Segmentation and Prediction of Mutation Status of Malignant Melanoma Whole-slide Images using Deep Learning

Elin Johansson and Fanny Månefjord

Master's Thesis in

Biomedical Engineering

Supervisors: Ida Arvidsson
Jonatan Eriksson
Melinda Rezeli

Examiner: Christian Antfolk

June 7, 2021

LUND UNIVERSITY

Department of Biomedical Engineering
Faculty of Engineering LTH

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

Malignant melanoma is an aggressive type of skin cancer. Gene mutations can make the disease progress faster, but specialised treatment exists. Today, gene mutations are detected with DNA-analysis which is costly and time-consuming. The aim of our thesis is to investigate whether deep learning can be used to differentiate whole-slide images of tumours with different gene mutations. This was done in two steps, first whole-slide images were segmented based on tissue types, and then classification of gene mutations was done.

The tissue segmentation was done using the deep convolutional network Inception v3, modified to a four class output. Image tiles of the size 244 x 244 pixels were used to train and evaluate the network, with F1-score 0.84 on tumour tissue.

Two different methods to predict mutation status were tested. First, image features extracted from the segmentation network were fed into binary classifiers to separate images of tumours with and without NRAS mutation. Due to unsatisfactory results, another method was tested. A new Inception v3 network was trained to distinguish between NRAS and BRAF mutated tumours. Data from the public database The Cancer Genome Atlas was used for training and evaluation. Further testing was done on two independent test sets. Only tiles with 90% or higher probability of being tumour according to the segmentation network were used. The classification network was tested tilewise (AUC 0.53-0.66) and patientwise with AUC-values around 0.60 for all datasets.

The results indicate that it is possible to separate tissue images based on gene mutations. We believe that deep learning networks like these have great potential of being integrated into diagnostics of malignant melanoma. This could lead to faster and more accessible gene mutation diagnostics around the world.

Keywords: deep learning, image analysis, malignant melanoma, tissue segmentation, mutation classification, Inception v3

Svensk sammanfattning

Malignt melanom är en aggressiv form av hudcancer. Genmutationer kan påskynda sjukdomsförloppet och spridningen av tumörer, men specialanpassad behandling finns att tillgå. Idag används DNA-analys för att upptäcka genmutationer, vilket är kostsamt och tidskrävande. Syftet med vårt examensarbete är att undersöka om djupinlärning (deep learning) kan användas för att hitta genmutationer från vävnadsbilder på malignt melanom. Detta har vi gjort i två steg, först genom att hitta tumörrik vävnad i mikroskopbilder, och sedan utföra klassificering av mutationer på dessa regioner.

Segmentering av olika vävnadstyper gjordes med hjälp av det djupa neurala nätverket Inception v3. Bildurklipp av storleken 244 x 244 pixlar användes för att träna och testa nätverket med F1-score 0,84 på tumörvävnad.

För att utföra klassificering av genmutationer testades två metoder. Först testade vi på att skilja på vävnadsbilder med och utan NRAS-mutationer med hjälp av s.k. *features*, numeriska värden som hämtats ut från segmenteringsnätverket. Försöket gav inte tillfredsställande resultat och därför tränades istället ett nytt Inception v3-nätverk till att göra klassificering av tumörbilder med NRAS- och BRAF-mutationer. Nätverket tränades på bilder från databasen The Cancer Genome Atlas och testades på ytterligare två separata dataset. Endast urklipp med mer än 90% sannolikhet att vara tumörvävnad enligt segmenteringsnätverket användes. Klassificeringen testades både urklippsvis (AUC 0,53-0,66) och patientvis med AUC-värden runt 0,60 för samtliga dataset.

Resultaten visar på att det är möjligt att skilja på bilder på tumörvävnad med olika genmutationer. Vi tror att liknande djupa neurala nätverk har stor potential att integreras i diagnostiken av malignt melanom. Det skulle kunna innebära snabbare och mer tillgänglig diagnostik av genmutationer.

Deep Learning - the Key to Revolutionise Skin Cancer Diagnosis?

POPULAR SCIENCE SUMMARY. **Malignant melanoma is an aggressive type of skin cancer that develops from moles. Gene mutations can make the disease progress faster, but if the mutations are detected, it is possible to specialise the treatment. Using deep learning as a complement in diagnostics is state of the art in many medical fields. It is a type of artificial intelligence that can detect patterns that are invisible for the human eye. In our thesis we have shown that it is possible to use deep learning to predict the mutation status of melanoma using microscopy images. With further development, this method could possible replace advanced, expensive and time consuming lab analyses. The technique could contribute to more rapid and accessible diagnostics around the world.**

Malignant melanoma is increasing at a high pace all over the world. With the exception of lung cancer in women, it is the cancer type that is increasing the most in prevalence. Specialised treatment is an important step of defeating cancer. Gene mutations in malignant melanoma enhance tumour growth which makes the disease progress faster. The two most common mutations are present in 40% and 20% of the cases, respectively. Since specialised treatment exists, detection of these mutations is crucial. Today, this is done with costly and time-consuming DNA analysis. However, recent studies show that deep learning can be used to detect the mutation status from tissue images alone. For a better chance at saving a patient's life, early detection and comprehensive patient investigation play vital roles. It is common to visually inspect cancer tissue in a microscope to mark out the tumour areas. However, this is a tedious task performed manually by a specialist.

Deep learning is a subfield of artificial intelligence and it can be used to automatically mark the different tissue types, without the need for human participation. In our thesis, we have trained a deep learning network that can identify four tissue types in melanoma biopsies which can assist in the segmentation procedure and save a great amount of time for the specialist. The segmentation network was trained and evaluated with a dataset from Skåne University hospital in Lund and its performance was visually evaluated on an independent dataset from the public database The Cancer Genome Atlas.

The segmentation network was used to find tumour-rich areas in the tissue and another deep learning network was trained to classify the mutation status. Even though further improvement is needed, the deep learning models developed in this thesis show high potential of being an integrated part of an automatic diagnostic tool. This tool would not only increase the speed but also make the melanoma diagnosis more accessible across the world since it only needs microscopy images and a computer.

Table of Contents

Acronyms

Adam Adaptive moment estimator.

ANN Artificial Neural Network.

AUC Area under the ROC-curve.

CEE Cross Entropy Error.

CNN Convolutional Neural Network.

H&E Hematoxylin and Eosin.

MM Malignant Melanoma.

PLS-DA Partial least square discriminant analysis.

ReLU Rectified Linear Unit.

RMSProp Root Mean Square Propagation.

ROC Receiver Operating Characteristic.

SVM Support Vector Machine.

TCGA The Cancer Genome Atlas.

WT Wild Type.

Chapter 1

Introduction

1.1 Background

The number of cases of Malignant Melanoma (MM) is increasing at a high pace all over the world. With the exception of lung cancer in women, MM is the cancer type that is increasing in occurrence the most [1]. Mutation of the oncogenes BRAF and NRAS are common in MM. The mutations are connected to an inferior prognosis but when detected early, targeted treatment can be possible. The analysis of the mutation status is done with time-consuming DNA-analysis [2]. Several recent research studies have explored the possibility of using deep learning and histopathological images for mutation status classification.

A common diagnostic procedure in medicine is histopathological evaluation, when tissue or cells are visually inspected using a microscope. For patients with suspected or diagnosed cancer, a tissue sample (biopsy) is collected. The frozen or chemically preserved tissue sample is sliced very thinly, and stained to make structures and components appear more distinctly. Hematoxylin and Eosin (H&E) staining is commonly used, where hematoxylin makes the nuclei blue and eosin stains cytoplasm and stroma pink. Histopathology gives a clear view of the disease and how it affects the tissue, since the preparation process preserves the tissue structure. Traditionally, the visual inspection of the slides is done manually by trained specialists. This is a tedious task, and to reduce the risk of human error it is common for more than one specialist to inspect each slide. Today the images are digital, which opens the opportunity to use computers for some of the inspection and analysis through automatic computer analysis [3]. It might take a while until a computer is authorised to set a diagnosis, but there are multiple ways the computers could decrease the workload for the specialists like highlighting interesting areas and interpreting high-volume data [4].

The concept machine learning refers to the procedure when a computer learns to recognise patterns and make predictions from data. Deep learning implements the machine learning concept using advanced deep neural networks that typically include an input layer, several hidden layers and one or more output values. After training, the network becomes an expert at finding features that represent the training data. It is essential to test a deep learning network on images that were not used during training and a popular approach is to supplement an institutional dataset with data from a public database such as The Cancer Genome Atlas

(TCGA) for either training or testing. Deep learning in medicine is a rising area of interest with a large amount of ongoing research projects. Possible applications are segmentation, disease staging and mutation status classification from image data [4].

1.2 Previous work

Several studies have used deep learning on histopathological images before, with promising results. Using whole-slide images of lung cancer tissue from TCGA, Coudray et al. [5] trained deep learning networks to predict several pathological attributes. An Inception v3 network was trained to classify the images into two of the most prevalent non-small cell lung cancer subtypes LUAD, LUSC or normal lung tissue. The result was similar to the classification by pathologists on the TCGA images. Additionally, they tested the classification on an institutional dataset with maintained performance. Furthermore, Coudray et al. attempted to predict the ten most common gene mutations in one of the lung cancer subtypes using Inception v3 and image tiles. Six of the ten gene mutations were shown to be predictable by the network. The mutation status of a whole-slide image was predicted by aggregating the probabilities of mutation of the image tiles. This was done by either calculating the average probability of the mutation, or the percentage of positively predicted tiles.

Kim et al. [6] used a CNN to classify histopathology images of primary tumours from 257 melanoma patients. They developed an automated model that first selects tumour-rich areas with high confidence, and second, predicts for the presence of mutated BRAF or NRAS. The network was tested on a test set from the institution, grouped into both ulcerated (broken skin membrane) and non-ulcerated tumours, and different tumour thickness. An additional test set from TCGA was used. The performance was higher for thinner BRAF and non-ulcerated NRAS tumours.

Dolezal et al. [7] explores how deep learning can be used to predict BRAF-RAS gene expressions. Slides of thyroid neoplasm were used to train a neural network (Xception) to predict the tumour subtype. Their results demonstrate that the histologic features associated with BRAF-RAS spectrum are detectable by deep learning, and they pose that the findings can help to give the patient a correct diagnosis quickly.

Inception v3 was trained to map driver mutations in papillary thyroid carcinoma (thyroid cancer) to histopathological subtypes by Tsou et al. [8]. The whole-slide images were obtained from TCGA and cropped into non-overlapping tiles of size 512 x 512 pixels. The model was trained on the tiles from the training set and the model with the highest accuracy on the validation set was chosen as the final model. Firstly, a tile was classified as mutated if it had over 80% predicted probability of one of the classes, otherwise it was classified as uncertain. Secondly, a whole-slide was predicted as mutated if over 80% of the tiles belonged to one class. The model resulted in an AUC of 0.88 on the validation set and 0.95 on the test set. The RAS mutation had higher accuracy than prediction of BRAF.

Van Zon et al. [9] successfully created a system that classified whole-slide images into the classes melanoma, nevus (harmless mole) and normal tissue. A U-net architecture was trained to perform semantic segmentation (assigning a class to each pixel) and the output of the U-net was fed into a Convolutional Neural Network (CNN). The CNN predicted a class label for the whole-slide image with a success rate of 173 out of 176 on the melanoma slides and 57 out of 62 on the nevi slides.

Couture et al. [10] used deep learning to predict breast cancer grade, ER status (expression of estrogen receptors) and subtypes with high accuracy. As a pre-processing step, features and properties of the images were captured with VGG16, which is a CNN. The network was pre-trained on the ImageNet dataset. The features were extracted by using the output from some convolutional layers (before max-pooling). A Support Vector Machine (SVM) classifier was trained to use the features from VGG16 as input.

1.3 Aim

This master's thesis aims to investigate whether it is possible to extract mutation status of BRAF and NRAS, solely from whole-slide images of MM tumour tissue. A segmentation network will be trained to find tumour-rich areas that can be used in a classifier. A part of the project will be to test the system on different datasets to investigate the generalisation performance.

1.4 Structure of the thesis

The introduction of the thesis is followed by a theory chapter, where the readers will gain knowledge about the foundations of MM, artificial neural networks and evaluation methods. Readers already familiar with these concepts can skip these parts. A presentation of the data and processing methods are presented in Chapter 3 and the method in Chapter 4. The results are presented in Chapter 5 and a discussion about the results can be seen in Chapter 6. Chapters 4, 5 and 6 are divided into the sections Segmentation, Feature extraction and Classification of BRAF versus NRAS.

The authors have contributed equally throughout the project.

