Designing a Bone Age Estimation CNN-Model using Hyperparameter Optimization

Student: Jiawei Ma
Professors: Michele Rossi, Francesca Meneghello
Course: Human Data Analytics 2020-2021

***Abstract*—Bone age estimation is a very important medical examination based on X-ray images for detecting skeletal system development abnormalities, it is traditionally performed manually and the estimation precision depends on the expertise of the doctor in charge. Powerful modern calculators and new deep learning techniques allow the computer vision community to automate this tedious procedure using computer algorithms which outperform the human specialists.**
In this work we experimented with the dataset prepared for the Radiological society of North America (RSNA) challenge 2017 and we achieved promising final results with a deep model comparing to the state-of-the-art score, having limited computational resources. We approached the problem concentrating on hyperparameter search despite contriving complex prediction model architectures. The hyperparameter optimization applied consisted of two steps; *ablation analysis and enhancement step*. It aimed to build an efficient model that can run on a machine with limited computational power and memory.

I. Introduction

In this work we focused on solving a bone age prediction problem given a labelled dataset composed by X-ray images of the left hand, from the wrist to fingertips. It's a common knowledge that during human organism development bones of the skeleton change in size and shape, discordance with the normal bone growth indicates a serious skeletal system dysfunction. Clinicians can rely on hand bones age estimation to verify the normal skeletal system development and these estimations are, traditionally, made manually, flipping through the *Greulich and Pyle atlas* to find the most similar example every time they are presented with a bone age study [1] or other tedious assessment methods.
As the calculators are more and more powerful, they were invented some computer vision based algorithms to automatically estimate bone ages given a X-ray image. Nevertheless, an important drawback related to these algorithms was that the precision of the estimations was not satisfactory due to the quality of the images (noises or alignment etc.) and complexity in designing a good image processing pipeline. Thanks to the new *data-driven* computer vision techniques developed in *Deep Learning (DL)* and to the big image databases available, we are now able to improve the traditional approaches in medical image analysis.
To encourage innovations the Radiological society of North America (RSNA) had organized a challenge in 2017 for pediatric bone age identification. This event succeed bringing out many possible strategies and most of them were based on DL models, the results were promising and the top models outperformed even the human experts.
Regarding our work, we preferred DL approaches and we were able to reproduce closely the best results obtained by the wining model in the 2017 RSNA challenge. Specifically, in addition to present another high precision model, we will also illustrate a model designing strategy that starting from a baseline model allows us to build a *efficient and outstanding* model architecture. This is a *hyperparameter search* strategy consisting of a *ablation step* and a *model enhancement* procedure, we found it very useful also in many general situations. In literature there are some hyperparameter optimization methods such as *grid search* and *random search* but they are infeasible when dealing with complex model structure, instead, engineers often leverage on personal experience. So, what we propose here is a good alternative that takes into account model efficiency, implying that the final model may require even less computer memory than the starting model or a strictly necessary amount, this feature is desirable in many application scenarios where computational cost and memory are crucial, for instance in mobile environment.

II. Related Work

BAA is one of radiological diagnosis tasks that are particularly fitted to be automated and over the past decades engineers/scientists have been aimed to achieve it by creating machine algorithms. Very first attempts required extensive feature engineering, manual annotation and specific field knowledge to solve this problem, then these methods have still returned no satisfactory results. Nowadays, deep learning models are becoming more and more popular and thanks to recent researches in this field there are available very sophisticated model architectures. Thus, experts from various disciplines are attracted by this new tools, desiring to improve existing automatic algorithms for their daily tasks. In image analysis, deep convolutional networks are proved to be very effective and almost all modern computer vision models benefit from this basic convolution operation.
Three of the new works related to bone age prediction problems caught our attention, all leveraged novel deep learning models. The first method was proposed for Radiological Society of North America (RSNA) challenge 2017 and won the first place. The designed model was quite simple and as feature extractor it used *InceptionV3* followed by two full connected hidden layers, it surpassed other methods, some of them were also very complex which combined classic

