INDEX

List of Figures:

List of tables:

ABSTRACT

Skeletal maturity progresses through discrete phases, a fact that is routinely used in paediatrics where bone age is compared to chronological age in the evaluation of endocrine, metabolic and growth disorders. Bone age assessment is a common clinical practice to investigate these disorders in children. It is generally performed by radiological examination of the left hand by using either the Greulich and Pyle (G&P) method or the Tanner-Whitehouse (TW) method. It is a safe and painless procedure that uses a negligible amount of radiation. The G&P method is the approach used by most radiologists because of its simplicity and speed and is based on the comparison between the complete X-ray of the non-dominant hand and a reference atlas. However, both clinical procedures show several limitations, from the examination effort of radiologists to significant intra- and inter-operator variability.

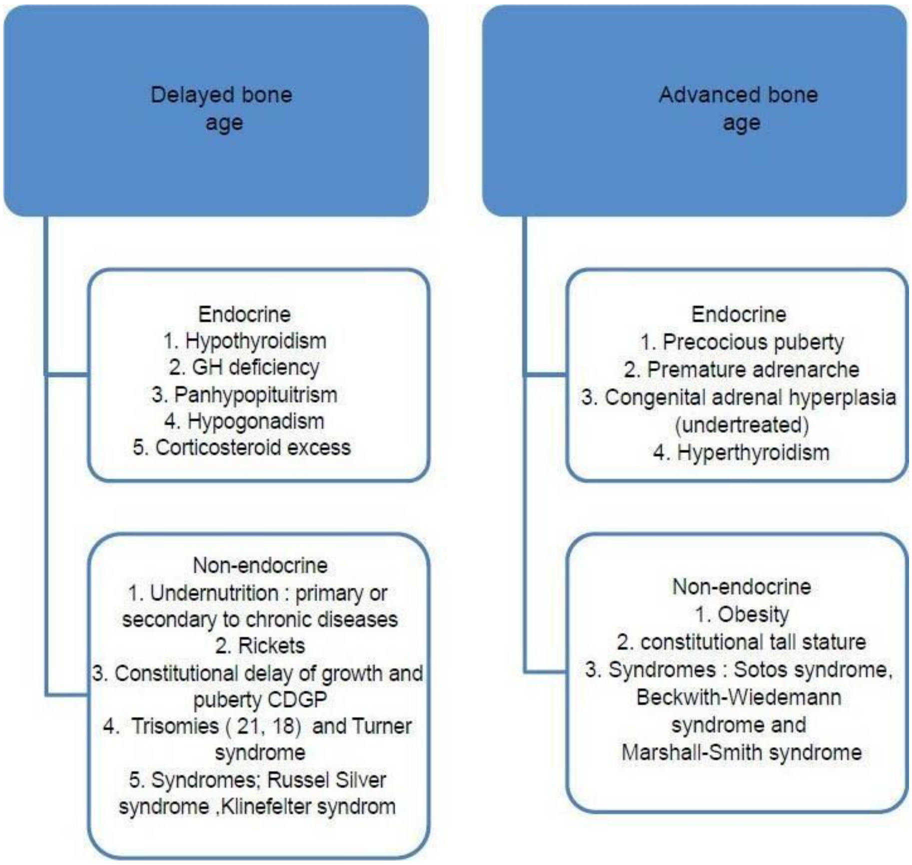
The latter half of the 20th century has seen a significant increase in the use of automation in the field of biomedicine. Moreover, the advancements in the field of electronics paved the way for computers having high computational power which led to solving problems unconventionally but efficiently through the use of Machine Learning.

In this Project, we propose a solution for automated Bone Age assessment by imbibing the concepts of Machine Learning. The project utilizes a number of deep learning networks which are used to segment a region of interest from our given input radio-graphs and perform a number of operations to standardize and pre-process the input. The results obtained from preprocessing neural networks pipeline are at par and sometimes better than the ones obtained with the traditional approach. The normalized radio-graphs are given to our regression model which assesses the bone age of the subjects.

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The assessment of skeletal maturity or ―Bone Age‖ is a common radiological examination used in paediatrics and pediatric endocrinology to determine any discrepancy between a child‘s bone age i.e. how far the child has advanced in the development of their bones and their chronological age. A discrepancy between these two values indicates abnormalities or a hormonal problem. Bone age assessment is an important clinical procedure in endocrinology to investigate genetic growth disorders in children. Some of the issues related to Bone Age have been covered in fig. 1



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4266871/

Figure 1 : Some of the medical issues pertaining to Bone Age

Bone age can be used to predict:

- How much time a child will be growing
- When a child will enter puberty
- What the child's ultimate height will be

The classical method of skeletal bone age assessment utilizes the recognition of changes in the radiographic appearance of the maturity indicators in a hand-wrist X-ray of the nondominant hand of an individual by comparison with a reference data set/atlas of known sample bones. These X-rays are classified according to sex and age. This examination is subjective and relies on the judgment of the radiologist and hence is time-consuming and prone to error. All calculations in this process are specific laboratory tool dependent and in case of bone-related surgeries or bone replacements, the measurements must be accurate or the surgery may fail due to overestimation or underestimation of bone length or bone angle and hence lead to disability or deformity.