Chapter 2

Theory

2.1 Malignant melanoma

MM is caused by malignant transformation of melanocytes, the skin cells that are specialised at producing melanin. Melanin makes the skin darker and serves as protection against UV radiation. The number of cases of MM is increasing at a high pace all over the world. The average age at diagnosis is 57 years and 75% of the patients are younger than 70 years old. The low average age differentiates MM from most other tumour cancers [1].

MM is divided into stage 0 to IV, where stage 0 is a tumour on the top layer of the skin. Deeper and/or more spread cancer cells correspond to higher stages and at stage IV the cancer has spread beyond the regional areas and lymph nodes to distant sites in the body [11]. A metastasis is when the tumour has spread from its primary site. MM stands for one third of deaths due to cancer, and the five-years survival rate decreases drastically from 98.4% to 22.5% if the cancer evolves from stage I to stage IV [12]. It is therefore of greatest interest to find the MM tumour as early as possible.

2.1.1 BRAF and NRAS mutations

Mutation of the oncogenes BRAF and NRAS are the most common genetic alterations in MM, detected in approximately 40% and 20% of the cases respectively. BRAF and NRAS mutations are shown to activate pathways that enhance tumour growth and, thereby, disease progression. The cancer tumours where no mutation can be found are called Wild Type (WT) [2]. The most common mutations in BRAF are located in the position V600 and 90% of the mutated BRAF have the mutation V600E, where valine is substituted with glutamic acid at position 600. It is possible to inhibit the activity of the mutated BRAF V600E protein with pharmaceuticals [13]. It has been shown that targeting the cellular activity of melanoma cells is effective since it delays tumour progression and prolongs patient survival [14].

Patients with stage III and IV MM are tested for mutations. Suspicious lesions are biopsied, embedded and sliced thinly. Every other slice is stained with H&E and the rest are used for immunohistochemistry or DNA sequencing [15].

2.2 Artificial neural networks

An Artificial Neural Network (ANN) is a computing system designed to process information. It consists of neurons (often denoted nodes), which are connected processors, inspired by biological neurons. The simplest type of an artifical neuron is a perceptron. The perceptron consists of inputs, an operator and an output. The neuron calculates a weighted sum of the inputs and has an activation function to calculate the output, see Equation 2.1. The output is denoted y, φ is the activation function, $\omega = (\omega_1, \ldots, \omega_N)$ the weights and $x = (x_1, \ldots, x_N)$ are the inputs. Neurons that are connected to each other are called an ANN [16].

$$y = \varphi(\omega_0 + \sum_{n=1}^{N} \omega_n x_n) \tag{2.1}$$

The trainable parameters in the model are the biases and weights. Each layer of nodes has a bias and the connection between two nodes has a weight. In feed forward networks, input values are propagated forward through layers of nodes to provide the output. A perceptron can have many layers, resulting in a deeper network and the layers between the input and output layers are called hidden layers. This kind of perceptron is called a multi-layered perceptron and a general scheme is presented in Figure 2.1. Each layer uses the output from a previous layer as input, including the bias term. More hidden layers allow the network to learn representations of the data with several levels of abstraction and this kind of model is called a deep learning network [17]. When all nodes in one layer is connected to all nodes in the next layer, it is called a dense layer. The dense layers in a model have many weights, and a way to decrease the number of trainable weights is to use a CNN [16].

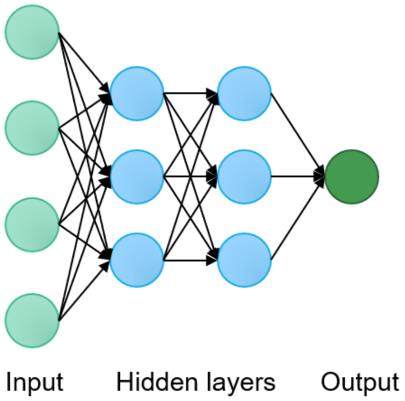


Figure 2.1: Schematic image of a multi-layered perceptron. The number of hidden layers and the number of nodes in every layer can be adjusted.

2.2.1 Convolutional Neural Networks

CNNs are neural networks that have at least one convolutional layer and they are commonly used for image analysis purposes. The input of the network is usually a multidimensional array of data and it can be of various sizes. In this project, 2-dimensional images with three colour channels are used as input. The input is convolved with a kernel. A kernel in a CNN is a matrix which will slide over the image, from top left to top right, and then all rows until the end of the image, performing the mathematical operation convolution. The formula for convolution of an image I with a 2D kernel K giving the output S, is presented in Equation 2.2. The numerical values of the kernel are called weights. The benefits of using convolution in a neural network are sparse connections and parameter sharing, since the weights are stored in the kernel that is used several times [16].

$$S(i,j) = (I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m,j-n) \tag{2.2}$$

The stride describes how much the kernel is moved before next convolution. A larger stride will reduce the number of pixels in the output. Figure 2.2 shows how a 3 x 3 kernel will move over an image with stride 2 [16]. If the kernel is smaller than the input, the weights will be shared and fewer parameters need to be stored. The kernels will extract elementary features (corners, edges) and these features are combined in other layers to detect higher level features. When multiple layers are used in a CNN, features of different detail level will be extracted and CNNs are therefore useful for image analysis tasks. It is also possible to detect how the features are positioned with respect to each other. The first convolutional layers in the network can detect small features such as corners and edges and the later layers can put these together to detect full objects [18].

Padding is a technique to preserve the image size and the information in the outer part of the images by adding pixels around the image. Without padding, the size of the image decreases due to the convolutions with the kernel. Padding can be done in different ways. Zero padding is when zeros are added around the image and copy padding is when pixels with the same intensity as the closest pixel are added [16].

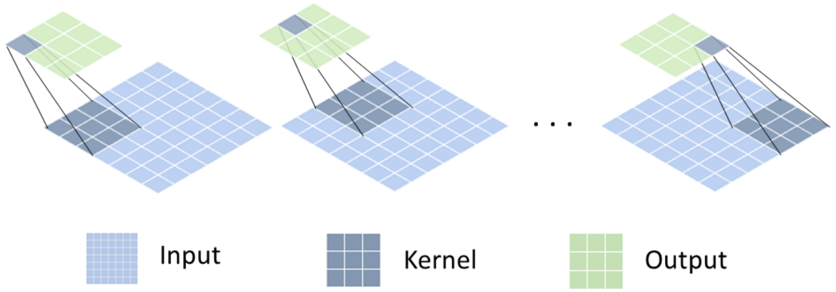


Figure 2.2: Convolution of a 2D matrix. A 3 x 3 kernel moves over an input image with stride 2. The resulting image is smaller than before the convolution.

2.3 Training a neural network

2.3.1 Loss minimisation and supervised learning

A neural network is trained to find patterns in the data by changing the values of the weights. The values are changed to minimise the difference between the predicted output and the target, by minimising a loss function [16]. This process is iterated, which is called training. Training is done with the data divided into batches, where the weights are updated after each batch (one iteration). When all batches have been seen by the network once, it has been trained one epoch. The updates of the weights are calculated using back-propagation, which is a practical application of the chain rule for derivatives. By starting at the output, and propagating back towards the input, all gradients are calculated. In supervised learning, when ground truth (or gold standard) is available, minimising the loss means finding the weights that make the network's output as close as possible to the ground truth. There are different types of loss functions suitable for different problems [17].

For classification networks with multiple classes, Cross Entropy Error (CEE) is typically used as the loss function. The equation for categorical CEE with four classes is presented in Equation 2.3, where d is the target and y is the output for N examples. For each class k, the target d_k is 1 for the correct class and 0 for the other classes. This is a multidimensional minimisation problem that likely has many local minima. Converging towards a minimum does not guarantee finding the optimal solution. Several attempts with different settings and starting points might be necessary to find the global minimum [17].

$$CEE = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{4} d_{nk}\ln(y_{nk}) \tag{2.3}$$

2.3.2 Activation functions

Biological neurons get activated when the inputs reach a threshold level. Artificial neurons are designed to work in a similar way, with an activation function that is increasing with a threshold behaviour. The Rectified Linear Unit (ReLU) activation function is recommended to use in most feed forward networks. The function is presented in Equation 2.4 and Figure 2.3. ReLU is piecewise linear and thus it preserves many of the properties of linear models that make models generalise well [16].

$$\varphi(x) = \max\{0, x\} \tag{2.4}$$

CEE is commonly used together with the softmax output activation function. The softmax function can be used to represent a probability distribution function of K different classes. The goal is that the correct class will have the highest value, and thus the highest probability. The outputs are bounded between 0 and 1 and the sum of all outputs is 1, and therefore they can be interpreted as

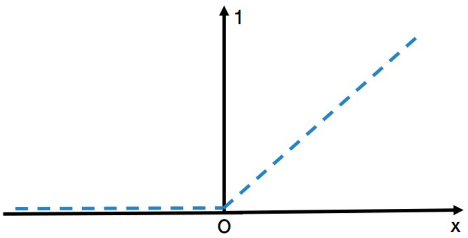


Figure 2.3: The Rectified Linear Unit (ReLU) activation function. This activation function is commonly used in neural networks.

probabilities. The softmax function is defined in Equation 2.5 for $i = 1, \ldots, K$, where $\boldsymbol{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$ is the input vector of K numbers [16].

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.5}$$

A pooling function can be used after the activation function to further modify the output of the layer. The function looks at a rectangular neighbourhood (for 2D data) and replaces the outputs by a single value. In max pooling, the maximum value of the neighbourhood is chosen, and in average pooling the average value of the neighbourhood is the output [16].

2.3.3 Optimisation methods

A popular optimisation method to minimise the loss is the Adaptive moment estimator (Adam) optimiser. The algorithm is gradient-based, which means that the minimisation is done based on the gradient with respect to the weights and biases. Additionally, it takes into account an adaptive estimation of lower-order moments. The algorithm uses exponential moving averages of the gradient (first moment) and the squared gradient (second moment) multiplied with tunable constants (β_1 and β_2). One of the important features of the Adam optimiser is its adaption of steplength for the updates. Near an optimum the step length decreases, allowing the algorithm to come closer to the real minimum [19].

An alternative to Adam is Root Mean Square Propagation (RMSProp), where the learning rate is adapted by dividing it by a moving average of the root squared gradient. The hyperparameters used in RMSProp are epsilon (ϵ), momentum and weight decay. ϵ is a constant added to avoid dividing by zero. Momentum determines how quickly the contributions of the previous gradients decay. The weight decay is used to prevent the loss of becoming too large [16].

2.3.4 Batch normalisation

One complication during training of deep neural networks is that the distribution of the input changes when the weights of the previous layer are updated. This makes the minimisation more difficult and slows the training down since lower learning rates are required. The adaption to changing input distributions can be described as 'chasing a moving target'. Batch normalisation is a method that is frequently used in deep networks to counteract the issue of the varying input distributions. Added in the network architecture, batch normalisation applies normalisation of each training batch. Batch normalisation allows for higher learning rates and less careful parameter initialisation which speeds up the training process [20].

2.3.5 Normalisation of data

Training an ANN with real data does in general demand some pre-processing of the data to improve the training performance. Statistical normalisation of the data is done by using the mean value and the standard deviation of each input vector. The equation for statistical normalisation is presented in Equation 2.6 where x'_i is the normalised value, x_i the former value, μ is the mean and σ the standard deviation. Normalisation can improve the training by making the network update the weights equally fast for all inputs. The risk of training on statistical outliers is reduced [21].

$$x'_i = \frac{x_i - \mu_i}{\sigma_i} \tag{2.6}$$

2.3.6 Generalisation performance

The goal when training a deep learning network is to get the best performance possible on data that is new to the network but comes from approximately the same statistical distribution as the training data. This ability is called generalisation performance. To choose the network architecture and evaluate the training, a common approach is to separate the data into three exclusive sets: training data, validation data and test data. The training data (usually the biggest part) is used to train the model and the validation data is used to continuously evaluate the performance and to choose the settings. When all the settings of the model are decided and the model has been trained to a good performance on the validation data, the test set is used to check the performance of the model on unseen data. The main challenge is to train the model on the training set to gain statistical information and learn features that describe the distribution but not details only found in the training set [16].

2.3.7 Overfitting

If the model is trained over too many epochs or the training set is too small for the network size, the model can be overfitted to the training data. The opposite of overfitting is underfitting, where the model has not learned enough. An overfitted model will perform extremely well on the training data but poorly on unseen data, whereas an underfitted model will have poor performance on both training data and unseen data. A common way to evaluate whether the model is either overfitted or underfitted is to plot its performance on the validation and training sets for every epoch of training [22]. In Figure 2.4 the accuracy of the training set is increasing while the accuracy of the validation set is decreasing after a certain point, and the model is overfitted. There are numerous approaches to reduce the risk of overfitting.

