Predicting Bone-Age Using X-ray Images - *Deep Learning*

Adhiraj Roka

Applied Statistics

I. INTRODUCTION

Detecting the bone age of a child's skeletal system is important as it can help determine the growth rate of a child. Tracking the difference between a child's bone age and their chronological age can aid doctors in diagnosing a growth problem. However, these growth problems are not always an issue; healthy children can have a bone age that differs from their chronological age. This means that the child may be developing at a faster or slower rate than average. The changes in growth plates are essential for determining the age of a bone. Growth plates are easy to spot on X-rays because they are softer and contain less mineral, giving them a darker appearance when compared to the rest of the bone [1]. Hands contain many growth plates and have such become the standard for estimating bone age in children over the age of three. Hands of children under the age of three do not develop many changes within their first few years, so X-rays of the knee or hemi-skeleton are more often used [2].

Due to the unique characteristics of growth plates, a radiologist can assign a bone age based on the appearance. This process can be done by comparing the X-ray of the child to the X-ray images in a standard atlas of bone development and deciding which stage most closely matches the child [1]. New commercial means have recently become available for automating the determination of bone age [2]. We created a neural network, which is a series of algorithms that mimic the way the human brain operates to process data. By creating a neural network trained on various X-ray images, one could create a model to estimate the bone age of a child without having the subjectivity of human error.

II. DATA

Data was found on Kaggle [4] and provided by The Radiological Society of North America (RSNA) as part of their 2017 challenge. The goal of this challenge was to train a model and try to correctly identify the age of a child from their X-ray. To train the model you need to feed the machine several examples of properly identified images until it can make predictions. For simplicity, no matter the age of the child, an X-ray of their hand was used. There were four files included in the dataset - a folder of training X-rays, a folder of test X-rays, a comma separated values (.csv) file for the training X-rays containing the ID, Bone Age, and Gender of each X-ray, and a .csv file for the test X-rays containing the ID and Gender of each X-ray. There were 12,611 X-rays in the training set and 200 X-rays in the test set. Due to the large size of the training set and small RAM space available for use, we subsampled the X-ray images to get a smaller sample size to create a model with. A stratified sample based on gender was used, as we expect the population to use the model to be about 50% female and 50% male because that is approximately the distribution of the child population. We sampled 2,250 male X-rays and 2,250 female X-rays from the training set, giving us a final sample size of 4,500 X-rays, which was split into 3,600 for training and 900 for validation, we want to use as much data for training as possible to have enough data for our model to learn. The final proportion of males to females in the training set matches the original proportion of males to females.

Before beginning the analysis, the X-ray images were reviewed to see if there were any major imperfections to be aware of. Some images were very dark, and the X-ray was barely noticeable, some were very zoomed out and shifted to the corner of the image, some were flipped upside down, and some have different lighting throughout the single X-ray (Figure 1). Another area of concern is that all the X-rays have a label on them giving information about the patient. These labels do not correlate in any consistent way. In order to compensate for some of these inconsistencies, the data was augmented before building our models using rescaling, shear range, rotation range, zoom range, width and height shift range, and horizontal flip. This will help the models from overfitting the data.

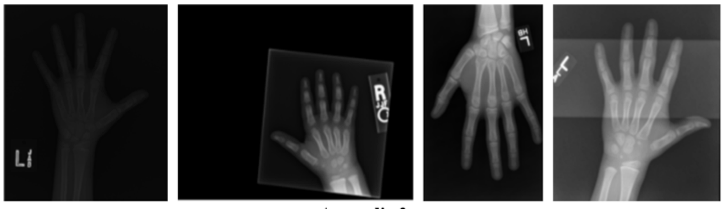


Figure 1: Sample images from dataset

III. LITERATURE REVIEW

As stated previously, this project was originally a Kaggle competition hosted by The RSNA in 2017. There were over 100 entries from 48 unique users in the competition. The models were evaluated using a testing set consisting of 100 images that were unavailable to participants. This training set is now available through Kaggle. The error was calculated using the mean absolute difference (MAD), which is the mean of all of the absolute values of fitted minus actual values. The RSNA reported details of the top 5 models, in terms of lowest MAD, reported [3], but the top two are discussed.

