RSNA Boneage Prediction

Joel Niklaus
University of Bern
Hochschulstrasse 6, 3012 Bern
joel.niklaus@students.unibe.ch

Abstract

Accurately determining the age of a patient based on a X-ray image is a difficult task, where many highly trained radiologists struggle. The winner of the 2017 RSNA competition achieved a mean absolute difference of 4.265 months on the test set. In this work we mainly investigate the influence of transfer learning in this task. But we also perform additional experiments providing further insight into the effectiveness of other approaches.

1. Introduction

1.1. Motivation

The Radiological Society of North America hereafter referred to as RSNA hosted a competition in 2017. The goal was to identify the age of a child based on a X-ray image of a hand. [8] Inspired by this competition Kevin Mader is hosting this dataset on kaggle in order to facilitate further research on this topic after the competition. [4])

1.2. Overview of the Idea

Our goal was to do further research on this topic. We tried to improve the existing model of Kevin Mader by investigating the following approaches:

1. Try to apply transfer learning.
    (a) Investigate the influence of the age range from the chest dataset (section 3.1.2) on the knowledge transfer
    (b) Investigate the influence of classification on the disease versus the regression on the age on the chest dataset on the knowledge transfer
    (c) Experiment with a different number of trainable layers
2. Investigate the difference in accuracy when doing classification or regression respectively
3. Investigate the difference between a model pretrained on imagenet or with random initialization respectively
4. Try to include the gender information into the model
5. Experiment with different architectures for the neural network. (InceptionNet, ResNet, VGG16, VGG19, etc.)
6. Experiment with different Hyperparameters.
7. Experiment with different settings for preprocessing of the input images.
8. Build model on top of age model to predict accuracy of age model

We also contacted him directly and he gave us valuable ideas to what to investigate further.

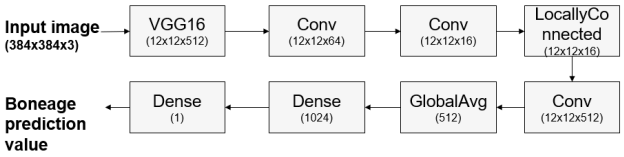
2. Prior Work

Kevin Mader published some of his work as kernels on kaggle [5]: Attention on a Pretrained VGG 16 Model (`https://www.kaggle.com/kmader/attention-on-pretrained-vgg16-for-bone-age`), MobileNet Model (`https://www.kaggle.com/kmader/mobilenet-for-bone-age`), InceptionV3Net Model (`https://www.kaggle.com/kmader/inceptionv3-for-age-gpu-hr`) and a pretrained InceptionV3Net Model (`https://www.kaggle.com/kmader/pretrained-inceptionv3-for-bone-age`).

2.1. Baseline

Because "Attention on a Pretrained VGG16" was the most popular kernel on the boneage dataset in April 2018, we decided to take his architectures (depicted in figure 1) as a baseline (included in the experiments and coded as "baseline") and tried to improve it. We learned a lot from his work, mainly regarding the use of keras and the processing of the data.

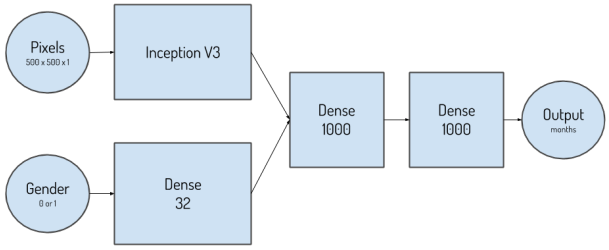
Figure 1. The ”baseline”-Architecture [5]



2.2. Winner

We further studied the work of the winner of the 2017 RSNA ML challenge. They proposed an interesting architecture (depicted in figure 2) which we included into our experiments (included in the experiments and coded as ”winner”). They also included the gender information as an additional input. With their model they scored a MAD of 0.36 years (4.265 months), which is already better than the investigated radiologists. [2]