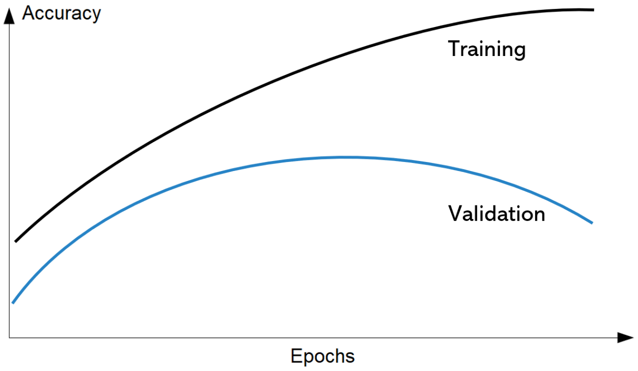


Figure 2.4: The accuracy of the training and validation data plotted against the training epochs. For an overfitted model, the validation accuracy is decreasing while the training accuracy is increasing. The goal is to stop training before the validation accuracy starts decreasing.

Dropout

In dropout regularisation some nodes are ignored, 'dropped', during each epoch of training [22]. Dropout makes the model learn more general features and prevents a small subset of nodes from becoming dominant. When dropout regularisation is used, one must choose the probability of dropping a node.

Transfer learning

The training of a deep neural network is computationally demanding and requires a large amount of training data. Even if a big dataset is available, it is common to use pre-trained weights instead of starting training with randomly initialised weights. The weights can thereupon be fine-tuned by continuing training and this method is called transfer learning [23]. A model developed for one task can therefore be reused as the starting point for another task [16].

Augmentation

Data augmentation can be used to increase the size of the data set without adding new images. Augmentation changes the image by e.g. rotation, flipping or changing other appearances while the label is kept. Data warping is a simple and often safe way to increase the size of the dataset. For histopathology images, flipping and rotating an image still makes the information in the image correct and useful for training, since the images do not have a correct orientation (compared to if you train a network to separate images with the numbers 6 and 9). In addition to geometrical warping, the colour of the image can be changed to increase variation of the training data. This can be useful to make the network's generalisation performance higher on data from different microscopes or staining techniques, especially if all the training data comes from one or a few hospitals [22].

2.3.8 Training with unbalanced classes

The deep neural network becomes an expert at the task it has been trained to do. Class imbalance in the training set usually results in the trained model over-predicting the majority class due to larger prior probability, which could result in poor performance. The effect of class imbalance during training can be prevented by alterations in sampling of the classes, by adding weights to the loss function, or a combination of them. Balancing through weights does not alter the distribution of the training data, but the learning process is shifted in favour of the minority classes. Weight functions suppresses class imbalance by giving a large weight to the minority class in the loss function and a small weight to the majority class. This leads to a higher penalty for miss-classifying the minority class [24].

2.4 Inception v3

Inception v3 is an architecture of a convolutional neural network. Microarchitecture units, called inception units (see Figure 2.5), increase the network's robustness against translations in the input images and its non-linear learning abilities. Each inception unit consists of several convolutional layers with non-linear activation functions. These inception units were first introduced by Szegedy et al. [25]. The inception unit acts as a feature extractor in multiple levels, by performing several convolutions within the same unit of the network. The outputs are stacked and then used as input to the next layer in the network [26]. The inception architecture is suitable for histopathology tasks since it can handle multiple resolutions and it has been successfully adapted to other tissue type classifications [5].

2.5 Evaluation methods

2.5.1 Accuracy

The accuracy of a model is the number of correctly classified samples divided by the total number of samples. The accuracy gives a number on how well the model can predict the classes correctly. Although, with multiple and/or imbalanced classes accuracy can be a misleading metric. If a large part of the data is from one class, the model could get high accuracy by only classifying objects of that class correctly. For these cases, more methods are needed to trust the evaluation and some examples are stated below [27].

2.5.2 Confusion matrix

To evaluate the network's ability to predict class labels in binary or multi-class problems correctly, a confusion matrix can be used. An example matrix for a two class-problem can be seen in Figure 2.6. Each predicted sample will be placed in

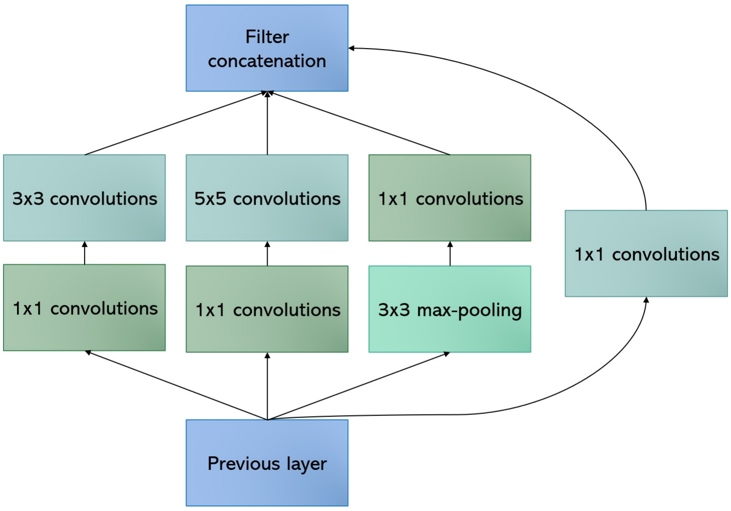


Figure 2.5: An inception unit. Multiple inception units are used in the Inception v3 deep learning network. The last operation concatenates the outputs from the parallel streams.

one of the boxes, with the true label vertically and the predicted label horizontally. Correct predictions will be in the green diagonal boxes [27].

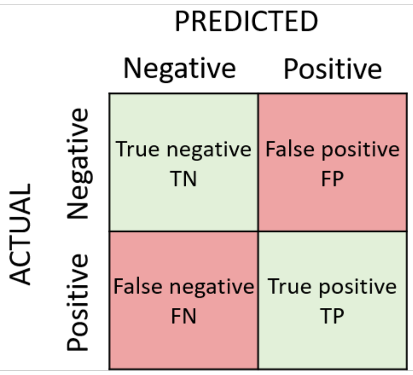
From the confusion matrix it is possible to see if the model predicts one or some of the classes more frequently. This kind of information is helpful to improve the network performance. The specificity, precision, recall and F1 score of the classification can be calculated from the information in the confusion matrix. Specificity is the fraction of true negatives and all samples that are actual negatives (see Equation 2.7). Precision is the fraction of true positives and all positive predictions (see Equation 2.8) while recall is the fraction of true positives and all cases that truly are positives despite their prediction (see Equation 2.9). Recall is also called sensitivity. The F1 score is a function of precision and recall (see Equation 2.10), that is used to find a balance where both of them are as high as possible [28].

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.8}$$

$$\text{Recall} = \text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.9}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.10}$$



		PREDICTED	
		Negative	Positive
ACTUAL	Negative	True negative TN	False positive FP
	Positive	False negative FN	True positive TP

Figure 2.6: A confusion matrix of two classes. A confusion matrix has the true labels in the vertical direction and predicted label horizontally. Correct classifications end up in the green diagonal boxes.

2.5.3 Receiver operating characteristic curve and area under the curve

The performance of a classifier can be evaluated using the Receiver Operating Characteristic (ROC) curve. The curve only works for binary problems, and must be plotted classwise for multi-class models. For each class the false positive rate (items incorrectly classified as this class) is plotted on the x-axis against the true positive rate (correct classifications of this class, also called sensitivity or recall) on the y-axis. The dashed black diagonal in the plot in Figure 2.7 represents random guessing whereas a good classifier reaches as far as possible towards the top left corner. The Area under the ROC-curve (AUC) brings a more comparable, numerical value from the ROC-curve. The AUC-value is between 0 and 1 and measures how good a classifier is by calculating the area between the ROC-curve and the x-axis. A perfect classifier has the AUC-value 1, which happens when the ROC-curve reaches the top left corner (green in the figure) and the random guessing has the AUC-value 0.5 [27].

2.5.4 Cross validation

When the dataset is small, the evaluation of the algorithm can be statistically uncertain since the test set is too small to yield an accurate generalisation estimate. Cross validation is a method to use all data in the estimation of the mean test error. Training and validation sets are created repeatedly of random subsets of the whole dataset. In K-fold cross validation, the dataset is divided into K non-overlapping subsets. In the first round, the first subset is used as validation data and the rest is used for training. This process is iterated, and all subsets are used as validation subset once. In total, K slightly different models will be created and tested and the final estimate of the generalisation performance is the average of the K validation results [16].

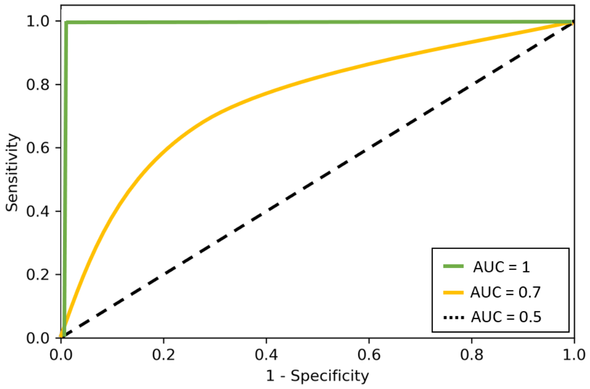


Figure 2.7: Receiver operating characteristic curve and area under the curve for different models. The green model is the best since it reaches the furthest towards the top left corner and has the largest AUC-value.

2.6 Segmentation

Image segmentation is the process of clustering pixels in an image in different classes. In semantic segmentation, all pixels belonging to a class are assigned a certain label, and if there are multiple objects of the same class in the image, they are assigned the same label. This requires very detailed gold standard annotation. Another approach of image segmentation is to split the image into small image tiles and assign each tile a class. Deep learning and convolutional neural networks can be used to perform the segmentation [23].

2.7 Classification

2.7.1 Image features

Images can be explained by features and these can in turn be used for classification tasks. The features can be hand crafted to describe certain characteristics of the image, i.e. intensity or width of an object. The aim is that the features alone can be used for an application instead of the image. Instead of hand crafting, the features can also be calculated by a CNN and extracted automatically to gain high level information about the image. Feature selection is the process of selecting a relevant subset of features, to reduce dimensionality and simplify the task. The number of features is reduced by removing irrelevant features or a feature that strongly correlates to another [29].

2.7.2 Logistic regression

Logistic regression is a regression model used for binary data, where the output variable is interpreted as a probability, since the value ranges between 0 and 1.

The logistic model is based on the mathematical logistic function, presented in Equation 2.11 [30].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.11}$$

2.7.3 Random forest classifier

The random forest classifier consists of multiple decision trees. The functionality of a decision tree is to use features to separate data into branches in a way that make the leaves (the two output nodes) of each branch as different as possible but with homogeneous groups inside the leaves. The tree is built up with the training data and depending on the decisions, the same training set can result in many different tree architectures. A random forest classifier uses a large amount of nearly uncorrelated decision trees that work as an ensemble. Each of the trees predicts a class for the tested data point and the majority prediction is chosen. To use the largest vote of many trees gives more robustness than to only use one classifier [31].

2.7.4 Partial least square discriminant analysis (PLS-DA)

PLS-DA uses dimensionality reduction and discriminant analysis in one combined algorithm and it is commonly used for classification tasks with few data points and high dimension (a large number of features). The algorithm is very flexible since it does not assume the data to fit any specific distribution [32].

2.7.5 Support Vector Machine (SVM)

A supervised (explained in Section 2.3.1) machine learning method used for classification is the SVM. The algorithm is mapping observations of the training data in a high dimensional feature space and separates the classes by a multidimensional decision plane [33]. The advantages of an SVM is that it is still effective for high dimensional data and it is versatile. A kernel function is a set of mathematical functions that are used to transform the input data into a desired form. Several different kernel functions can be used to determine the decision plane, such as linear, Gaussian and radial basis function kernels [34].

Chapter 3

Data

3.1 Data description

The data in this project is from three sources, a cohort from Lund University hospital, images from the TCGA database and a cohort from Semmelweis University. The cohorts consist of whole-slide images of flash frozen MM tumours (both primary tumours and lymph node metastases). The tissue was stained with H&E, where hematoxylin colours the nuclei purple and the other part of the tissue is stained in different shades of pink by the eosin [35].

The Lund cohort consists of 144 surgically removed flash frozen MM tumour tissues, with a majority of lymph node metastases, from the Melanoma biobank (BioMEL), Region Skåne, Sweden. The study was approved by the Regional Ethical Committee at Lund University, Southern Sweden (DNR 191/2007, 101/2013 and 2015/266, 2015/618). All patients included in the study provided written, informed consent. The whole-slide images were annotated by an expert pathologist classifying the tissue compartments/regions into four different classes: tumour, necrosis, immune cells and stroma. An example whole-slide image of a tumour sample and its corresponding labels can be seen in Figure 3.1. Tiles from each of the four classes are presented in Figure 3.2. Clinical data, including survival data and mutation status, were available for the cohort. BRAF V600E mutation status was available for a majority of the samples and NRAS mutation status was also examined for samples carrying BRAF WT.