We are developing an automated method using deep learning to have a Convolutional Neural Network (CNN) to predict the bone age.

1.2 SCOPE

The main objective is to predict the age of the person from the X-ray using CNN. The age is limited up to 15 years as the doctors can cure growth disorders only till a certain age threshold. In order to realize our objective, we have used some state-of-the-art CNN architectures. These include:

U-net Architecture: This is a convolutional neural network mainly used in Biomedical Image Segmentation. This architecture is the winner of various IEEE International Symposium on Biomedical Imaging Challenge 2015. We are using this architecture for Semantic Segmentation in our project.

VGG16: This is a deep convolutional network for large-scale image recognition. This architecture secured first place for image localization in ImageNet challenge 2014. We are using this architecture for keypoint detection in our project.

Densenet121: This is a network architecture wherein each layer is directly connected to every other layer in a feed-forward fashion. We are using this architecture for predicting the age.

Semantic segmentation and keypoint detection are a part of the preprocessing pipeline.

After the successful implementation of the CNN, we developed a web application using Django Python framework wherein users can upload their X-ray image and get the predicted age within seconds.

1.3 ORGANIZATION OF REPORT

- **Chapter 1: Introduction and scope**

  This chapter provides an overview of the basic functionality of the system and describes its scope of expansion

- **Chapter 2: Literature Survey and present scenario**

  This chapter enlightens the literature survey of the work done in this field so far as well as the present scenario.

- **Chapter 3: System Block Diagram and Flowchart**

  Explains in detail the design and development process of the system. Includes system specifications block diagram, description of each block.

- **Chapter 4: System Design**

  It includes all the algorithms used for implementation of this project.

- **Chapter 5: Result and Conclusion**

  It includes results obtained during various phases, conclusion and future scope of the project.

- **Chapter 6: Reference**

  Includes important research papers referred for algorithms of the system.

CHAPTER 2
REVIEW OF THE RELATED LITERATURE

2.1 LITERATURE

There are several papers based on identifying bone age using the Tanner-Whitehouse method and Greulich-Pyle method. Multiple papers have been written on segmentation of the X-ray images. We have summarized them below:-

- Xiaodong Zhang, Fucang Jia, Suhuai Luo, Guiying Liu, and Qingmao Hu proposed a marker-based watershed method for X-ray segmentation to segment background of Xray images. The method consisted of six parts: image pre-processing, gradient computation, marker extraction, watershed segmentation, region-merging and background extraction.
- Chengsu Ouyang, Yongxuan Huang proposed using a snake model to map the contours for segmentation. Snakes are energy minimizing curves which deformed under the influence of internal forces and external force. The internal forces within curve itself and external force derived from the image are used to minimize the energy function. Thus the problem of finding an object boundary can be regarded as an energy minimization process.
- C. Spampinato, S. Palazzo, D. Giordano, M. Aldinucci, R. Leonard, in their paper proposed and tested several deep learning approaches to assess bone age automatically; the results showed an average discrepancy between the manual and automatic evaluation of about 0.8 years, which was state-of-the-art performance. Furthermore, this was the first automated skeletal bone age assessment work tested on a public dataset and for all age ranges, races and genders, thus representing an exhaustive baseline for future research in the field. Besides the specific application scenario, this paper provides answers to more general questions about deep learning on medical images: from the comparison between deep-learned features and manually-crafted ones to the usage of deep-learning methods trained on general imagery for medical problems, to how to train a CNN with few images.

2.2 PRESENT SCENARIO

Currently, there are 2 methods for manual bone age assessment. The Greulich and Pyle (GP) method for Bone Age Assessment involves the radiologist referencing an atlas of X-rays of healthy children and selecting the age which looks the most similar. The atlas used in this method was made in the 1950s and is not applicable anymore, this is a major drawback of this method. Another alternative is the Tanner-Whitehouse method, where individual bones are given a score from (A-I) and the final bone age is calculated by averaging the score. This is a more robust method than GP. These methods are very time consuming so automated methods need to be developed.

Currently, there are certain systems already in place for assessing bone age. However, they have been developed only for specific ethnicities (eg. Caucasians) and there are no available systems for Asians. The bone sizes differ with changes in ethnicities and thus different system have to be developed. Bone Age Assessment becomes harder with the change in skeletal sizes which are different for Indians and thus incompatible with the systems in place. There is no system for Bone Age Assessment commercially available in India at the moment.