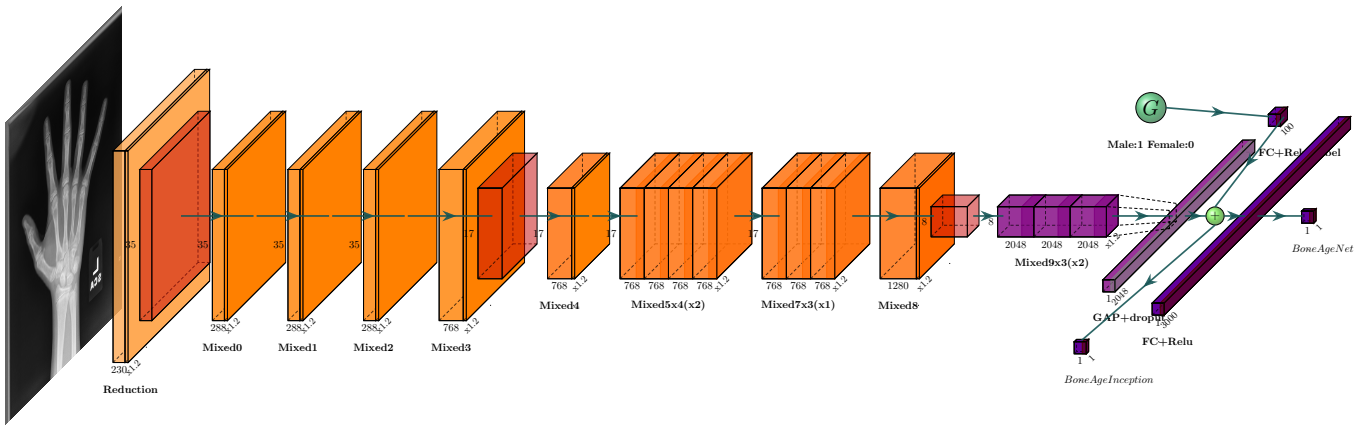

Fig. 1: The best model's structure. *BoneAgeInception* net was the initial network architecture used for hyperparameter optimization. MaxPooling layers are highlighted with dark orange color.

image preprocessing techniques and deep learning models. The approach followed in this case was to find out the best hyperparameters and to perform heavy model training instead of designing new model structure ad-hoc for that competition, obviously, this implied inefficient use of resources, in terms of computational cost and memory. Moreover, this method suffered as for many other deep learning based models the problem of *black box* prediction making interpretation of the results very difficult. Other approaches used in the challenge are described in [2]. The second method was presented in [3] and in addition to predicting ages it provided also hand pose estimation, indeed, this information was a key ingredient to perform the regression analysis for age prediction and it achieved the state of the art precision on the same dataset as in the RSNA challenge 2017. The main drawback of the method proposed was that manual annotation was need and, in general, this is an expensive and time consuming procedure. The last work was published in [4], always using RSNA dataset the authors worked out an *attention-guided* framework and it solved the problem of expansive data preprocessing required in the second approach (in practice, this data preparation was done by a separated CNN based model) and this attention based model gave more interpretable results with respect to the first model described in this section. Nevertheless, a disadvantage of the latter is that it must be trained through two stages, one is for hand localization and another is for age prediction, using two different deep models (memory consuming), so this is still a laborious working pipeline.

Regarding our work, we tried notable deep learning models such as *DenseNet*, *InceptionResNetV2* and *Xception* etc. We found the *InceptionV3* architecture is best baseline model. Firstly, we attempted to boost the initial model with several *attention mechanisms*, both *soft attention* proposed in [5] and *hard attention* illustrated in [6], the main difference between two strategies is that, first method returns an *end-to-end* trainable model with slight increase of computational cost and memory, on the contrary, the second method required either two stages training procedure or one-phase training with significant increase of training time (to our best knowledge, currently, there is not optimized framework to training such models). Nevertheless, previous models didn't give satisfactory results, in some cases they were even worse than the baseline model, we may need to adjust the general architecture for this particular problem. So, our major focus was on baseline model's hyperparameters, specifically, we did an *wise hyperparameter search* and we ended up with a model that we called *BoneAgeNet*, it has the following features:

- It almost achieved the state-of-art precision on the test dataset (mae$=5.6$ months vs mae$=4.3$ months).
- It has similar structure as the winner model in the RSNA competition 2017, but it's more compact in terms of parameters, 30% less, and it was trained using lower resolution images and through $1/5$ of the number of epochs (100 instead of 500), all these modifications imply more computational and memory efficiency. This point is important from a practical point view, since the final model can be trained using a standard machine with a single 16GB (or even less) GPU and in a reasonable time.
- It uses *GAP (Global Average Pooling)* layer, this allowed us to visualize *CAMs (Class Activation Maps)* [7] in order to verify if the predictions are made based on hand regions we expect. Moreover thanks to GAP we could keep quite low the number of parameters and this number is not sensitive to the input images resolution but only to the number of filters.

III. Processing Pipeline

Our main idea is inspired by the work related to *EfficientNet* [8], in which the authors started from a baseline model (a small version of *MobileNet*) and they strived to work out the best model scaling strategy for *efficiently* increasing the

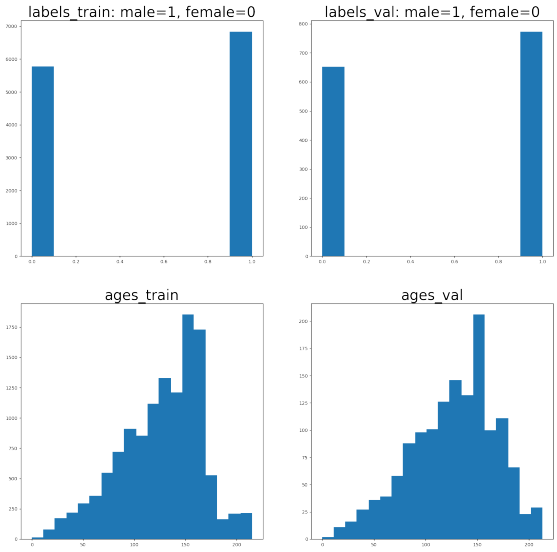

Fig. 2: Dataset: two figures on top are label distribution of the train set and validation set, the histograms on the bottom are related to age distribution of the data.

performance and they noticed that trivial enlarging of the baseline model didn't yield better results, instead, carefully balancing network depth, width, and resolution could lead to better performance. Thus, they derived a mathematical relationship between depth, width and resolution increments. Clearly, this mathematical formula it's model specific, so we need to find out the model scaling strategy that is most suitable for our problem and baseline model.

Our initial model is based on *InceptionV3* model and it's named *BoneAgeInception*, see Fig. 1. Inception models are proved to be an excellent feature extractors thanks to the very innovative idea of *Inception block* which is concatenation of different type of convolution layers and this allows the model to capture more features from image data. Then, fixed input image resolution to $300 \times 300 \times 1$, we proceeded in the backward direction for the baseline model's depth, this resembles the first step of *backward stepwise selection* in Statiscal Learning and in Deep Learning this is called *ablation analysis*, usually this operation serves as a way to understand the contribution of the components to the overall system. We investigated also the model width in a *bidirectional way* that is either increasing or decreasing it by a multiplication factor. The next step was to focus on a particular aspect, in this case on the depth, and train several models with different sizes. Once we had clear idea about how to scale the model, we decided to build the final model according to what we tested previously and additionally, we stacked another fully-connected (FC) hidden layer after the GAP because it could be useful to further increase the performance. The final model is also depicted in Fig. 1, in the same figure we highlighted the modifications we brought to the original architecture.

There was a very important decision we had to take on the nature of the problem before starting the model design, that is to define this problem as a regression or classification task, this was a reasonable doubt but observing the age distribution it seemed that a regression model was the correct answer since large number of classes would make problem harder and small number of classes would not be able to predict bone ages with good precision.

Finally, we would like to point out that the idea of ablation analysis used for model scaling can be understood recalling the basic knowledge of CNNs, specifically about the size of receptive field that corresponds to the size of kernels; the shape of receptive field applied to a image depends on the data and some are more effective than the others in that particular problem.