Figure 2. The ”winner”-Architecture [2]



3. Methods

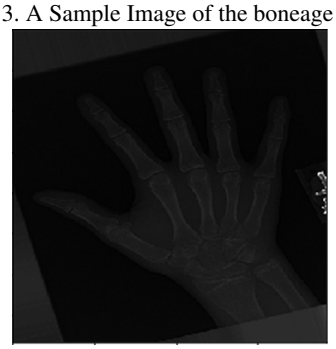
Our code and results are available on github: `https://github.com/lukaszbinden/jmcs-atml-bone-age-prediction`

3.1. Datasets

3.1.1 Boneage

The data is available on `http://rsnachallenges.cloudapp.net/competitions/4#results` and on `https://www.kaggle.com/kmader/rsna-bone-age/data`. The data contains the age of the child in months, the image of a hand X-ray and the gender information. [6, 4, 8]

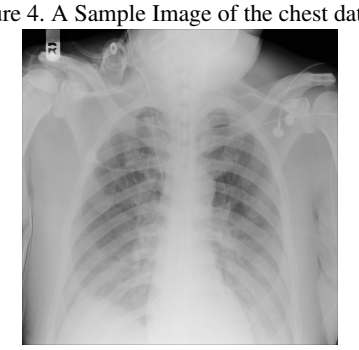
Figure 3. A Sample Image of the boneage dataset



3.1.2 Chest X-Rays

The data is available on `https://nihcc.app.box.com/v/ChestXray-NIHCC`. The ChestX-ray dataset comprises 112,120 frontal-view X-ray images of 30,805 unique patients with the text-mined fourteen disease image labels (where each image can have multi-labels), mined from the associated radiological reports using natural language processing. [1]

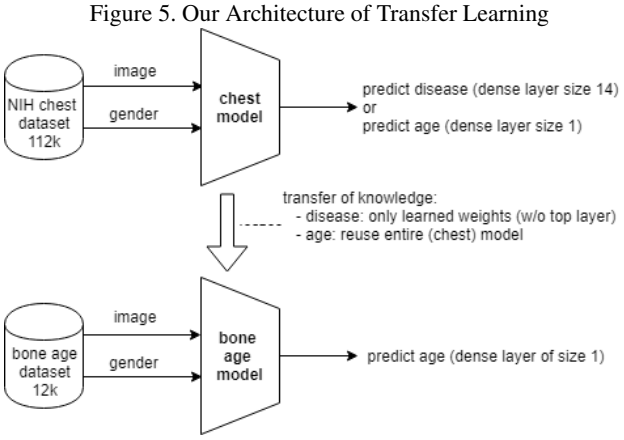
Figure 4. A Sample Image of the chest dataset



3.2. Transfer Learning

Figure 5 shows our design for transfer learning.

Figure 5. Our Architecture of Transfer Learning



4. Experiments

For our experiments we used the python library keras. ”Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.” [3] It definitely holds what it is promising. In our experience the development speed is higher than using *e.g.* pytorch. On the cluster which was provided to us by the university we used keras on top of TensorFlow. The cluster has 12 Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz and we had 4 GB Memory to our disposal. Unfurtunately there was only one GPU available leading to ResourceExhaustedErrors when training multiple models sequentially.

4.1. Setup

In all our experiments we run the training for 5 epochs because of time issues. This is by far not enough. The team of the winner model run the algorithm for 500 (!) epochs. [2] In all our experiments we used the ”winner” model as described in section 2. ”mae” is an abbreviation for mean absolute error. In the following experiments it describes the deviation in months from the ground truth. It has been used as the loss function for regression. For classification we used categorical accuracy as a loss function. The whole dataset is split into training and validation dataset with ratio 4:1. The gender information is normally excluded. We used the following data augmentation: left-right flip, random shift (20%), random rotation (20 degree), zoom (0.2). We used a batch size of 16 for the experiments, as we would run into ResourceExhaustedErrors with larger batches, because not all of the images in one batch could be allocated in 4 GB of Memory. All the networks are training from scratch except the baseline network (which applies imagenet weights). The Adam optimizer is applied for all networks and the initial learning rate is 1e-3.

