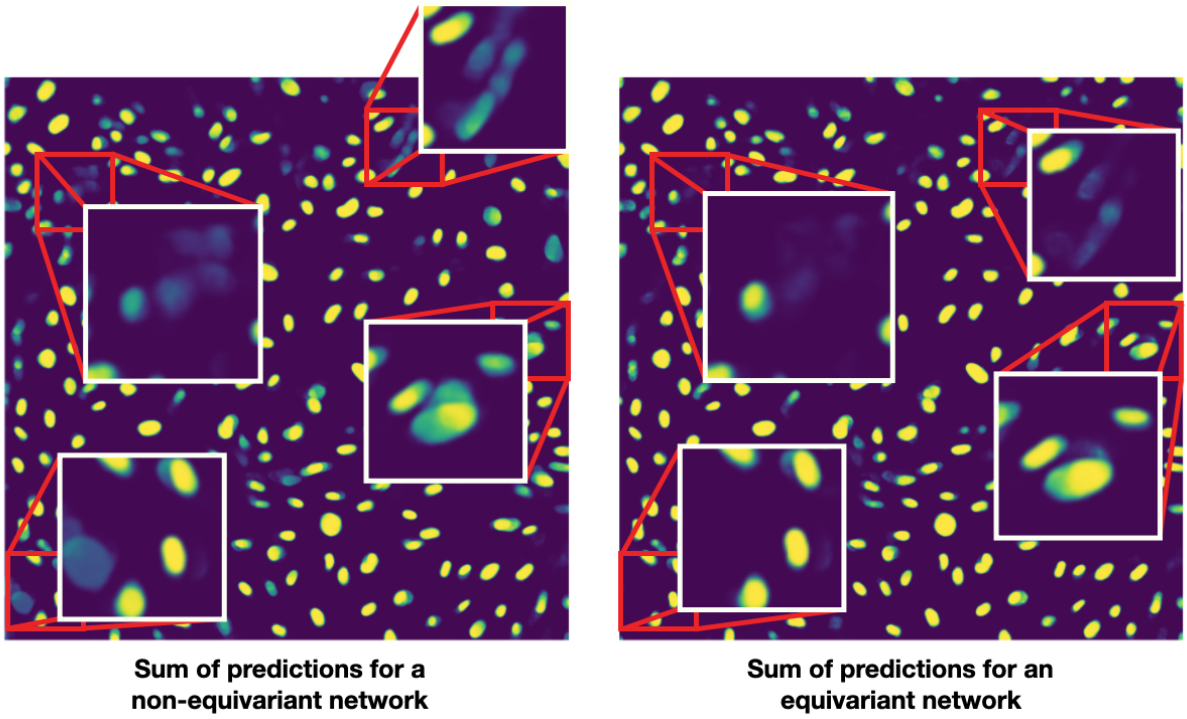

Figure 3. Probability maps of brightfield images. The goal of the figure is to show what a rotation equivariant model and a rotation non-equivariant model would produce if given four images; each one rotated a different amount. The image on the left comprises the sum of predictions for four images, which have been turned respectively by 0-, 90-, 180- and 270-degrees. The photo on the right predicts a single image that has not been rotated. Red squares have been used to point out differences. When looking at the corresponding highlighted parts of both images, it can be seen that on the left image, some rotations have introduced parts of nuclei that the representation of a rotation equivariant network has not seen. This is an expected result from a network that is not equivariant to rotations. However, a rotation equivariant model would produce an effect similar to the one on the right. It does not create duplicates and does not find cells on different rotations because the image rotation does not improve or worsen the output.

In addition to Figure 3, an example to understand equivariance would be that if a rotation non-equivariant model were given an image of an upright car, it would predict it as a car. Now, if the image is rotated 90-degrees, it might think it is an image of a tree. A rotation equivariant model would predict the image as a car regardless of the 90-degree rotational transformation.

This concept of models predicting the same output regardless of the rotation of the input is inspired by group theory. Group theory is a branch of mathematics and one of the critical components in mathematical algebra that deals with the study of symmetries. “When an object appears symmetric, group theory can help us study it.” - (Keith Conrad, 2014). In the current thesis, we focus on rotation equivariant networks which are a subclass of a bigger entity called group convolutional neural networks (G-CNNs). A group convolutional network can be equivariant to a number of different groups of symmetry such as translations (or shifts), 90-degree rotations and reflections. With these actions or transformations, specific functions can be defined that would define a group convolution, which is the central part of a G-CNN. To be more precise, a group convolution or a G-convolution is a function that defines a group of interest which can then be used to utilize the fact that cells remain equivariant under rotation.

For easier understanding, two functions *m*, which mirrors an image and *r*, which rotates an image 90-degrees, can be defined. Figure 4 illustrates the two action functions *m* and *r*. As a side note, all the following visualisations in the current chapter are inspired by the blog post Geometric Deep Learning (Casper van Engelenburg, 2020).

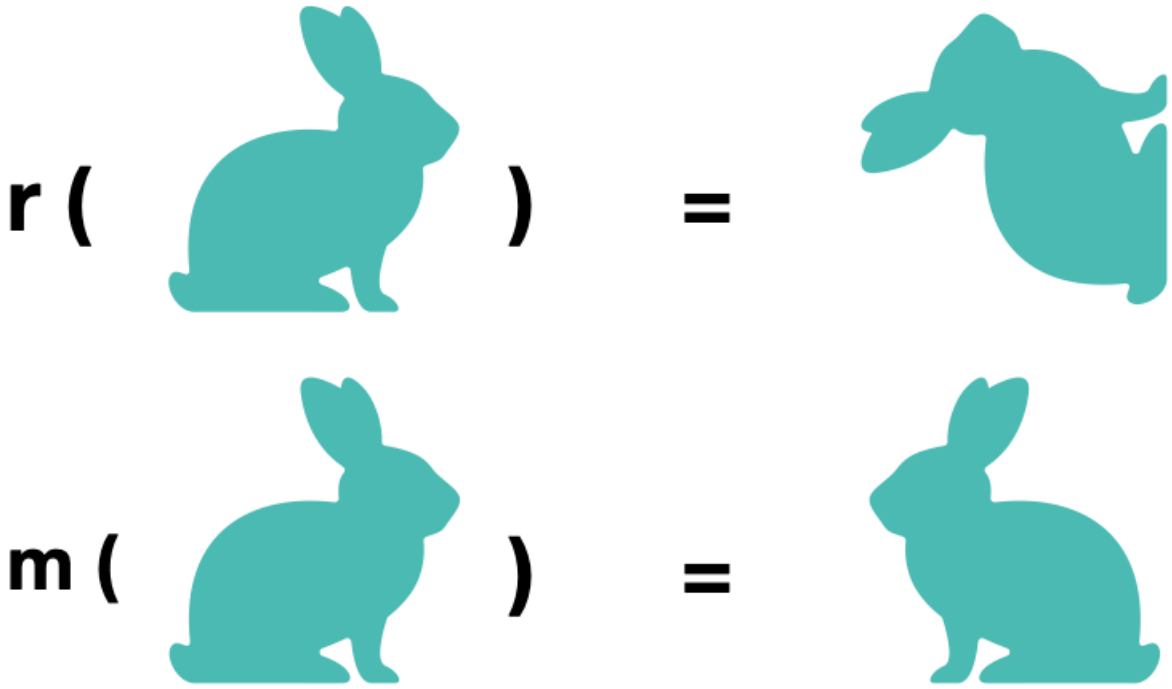

Figure 4. Examples of a rotation and a reflection being applied to an animal's mask. The top-most function *r* displays image rotations, and the bottom function *m* shows a mirroring of the image when the function is applied.

It is clear that some of the combinations of the two actions provide an equivalent outcome (e.g. rotating an object four times or flipping it twice leaves an object unchanged). These combinations help define groups of symmetry and this is important because then the steps needed to take to achieve equivariance are clear. This logic will be used later in this chapter to explain the working cycle of a group convolutional network. When defining equivariant groups, the idea is to have graphs where the nodes denote possible rotated orientations of the object, and edges symbolise transformations. The groups investigated in the thesis are p4, and p4m, where p4 is a group of all possible 90-degree rotations and p4m represents all rotations and possible mirrorings of the input. See Figure 5 for a visual representation of the two groups.