IV. Dataset and Data Preprocessing

***Data**:* The dataset is freely available on the website of Stanford Medicine School [1] and it's the same dataset used in the RSNA challenge 2017. Thus, it's already split into three partitions, training, validation and test set. The training set consists of 12611 images with mean patient age of 127 months and the validation set is composed of 1425 images (about 10%) with the same mean patient age as for the training set, the test set contains only 200 images with mean patient age of 132 months. The raw image resolutions are different from each others,they are around 2000×1500 pixels. Along with ground truth estimates for each image, we are provided also binary labels indicating gender of the patients; 0 indicates female and 1 indicates male. The gender information are revealed to be a relevant feature to predict the correct bone ages.

Looking at the data we observed that training-validation split was well-done meaning that both training and validation set are *balanced* ($50\% - 50\%$ class division). Moreover, also ages distribution are similar for training and validation set, we show these in Fig.2. We believe that, this check operation should be always done before building any predictive model, otherwise the results returned during training phase would be not reliable meaning that we are not ensured about the model generalization ability.

***Data preprocessing**:* The data preprocessing could be very difficult as we have discussed in the Section II, using particular techniques of feature engineering, but deep learning models usually don't require complex preprocessing techniques. For our problem we tested many *data augmentation* options such as random translation, random brightness augmentation and random flip, it turned out that, according to prediction accuracy on validation set, the only meaningful augmentation operations are:

- resize images to 300×300 grayscale images;

[1] https://stanfordmedicine.app.box.com/s/4r1zwio6z6lrzk7zw3fro7ql5mnoupcv

- feature scaling, that is dividing intensity of pixels by 255 (unit8 image encoding);
- random zooming;
- random flip;
- horizontal random flip.

We remark that feature scaling is important for the convergence of network optimization (backpropagation) and generally, feature standardization is preferable, since *batch normalization layer* is implemented and the optimization proceed smoothly, then we didn't really concern about standardization. Obviously, the choice of image resolution was due to computation resource constraints.
As the last comment, we would like to bring out the issue that within training data set there are few outliers with high or low bone age estimation, we may need to normalize the ages, but during the experiments we rather preferred to concentrated on model designing.

V. Learning Framework

Our approach is inspired by the work that introduced the EfficientNets which are novel deep learning architectures contrived starting from a baseline model and step by step wisely scaling the latter, along three axis: **width**, **depth** and **resolution**. This leads to build efficient models in terms of computational cost and number of parameters needed to be tuned. Our baseline model is InceptionV3 which is composed of different inception blocks (denoted by mixed block) preceded by a reduction block, this last block is served as resolution reduction block.
We can summarize what we did as a *three steps* method:

1) We started with a simple model consisted of InceptionV3 block and on top of it we stacked a GAP layer, a fully connected layer for the gender input and a concatenation operation followed by a final single unit output layer, we call it BoneAgeInception. We used L_2 regularization for the last output layer, that is L_2 penalization terms for the output layer's weight updates, this helps to avoid overfitting and we set the shrinkage coefficient to a common value. Regarding the loss function we defined a *Mean Square Error (MSE)* function and as the evaluation metric we followed the previous works on the dataset computing *Mean Absolute Error (MAE)*. For the model optimization we preferred the *Adam* algorithm which is in general the best one among other available *Gradient Descent* based method, then we set the batch size to 32 which is good trade-off between good gradient approximation and memory requirement. The initial learning rate was set to 10^{-2} and *stepwise learning rate decay* was implemented with minimum rate equal to 10^{-5}. So we have:

$$Loss = \frac{1}{B}\sum_{i=1}^{B}(y_{true}^{(i)} - y_{pred}^{(i)})^2 + \alpha\|\boldsymbol{w}\|_2,$$

