Bone age assessment for Asian children using Convolutional Neural Networks

Shankara van de Ven

11000791

Bachelor thesis

Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam

Faculty of Science

Science Park 904

1098 XH Amsterdam

Supervisor

Dr. S van Splunter

Queen Mary Hospital

102 Pok Fu Lam Road

Hong Kong

June 29th, 2018

Abstract

Skeletal bone age assessment is performed to evaluate whether the bone age is advanced or delayed compared to the patients chronological age. A delayed or advanced bone age can indicate growth disorders. It is generally performed by using either the Greulich and Pyle method or the Tanner-Whitehouse method. However, inter- and intra-observer differences occur that could be resolved by developing an automatic system. Convolutional neural networks have been implemented with great success in the computer vision field. This thesis investigates different ways of building a simple network that performs bone age assessments reasonably well and can be run on low-end hardware. Three experiments are done focusing on adding sex data, regularisation and pre-training. It can be concluded that adding sex data, applying image generation and pre-training the network increases the networks performance.

Acknowledgements

I would like to thank Dr. Sander van Splunter for the incredible support I received during the course of this thesis. His immense enthusiasm helped me to stay focused, motivated and excited during these last three months. Thank you so much for all the hours revising my work and video calling every Friday to catch up on the progress made that week.

I also want to express my gratitude to Dr. Benjamin Fang from Queen Mary Hospital. Without you I would not have had the amazing experience of doing my thesis in Hong Kong. Thank you for going out of your way to make all of this possible and putting so much energy into this random guy from the Netherlands. It was a pleasure working together and sharing our passion for AI and medicine.

Contents

1 Introduction

In recent years Artificial Intelligence (AI) has been applied more and more within the medical field which has led to some novel insights. Noticeable examples of these are, e.g., IBM Watson [4] identifying cancer and heart diseases, telerobotic surgery performed with the da Vinci surgical system [2] and mental health chatbots such as Woebot [6].

This thesis focuses on applying AI in the radiology department with the goal being the assessment of skeletal bone age from X-ray scans. Bone age assessment is performed to evaluate whether the bone age is advanced or delayed compared to the patients age. A delayed or advanced bone age can indicate pediatric disorders [18]. In that case serial measurements can help to assess if a treatment is succeeding [13].

A strong reason for automatic bone age assessment are the inter- and intra-observer differences occurring in bone age assessment. Inter-observer differences can range from 0.07 to 1.25 years [3]. These differences between doctors can be resolved by developing an automatic system that does all the assessments. Another reason is the time/cost efficiency of an automated system in contrast to needing individual assessments from a radiologist. There have been multiple approaches in the past years for the automation of bone age assessment. A thorough review can be found in [12]. The earlier methods extract the regions of interest of the scan. After this, different automatic measurements are applied to determine the bone age, one of them is the CASAS system [21], a computerised version of the Tanner Whitehouse method described in Section 2.2.2.

The hardware available for this thesis is not capable of training the type of large sized networks currently used for bone age assessment. Therefore the goal of this thesis is to explore how a smaller network can still predict bone age reasonably well. This is done by applying a downsized version of the chosen network, BoNet [19], in different ways and report the resulting predictive power. The research is conducted in Hong Kong, therefore the scans are mostly from Asian patients. The question that this thesis tries to answer is: "How can a

smaller network predict bone age reasonably well for Asian children?". Multiple techniques are expected to have a positive effect on the performance of the network. The three different experiments done to examine these techniques are the following:

1. What is the effect of adding Male/Female data as an extra input on the accuracy of the network?

2. How do image generation and dropout layers effect the regularisation and subsequently the accuracy of the network?

3. Does training the network on chronological age and then refining on bone age result in a higher precision than training purely on bone age?

This thesis is structured in the following order: In the next section the required literature is reviewed, for the topics of convolutional neural networks and bone age assessment. Section 3 discusses the approach, including the outline of the data, the processing of this data, and a description of the network. In Section 4 the experiments and results are covered that answer the three questions stated above. Section 5 discusses the results and concludes this thesis.

2 Literature Review

Firstly this section gives a theoretical foundation of convolutional neural networks (CNN's). Secondly two classic bone age assessment methods: Greulich & Pyle and Tanner Whitehouse are described. The following information about CNN's stems from the online Stanford course CS231n [24] and the online course about CNN's by deeplearning.ai on Coursera [15]. This thesis only covers the basics of CNN's and does not go into detail on some specific techniques like batch normalization which can deal with covariance shifts.

2.1 Convolutional Neural Networks

CNN's are deep learning networks commonly applied in computer vision. These networks have been implemented with great success in the last couple of years. One success for convolutional neural networks was in 2012 when AlexNet had an error rate of 15.3% on the ILSVRC-2012 contest compared to the second best entry which had a 26.2% error [10]. In the years thereafter results improved until in February 2015 a convolutional neural network from researchers at Microsoft surpassed human performance. Human performance is estimated on a 5.1% error [17] while the Microsoft Research network achieved 4.94% error on the ILSVRC-2012 data [8].

2.1.1 A convolutional layer

The basics how a CNN works are the same as a normal deep learning network: there are multiple layers with multiple weights that can be trained. However, a CNN distinguishes itself from a normal deep network in a few ways. When your task is to classify an RGB image of a 1000 by a 1000 pixels you have 1000 x 1000 x 3 = 3.000.000 input variables. Instead of learning the weights for all separate variables a CNN uses convolutional operations. In case of a 2D image a convolutional operation can be seen as a flashlight shining on part of the image, for example a region of 7 by 7 pixels. This flashlight is called a filter and the region is called the receptive field. The filter consists of an array of numbers that are also called weights and can be trained by the network. In order for this filter to work it needs the same depth as the input, which in this case was a RGB image. So the dimensions of this filter are 7 x 7 x 3. This reduces the number

of weights greatly compared to a normal neural network. Instead of 3.000.000 weights, the only weights that are trained in a CNN are the weights from the filters. These filters will convolve over the image and multiply their values with the corresponding values of the image, sum them and produce one output value for each position in the image. In a 32 x 32 image there are 26 x 26 different positions where the 7 x 7 convolution is applied (see Section 2.1.3). In order to learn more features from an image multiple filters are applied. Applying 48 filters like in the first layer of AlexNet [10] will result in an output with dimensions of 26 x 26 x 48. These different filters together form a convolutional layer in the CNN.

2.1.2 Filter

The filters can be seen as feature identifiers. For example one of the defining features in a set of lungs could be a slightly curved line as shown in Figure 1. A CNN can use a filter that is able to detect this line. If a part of the image matches this section it will score high with this filter where another line will score low, see Figure 2. In contrast to these simple examples of a filter, Figure 3 shows an example of how actual filters from a CNN can look.

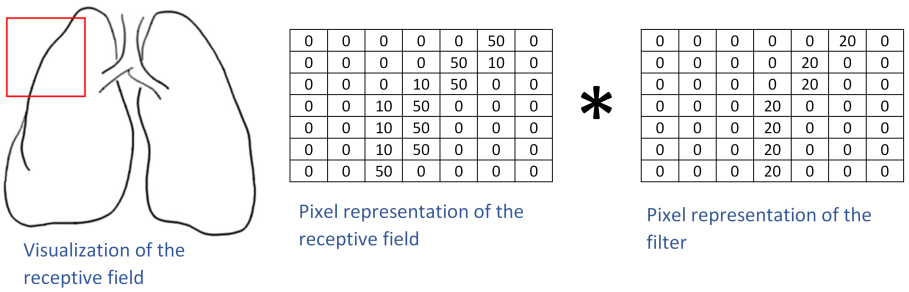

Figure 1: Filter activation resulting in a high value (20*0 + 20*50 + 20*50 + 20*50 + 20*50 + 20*50 + 20*50 = 6000)

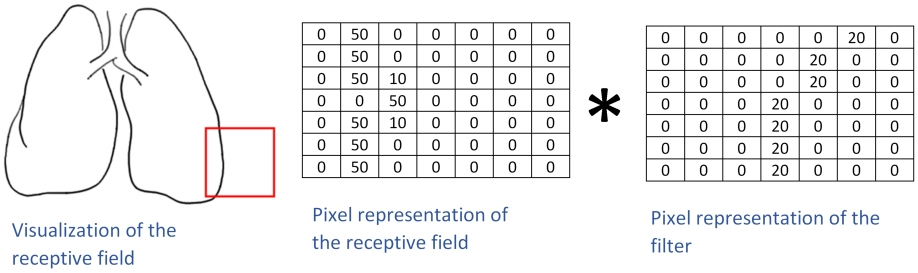

Figure 2: Filter activation resulting in zero value (20*0 + 20*0 + 20*0 + 20*0 + 20*0 + 20*0 + 20*0 = 0)