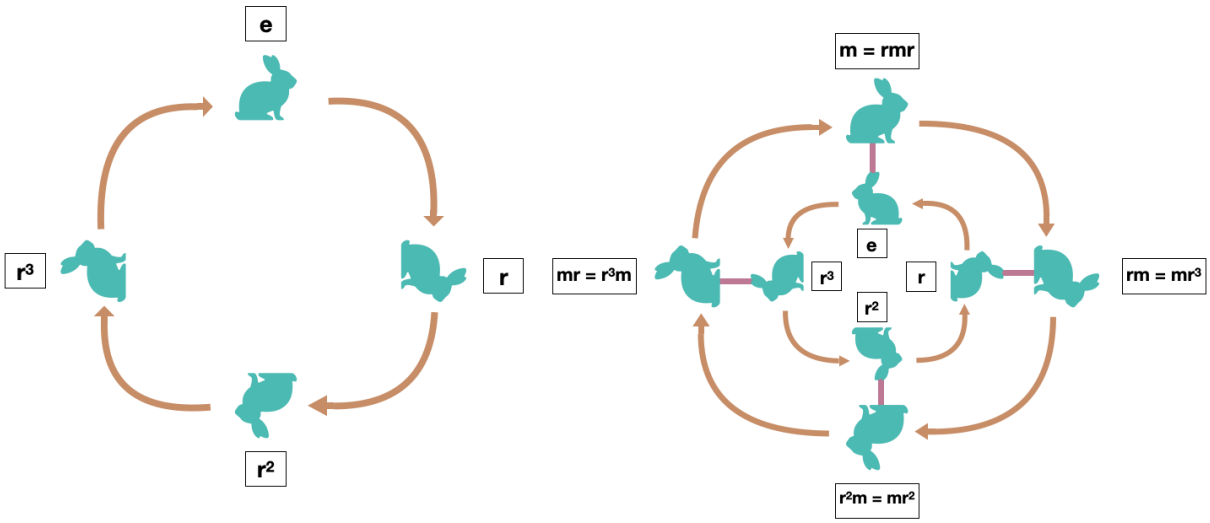

Figure 5. Graphical representations for the group p4 (left) and the group p4m (right). The brown arrows on the figures represent transformations such as mirrorings or rotations and the images display the state of the image after the modifications have been applied. The letter *r* represents a rotation function and the number in superscript next to it indicates how many times this action has been performed to reach the current state. The same goes for *m*, which indicates the mirroring of an image. The group p4 consists of all possible 90-degree rotations. The group p4m has mirroring at every rotation in addition to the rotations.

Since regular CNNs are invariant to shifts but not equivariant to other transformations such as rotations, changes must be implemented to the architecture of the layers (Taco S. Cohen & Max Welling, 2016). Group p4 can be considered to describe the differences between a regular and a group p4 equivariant network. To create a network equivariant to the group p4, the following must happen: when an image is fed into the network, then the filters in each layer must all be transformed according to every possible pose of the group, and then regular convolution is performed to create four feature maps, as there are four different rotations that can be performed. Figure 6 shows a visual representation of a filter being transformed according to the group p4.

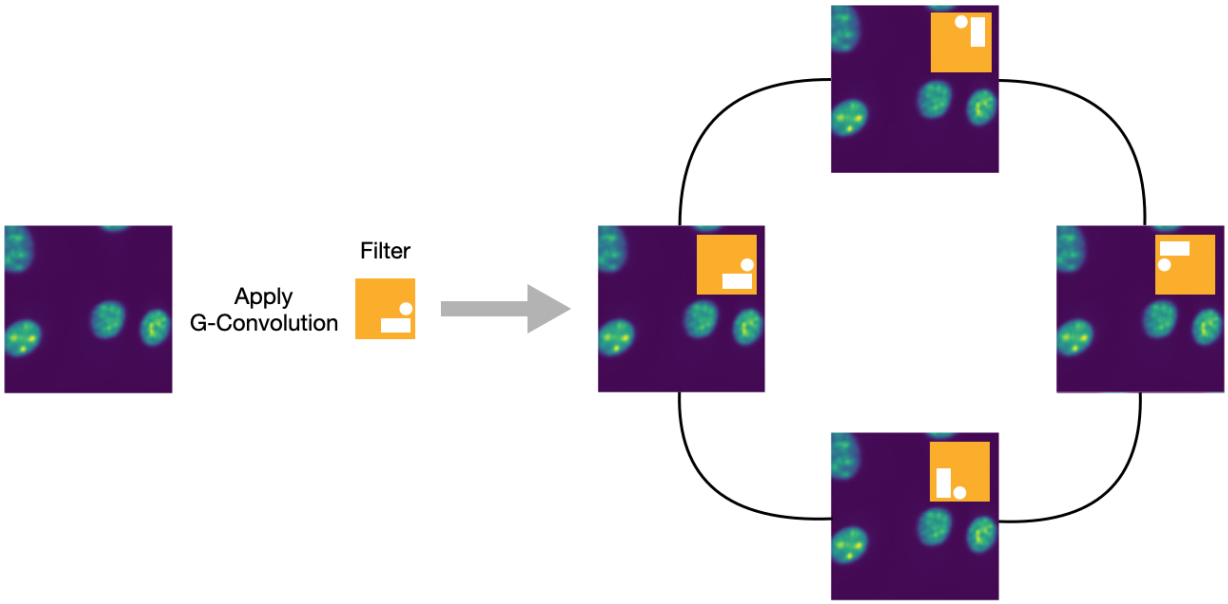

Figure 6. A visual representation of a filter being allocated to every possible pose according to group P4 on an image of cell nuclei.

In case a structured feature map has already been created, then another homogeneous structured feature map is generated, and the existing one is passed onto it (Taco S. Cohen & Max Welling, 2016). There will be two actions that need to be done to achieve equivariance.
The first action is transformation. This step implies two actions: the individual rotation of filters and allocation to the next node.

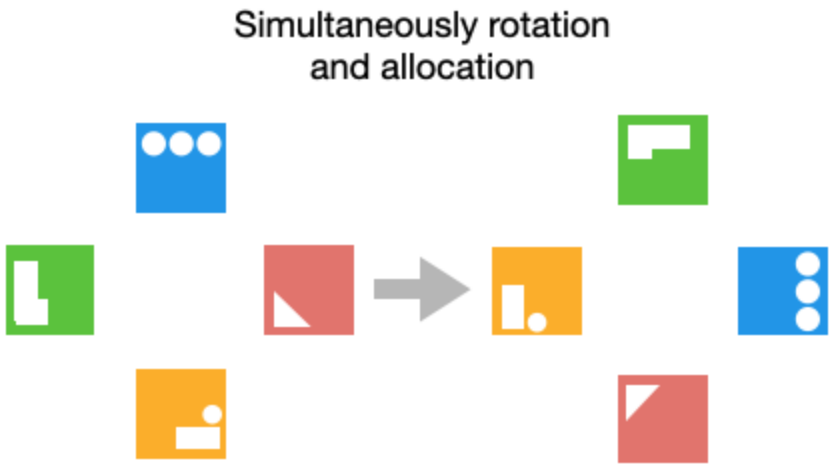

Figure 7. Simultaneous allocation and rotation of filters. Each filter is rotated individually 90 degrees, and all four filters are allocated together.

The second step is taking the dot-product of two structured objects by taking the sum of pointwise values. See Figure 8 for a visual representation.

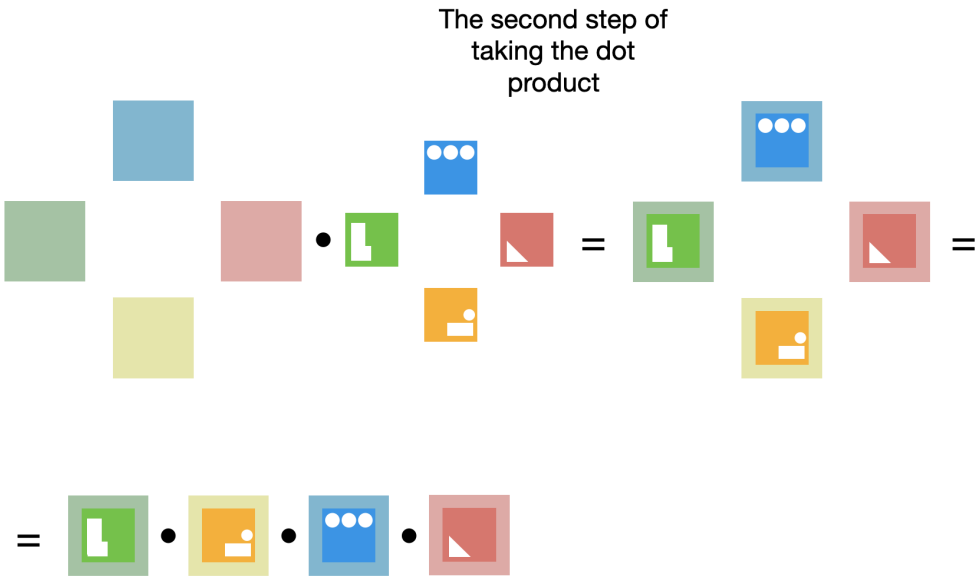

Figure 8. The dot product of two structured objects. The pictures of cells have been replaced with coloured boxes because, at this point, the feature maps do not show cells but something more abstract.

Now, the entire cycle of a G-Convolutional network can be defined. The first step is creating filters for all possible transformations (in the current example, there are four possibilities because you can rotate an object 90-degrees four times). The current step will act as a bridge from group Z2, which is equivariant to shifts and is the most common group that convolutional neural networks operate on, to a rotation and shift equivariant group P4. The four generated feature maps are then passed on to the next layer, where there are now four feature maps waiting for input. The process continues as it previously has; each filter is simultaneously rotated and allocated to all feature maps, and the dot product is taken at the end.