$$MAE = \frac{1}{D}\sum_{i=1}^{D}|y_{true}^{(i)} - y_{pred}^{(i)}|,$$

where B is the batch size, D is the total number of elements in the training set and $\boldsymbol{w}$ is a vector representing the last layer's weights.
Once designed the model we trained it for few epochs (30 epochs) and then we modified the model in a backward direction by *eliminating some of the inception blocks*, one block by step, this gave us insights about which is the most relevant part of the network according to MAE value returned on the validation set but also observing the convergence speed on the training set.
After investigated in depth we moved to the model width and what we did was simply to *multiply the original number of filters of each convolutional layer by a factor* β which could be greater or smaller than 1, but not too much, depending on if we wanted to increase or decrease the width.
The reason why we tried also to downsize the model by decreasing the width was that, not always wide model performs better than less wider ones due to the complexity of loss optimization problem (highly non-convex problem), there is no guarantee for the algorithm convergence, more parameters needed to be tuned more difficult will be the optimization procedure. Same thing we can say for the depth counter part. Thus, it would be desirable to achieve the *maximum model efficiency*, that is, enhance what are useful and diminish what are useless. We recall that the dimension of the input data was fixed to $300 \times 300 \times 1$ since we have limited computational resource and keep fixed the resolution was the most convenient strategy.

2) After we identified the key inception blocks, we concentrated on the depth instead of the width (β) which is fixed at a reasonable value, because it was cheaper adapting the depth of the network and otherwise the search space would be too large. The most easy way to modify the depth was to change the number of inception blocks, those detected in the previous step and this is what we did, until we reached certain values for which the model stopped to improve. Then we obtained an inception network with an optimal number of layers.

3) In this last step we refined the architecture by setting the final β (which is greater than 1, see the next section) and we inserted another fully connected layer between the output layer and the output of the concatenation

operation, we expected this additional layer could improve the performance by learning complex relationship among the output of the GAP, so the final convolutional filters, and along with gender information. Moreover, in order to further regularize the network, we added, on top of the GAP, a *Dropout* layer with a optimal drop rate chosen observing the model convergence, namely, we discarded the values which didn't allow the model to converge and we selected the maximum possible rate. The final model is showed in Fig. 1

The experiments are all carried out using **TensorFlow-Keras** deep learning framework installed in a *quad-core* cpu machine equipped with a GPU *NVIDIA TESLA P100* 16GB of ram.

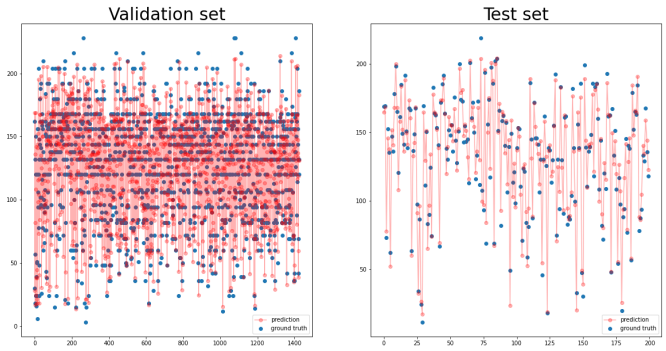

Fig. 3: Bone age predictions on validation set at the left side and test set at the right side.

VI. Results

Now we can discuss about the results we obtained following the three step described in the previous section.

1) During the first step we noticed that the inception blocks which are crucial for a good bone age prediction are the *middle and last part of the network*, specifically, they are, using the notation as in the codes, *Mixed5, Mixed7 and Mixed9*. When we eliminated these blocks the model performed much worse than the original version and when we deleted the others the performance didn't vary a lot. Related to width, we discovered that, fixed the depth unchanged, decreasing β to around 0.7 the model we didn't noted worse results with respect to the initial model, whereas increasing β we gained little more precision until at a value about 1.4, after that value the model stopped to improve and the training time increased more than linearly. So we can conclude that the baseline model was not efficient, 30% of the parameters were useless, and in order to increase the performance we need to create a not too wider model.

2) In TABLE 1 are reported MAE scores given by the different models we tested in the second step. It can be seen that at each modification of the baseline model we increased the number of the inception blocks which we considered the most relevant and we kept β a little bit greater than 1 so that the total number of trainable parameters were below 70 millions, this is the model size we could afford with the available computation and memory resources. It turned out that the best combination was with *n. Mixed5=4, n.Mixed7=3 and n.Mixed9=3* which has 25% more parameter than the starting model, further increase the layers led to worse models. Although the precision gains we had were not very high, this might due to a bad learning rate decay strategy, but we believed that at this stage we had to select the best model structure and we would refine it in the next step. However, it would make quite big difference when we evaluate the model with the test set which is much smaller than the validation set in terms of number of samples.
Lastly, we observe that while enlarge the baseline model we decrease training loss (MAE) the validation loss didn't drop with same speed and this implies that we had to strengthen the regularization coefficient or add a dropout layer as we did at the end.

