JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by

Dominik Heindl

Submitted at

Institute for

Machine Learning

Supervisor

Univ.-Prof. Dr. Sepp Hochreiter

Co-Supervisors

Markus Hofmarcher, MSc

Elisabeth Rumetshofer, MSc

October 2020

Attention Based High Resolution Image Classification

Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelors's Program

Bioinformatics

JOHANNES KEPLER

UNIVERSITY LINZ

Altenbergerstraße 69

4040 Linz, Österreich

www.jku.at

DVR 0093696

Bibliographical Detail

Heindl, D., 2020: Attention Based High Resolution Image Classification. Bachelor Thesis, in English. – 39 p., Institute for Machine Learning Johannes Kepler University, Linz, Austria

Annotation

Modern digital images, especially in the field of medicine, have extremely high resolutions. Current state-of-the-art image recognition techniques, like Convolutional Neural Networks, cannot handle such high dimensional inputs. In this thesis I compared the standard approach of classifying images by downscaling them with an attention-based Multiple Instance Learning approach where the original image is split up into several smaller patches and low dimensional embeddings are calculated for each patch by a Convolutional Neural Network. All low dimensional embeddings are then again processed in a MIL fashion, where attention-pooling is used to determine class label and additionally the importance of each patch. The data set for this thesis consisted of ultra high resolution histological slides of human skin which were classified to contain Basal Cell Carcinoma or not.

Declaration

I hereby declare that I have worked on my bachelor's thesis independently and used only the sources listed in the bibliography.

I hereby declare that, in accordance with Article 47b of Act No. 111/1998 in the valid wording, I agree with the publication of my bachelor thesis, in full to be kept in the Faculty of Science archive, in electronic form in a publicly accessible part of the IS STAG database operated by the University of South Bohemia in České Budějovice accessible through its web pages. Further, I agree to the electronic publication of the comments of my supervisor and thesis opponents and the record of the proceedings and results of the thesis defence in accordance with aforementioned Act No. 111/1998. I also agree to the comparison of the text of my thesis with the Theses.cz thesis database operated by the National Registry of University Theses and a plagiarism detection system.

Place, Date

Dominik HEINDL

Contents

Abstract

Convolutional neural networks (CNNs) are the most widely used image recognition technique. They are used in a variety of different environments to a high degree of success. In a clinical setting, CNNs can be used to diagnose cancer in histological slides. Basal cell carcinoma (BCC), for instance, is the most common form of skin cancer and must be diagnosed by pathologists on a regular basis. Histological slides however, have an exceptional high resolution in order to keep as much details as possible of the cellular structure. Current state-of-the-art CNN architectures, unfortunately, cannot handle such high dimensional inputs without downscaling the images to a reasonable size and therefore loose valuable information, which could otherwise be beneficial during training. In this thesis I try to compare standard CNN architectures, namely VGG16 and ResNet50, with an attention-based Multiple Instance Learning (MIL) approach. These attention-based models are not only capable of using more information than their standard CNN counterparts but are also able to find key regions of an image which in this case shows cells with BCC. These highlighted regions then could assist pathologists during the diagnosis process. I also show that this attention-based approach does not only match standard CNNs in terms of classification accuracy, but is also able to outperform them. While the best standard CNN achieved an Accuracy of 73.6% and an AUC of 0.91, the suggested attention-based MIL approaches reached an Accuracy of up to 94.6% and an AUC of 0.99 on the same dataset.

1 Introduction

Convolutional Neural Networks (CNNs) are the most widely used model class for image recognition tasks today [16, 18]. There are a wide variety of different architectures trained for different purposes, like facial recognition or cancer detection. One common theme for this type of neural networks is the size of the input images. Most CNNs are using, at least by today's standards, rather small images with a resolution of up to 4,096 $\times$ 4,096 pixels. Digital images nowadays, can be of much greater dimensions. Medical images for instance are often several giga pixels in size in order to examine cellular components. To overcome the limitation of modern CNNs, regarding the efficiency of modern algorithms and the available computing power, the high resolution images are usually downscaled to a manageable resolution. By resizing the image, a lot of information is lost and therefore unavailable for training. To avoid the problem of information loss, the original high resolution image can be divided into smaller patches, which then can be processed by a standard CNN architecture. The CNN is used to extract low dimensional embeddings from each image patch. All embeddings of one image are then treated as one bag for an Multiple Instance Learning (MIL) approach, where one bag with all embeddings then only has one label. To overcome the now arising weak labeling problem Ilse et al. [12] proposed a special pooling function which does not utilize a maximum- or average pooling but is an attention-based approach. This new pooling method is also capable of finding key instances in a bag which are responsible for triggering a specific bag label.

The aim of this thesis is to compare the proposed approach by Ilse et al. [12] using a standard VGG16 [26] and ResNet50 [8] as feature extractor, against a baseline model which utilizes the standard approach of image recognition by downscaling the original image. The baseline models are also based on a VGG16 and ResNet50. The different models are tested on a data set of histological slides of skin which either show Basal Cell Carcinoma or normal skin. Furthermore, the performance of the models with different embeddings is compared to see if one architecture is superior in terms of feature extraction in a medical setting. Finally, embeddings where the CNN for feature extractions was previously refined with the medical images (self-trained) are compared with embeddings from a standard CNN with the pre-computed weights according to the ImageNet data [4].

2 Related Work

2.1 Convolutional Neural Networks

Convolutional Neural Networks (ConvNets or CNNs) [19, 22] are a special form of artificial neural networks (ANNs) which are often used for computer vision tasks. Although CNNs were already introduced in 1998 by LeCun et al. [18] they only really gained attention in 2012, when Krizhevsky et al. [16] won the ImageNet ILSVRC challenge [24] with their implementation of a CNN called 'AlexNet'. Since then, the ImageNet ILSVRC challenge was repeatedly won by different CNN architectures. Unlike traditional ANNs, CNNs are made specifically to work with inputs like images and videos. These kinds of inputs would be too high dimensional and therefore too much to handle for a fully connected ANN. The number of parameters and computational time it would take for the ANN to process this data would become unfeasible very quickly. For instance, if the input for a standard ANN would be an RGB image with only 64×64 pixels, the number of connection of a single neuron in the first hidden layer would already be 12,288. CNNs use several functions and techniques to overcome this problem and limit the number of weights and biases.

2.1.1 Building Blocks of CNNs

In general a CNN is a combination of three different layers: (i) Convolution Layer with non-linear activation function, (ii) Pooling Layer, (iii), Fully-connected (FC) Layer. A CNN architecture consists of a different number of these layers in an alternating sequence. The general trend of CNNs in recent years was to get deeper as the computational power increased.

Convolution Layer A major part of the convolution layer is a kernel or filter. This filter is usually small in size, e.g. 3×3 pixels, but covers the whole depth (color channels of an RGB image). With this small size, the kernel is only able to cover a small fraction of the input at one time. The network then computes the dot product of the kernel and the small part of the image which it covers. By sliding the kernel along the width and height of the image and repeatedly calculating the dot product, see Figure 1, a 2D activation map is created. Based on the used kernel, this activation map then shows certain structures of the image, like edges, corners or even more complex structures like animals or faces in deeper layers of the network. Usually

a non-linear activation function like ReLU ($max(0, x)$) is applied to the output of these dot products.

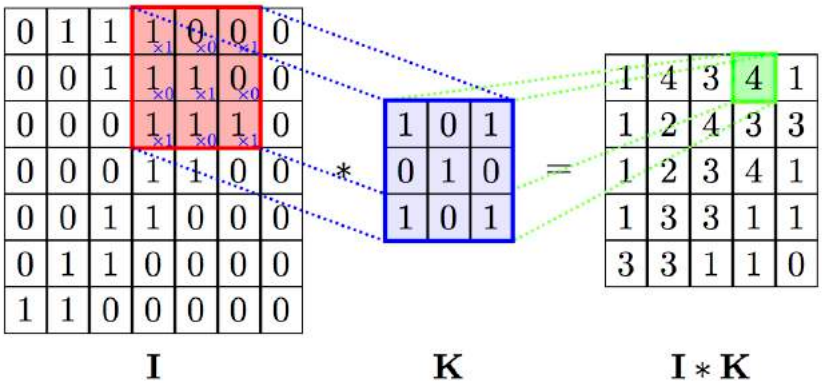


Figure 1: Visualization of a 2D convolution operation. **I** is the original input, The blue square labeled **K** is the kernel. The result on the right side is the dot product of the input and the kernel. [30]

Important parameters for controlling the convolution step in a CNN are the number of activation maps, the stride and the kind of padding. The number of activation maps is controlled by the number of kernels which are connected to a single receptive field, each kernel is then able to detect different structures in the small image patch. The receptive field in this case is the region of the input which is connected to neurons in the CNN. Several neurons can be connected to the same region of the input. With the stride parameter it is controlled how many steps, in the case of images, pixels, the kernel slides across the input. In general, the stride is very small with only 1 or 2 pixels. This leads to a large overlap of the different dot products. The output volume of a convolution operation will be smaller than the input if the kernel is bigger than 1×1 pixel. To compensate this, padding can be used. Zero-padding, for instance, adds empty pixels to the borders of the input image. This way it is possible to control the size of the output of the convolution layer. To calculate the size of the output the following formula can be used:

$$\frac{(\mathrm{V} - \mathrm{R}) + 2\mathrm{Z}}{\mathrm{S}} + 1, \tag{1}$$

where V is the input volume size (in the case of an image: height $\times$ width), R is the size of the receptive field, Z is the number of pixels added through zero-padding and S is the stride size.