CHAPTER 3
SYSTEM SPECIFICATION

3.1 BLOCK DIAGRAM

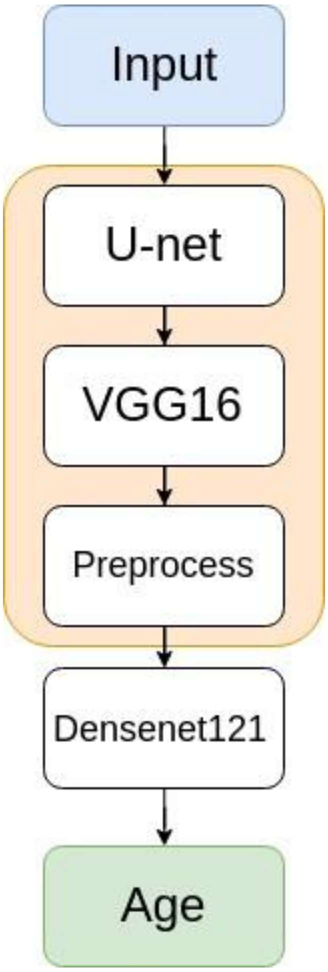


Figure 2: System block diagram

3.2 PROJECT SPECIFICATION

The main aim is to develop a robust system which can predict the age of the given X-rays.

Hardware used:

- Nvidia Titan X GPU

Specifications:

NVIDIA CUDA® Cores	3584
Memory Speed	10 Gbps
Standard Memory Config	12 GB GDDR5X
Memory Interface Width	384-bit

Platform / Language used:

- Python 2.7
- Django 1.11.0 Python Web - Framework

Libraries used:

- Keras - Deep Learning Library
- Scikit-learn - High-Level Machine Learning Library
- Numpy - Handling Matrices
- Matplotlib - Plotting Graphs and Images
- Scipy- Mathematical tools

Online Tools used:

- Supervisely

3.3 COMPLEXITIES INVOLVED

We faced the following difficulties while implementing our system:

- Due to noisy X-rays, we had to implement a pre-processing pipeline in our system. We also had to discard the X-rays of damaged hands since they were detrimental to the training of the network.
- The skeletal structures of children below the age of 3 and children above the age of 15 are somewhat similar and that can cause mispredictions in the system.
- We had to test various neural networks such as DenseNet to find out which one gives the best possible results.
- The RSNA dataset does not have an equal amount of X-rays for each month between the ages of 3 & 15. So, we had to augment the dataset to an extent and use a regression model.
- Determining whether a Neural Network trained on a limited dataset would give better results than the traditional methods.
- We have currently implemented a system that does not take into account the gender of the subject but we need to train two different networks for both genders to get better results during classification.

CHAPTER 4
SYSTEM DESIGN

4.1 Data Collection:-

We obtained a database of X-rays of the left hands of individuals of different age groups through an arrangement with Dr Vaman Khadilkar of Jahangir Hospital. The X-rays are 1024x1024 JPEG images. Due to the limited availability of dataset of Indian children we have used the publicly available dataset provided by Radiological Society of North America (RSNA) to determine if our system is working correctly. The dataset consists of 12,612 images of ages 0 months to 228 months. The images are from two U.S. hospitals and have been donated to the RSNA. These images came with labels for their skeletal age in months and the gender of the patient.