3) Finally, the results demonstrate that, firstly, with the final version of the model ($\beta = 1.4$) both the training and validation loss could be reduced but most importantly, the additional fully connected layer and dropout layer gave contribution in returning the best scores, the model could generalize very well the data. The full neural network doubled the number of parameters but it improved significantly, relatively to this regression problem, the predictive performance.
Furthermore, we see that without gender information we obtained worse results meaning that this information was very crucial.

We could exploit the GAP layer to visualize CAMs produced by the last convolutional layer, we had two objectives in mind in doing this operation; on one side it could be considered as a model debug procedure allowing us to verify if predictions were made by really extracting relevant features from the images, on the other hand in this way we can understand which were the image regions the model was focus on during the inference. The Fig. 4 illustrates the results. Interestingly, and it's by some means expected, the heatmaps show that the model was able to localize the hand correctly and the highlighted regions, mostly finger regions, were some parts of the hand that a real radiologist would concentrate on doing the same task.
Another debug operation we could do is to compare predictions and ground truths and the outcomes are reported in Fig. 3. We see that the final model is trained correctly and scores computed are reliable.

VII. Conclusions

In this work we approached the bone age prediction problem using a simple hyperparameter optimization strategy, that is enhance what are useful and diminish what are not in

Name	β	n.Mixed5	n.Mixed7	n.Mixed9	Train MAE	Valid. MAE	n.params	Test MAE
Baseline	1	2	1	2	8.67	8.33	$\sim$ 28M	N/A
Modified 1	1.2	4	1	2	7.79	8.26	$\sim$ 36M	N/A
Modified 2	1.3	5	1	3	7.69	8.13	$\sim$ 55M	N/A
Modified 3	1.2	4	3	3	7.76	8.03	$\sim$ 51M	N/A
Modified 4	1	6	4	4	7.78	8.12	$\sim$ 47M	N/A
Modified 5	1.2	5	4	4	8.24	8.28	$\sim$ 65M	N/A
Modified 6	1.1	6	5	5	8.56	8.78	$\sim$ 66M	N/A
Final no FC layer	1.4	4	3	3	7.35	7.95	$\sim$ 69M	N/A
Final no FC and gender info.	1.4	4	3	3	9.12	9.44	$\sim$ 69M	N/A
Final	**1.4**	**4**	**3**	**3**	**6.85**	**7.21**	$\sim$ 78**M**	**5.6**

TABLE 1: The results related to second step described. The final models are depicted in Fig. 1.

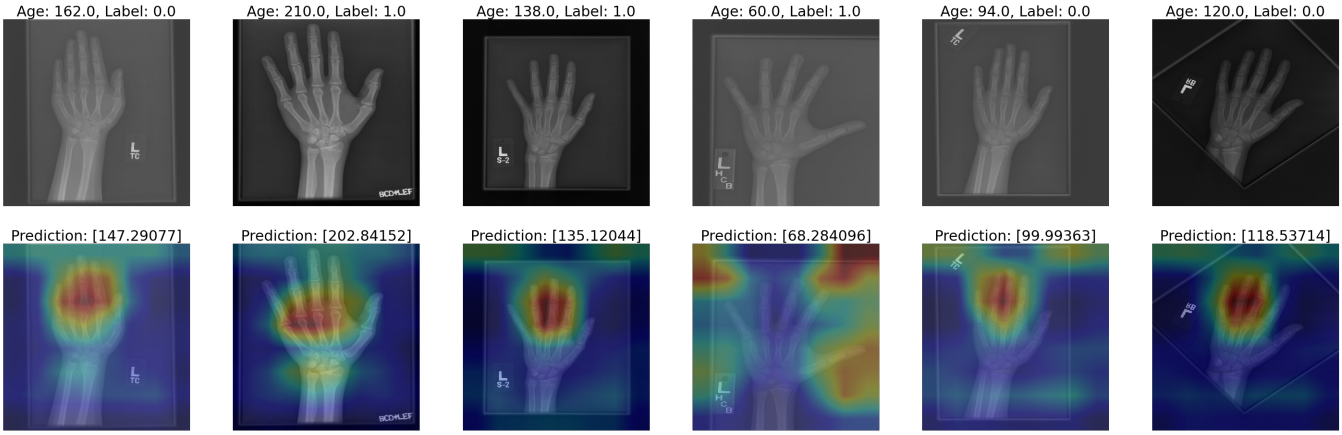

Fig. 4: CAMs produced by the best *BoneAgeInception* net (Modified 3 in TABLE. 1).