4.2. Transfer Learning

4.2.1 Influence of Age Range

The age range of the boneage dataset is from 0 to 20 years. We were interested in finding out if reducing the samples from the chest dataset to the age range of the boneage dataset would influence the performance on the boneage dataset.

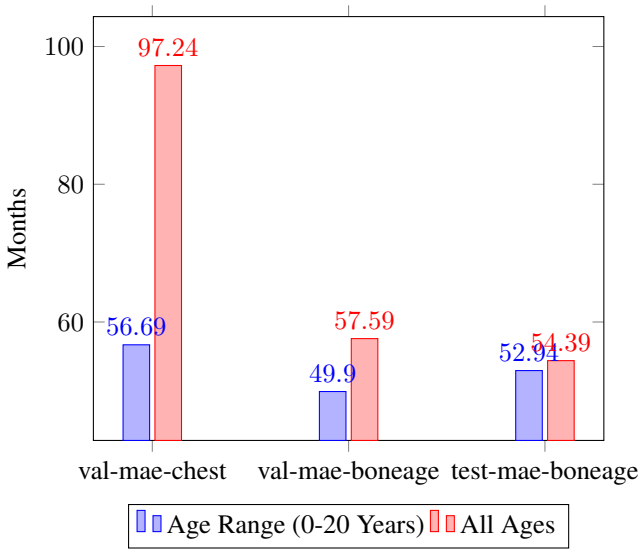


Figure 6. Influence of reducing chest dataset to age range of bonage dataset on transfer learning

Figure 6 shows that all the measured metrics are better when the chest dataset has been reduced to the boneage age range. So, it seems beneficial for the network if it only looks at samples in the required age range. When it takes the whole chest dataset, it is perhaps getting confused with images from much older people (the chest dataset contains images from people of all ages).

It has to be noted that there are 15 times more images in the entire chest dataset in comparison to when it is reduced to the boneage age range. Although there are much less data available it still performs better. Additionally it comes with the advantage that the training time is smaller.

4.2.2 Classification on Disease

Here we had to pre-process the data in order to make the classification. For every image there were 0 to 14 diseases possible. For reasons of simplicity we only used the samples which had only one disease assigned. We converted it into a sparse binary vector in order to do the classification. The model had 72% top-5 accuracy on the predicted disease. In our opinion this is an acceptable achievement for 5 epochs of training.

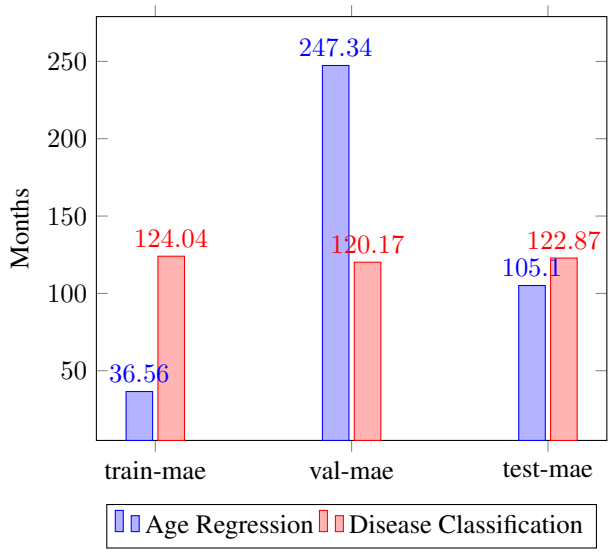


Figure 7. Influence of regression on age versus classification on disease on chest dataset on transfer learning

Figure 7 shows that training on disease classification on the chest dataset and then on age regression on the bonage dataset does not improve the accuracy on the test set. Maybe this would change had we had the possibility to train for more epochs. But our findings indicate that training directly on age regression is better than the detour over the disease classification.