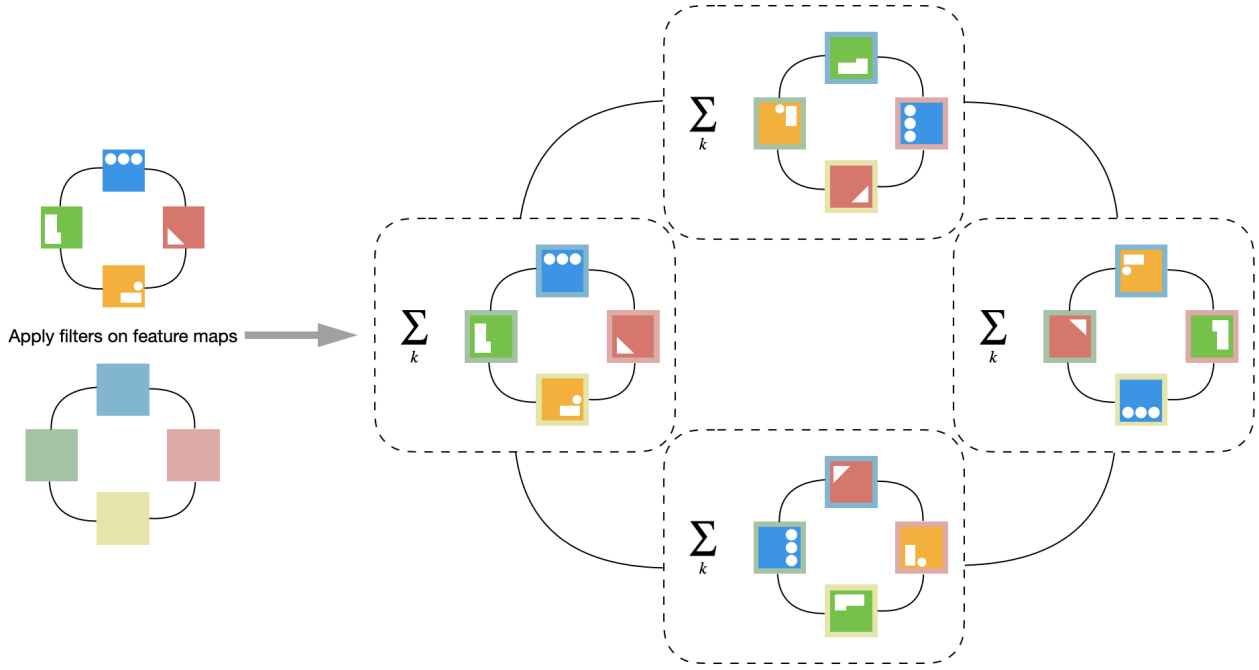

Figure 9. A convolutional neural network at work with the group P4. The k is the number of filters and for every box surrounded by dashed lines, the dot-product is taken. The lines between boxes indicate the process of rotation and reallocation onto the following feature map.

It is crucial to notice that each feature map will give partial information to the rotating filters. However, all the four feature maps together form a complete permutation of all possibilities on how to learn the rotations.

When the model reaches the end of its prediction cycle, the P4 feature maps need to be converted back to a single Z2 feature map to have a single feature map as output, not four. In order to convert the P4 group back into Z2, group pooling can be used, which in essence stacks all the structured feature maps together to have a prediction in the same plane as the input. In this work, we attempted to use P4 rotation equivariant convolutional networks to segment nuclei from microscopy images.

3.3 Semantic segmentation

Image segmentation is one of the most popular topics in image analysis and computer vision (Shervin Minaee et al., 2020). Semantic segmentation aims to define an object of interest on an image with a mask. See Figure 10 for an example.

The task of semantic segmentation can also be rephrased as a classification problem. A segmentation model tries to classify each pixel as part of the object of interest (for example, a cell nucleus) or as the background.

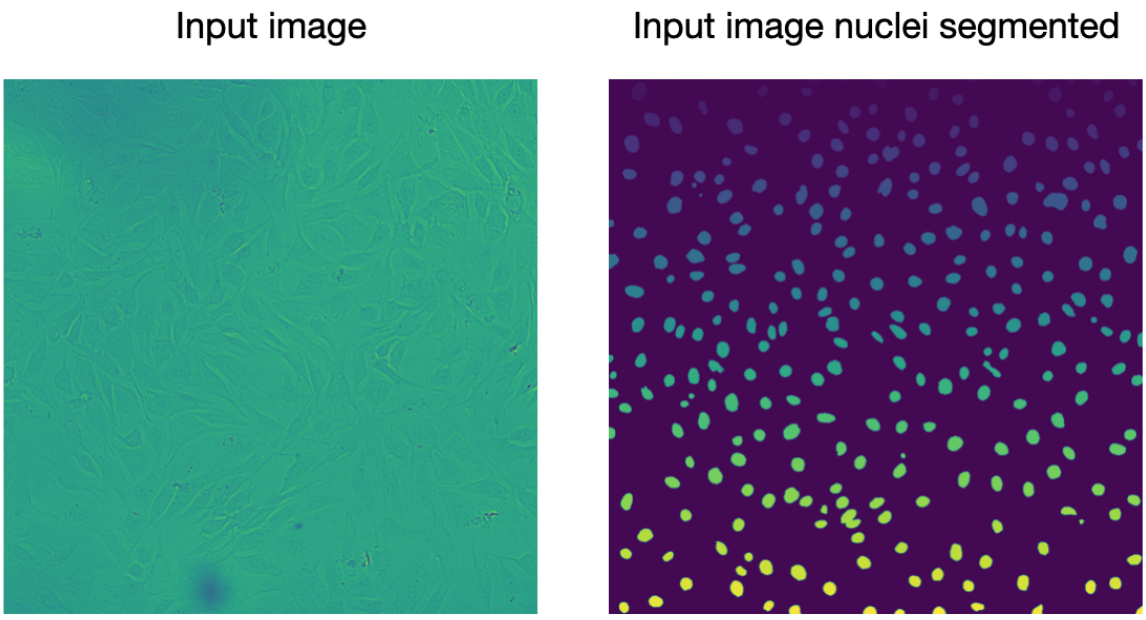

Figure 10. The results of semantic segmentation on a brightfield image. The image on the right depicts the predictions a model has made about the locations of cell nuclei by semantically segmenting them.

When an image has been segmented and pixels have been classified, the evaluation of results must be done to determine the quality of the segmentation. The most common metrics for evaluating segmentation quality are intersection over union (IoU), where the predicted mask and the ground truth are compared to each other; mean pixel accuracy (MPA), where the amount of correctly classified pixels is computed and averaged over all classes (Shervin Minaee et al., 2020).

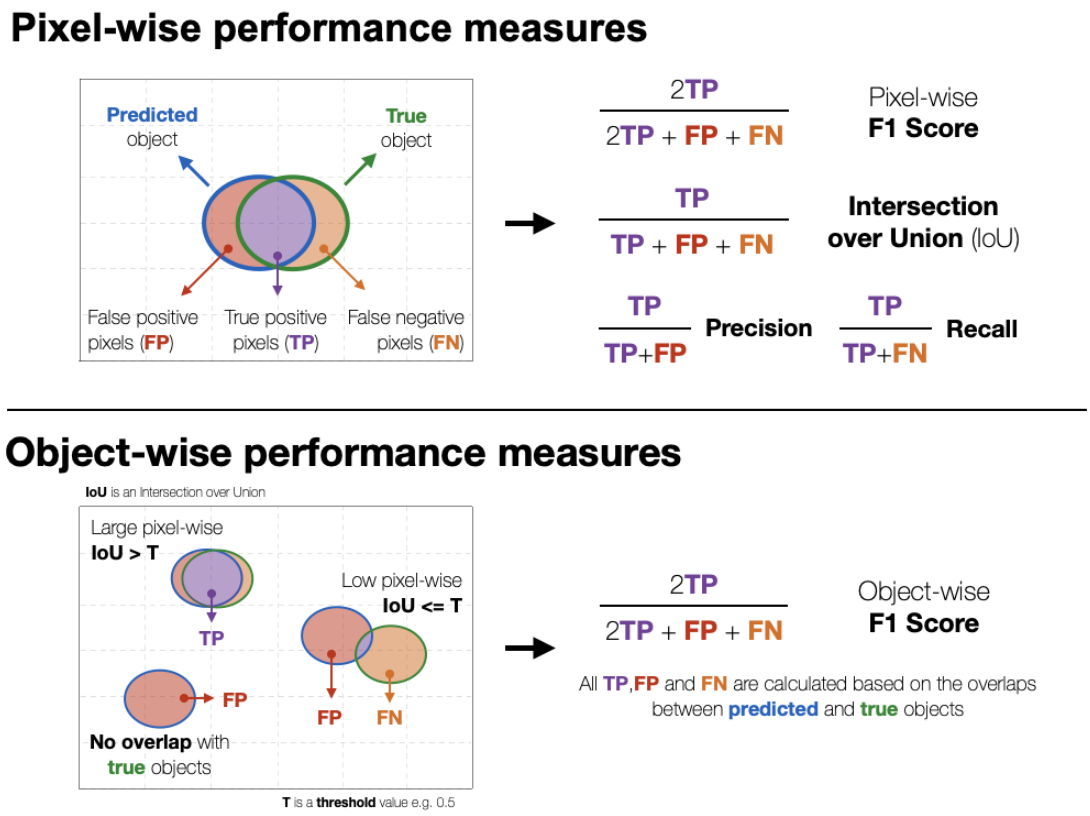

Figure 11. Pixel-wise and object-wise performance metrics (Ali et al., 2022).