Using Parameter Sharing, a CNN is able to drastically reduce the number of parameters. This is achieved by the assumption that a feature, e.g. simple edge detection, which is useful

in one part of the image, can also be useful somewhere else in the same image. Based on this assumption, the weights are shared between neurons in one activation map. In case of the first convolution layer of the VGG16 model which uses a receptive field of size 3×3 on a three channel RGB image and connects 64 kernels to each receptive field, this would lead to $(3 * 3 * 3) * 64 = 1,728$ individual weights. In the second convolution layer, the number of weights would increase to $(3*3*64)*64 = 36,864$ weights due to the deeper output of the first convolution layer. The depth of the output volume of this convolution layer is 64 because of the number of kernels used, each of this depth slices then shares the weights and bias between the $224 * 224 = 50,176$ neurons. During the backpropagation the gradients of each neuron of one depth slice are added up and only update the weights of one of these depth slices.

Pooling Layer This layer is used to reduce the dimensionality of the input image. Like in the convolution layer, a filter is sliding across the input but now the filter takes the maximum or average of the input which is covered by the filter. An illustration of this step is shown in Figure 2. The most common type of pooling is the max-pooling, mostly in combination with a filter size of 2×2 and a stride of 2. Pooling layers in general lead to a loss of information, by avoiding larger filters in this layer, the information loss can be kept in check. However, there are some researchers who suggest to discard the pooling layer as a whole and use another convolution layer with a bigger stride instead. [27]

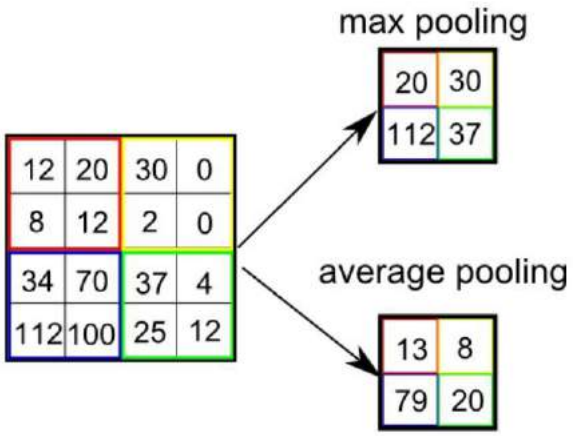


Figure 2: Visualization of a max- and average pooling function. Either the maximum or the average value of each 2×2 square on the left is taken during the pooling process. [25]

Fully-connected Layer This type of layer is usually used at the end of a CNN. Like traditional ANNs, neurons in this layer are connected to all activations of the previous layers. Fully-connected layers also derive the class scores from the activations which are then used for classifying the input image.

A simple example of a CNN is shown in Figure 3. The input, in this example a black and white image with 28 $\times$ 28 pixels, shows a handwritten digit which the CNN should classify. First, a convolution layer with a kernel size of 5 $\times$ 5 is used, followed by a max-pooling step. These steps are repeated a second time before the output of these operations is fed into a fully connected layer where a neural network learns the differences between the digits.

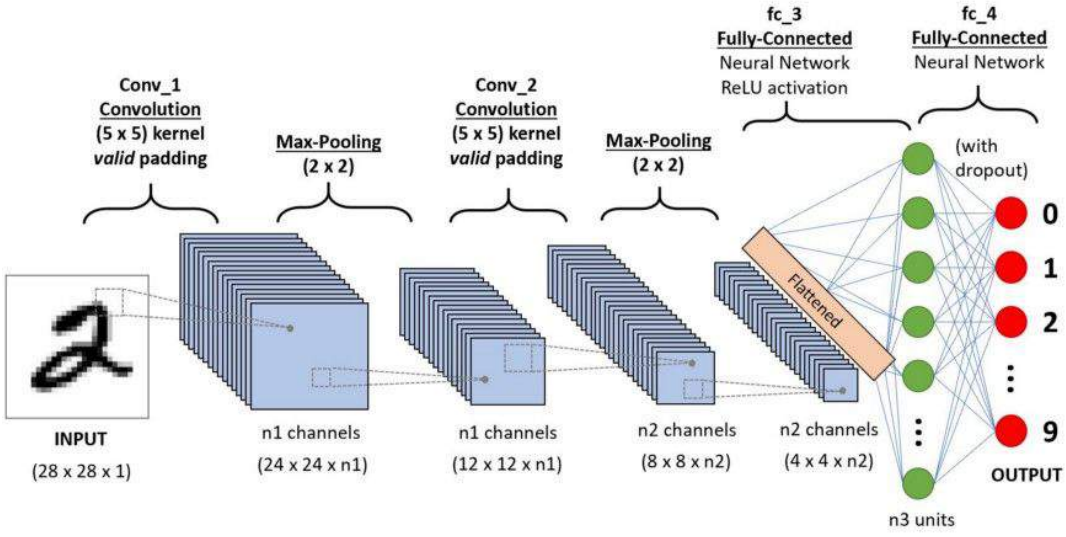


Figure 3: Simple example of a CNN which classifies handwritten digits. [25]

2.1.2 VGG

The VGG net was introduced by Simonyan et al. [26] in 2014. They also submitted their work to the ImageNet ILSVRC challenge in the same year. The top 5 test error rate of a single VGG net was 7.0%. Simonyan et al. tested different numbers of convolution layers for their CNN architecture and its effect on the model performance. Their results showed a significant improvement of the accuracy by using a deeper model. In contrast to previously known models like AlexNet [16], VGG nets used way more convolution layers and a significantly smaller kernel size.

The input for VGG net is a 224 $\times$ 224 pixel RGB image. This image is then passed trough

several convolution layers where the kernel is set to 3×3 pixels. The number of convolution layers is the only difference between the different VGG architectures e.g. VGG11, VGG16 and VGG19. The stride is set to 1 pixel and zero-padding is used to preserve the original input size until a pooling layer. In total five pooling layers are used after some convolution layers. The filter size for the max-pooling operation is 2×2 pixels, with a stride of 2. After the last pooling layer, three FC layers are utilized to infer the final class label. The first and second FC layer have an output size of 4,096 while the last FC layer only has 1,000 output units (one for each class in the ImageNet ILSVRC challenge). As a last step, soft max is applied to the output of the last FC layer.

An overview of the VGG architecture is shown in Table 1. Configuration **D** in this Figure shows the VGG16 architecture, which was used during this project. VGG16 has 13 convolution layers, as previously described, and 3 FC layers. The total number of parameters for this model is 138 million.

2.1.3 Residual Nets

Residual Nets (ResNet) were introduced by He et al. [8] in 2015. In the same year, He et al. won the ImageNet ILSVRC challenge where their single ResNet-152 architecture achieved a top 5 test error rate of 4.49%.

He et al. tried to overcome the problem of vanishing/exploding gradients, which plagues very deep neural networks, by utilizing a residual learning network. Deep neural networks tend to saturate in accuracy when they have too many layers. This stagnation in accuracy is not caused by overfitting of the model. ResNets utilize layers with identity mapping to reduce this phenomena of deep neural networks. The idea is that the weight layers in the network do not learn a direct mapping of the input to an output in the form of $\mathcal{F}(\mathrm{x})$ but rather a residual mapping in the form of $\mathcal{F}(\mathrm{x}) + \mathrm{x}$. This new mapping can be easier to learn and is realized by using an identity function, a function which returns only its input without

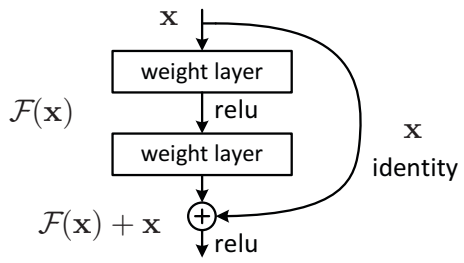


Figure 4: Residual building block [8]

A	A-LRN	B	C	D	E
11-layer	11-layer	13-layer	16-layer	16-layer	19-layer
input (224 × 224 RGB image)					
3×3, 64	3×3, 64	3×3, 64	3×3, 64	3×3, 64	3×3, 64
	LRN	3×3, 64	3×3, 64	3×3, 64	3×3, 64
2×2 max pool, stride 2					
3×3, 128	3×3, 128	3×3, 128	3×3, 128	3×3, 128	3×3, 128
		3×3, 128	3×3, 128	3×3, 128	3×3, 128
2×2 max pool, stride 2					
3×3, 256	3×3, 256	3×3, 256	3×3, 256	3×3, 256	3×3, 256
3×3, 256	3×3, 256	3×3, 256	3×3, 256	3×3, 256	3×3, 256
			1×1, 256	3×3, 256	3×3, 256
					3×3, 256
2×2 max pool, stride 2					
3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512
3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512
			1×1, 512	3×3, 512	3×3, 512
					3×3, 512
2×2 max pool, stride 2					
3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512
3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512	3×3, 512
			1×1, 512	3×3, 512	3×3, 512
					3×3, 512
2×2 max pool, stride 2					
FC-4096					
FC-4096					
FC-1000					
soft max					

Table 1: Each column of the table shows a different VGG model architecture (A-E), ranging from 11 (left) to 19 layers (right). The rows represent the different layers of a CNN. The parameters of the convolution layers are denoted as: <kernel size, number of kernels>. [26]

applying any new transformation. This residual mapping is depicted in Figure 4. This building block of a residual network can be implemented via a feed forward neural network and "shortcut connection". These shortcut connections skip a few layers before the output of a shortcut connection is added to the output of a few stacked weighted convolution layers. After adding both outputs together, a nonlinearity function like ReLU is applied. The identity function does not add any further parameters nor computational cost to the model. Outputs of the weighted layers and the identity function must be of the same dimension. Otherwise a linear projection has to be applied on the output of the identity function, like zero padding or a 1×1 convolution.