The first-place team achieved an MAD of 4.2 months. The architecture of their model can be seen in Figure 2. There were two inputs in this model, namely the image and gender. The image input was a 500x500 pixel input with one channel. It was passed through an InceptionV3 transfer learning model. On the gender input, it was passed through a dense layer with 32 neurons before being concatenated with the image model. After concatenation, the data was passed through two more dense layers each with 1000 neurons in an attempt to capture the relationship between gender and pixel. The two steps that greatly improved the performance of the model were noted to be data augmentation and an ensemble approach at test time [3].

The second-place team obtained a slightly higher error with a MAD of 4.4 months. This group trained gender-specific models using contrast-enhanced patches of the image, each of size 224x224 pixels. These patches were overlapped for a total of 49 patches for each image. Again, transfer learning was used with a fine-tuned ResNet-50 architecture. The final prediction was calculated by taking the ~50th percentile of the patch predictions. Finally, just as in the first-place model, data augmentation and ensembling on nine models were used to improve model performance [3].

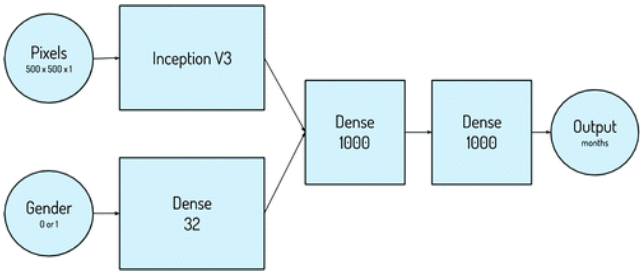


Figure 2: Architecture of first-place model in Kaggle competition

IV. MODELS

Numerous models were fit using three different input designs. Models, such as a Convolutional Neural Network (CNN) and a transfer learning model, were fit using inputs including gender only, X-ray image only, and both X-ray image and gender. The CNN model is primarily used for object identification and image classification, it uses a subsampling method to improve learning at each stage. Transfer learning models are designed with pre-existing models that have been previously trained on a larger image classification dataset. Three models will be discussed: a CNN model with only the X-ray images as an input, a transfer learning model with only the X-ray images as input, and a combined model consisting of transfer learning with the X-ray images as input and a perceptron model with gender as input. The mean absolute error (MAE) will be used to evaluate the performance of these models. This is computationally the same as the MAD used in the Kaggle competition described in section III.

i. Model 1: CNN Model with Image Input

The first model was a general CNN with the X-ray images as an input. Each X-ray image was resized to 64x64 pixels and consists of three channels for red, green, and blue. A few different sizes were tried as well, but none of them appeared to give a difference large enough to justify changing. Next, it was passed through two layers of convolution, each consisting of 32 filters and a kernel size of 3. The Rectified Linear Unit (ReLU) activation function was used in these two layers, as well. Preceding each convolution layer is a max-pooling layer with a pool size of 2 and stride of 2. This helped prevent our model from overfitting by preserving the features and reducing the number of parameters in the model. The model is then flattened to create a fully connected layer. This fully connected layer consists of a dense layer with 128 hidden neurons and the ReLU activation function. In this layer, the weights and features will be adjusted again. Finally, it is passed through the last dense layer, which has 1 hidden neuron and a linear function to get the predicted output of bone age in months (Figure 3).

