Predicting Pediatric Bone Age from X-rays Using Convolutional Neural Network

Paras Memon
Department of Computer Science
Colorado State University
Fort Collins, CO 80526
memon@cs.colostate.edu

Joaquin Cuomo
Department of Computer Science
Colorado State University
Fort Collins, CO 80526
jcuomo@colostate.edu

May 20, 2019

1 Problem Formulation

A common practice in a pediatric assessment is to evaluate the bone age of a child to diagnose diseases to an early detection of any potential anomaly. Bone age is the average age at which children reached certain bone maturity and combined with the chronological age a wide variety of conclusions can be withdrawn. For example, the growth potential and disorders related to it, such as hypothyroidism, growth hormone deficiency and pubertal delay. Since bone maturation is more correlated to linear growth than to chronological age, the epiphyses of the bones, which are the rounded end parts of a bone in a joint with other bones, are used to determining the bone age of a child. The easiest way of assessing the epiphyses is through radiography of the non-dominant hand [11].

The most widely used methods to determine the bone age are the "Greulich and Pyle", "Bayley-Pinneau, Roche et al." and "Tanner-Whitehouse" methods [7]. These all methods are manually done and considered to be time consuming. To solve this kind of a problem, the Radiology Society of North America (RSNA) launched a competition in 2017, to automate the assessment of pediatric bone age base on x-Rays.

The goal of this project was to build a system that can be used to predict the bone age giving the hand radiography of a child. The dataset used consists of 9 GB of labeled images, which demands a great amount of computer power or parallelism techniques to handle its processing. The developed system aims to at least match the manual techniques precision while outperforming in processing time. Ideally, a fine tuned model of this system could be used by physicians

to assist them on the evaluation of radiography to save them time and make it independent of human bias.

2 Strategy to Solve the Problem

The dataset can be found in Kaggle [4] and contains more than 12,000 images with the correspondent bone age and sex of each patientâs X-Ray. Some of the x-Rays are from the left hand and others from the right hand with no label on that feature.

To solve the image prediction problem our main strategy will be using a pre-trained convolutional neural network. The framework we found the most suitable to do it is Tensor Flow On Spark, as we can take advantage of the cluster we already have with Spark and Hadoop and at the same time we can use the vast variety of functions provided by Tensor Flow. Hadoop Distributed File System will be the file system to support the dataset distribution in the cluster.

The framework to run our software was Hadoop Distributed File System to store the dataset and all intermediate files. The process parallelism was achieved using an Apache Spark cluster with 10 workers nodes and the API was TensorFlowOnSpark. Specifically, we used Keras with TensorFlow as the backend and then converted the Keras model into a TensorFlow Estimator feed with RDD so when run in PySpark it can distribute the processing among the workers.

To improve the accuracy of the prediction two different techniques were applied. First, we used a specific histogram equalization algorithm named CLAHE (Contrast Limited Adaptive Histogram Equalization) as shown in the Figure 1, to enhance the contrast of the x-Ray locally while limiting the amplification of noise.

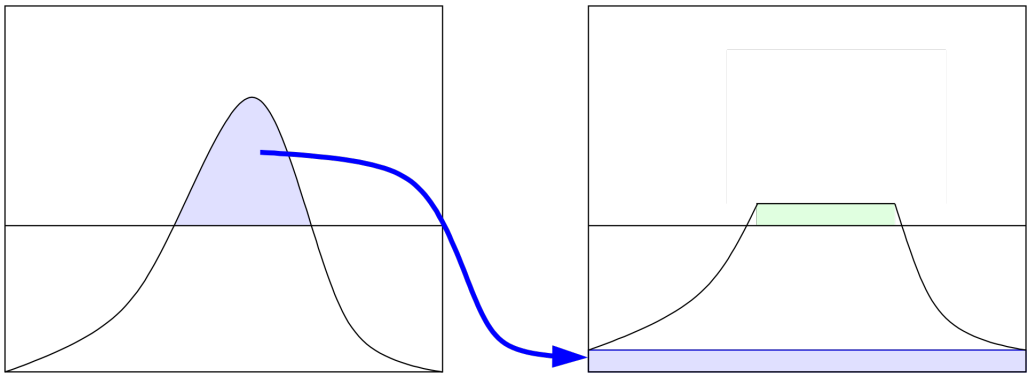

Figure 1: Histogram equalization [1]

The main idea behind CLAHE is limit the amplification of contrast in near-constant regions of the image by means of a threshold in the histogram and then redistribute the excess in an uniform way as shown in Figure 1.