The overall architecture of the used ResNet50 model is shown in Figure 2. Like VGG16, ResNet50 uses 224×224 pixel RGB images as input. The first convolution layer in the ResNet50 uses a 7×7 kernel with a stride of 2, followed by a max-pooling layer with a stride of 2. Afterwards four different residual learning blocks, without additionally pooling layers are used. Average-pooling is used after the last residual block, followed by a FC layer and soft max.

2.2 Multiple Instance Learning

In traditional supervised learning settings all instances of a data set, $X = \{x_1, \ldots, x_n\}$, have a target variable or label y. In the binary classification task the label is $y \in \{0, 1\}$. In the case of Multiple Instance Learning (MIL) [2, 5, 12, 20] several instances form a set, called bag, and only the bag as a whole is labeled but not each instance in it. In this case the data is weakly labeled. In a simple MIL setting, the bag label is only negative if all instances in it are negative as well. If at least one instance is positive, then the bag label also turns positive. This can be expressed by:

$$Y = \begin{cases} 0, & \text{iff} \sum_n y_n = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

where Y is the bag label and y_n represents the labels on an instance level.

A general MIL framework will consist of 3 steps: (i) feature extraction of the instances, (ii) aggregation of the extracted features, also called pooling and (iii) a final transformation of aggregated features to infer a bag label. In case the instances do not need any further process-

layer name	18-layer	34-layer	50-layer	101-layer	152-layer
	input (224 × 224 RGB image)				
conv1	7×7, 64, stride 2				
	3×3 max pool, stride 2				
conv2_x	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	average pool				
	FC-1000				
	soft max				

Table 2: Each column of the table shows a different ResNet model architecture, ranging from 18 (left) to 152 layers (right). The rows represent the different layers of a CNN. The parameters of the residual blocks are denoted as: <kernel size, number of kernel × how often this block is repeated>. [8]

ing, the first step of feature extraction can be omitted. For several tasks, like image recognition, it is necessary to compute or transform the instances into computer usable features. The second step, in the general MIL framework, also called MIL pooling can be realised in several different ways. This step is responsible for aggregating the features on an instance level and compute a representation for the whole bag. Pooling can be realised in various different forms, if they ensure two properties: (i) the function must be permutation-invariant and (ii) the function must work with different bag sizes. Two common examples for this pooling function are the mean- and max- pooling operator. Mean-pooling obtains a bag representation by calculating the mean over the extracted features of a bag. The max-pooling function on the other hand uses the maximum of the extracted features of one bag as the bag representation. Other pooling functions, like 'attention-pooling' introduced by Ilse et al. [12] offer several advantages over

the simple mean and max operations, see Section 2.4 The last step of a basic MIL framework is the transformation of the bag level aggregation into a bag label Y. This transformation, as well as the first feature extraction step, can be realized using neural networks. The use of neural networks for these tasks has the advantage that the whole process is trainable in an end-to-end fashion if the MIL pooling operation is also differentiable.

Because certain problems can be easily reformulated into a MIL problem, it is a good fit for several topics in different fields of research. For instance, Dietterich et al. [5] used MIL in chemoinformatics when they tried to predict if a desired drug activity is achieved by certain molecule configurations. Because it would be unfeasible to test every molecule configuration individually, MIL was used. More recently, Kimeswenger et al. [14] used a MIL approach to classify basal cell carcinoma in histoligical slides. To avoid loosing valuable data during the downsampling for a standard CNN, they divided the high resolution slides into smaller patches and treated all patches of one image as a bag for MIL. But MIL is also used in other computer vision tasks. Usually images contain bounding boxes and local annotations but these involve immense work from humans in order to obtain a reasonable sized data set. Therefore, Wu et al. [32] used weakly labeled data from the internet for their image classification and auto annotation experiments.

The acquisition of data is another advantage of MIL. Weakly labeled data, for instance for object detection in images, or finding cancerous cells in medical slides, is easier to find and collect than normal labeled data as it does not involve much additional manual labour.

2.3 Attention

Humans have the ability to focus on key regions of images or sentences and identify the object in an image or the meaning of a sentence. Intuitively, humans lay their attention on important parts which help them understand the context of an image or sentence. Figure 5, for instance, shows a dog in a sweater. While humans look at this image and instantly focus on the dog features, like the pointy ears or the black nose and neglect the sweater to confirm that there is a dog in this picture, machines do not work as intuitively. For a computer vision algorithm each part of the image would be equally important. In examples like depicted in Figure 5 not all areas of an image are of equal importance when trying to identify the main object in an

image. For machines to ”think” more human-like and not waste resources, they also have to focus on the important parts of images or sentences. To identify the important regions, attention [1, 31, 33] was introduced. In general, attention can be interpreted as an ”importance weight” for different parts of an image or sentence which measures how much this part correlates with other parts of the input.

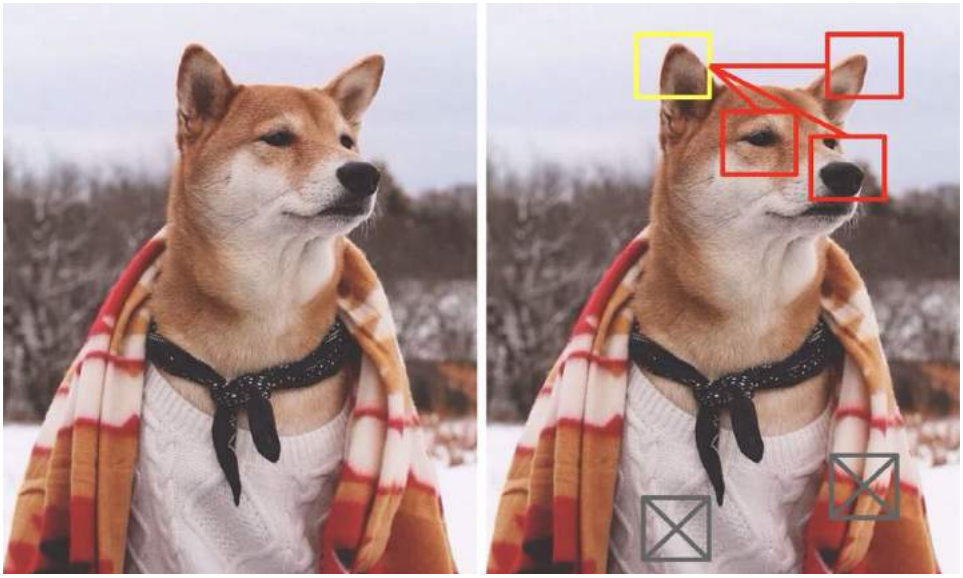


Figure 5: Coloured boxes show ”typical” dog features on which humans lay their attention when looking at this image. They grey boxes indicate unimportant features for the classification of a dog. [31]

The idea of attention in deep learning originated in the field of machine translation. Prior to using an attention mechanism, encoder-decoder models as explained by Sutskever et al. [28] were used for machine translation. These models are based on recurrent neural networks (RNNs) where the encoder part tries to transform the information of an input sequence into a context vector with a fixed length. The decoder part of such models then takes this vector as an input and gives an according output, for example the prediction of the next word in another language. This approach of machine translation however, has problems with longer input sentences. The produced context vector of the encoder part should be a good representation of the input sequence. With this approach, an encoder does not take into account the length of the sentence, which can lead to problems if the input is very long. The RNN used in the encoder can ”forget” what the beginning of the sentence was once it reaches the end, leading to a low quality representation of the input. To overcome this problem with longer sentences Bahdanau

et al. [1] introduced an attention mechanism. The context vector of the model of Bahdanau et al. is a weighted sum of annotations which are mapped to the input sentence by an encoder. The annotations used in this approach hold information about the whole input sequence but mostly focus on the words which come close before and after the *i*-th word of the input. The weight for the individual annotations is determined by a simple neural network. This way, the context vector is no longer of a fixed length and the information is distributed more evenly across the vector for the decoder.

Driven by the success of this approach, the idea of attention in machine learning gained further momentum and was also applied in other fields. Xu et al. [33] used attention in computer vision to produce computer generated descriptions of images. Which is a non-trivial task, as the computer not only has to identify the object or scenery in an image, it also has to produce a natural language description. Xu et al. described different approaches of attention in their paper. They differentiated between a ”soft” and a ”hard” attention mechanism.

Soft Attention This type of attention is a deterministic approach where all parts of an image, or sentence, are considered when calculating the attention weights. The attention vector then is also laid over the whole input. The calculation of the individual attention weights can become computational expensive if the input is too large. One advantage of this form of attention is the fact that it is differentiable and can therefore be trained using back-propagation. This attention mechanism was used by Bahdanau et al. [1], when they first introduced the idea of attention and also in the work of Ilse et al. [12] and Kimeswenger et al. [14]. Soft attention is also used in this thesis, as it allows the visualization of key regions in the histological slides, see Figure 11.

Hard Attention This attention approach on the other hand is a stochastic approach, where each part of the input is viewed and attended at a time. This leads to fewer and more efficient calculations but has the disadvantage that the training of a model is not as straight forward as with soft attention. Because the hard attention mechanism is not differentiable, other methods like reinforcement learning have to be used to train the model.