Additionally, precision, recall and f1-score are popular ways to measure the network's performance. The f1-score metric combines precision and recall and has been the primary metric to evaluate model performance in this work. Figure 11 gives an intuitive explanation of the object-wise and pixel-wise f1-scores. In order to better understand a model's susceptibility to rotations, we defined our own metric, which we will refer to as f1-score_difference. We use the underscore to not confuse the reader when we talk about the differences between f1-scores. The

f1-score_difference is calculated by first calculating the f1-score of the original image and then subtracting the f1-score of the same image, which has been rotated either 90, 180 or 270 degrees. The results of the subtraction will show how similar the two predictions are. For a completely rotation equivariant model, the expected f1-score_difference would equal zero. We have chosen to use f1-score as the main metric due to its popularity in determining segmentation quality in the past (B. Chidester et al., 2019; Pan et al., 2018; Sten-Oliver Salumaa, 2018).

4. Related Work

The segmentation of the nucleus offers researchers a plethora of information. With the results of segmentation, it is possible to further determine cell types (Liu & Long, 2019) and go as far as to diagnose diseases (Chen et al., 2021) and evaluate the effectiveness of drugs (Dinh et al., 2022).

The first approaches to image segmentation were based on colour and edge- and region-based features (Sameena Banu, 2012). Such methods albeit simple, need manual input like finding the suitable threshold values for pixel intensity to operate well. Additionally, each of those methods has big drawbacks such as the resulting segments being too square with the region-based approach or that thresholding methods do not work on photos without any significant peaks on the histogram, where each peak denotes a different region on the image (Sameena Banu, 2012). A peak on these histograms could, for example, indicate the location of a nucleus and when a nucleus does not differ from the background, it does not show up on the histogram, thus it might stay below the threshold and not get detected. These and other reasons have motivated the development of new and more sophisticated ways of segmentation such as artificial neural networks (Pham et al., 2000).

It has become increasingly popular to use machine learning for cell nuclei segmentation (Pan et al., 2018). However high variability of shapes, sizes and cell types, image quality differences, and many other factors make accurate segmentation of nuclei a difficult problem even for artificial neural networks. Convolutional neural networks have proven to be one of the most efficient ways to address these challenges (B. Ehteshami Bejnordi et al., 2016; Litjens et al., 2017; Yun Liu et al., 2017).

Researchers also note that even though machines can outperform humans in specific cases, the algorithms are still far from perfect (B. Chidester et al., 2019; Bastiaan S. Veeling et al., 2018; Chidester et al., 2019). One of the reasons for imperfections is that convolutional neural networks are not equivariant to rotations, whereas cells as separate objects are. Rotating a cell does not change its value in any way, because there is no right orientation to look at a cell from. The hypothesis is that if a model were to be equivariant to rotation, it would recognise cell nuclei in a much easier manner due to the information it holds on different rotations of thousands of

cells from the dataset. Thus, the equivariant model would be more accurate in its predictions. One of the most common ways to leverage the cells' equivariance to rotation is by applying augmentations to the input data. Such augmentations typically include random rotations and flips (Yun Liu et al., 2017). Although augmentation may improve the generalisation capabilities of the model to some extent, it will not make a model equivariant for all layers, because the layers learn the rotations of an object collectively not individually. The shortcoming of the layers collectively learning the rotations is a waste of resources when they could be focusing on learning the details instead e.g. morphological features. In order to counter these shortcomings, group equivariance can be used, which is a robust way to learn rotations of the input and enables the model to focus on other details of the nucleus, not the orientation.

In 2018, Bastiaan S. Veeling published a research paper (Bastiaan S. Veeling et al., 2018) with his team introducing a rotation equivariant CNN for digital pathology. The model they developed was called a GDenseNet, which was based on the DenseNet architecture. The GDenseNet outperformed traditional CNNs in patch classification on histopathology images. The model was equivariant to rotation and reflection and was two percentage points more accurate than the regular DenseNet. One year later, Benjamin Chidester and his team published two research papers about rotation and reflection equivariant CNNs (B. Chidester et al., 2019; Chidester et al., 2019). The team developed two models called G-U-Net and CFNet. The G-U-Net was trained on histopathology images and the CFNet was trained on biomarker images. Both of the models outperformed the conventional CNNs. This work focuses on the analysis of fluorescent and brightfield images which are two modalities that have not been tested with rotation equivariant models.

5. Data

The dataset used in this work covers images of seven cell types. Each image has a complementary image referred to as a mask. Mask is a simplified copy of the original image that contains only three values, indicating the locations of the background, cell border and the nucleus. The size of the mask is 1080x1080, which is identical to the images. There are a total of 3024 images with corresponding masks divided into train, test and validation sets, with 2016, 504 and 504 images, respectively. The images are grayscale and come in two modalities, which will be described in more detail in parts 5.2 and 5.3: fluorescent and brightfield.

5.1 Cell types

The following chapter briefly summarises all seven cell types found in the seven cell lines dataset.

NIH3T3: One of the most common cell lines that originate from mice. NIH3T3 have an essential role in holding tissue together and helping with wound healing.

MDCK: Kidney cells that are typically taken from female dogs. The cell line is used in virus and vaccine productions.

HeLa: A durable cancer cell line from 1951, also known as the oldest and most frequently used human cell line. It originates from cervical cancer cells extracted from a woman named Henrietta Lacks.

MCF7: Human breast cancer cells that were discovered in 1970.

A549: Cancerous lung cells that are used for drug research and to study the disease.

HepG2: A human cell line that originates from a liver tumour.

HT1080: Soft-tissue cancer cell line that wraps itself around tendons, ligaments and muscles.

5.2 Fluorescent images

Fluorescent images rely on fluorescent molecules, which light up when light with a specific wavelength is shone on them (Zhang et al., 2020). The molecules attach themselves to DNA, thus illuminating the cells.

5.3 Brightfield images

Brightfield imaging is the simplest way of making pictures of cells. A light is shone through the Petri dish and an image is taken with the use of a brightfield microscope. See Figure 12 for an example.

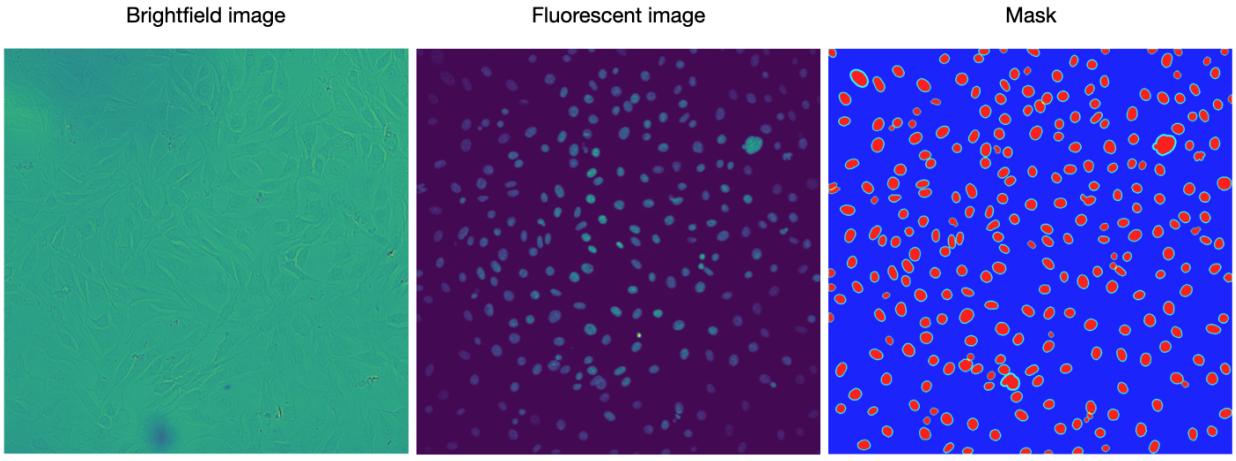

Figure 12. Brightfield (left) and fluorescent (centre) images with their respective mask (right). It can be concluded from the photo that brightfield images are much harder to analyse for the human eye since the contours of the cells are not as easily visible. The red areas on the mask indicate the position of cells on both images. Images are provided by PerkinElmer.