Then a convolutional neural net was trained to do semantic segmentation on the x-Ray and determine which part is the hand so we could masked it and discard the other parts such as tags or borders. We opted for using the U-Net in Figure 2 which was originally created for biomedical image segmentation and outperforms all current methods, and we based our implementation on zhixuhao [12] and zizhaozhang [13] implementations.

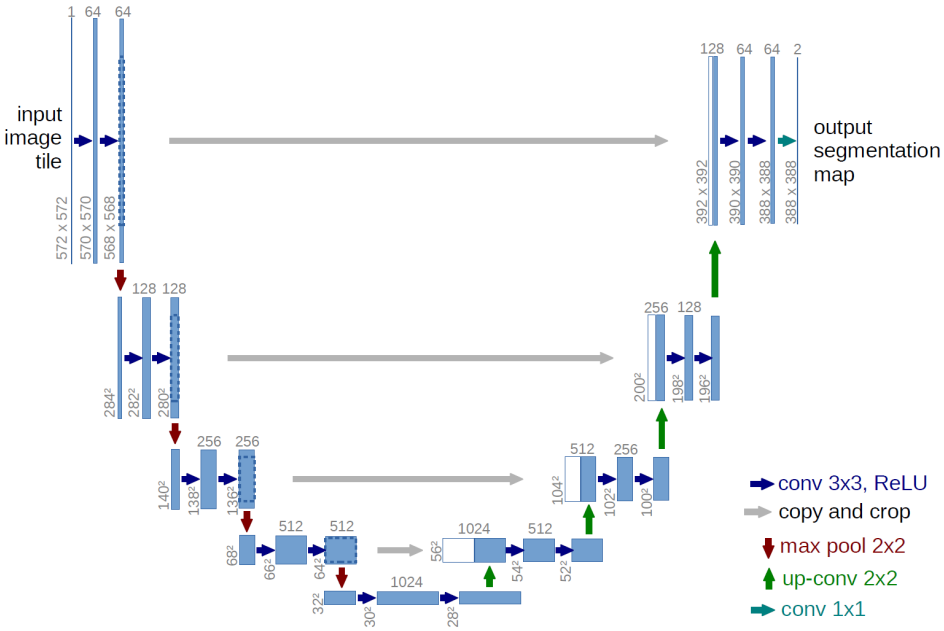

Figure 2: U-Net architecture [8].

U-Net is a fully convolutional network (FCN [2]). First it runs an encoder (stack of convolutional and max pooling layers) to capture the context in the image and then a decoder (symmetric to the decoder) to enable precise localization using transposed convolutions. The main difference with other FCN is that it is symmetric and it skips connections between the downsampling path and the upsampling path by applying a concatenation operator instead of a sum.

To create the training/testing dataset we used the Pixel Annotation Tool [6] to create the labels on the x-Rays. Because, hand-labeling images is a tedious and time-consuming work we only did it on 30 images and used data augmentation by means of slights rotations, scaling and mirroring to have enough images to train the U-Net.