Whole-slide images from the TCGA database [36] were downloaded. 319 samples of flash frozen tumours of mostly metastases were used. Information about the mutation status of NRAS and BRAF was included. The distribution of the samples were: 182 BRAF mutated, 106 NRAS mutated and 31 WT.

The Semmelweis historical cohort consists of autopsy samples from multiple metastatic MM tumour tissue from 19 patients (flash frozen tissue). The study was approved by the Semmelweis University Regional and Institutional Committee of Science and Research Ethics (IRB, SE TUKEB 114/ 2012). Patient consent to participate was waived by the Ethics Committee of the Semmelweis University by reason that metastatic samples were collected at the time of autopsy. Clinical data about NRAS and BRAF was available for most samples.

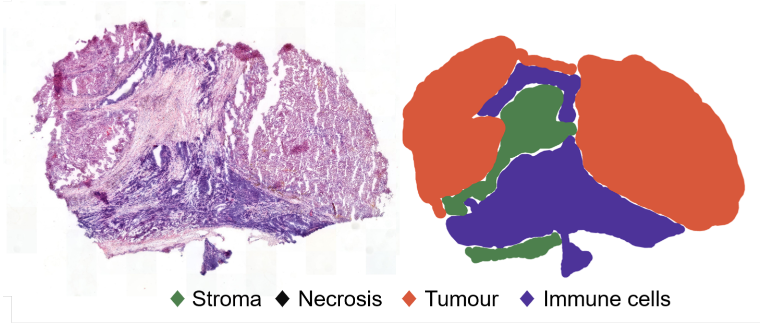


Figure 3.1: Example image from the Lund cohort with its corresponding mask. Whole-slide H&E stained tissue image to the left and the corresponding annotated image to the right.

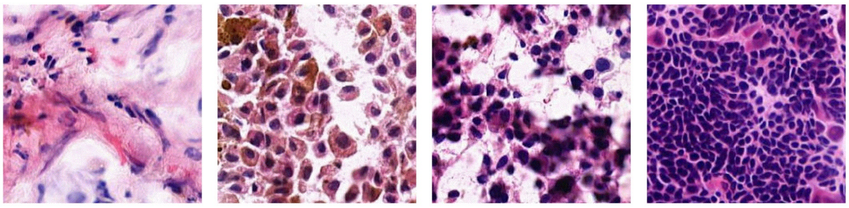


Figure 3.2: Tiles from the four tissue classes of an image in the Lund cohort. From left to right: stroma, necrosis, tumour, immune cells.

3.2 Data processing

3.2.1 Initial cropping

The images were first cropped in QuPath, which is an open source software for whole-slide image analysis [37]. QuPath was needed due to the size and format of the microscopy images. Images of size 10 000 x 10 000 pixels were saved and for the Lund cohort additional corresponding masks were saved. The images were down-sampled to half of the original resolution, resulting in 20x magnification. One pixel in a down-sampled image corresponds to 0.5 µm. The script was provided by the author of QuPath [38].

3.2.2 Tiling of the cropped images

From the cropped images, multiple tiles were exported. One tile had the size 244 x 244 pixels. Tiles with more than 59% white pixels were removed. A pixel was classified as white if the grey-level intensity was 237 to 255. The thresholds were empirically chosen to remove images that were hard to gain any information from.

3.3 Datasets

3.3.1 LundSeg dataset

For training and evaluation of the segmentation network, samples from the Lund cohort were used. Whole-slide images from 116 patients were used, resulting in 844 961 tiles. 60% of the tiles belonged to the tumour class, making it the majority class. To make the classes more balanced in the dataset, the tumour class was undersampled by only keeping every fourth tile. The number of tiles in each class before and after undersampling is presented in Table 3.1. The stride used when tiling the images was 122 pixels, half of the image size 244 pixels. Therefore, the tiles were overlapping and the undersampling of the tumour class will not cause a great loss of data. The undersampled dataset will be referred to as the LundSeg dataset.

Table 3.1: The class distribution of all tiles before and after undersampling of the tumour class. The undersampled set is called LundSeg dataset. More than 60% of the tiles produced from the whole-slide images were tumour. The classes are more balanced after undersampling the tumour class to $\frac{1}{4}$ of its former size.

	Full set		Undersampled set	
Class	No of tiles	Share	No of tiles	Share
1. Stroma	139 269	16.5%	139 269	31.6%
2. Necrosis	97 486	11.5%	97 486	22.1%
3. Tumour	539 340	63.8%	134 837	30.6%
4. Immune cells	68 866	8.2%	68 866	15.6%
Total	844 961		440 458	

The next step was to divide the dataset into three parts, training set (70%), validation set (15%) and test set (15%). All tiles from a single patient were added to a set together, to prevent the sets from becoming too similar. The sets and the number of samples are presented in Table 3.2.

Table 3.2: Division of the LundSeg dataset into three subsets: training, validation and test. The subsets do not contain tiles from the same patient. This table presents the number of tiles in each subsets and how many patient samples the tiles were produced from.

Set	Patient samples	Number of tiles
Training	85	309 900
Validation	15	64 745
Test	16	64 813

3.3.2 LundClass testset for NRAS/BRAF classifier

Samples with BRAF or NRAS mutation were extracted from the LundSeg dataset to form a new testset for a BRAF/NRAS classifier, which will be called LundClass dataset. There were 48 samples that were BRAF mutant and 29 NRAS mutant. The tiles that had a higher predicted probability than 90% of being tumour according to the segmentation network were included (Inception v3 trained on tissue segmentation with the LundSeg dataset, more details are presented in Method and Results). The tiles were zero padded to the size 299 x 299 pixels.

3.3.3 TCGABin dataset, NRAS versus WT

Images from 62 patients (31 NRAS mutated and 31 WT) were used in TCGABin dataset. The images were cropped and tiled according to Section 3.2.1 and 3.2.2.

3.3.4 TCGAClass dataset, NRAS versus BRAF

The images in TCGABin dataset with confirmed NRAS mutation were supplemented with 76 NRAS mutated samples (resulting in 107 NRAS samples) as well as 182 samples with BRAF mutation. Samples with both BRAF and NRAS mutations were not used in this dataset. The images were cropped and tiled, described in Section 3.2.1 and 3.2.2. The tiles were separated into the sets training ($\sim$ 70%), validation and test ($\sim$ 15% each) without patient overlap. The tiles were run through the segmentation network (Inception v3 trained on tissue segmentation with the LundSeg dataset, more details are presented in Method and Results). Only the tiles with 90% or higher probability of belonging to the tumour class were used. The sizes of the final three sets are presented in Table 3.3. Zero-padding changed the tile-size from 244 x 244 pixels to 299 x 299 pixels and the training set was oversampled to balance the classes and increase the amount of data, resulting in 500 000 training tiles from each class. Randomised rotation and mirroring was done to the oversampled tiles.

Table 3.3: The data in TCGAClass dataset was divided into the subsets training, validation and test without patient overlap. The training set was oversampled with augmentation.

	Training tiles		Validation tiles	Test tiles
Class	Originally	Oversampled		
BRAF	166 347	500 000	38 158	31 815
NRAS	105 031	500 000	21 401	20 678
Total	271 378	1 000 000	59 559	52 493

3.3.5 Semmelweis dataset

One whole-slide image was used from every patient, and the samples without NRAS or BRAF mutation were removed. Furthermore, some whole-slide images were removed due to artefacts in the images. The final dataset consisted of whole-slide images from 15 patients (12 BRAF-mutated and 3 NRAS-mutated). The images were cropped and tiled according to Section 3.2.1 and 3.2.2. The tiles that had over 90% probability of being tumour according to the segmentation network were zero-padded to the size 299 x 299 pixels. This resulted in 31519 tiles.

3.3.6 Summary of datasets

All datasets and their corresponding number of patients and tiles are presented in Table 3.4.

Table 3.4: All datasets used in this project with the number of patient samples and tiles. The datasets for the classification (TCGAClass, LundClass and Semmeweis) only contain tumour tiles.

Dataset	Number of patients	Number of tiles
LundSeg	116	440 458
TCGABin	62	87 052
TCGAClass	289	1 112 052
LundClass	77	238 822
Semmelweis	15	31 519

Chapter 4

Method

4.1 Overview

This chapter describes the methods of the project in detail. Firstly, the method of training and evaluation of the segmentation network is presented. This network was trained on the LundSeg dataset. Secondly, the attempts of using feature extraction to classify NRAS or WT using the TCGABin dataset are described. The features were extracted from different layers of the segmentation network and a subset of features were given as input to different binary classifiers. Thirdly, another deep learning network was trained to classify tumours based on the mutation status BRAF or NRAS. This method provided promising results and the whole process is visualised in Figure 4.1. The classification network was trained and evaluated with TCGAClass dataset and additional testing was done on the datasets LundClass and Semmelweis.

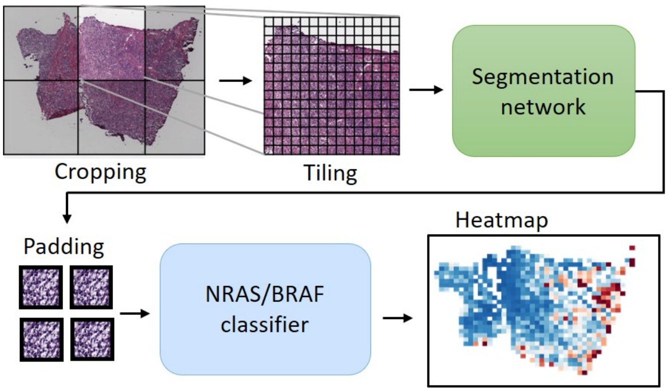


Figure 4.1: The workflow of prediction of NRAS/BRAF mutation status. Whole-slide images from TCGA were cropped and tiled. The tiles were segmented using the segmentation network and the tiles with over 90% predicted probability of being tumour were zero padded and used as input to the next deep learning network. The last network produced probabilities of the mutation statuses BRAF and NRAS.

4.2 Setup

The training and evaluation of the networks in this project have been coded using Python 3 in Jupyter Notebook [39], a web-based interactive computational environment. Tensorflow [40] and the Keras [41] library have been used. Other Python libraries used in this project are scikit-learn [42], Matplotlib [43] and OpenCV [44]. QuPath [37] has been used to look at and crop whole-slide images and annotations. The training was done using a NVIDIA GeForce RTX Super 2070 (8 GB) using the CUDA toolkit [45].

4.3 Segmentation

4.3.1 Architecture

The segmentation was done using the deep learning network Inception v3 [46], pre-trained with ImageNet data [47]. The ImageNet data consists of more than 1.2 million images of 1000 classes, such as balloon, car and volleyball. It has been shown that CNN:s pre-trained on ImageNet data can transfer well to other classification problems, including biomedical datasets [48]. The top layers of the model (where the classification into different classes happens) were replaced with global average pooling, a fully connected layer of 1024 nodes with ReLU activation functions and a final fully connected layer with four nodes and the softmax activation function. A visualisation of the Inception v3 network is presented in Figure 4.2, where the boxes in different colours represents slightly different inception units, shown in Figure 2.5. Dropout was used between the base network and the fully connected dense layer (orange layer in the figure), with the probability to drop a node set to 0.2. The architecture of the top layers was inspired by previous projects that use Inception v3 for similar tasks [49–51].

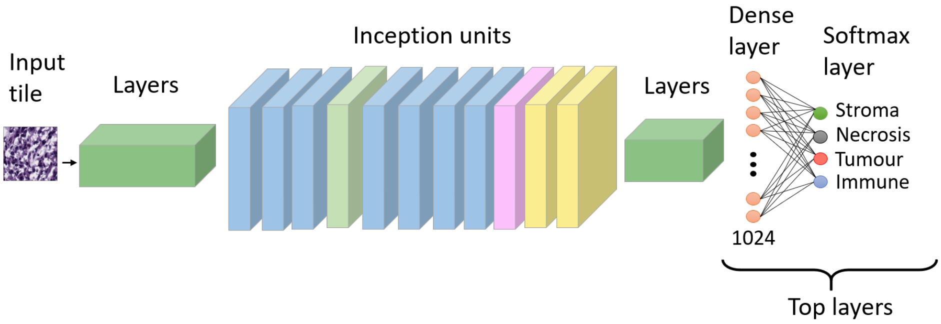


Figure 4.2: A visualisation of the segmentation network. Tiles are used as input to the network and the softmax output provides probabilities for the four classes. For simplification, the many layers and operations in the Inception network are visualised with boxes, where different colours represent slightly different internal architecture in the inception units.