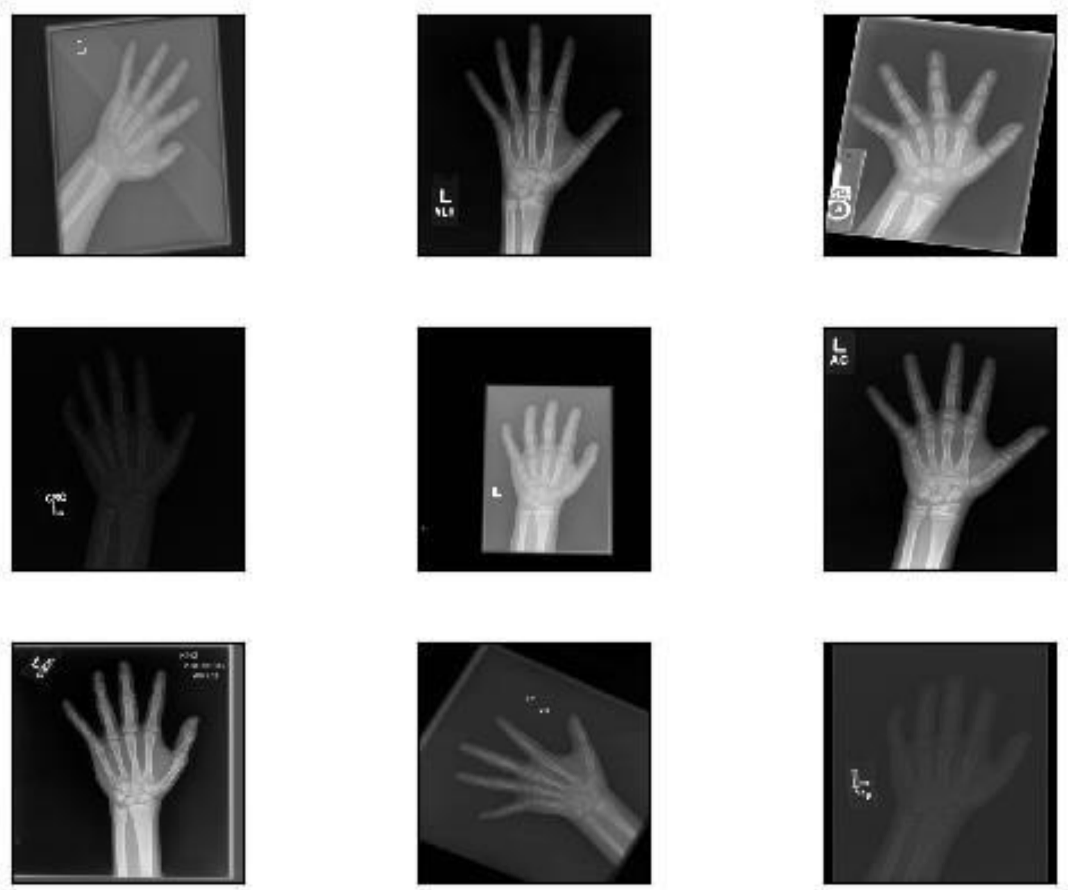


Figure 3: Sample RSNA Dataset

4.2 Preprocessing:-

As we can see in the above images, they have a varied level of contrast, brightness and noise in the background. The actual hand itself is at different positions in the image covering a small portion of the actual and with random rotations. Hence simple foreground segmentation techniques did not work. We have used deep learning for preprocessing as well.

4.2.1 Semantic Segmentation:-

Semantic segmentation is a type of image segmentation technique in which each pixel is assigned a label. It can be thought of as a classification task where each pixel is given a particular class.

The typical use of convolutional networks is on classification tasks, where the output of an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image.

4.2.1.1 Mask Creation:-

To facilitate ease in creating masks for the semantic segmentation we used an online tool called Supervisely. We were able to create 300 masks manually with each image requiring 2 minutes for masking

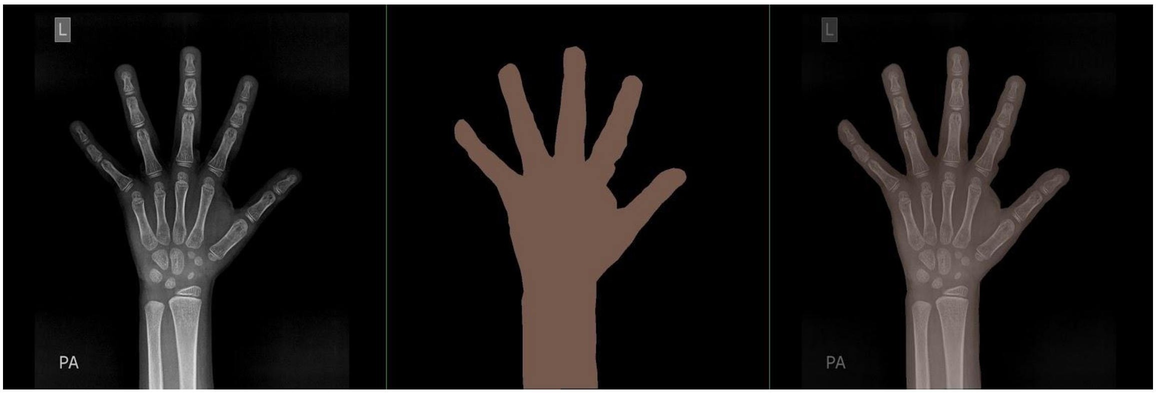


Figure 4: Image and corresponding Mask

4.2.2 U-net: Convolutional Networks for Biomedical Image Segmentation:-

U-net is a very popular Convolutional Neural Network used in the segmentation of BioMedical images. This network and training strategy relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path which downscales the image to feature vectors to capture context and a symmetric expanding path that upscales the image to the original size that enables precise localization. The network also contains skip connections which allows it to use higher resolution features during the upscaling phase. Since up-sampling is a sparse operation we need a good prior from earlier stages to better represent the localization.

We were able to train the end-to-end network using very few images (300 for a dataset of 12,000 images) using the dice coefficient as a loss function.