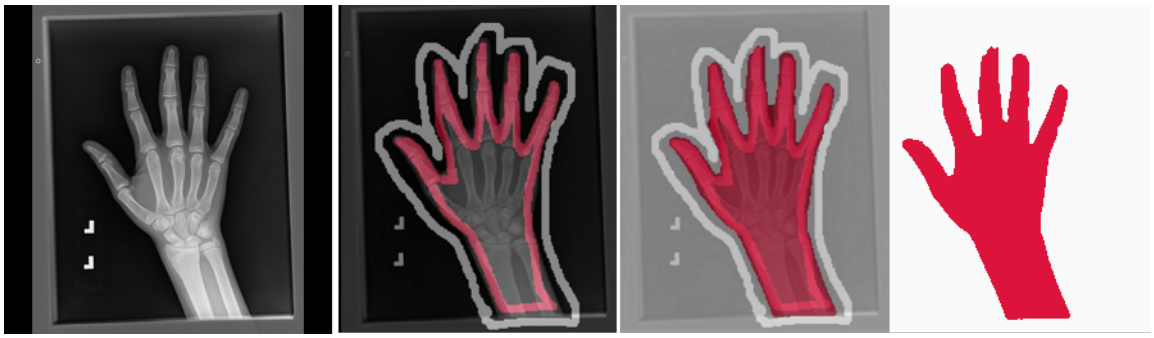

Figure 3: Labeling process using Pixel Annotation Tool. From left to right: raw x-Ray, areas tagging, automatic area detection, final mask.

To further 'simplify' the work to the neural network we attempt to increase the similarity of each x-Ray by rotating them so the mayor axis (in line with the middle finger) was straight. To find the right rotation angle we fitted the hand contour to an ellipse and the found the angle of the mayor axis.

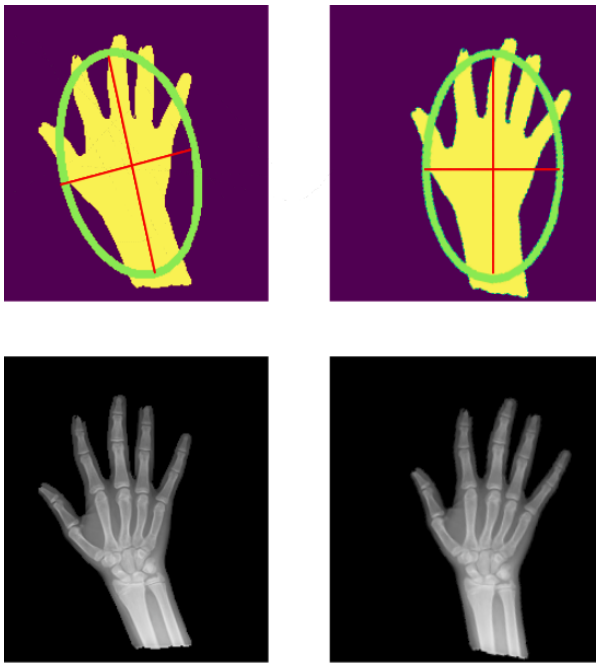

Figure 4: Example of a rotation

2.1 Convolutional Neural Network

The bone age is predicted using the Convolutional Neural Network (CNN) as shown in the Figure 5. CNN consists of an input, output layer, and multiple hidden layers. Inside CNN, the hidden layers consist of an input, output layer, and multiple hidden layers. Inside CNN, the hidden layers consist of convolutional layers, pooling layers, fully connected layers.

First, the processed x-ray images used as the input data to the input layer of the convolution network, the dimension of the input data is 256*256*1. Then in the hidden layers, data is given to the convolutional layers. First convolution layer in the learning network is composed of 16*3*3 convolution kernels; the second convolution layer composed of 32*3*3, the third convolution layer consists of 64*3*3, and the fourth convolution layer consists of 128*3*3. After the convolutional layers, the data moved to the pooling layer, which is also known as downsampling layer; it reduces the input dimension of the input features but does not change the number of features. In our algorithm, all the pooling layer used the 2*2 filter. Furthermore, the sample features integrated after convolution and pooling. The features that are most conducive to the model extracted. The full connection layer is one dimensional feature vector by using the flatten operation, which directs the features that are obtained from the learning network to the output layer to predict the decision output. Our convolutional network predicts the bone age using x-rays data.