Brightfield images usually have very low contrast making it very hard even for a human eye to detect cells and nuclei compared to fluorescent images (Fishman et al., 2021).

The size 1080x1080 that brightfield and fluorescent images share in this dataset is not optimal for training, because there is already enough information on smaller images to successfully make

predictions. Additionally, there are standard practices such as image standardisation that are done before training.

5.4 Dataset preparation

The dataset preparation for training on all models included standardisation for the images and mask binarization from a three-class mask (background, border, nuclei) to a two-class (background, nuclei) mask. Mask binarization was done because the work wants to explore equivariance and for this the information about cell border is irrelevant. The dataset construction was done with the help of the proprietary Uniseg library created by the biomedical image analysis group at the Institute of Computer Science. The library utilises the Tf.Data module from Tensorflow to prepare images and masks for training. The library also allows additional arguments to conduct data augmentations, image and mask preprocessing, cropping and batch size. For the current thesis, crop sizes of 512 and 256 pixels were used depending on the memory consumption of the models.

Brightfield and fluorescent images described in this chapter have never been tested on rotation equivariant networks. These networks have previously been shown to improve the accuracy of semantic segmentation in the field of medical image analysis, in particular cell segmentation from histopathology slides (B. Chidester et al., 2019; Bastiaan S. Veeling et al., 2018). Such networks however have never been tested on the fluorescent and brightfield images (Bastiaan S. Veeling et al., 2018; Chidester et al., 2019). The next chapter will dive deeper into the components of rotation equivariant networks.

6. Methods

We have decided to look for existing architectures and models as building our own would likely fall outside of the scope of a BSc thesis. The two main architectures introduced are the U-Net and Densenet architectures, which have previously been used to develop rotation equivariant networks (B. Chidester et al., 2019; Bastiaan S. Veeling et al., 2018). Below, we will describe all the rotation equivariant variations of the two architectures that have been used in the work.

6.1 Baseline U-Net models

In order to test a hypothesis that equivariant models indeed improve the segmentation results, we need to compare the results of these models with non-equivariant models, in other words, baselines. If an equivariant model were to show better results than the baselines model, it would prove that group equivariance does improve semantic segmentation for cell nuclei. The first baseline model will be called “raw baseline”, which is going to be trained in a traditional way. The second baseline model will be referred to as the “augmented baseline” as it is trained using not only original images but also images transformed using the most common augmentations: rotations and flips.

Both aforementioned baseline models are based on the U-Net architecture (Olaf Ronneberger et al., 2015). The motivation to use U-Net came from articles showing its capabilities to perform semantic segmentation on brightfield images of cell nuclei (Ali et al., 2021; Fishman et al., 2021; Long, 2020). The U-Net model has an encoder-decoder architecture. The encoding part compresses the input information with a stream of convolutional and max-pooling layers into an encoder vector. After the input has been compressed, data gets sent through the bottleneck and from there, the decoder analyses the vector for localisation features through convolutional and upsampling layers (Olaf Ronneberger et al., 2015).

Sten-Oliver Salumaa implemented the U-Net model used in this work for the biomedical image analysis group at the Institute of computer science at the University of Tartu. See Figure 13 for a visual representation of the U-Net architecture.

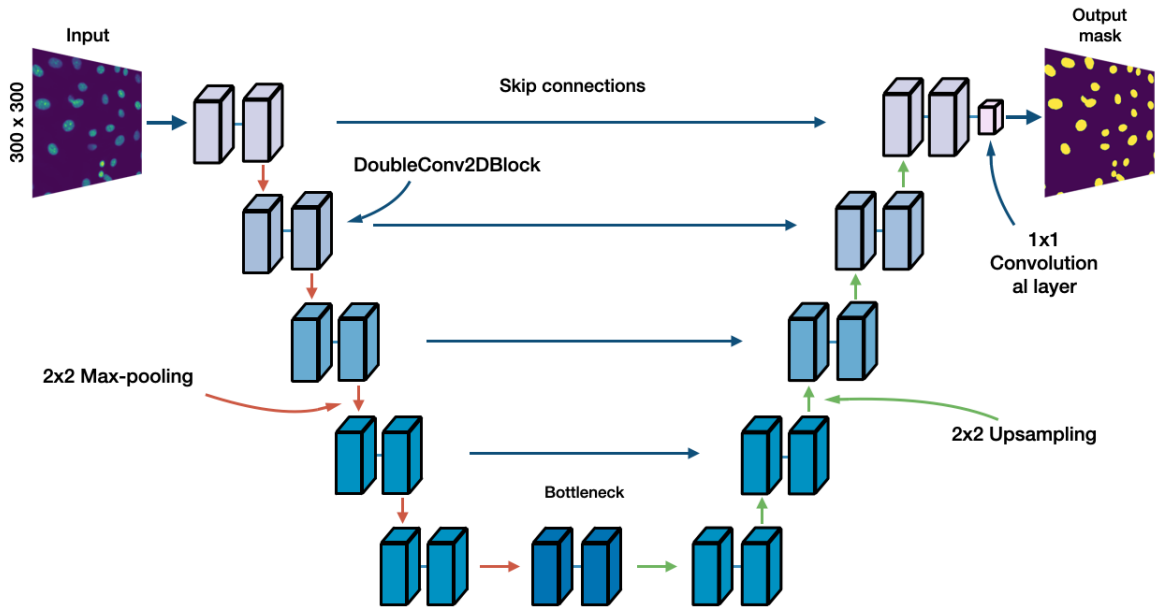

Figure 13. Representation of the U-Net architecture from input to output. The model takes a 300x300 fluorescent image of cell nuclei as input and outputs a mask of the same size. The U-Net on the figure comprises five levels of DoubleConv2DBlocks and a bottleneck. To the left of the bottleneck, the model depicts the encoder, and the right side shows the decoder. Skip connections go from each encoder level to the respective decoder part. DoubleConv2DBlocks consist of four convolutional layers, batch normalisation and ReLU. The orange arrows pointing down denote max-pooling layers, and the green arrows show upsampling layers.

The model has a single input channel and an output channel. The encoder part consists of five repetitions of two DoubleConv2DBlocks which contain convolutional layers, each with 3x3 kernel sizes, alternating with BatchNormalization and ReLu activation layers and ending with a max-pooling layer with a 2x2 kernel size and a skip connection to the decoder part. The skip connections are used so that the gradient would have an alternative route to perform backpropagation. The bottleneck has 2 DoubleConv2DBlocks, which means four convolutional layers similarly intertwined with batch normalisation and ReLU. The decoder part is symmetrical to the encoder, with the difference that it begins with a 2x2 upsampling layer and does not end with a max-pooling layer. There are 64 filters in each of the convolutional layers throughout the model.

6.2 Group Equivariant models

Because the U-Net architecture has shown noticeable cell segmentation results, this work will focus on making it equivariant using group convolutional layers. To not be too dependent on one model, ready-made models that have been shown equivariance properties based on articles on rotation equivariant networks were also incorporated into the experiments (B. Chidester et al., 2019; Bastiaan S. Veeling et al., 2018; Chidester et al., 2019).

6.2.1 Equivariant U-Net

The equivariant U-Net model differs from the baseline models because the convolutional and batch normalisation layers have been swapped with GConv and GBatchNorm. All group equivariant layers have been implemented using the GrouPy library (Taco S. Cohen & Max Welling, 2016). Additionally, the convolutional layer with a kernel size of 1x1 at the end is substituted with a group pool layer followed by a Keras GlobalAveragePooling2D layer and a Reshape layer to compensate for the shape loss by the average pooling layer (TensorFlow v2, 2019). The author has developed this model on the basis of the baseline U-Net model.

6.2.2 GDenseNet

GDenseNet is a model that has shown great promise in achieving better accuracy than augmentation with the Camelyon16 dataset (Ehteshami Bejnordi et al., 2017, Bastiaan S. Veeling et al., 2018). The model architecture is based on DenseNet (Gao Huang et al., 2016) and changes have been made to make it equivariant (see Figure 14).