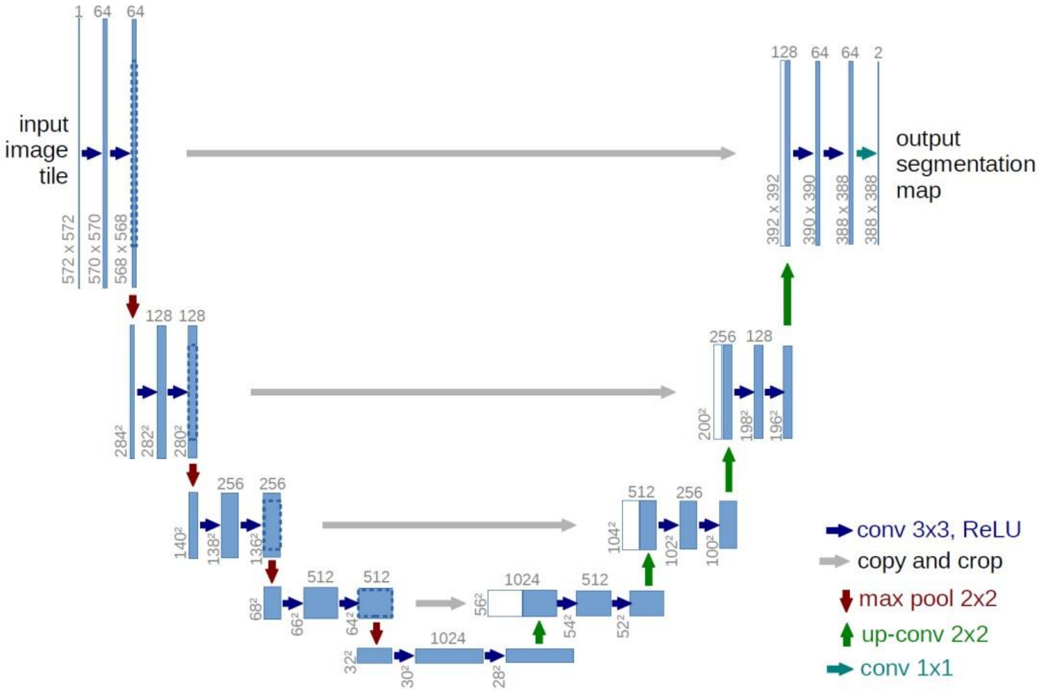


Figure 5: U-Net Architecture

For training of u-net, we are minimizing the dice coefficient. Dice Co-efficient is a statistic for measuring the similarity of two sets. In case of image segmentation tasks, we measure the similarity between the two images. One will be the Target images mask and the other will be the predicted image mask. The negative of the Dice coefficient is called the dice loss which is being minimized in the network. There are other similarity index like cosine similarity and Jaccard index.

U-net network fundamentally consists of:

- Convolution layers with ReLU Activation Function
- Max Pooling layers

4.2.2.1 Convolutional Neural Network:

A convolutional neural network (CNN) is a special kind of neural networks that have been widely applied to a variety of pattern recognition problems, such as computer vision, speech recognition, etc.

We are focusing on two-dimensional convolutional neural networks. The basic idea of CNN is to build invariance properties into neural networks by creating models that are invariant to certain inputs transformation. This idea originates from a problem that often occurs in the feedforward neural networks, especially multilayer feed forward neural network (MLP). The problem is all MLP layers fully connected to each other. It removes the spatial information of the inputs which are needed for the computational.

Unlike the ordinary neural networks, CNN has a special architecture. The architecture of CNN usually is composed of a convolutional layer and a sub-sampling layer as presented in Figure 1. The convolutional layer implements a convolution operation, and a sub-sampling operation was done in the sub-sampling layer. Here, the sub-sampling is also known as a pooling. A CNN is built based on three basic ideas, i.e., local receptive fields, weight sharing, and pooling.

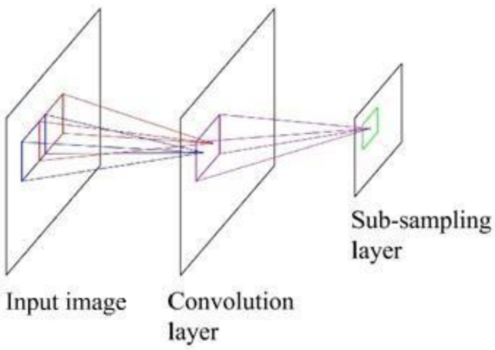


Figure 6: The architecture of a standard CNN

Local receptive fields:

In the feed forward neural network, the input is fully connected to the next hidden node for every neuron. In contrast, the input of CNN only makes the connection within a small region. Each neuron in a hidden layer will be connected to a small field of the previous layer, which is called a local receptive field. For example, if the field has a 3 × 3 area, a neuron of the first convolutional layer is corresponding to 9 pixels of the input layer. Figure 2 illustrates the small coloured box as the local receptive fields, and the coloured lines represent where the neuron is connected to.

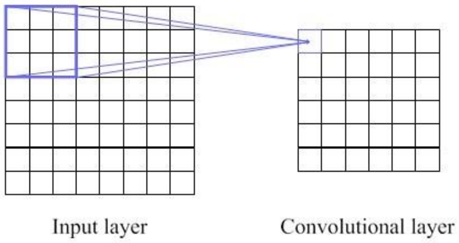


Figure 7: An example of local receptive fields on a convolutional layer.

Weight Sharing

In the convolutional layer, the neurons are organized into multiple parallel hidden layers, which so-called feature maps. Each neuron in a feature map is connected to a local receptive field. For every feature map, all neurons share the same weight parameter that is known as filter or kernel.

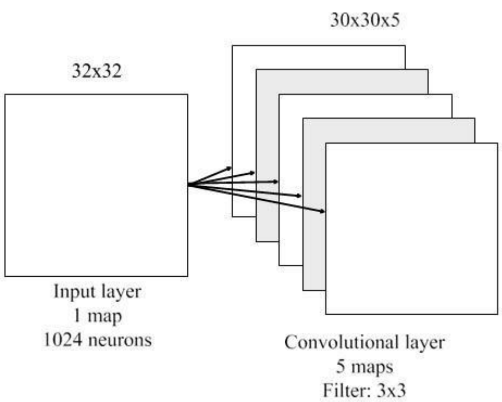


Figure 8: An example of a convolutional layer.