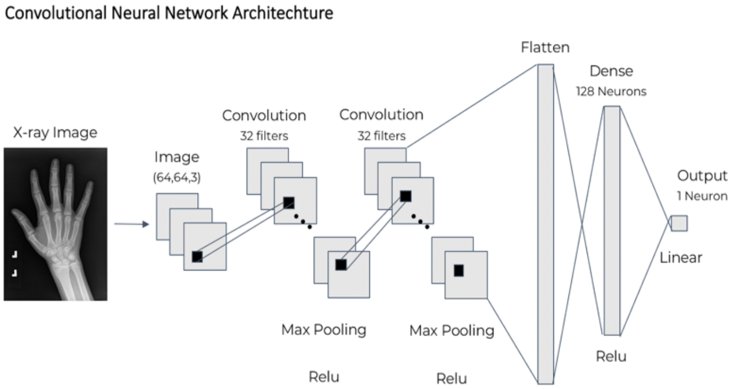


Figure 3: Model 1 CNN architecture

When running the model, the Adam optimizer was implemented with a loss function of mean squared error (MSE) and the performance metric of mean absolute error (MAE). The model was fit using the validation set to monitor how well the model was being trained. Fitting the model with 30 epochs was sufficient since the metrics began to converge prior to the 30th epoch. A plot of the training and validation MAE can be seen in Figure 4. Looking at this plot, we can see that the training MAE decreases fast for the first few epochs and then begins to level off and that the validation MAE is consistently lower than the training MAE with the exception of one epoch. This gives us a clue that our model may be underfitting the data since there is still room for improvement on the training data. The final validation MAE for this model was 29.29 months.

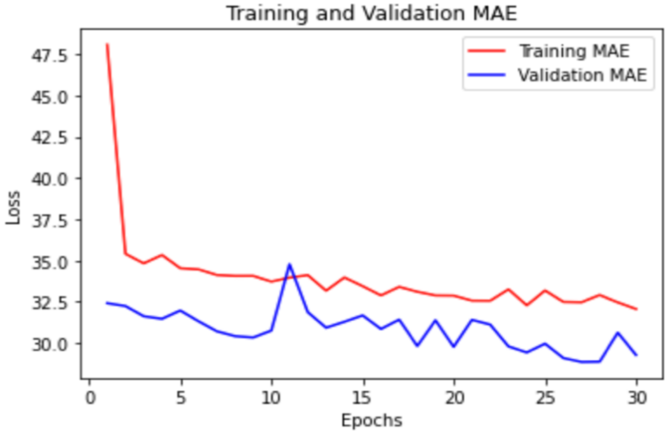
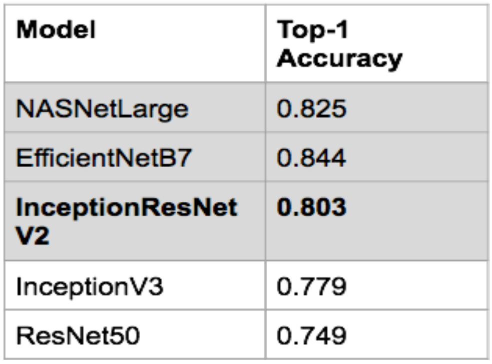


Figure 4: Training and Validation MAE on Model 1

ii. Model 2: Transfer Learning Model with Image Input

The second group of models consisted of a transfer learning model with the X-ray images as input. The top models in the Kaggle competition used InceptionV3 and ResNet50 for their pretrained models. Before building the transfer learning model, it was necessary to research other pretrained models to see if there was anything else appropriate to use. It was found that there were some newer and less trained models that had higher Top-1 accuracies (the accuracy of the model on the popular ImageNet validation set) in comparison to the previous models used. A table of the pretrained models are listed in Table 1.



Model	**Top-1 Accuracy**
NASNetLarge	0.825
EfficientNetB7	0.844
InceptionResNet V2	**0.803**
InceptionV3	0.779
ResNet50	0.749

Table 1: List of Transfer Learning Models Performance on ImageNet Dataset

NasNetLarge, EfficientNetB7 and InceptionResNetV2 were all attempted for this project. NasNetLarge had the best overall performance but consisted of an input size of 331x331 pixels. Due to RAM and Disk constrictions, Model 2 will focus solely on InceptionResNetV2 since it is computationally less expensive than NasNetLarge. In order to input the images into the InceptionResNetV2 model, they were resized to 150x150 pixels and 3 channels. The images were then trained on an InceptionResNetV2 transfer learning model. This was followed by a layer of flattening, a dense layer consisting of 256 hidden neurons with a ReLU activation function, and then passed to the output layer of one node with a linear activation function. This full architecture can be seen in Figure 5.