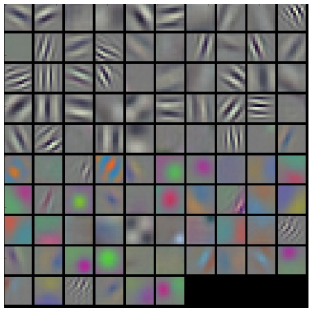

Figure 3: Different filters used in a convolutional neural network [24]

2.1.3 Stride and padding

Apart from the filter size, the stride and padding needs to be decided. The stride is the amount with which the filter shifts, the padding is the amount of pixels added on the sides of an image. In Figure 4 the normal case of a 3 x 3 filter with stride 1 can be seen, resulting in a 5 x 5 output. Important to note here is the decrease of size. As can be seen in Figure 5 a stride of 2 results in an even bigger decrease of size. Depending on the task this can be helpful for the application. If the image is of high resolution it might be good to decrease the size, which can be done by increasing the stride. If on the other hand a decrease of size is undesired a padding can be beneficial. In Figure 6 it can be seen that a padding of 1 results in the size being unchanged for a 3 x 3 filter with a stride of 1.

The formula for the output size O with an input of size W, a filter of size K, a padding of size P and a stride of S is as follows: $O = (W - K + 2P)/S + 1$.

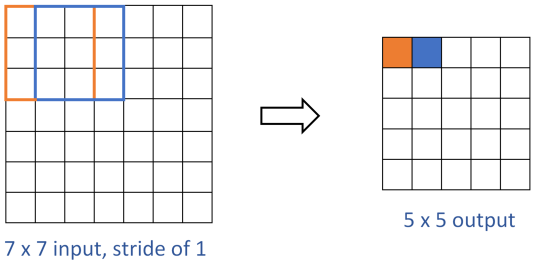

Figure 4: A 7 x 7 input and a 3 x 3 filter with a stride of 1 resulting in a 5 x 5 output

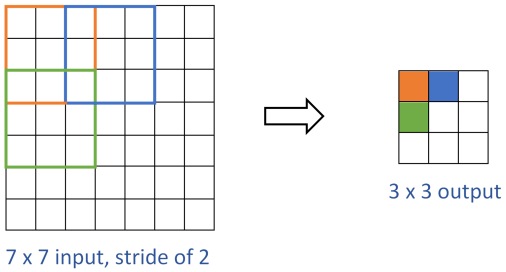

Figure 5: A 7 x 7 input and a 3 x 3 filter with a stride of 2 resulting in a 3 x 3 output

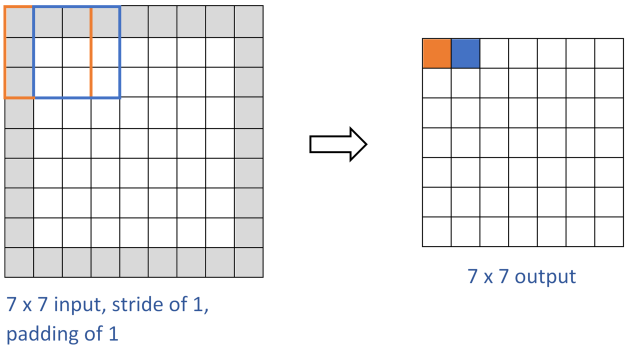

Figure 6: A 7 x 7 input with a padding of 1 and a 3 x 3 filter with a stride of 1 resulting in a 7 x 7 output

2.1.4 Depth

Whereas the width and length from a filter can be chosen, the depth of a filter in a 2D convolution is dependent on the amount of input channels. Take for example a RGB 2D image which contains 3 channels. When doing a 2D convolution over the image the filter needs a depth of 3 in order to acquire the information of every channel. Figure 7 gives an illustration of a 2D convolution applied on a 1-channel input and a 2D convolution on a 3-channel input. When multiple filters are used the output will get one dimension more. In case of a 2D input (1-channel or 3-channel, does not matter) the resulting output will have a depth that is the same as the amount of filters that has been used.

When a convolution is calculated the weights of the filter are multiplied with the values from the input. When the input has multiple channels (the third dimension from a 2D image) there will be multiple filters going over the multiple channels, see Figure 7. All the multiplications over the different channels will be add together so the resulting output will only have one channel. However, because normally multiple filters are used the output will contain multiple channels again (for each filter one).

This is what happens in convolutional layers. In pooling layers 2.1.5 the multiplications over the different channels will not be added together and each channel results in its own channel in the output. In a pooling layer only one filter is used but because each channel results in it's own channel the amount of channels does not change, only the size of the first dimensions change because of stride and padding.

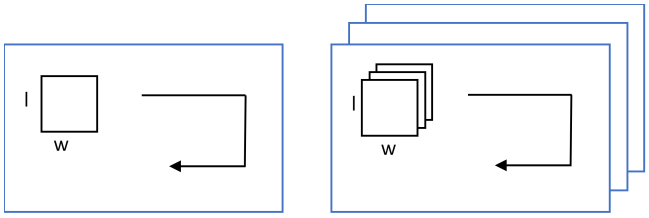

Figure 7: 2D convolutions. **Left:** a 2D convolution on a 1-channel input. **Right:** a 2D convolution on a 3-channel input.

2.1.5 ReLU, pooling, fully connected and dropout layers

Apart from convolutional layers a CNN can also include ReLU, pooling, fully connected and dropout layers. Usually after a convolutional layer follows a non-linear layer, also called an activation layer. The purpose of this layer is to introduce non-linearity to the network. A common approach these days is to use a so called ReLU layer. This layer applies the function $f(x) = max(0, x)$ to the input values, which will change the negative values to 0.

Another type of layer is the pooling layer, which is also called the downsampling layer. This downsampling is done by using a filter (normally 2 x 2), which takes a subregion of the image and performs a pooling operation on the numbers. The type of pooling that is most commonly used is maxpooling. This simply takes the maximum number of the provided input numbers and returns it as the output.

Another type of layer is the fully connected layer. The input for this layer can be whatever the preceding layer (pooling, ReLU or convolutional) is outputting. It then combines the different features from this input and returns the output as an array of numbers. This layer is commonly used in the end of the network and can produce different output classes. For example, when classifying dog pictures this layer will receive high level features like a pawn or a tail. It then can combine these and come to the conclusion that the image is either a dog or not.

Dropout is a regularisation technique introduced by Srivastava et al. [20]. In this technique randomly selected nodes are ignored during training. Without dropout the nodes in the network become too specialised to the training data. By taking out nodes randomly the nodes will not overspecialise and will represent a more generalised model of the data which results in less overfitting.

2.1.6 Training

The previous sections covered the structure of a CNN, this section discusses how the CNN works during the training process. As stated earlier the variables in

this network that will be trained are the weights in the filters. Before training the network these numbers are randomly initialised. After that the network goes through a number of training iterations. For training the network a labelled data set is needed, in this thesis the goal is to predict bone age given a bone scan. The input are scan which are labelled with the corresponding bone age. The training iterations can be separated into 4 steps: the forward pass, the loss function, the backward pass and the weight update.

In the *forward pass* a training image is sent through the network. This results in a prediction which, depending on the type of problem, can either be a classification or a continuous value.

This value then gets passed into the *loss function*, where the difference between the predicted value and the actual value gets calculated. There are many different loss functions that can be applied. The two most common loss functions for regression are the Mean Absolute Error (MAE) and Mean Squared Error (MSE).

$$MAE : \sum_{i=1}^{n} |y_i - y_i^p|$$

$$MSE : \sum_{i=1}^{n} (y_i - y_i^p)^2$$

When the loss is calculated the weights can be updated accordingly, this is called the *backward pass*. The goal here is to minimise the loss, in the end the network should give predictions as close to the actual values as possible. Mathematically this can be showed by Figure 8. There the loss function is depicted as the dependent variable and the two independent variables are two weights of the network (just for simplicity reasons, there are of course way more than two weights). The goal is to find the weight values that contributed most to the loss and update them accordingly. In order to do this, the derivative of the loss function with respect to the weights is calculated (dL/dW).

The last step is to perform a *weight update*. Here the weights are updated so that they change in the opposite direction of the gradient. The weight update is

mathematically depicted as: $w = w_{old} - \eta \frac{dL}{dW}$, where w is the new weight, w_old the old weight, η the learning rate and $\frac{dL}{dW}$ the derivative of the loss function.