Pooling:

A CNN contains not only convolutional layers, but sometimes also pooling layers. When there is a pooling layer, it is usually used immediately after a convolutional layer. It means the outputs of the convolutional layer are the inputs to the pooling layer of the network. The idea of a pooling layer is to generate translation invariant features by computing statistics of the convolution activations from a small receptive field that corresponds to the feature map. The size of a small receptive field in here depends on the pooling size or kernel pooling.

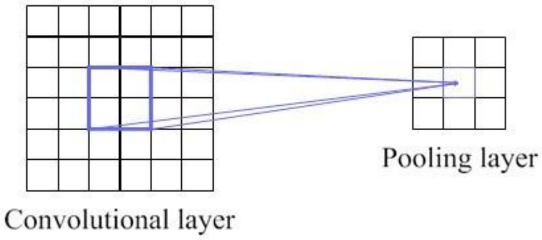


Figure 9: An example of a pooling layer in a feature map. This example uses 2 × 2 pooling size

There are two methods of pooling popularly used in the literature: Max pooling and Mean/Average pooling.

Max Pooling: The maximum value out of all the values in the region/kernel is taken as the output. Max pooling is sensitive to the existence of some pattern in the pooled region

Mean/Average Pooling: The mean value of all the values is taken as the output.

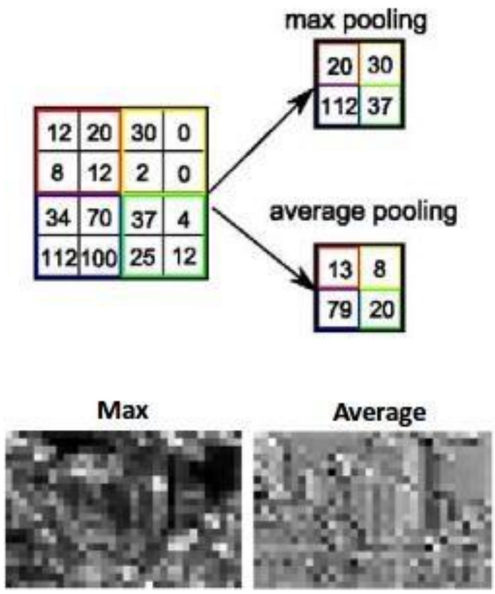


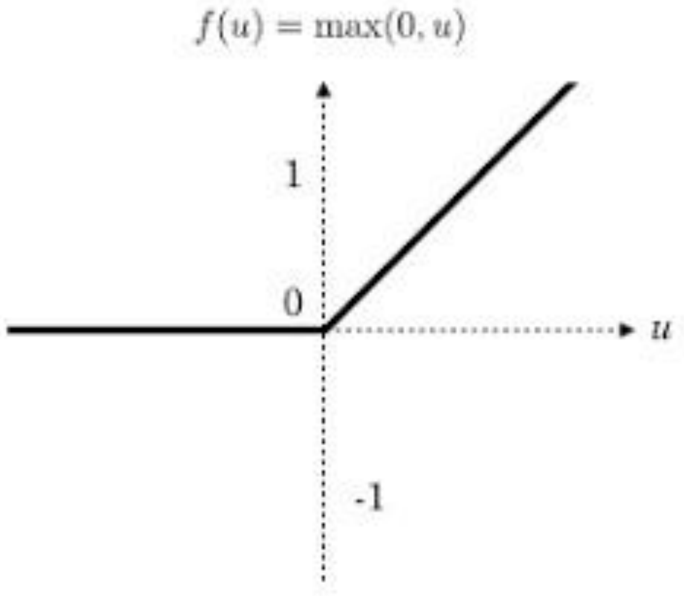
Figure 9(b): Pooling

ReLU :

The Rectified Linear Unit has become very popular in the last few years. It computes the function f(x)=max(0,x)f(x)=max(0,x). In other words, the activation is simply thresholded at zero (see image above on the left). There are several pros and cons to using the ReLUs: (+) It was found to greatly accelerate (e.g. a factor of 6 in Krizhevsky et al.) the convergence of stochastic gradient descent compared to the sigmoid/tanh functions. It is argued that this is due to its linear, non-saturating form.

(+) Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.

(-) Unfortunately, ReLU units can be fragile during training and can ―die‖ . For example, a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any data point again. If this happens, then the gradient flowing through the unit will forever be zero from that point on. That is, the ReLU units can irreversibly die during training since they can get knocked off the data manifold. For example, you may find that as much as 40% of your network can be ―dead‖ if the learning rate is set too high. With a proper setting of the learning rate, this is less frequently an issue.



Adam Optimizer:

Adam (short for Adaptive Moment Estimation) is an update to the RMSProp optimizer. In this optimization algorithm, running averages of both the gradients and the second moments of the gradients are used. Given parameters w(t) and a loss function L(t), where t indexes the current training iteration (indexed at 1), Adam's parameter update is given by:

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1)\nabla_w L^{(t)}$$
$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1 - \beta_2)(\nabla_w L^{(t)})^2$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1 - \beta_1^t}$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1 - \beta_2^t}$$

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon}$$

Back-propagation:

- Method of training artificial neural networks used in conjunction with an optimization method such as gradient descent.