2.4 Attention-based Multiple Instance Learning

Proposed by Ilse et al. [12] attention-based MIL differs from more traditional MIL approaches by utilizing a different pooling function. Widely used pooling functions like mean and max pooling posses unwanted characteristics. The mean pooling operator for instance could be a good choice in the case of predicting whole bag labels, but it would fail in case the labels on an instance level should be predicted. Max pooling on the other hand would be more suitable for the instance level predictions and not for whole bag labels. These pooling functions neither can be fine-tuned nor adapted in any way to a new data set or problem.

In order to convert the problem of high resolution image classification into a MIL problem, the high resolution images are divided into smaller patches. Features which are calculated based on these patches are then used as an input for a final classification network. All features of one image are combined into a bag. One bag of features only has one bag label associated with it. By the proposal of Ilse et al. the pre-processing or feature calculation step is done by a neural network to obtain trainable features. Ilse et al. used a self implemented neural network to calculate low dimensional representations of the image patches and used these as an input for a final classification network. In this project the extracted features of the images were the activations of the last layer of a CNN rather than a self implemented feature extractor. Here, VGG16 [26] and ResNet50 [8], were used as feature extractors, where the activation of the last layer was saved. This feature extraction leads to an immense reduction in dimensionality where one image patch can then be described by a single vector.

In contrast to some other MIL approaches, mean or max pooling is not utilized to infer the bag label. Rather the idea of attention is used, where every instance in a bag is associated with an attention weight. This weight is determined by a neural network. The attention is also helpful in finding key instances of a bag which trigger the associated label. A bag of K embeddings is represented by $H = \{\mathrm{h}_1, \ldots, \mathrm{h}_K\}$ then the proposed attention MIL pooling is:

$$\mathrm{z} = \sum_{k=1}^{K} a_k \mathrm{h}_k, \tag{3}$$

where:

$$a_k = \frac{\exp \mathrm{w}^\top \tanh\left(\mathrm{V}\mathrm{h}_k^\top\right)}{\sum_{j=1}^{K} \exp \mathrm{w}^\top \tanh\left(\mathrm{V}\mathrm{h}_j^\top\right)}, \tag{4}$$

where $\mathrm{w} \in \mathbb{R}^{L\times 1}$ and $\mathrm{V} \in \mathbb{R}^{L\times M}$ are trainable parameters.

2.5 Basal Cell Carcinoma

Basal Cell Carcinoma (BCC) [3, 7, 13, 17] is the most common form of skin cancer today. BCC usually appears in the adult Caucasian population during their fourth life decade and beyond. BCC mostly develops in sun exposed areas of the body like face, neck and arms but can also appear on other body parts. Although widely spread, BCC is highly curable if the diagnosis is made early.

Basal Cells BCC occurs in basal cells, a type of cell in the outer most layer of the skin, the epidermis. The epidermis forms the top layer of human skin and serves as a protection layer from environmental influences. In the bottom layer of the skin, the dermis, hair follicles, blood vessels and melanocytes can be found. The basal cells can be found on the edge between the dermis and epidermis. Figure 6 shows the basic structure of human skin and the location of basal cells between the dermis and epidermis. They are relatively undifferentiated but can reproduce promptly and support the growth of new skin cells.

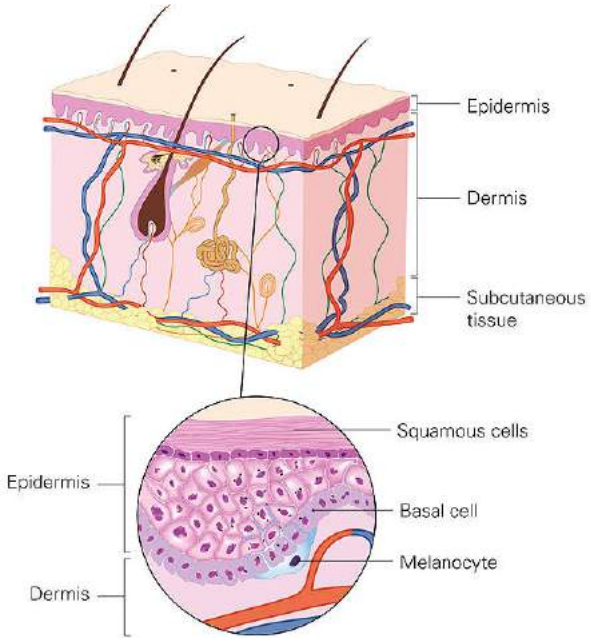


Figure 6: Structure of human skin and the location of basal cells between the epidermis and the dermis. [29]

Histopathology and appearance BCC can be divided into several sub-groups which differ in their appearance and growth-pattern. Correctly identifying the sub-type of BCC in this spectrum is crucial to find the most suitable treatment. First and foremost, BCC can be sep-

arated into differentiated and undifferentiated tumors, based on their specific growth-pattern and differentiation from the surrounding cells. Undifferentiated BCC tumors, do not show specific differentiation features e.g. to sebaceous or eccrine cells. This sub-group can be further separated into indolent- and aggressive-growth tumors. Nodular BCC, part of the indolent subgroup, is the most common form of BCC. Aggressive-growth tumors among others include the infiltrative growing BCC. Differentiated BCC is less common and shows clear differentiation from surrounding cell lines. Sub-types of this group are Follicular, Pleomorphic or Recurrent BCC.

Depending on the body part, age, severeness and sub-type of the disease, the appearance of BCC can be drastically different. BCC typically appears as a pink or flesh colored papule with small visible veins. The edges of this histomorphological sub-group of BCC are clearly separated from the surrounding skin, see the left example in Figure 7. Additionally also bleeding or crusting around the tumor can appear, some lesions may also be slightly translucent. Other, more aggressive growing, subtypes of BCC, as shown on the right side in Figure 7, are not as clearly differentiated. This infiltrative growing BCC is depressed in the skin and has a more pinkish to reddish color. Its appearance reminds of a scar or open sore. On some occasions the BCC can be surrounded by a area of hyper-pigmented skin. Occasionally, aggressive forms of BCC can lead to tenderness or pain but in general BCC does not induce any other complaints aside from local destructive behaviour in the region of eyes and ears.

Risk Factors By far the biggest risk factor for BCC is prolonged exposure to sun light. The UV-radiation can damage the DNA in the skin cells permanently and lead to irreparable mutations. Further risk factors are the exposure to arsenic, coal tar derivatives or inflammatory skin conditions. Certain phenotypes are also able to facilitate the growth of BCC. People with light-colored hair or eyes as well as a fair skin tone are especially in danger of developing BCC.

Treatment The typical method for removing BCC includes a small surgical excision which removes the tumor. ”Mohs surgery” is a special type of surgical procedure, where thin layers of skin are removed one at a time and examined under a light microscope. If no cancer cells are found in a layer of skin, the surgery is finished. If the lesion is small enough, it can also

Figure 7: Example images of different BCC sub-groups. **Left**: Nodular BCC - the most common form of BCC. **Right**: Infiltrative growing BCC. [3]

be freezed and destroyed with liquid nitrogen. Another approach would be the use of a special tool called curettage which has a sharp, ring-shaped tip. This instrument is used to remove the majority of the cancer cells, the remaining cells are treated with electricity.

If these methods cannot be used due to the age of the patient, immune deficiencies or other complicating factors, there are also non-surgical treatments to remove BCC. If the diagnosis is made early enough, local therapies can be utilized such as Photodynamic, Radiotherapy or the use of Ingenol mebutate (IM). Photodynamic therapy (PDT) uses a topically applied gel and special wavelength light. The light will start a reaction in the gel which produces highly reactive oxygen singlets. These oxygen singlets will start the destruction of tumor cells. Radiotherapy utilizes high-energy electromagnetic waves such as X-rays or electron beams, which lead to severe damage on a DNA level and ultimately cell death. Ingenol mebutate is a macrocyclic diterpene ester, which will start cell necrosis shortly after the topical application. An inflammatory response will follow after this treatment.

BCC rarely spreads out, only bigger tumors which are larger than three cm in diameter increase the risk of metastases. If the tumor spread, then the localized therapies may not be sufficient to remove the cancer nests. In such cases, systemic therapies such as chemotherapy or Hedgehog pathway inhibitors can be used. Chemotherapy is not frequently used to treat BCC as there are alternatives with a higher cure rate and less adverse effects. The use of Hedgehog

pathway inhibitors is a relatively new treatment. These inhibitors block the Hedgehog signaling pathway, which is crucial for the building of new cancer cells. By inhibiting this protein synthesis, cancer cells can be prevented from further growth.

2.6 Histopathology and Histological Slides

Histology [6, 9] is a field of biology which studies the microscopic structure of tissues and cells. It is used to understand the function of tissues. Forensics also uses histology to gain information about unexplained deaths or the surrounding environment of the tissue. A special field of histology, called histopathology focuses on the diagnosis of diseases and the effect of the treatment. Trained pathologists or histopathologists are able to identify abnormal conditions of the tissue, like cancer.

To examine the fine structure of cellular components it is necessary to view them under a microscope. To achieve this, histological slides [9, 10, 11, 21] have to be prepared. The preparation of these slides can be divided into five different steps: (i) Tissue Fixation, (ii) Dehydration, Clearing, (iii) Embedding, (iv) Sectioning and (v) Staining

Tissue Fixation Cells release special enzymes after they die, which start an autolysis process. The speed of this process varies between different types of tissues. Additionally, the tissue could also decay because of bacterial or fungal contamination. To prevent the degradation of the tissue, make it more resilient and preserve the cellular structure, the tissue must undergo the process of fixation. The fixation method depends on factors like: type of tissue, type of desired staining method, type of experiments and time restrictions. Generally, samples should undergo the process of fixation immediately after collecting the sample to ensure good results.