4.3.2 Training

A training set was created using the Keras function ImageDataGenerator. All tiles were pre-processed using normalisation and by dividing all pixel values by 255. To compensate for the unbalanced classes, class weights were added to the loss function in the training. The weight of each class was calculated using Equation 4.1.

$$\text{weight} = \frac{\text{total number of samples}}{\text{number of samples in class}} \tag{4.1}$$

The training was divided into two phases, where the first phase was transfer learning. In the transfer learning phase, all the weights of the inception layers were locked. Hence, only the top layers consisting of 1024 nodes fully connected to 4 output nodes were trained, see Figure 4.2. The hyperparameters are presented in Table 4.1. The second phase of the training was fine tuning, where the last two inception units closest to the top layers were unlocked and made trainable (yellow boxes in Figure 4.2). The top layers were still trainable. The learning rate was decreased to avoid overfitting, see the hyperparameters in Table 4.1.

Table 4.1: The hyperparameters chosen for training of the segmentation network. The learning rate was lowered in the fine tuning phase to avoid overfitting.

	Phase 1: transfer learning	Phase 2: fine tuning
Epochs	35	30
Optimisation method	Adam	Adam
Learning rate	$5 * 10^{-3}$	$2 * 10^{-6}$
Batch size	100	100
β_1	0.9	0.9
β_2	0.999	0.999
ϵ	10^{-7}	10^{-7}

4.3.3 Evaluation of the results

The network performance was tested on the validation data and the unseen test data of the LundSeg dataset. The classwise precision, recall and F1 score were calculated and confusion matrices were used to visualise the classification performance. The tiles from some image samples from the test set were run through the segmentation network and segmentation masks were recreated from the predictions. The recreated segmentation masks were visually inspected, to get a view of where the incorrect classifications were in the image.

4.4 Feature extraction and classification of NRAS versus WT

Features were extracted from the segmentation network with the goal to predict the NRAS mutation status, in the TCGABin dataset. Features were extracted from different layers in the segmentation model, both from the dense layer of 1024 nodes and the last convolutional layer in the inception network (6912 features). The features were extracted before the activation function. To extract features, all tiles from each patient were segmented by the segmentation model and the 10 tiles with the highest accuracy for tumour were selected. The mean feature values from these 10 tiles were saved as features for the patient sample.

Different methods of feature selection and binary classifiers were tested. The tested classifiers were logistic regression, random forest classifier, PLS-DA and SVM (with linear and radial basis function kernels). Cross validation was used to test the generalisation performance despite the small amount of data (62 samples). The 95% confidence interval was investigated to see whether it covered 50% which is a random binary classifier.

4.5 Classification of BRAF versus NRAS mutation

A new deep learning network was trained to do binary classification of NRAS/BRAF mutation status and the model was based on a modified Inception v3 architecture [46]. The dataset used was TCGAClass dataset. The code from Coudray et al. [5] was modified to allow binary output classification (`DeepPath,https://github.com/ncoudray/DeepPATH`). The network was pre-trained on ImageNet [47] data. The image tiles were converted into TFRecord format, which is a simple format for storing a sequence of binary records. The default hyperparameters were used in the training and they are presented in Table 4.2. Other settings were tested as well without satisfactory results. The batch size was adapted to fit the GPU capacity.

Table 4.2: The hyperparameters chosen for training of the classification network.

Hyperparameter	Setting
Iterations	500 000
Optimisation method	RMSProp
Learning rate	0.1
Weight decay	0.9
Momentum	0.9
ϵ	1.0
Batch size	36

Checkpoints were saved every 10 000th iteration of training and the network was evaluated on every 20 000th iteration using the validation data. At the interval with highest AUC-values on the validation data (70 000 - 150 000 iterations), the model was evaluated more frequently (every 10 000th iteration). The best model was chosen as the checkpoint with highest AUC on the validation data. The network was further tested on the test data, and the independent datasets LundClass and Semmelweis. ROC-curves and AUC were presented both tilewise and patientwise for the different datasets. The patientwise prediction was done by examining the average of the predicted probabilities of all tiles.

Heatmaps of the predicted probabilities were generated. A tile was given a colour based on the predicted probability and the darker the colour, the higher probability for a certain class. Tiles that have high probability of being NRAS are dark red and tiles that are light red are still predicted as NRAS, but with a lower probability. Predicted probabilities of BRAF behave the same way but in blue.

Chapter 5

Results

5.1 Segmentation of tissue types

5.1.1 Gold standard annotation

The images in the LundSeg dataset were annotated in QuPath [37]. The annotations were sometimes overlapping and some parts of the tissue were therefore belonging to two classes. In Figure 5.1 it is possible to see that the annotation is inexact. When the annotation of one class were put on top of another class, the exported mask was only showing one of the annotations. In some cases, this caused some areas to get the wrong class. An example of this is presented in Figure 5.2, where one blue immune cell area in the left image is beneath the red tumour annotation, which lead to the immune cell area getting labelled as tumour. The specialist manually annotating the images chose to classify the tissue by the dominant class and when there were small parts standing out from their surroundings this was not taken into account.

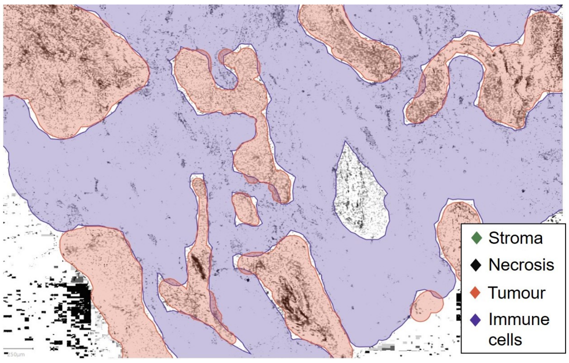


Figure 5.1: The image shows a part of a Lund cohort tissue sample with inexact annotation of tumour and immune cells. The annotated areas sometimes overlap in the Lund cohort. In this example, tumour and immune cells cover the same pixels in multiple locations. The white part in the middle of the image does not belong to any class. To visualise the annotation, the colour of the image is distorted.

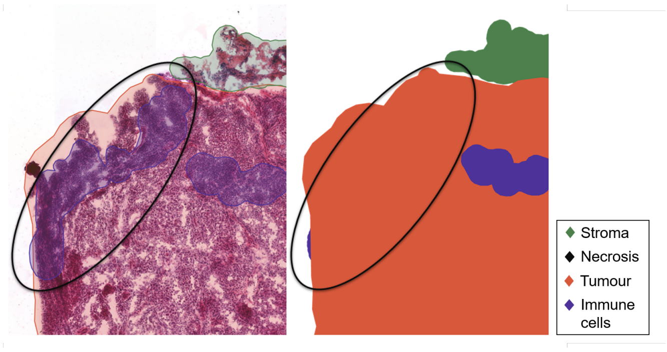


Figure 5.2: To the left is an annotated tissue sample from the Lund cohort with the annotations as transparent colours. The exported mask is shown to the right. The mask should have two blue areas of immune cells but since the leftmost immune cell area is placed beneath the red tumour annotation, the area is not present in the exported mask. The gold standard of these tiles will be incorrect.

5.1.2 Average network performance

The segmentation network was trained for 65 epochs and the average accuracy is presented in Figure 5.3 and the plotted categorical cross entropy loss is presented in Figure 5.4. In epoch 35 the training was changed from transfer learning (training of top layers only) to fine tuning. This means that the last two inception units were trained as well. After the 65 training epochs, the network was tested on the validation and test data. The classwise performance is presented in Table 5.1 and the confusion matrices are shown in Figure 5.5.

Table 5.1: The performance of the segmentation network on the Lund-Seg validation and test data. On the validation data, the performance was higher on the classes stroma and immune cells. The performance on the test data was better than the validation performance with F1 score 0.84 or higher on all classes.

	Validation data			**Test data**		
Class	Precision	Recall	F1	Precision	Recall	F1
1. Stroma	0.86	0.88	0.87	0.87	0.86	0.86
2. Necrosis	0.85	0.67	0.75	0.88	0.86	0.87
3. Tumour	0.73	0.81	0.77	0.83	0.85	0.84
4. Immune	0.89	0.91	0.90	0.90	0.91	0.90

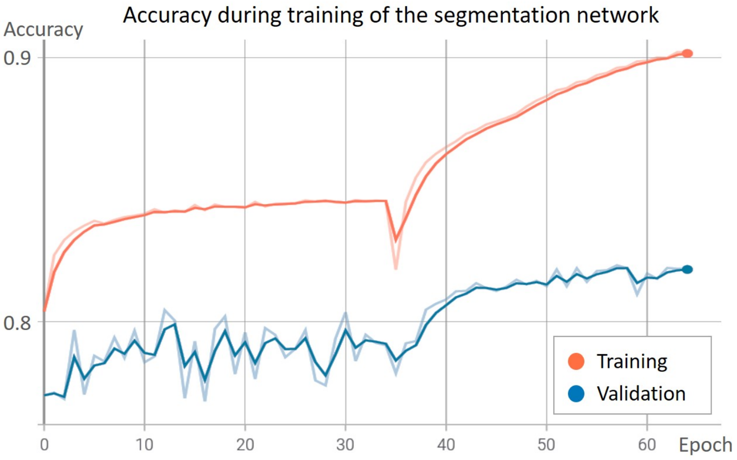


Figure 5.3: The accuracy of the LundSeg training and validation data during the training of the segmentation network. The sharp decline at epoch 35 is where the training went from transfer learning to fine tuning.

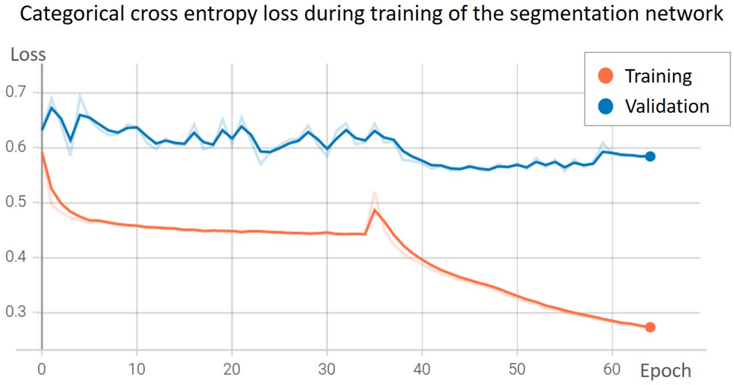


Figure 5.4: The categorical cross entropy loss on the LundSeg training and validation data during the training of the segmentation network. The peak at epoch 35 is where the training went from transfer learning to fine tuning.

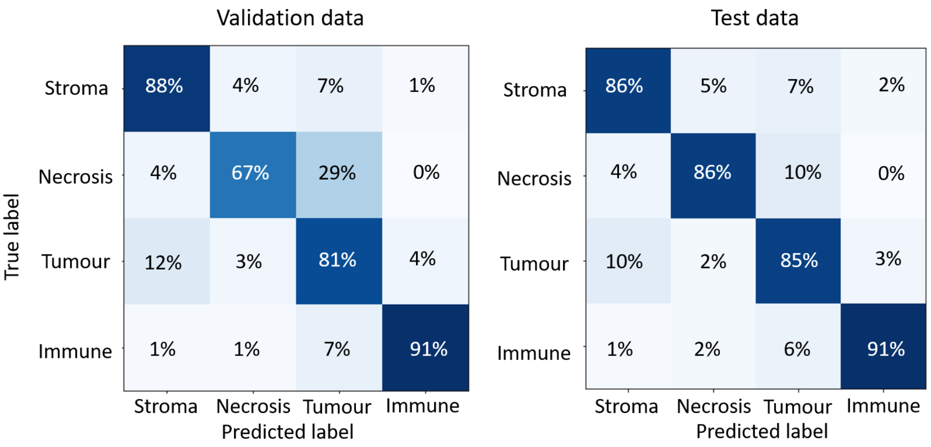


Figure 5.5: Normalised tilewise confusion matrices of the segmentation network. The LundSeg validation data is presented to the left and the test data to the right.

5.1.3 Segmentation of images from the test set

Images of the test set were segmented using the model, see Figures 5.6, 5.7 and 5.9. In Figure 5.6 there are folding artefacts, which are the darker parts in the whole-slide image to the left, however the prediction does not seem to be affected by the artefacts. Three of the classes are present: tumour, immune cells and stroma. The white parts are tiles that were removed prior to segmentation, since they contained less than 80% of one class or because they were too bright.