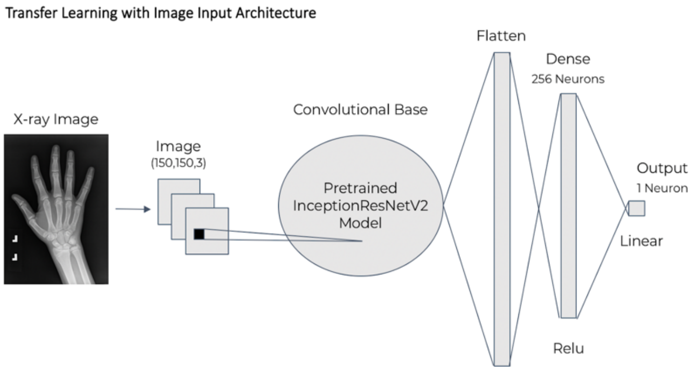


Figure 5: Model 2 (Transfer Learning with Image Input) Architecture

When running the model, the Adam optimizer was used with a learning rate of 0.0001 with MSE as the loss function and MAE as the performance metric. Running the model for 30 epochs proved to be sufficient since the metrics began to converge before the 30th epoch. The model was fit using the validation set to monitor how well the model was being trained. A plot of the training and validation MAE can be seen in Figure 6. This plot shows the training MAE decreasing and eventually converging. From this plot, we can assume the model is fitting better than the previous model since the validation MAE is not consistently higher or lower than the training MAE. The final validation MAE for this model was 24.13 months.

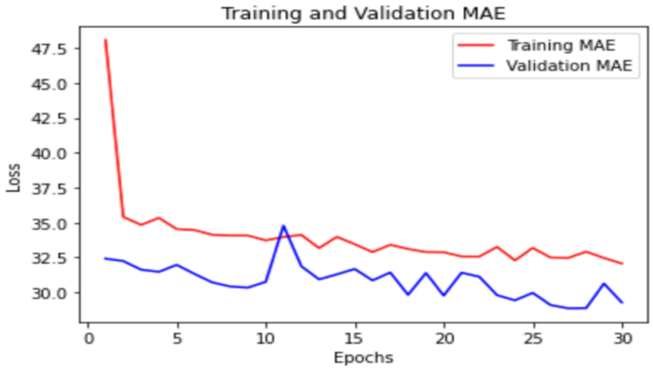


Figure 6: Training and Validation MAE on Model 2

iii. Model 3: Transfer Learning Model with Image Input Combined with Gender

The third model was a multiple inputs model consisting of an image input (X-rays) and a non-image input (gender). This architecture was inspired by a combination of the top two models from the Kaggle competition as stated in the Literature Review. In order to combine these two inputs, the widely used method *flow_from_dataframe()* will not work. A new method called *flow()* was researched and used instead. This method allows the augmented images and gender to be combined into a list and passed through model fitting. The architecture for this model is designed as two separate models that are linked to create one single model. The first input, the X-ray images, were converted into a 3-dimensional numeric array and passed through a convolutional base containing the pretrained InceptionResNetV2 model. Similarly, the second input, gender, was converted to a 1-dimensional numeric array and passed through a dense layer consisting of 32 hidden neurons and the ReLU activation function. Each of these models are then flattened to create a 1-dimensional vector. Taking these two flattened layers, the vectors were concatenated and passed through two dense layers having 256 hidden neurons and a ReLU activation function. Two dense layers were used to give the model enough parameters to analyze the relationship between the two inputs, X-ray and gender. The final dense layer was connected to an output layer that had 1 neuron and a linear activation function. The architecture for this model can be seen in Figure 7.

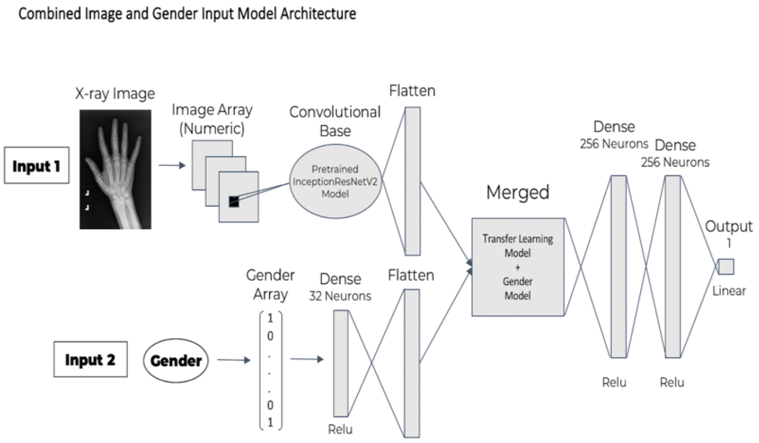


Figure 7: Model 3 (Image Input with Transfer Learning Concatenated with Gender Input)