The main fixation method today uses 10% neutral buffered formalin (Formaldehyde) to preserve the tissue. Other aldehydes, such as Paraformaldehyde or Gluteraldehyde can also be used. The aldehyde forms covalent bonds between neighbouring amine-groups, thus lowering the reactiveness of the cross-linked molecules. The specimen is placed in a small container together with the aldehyde, after 24-48 hour the fixation process is completed.

A much more rapid way to fix the tissue is to freeze it. This technique is used when the tissue has to be examined rather quickly, like in surgical biopsies. This type of fixation

does keep some enzyme and proteins and fats in tact, which would otherwise be destroyed or washed away with other fixation methods. Very fine structure of the cells on the other hand can be destroyed by ice crystals. Furthermore, freezing the sample requires special equipment to keep it cold enough during the ongoing procedure. Thawing could potentially lead to a degradation of the tissue which makes it unsuitable for further experiments.

Under special circumstances, like examining antibody binding sites, the usage of aldehyde is counter intuitive as it destroys this type of bond during the fixation. In such cases, other organic solvents can be used for fixation. Methanol and Chloroform-containing fixatives are cooled and the sample is immersed. This method however leads to the destruction of the 3D organization or shrinkage of the sample. But, like freezing, it is a more rapid way to fix a specimen.

Dehydration, Clearing By immersing the sample in solutions with increasingly higher concentration of alcohol, water and formalin is washed away. To remove the alcohol, organic solvents, like xylene, are used. This allows the sample to be embedded in paraffin.

Embedding The fixated and dehydrated tissue alone has not enough structural rigidity to be sliced into thin pieces, therefore it is necessary to embed the tissue in a supporting material. In most cases this is done by submersing the sample in liquid paraffin wax which is then cooled down to solidify the wax.

Sectioning Excessive wax from the solidified block is removed to expose the sample inside. Then, in most cases, a microtome is used to cut thin slices (1-50 µm) off the samples. The cut off slices are then transformed onto the surface of a warm water bath. Microscopy slides are then placed under the floating slices to lift them out of the water and place them on the slide. Other cutting devices which are used for sectioning the embedded samples are a vibrotome, which uses vibrating blades to cut slightly thicker slices (100-200 µm) or a cryotome, which is an actively cooled version of a normal microtome to handle frozen samples.

Staining The majority of cells are transparent and therefore it is very hard to examine or distinguish them under a light microscope. With special stains this problem can be elimi-

nated. There exist several different stainings to highlight different elements or parts of the cell. The most common staining technique in histology uses combination of hematoxylin and eosin (H&E). Eosin stains cytoplasm pink/red and hematoxylin is used to stain cell nuclei blue. This is achieved by submerging the slides with the specimen on it in a small container of the hematoxylin, rinse them down with tap water and then stain them again with eosin. This procedure makes the specimen observable under a microscope. Although widely used, H&E is not suitable for all types of staining. Particular tissues or structures need other staining techniques to make them visible. For instance, to stain and identify different blood cells giemsa combined with eosin is used. Luxol on the other hand can be used to stain myelin, a part of the nervous system.

3 Experimental Setup

3.1 Data Set

The data set consists of 811 images of histological slides of skin. The images either show a single slice of a probe (Figure 9, left) or multiple slices of the same probe (Figure 9, right). 599 of the 817 samples contained BCC while the rest shows normal skin probes. The data set was randomly split into a training set with a relative size of 84% and a test set of 16%. 25% of the training set was randomly chosen as validation set. The slide images were retrospectively collected by the Kepler University Hospital and the Medical Hospital of Vienna according to ethics votes number 2085/2018 and 1119/2018 from the Ethics committees Upper Austria and Medical University of Vienna respectively. The height of the images ranged from 6,884 to 47,939 pixels and the width from 7,360 to 99,568 pixels. With a few exceptions, it was the same data set used by Kimeswenger et al. [14] for their related work on this topic.

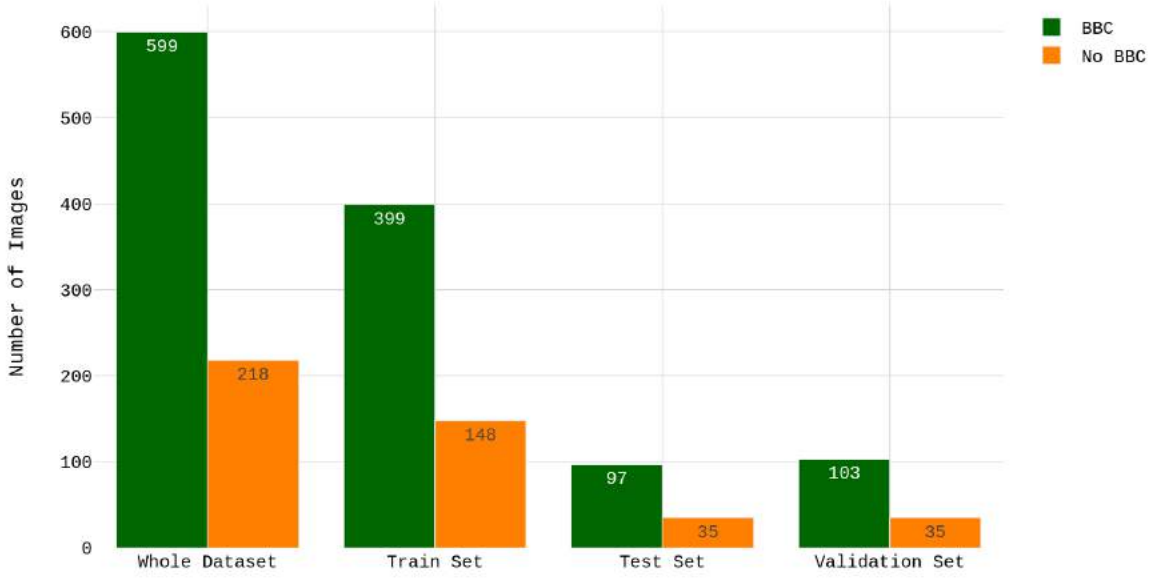


Figure 8: The figure shows the imbalance of the whole dataset, as well as all the different splits. All splits have approximately the same ratio of BCC to non BCC examples: 73%:27%.

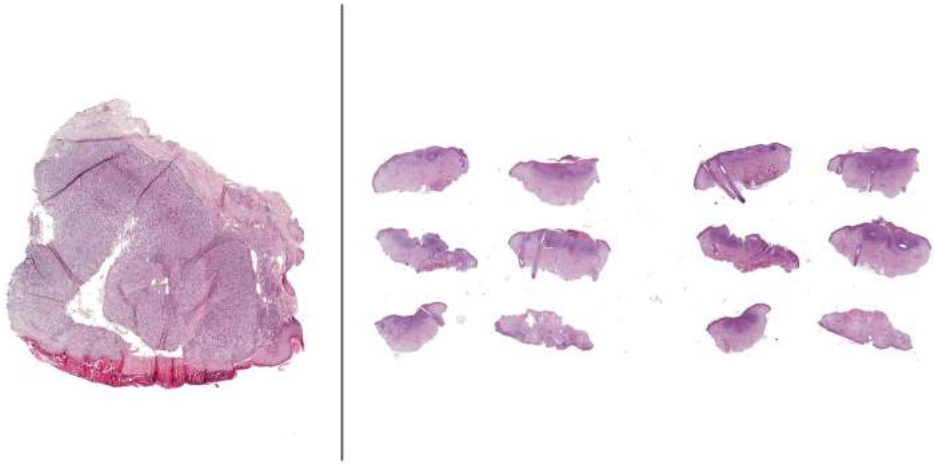


Figure 9: Examples of Histological slides **Left**: a single slice of one probe. **Right**: multiple slices of one probe.

3.2 Methods

3.2.1 Baseline Model

The baseline models were standard VGG16 and ResNet50 implementations. Both models were initialized with pre-trained weights, calculated on the ImageNet [4] data set, and fine-tuned for 100 epochs with downscaled versions (224×224 pixels) of the images of the training set. Model hyperparameters were optimized using the previously mentioned validation set. Final performance metrics were evaluated on the test set, with the best performing model on the validation set. All models utilized the Adam Optimizer (*lr: 2e-04, weight decay: 5e-04*).