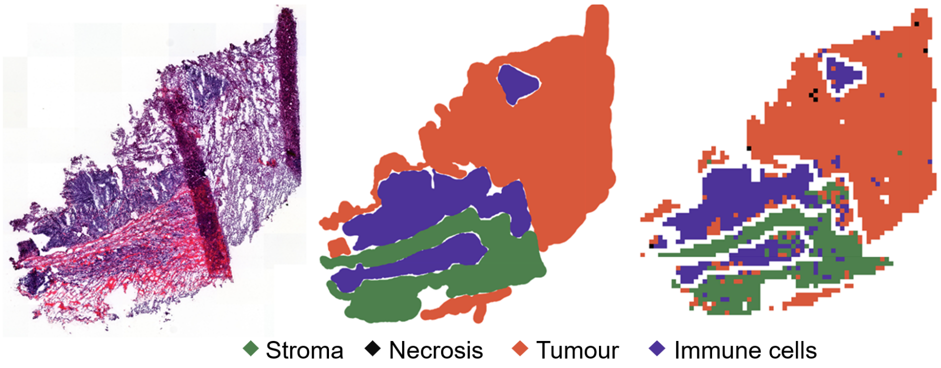


Figure 5.6: The segmentation results on an image from sample MM1073 in the LundSeg test set. From left to right: whole-slide image, annotation and prediction. Folding artefacts are visible in the whole-slide image to the left (two darker lines). The predicted tiles are not affected by the experimental artefacts.

Figure 5.7 contains tumour and necrosis. The tumour parts are well classified. Some necrosis tiles are classified as tumour. A magnification of a part of the tissue that was differently classified is presented in Figure 5.8. It is visible that there is some internal variation in the tissue structure, which was not taken into account so thoroughly during the manual annotation where the dominant class was selected for the whole area.

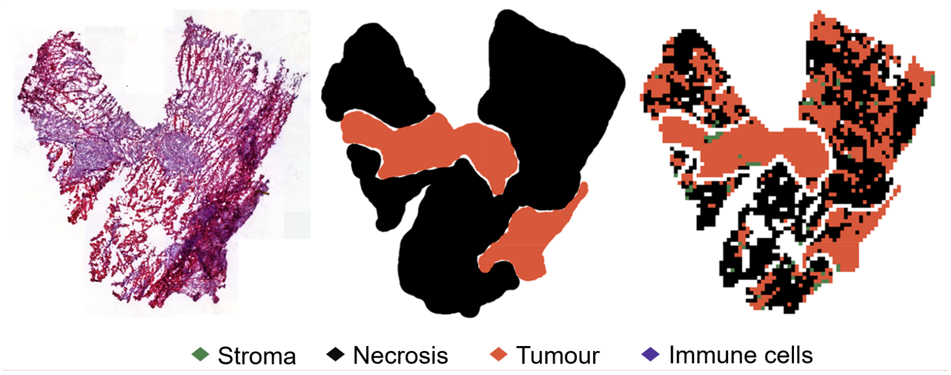


Figure 5.7: The segmentation results of an image from sample MM1265 in the LundSeg test set. From left to right: whole-slide image, annotation and prediction. The tumour tiles were well classified. Some necrosis tiles were classified as tumour.

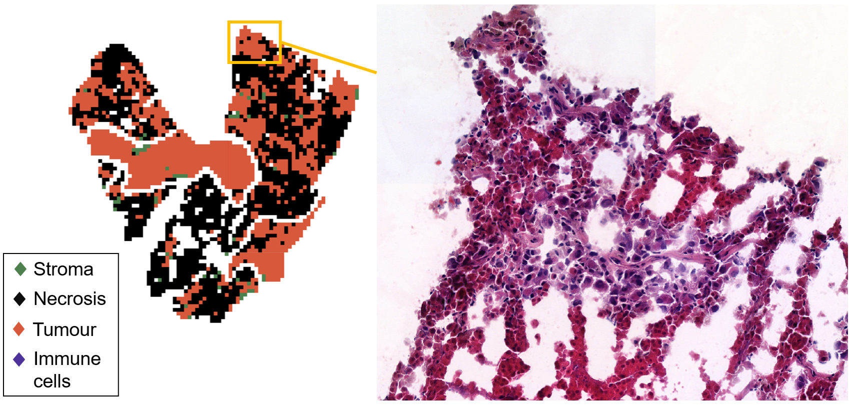


Figure 5.8: Miss-classified parts that stand out from their surrounding. Magnified parts of the whole-slide image from sample MM1265 in the LundSeg test set. These parts were incorrectly classified with respect to the annotation, see Figure 5.7. However it is visible that there are variations in the tissue structures.

Figure 5.9 contains immune cells, tumour and stroma. Most of the tiles were correctly classified. Two parts of the whole-slide that were differently classified are magnified in Figure 5.10. To the left in the image, it is a part that has been classified as necrosis, although the gold standard is tumour. The magnified version to the left shows that the tissue is darker in this part. To the right there is a section classified as tumour in the blue immune cell area. The magnified image shows that the part being classified as tumour instead of immune cells is more pink than the surrounding tissue.

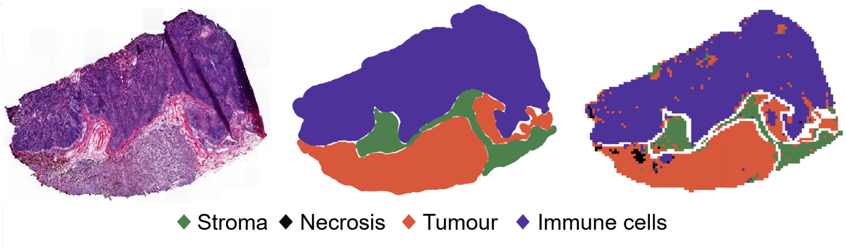


Figure 5.9: The segmentation results of an image from sample MM710 in the LundSeg test set. From left to right: whole-slide image, annotation and prediction.

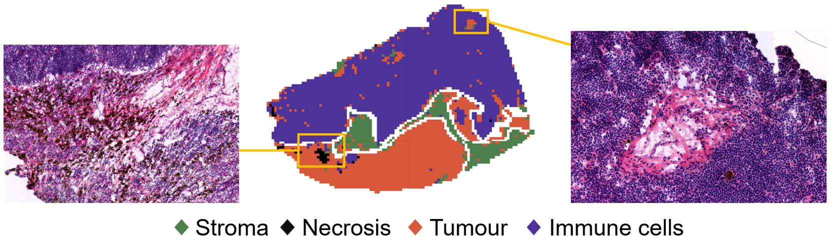


Figure 5.10: Miss-classified parts that stand out from their surrounding. Magnified parts of the whole-slide image from sample MM710 in the LundSeg test set. These parts were miss-classified with respect to the annotation, see Figure 5.9, however it is visible that these parts stand out from their surroundings.

5.1.4 Segmentation of images from TCGA

Images from TCGA were segmented using the network, see Figures 5.11 and 5.13. These images were not annotated in the database, and hence no true annotation is available. By a visual comparison of both the authors and a biologist at the institution, the results look reasonable. In Figure 5.11 it is a clear border between a part that looks more pink and a part with more purple. These parts are predicted as stroma respectively tumour. In the stroma part, there are some tiles that are predicted as tumour. In Figure 5.12 it is possible to see two magnified versions of the tissue classified as tumour. The tiles that have been predicted as tumour are more purple and they stand out from the surrounding tissue.

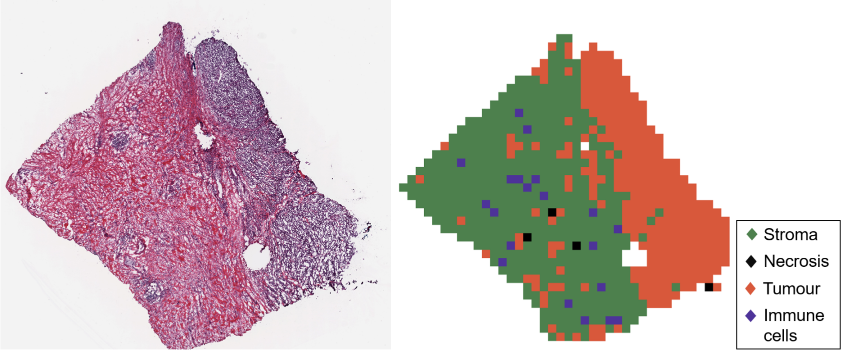


Figure 5.11: Segmentation results of image from TCGA. The whole-slide image of TCGA-EE-A29C to the left and the predicted classes to the right. There is no gold standard available from TCGA. It is possible to see a border between a more pink area to the left and more purple to the right, and these areas have been predicted as different classes.

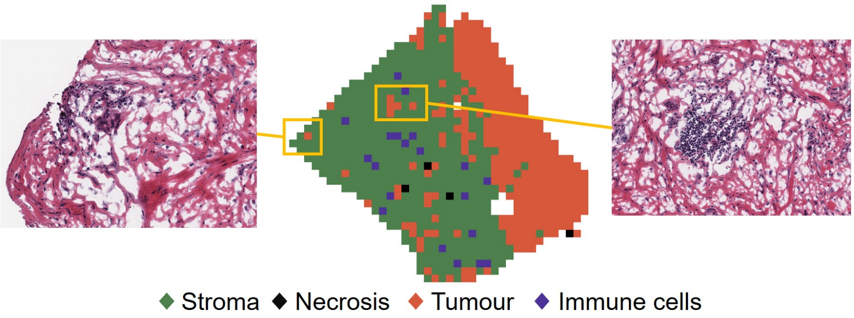


Figure 5.12: Magnified parts of the segmented TCGA-EE-A29C image. It is possible to see differences between the tiles that were predicted as tumour and the surroundings of stroma.

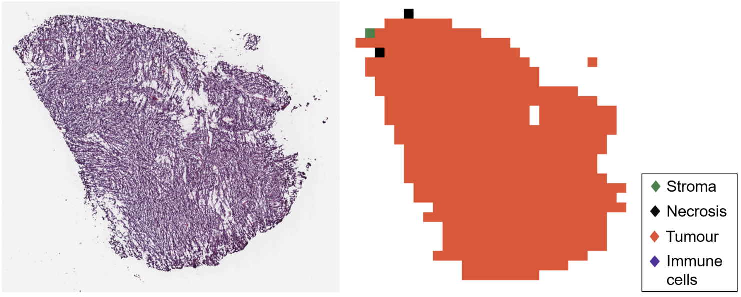


Figure 5.13: Segmentation results of image from TCGA. The whole-slide image of TCGA-EE-A3AE to the left and the predicted classes to the right. There is no gold standard available from TCGA.

5.2 Feature extraction and classification of NRAS versus WT

Features were extracted from two different layers of the segmentation model. The different methods of feature extraction of a relevant subset of features gave different results and the features did not seem to generalise. The tested classifiers logistic regression, random forest, PLS-DA and SVM gave poor results for all tested subsets of features. It was possible to get acceptable results on the training data but generalisation performance with cross-validation did not get considerably better than chance (50%). Due to this, no results of the tested classifiers are presented. The number of features from the different layers in the segmentation model are presented in Table 5.2 together with the number of samples.

Table 5.2: The layers used for feature extraction, the number of features and samples. For both tested layers, the number of features are substantially larger than the number of samples.

Layer in model	Number of features	Number of samples
Last conv. layer	6912	62
Dense layer	1024	62

5.3 Classification of NRAS versus BRAF

The classification network was trained on the TCGAClass dataset for 500 000 iterations which corresponds to about 18 epochs. Checkpoints of the weights were saved and the AUC-values for the training and validation set were calculated for each evaluation point, see Figure 5.14. The checkpoint at 70 000 iterations was chosen as the final model since it had the highest AUC-value on the validation data.

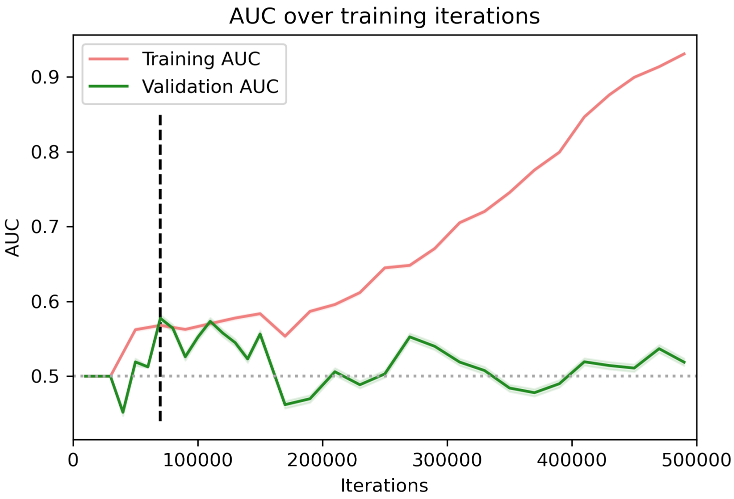


Figure 5.14: The AUC of the TCGAClass training and validation data with 95% confidence interval during the training of the classification network. The highest peak for the validation data is at 70 000 iterations (marked with a dashed line) and this model is chosen as the final model. The confidence intervals for the training data are very narrow and hence not visible in the figure.