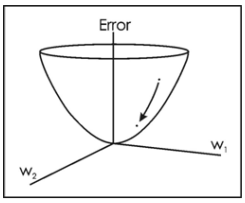

Figure 8: Visualisation of the loss function in a 3D graph

2.1.7 LeNet-5

This section explains LeNet-5, one of the very first CNN's [11]. This will give a sense of how a CNN looks and works. LeNet-5 is very basic compared to the newer networks, but this will simplify understanding the different steps. This network was built to classify handwritten digits from the MNIST dataset. The grey scale input images were 32 x 32 x 1.

In the first convolutional layer 6 filters of 5 x 5 with a stride of 1 are used. This results in a 28 x 28 x 6 output: O = (32 – 5)/1 + 1 = 28.

After that, a pooling layer follows that uses average pooling. Average pooling is more common in older networks, these days almost everyone uses max pooling. The pooling layer uses a 2 x 2 x 6 filter with a stride of 2 resulting in a 14 x 14 x 6 output: O = (28 – 2)/2 + 1 = 14.

Then a second convolutional layer with 16 filters of 5 x 5 was applied. Again with the formula this results in a 10 x 10 x 16 output. Followed by another pooling layer, which results in a 5 x 5 x 16 output. After these two convolutional layers and the two pooling layers there are 400 resulting nodes (5 x 5 x 16 = 400). These nodes are then connected to a fully connected layer of 120 nodes, which is then connected to the last fully connected layer of 84 nodes. This last layer is connected to 10 nodes where each node stands for one of the ten digits and returns the probability that that digit is given in the input.

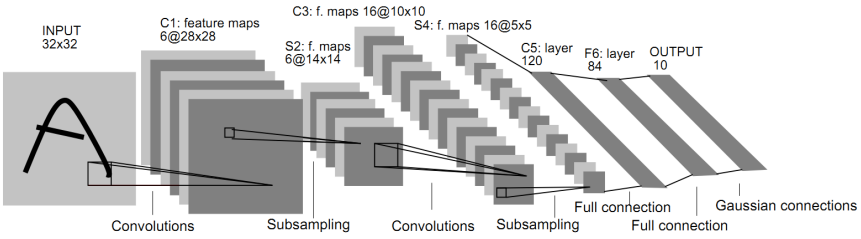

Figure 9: Architecture of LeNet-5 [11]

2.2 Bone age assessment

The medical field uses several methods to determine the bone age of a patient [14]. The two most common methods are the Greulich and Pyle (G&P) method and the Tanner–Whitehouse (TW) method [18].

2.2.1 Greulich and Pyle

This method is based on "The Radiographic Atlas of Skeletal Development of the Hand and Wrist" [7] by Dr. William Walter Greulich and Dr. Sarah Idell Pyle. This book contains reference images of the left hand from birth till 18(F)/19(M). Bone age is determined by comparing the patients hand scan with the different scans in the atlas and finding the closest match. However, the atlas is based on scans of children living in Cleveland, Ohio, United States in the years from 1931 till 1942. This results in at least two problems: it has been reported that secondary sex characteristics develop earlier nowadays than back in the 1940's [5] and there is a difference between Asian and Caucasian children in terms of bone growth [25]. The data set that is used for this thesis contains images from mainly Asian patients, whereas the atlas is mainly based on Caucasian children.

2.2.2 Tanner Whitehouse

This thesis is written in the Queen Mary Hospital in Hong Kong. Here the doctors mostly use the Tanner Whitehouse (TW) method for bone age assessment. The two different versions that are used are the TW2 and TW3 methods. Over the years the scoring system, maturity stages, skeletal ages and equations for adult height prediction have been adjusted to represent the population more

accurate. This resulted in the TW3 version. However, the rating of bone scans remained the same and results in the same maturity score (this score is just interpreted different) [16]. In the Queen Mary Hospital doctors use the book [23] but use the scoring tables from the TW2 method.

In the TW methods there are different systems to get the maturity score, in the Queen Mary Hospital the RUS method is used. RUS stands for radius-ulna-short bones, in this method 13 long and short bones are evaluated. This is done by looking at the radius, the ulna and the short bones of the first, third and fifth fingers [18].

Figure 10: Example of a scan used for bone age assessment showing the different hand bones

3 Approach

This section describes the initialisation of the experimental setup. Firstly a description of the data is given. Secondly the distribution of bone age of this data is given. Thirdly an elaboration of the data preprocessing, splitting of the data and image generation conducted is given. Lastly the network is presented, which includes the implemented architecture, loss function and optimiser.

3.1 Data

The network is trained and tested on a data set made available by the Queen Mary Hospital. The total data set contains 13723 scans of patients between the ages of 1 to 18. The scans are in DICOM format that contain image data as well as information about the patient and the scan itself such as the date of the scan and the amount of radiation used. In this thesis the following aspects of the scans are used: the pixel data, the patients sex, the patients birth day and the day of the scan. The patients birth day, and the day of the scan are used to calculate the age of the patient at the time of the scan.

The pixel data of the scans is varying in size, usually around 2300x1700. The network requires images of the same size and since the hardware puts restrictions on the data size all images are scaled to 300x200.

The goal of this thesis is not to predict the actual age of the patient but rather the bone age. In order to do this a xlsx file is provided with the bone age assessments from Queen Mary Hospital for part of the scans. Each entry contains, among other things: the name of the scan (which can be linked to the scans provided by the hospital), the sex of the patient, the chronological age and a written bone age assessment. From the bone age assessments the bone age has to be distilled. Two examples of assessments are shown here:

BONE AGE (Left hand):According to Tanner & Whitehouse system, the RUS (TW2) score is 581, with an estimated bone age of 14.9 years.

Bone Age (left wrist)According to Tanner & Whitehouse Standard, the RUS is 205, bone age is 6.6 years old (TW2).

The bone age is extracted from these sentences by taking the part between

"bone age" and "years" and then distil the value by using regex. As can be seen the Tanner Whitehouse method is used which is explained in Section 2.2.2. The bone age distribution of these scans are shown in Table 1. The average difference between the chronological age and the bone age is 1.32 with a median of 1.07.

An issue with the bone age assessments is that certain bone age assessments are unreliable, e.g., the ones for young infants for who a foot scan is recommended instead of the usual left hand scan. To avoid unreliable assessments, these were removed manually from the data set. After manual pre-selection a total of 1616 scans between the ages of 1-18 remained.

12107 of the 13723 scans do not have a bone age assessment, only the chronological age can be calculated by taking the patients birth day and the day of the scan. 3252 of these non-assessed scans are used in Experiment 3 to pre-train the network.

3.2 Distribution

The assessed data set which contains the bone age is distributed normally from an age perspective with 702 scans from the ages 1-9 and 972 scans from the ages 10-18, as can be seen in Table 1. Experiment 3 uses the non-assessed data set. This data set is skewed: there are 668 scans from the ages 1-9 and 11325 scans from the ages 10-18. To refrain the network from over fitting on the older scans the amount of scans is restricted to a maximum of 300 scans per age.

Table 1: Distribution for the reduced data set with trusted bone age assessments

Age	**Male**	**Female**
1	4	3
2	12	11
3	13	20
4	17	30
5	42	30
6	41	50
7	38	77
8	48	50
9	47	95
10	47	153
11	48	141
12	62	96
13	56	109
14	69	41
15	68	20
16	34	13
17	18	0
18	13	0
Total:	677	939

3.3 Splitting the Data Set

The data set of 1616 bone age scans is divided into a training, validation and test set. This is done by first splitting the set into 80%/20%, with the 20% part being the test set. The remaining set is then again split into 80%/20%, with the 80% being the train set and the remaining scans being the validation set. The training set consists of 1035 scans, the validation set contains 260 scans and the test set 321 scans, which roughly converts to 64%/16%/20% for train/val/test. After this split the data sets are centred and normalised, as explained in the next section.

3.4 Preprocessing and Image Generating

3.4.1 Preprocessing

There are multiple ways of preprocessing data, with mean subtraction and normalisation being the most common when altering data for CNN's [24]. Both mean subtraction and normalisation are applied in this thesis. While training

the network the initial inputs will be multiplied by weights and biases will be added. These weights and biases will change while training to fit the data better. The rate of how fast these values change is called the learning rate. When the input values are varying a lot this learning rate might be the correct learning rate for one feature but not for a different feature. In order to avoid this the data is centred and normalised. Centring is done by subtracting the mean from each feature for the input images. The normalisation is done by dividing each feature by its standard deviation.