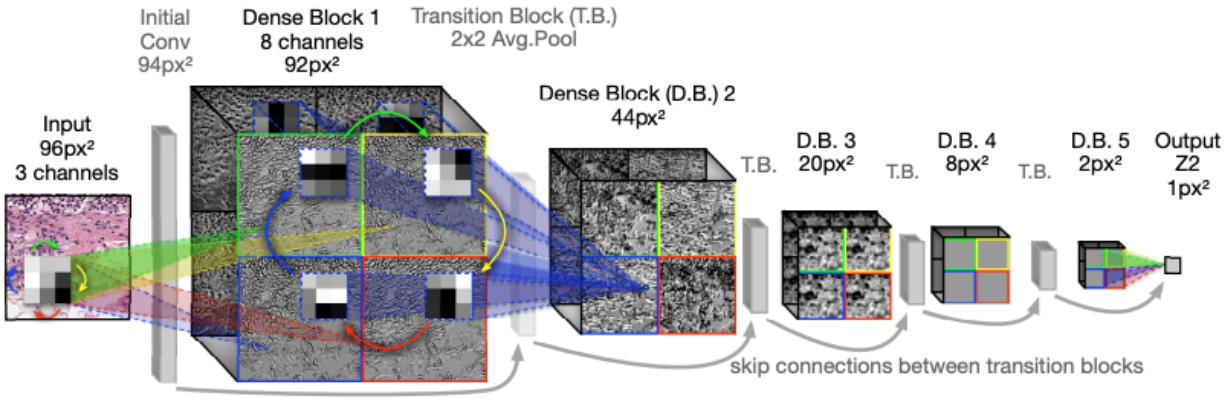

Figure 14. GDenseNet architecture visualised. The figure was created by the authors of the article "Rotation Equivariant CNNs for Digital Pathology", and the model displayed is equivariant to all 90-degree rotations (Bastiaan S. Veeling et al., 2018).

The overall architecture of the model consists of dense blocks with layers, which take all previous layers as input. The given model starts with a group convolutional layer GConv with strides 1x1 following five dense blocks consisting of a convolutional layer, batch normalization and ReLu activation layer consecutively. Each dense block ends with a transition block. Transition blocks include elements used in dense blocks and an additional average pooling layer. The final layer consists of a group pooling and a dense layer with the number of classes declared beforehand.

6.2.3 Simplified G-U-Net

The simplified G-U-Net is a name given to the model by the author. The model is a shallower and more primitive version of the original U-Net. The architecture contains four encoder blocks, where every block contains two group convolutional layers, each using the ReLU activation function and one max-pooling layer. Each encoder block has respectively 8, 16, 36 and 64 filters for each group convolutional layer. The last encoder block is connected to the bottleneck, which has two group convolutional layers, with 128 filters per layer. The bottleneck's other end is connected to the decoder part, which uses a transposed convolutional layer, concatenation with a respective encoder part at the same depth and two group convolutional layers. The group convolutional layers contain 64, 32, 16 and 8 filters per block and use ReLU activation functions.

All of the filters for the previous convolutional layers are with size 3x3. The model ends with a regular convolutional layer with a single 1x1 filter that uses the sigmoid activation function. The main difference between the equivariant simplified U-Net and the equivariant U-Net is that the simplified version does not use batch normalisation.

6.3 Training and evaluation

All models have been compiled with the Adam optimizer (Diederik P. Kingma & Jimmy Ba, 2014). Keras wrapper on top of Tensorflow was used to run the models. The following performance metrics were collected during training: BinaryAccuracy, Recall, and Precision (*Keras: The Python Deep Learning Library*, 2015). The two main evaluation metrics to assess model performance were f1-score and f1-score_differences. In order to understand how rotation equivariant the models are, we used the f1-score_difference metric introduced in chapter 3.3. The hypothesis is that the f1-score_difference would decrease as the models become more rotation equivariant via training using augmented images and group convolutional layers. The f1-score_differences were calculated for rotations of 90-, 180- and 270-degrees of the same image. After that, the rotated samples are separately deducted from the f1-score of the original image, resulting in three values.

7. Experimental results

The following chapter will introduce the results of experiments for the equivariant and simplified U-Nets, the GU-Net and the GDenseNet. However, refactoring the deprecated source code had to be done to obtain any results. Most of the libraries and models used for experiments were outdated and did not comply with the Tensorflow and CUDA environment versions used in the thesis. Chapter 6 focused on explaining the components of the models used in the work, and here the experimental results will be described, starting with the baseline model.

It is important to note that this works aims to improve the equivariance properties of existing networks but does not try to achieve complete equivariance.

7.1 Equivariance of the baseline models

The raw baseline and augmentation U-Net models were trained on the whole dataset. The raw baseline model achieved an average pixel-wise f1-score of 0.81, and the augmented model an average pixel-wise f1-score of 0.84 on the test set. The equivariance to rotations improved when the model was trained on augmented data. For the results to be comparable, the absolute values were taken from f1-score differences. The baseline model achieved a 0.030 average of f1-score differences and the augmentations model achieved a score of 0.017. The 0.017 f1-score difference from the augmentations model compared to the baseline, indicates that the model trained on augmented data is less susceptible to rotations and thus has shown better equivariance properties than the basic model (Figure 15).

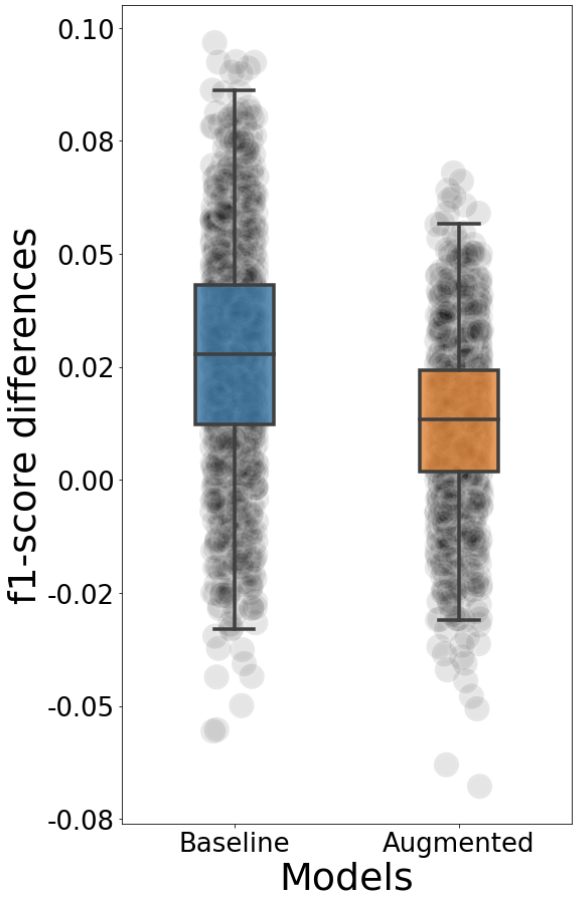

Figure 15. Box plot demonstrating f1-score_differences between rotated images for the baseline model and the model with augmented data. The left box shows f1-score_differences for a model without prior augmentation to the dataset, and the box on the right shows f1-score_differences with augmentations before training. Positive numbers show that the original was segmented better, and negative values indicate that rotated images were segmented better.

Results obtained using the baseline model, and the baseline model on rotation rich augmented data seemed to confirm the hypothesis that rotation and flipping improve the equivariance of the model to rotation. Hence the smaller average f1-score_difference value.

7.2 Results of equivariant segmentation models

In order to integrate equivariance into the convolutional neural networks, we have the used previously described approaches: the equivariant baseline U-Net model, the simplified G-U-Net, the GU-Net and a GDenseNet adopted for segmentation.

Equivariant baseline U-Net

The baseline U-Net model was used to make the model equivariant by substituting conventional layers with GConv, GBatchNorm and GroupPool layers (Bastiaan S. Veeling et al., 2018). In order to make the code run, refactoring had to be done due to the Tensorflow version incompatibilities of version 2, which is used in the work and version 1, which was used to build the group layers. One of the main difficulties were the dimensional differences that the functions in the versions brought forward. Some of the older functions had retained the dimensions of a single Keras Tensor in the model, whilst the newer version did not. Workarounds were found to compensate for the dimensional loss. However, after the integration, all predictions were an array of zeroes, indicating that the model could not learn.

Since there is no excellent way to debug a model and see the after each layer, different ways were tested to make the architecture work. Every different model variation has not been documented due to the minority of the changes and the lack of results it produced. The following descriptions of ideas tried and tested represent high-level overviews of what had been done in the effort to produce tangible results.

At first, we started removing different types of group equivariant layers one by one, starting with GBatchNorm. The outcome was the predictions being arrays of ones this time instead of zeroes. Without the group pool layer, the model was once again predicting zeroes. The removal of GConv2D layers was pointless since the papers described them as crucial for group equivariance. Since removing layers was not helpful, replacing the convolutional layer at the end with a Dense layer was tried, producing a final prediction similar to the DenseNet architecture. Due to the nature of the segmentation problem, where the model needs to predict each pixel label, not the whole image, then the input of 256x256 was given to the layer, which made the GPU run out of memory due to the size of the tensor. A smaller input size was also tried, but with no results.

Due to time constraints and the author exceeding the scope of a bachelor's thesis whilst doing minor tests on other models in parallel, other ways had to be more thoroughly looked into, and this model could be written off. The possible reasons for the model not working could be the

limited knowledge of a bachelor's student in regards to artificial neural networks, changes in different versions of Tensorflow v1 and v2 that the author was not aware of and did not become apparent when reading the documentation and finally the layers used having the incompatibleness to solve a segmentation problem.

GDenseNet

The second model that was tried was a variation of the DenseNet. The author attempted to transform this architecture which was initially designed for classification, into a segmentation model.