The model was evaluated on the validation and test data, as well as the two independent test sets LundClass and Semmelweis dataset. The tilewise and the patientwise ROC-curves are shown in Figure 5.15 and 5.16 for the different datasets. The model performs better on the test data than on the validation data. The tilewise AUC for the LundClass dataset is quite low (0.53) but the patientwise prediction has a higher AUC of 0.59. The AUC-values of the Semmelweis dataset is close to 0.6 for both the tilewise and patientwise predictions. Although, the Semmelweis dataset only has 3 NRAS mutated samples and the patientwise AUC can be misleading.

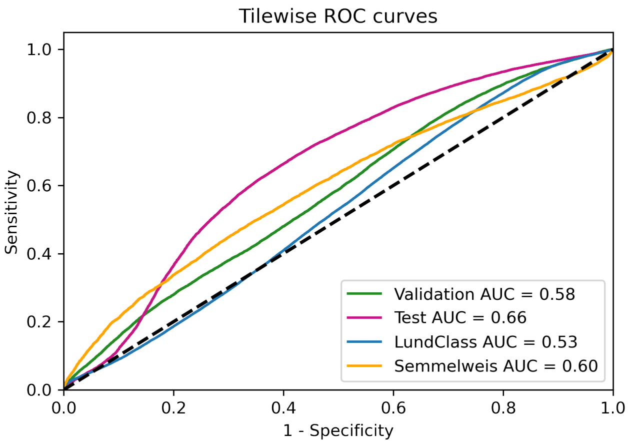


Figure 5.15: ROC-curves and AUC-values for the four datasets on tilewise classification of BRAF versus NRAS mutation status. The model shows good potential at predicting BRAF vs NRAS with AUC over 0.6 on two datasets. The model performance on the LundClass dataset is close to 0.5.

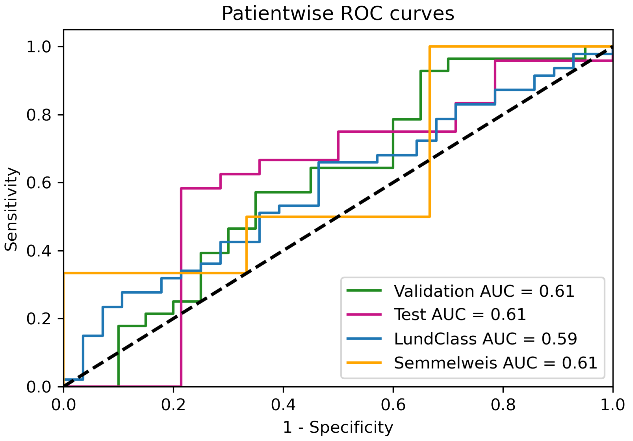


Figure 5.16: ROC-curves and AUC-values for the four datasets on patientwise classification of BRAF versus NRAS mutation status. The AUC-values for the patientwise predictions are close to 0.6 for all four datasets.

The results of the predictions are visualised with heatmaps in Figure 5.17 and 5.18. Every tile is given a colour based on the predicted probability. Tiles predicted as BRAF are blue (the darker blue the higher probability) and tiles predicted as NRAS are red (the darker red the higher probability of NRAS). Overall, when a tumour is BRAF mutated more tiles are correctly classified as BRAF. In the BRAF examples from the test set and the Semmelweis dataset, some darker areas are predicted as NRAS (artefacts). For the NRAS mutated tumours, there are more tiles with low probabilities (light blue and light red). There are also tiles predicted as BRAF in all NRAS mutated examples. In the whole-slide images in the LundClass dataset in Figure 5.18, some parts of the tissue is predicted as stroma (marked yellow in the image) and therefore these tiles are not fed into the classification network.

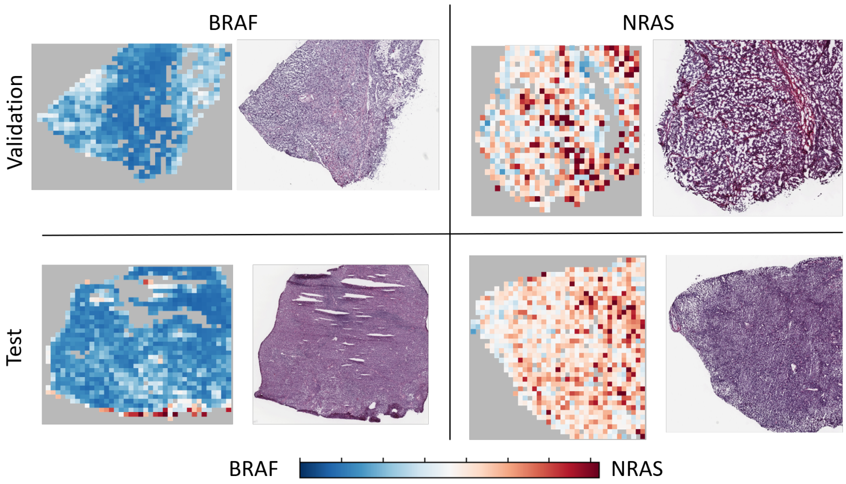


Figure 5.17: Slides and probability heatmaps for BRAF and NRAS positive samples from the validation and test sets. The heatmaps of the BRAF samples are overall predicted as BRAF while the NRAS samples are more ambiguous.

A slide from the validation set with an overlayered heatmap is presented in Figure 5.19. The gold standard of the tumour is BRAF, although a part of the whole-slide is predicted as NRAS (the red area in the figure). Two magnified areas are shown to the right in the image, and it is evident that the two areas differ in both colour and morphology.

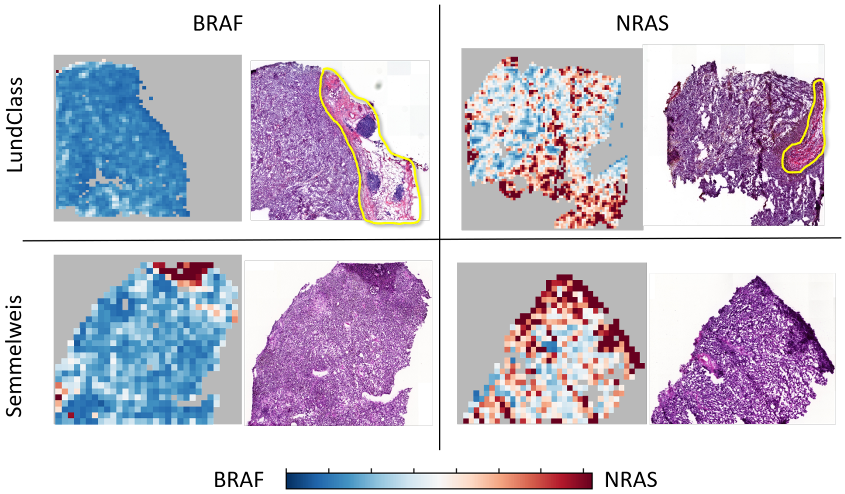


Figure 5.18: Slides and probability heatmaps for BRAF and NRAS positive samples from the LundClass and Semmelweis datasets. The heatmaps of the BRAF samples are in general predicted as BRAF except for a dark area (artefact) in the Semmelweis sample. Like for the validation and test sets, the NRAS samples are more ambiguous. In both LundClass slides, areas with stroma (marked yellow) have been removed by the segmentation network.

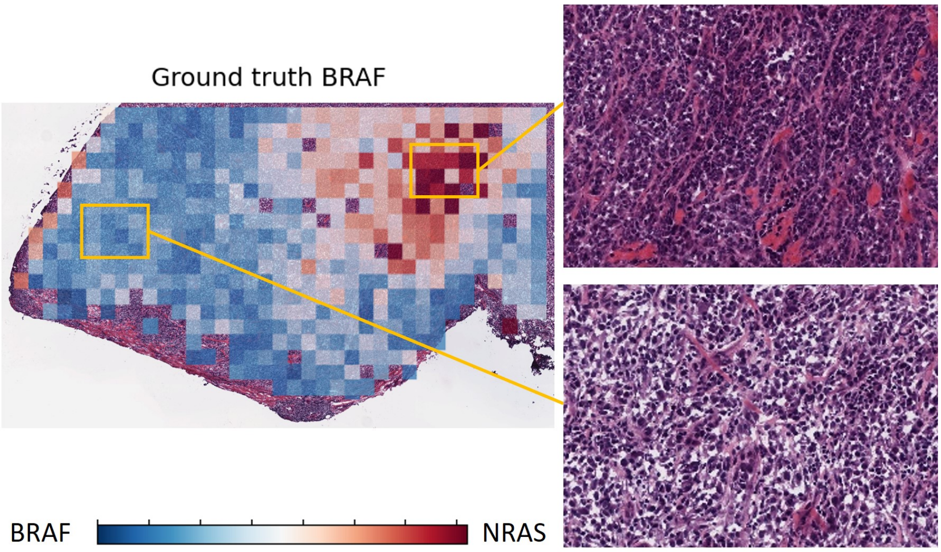


Figure 5.19: A slide from the validation set with probability heatmap over-layered and magnified areas. The magnified parts show differences in structure and colour between the NRAS and BRAF classified tiles. Note that the area marked in red is incorrectly classified.

Chapter 6
Discussion

6.1 Segmentation model

6.1.1 Training and overfitting

The Inception v3 network has a very deep architecture with many trainable weights. The more trainable weights, the larger amount of data is needed to avoid overfitting. The choice of dividing the training into phase 1 and 2 (transfer learning and fine tuning) was made to control the amount of trainable weights. During the model selection, a span of different learning rates was tested and it was clear that the model was easily overfitted for larger learning rates during phase 2 (fine tuning). With the chosen settings, the segmentation network does not seem overfitted, since the validation performance never starts to decrease, see Figure 5.3.

6.1.2 Data and performance

The performance of a neural network is greatly dependent upon the training data. Some parts of the tissue were assigned two labels or one incorrect label (at the border between two classes). This will lead to several tiles having the incorrect class which will make the training process harder. Additionally, the evaluation results can be misleading. In Figure 5.10, there are some parts that are miss-classified according to the annotation. However, these parts do stand out from their surrounding tissue and it is possible that the segmentation network is more precise than the annotations. The same behaviour can be seen in Figure 5.8, where it is visible that some parts of the gold standard necrosis can be differentiated from the rest, and they are predicted as tumour. To summarise, it is hard to tell if the measured performance of the network is correct or if it is too pessimistic due to the imprecise annotation. The accuracy is better for the test set than for the validation set, which suggests that the validation set is harder to segment than the test set. Another possible explanation is that the test set is more similar to the training set.

It is possible to be more selective in the process of creating the dataset. One way of doing this is to remove tiles that are close to the border of two labels, or close to the border of the tissue sample. Another way of being selective is to remove whole-slide images that are considered hard to classify. We chose to keep as many tiles as possible to make the dataset versatile and comprehensive. The model should therefore be more general, but this may also lead to lower accuracy scores.

Whole-slide images from different cohorts may differ due to various equipment and imaging methods. It is therefore important to test a model on independent cohorts from other hospitals to evaluate the generalisation performance. The segmentation model in this project was trained and tested on data from the Lund cohort and an additional test was made on images from TCGA. The TCGA database did not include tissue type annotations and it was not possible to numerically evaluate the performance. However a visual examination could be done by both the authors and a biologist at the institution and the performance seems adequate. Before a segmentation model like this can be used clinically, it is crucial to confirm the performance on more independent datasets with annotations made by several experts. Another way to confirm the segmentation result could be to compare the results with another well-known segmentation system.

To get trustworthy results, it is important that there is no overlap between the training, validation and test sets. Otherwise, the validation or test performance could be too optimistic since the data is similar to the training data. We chose to never have tiles from the same patient in different datasets to avoid this effect.

6.2 Attempt of classification with image features

As presented in the Results, Section 5.2, no satisfactory outcome was obtained for the classifier with image features. The features were extracted from the segmentation model at two different layers, a dense layer and a convolutional layer. The dense layer in the top layers is trained to be specialised at separating the tissue types. Since only tumour tiles were used when extracting features, it is reasonable that the features are too similar to differentiate NRAS versus WT tumour tiles. The convolutional layer is further from the top than the dense layer and it was only trained on tissue segmentation during phase 2. It was therefore thought to be less specialised on the tissue types. 6912 features were extracted from images from 62 patient samples. This means that one feature could describe patterns connected to only one sample. The validation performance did never reach adequate levels which implies that the classifiers were only able to find patterns in the training data but the classifier were not able to generalise.