4.2.3 Number of Trainable Layers

We wanted to experiment with different numbers of trainable layers of the backbone convolutional network when training on the boneage dataset after having trained all layers on the chest dataset. The more layers we set trainable the more the network can adapt to the new dataset, but the more it is also prone to overfit to the samples seen in the training set.

Unfortunately we were not able to run this experiment because it was not possible to run more than two models sequentially. Somehow the GPU memory was not cleared after a model was not used any more. This is a well known problem in keras or tensorflow respectively. But the solutions proposed in the internet do not yet solve this problem reliably.

Therefore we cannot present any results here, but we think that the optimal number of trainable layers is around two thirds of the length of the network because like that the low level features learned at the start of the network can be preserved well.

4.3. Classification versus Regression

In this experiment we investigated the difference on the performance when doing classification on the age in months versus doing regression. For classification we used top-1 and top-5 accuracy. Given our normal regression performance of 10 to 20 months we could have experimented with *e.g.* top-10 or top-20 accuracy as well. As there is an almost 5 fold increase in accuracy from top-1 (8%) to top-5 (35%) (table 1) it could well be that top-10 accuracy is around 65%. Continuing this, top-20 accuracy would be around 100%. This is pretty plausible: Given the regression performance of around 10 months on average (table 2) top-20 accuracy could cover 10 months in either direction.

Table 1. Classification versus Regression

	val-loss	val-top-1-acc	val-top-5-acc	val-mae
class. top-1	5.5084	0.0821		
class. top-5	9.4924		0.3502	
regression	18.1380			18.1380

4.4. Pretrained on ImageNet versus Random Initialization

Here we investigated the influence of the transfer learning from the chest dataset to the boneage dataset. Once we ran the experiment with the model pretrained on imagenet and once randomly initialized. We did not freeze any layers.

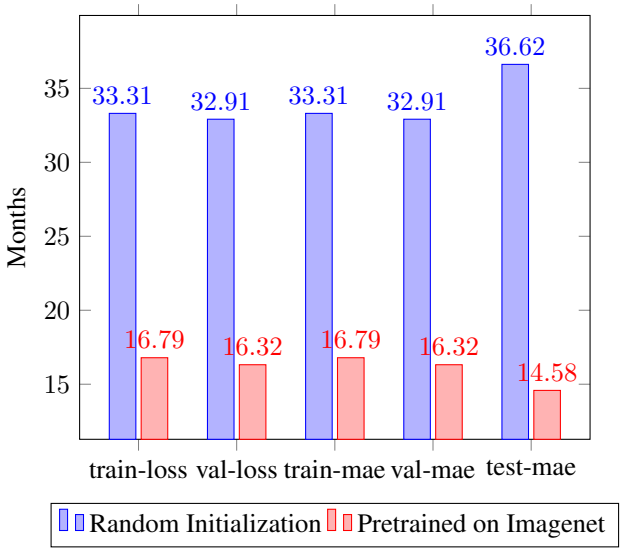


Figure 8. Influence of pretrained model versus random initialization on transfer learning

Figure 8 shows, that the pretrained model has much lower error rates than the randomly initialized one (factor 2). This strongly indicates that the network can transfer knowledge from classifying images from ImageNet to our hand X-ray age regression task.

4.5. Gender Model

We investigated if the inclusion of the gender information had any effects on the accuracy. Figure 2 shows that

gender information is coded as a boolean value and then fed into a Dense 32 layer. Then it is concatenated with the output of the InceptionV3-Net analyzing the input images.