An important thing to note is that the centring and normalisation is done on the whole data set but the mean and standard deviation is calculated from solely the training set. The validation/test data should never contribute in any way to the training of the model (this includes the preprocessing) and can only be used to evaluate the model.

3.4.2 Image Generating

This section discusses the image generation for improving the training of the data. A challenge, when training a neural network using a limited data set, is that the network can overfit the data. Image generation deals with this by creating more images. The idea is to augment the data in such a way that a human would still predict the same output but the network has to deal with a new set of input values. In case of images there are a couple of different operations that can be used. The different operations used in this research are the following:

1. **Rotation**: random rotations are done up to 20 degrees.
2. **Horizontal shift**: random horizontal shifts are done up to 20% of the original width.
3. **Vertical shift**: random vertical shifts are done up to 20% of the original height.
4. **Shear**: random shears are done with a shear factor up to 0.2.
5. **Zoom**: random zooming is applied up to 20%.

3.5 The network

This section provides details about the implementation of the network. First of all the networks architecture is laid out. After that an explanation is given for the choice of loss function and optimiser.

3.5.1 Architecture

The architecture of the network used in this thesis is derived from BoNet used in [19]. In this paper their trained network BoNet is compared with the pre-trained networks OverFeat, GoogLeNet and OxfordNet. BoNet resulted in the smallest error of 0.79 years and is therefore chosen as a suitable network for this thesis. The original network architecture is shown in Appendix B.1. However, this network contains convolutional layers with up to 2048 filters. This amount of filters is not possible with the hardware available, so the network is reduced to what is shown in Figure 11.

As can be seen the simplified BoNet architecture contains 5 convolutional layers, each followed by a max-pooling layer. The first convolutional layer contains 96 filters of size 7 x 7. The second convolutional layer contains 128 filters of size 5 x 5. The last three convolutional layers contain 128 filters of size 3 x 3. All convolutional layers have a ReLU activation. All max-pooling layers are of size 2 x 2. After the last max-pooling layer the nodes get flattened. These are connected to a fully connected layer of 512 nodes, which is then connected to a fully connected layer of 1 node with a linear activation. In Experiment 1, see Section 4.3, Male/Female data is added, the architecture of this network can be found in Appendix B.2. In Experiment 2, see Section 4.4, dropout layers are added which results in the architecture shown in Appendix B.3.

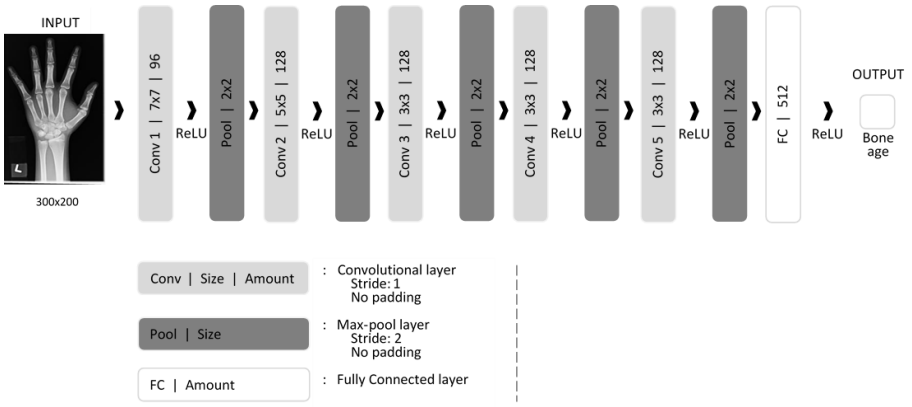

Figure 11: Architecture of the simplified BoNet network.

3.5.2 Loss-function

As explained in Section 2.1.6 two loss functions commonly used are the MSE and the MAE. Just like in [19] this thesis uses the MSE to train the network and the MAE to visualise the loss. Both of the loss functions have at least one main advantage over the other. The disadvantage of MAE is that the gradient of MAE stays the same throughout training, this means that the absolute minimum for the loss function can be missed. The gradient of MSE decreases when training goes on and the loss decreases. The advantage of MAE over MSE is when the data has outliers. Taking the square of the error of an outlier results in a tremendous error compared to taking the absolute error. However, the data used in this thesis does not contain such outliers so this advantage from MAE over MSE is not applicable here. Therefore the MSE is used as the loss function.

3.5.3 Optimiser

While training the network updates the weights, as explained in Section 2.1.6. Updating weights can be done in several different ways, the algorithms that take care of updating the weights are called optimisers. One of the newer optimisers is called Adam [9] and has proven to work well on other bone scans, see Appendix A. The chosen optimiser parameters are the standard values given by Keras: a learning rate of 0.001, a Beta 1 of 0.9 and a Beta 2 of 0.999.

4 Experiments and Results

The aim of this thesis is to find techniques that increase the performance of a simple network for bone age assessment that can be run on low-end hardware. This sections starts with an overview of the hardware and software that has been used. After that follows a subsection that explains how the training of the network is split up in epochs and batches. Following that, three experiments and their results are shown that answer the three research questions of this thesis. After each experiment the network design that performed best on the validation set is used for the next experiment. Since the test set should never contribute towards choices made for the network, the test set is only used to determine what network performed best.

4.1 Setup

The hardware used for running the network is a PC with an Intel Core i7-4770K CPU and two NVIDIA GeForce GTX 780 TI's using Windows 7 as the operating system. The software in which the network is written is Python using TensorFlow for GPU with Keras running on top. For TensorFlow to work on a GPU CUDA and cuDNN are installed. The software versions are shown in Table 2.

Table 2: Software and versions as used in this thesis

Name	Version
Python	3.5.2
TensorFlow	1.8.0
Keras	2.1.2
CUDA	9.0.176
cuDNN	7.1.3

4.2 Epochs and batches

For each of the following experiments the networks are trained for 150 epochs, a 100 batches per epoch, with 15 scans per batch. The batches could not exceed 15 scans, as the NVIDIA GeForce GTX 780 TI's have 3GB of memory, of which around 2.5GB can be freed for training. Training with small epochs results in a fluctuating accuracy from epoch to epoch. A 100 batches per epoch

gives a training cycle that does not fluctuate too much from epoch to epoch in accuracy. Each network is trained with 150 epochs. As can be seen in the training cycles of later networks, see Figure 15, after approximately 100 epochs the validation loss does not decrease anymore. In total each network is trained with $150 * 100 * 15 = 225000$ scans. Each scan in a batch is randomly drawn from the training data set.

4.3 Experiment 1: Male/Female

The first question to answer is: "What is the effect of adding Male/Female data as an extra input on the accuracy of the network?". The hypothesis here is that adding Male/Female data will increase the accuracy of the network. Because males and females have different growth rates, different conversion tables are used in the Tanner-Whitehouse method [22]. The same scan is therefore scored differently for males and females. On average the bones of a female person reach adulthood at a chronological age of 16 years. Therefore, as can be seen in Table 1, there are no female scans with a bone age over 16 years. A maturity score of 1000 results in 18.2 years for a male and 16 years for a female [22]. The maturity scores do not depend on sex, so the same scan should result in around 2 years difference in bone age assessment between males and females.

There are different ways to deal with extra input in a CNN. A simple way would be to add either a one or a zero to the image as a pixel, but there is no guaranty that the first convolutional layer will be trained in such a way that the extra input has a meaningful impact. Therefore, in this thesis an extra input stream is created that is connected to the network. This is done by creating an extra fully connected layer after the first fully connected layer, which is then combined with the extra input, resulting in the final output, as can be seen in Appendix B.2.

There are two networks used in this experiment, Network 1 has been shown in Section 3.5.1, Network 2 is shown in Appendix B.2. Network 1 has no sex input, while Network 2 has. As can be seen in Figure 12 the training accuracies of the two networks are very similar. The lowest training accuracy in Network 1

is 0.07621 and the lowest training accuracy for Network 2 is 0.1233, see Table 3. However the validation error is much higher for both networks, as can be seen in Figure 13. This means that the network does not actually learn generalised features but overfits on the training set. The validation accuracies are therefore not reliable and some regularisation should be performed first before the effect of adding sex data can be assessed. To explore such regularisation, Experiment 2 is performed.