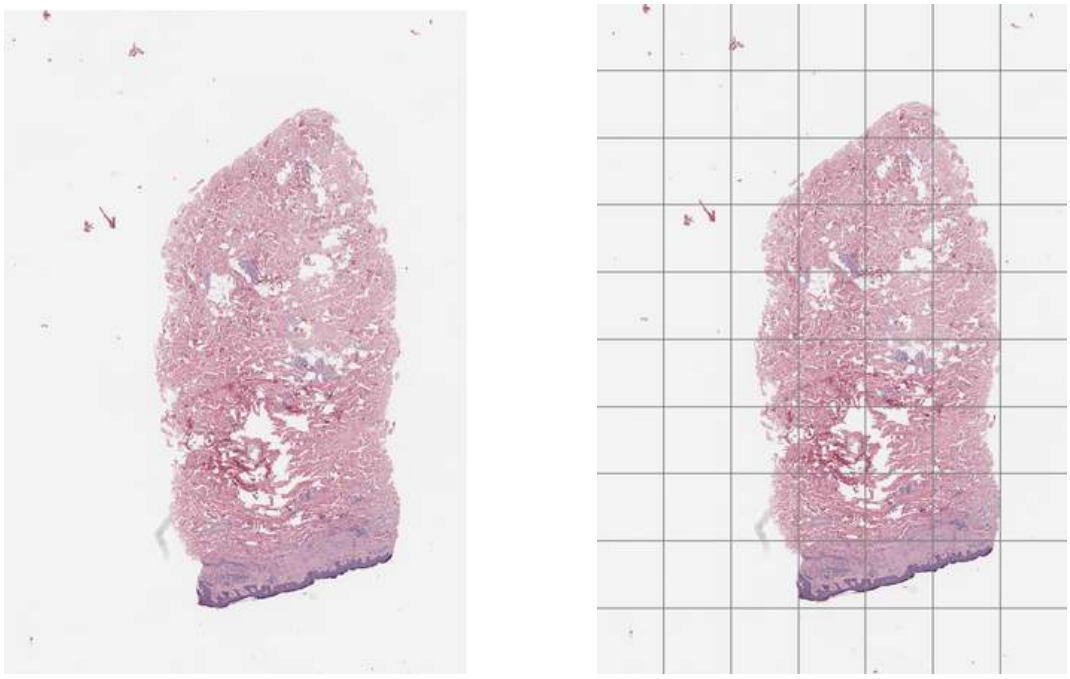
3.2.2 Embeddings

For the attention-based MIL models it was necessary to compute low-dimensional representations of the images, also called embeddings, as input for the second classification network. These embeddings were calculated using the PyTorch [23] implementation of VGG16 and ResNet50. The activation of the last layer of each image and model was saved as a one dimensional vector. Each CNN architecture computed these embeddings twice. First, the embeddings were calculated with the original pre-trained weights, trained on the ImageNet [4] data set. In a second step, the embeddings were calculated with the already fine-tuned baseline models. This lead to the self-trained embeddings, which are hypothesized to yield better representations of the images than their pre-trained counterparts.

3.2.3 Attention-based MIL Model

For the attention based approach, the original images were divided into patches with a size of 224 $\times$ 224 pixels, zero-padding was used where necessary. A visual representation of this step can be seen in Figure 10. Patches with no information gain (empty patches) were excluded from the experiments to optimize the runtime as well as the memory requirements. For this step, the average color intensity c_p for all patches p was calculated and all patches with a max c_p of more than 95% of the maximum $c_{max} = max_p c_p$ were discarded. Finally, all patches were normalized to zero mean and unit variance without any stain-normalization.

Each of the remaining patches was processed to obtain a low dimensional representations as mentioned in Section 3.2.2. All embeddings of one histological slide were then combined and fed in a classification network with attention-pooling, see Equation 4. This model was trained for 50 epochs, the hyperparameters were tuned manually with the performance of the validation set and the SGD optimizer was used (*lr: 1e-03, weight decay: 5e-04*).



(a) Original Image (b) Image Patches

Figure 10: Preparation of the histological slides for the attention-based MIL model. **Left**: the original slide. **Right**: slide divided into smaller patches. Empty patches got excluded from the MIL model.

3.3 Evaluation

The main metric for comparing different models was the area under roc curve (AUC). This metric was also used for comparing the learning phase of the different models in Figure 12 and Figure 13. Other performance metrics include the F1 Score, Accuracy, Precision and Recall. The mean $\pm$ standard deviation for all performance metrics was calculated by re-training the models ten times and averaging performance values.

4 Results

The results are split into a performance metrics and visualization part. The performance metrics part includes the attention-based as well as the baseline models and compares AUC, F1 Score, Accuracy, Precision and Recall of all models while testing only AUC for significance. The visualization part of the results only compares the results of the attention-based models and their ability to find key regions in an image. Key regions in this case are cells with BCC. The attention weights of all models are compared with human labeled examples.

4.1 Performance Metrics

Performance metrics of all evaluated models are shown in Table 3. These results are the average of the model performances on the test set of all ten trained models. The table shows that the attention-based ResNet50 model with self-trained embeddings yields the highest AUC, F1 Score and Accuracy, making it the best performing model despite two models with higher Precision or Recall. In general, the attention-based MIL models outperform the traditional CNN implementations used as a baseline in almost all metrics. Table 5 supports this observation by showing a significant difference between the combined AUC of both baseline models and attention-based models with self-trained embeddings. Even when examining a single baseline model and its attention-based counterpart, a significant difference can be determined in favor of the attention model.

Table 6 then shows that there is no significant difference in terms of the combined AUC of the attention-based VGG16 and ResNet50 approach, with pre- and self-trained embeddings. However, there is a significant difference when comparing the AUC of the attention-based models with pre- or self-trained embeddings individually. While the VGG16 based attention model performs better with pre-trained embeddings, the ResNet50 model significantly outperforms it when using self-trained embeddings.

In Table 7 it can be seen that there is also a significant difference between the combined results (ResNet50 and VGG16) of models with pre- or self-trained embeddings. The models with self-trained embeddings seem to slightly outperform the models with pre-trained embeddings. However, there is a difference between the different embeddings when looking at the VGG16

and ResNet50 models individually. In the case of the attention-based ResNet50, the model with self-trained embeddings performs significantly better than the model with pre-trained embeddings. The VGG16 model with pre-trained embeddings, on the other hand, outperforms the model with self-trained embeddings.

4.2 Ability of identifying Key Regions

All attention-based models deliver comparable results and are able to classify most of the images correctly. One advantage of the attention-pooling method is, that the network does not only provide a class label but also key regions of the image itself. Figure 11 (a)-(d) visualizes the patches with an high attention-weight in green. The attention weights a_k provided by the networks were scaled using $a'_k = (a_k - \min(\mathbf{a}))/(\max(\mathbf{a}) - \min(\mathbf{a}))$. The more saturated a green patch the higher is the attention on this region. Patches with low attention were excluded from this visualization for a less cluttered image. To compare the computer generated attention-weights with a human labeling, Figure 11 (e) [15] shows the average fixation time of a trained pathologist and Figure 11 (f) shows the exact location of cancer cells marked by a human. While all attention-based models were able to predict the correct class in this example, the ResNet50 model with self-trained embeddings was additionally able to give the most accurate representation of key regions in this sample image. This would make this model the most useful one in practice, as it could provide additional insights when marking cancerous regions in histological slides. The VGG16 model with pre-trained weights, see Figure 11 (c) was also able to identify some of the important regions in the image. The ResNet50 model with pre-trained weights, see Figure 11 (a) as well as the VGG16 one with self-trained weights, see Figure 11 (d) were not able to identify key regions in a satisfying manner. Both of these models were able to predict the right class label, however due to their inferior visualization capabilities they are less suitable for computer-aided diagnosis than the other two models.

Model	AUC	F1 Score	Accuracy	Precision	Recall
Base. VGG16	0.9142 ± 0.082	0.8441 ± 0.005	0.7318 ± 0.006	0.7366 ± 0.004	$\mathbf{0.9887} \pm 0.016$
Base. ResNet50	0.9009 ± 0.033	0.8064 ± 0.028	0.6879 ± 0.031	0.7386 ± 0.006	0.8907 ± 0.065
A. VGG16 pre	0.9834 ± 0.003	0.9573 ± 0.008	0.9386 ± 0.010	$\mathbf{0.9785} \pm 0.000$	0.9371 ± 0.014
A. VGG16 self	0.9778 ± 0.005	0.9582 ± 0.003	0.9394 ± 0.005	0.9725 ± 0.007	0.9443 ± 0.007
A. ResNet50 pre	0.9757 ± 0.004	0.9460 ± 0.007	0.9212 ± 0.011	0.9541 ± 0.014	0.9381 ± 0.000
A. ResNet50 self	$\mathbf{0.9865} \pm 0.001$	$\mathbf{0.9633} \pm 0.004$	$\mathbf{0.9462} \pm 0.006$	0.9668 ± 0.004	0.9598 ± 0.006

Table 3: Performance metrics of the Baseline (Base.) and Attention-based (A.) models. Result of the best performing model for each metric in bold.

Model	TP	FN	FP	TN
Base. VGG16	96	1	34	1
Base. ResNet50	86	11	31	4
A. VGG16 pre	91	6	2	33
A. VGG16 self	92	5	3	32
A. ResNet50 pre	91	6	3	32
A. ResNet50 self	93	4	3	32

Table 4: Classification performance of the Baseline (Base.) and Attention-based (A.) models.

Model	AUC	p-value
Baseline (combined)	0.9076	
Attention (combined)	0.9822	**3.9e-08**
Baseline (VGG16)	0.9142	
Attention (VGG16)	0.9778	**9.1e-05**
Baseline (ResNet50)	0.9009	
Attention (ResNet50)	0.9865	**9.1e-05**

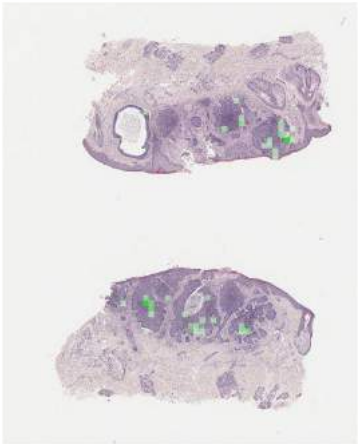
Table 5: Comparison between baseline and attention-based models with self-trained embeddings. A method is compared with the method beneath it, double lines separate individual tests. The p-value is the result of a Mann-Whitney-U-Test. Significant p-values are bold.