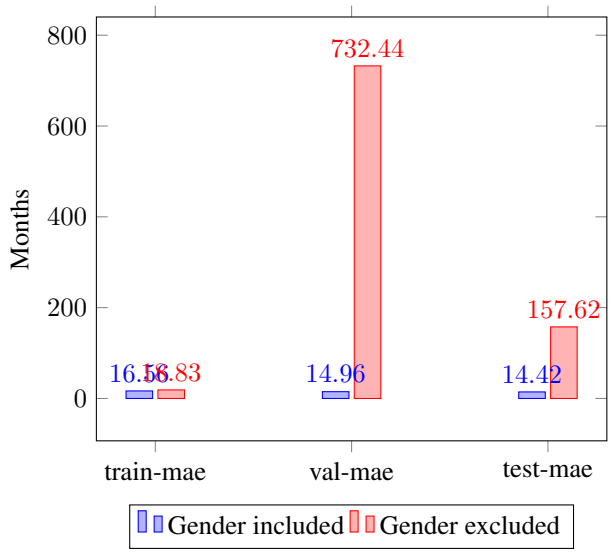


Figure 9. Influence of the inclusion of the gender information on the prediction performance

Figure 9 clearly indicates that the inclusion of the gender information is worthwhile, increasing the accuracy. The validation mean absolute error and the test mean absolute error in the case where the gender was excluded are very high. This could be due to coding errors or the small number of epochs. Figure 10 shows that the validation loss of the "winner" model is much higher than the training loss in the early training.

4.6. Different Architectures

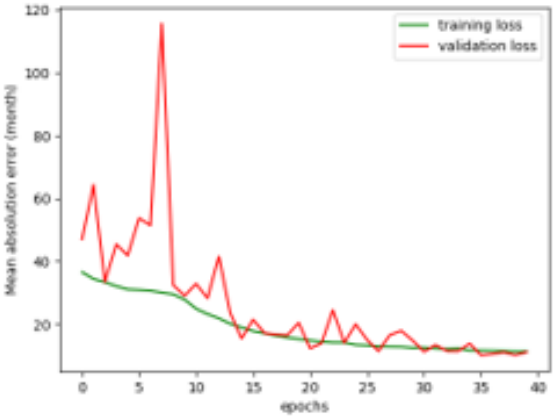
Table 2 shows that the "winner" model has indeed the best accuracy but we can see that if we replace the InceptionV3Net with a SeResNet50, the accuracy is almost the same but it requires only half the parameters. In fact the "winner" model is by far the largest model with over 123 million parameters.

Table 2. A comparison of different investigated architectures (trained for 40 epochs)

Network	Mae	Parameters
Baseline Net ("baseline")	13.48	15,279,905
16BitNet ("winner")	10.18	123,157,209
ResNet50-backbone	11.79	57,352,441
ResNetXt50-backbone	14.89	56,812,857
SeResNet50-backbone	10.51	59,825,465

Figure 10 shows the development of the training and validation loss in the course of the training. The training loss decreases smoothly and regularly. The validation loss on the other hand oscillates strongly and is very erratic in early epochs particularly.

Figure 10. The training and validation loss of the "winner" model



4.7. Different Hyperparameters

We did investigate different hyperparameters using an unstructured random search. At the end we used the best hyperparameters we encountered. This is by no means the end of all wisdom, and the search could *e.g.* be extended with a structured grid search.

4.8. Image Preprocessing

Figure 11. An image before and after applying histogram equalization

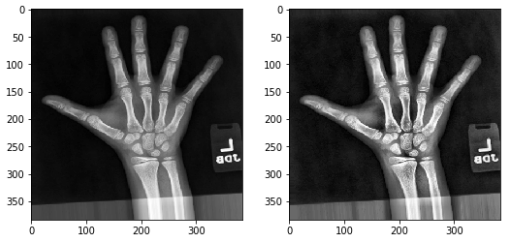
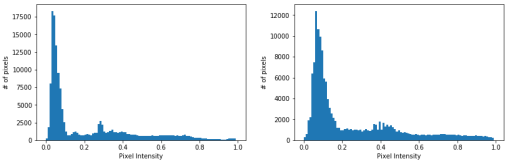


Figure 12. An image before and after applying histogram equalization



We tried many different parametrizations for the `ImageDataGenerator` class available in keras as described in 4.1. Additionally we applied histogram equalization (figures 11 and 12) by using the scikit-learn function `equalize_adapthist`. [9] Unfortunately, none of the investigated approaches yielded improved results.