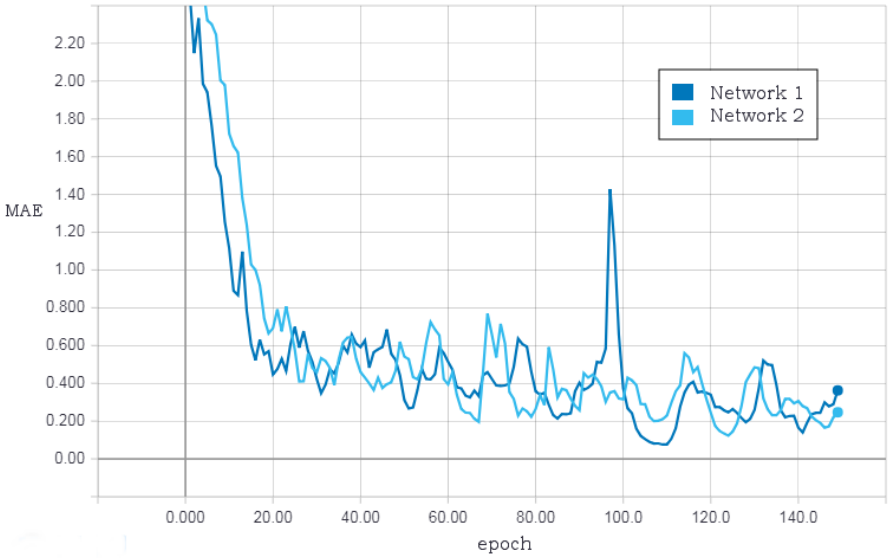

Figure 12: Training accuracy of **Network 1** and **Network 2**

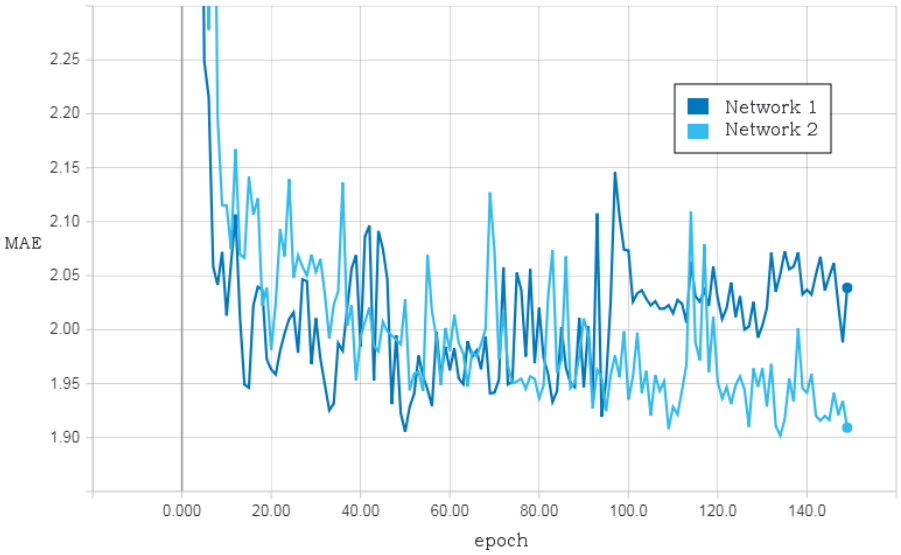

Figure 13: Validation accuracy of **Network 1** and **Network 2**

4.4 Experiment 2: Image Generating and Dropout layers

Because the network overfits, this section will focus on preventing overfitting by answering the question: "How do image generation and dropout layers effect the regularisation and subsequently the accuracy of the network?". Firstly Network 1, without sex input, is trained after applying the image generation explained in Section 3.4.2. The accuracy while training never reaches the 0.07621 in Experiment 1, the lowest error was 0.9834, but the score that matters is the error on the validation set. The lowest validation error is 1.168 compared to 1.905 in Experiment 1, as can be seen in Table 3.

When training Network 2, with sex input and image generation, the same increase in training error and decrease in validation error is witnessed. The lowest training error increases from 0.1233 to 0.8007 and the lowest validation error decreases from 1.908 to 1.003, see Table 3. Both Network 1 and Network 2 show that image generation successfully decreases the validation error.

Because the validation results now show the model does not overfit as much

as in Experiment 1, the question of Experiment 1 can be answered. As can be seen in Figure 14 and Figure 15 both the train and validation errors are lower for Network 2 than that they are for Network 1. This shows that adding data about the sex of a patient increases the accuracy of the network.

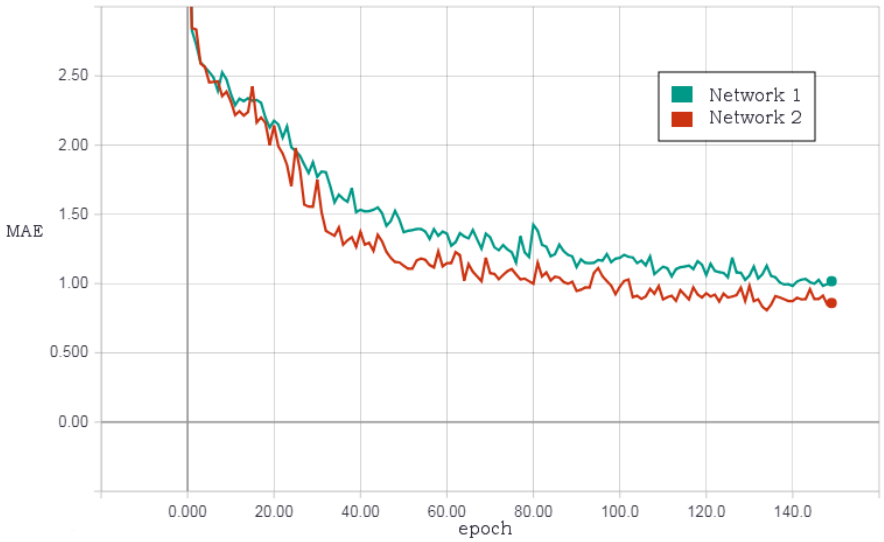

Figure 14: Training accuracy of **Network 1** and **Network 2** with image generation applied

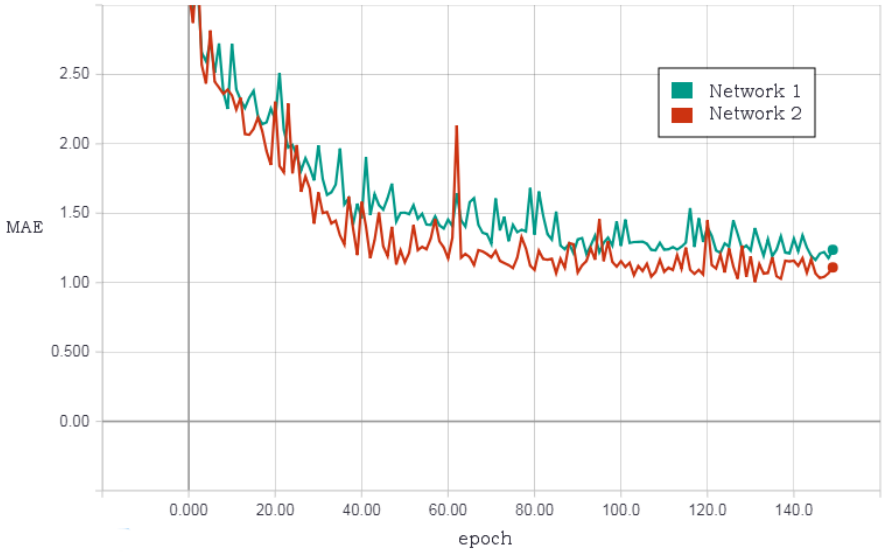

Figure 15: Validation accuracy of **Network 1** and **Network 2** with image generation applied

In the second part of this experiment Network 3, which is Network 2 with added dropout layers, is trained to determine if dropout layers increase the validation accuracy of the network. Two dropout layers are added, one after the second pooling layer and one after the fourth, as shown in Appendix B.3. The training and validation accuracies of Network 2 with image generation and Network 3 with and without image generation can be found in Appendix C.1: Figure 21 and Figure 22. The validation errors of Network 3 are higher than the validation errors of Network 2 and therefore Network 2 is used in Experiment 3.

4.5 Experiment 3: Fine tuning after training on chronological age

In this experiment a subset of 3252 scans is taken from the remaining images without a bone age assessment. These scans only have a chronological age available and are used to pretrain the network to answer the following question: "Does training the network on chronological age and then refining on bone age result in a higher precision than training purely on bone age?". Network 2 with

image generation has shown the lowest validation error so far and is therefor chosen for this experiment as well.