- Requires a known desired output for each input value in order to calculate the loss function gradient.

- Back-propagation requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method, although it is also used in some unsupervised networks such as auto-encoders.

- It is a generalization of the delta rule to multi-layered feed-forward networks, made possible by using the chain rule to iteratively compute gradients for each layer. Backpropagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable.

- The number of hidden layers and the number of neurons decide its efficiency.

- The back-propagation learning algorithm can be divided into two phases: propagation and weight update.

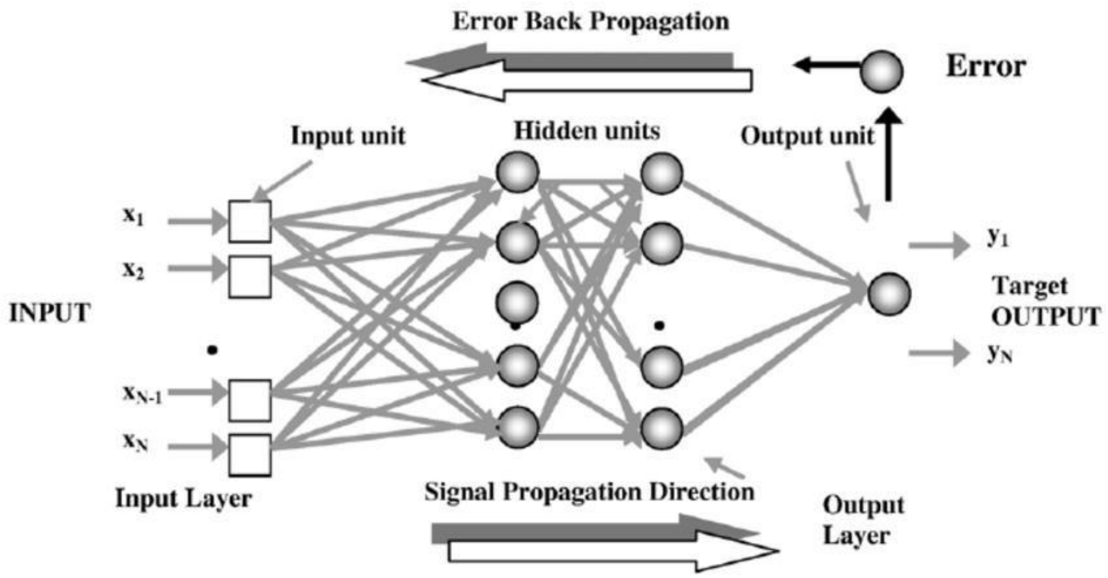


Figure 10: Back Propagation diagram

Phase 1: Propagation

- ✓ Forward propagation of a training pattern's input through the neural network in order to generate the propagation output activations.

- ✓ Backward propagation of the propagation output activations through the neural network using the training pattern target in order to generate the deltas (the difference between the targeted and actual output values) of all output and hidden neurons.

Phase 2: Weight update

- ✓ Multiply its output delta and input activation to get the gradient of the weight.

- ✓ Subtract a ratio (percentage) from the gradient of the weight. This ratio (percentage) influences the speed and quality of learning; it is called the learning rate. The greater the ratio, the faster the neuron trains, but the lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing; this is why the weight must be updated in the opposite direction.

Dropout:

Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons. Dropout refers to ignoring units (i.e. neurons) during the training phase of a certain set of neurons which is chosen at random. These units are not considered during a particular forward or backward pass. , At each training stage, individual nodes are either dropped out of the net with probability 1-p or kept with probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed. A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.

Dropout is used in all state-of-art networks to reduce over fitting. It is a harmless technique which always improves the performance of the network. One important thing to note is that dropout is only used during the training phase because we want the network to learn alternative more robust routes. It is not used in the testing phase where we test is the network has learned these alternative routes effectively or not. Dropout roughly doubles the number of iterations required to converge. However, training time for each epoch is less.