4.9. Add Meta-Model to Predict Accuracy of Age-Model

The goal of this experiment was to add a model on top of the age model to predict the accuracy of the age model. So it would *e.g*. learn that the age model is less than half a year away from the real age. With the help of this model we would be able to reject the prediction of the age model if it is very insecure. So if we increase the tolerance of the meta model the coverage increases but the accuracy decreases and vice versa.

The team member who was responsible for this part quit the course one day before the deadline of this report. Therefore we cannot present any results here.

5. Discussion

We initially started out as a group of six people. But in the course of the semester more and more people quit the course. Some team members were quite inactive leading to the fact that this project was mainly developed by two people.

5.1. Learnings

We created a csv file containing only 18 entries in order to make the debugging easier. Like this it is much easier to keep the overview. But debugging on the cluster was a pain, because the editors are bad and the debugging can only happen with print statements. So for the next time we should probably download a small dataset to the local machine and debug the code locally. Once we are sure it runs locally we can push it to the cluster, make the necessary path adjustments and possibly other changes and then only perform the big experiments on the cluster. That way we probably could have saved much time. Additionally, the cluster only has 4 GB of Memory. This made it impossible to run two experiments at the same time further pejorating the problem with the low speed of development.

6. Conclusions

None of our image preprocessing attempts caused significant improvement. The use of different architectures demonstrated that performance gains are possible but it requires further investigation to draw conclusions. Likewise, we experimented with transfer learning on the chest dataset and reused the weights for age prediction on the pediatric X-rays. While we were able to show significant improvements between some sets of hyperparameters, at this point we cannot conclude whether the use of the large chest dataset towards transferring knowledge to the bone age prediction task actually does bring a notable advantage or not. This question should be investigated further as part of future work.

7. Further Work

7.1. Attention Based Approaches

Kevin Mader suggested an attention based approach to improve the accuracy. If someone were to do further research, this could be an interesting topic to further investigate.

7.2. Hyperparameter Tuning

Because of our limited resources (cluster, people, time) we were not able to investigate many different hyperparameters like *e.g*. learning rate, loss functions or optimizers. The default hyperparameter values are good for a wide spectrum of application but may not be well suited for our individual problem. So there could be much potential to increase the accuracy of the model with additional tuning of the hyperparameters. [7]

7.3. Combining Results

We experimented with different approaches and got results indicating which approaches are improving the accuracy and which ones are not. Bringing together all the improving approaches could result in a combined "supermodel" further exceeding the accuracy. Additionally it would be very interesting to train this supermodel for 500 epochs like the winner team as our 5 epochs are by far not enough.

7.4. Ensemble Methods

We tried out different models and architectures. Some may work well for some data samples and others might work well for others. Ensemble methods combine several (sometimes different) base learning models into one with a potentially increased accuracy. Further work could investigate the improvement of such methods (*e.g*. Voting, Bagging) over the base methods.

References

[1] N. C. Center. Nih chest x-rays dataset.

[2] M. Cicero and A. Bilbily. Machine learning and the future of radiology: How we won the 2017 rsna ml challenge.

[3] Keras. Keras: The python deep learning library.

[4] K. Mader. Kaggle rsna bone age.

[5] K. Mader. Kaggle rsna bone age kernels.

[6] S. H. MD. Stanford medicine bone age datasets.

[7] P. Probst, B. Bischl, and A.-L. Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms, 2018.

[8] RSNA. Rsna bone age prediction challenge.

[9] Scikit. scikit-learn: Machine learning in python.