Model	AUC	p-value
ResNet50 (combined)	0.9811	
VGG16 (combined)	0.9806	1.4e-01
ResNet50 (pre-trained)	0.9757	
VGG16 (pre-trained)	0.9834	**2.9e-03**
ResNet50 (self-trained)	0.9865	
VGG16 (self-trained)	0.9778	**9.1e-05**

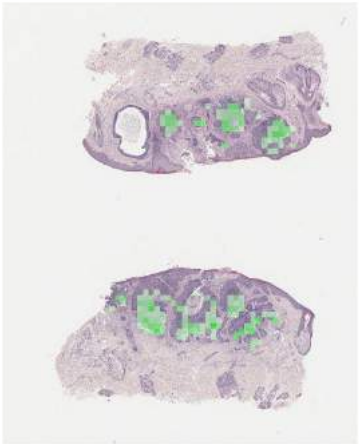
Table 6: Comparison between the attention-based models with the embeddings of an underlying ResNet50 and VGG16. The comparison includes the result of these models with pre-trained and self-trained embeddings. A method is compared with the method beneath it, double lines separate individual tests. The p-value results from a Mann-Whitney-U-Test. Significant p-values are bold.

Model	AUC	p-value
Pre-trained (combined)	0.9795	
Self-trained (combined)	0.9822	**4.3e-02**
Pre-Trained (ResNet50)	0.9757	
Self-trained (ResNet50)	0.9865	**1.4e-04**
Pre-Trained (VGG16)	0.9834	
Self-trained (VGG16)	0.9778	**8.4e-04**

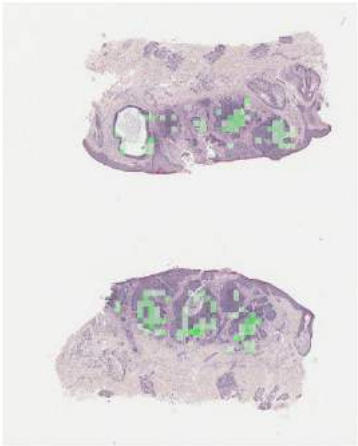
Table 7: Comparison between the attention-based models with pre-trained and self-trained weights. Comparison includes the results of the VGG16 and ResNet50 with both, pre-trained and self-trained weights. A method is compared with the method beneath it, double lines separate individual tests. The p-value results from a Mann-Whitney-U-Test. Significant p-values are bold.



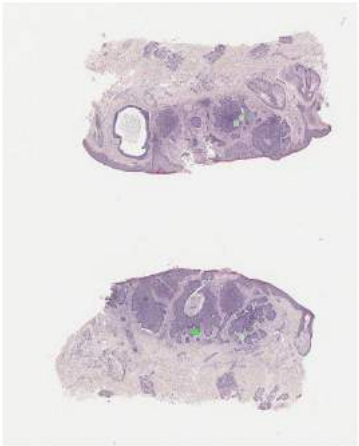
(a) ResNet50 pre-trained | True: BCC | Predicted: BCC



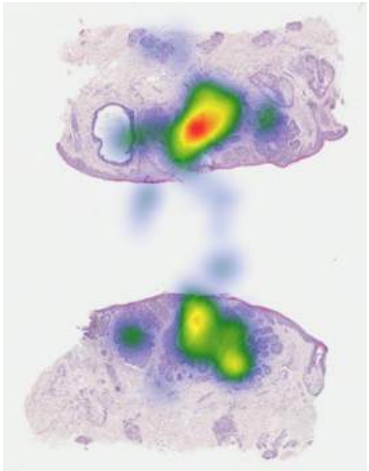
(b) ResNet50 self-trained | True: BCC | Predicted: BCC



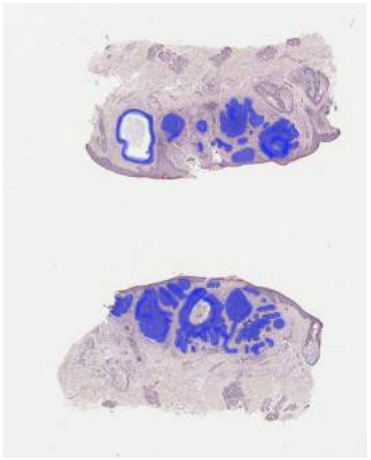
(c) VGG16 pre-trained | True: BCC | Predicted: BCC



(d) VGG16 self-trained | True: BCC | Predicted: BCC



(e) Average fixation time on key areas from a trained pathologist, Kimeswenger et al. [15]



(f) Human labeled location of cancer cells

Figure 11: Key regions of a histological slide as identified by attention-based MIL models and a human.

5 Discussion

The presented results show that the MIL models with attention pooling by Ilse et al. [12] are able to outperform current state-of-the-art CNN architectures significantly in a binary classification task with high resolution medical images. All attention-based models have a noticeable higher AUC, F1 Score, Accuracy and Precision than the baseline models. Not only are the attention-based MIL approaches better regarding the performance metrics, Figure 12 and Figure 13 also show that they are much faster to train. Attention-models reached their peak AUC on the validation set approximately after 15-20 epochs of training, while the baseline models needed much longer to reach their top performance. The faster convergence in combination with the low dimensional inputs, made the whole training phase (number of epochs set in the beginning) of the attention-based models on average four times faster compared to the baseline models. Figure 12 shows that the number of epochs for the attention models can also be drastically reduced while still retaining the best possible performance of the model. This increase in speed does not take into account the time needed to divide the original whole slide images into patches and compute the low dimensional representations. The time needed to execute all preprocessing steps is made up by the extremely fast convergence and the overall good performance of the attention models.

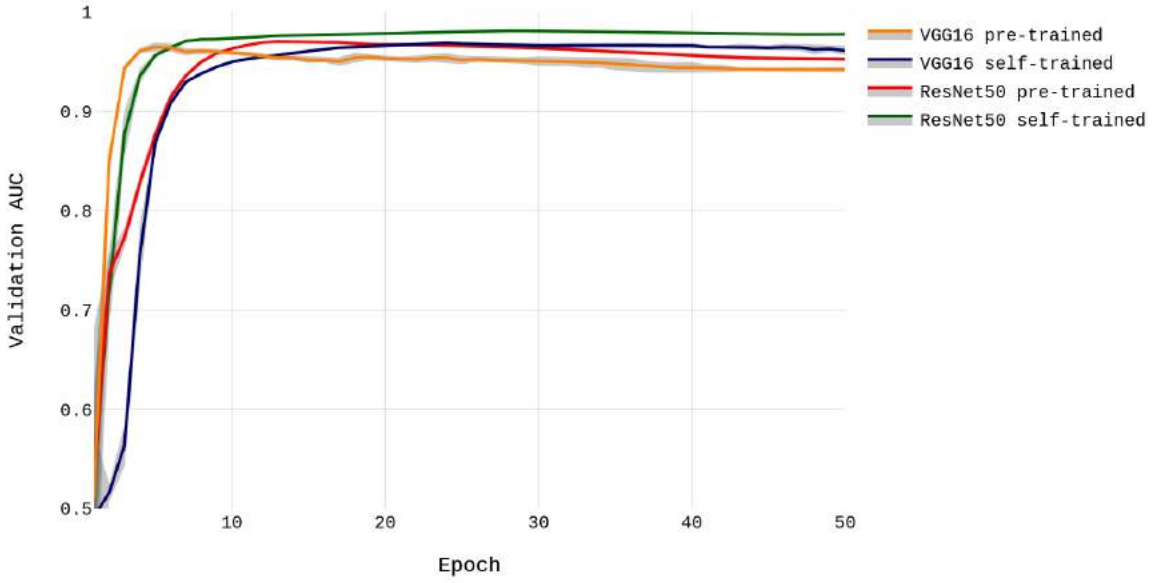


Figure 12: AUC on the validation set of the attention-based models during training.

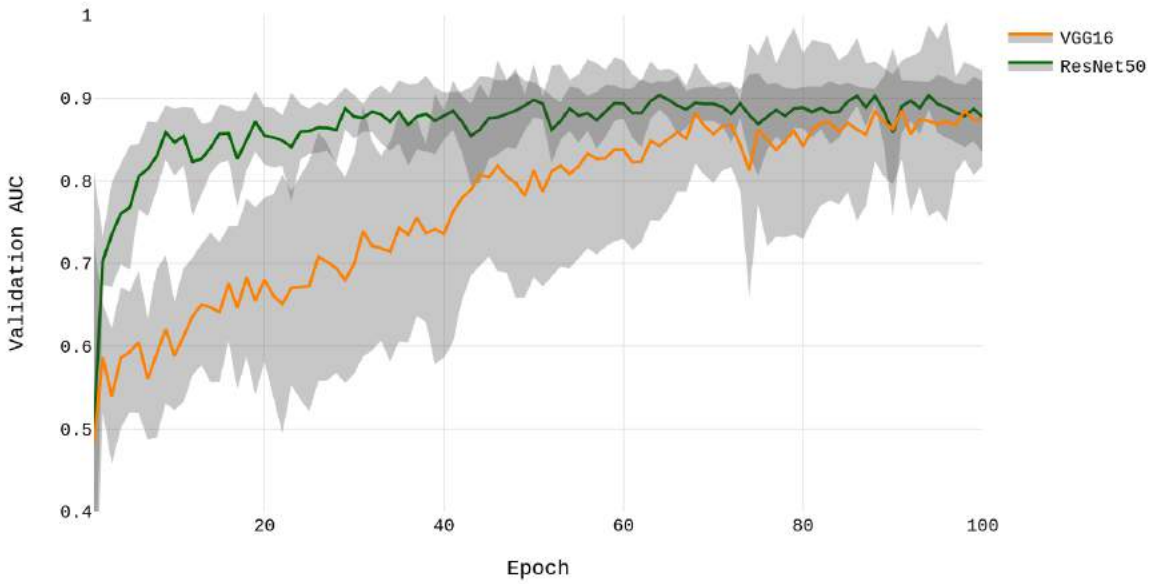


Figure 13: AUC on the validation set of the baseline models during training.