At first, the model was tested on a classification task based on the CIFAR-10 dataset to check if equivariance is observed. The test included an untrained GDenseNet model that predicted different rotations of the same image. The test was successful, and the assumption was made that conversion to a segmentation network should be possible and it would produce equivariant results.

Converting the GDenseNet included changing the final layer input units and other small tests such as removing as many group equivariant layers as possible and also, at times, adding layers, where jumps between layer sizes seemed too heavy. The main problems were still the lack of values in predictions and out of memory errors due to the sizes of the layers. As of this moment, two models with different architectures that used the same group equivariant layers had been tested, which led to a more thorough overall test of the GDenseNet model on the CIFAR-10 dataset. Turned out that the model could not learn even with the CIFAR-10 dataset and constantly predicted the same class regardless of the image.

In retrospect, it seems that the issue why the model did not work had again to do with either Tensorflow versioning or the model given in the Github repository is not the same or has subtle differences compared to the model used in the article (Bastiaan S. Veeling et al., 2018).

GU-Net

As the name suggests, GU-Net is a group equivariant U-Net model. GU-Net has been initially designed for cell nuclei segmentation (B. Chidester et al., 2019). This model was developed in 2019, making it less likely to have problems with versions. The model has been built using the familiar to the reader GroupPy library. Unfortunately, the model was developed using TensorFlow v1, which is deprecated and after refactoring, compiling errors persisted due to the TensorFlow Session not being initialised correctly. Additionally, the model kept getting stuck, throwing a new error after the previous one got fixed. After two weeks of trying to fix the errors, the model did not compile, and due to the time limit of the work, the search for other solutions seemed a more reasonable way forward.

Simplified G-U-Net

Because of the model's more straightforward structure, only the convolutional layers had to be replaced with the group equivariant layers. For this particular model, the baseline versions with the same architecture were also generated so that the results could be compared to models with the same architecture. This way, it can be examined how the model performance is affected by different layers, and architectural differences do not have to be included in the interpretation of results. The first baseline models were trained on brightfield images with the size of 256 x 256 pixels and 75 epochs. The outcome was very pixelated and blurry (Figure 16). Furthermore, the model did not recognise all the cells in the image. We concluded that the problem might occur due to the simplicity and size of the model and the complexity of brightfield images.

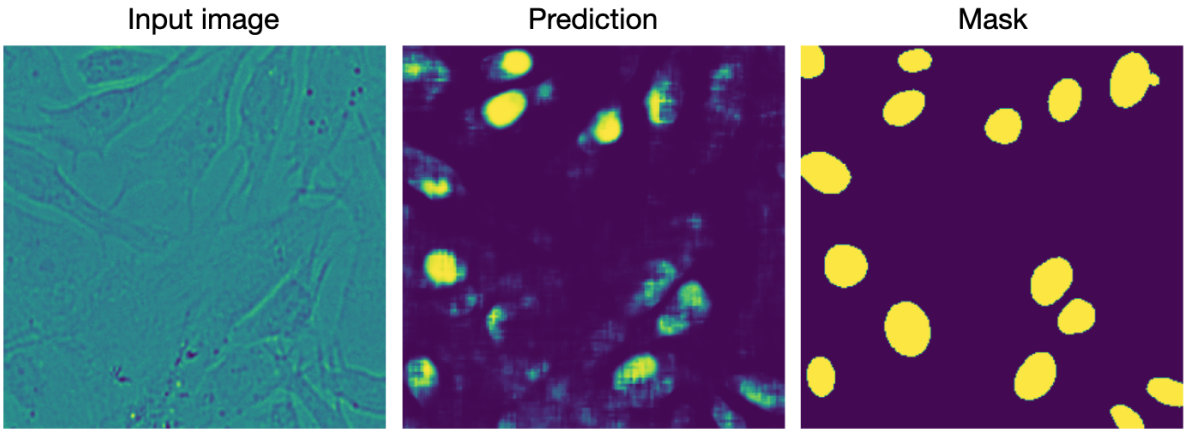

Figure 16. Pixelated prediction on a brightfield image. The input image size is 256 x 256 pixels. Bad performance is likely due to the simplified structure of the model.

Because fluorescent images are less complex for models to predict, we decided to use fluorescent images to achieve equivariance. In order to have more representable results, we took the absolute values from all f1-score_differences. The average object-wise f1-score_difference for the simplified model without augmentation on the dataset was 0.015. For the augmented dataset version, it was 0.014. The difference is not as sizeable in this case, and the possible reason is that fluorescent images are much easier to process than brightfield, thus, augmentation does not have a substantial effect on the equivariance properties of the model.

Additionally, the average f1-score_difference for the simplified G-U-Net was 0.010 and 40.25% out of the 6000 comparisons made resulted in 0, which means that two different predictions were identical (Figure 17). The 0.004 difference indicates that the layers had some kind of an effect on the model generalisation capabilities.

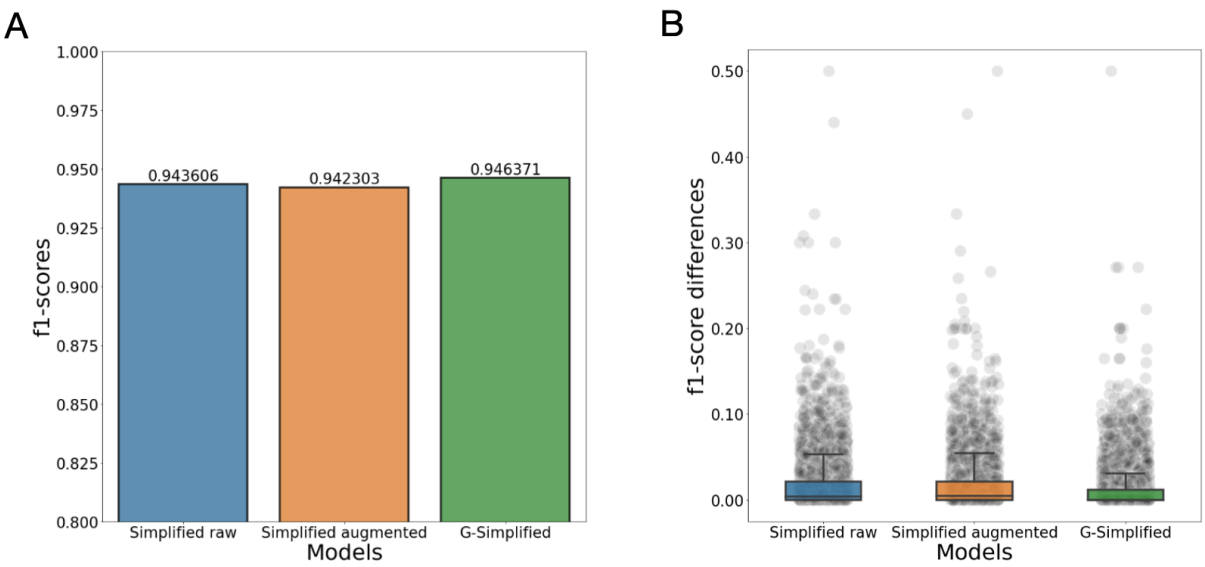

Figure 17. The left barplot (A) shows the average object-wise f1-scores of the three variations of simplified U-Net. As can be seen from the plot, the f1-score has not drastically improved through the group equivariant layers. The boxplot on the right (B) shows the object-wise absolute f1-score_differences for the simplified U-Net. As can be seen from figure B, the G-Simplified (simplified G-U-Net) model is substantially more equivariant to rotations than the other two. All models have outliers marked with black dots outside of the boxes. The figure proves that group equivariant layers make a model less susceptible to rotations, but not entirely.

In addition to looking at the f1-score_differences, we also investigated whether or not the f1-score improved. There were minor differences between f1-scores when comparing the simplified G-U-Net model to the baseline models (Figure 17), but no conclusions could be made because the differences were minor. Thus, it cannot be said that for the given dataset and with the simplified U-Net, the group equivariant layers improved the f1-score of the model.

8. Conclusions and Summary

Due to the vast amounts of data generated in medical imaging, accurate information processing is crucial. Because of the rapid development of machine learning in biomedicine, the usage of artificial neural networks for semantic segmentation of cell nuclei is higher than ever.

The segmentation of cell nuclei provides researchers with lots of information, from the number of cells in the image to doing morphological analysis on cell nuclei. These steps are often used in fields such as drug research or pathology. Despite artificial neural networks already achieving high accuracy in nuclei segmentation, there is still room for improvement. This work investigates the possibilities of using rotation equivariant networks for the semantic segmentation of fluorescent and brightfield images to achieve better performance than most common methodologies to solve the given problem.