First the 3252 chronological age scans are split into a training set of 2600 (80%) scans and a validation set of 652 (20%) scans, there is no need for a test set. The model is then trained and only the weights from the epoch with the lowest validation error are saved. The training and validation accuracies on the chronological age data set can be found in Appendix C.2. These weights are then imported as a new model.

The new model should predict bone age instead of chronological age therefore the weights of the last layer of this new model are deleted, which is the last fully connected layer that outputs the age prediction. After that the new model is trained on the bone age data set of 1616 scans. When finetuning a network the learning rate should be decreased [24], otherwise the network starts overfitting on the bone age data set and forgets everything from the chronological age data set. The learning rate is therefore decreased from 0.001 to 0.0001.

The resulting training and validation accuracies are shown in Figure 16 and Figure 17. A clear difference in the start can be seen where both the training and validation accuracies are lower for the pre-trained network. However, in the end the training accuracies are very similar. The validation accuracy of the pre-trained network is slightly lower than the validation error of the normal trained network. This shows that the extra 3252 images might help a bit against overfitting.

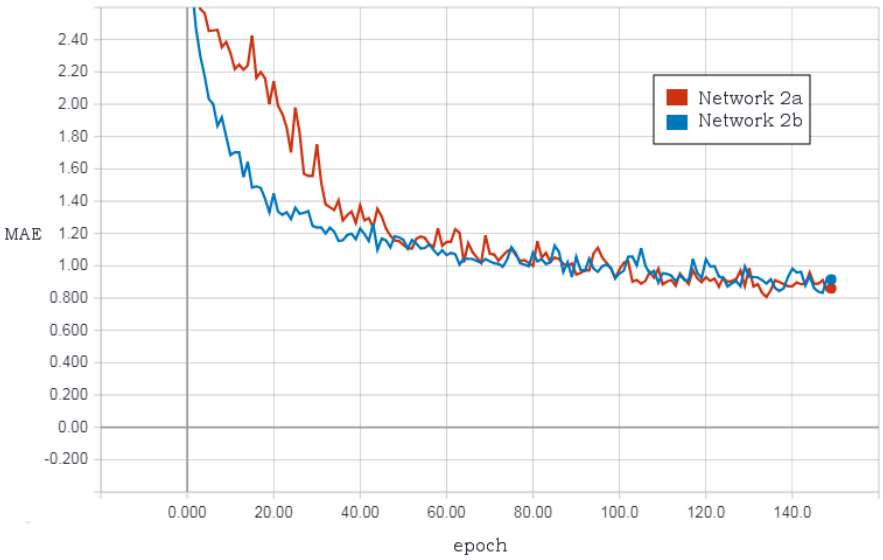

Figure 16: Training accuracy of **Network 2a** without pre-training and **Network 2b** with pre-training applied

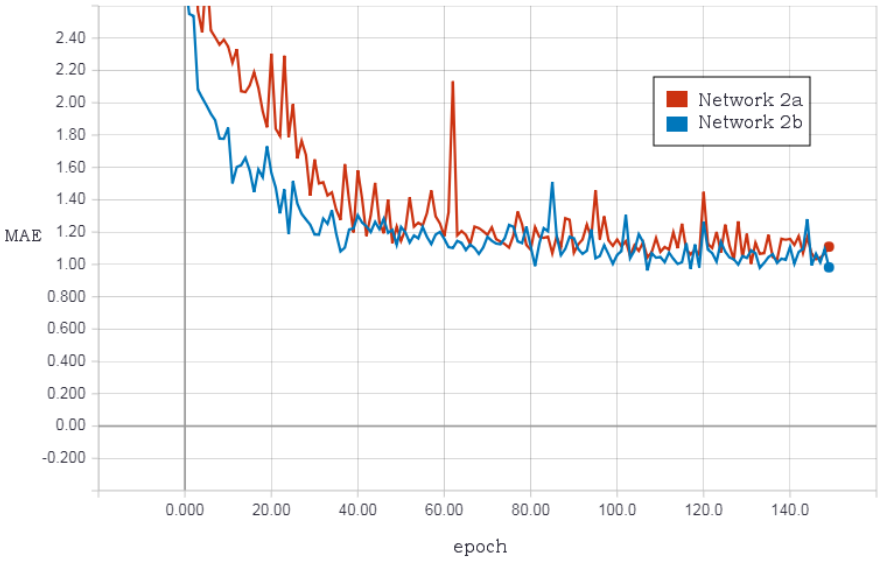

Figure 17: Validation accuracy of **Network 2a** without pre-training and **Network 2b** with pre-training applied

4.6 All results

All the results are shown in Table 3. The training and validation error showed in this table are the lowest errors encountered while training. Only the validation errors are used to determine what network to use in the experiments.

Table 3: Training, validation and test results for all experiments

Experiment	Network	Image generating	Train error	Validation error	Test error
Experiment 1	Network 1	No	0.0762	1.919	1.9697
	Network 2	No	0.1233	1.902	2.0324
Experiment 2	Network 1	Yes	0.9834	1.164	1.1729
	Network 2	Yes	0.8077	1.003	1.0686
	Network 3	No	0.4264	2.428	2.3368
	Network 3	Yes	1.002	1.246	1.2865
Experiment 3	Network 2 pre-trained	Yes	0.8330	0.9631	**1.0232**

Network 2 with image generation and pre-trained on the chronological age data set has the lowest test error. Table 4 shows how much each age group contributed to the test error. The network had the highest error when predicting scans of one-year-olds and two-year-olds.

Table 4: Test error per age group for Network 2 with image generation and pre-training.

Age	**Test error**
1	4.964
2	2.328
3	0.833
4	0.875
5	0.366
6	0.756
7	1.036
8	1.065
9	1.175
10	0.903
11	0.895
12	0.900
13	0.964
14	0.904
15	1.051
16	1.789
17	1.672
18	1.068

5 Conclusions and discussions

Multiple conclusions can be drawn from the three experiments. In the first experiment it becomes clear that without any form of regularisation the network overfits on the training data. The second experiment shows that adding Male/Female data decreases the validation error. It also becomes clear that performing image generation reduces overfitting, which results in a smaller validation error while dropout layers increase the validation error. The last experiment shows that using a pre-trained network decreases the training time. These conclusions will be discussed in-depth in the next paragraphs.

The goal of the first experiment was to answer the following question: "What is the effect of adding Male/Female data as an extra input on the accuracy of the network?". After training Network 1 without sex data and Network 2 with sex data it becomes clear that the network overfits on the training data set. Therefore, regularisation should be performed before answering this question.

In Experiment 2 two different ways of regularisation are tested when answering: "How do image generation and dropout layers effect the regularisation and subsequently the accuracy of the network?". First image generation is performed, and although it increases the training error it decreases the validation error. Therefore it can be concluded that image generation is a suitable method for regularisation. In this thesis only simple image augmentation methods are used like rotations, shifts and zooming. In bone scintigraphy, see Appendix A, it showed that occlusion significantly improved the classification capabilities of the network. When no occlusion is performed there is a risk that the network focuses too much on a certain area in the image and "forgets" about other features. A suggestion for future research is to occlude certain regions of interest, e.g., some of the short bones, the radius or the ulna so that the network learns to focus on different regions.

Now that the network does not overfit as much, the question from Experiment 1 can be answered. When bone age assessment is done manually with the Tanner Whitehouse method the same scan is given a different bone age depending on

the patients sex. A scan that depicts an adult bone age results in 16 years for females and 18.2 years for males. The expectation here is that a network without sex input would predict 17 years on average and would have an error of at least 1 year higher than a network without sex input. However, the results only show an increase of around 0.2 years when adding Male/Female data. Presumably the convolutional part of the network already learns the differences between male and female scans. To investigate this, the network could be changed into a classification network that predicts the sex of a given scan instead of the bone age.

The last part of Experiment 2 focuses on dropout layers. In [24] it is stated that dropout layers often work well for classification problems but not for regression. From Experiment 2 it can be concluded that adding dropout layers results in a higher error on the validation data set and is therefore not recommended for bone age assessment CNN's and not used in Experiment 3.

Experiment 3 answers the question: "Does training the network on chronological age and then refining on bone age result in a higher precision than training purely on bone age?". The results show no significant difference in the test accuracy. However, the big advantage of fine tuning a pre-trained network is the reduced training time. A publicly available network that is trained on a varied data set that can be fine tuned by hospitals on their own patients would be beneficial. Future research could focus on a larger scaled project that accomplishes this.