Even though the results for the classification with image features were unsuccessful in this project, it might be possible to obtain better results. Future work could include a larger dataset with more samples since a larger amount of samples would make it easier to find general features. Another possible approach could be to extract features from another layer in the network, or from a network that is trained for another task. The classification network described in this project is trained to differentiate between NRAS and BRAF and features from this network might be better to feed to a binary classifer.

6.3 Classification network

6.3.1 Training and evaluation

The classification network was trained for 500 000 iterations and the final model was chosen at iteration 70 000 since it had the best performance on the validation data. The AUC is plotted against iterations in Figure 5.14. The model gets overfitted to the training data since the AUC-values of the training data increases while the AUC-values of the validation data decreases. The model is not overfitted at iteration 70 000. The model performance never converges to a specific AUC-value for the validation data. The spiky behaviour of the AUC-values for the validation data can indicate that the learning rate is too high. However, lower learning rates were tested as well but the model seemed to get stuck in local minima. The learning rate of 0.1 was tested because it gave good results for similar tasks [5] [6].

The classification model is general since the patientwise AUC is close to 0.6 for all datasets (see Figure 5.16). This is promising since the model performs well on data from other sources than the source of the training data. To improve the model performance, the model needs to be trained on a larger dataset, and more specifically with images from different sources. Further testing would also be needed. Another possible method of improvement could be to train a three-way classifier instead of a binary classifier, which is used in this project. The three-way classifier could include a WT class. This could improve the model performance since tiles with low probability of being BRAF not necessarily have to be predicted as NRAS.

The heatmaps of the predicted probabilities (Figure 5.17 and 5.18) show that the BRAF mutated example samples are uniformly predicted as BRAF. There are only artefacts, small darker parts on the edges of the tissue samples that have been predicted as NRAS. For the NRAS mutated samples, some tiles are predicted as BRAF and some have very low probabilities. The training data consists of more BRAF examples which may explain why the network is better at predicting BRAF correctly.

6.3.2 Alternative approaches

The segmentation model is used to find the tiles that have over 90% predicted probability of being tumour. Coudray et al. [5] emphasises the importance of choosing a good region of interest to feed to the classification network both during training and testing. Coudray et al. used a manually found region of interest and Kim et al. [6] implement a segmentation network to find a region of interest. For subsequent improvement of our workflow, the tiles fed into the classification network could be chosen more carefully, e.g. by making the segmentation network find an area with connected tumour tiles.

Each patient sample generates multiple tiles which are fed into the classification network. While the tilewise performance is straightforward to evaluate, the patientwise performance can be investigated with various approaches. We chose to use all tiles belonging to a patient sample and use the mean of the probabilities as the total probability for that sample. Tsou et al. [8] used more restrictive classification demands, only giving the patient sample a class if 80% or more of the tiles were predicted in one of the classes. Additionally, Tsou et al. classified tiles with low probabilities in a third class prior to patient classification. It might have been possible for us to induce a higher patientwise sensitivity if tiles with low probability of both classes were classified as uncertain.

6.3.3 Clinical aspects

To get a fully working classification system, a probability threshold must be defined. Since we only evaluated our classification results with ROC-curve and AUC, we did not need to find the optimal threshold for our classification. The colour scale of the heatmaps was chosen as white with smooth transitions on the border between the classes, so no definite threshold was defined there either. One of the biggest challenges when defining a threshold will probably be to make it fit other datasets than the one it was chosen from. If uncertain tiles were removed, a general threshold would likely be easier to find. Likewise if a three-way classifier was used, the classification would seemingly be less sensitive to a chosen threshold.

When using a classification system clinically, it is important to adapt the settings based on how the results will be used. If all mutant positives are investigated further, it would be preferable to have some false positives rather than false negatives, i.e. increase the sensitivity of the system. To make such an adjustment of the system, it would be necessary to predict mutation versus WT for every mutation.

A binary classifier of NRAS versus BRAF mutation status would not be convenient in clinical use. First of all, not all MM tumours carry one of the mutations and secondly, a low probability of BRAF mutation should not be equivalent to a high probability of NRAS mutation. Since we did not include WT tumours in any of the datasets used in the classifier, the binary classification was a simple and straightforward way to see if it was possible to train a network to discriminate between the mutations.

The mutation status of MM tumours are currently being examined with DNA-analysis which requires special laboratory equipment. The classification model, on the other hand, only requires a whole-slide image and a computer to run the classification on. A deep learning model could probably increase the availability of mutation status analysis around the world. However, this would require a more stable model with higher AUC-values.

Chapter 7
Conclusion

A system for classification of NRAS versus BRAF mutation status has been developed. Tiles from whole-slide images were first segmented into the tissue types stroma, necrosis, tumour and immune cells. The segmentation network had an F1-score of 0.84 for the tumour class on test data. The tiles with over 90% predicted probability of being tumour were passed to the classification network, where a binary prediction between NRAS and BRAF were made. The classification network had the tilewise AUC-values of 0.66 on the test set and 0.53 respectively 0.60 on independent datasets. The patientwise predictions had AUC-values around 0.60 for all datasets.

Features extracted from the segmentation network could not be used to separate NRAS and WT tumours with binary classifiers. The attempt was made on data from 62 patient samples and the classifiers tested were logistic regression, SVM, PLS-DA and random forest.

Deep learning models, more specifically Inception v3, have potential in being used in mutation status analysis of MM clinically. The promising results show that it is possible to predict mutation status from solely whole-slide MM tumour images. We believe that deep learning models can provide a cheaper and faster alternative to DNA-analysis for the detection of some cancer mutations in the future.

Acknowledgements

We would like to extend our gratitude to our supervisors Ida Arvidsson and Jonatan Eriksson for support during this project. The weekly support with everything from coding and report writing to motivation and encouragement has been very valuable. We would also like to thank Melinda Rezeli and her team for help with the data and helpful discussions about malignant melanoma.

The results shown in this report are in part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

References

[1] Heistein, Jonathan B. and Acharya, Utkarsh. Malignant melanoma. `https://www.ncbi.nlm.nih.gov/books/NBK470409/`. [Online: accessed 2021-02-15].

[2] Markus V. Heppt, Timo Siepmann, Jutta Engel, Gabriele Schubert-Fritschle, Renate Eckel, Laura Mirlach, Thomas Kirchner, Andreas Jung, Anja Gesierich, Thomas Ruzicka, Michael J. Flaig, and Carola Berking. Prognostic significance of BRAF and NRAS mutations in melanoma: a German study from routine care. *BMC Cancer*, 17, August 2017.

[3] Metin N Gurcan, Laura Boucheron, Ali Can, Anant Madabhushi, Nasir Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009 2009.

[4] Kaustav Bera, Kurt A. Schalper, David L. Rimm, Vamsidhar Velcheti, and Anant Madabhushi. Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology. *Nature Reviews Clinical Oncology*, 16(11):703–715, 2019.

[5] Nicolas Coudray, Paolo Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature Medicine*, 24, 10 2018.

[6] Randie H. Kim, Sofia Nomikou, Zarmeena Dawood, George Jour, Douglas Donnelly, Una Moran, Jeffrey S. Weber, Narges Razavian, Matija Snuderl, Richard Shapiro, Russell S. Berman, Nicolas Coudray, Iman Osman, and Aristotelis Tsirigos. A deep learning approach for rapid mutational screening in melanoma. *bioRxiv*, 2019.

[7] James M Dolezal, Anna Trzcinska, Chih-Yi Liao, Sara Kochanny, Elizabeth Blair, Nishant Agrawal, Xavier M Keutgen, Peter Angelos, Nicole A Cipriani, and Alexander T Pearson. Deep learning prediction of braf-ras gene expression signature identifies noninvasive follicular thyroid neoplasms with papillary-like nuclear features. *Modern Pathology*, pages 1–13, 2020.

[8] Peiling Tsou and Chang-Jiun Wu. Mapping driver mutations to histopathological subtypes in papillary thyroid carcinoma: applying a deep convolutional neural network. *Journal of clinical medicine*, 8(10):1675, 2019.

[9] M. van Zon, N. Stathonikos, W. A. M. Blokx, S. Komina, S. L. N. Maas, J. P. W. Pluim, P. J. van Diest, and M. Veta. Segmentation and classification of melanoma and nevus in whole slide images. pages 263–266, 2020.

[10] Heather D Couture, Lindsay A Williams, Joseph Geradts, Sarah J Nyante, Ebonee N Butler, JS Marron, Charles M Perou, Melissa A Troester, and Marc Niethammer. Image analysis with deep learning to predict breast cancer grade, er status, histologic subtype, and intrinsic subtype. *NPJ breast cancer*, 4(1):1–8, 2018.

[11] AIM at Melanoma Foundation. Stages of melanoma. `https://www.aimatmelanoma.org/stages-of-melanoma/`. [Online; accessed 2021-03-26].

[12] Melanoma Research Alliance. Melanoma Survival Rates. `https://www.curemelanoma.org/about-melanoma/melanoma-staging/melanoma-survival-rates/`. [Online; accessed 2021-03-26].

[13] Gisela Helenius. Analys av mutationer i braf. *Universitetssjukhuset Örebro*, 2013. `https://usorebro.se/sv/Behandlingar/Diagnostik/Laboratorieanalyser/Tumorgenetik/Analys-av-mutationer-i-BRAF/` [Online; accessed 2021-04-27].

[14] Marcus C Ravnan and Mazen S Matalka. Vemurafenib in patients with braf v600e mutation–positive advanced melanoma. *Clinical therapeutics*, 34(7):1474–1486, 2012.

[15] Liang Cheng, Antonio Lopez-Beltran, Francesco Massari, Gregory T MacLennan, and Rodolfo Montironi. Molecular testing for braf mutations to inform melanoma treatment decisions: a move toward precision medicine. *Modern Pathology*, 31(1):24–38, 2018.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[21] T. Jayalakshmi and Santhakumaran A. Statistical normalization and back propagation for classification. *International Journal Computer Theory Engineering (IJCTE)*, 3:89–93, 01 2011.

[22] Connor Shorten and T. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019.

[23] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017.

[24] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, March 2019.

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[26] Adrian Rosebrock. Imagenet: Vggnet, resnet, inception, and xception with keras, 2017.

[27] Ajay Kulkarni, Deri Chong, and Feras A. Batarseh. 5 - foundations of data imbalance and solutions for a data democracy. pages 83–106, 2020.

[28] Cyril Goutte and Éric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *ECIR*, 2005.

[29] Scott E Umbaugh. *Computer imaging: digital image analysis and processing.* CRC press, 2005.

[30] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression.* Springer, 2002.

[31] Yiu, Tony. Understanding Random Forest. `https://towardsdatascience.com/understanding-random-forest-58381e0602d2`. [Online; accessed 2021-04-02].

[32] Loong Chuen Lee, Choong-Yeun Liong, and Abdul Aziz Jemain. Partial least squares-discriminant analysis (PLS-DA) for classification of high-dimensional (HD) data: a review of contemporary practice strategies and knowledge gaps. *Analyst*, 143(15):3526–3539, July 2018.

[33] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[34] 1.4. Support Vector Machines — scikit-learn 0.24.1 documentation. `https://scikit-learn.org/stable/modules/svm.html`. [Online; accessed 2021-04-02].

[35] Giuseppe Lippolis. *Image analysis of prostate cancer tissue biomarkers.* PhD thesis, Urological cancer, Malmö, 2015.

[36] Robert L Grossman, Allison P Heath, Vincent Ferretti, Harold E Varmus, Douglas R Lowy, Warren A Kibbe, and Louis M Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112, 2016.

[37] Pete Bankhead et al. Qupath: Open source software for digital pathology image analysis., 2017.

[38] Bankhead, Pete. Qupath: Exporting annotations as labelled images. `https://gist.github.com/petebankhead/d2fd20f05b30b18380f0cd2a85a4a6cf#file-qupath-exporting-annotations-as-labelled-images-groovy`. [Online; accessed 2021-01-29].

[39] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[40] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[41] François Chollet et al. Keras. `https://keras.io`, 2015.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[44] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[45] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[47] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[48] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[49] Jonatan Nyström. Automated histopathological evaluation of tumor images using cnns, 2020. Student Paper.

[50] Tejan Irla. Image classification. `https://github.com/tejanirla/image_classification`. [Online; accessed 2021-03-31].

[51] Kaggle. Dogs vs cats binary classifier (inception v3). `https://kaggle.com/pranaykankariya/dogsvscats-binary-classifier-inception-v3`. [Online; accessed 2021-03-31].