Two architectures and four different variations of the two were tested to achieve equivariance: Equivariant U-Net, GDenseNet, GU-Net and the simplified G-U-Net. All of the models were trained on brightfield and fluorescent images separately. None of the models except the simplified U-Net produced any results due to implementation complexity and versioning issues. The simplified U-Net produced very pixelated predictions for the brightfield modality due to model simplicity, thus, fluorescent images were used for testing this model. We generated two baseline models for the simplified U-Net, one trained on a dataset without augmentations and the other with augmentations. The third model, simplified G-U-Net, utilised group equivariant layers. The simplified G-U-Net had a 28% lower f1-score_difference metric than the better baseline model, showing that group equivariant layers aid in model generalisation and increase equivariance capabilities. The differences between f1-scores in variations of the simplified U-Net were so small that no conclusions could be made.

It can be concluded that group equivariance increases the generalisation and equivariance properties of a model but based on current experiments it cannot be said it also improves the accuracy of a model. Moreover, the implementation of the models is complex, and documentation on the source code is scarce. The simplified U-Net could be further developed by increasing the model size by creating the needed batch normalisation and group pooling layers

individually. This way, model capabilities might improve even more, and it could also be used to segment brightfield images. The work done in this thesis will be of great help to the biomedical image analysis group at the University of Tartu to guide them in decisions regarding which methods to use in their future work.

9. Bibliography

Ali, M. A. S., Hollo, K., Laasfeld, T., Torp, J., Tahk, M.-J., Rinken, A., Palo, K., Parts, L., & Fishman, D. (2022). ArtSeg: Rapid Artifact Segmentation and Removal in Brightfield Cell Microscopy Images. *BioRxiv*, 2022.01.24.477467. https://doi.org/10.1101/2022.01.24.477467 (05/07/22)

Ali, M. A. S., Misko, O., Salumaa, S.-O., Papkov, M., Palo, K., Fishman, D., & Parts, L. (2021). Evaluating Very Deep Convolutional Neural Networks for Nucleus Segmentation from Brightfield Cell Microscopy Images. *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, *26*(9), 1125–1137. https://doi.org/10.1177/24725552211023214 (09/03/22)

B. Chidester, T. Ton, M. Tran, J. Ma, & M. N. Do. (2019). Enhanced Rotation-Equivariant U-Net for Nuclear Segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1097–1104. https://doi.org/10.1109/CVPRW.2019.00143 (28/12/21)

B. Ehteshami Bejnordi, G. Litjens, N. Timofeeva, I. Otte-Höller, A. Homeyer, N. Karssemeijer, & J. A. van der Laak. (2016). Stain Specific Standardization of Whole-Slide Histopathological Images. *IEEE Transactions on Medical Imaging*, *35*(2), 404–415. https://doi.org/10.1109/TMI.2015.2476509 (25/02/22)

Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, & Max Welling. (2018). *Rotation Equivariant CNNs for Digital Pathology*. https://arxiv.org/abs/1806.03962 (28/12/21)

Casper van Engelenburg. (2020, May 6). *Geometric Deep Learning: Group Equivariant Convolutional Networks*. https://medium.com/swlh/geometric-deep-learning-group-equivariant-convolutional-netw

orks-ec687c7a7b41 (20/04/22)

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *CoRR*, *abs/2102.04306*. https://arxiv.org/abs/2102.04306 (02/05/22)

Chidester, B., Zhou, T., Do, M. N., & Ma, J. (2019). Rotation equivariant and invariant neural networks for microscopy image analysis. *Bioinformatics*, *35*(14), i530–i537. https://doi.org/10.1093/bioinformatics/btz353 (28/12/21)

Diederik P. Kingma & Jimmy Ba. (2014). *Adam: A Method for Stochastic Optimization*. https://arxiv.org/abs/1412.6980 (14/02/22)

Dinesh D. Pati & Ms. Sonal G. Deore. (2013). *Medical Image Segmentation: A Review*. *2*(1). https://www.ijcsmc.com/docs/papers/january2013/V2I1201306.pdf (12/03/22)

Dinh, T. L., Lee, S.-H., Kwon, S.-G., & Kwon, K.-R. (2022). Cell Nuclei Segmentation in Cryonuseg dataset using Nested Unet with EfficientNet Encoder. *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–4. https://doi.org/10.1109/ICEIC54506.2022.9748537 (10/04/22)

Ehteshami Bejnordi, B., Veta, M., Johannes van Diest, P., van Ginneken, B., Karssemeijer, N., Litjens, G., van der Laak, J. A. W. M., & and the CAMELYON16 Consortium. (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*, *318*(22), 2199–2210. https://doi.org/10.1001/jama.2017.14585 (27/04/22)

Fishman, D., Salumaa, S.-O., Majoral, D., Laasfeld, T., Peel, S., Wildenhain, J., Schreiner, A., Palo, K., & Parts, L. (2021). Practical segmentation of nuclei in brightfield cell images with neural networks trained on fluorescently labelled samples. *Journal of Microscopy*,

284(1), 12–24. https://doi.org/10.1111/jmi.13038 (08/03/22)

Gao Huang, Zhuang Liu, Laurens van der Maaten, & Kilian Q. Weinberger. (2016). *Densely Connected Convolutional Networks*. https://arxiv.org/abs/1608.06993 (22/03/22)

Juan C. Caicedo. (2019). *Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images*. https://www.biorxiv.org/content/10.1101/335216v4 (22/03/22)

Keith Conrad. (2014). *Why groups?* https://kconrad.math.uconn.edu/blurbs/grouptheory/whygroups.pdf (23/03/22)

Keras: The Python Deep learning library. (2015). https://keras.io/ (14/01/22)

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. In D. A. Forsyth, J. L. Mundy, V. di Gesú, & R. Cipolla (Eds.), *Shape, Contour and Grouping in Computer Vision* (pp. 319–345). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-46805-6_19 (02/05/22)

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, *42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005 (09/04/22)

Liu, Y., & Long, F. (2019). Acute lymphoblastic leukemia cells image analysis with deep bagging ensemble learning. *BioRxiv*. https://doi.org/10.1101/580852 (18/04/22)

Long, F. (2020). Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinformatics*, *21*(1), 8. https://doi.org/10.1186/s12859-019-3332-1 (19/02/22)

Lu, Y., Qin, X., Fan, H., Lai, T., & Li, Z. (2021). WBC-Net: A white blood cell segmentation network based on UNet++ and ResNet. *Applied Soft Computing*, *101*, 107006. https://doi.org/10.1016/j.asoc.2020.107006 (05/03/22)

Olaf Ronneberger, Philipp Fischer, & Thomas Brox. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. https://arxiv.org/abs/1505.04597 (22/02/22)

Pan, X., Yang, D., Li, L., Liu, Z., Yang, H., Cao, Z., He, Y., Ma, Z., & Chen, Y. (2018). Cell detection in pathology and microscopy images with multi-scale fully convolutional neural networks. *World Wide Web*, *21*(6), 1721–1743. https://doi.org/10.1007/s11280-017-0520-7 (20/03/22)

Pham, D. L., Xu, C., & Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, *2*, 315–337. https://doi.org/10.1146/annurev.bioeng.2.1.315 (20/03/22)

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0 (24/03/22)

Sameena Banu. (2012). *The Comparative Study on Color Image Segmentation Algorithms*. *2*, 1277–1281.

Sergey Ioffe & Christian Szegedy. (n.d.). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Retrieved May 8, 2022, from https://arxiv.org/abs/1502.03167 (07/04/22)

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, & Demetri Terzopoulos. (2020). *Image Segmentation Using Deep Learning: A Survey*. https://arxiv.org/pdf/2001.05566v5.pdf (26/02/22)

Sten-Oliver Salumaa. (2018). *Convolutional Neural Networks for Cellular Segmentation* [University of Tartu]. https://dspace.ut.ee/bitstream/handle/10062/66166/thesis.pdf?sequence=1&isAllowed=y

(28/03/22)

Taco S. Cohen & Max Welling. (2016). *Group Equivariant Convolutional Networks*. https://arxiv.org/abs/1602.07576 (10/02/22)

TensorFlow v2. (2019). *Keras GlobalAveragePooling2D layer documentation*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/GlobalAveragePooling2D (12/02/22)

Vergés Llahí & Jaume. (2005). *Color Constancy and Image Segmentation Techniques for Applications to Mobile Robotics*. 33–40.

Wang, Y., Li, Y., Song, Y., & Rong, X. (2020). The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition. *Applied Sciences*, *10*(5). https://doi.org/10.3390/app10051897 (15/03/22)

Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Greg S. Corrado, Jason D. Hipp, Lily Peng, & Martin C. Stumpe. (2017). *Detecting Cancer Metastases on Gigapixel Pathology Images*. https://arxiv.org/abs/1703.02442 (26/03/22)

Zhang, X., Smith, N., & Webb, A. (2020). Chapter one—Medical imaging. In D. D. Feng (Ed.), *Biomedical Information Technology (Second Edition)* (pp. 3–49). Academic Press. https://doi.org/10.1016/B978-0-12-816034-3.00001-8 (20/03/22)

10. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Marten Türk,

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

Improving Semantic Segmentation of Microscopy Images Using Rotation Equivariant Convolutional Networks,

supervised by Dmytro Fishman, PhD.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Marten Türk
09 / 05 / 2022