At the end of Experiment 3 the test error is analysed and it is shown that the test error is highest for one- and two-year-olds. A major reason for this is that the data set does not contain enough scans in this age group. There are 7 scans of one-year-old patients and 23 scans of two-year-olds. Dividing these scans over a training, validation and test set results in a lacking training set. A better accuracy could be achieved by increasing the number of scans in these age groups.

5.1 Final conclusion

The goal of this thesis was to answer the following question: "How can a smaller network predict bone age reasonably well for Asian children?". The results of the three conducted experiments show that a smaller network can predict bone age reasonably well by implementing the following features. First of all, sex input should be added. Secondly, image generation should be applied. Thirdly, a pre-trained network can be used to speed up the training process. Lastly, dropout layers should be avoided when creating a bone age prediction network.

References

[1] Yong-Whee Bahk. *Combined scintigraphic and radiographic diagnosis of bone and joint diseases: including gamma correction interpretation.* Springer Science & Business Media, 2012.

[2] Garth H Ballantyne and Fred Moll. The Da Vinci telerobotic surgical system: the virtual operative field and telepresence surgery. *Surgical Clinics*, 83(6):1293–1304, 2003.

[3] Matthew J Berst, Lori Dolan, Marta M Bogdanowicz, Max A Stevens, Shirley Chow, and Eric A Brandser. Effect of knowledge of chronologic age on the variability of pediatric bone age determined using the Greulich and Pyle standards. *American Journal of Roentgenology*, 176(2):507–510, 2001.

[4] Ying Chen, JD Elenee Argentinis, and Griff Weber. IBM Watson: how cognitive computing can be applied to big data challenges in life sciences research. *Clinical therapeutics*, 38(4):688–701, 2016.

[5] Susan Y Euling, Marcia E Herman-Giddens, Peter A Lee, Sherry G Selevan, Anders Juul, Thorkild IA SØrensen, Leo Dunkel, John H Himes, Grete Teilmann, and Shanna H Swan. Examination of US puberty-timing data from 1940 to 1994 for secular trends: panel findings. *Pediatrics*, 121(Supplement 3):S172–S191, 2008.

[6] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): a randomized controlled trial. *JMIR mental health*, 4(2), 2017.

[7] William Walter Greulich and S Idell Pyle. Radiographic atlas of skeletal development of the hand and wrist. *The American Journal of the Medical Sciences*, 238(3):393, 1959.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Marjan Mansourvar, Maizatul Akmar Ismail, Tutut Herawan, Ram Gopal Raj, Sameem Abdul Kareem, and Fariza Hanum Nasaruddin. Automated bone age assessment: motivation, taxonomies, and challenges. *Computational and mathematical methods in medicine*, 2013, 2013.

[13] David D Martin, Jan M Wit, Ze'ev Hochberg, Lars Sävendahl, Rick R Van Rijn, Oliver Fricke, Noël Cameron, Janina Caliebe, Thomas Hertel, Daniela Kiepe, et al. The use of bone age in clinical practice–part 1. *Hormone research in paediatrics*, 76(1):1–9, 2011.

[14] Arsalan Manzoor Mughal, Nuzhat Hassan, and Anwar Ahmed. Bone age assessment methods: A critical review. *Pakistan journal of medical sciences*, 30(1):211, 2014.

[15] Andrew Y Ng (deeplearning.ai). Convolutional neural networks, 2018. https://www.coursera.org/learn/convolutional-neural-networks/home/welcome.

[16] Ana Isabel Ortega, Francisco Haiter-Neto, Gláucia Maria Bovi Ambrosano, Frab Norberto Bóscolo, Solange Maria Almeida, and Marcia Spinelli Casanova. Comparison of TW2 and TW3 skeletal age differences in a Brazilian population. *Journal of Applied Oral Science*, 14(2):142–146, 2006.

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[18] Mari Satoh. Bone age: assessment methods and clinical applications. *Clinical Pediatric Endocrinology*, 24(4):143–152, 2015.

[19] Concetto Spampinato, Simone Palazzo, Daniela Giordano, Marco Aldinucci, and Rosalia Leonardi. Deep learning for automated skeletal bone age assessment in X-ray images. *Medical image analysis*, 36:41–51, 2017.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] James M Tanner and Robert D Gibbons. A computerized image analysis system for estimating Tanner-Whitehouse 2 bone age. *Hormone Research in Paediatrics*, 42(6):282–287, 1994.

[22] James Mourilyan Tanner. *Assessment of skeletal maturity and prediction of adult height (TW2 method)*. Academic Pr, 1983.

[23] JM Tanner, MJR Healy, H Goldstein, and N Cameron. Assessment of skeletal maturity and prediction of adult height: TW3 Method Saunders, 2001.

[24] Stanford University. Cs231n: Convolutional neural networks for visual recognition, 2018. http://cs231n.github.io/convolutional-networks/.

[25] Abdul Mueed Zafar, Naila Nadeem, Yousuf Husen, and Muhammad Nadeem Ahmad. An appraisal of Greulich-Pyle Atlas for skeletal age assessment in Pakistan. *JPMA. The Journal of the Pakistan Medical Association*, 60(7):552, 2010.

A Classification in Bone Scintigraphy

The first month of this 3-month project was spent on a different problem. This period was spent on a problem already solved by Dr. Benjamin Fang, in order to build up knowledge about CNN's and medical data. Comparing the acquired results with Dr. Fang's results gave an indication of the correctness of the chosen approach.

The goal in this problem was to classify if a bone scintigrapy image represents cancer metastates or not. Bone scintigraphy is a nuclear bone imaging technique in which the patient is injected with a radioactive substance and later scanned by gamma cameras [1]. The radioactive substance will connect to osteoblastic sites, which are spots in the body that contain a lot of osteoblats (cells that develop bone). Some of these spots indicate cancer, some of these do not, it is the objective for the network to learn to differentiate between the two.

During the development of this network it turned out that the Adam optimiser works well for bone scans, so Adam is used for bone age assessments as well. It also became clear that using Keras on top of TensorFlow brought some useful visualising capabilities and made the code look more clean and manageable. Also the different image generating tools proved to work well on classification in bone scintigraphy and were later applied in bone age assessment as well. One image generating method called occlusion worked very well in bone scintigraphy but is not explored for bone age assessment.

B Network architectures

B.1 Original BoNet

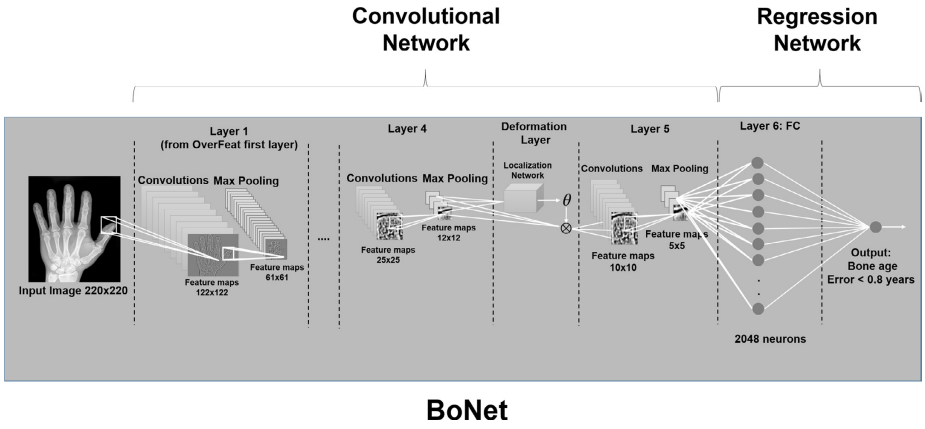

Figure 18: Architecture of the original BoNet network [19].

B.2 Network 2

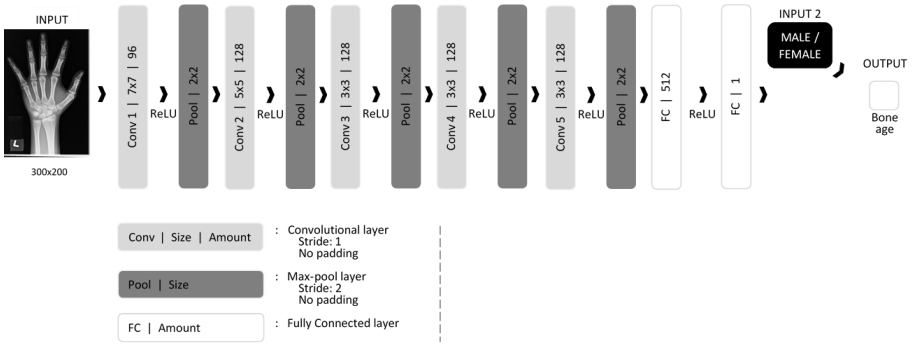

Figure 19: Architecture of Network 2, with sex input.