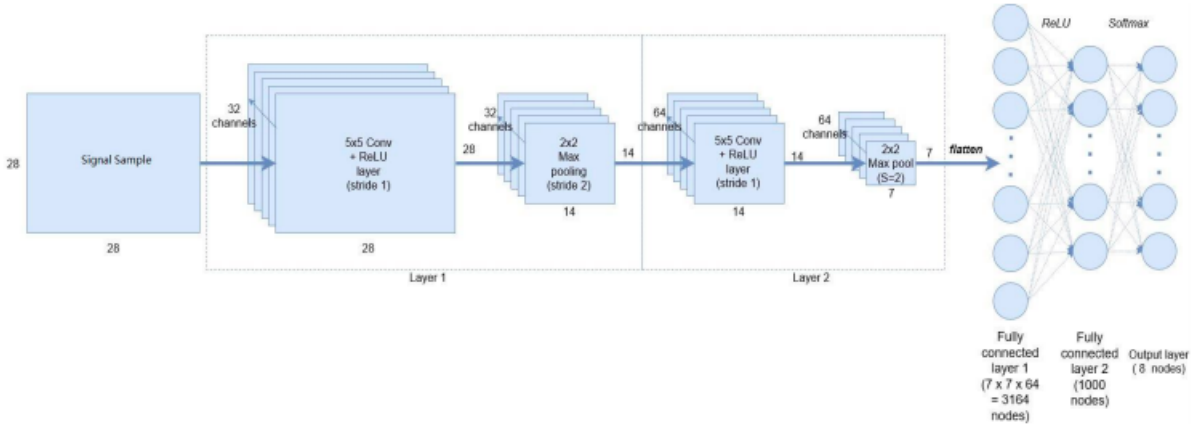

Figure 5: Convolutional Neural Network [9].

Figure 6 summarize the process flow of our system.

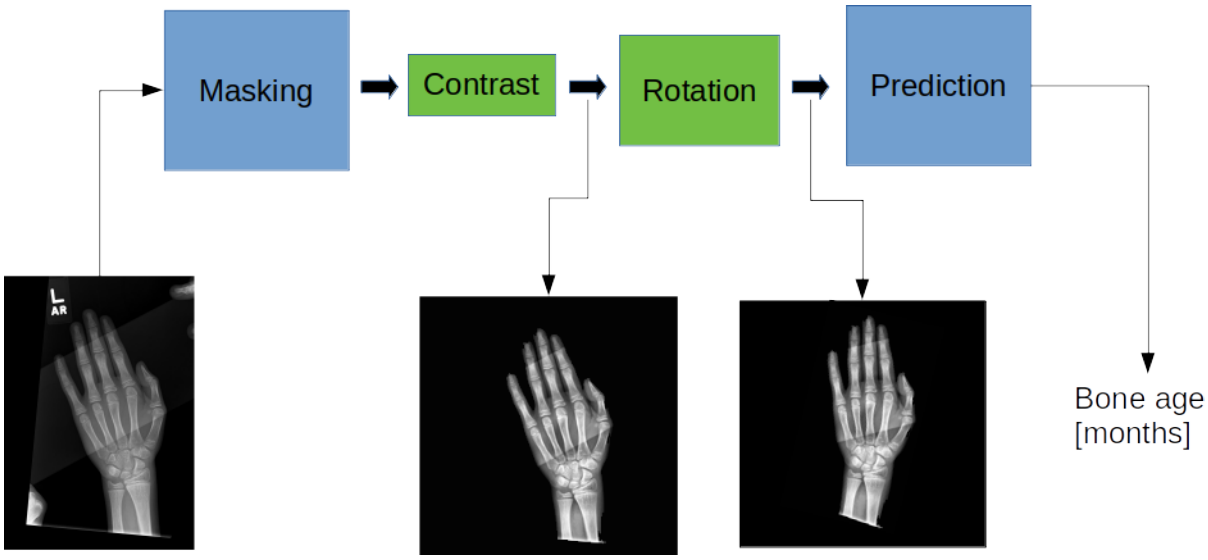

Figure 6: Summary process flow to predict bone age.