During the compilation of the model, MSE was used as the loss function, MAE was used as the performance metric and the Adam optimizer was used with a one-tenth of a smaller learning rate than the default value in order to give more training time and to better converge to the local minima. Running the model for 30 epochs proved to be sufficient since the metrics began to converge before the 30th epoch. A plot of the training and validation MAE can be seen in Figure 8. The training MAE and validation MAE appear to converge, and the validation MAE is not consistently higher or lower than the training MAE. Thus, we can conclude this model is not under- or overfitting the data. After running the model, the final validation MAE was 20.96 months. As shown in Appendix 10, we can see that the model performance is much better than the previous two models discussed, and the validation MAE is also smaller.

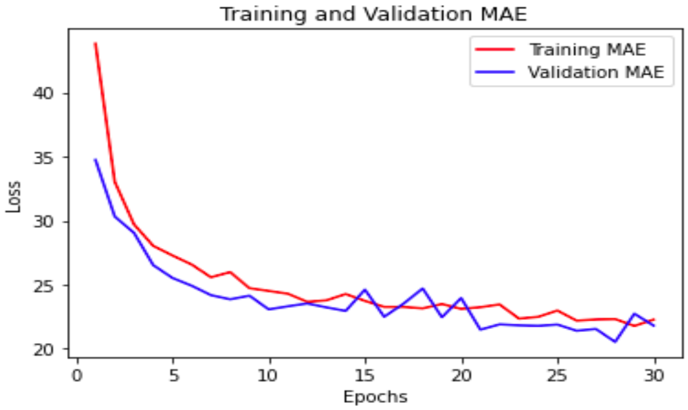
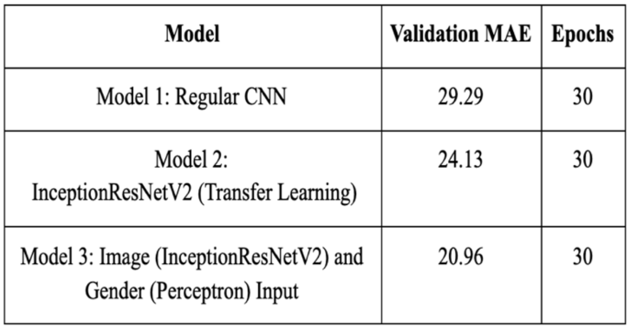


Figure 8: Training and Validation MAE on Model 3

In an attempt to improve the performance of the model, neurons were added to the last two dense layers. The number of hidden neurons was increased from 256 to 500. There was a very small increase in performance, but not enough to justify making that change. Moreover, there was somewhat of a bottleneck performance. Therefore, we conclude our final model to be the one described above and in Figure 7.

V. RESULTS

The overall results of the validation MAEs for the three models discussed are listed in Table 2. The first CNN model using only the images as input had an MAE of 29.29 months on the validation set. The second model using the InceptionResNetV2 transfer learning model had an MAE of 24.13 months on the validation set. The third model with multiple inputs where the images were passed through a pre-trained model and the gender input was passed through a perceptron and then concatenated gave an MAE of 20.96 months on the validation set.



Model	Validation MAE	Epochs
Model 1: Regular CNN	29.29	30
Model 2: InceptionResNetV2 (Transfer Learning)	24.13	30
Model 3: Image (InceptionResNetV2) and Gender (Perceptron) Input	20.96	30

Table 2: Performance (MAE) of all Three Models

To test the model's accuracy, we selected the best model from above and predicted the bone age of the 200 test set images provided by Kaggle. Since the last model, consisting of both X-ray images and gender inputs, gave the best performance compared to the individual models, we will use this model to do predictions on. A graph of the predicted bone ages vs. the actual bone ages in months can be seen in Figure 9. The MAE on the test set using our combined model was approximately 42 months. Looking at the actual values, we can see that the model does a better job at predicting the bone age for children of older age groups. However, the model has difficulty predicting the bone age for children of smaller age groups, as the predicted values are much more deviated from the actual line. Such an occurrence is possibly due to our training set having fewer images of children in that age group and can also be attributed to subsampling the original dataset.