Table 7 shows that the self-trained embeddings perform significantly better than the pre-trained ones, however, this is not the case for both models. The attention-based MIL model with the self-trained ResNet50 embeddings performed better with pre-trained embeddings and vice versa when the VGG16 is used as an feature extractor. This performance of the VGG16 based attention-based model is surprising given the fact that the initial weights for this and all other models were trained using the ImageNet data set, which does not include any medical images. The different outcomes of the ResNet50 and VGG16 attention models also makes it difficult to come to a final conclusion about the better embeddings choice. I would suggest further experiments with more CNN feature extractors to find out which implementations calculate satisfying embeddings with the initial weights and which ones can really benefit from the additional training. Overall I would argue that it is sufficient to use the pre-trained embeddings, this way the training of a baseline model with the available data can be omitted and time can be saved while keeping the performance on a comparable high level.

As Table 6 shows, there is no difference between ResNet50 and VGG16 based low dimensional embeddings when comparing the combined results. However, there is a difference when the two embedding types, self- and pre-trained, are compared from each model. I would suggest that the CNN used as a feature extractor is chosen based on the type of the desired embedding. If there is no time for self-training the feature extractor, then the VGG16 should be

chosen as it seems that the pre-trained weights in this model are well suited for the calculation of low dimensional embeddings. The ResNet50 is better suited for feature extraction if there is enough time to self-train the model. Further experiments could show that other CNNs are even better suited for the task of calculating embeddings from histological slides.

The attention-based MIL approaches showed a better performance in all relevant metrics and also very consistent behaviour during all ten reruns. The baseline models on the other hand showed extreme fluctuations. Despite the unstable performance during the training the baseline VGG16 still outperformed the baseline ResNet50 for almost all metrics. Both baseline models seem to have troubles to learn the difference between the two classes 'BCC' and 'no BCC' as both achieve an accuracy of about 70%, which is the portion of the data set labeled with 'BCC'. The low Precision, paired with high Recall are further indicators for this. Table 4 then shows the numerous false positives for the baseline models. While the attention-based models only produce 2 - 3 false positives the baseline models falsely predict the class label 'BCC' more than 30 times in a test set with 132 examples. Both baseline models were not able to correctly identify negative instances in the test set and therefore were only able to correctly identify less than 5 of the 35 images with no BCC in the test set. The attention-based models on the other hand were able to classify almost all negative instances correctly. These kinds of miss-classifications could lead to unnecessary treatments or other forms of complications for patients due to wrong diagnosis.

A further advantage of the attention-based MIL models is their ability to mark key regions of an image, as seen in Figure 11. This is especially useful on histological slides, where each cell individually could contain BCC or not. While all attention models are able to predict the right class label for this example image, the two best performing attention models are also able to give the best prediction on where the BCC cells are located. The regions with high attention weights overlap with the human labeled regions of interest. This makes this visualization well suited for computer-aided diagnosis, as it indicates potential high risk regions for a human to further look into. In the current state it is necessary for a trained pathologist to examine each histological slide individually, in the future it could be possible that a neural network can help to speed up and make the diagnosis more reliable, even if no trained pathologist is available to examine the histological slides.

6 Conclusion

In this thesis I compared the proposed attention-based MIL pooling method from Ilse et al. [12] with standard CNN architectures by classifying high resolution medical images. While the preprocessing and feature extraction steps of the attention-based models can be a time consuming process, it proofed to be worth the extra effort as these models outperformed the baseline models significantly in classifying histological slides with BCC. Not only are the attention-models faster to train, once the features are extracted, they also have the ability to show key regions of an image, which triggered the bag label in this MIL approach. This can be especially useful in the case of finding cancerous cells in such a setting. Because the results did not show that one CNN is significantly better at calculating the low dimensional embeddings than the other, I would suggest further experiments to see which CNN architectures are capable of producing the best features.

List of Figures

List of Tables

References

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).

[2] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. “Multiple Instance Learning: A Survey of Problem Characteristics and Applications”. In: *Pattern Recognition* 77 (Dec. 2016), pp. 329–353.

[3] A. Neil Crowson. “Basal cell carcinoma: Biology, morphology and clinical implications”. In: *Modern Pathology* 19.SUPPL. 2 (Feb. 2006), S127–S147.

[4] Jia Deng, Wei Dong, Richard Socher, LI-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: (2009), pp. 248–255.

[5] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. “Solving the multiple instance problem with axis-parallel rectangles”. In: *Artificial Intelligence* 89.1-2 (Jan. 1997), pp. 31–71.

[6] Editors of Encyclopaedia Britannica. *Histology - Encyclopaedia Britannica*. URL: https://www.britannica.com/science/histology (visited on July 31, 2020).

[7] SE Radiation Oncology Group. *What is Basal Cell Carcinoma?* URL: https://treatcancer.com/basal-cell-carcinoma-bcc/ (visited on Aug. 26, 2020).

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem (Dec. 2016), pp. 770–778.

[9] Anne Marie Helmenstine. *Defining Histology and How It's Used*. URL: https://www.thoughtco.com/histology-definition-and-introduction-4150176 (visited on July 31, 2020).

[10] *Histology fixatives - CellBiology*. URL: https://cellbiology.med.unsw.edu.au/cellbiology/index.php/Histology_fixatives (visited on Aug. 1, 2020).

[11] *How histological slides are produced - Open University*. URL: https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=65372&section=1.2 (visited on July 16, 2020).

[12] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. "Attention-based Deep Multiple Instance Learning". In: *35th International Conference on Machine Learning, ICML 2018* 5 (Feb. 2018), pp. 3376–3391.

[13] Julie Karen and Ronald Moy. *Basal Cell Carcinoma - The Skin Cancer Foundation*. 2019. URL: https://www.skincancer.org/skin-cancer-information/basal-cell-carcinoma/ (visited on July 16, 2020).

[14] Susanne Kimeswenger, Elisabeth Rumetshofer, Markus Hofmarcher, Philipp Tschandl, Harald Kittler, Sepp Hochreiter, Wolfram Hötzenecker, and Günter Klambauer. "Detecting cutaneous basal cell carcinomas in ultra-high resolution and weakly labelled histopathological images". In: (Nov. 2019).

[15] Susanne Kimeswenger, Philipp Tschandl, Petar Noack, Markus Hofmarcher, Elisabeth Rumetshofer, Harald Kindermann, Rene Silye, Emmanuella Guenova, Sepp Hochreiter, Günter Klambauer, and Wolfram Hötzenecker. "Comparison of histological pattern recognition of basal cell carcinomas by convolutional neural networks and pathologists". unpublished.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[17] Julien Lanoue and Gary Goldenberg. "Basal cell carcinoma: A comprehensive review of existing and emerging nonsurgical therapies". In: *Journal of Clinical and Aesthetic Dermatology* 9.5 (May 2016), pp. 26–36.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2323.

[19] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". In: (Nov. 2015).

[20] Mustafa Umit Oner, Jared Marc Song Kye-Jet, Hwee Kuan Lee, and Wing-Kin Sung. "Studying The Effect of MIL Pooling Filters on MIL Tasks". In: (June 2020).

[21] N. Parry. *How Histology Slides are Prepared*. URL: https://bitesizebio.com/13398/how-histology-slides-are-prepared/ (visited on July 16, 2020).

[22] K. Parthy. *Convolutional Neural Networks for Visual Recognition*. 2018. URL: https://cs231n.github.io/convolutional-networks/#overview%20http://cs231n.github.io/neural-networks-3/ (visited on July 7, 2020).

[23] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: (Dec. 2019).

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Dec. 2015), pp. 211–252.

[25] Sumit Saham. *A Comprehensive Guide to Convolutional Neural Networks*. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (visited on Aug. 30, 2020).

[26] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Sept. 2015).

[27] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. "Striving for Simplicity: The All Convolutional Net". In: *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings* (Dec. 2014).

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems* 4 (Jan. 2014), pp. 3104–3112.

[29] *Treatment for Advanced Basal Cell Carcinoma*. URL: https://www.clinicaltrialsarena.com/projects/erivedge-treatment-for-advanced-basal-cell-carcinoma/ (visited on Sept. 3, 2020).

[30] Petar Veličković. *2D Convolution*. URL: https://github.com/PetarV-/TikZ/tree/master/2D%20Convolution (visited on Aug. 28, 2020).

[31] Lilian Weng. *Attention, Attention!* 2018. URL: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html (visited on Aug. 6, 2020).

[32] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. "Deep multiple instance learning for image classification and auto-annotation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June (2015), pp. 3460–3469.

[33] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. "Show, attend and tell: Neural image caption generation with visual attention". In: *32nd International Conference on Machine Learning, ICML 2015* 3 (2015), pp. 2048–2057.