3 Functions of the Software

We provide several python scripts that guide the user to the prediction of the bone age, but can also be used independently in different classification/prediction tasks.

python 1_*Preprocess_images.py* < *image_dir* >< *mode* >: depending on the mode (re-size or contrast) it will reshape the images in the given directory to the desire size preserving the aspect ratio and filling the background with a white solid color, or apply CLAHE algorithm to enhance the contrast.

```
${SPARK_HOME}/bin/spark-submit
--master ${MASTER}
--conf spark.cores.max=${TOTAL_CORES}
--conf spark.task.cpus=${CORES_PER_WORKER}
--conf spark.executorEnv.JAVA_HOME="$JAVA_HOME"
/trainUnet.py
```

will trained the U-Net under Spark. The output is the saved trained model.

python 3_*maskImages.py* < *image_dir* >< *model* >: given an image directory and a U-Net model it will run the neural net on the images. The output is all the images masked.

python 3_*trainNN.py* < *image_dir* >< *label_dir* >: given an image and label directories it will train the neural net on the images. The output is all

training/testing images prediction.

python 3_predictBoneage.py < image >: given an input image it will output the predicted bone age.

4 Testing process of your software

For testing the U-Net, as the labeling was done by hand we did not have as many images as we would want for training and testing, so we hand-labeled only 6 x-Ray for the testing. An important aspect of the evaluation of the U-Net, as it is image segmentation and even the labelling is not perfect, is the visualization of output images to see if they include the main bones that are looked when assessing the bone age of a child and that they do not include any non-hand part of the x-Ray.

The accuracy of the U-Net was measured not with the direct output of the neural net but against a post-processing consisting on selecting the proper predicted mask (based on the expected area of a hand in a x-Ray) as there were some artifacts in the prediction. In Figure 7 the second image illustrates this as we can observed a small 'spot' on the bottom right part, and the third image is after the before mentioned processing.

To test the regression model a portion of training dataset (randomly picked) was set aside to be used only for testing purpose. Despite Kaggle provided a testing dataset they did not release any results from other implementations so we choose not to use it. Then, the NN was tested using non-exhaustive cross-validation. The partitioning of data is done using k=5 equally-size subsamples and over k iterations using each subsample once as the validation set. Finally, the results were averaged in order to obtain a single estimation.

The framework of this work was TensorFlow on a Spark cluster with HDFS as the distributed file system. Despite that, not all the code was prepared to run in cluster mode. We focus on running only the U-Net on parallel because it was the one that consumed the most time during training. For that we adapted the code to be able to run on Spark basing us on the official documentation of TensorFlowOnSpark [5]. Besides taking care of version compatibility, the main modification was on how the data was being feed to the NN. We used the input mode *InputMode.SPARK* so the images, saved in HDFS as CSV files, were distributed among all workers instead of using the input mode *InputMode.TENSORFLOW* in which the whole dataset was loaded in memory on each worker. For using SPARK input mode the images were converted to RDD and fed to the NN with a generator.

5 Results and Evaluation

The overall results of semantic segmentation was satisfactory. An example can be seen in Figure 7.

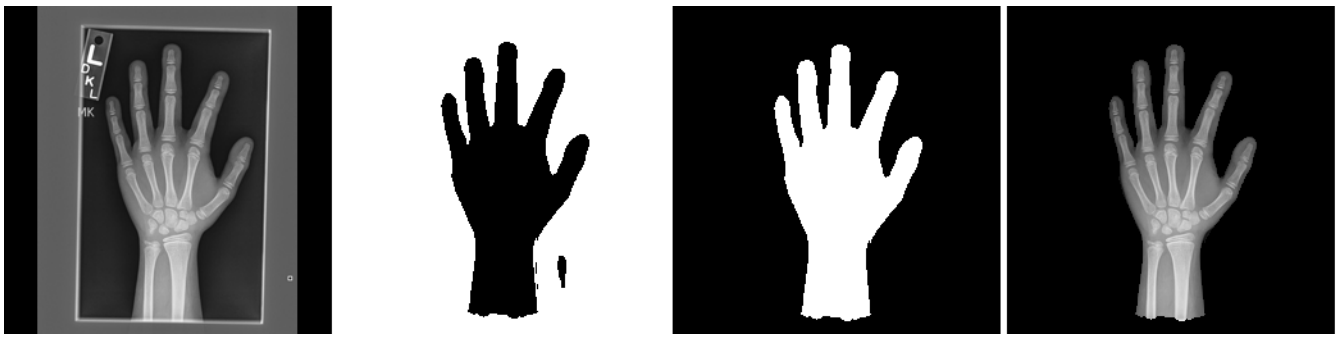

Figure 7: Example of an x-Ray masked by the U-Net

To evaluate the semantic segmentation model we used the Jaccard coefficient which represents the overlapping between the ground truth and the prediction. It counts the number of pixels in common between the ground truth and predicted masks divided by the pixels present across both masks.