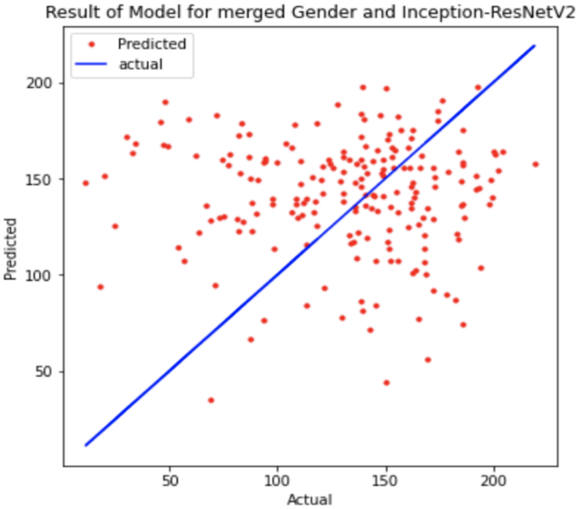


Figure 9: Actual vs. Predicted Values on Test Set for Model 3

VI. CONCLUSION, LIMITATIONS AND FUTURE WORK

Overall, our models have room to improve, especially when compared to the top models in the Kaggle competition. Our best model, Model 3, had an MAE of approximately 42 months. This would equate to about 3.5 years of error within our predictions. Thus, there is undoubtedly a lot more work to be done with this model to have an acceptable prediction to use in the medical field.

The main limitation incurred was limited processing power, causing the programs to crash or to have significant run times. Due to this limitation, the training set had to be cut from 12,611 images to only 4,500 images before splitting for validation. It is expected that this took a significant toll on the MAE for all of the models. Additionally, it was very difficult to have larger input images in terms of pixels due to processing time and code crashes. An additional limitation was the nature of the images. As discussed in the Data section, most images have excess text in as well as some lighting, shape, and background inconsistencies. These issues made it much more difficult to fit a model that can account for all of these situations.

In the future, it would be beneficial to address these limitations as well as fine tune the models and hyperparameters. If the processing power were to be increased enough to where all of the training data could be included in the fitting of the model, it is likely that the performance will increase greatly. Increasing processing power also will allow for larger inputs and more dense layers, which would also likely boost performance. It would specifically be interesting to test our Model 3 with 500x500 input and two 1000-neuron dense layers at the end, as in the first Kaggle model. This would allow for a better comparison, and it is possible our model would outperform the Kaggle model due to the higher accuracy, in terms of Top-1 accuracy, that InceptionResNetV2 has over InceptionV3.

Almost all of the Kaggle submissions suggest data augmentation as an imperative step to fitting the model. More work should be done to address the image inconsistencies and attempt to augment the data in a more useful way. One approach in the Kaggle submissions was to remove the background off all the X-ray images leaving just the hand behind. They then overlay the image on top of a solid black background to guarantee all of the backgrounds are the same and to remove some of the error. Another useful technique that almost all of the Kaggle submissions implemented was ensemble methods, which would also likely be useful to decrease our overall error.

To conclude, there are quite a few ways in which the models can be improved. Addressing the issues and implementing some new methods as well as increasing our training size is expected to put our model performance in competition with some of the other models that were researched.

ACKNOWLEDGEMENTS

I would like to acknowledge Kim Bianchi and Katie Schmidt for helping me with the findings of this paper. Additionally, I would also like to recognize Dr. Yimin Zhang for providing insights and guidance through her past experiences in this field of study.

REFERENCES

Kidshealth.org. (2020). X-Ray Exam: Bone Age Study (for Parents) - Nemours KidsHealth. [online] Available at: https://kidshealth.org/en/parents/xray-bone-age.html.

Creo, A.L. and Schwenk, W.F. (2017). Bone Age: A Handy Tool for Pediatric Providers. Pediatrics, 140(6), p.e20171486.

H.H., Chen, L., Shih, G., Andriole, K., Kohli, M.D., Erickson, B.J. and Flanders, A.E. (2019). The RSNA Pediatric Bone Age Machine Learning Challenge. Radiology, 290(2), pp.498–503.

kaggle.com. (n.d.). RSNA Bone Age. [online] Available at: https://www.kaggle.com/kmader/rsna-bone-age.