the baseline model, the results we obtained, having a limited computational and memory resources, were satisfactory and the model training method we adopted was successful.

As a very important remark is that, we have no mathematical proofs about how the illustrated hyperparameter optimization method would work in another context with a different starting model, the various blocks in a model may be correlated in a complex way and this may be a model specific approach. Nevertheless, we realized that once we got the right intuition about how to improve the baseline model, the efforts required to find the final model was worthy and acceptable, , for instance comparing with grid search or random search.

Finally, we noticed that setting width parameter β smaller than 1, namely downsizing the model, for the final model or for the baseline model the results we obtained were not so much worse than the original ones, on the other side we saved a lot computational time and memory. Thus, if we are allowed to slightly increase the prediction errors then we would gain a very efficient model. Importantly, perhaps in some application cases reducing width parameter magnitude is even beneficial for the model performance.

We could also investigate more on the factor β, applying this only for some specific layers in the network instead of all layers as we did and we are convinced that increasing the input image resolution we can further improve the results.

REFERENCES

[1] C. M. Gaskin, M. M. S. L. Kahn, J. C. Bertozzi, and P. M. Bunch, *Skeletal development of the hand and wrist: a radiographic atlas and digital bone age companion*. Oxford University Press, 2011.

[2] S. S. Halabi, L. M. Prevedello, J. Kalpathy-Cramer, A. B. Mamonov, A. Bilbily, M. Cicero, I. Pan, L. A. Pereira, R. T. Sousa, N. Abdala, *et al.*, "The rsna pediatric bone age machine learning challenge," *Radiology*, vol. 290, no. 2, pp. 498–503, 2019.

[3] M. Escobar, C. González, F. Torres, L. Daza, G. Triana, and P. Arbeláez, "Hand pose estimation for pediatric bone age assessment," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 531–539, 2019.

[4] C. Chen, Z. Chen, X. Jin, L. Li, W. Speier, and C. W. Arnold, "Attention-guided discriminative region localization and label distribution learning for bone age assessment," *arXiv e-prints*, pp. arXiv–2006, 2020.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[6] J. Liu, G. Zhao, Y. Fei, M. Zhang, Y. Wang, and Y. Yu, "Align, attend and locate: Chest x-ray diagnosis via contrast induced attention network with limited supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10632–10641, 2019.

[7] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[8] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

APPENDIX

What I learnt from this project and major difficulties: I found this project very instructive from either theoretical or practical point view.

It gave opportunity to me to deepen some advanced deep learning topics such as Efficientnets, DenseNets, many attention based model architectures etc. I improved the ability to quickly capture the most important concepts in a research paper and now I have a general view of this field related to deep learning applied to computer vision, thanks to also other courses in my master degree program.

From a practical point view, I became more familiar with the TensorFlow framework, I learnt the basic and advanced data processing pipeline using this tool and I can now exploit some low-level and powerful features of this library. I'm able to understand some parts of the source codes and then modify them creating a customized model. I learnt I understood how a medium sized cluster works and how to submit a machine learning job on it. I also had possibility to use private cloud tools for machine learning (AWS, GCP, Alibaba cloud etc.) such as VM or Jupyter notebook instances powered by a GPU, thanks to the free credits.

Regarding the difficulties I encountered during the project, I realized that computational resources are very precious and costly and even with a small dataset a single GPU strived to get model training phase done in a reasonable time, unless we really pay a lot of attention on the model efficiency, so I believe that the next deep model generation will take into account this issue (EfficientNets are some examples). Then, at the beginning I found difficult to write a good report, thanks to this opportunity I could train my writing skill but I'm still not fluent, so I need more exercises.