$$Jaccard = \frac{target \cap prediction}{target \cup prediction} \tag{1}$$

The score is calculated for each test image individually and then averaged over all images to provide a global median score of our semantic segmentation prediction.

Figure 8: Example of Jaccard coefficient for image segmentation

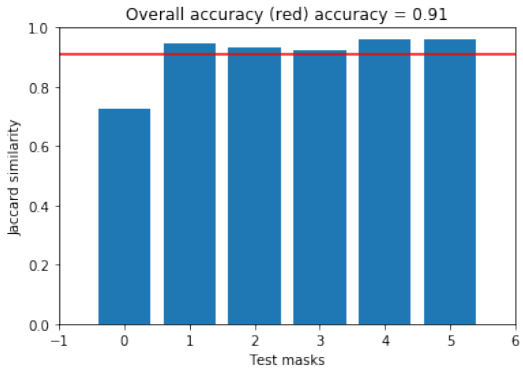

Figure 9: Accuracy of the U-Net on the testing set

The training of the U-Net yield an accuracy of 94% despite the training set was augmented with a seed of only 30 x-Rays. Because the labeling was done by hand and was not perfect the measurement is also not truly a ground truth so a visual inspection was done of the results and the resulting masks were very satisfactory as all of them included both the carpal bones and the proximal phalanges, which are the features of the hand that are used to determine bone-age and discarded all unwanted artifacts like tags and borders.

The x-Rays were highly improved using CLAHE algorithm as show in Figure 10.

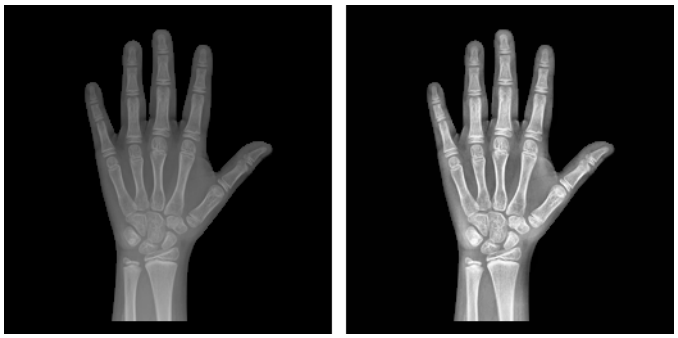

Figure 10: Example of an x-Ray enhanced using CLAHE

The training and testing regression plots are shown in Figures 11 and 12. These plots shows the ground truth against the predicted output values. If the accuracy of the predictions was higher we would see plots with data cluster close to a 45° line.

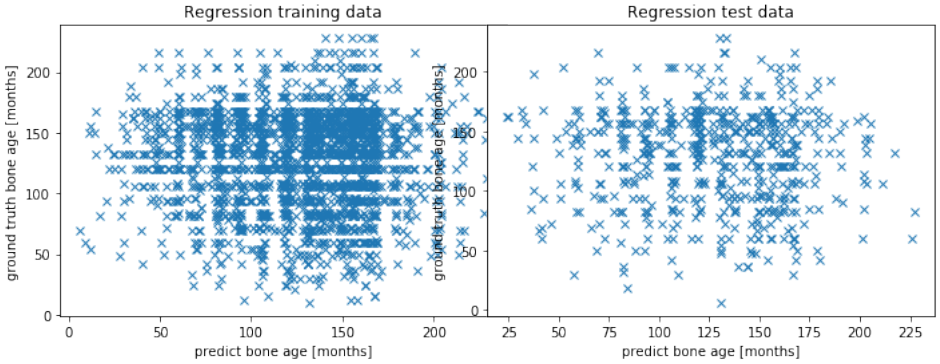

Figure 11: Training Accuracy Plot

Figure 12: Testing Accuracy Plot

To evaluate the regression model the Mean Absolute Deviation (MAD) (see Eq. 2) was computed to compare the results of a pre-defined testing dataset (from Kaggle) with other implementations of the prediction model [3].