B.3 Network 3

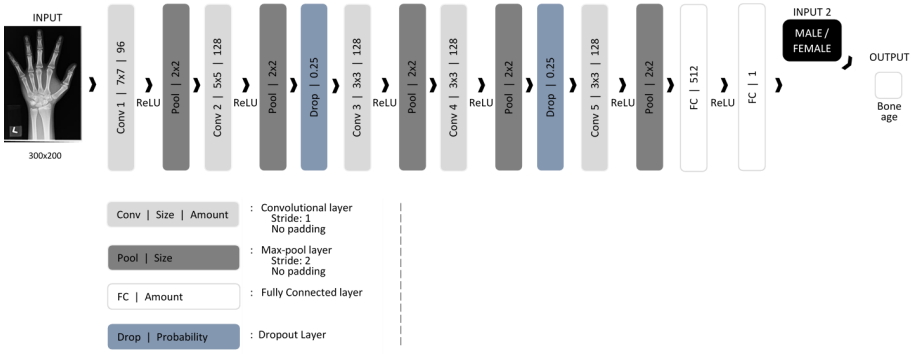

Figure 20: Architecture of Network 3, with sex input and dropout layers.

C Training and validation accuracies

C.1 Experiment 2: Dropout accuracies

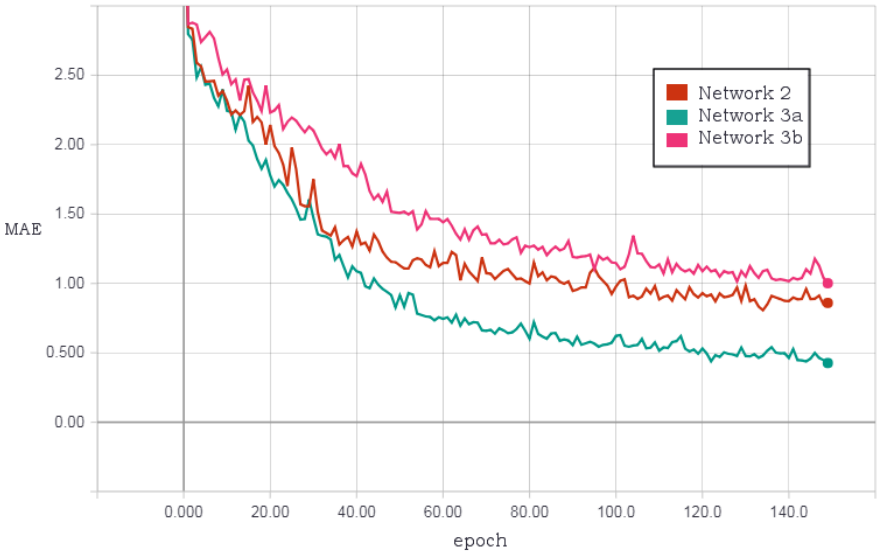

Figure 21: Training accuracy of **Network 2** with image generating, **Network 3a** without image generating and **Network 3b** with image generating

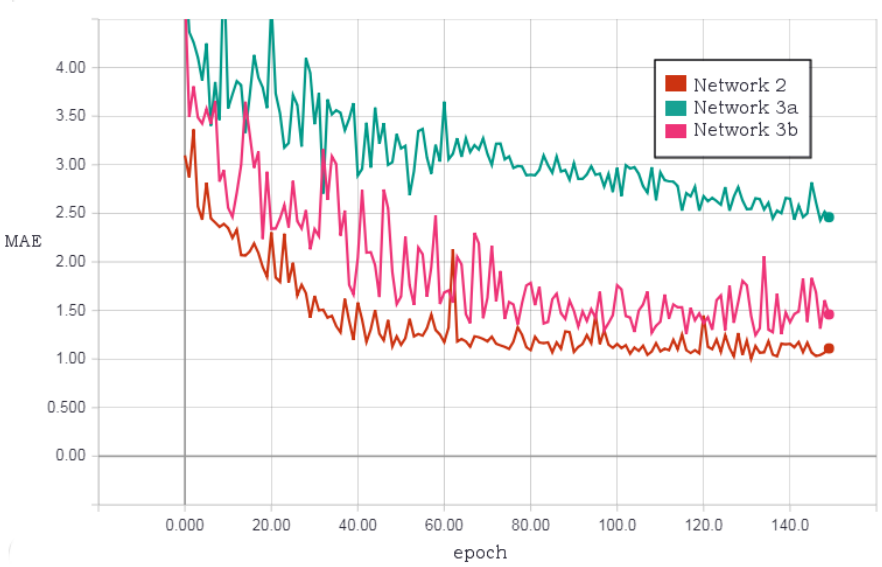

Figure 22: Validation accuracy of **Network 2** with image generating, **Network 3a** without image generating and **Network 3b** with image generating

C.2 Experiment 3: Chronological age accuracies

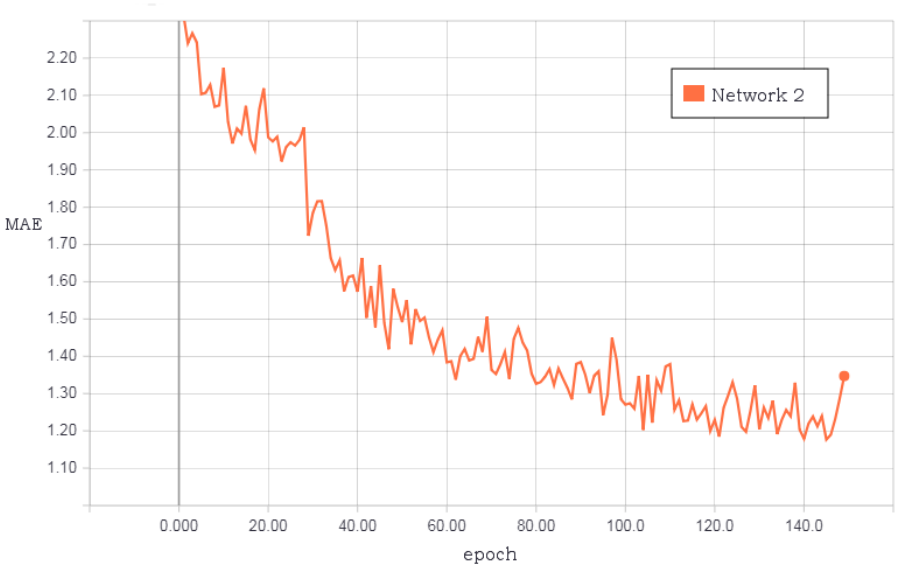

Figure 23: Training accuracy of **Network 2** with image generating on the chronological age data set

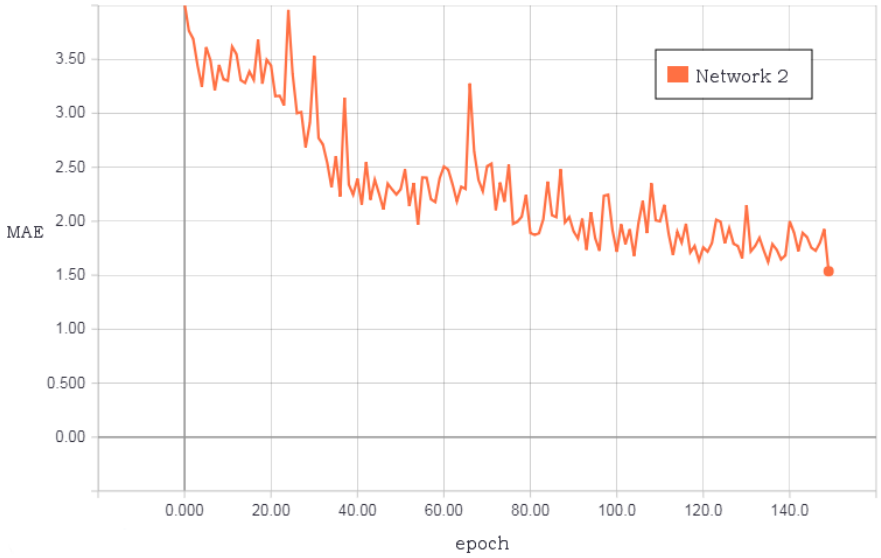

Figure 24: Validation accuracy of **Network 2** with image generating on the chronological age data set