$$MAD = \frac{\sum_{i=1}^{n} |x_i - \bar{x}|}{n} \tag{2}$$

The Root Mean Square Error (RMSE) (see Eq. 2) was also calculated to compare and improve the implementations and tuning of the algorithm.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x_i} - x_i)^2} \tag{3}$$

Table 1 shows the error values of the training and testing dataset.

Table 1: Training and Testing Error Values

Dataset	RMSE	MAD
Training	55.98	44.52
Testing	56.78	45.97

When we evaluate our model we cannot say it can predict the bone age as the regression plot shows a randomize behaviour. On the other hand, if we compare the performance of other implementations as shown in Table 2 it seems that only four groups were able to achieve reasonably good metrics.

Table 2: Results of Kaggle RSNA challenge

#	MAD
1	0.000 (1)
2	7.634 (2)
3	7.968 (3)
4	8.723 (4)
5	43.300 (5)
6	62.091 (6)
7	62.091 (6)

Initially, the regression results using directly the raw x-Ray as input to the model were not good, so we did the fore mention pre-processing to the images in the attempt to improve the accuracy. Unfortunately, the predictions were inaccurate to make any meaningful comparison of were this pre-processing could be useful for improving the prediction accuracy.

In conclusion, we consider that the prediction of the bone age given an x-Ray should be done in a much fine grain, meaning that probably a better extraction of the relevant features should be done prior to train the neural net. Thodberg et al. [10] did some good work on this regard isolating each epiphyses as shown in Figure 13.

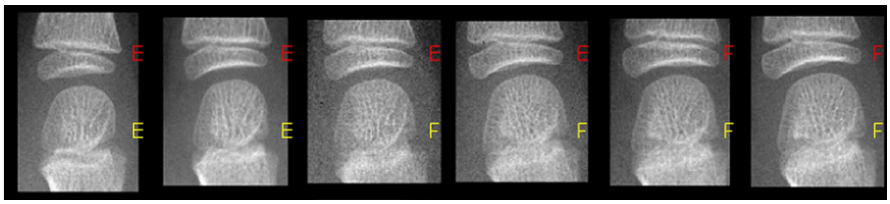

Figure 13: Fine Features Extraction [10]

6 Contributions

1. Joaquin Cuomo
    - U-Net
    - Contrast enhancement
    - Rotation
    - Spark cluster
    - Report
2. Paras Memon

- Tensorflow CNN
  - Pilot Experiment
  - Big Data Experiment
- Report

References

[1] Adaptive histogram equalization.

[2] Review: Fcnâââfully convolutional network (semantic segmentation).

[3] Competition - rsna pediatric boneage challenge. 2017.

[4] Rsna bone age — kaggle. 2017.

[5] Tensorflowonspark, 2019.

[6] abreheret. Pixelannotationtool.

[7] V. Gilsanz and O. Ratib. *Hand Bone Age: A Digital Atlas of Skeletal Maturity*. Springer Berlin Heidelberg, 2005.

[8] Philipp Fischer Olaf Ronneberger and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science*, 9351, nov 2015.

[9] Weiguo Shen and Wei Wang. Node identification in wireless network based on convolutional neural network. In *2018 14th International Conference on Computational Intelligence and Security (CIS)*, pages 238–241. IEEE, 2018.

[10] Hans Henrik Thodberg, Sven Kreiborg, Anders Juul, and Karen Damgaard Pedersen. The bonexpert method for automated determination of skeletal maturity. *IEEE transactions on medical imaging*, 28(1):52–66, 2009.

[11] M. Zacharin. *Practical Pediatric Endocrinology in a Limited Resource Setting*. Elsevier Science, 2013.

[12] zhixuhao. Implementation of deep learning framework – unet, using keras, 2019.

[13] zizhaozhang. unet-tensorflow-keras, 2017.