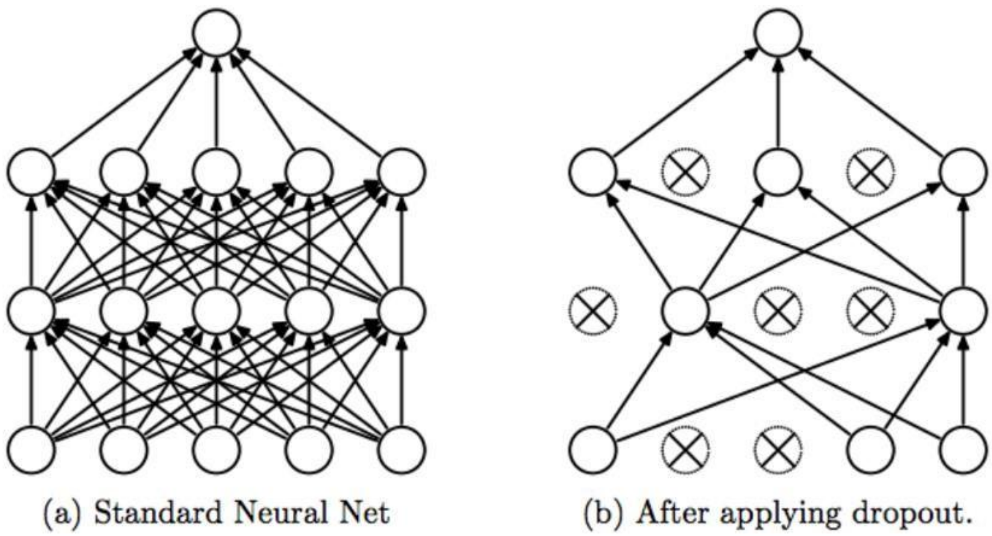


Figure 11: Dropout

Batch-Normalization:

A common practice in deep learning is that we normalize the input layer by adjusting and scaling the activations. For example, when we have features from 0 to 1 and some from 1 to 1000, we should normalize them to the range [0,1] or [-1,1] to speed up learning. The whole network benefits from it. Batch normalization allows each layer of a network to learn by itself a little bit more independently of other layers. Batch normalization reduces the amount of covariance shift caused by the hidden layers.

To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. However, after this shift/scale of activation outputs by some randomly initialized parameters, the weights in the next layer are no longer optimal. SGD (Stochastic gradient descent) undoes this normalization if it‘s a way for it to minimize the loss function. Consequently, batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a ―standard deviation‖ parameter (gamma) and add a ―mean‖ parameter (beta). In other words, batch normalization lets SGD do the denormalization by changing only these two weights for each activation, instead of losing the stability of the network by changing all the weights.

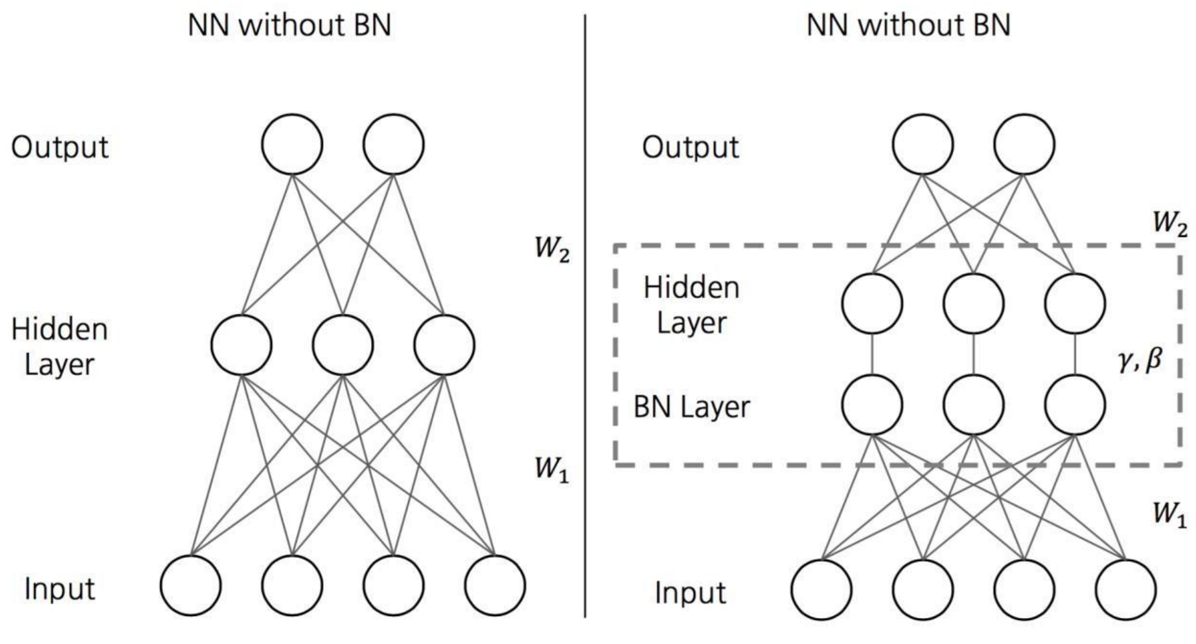


Figure 12: Batch-Normalization

We can use higher learning rates because batch normalization makes sure that there‘s no activation that‘s gone really high or really low. And by that, things that previously couldn‘t get to train, it will start to train.

It reduces overfitting because it has a slight regularization effect. Similar to dropout, it adds some noise to each hidden layer‘s activations. Therefore, if batch normalization is used, less dropout can be used, which is a good thing because we are not going to lose a lot of information. However, complete dependence only on batch normalization for regularization is not good; it should always be used with dropout.

4.2.2.2 Positive Mining

Expanding the initial training set with the additional good quality masks increases the size of the labelled images for segmentation procedure and improves segmentation results. To achieve an acceptable quality of the whole training set, we repeated this procedure three times. This is called as Positive Mining.

After masking the images, we have a set of images with varying size, intensity and orientation. This can be seen by the intermediate result after our U-net network as shown in figure 11. In order to standardize the images, we need to find few localized points which will help us in standardizing the images.

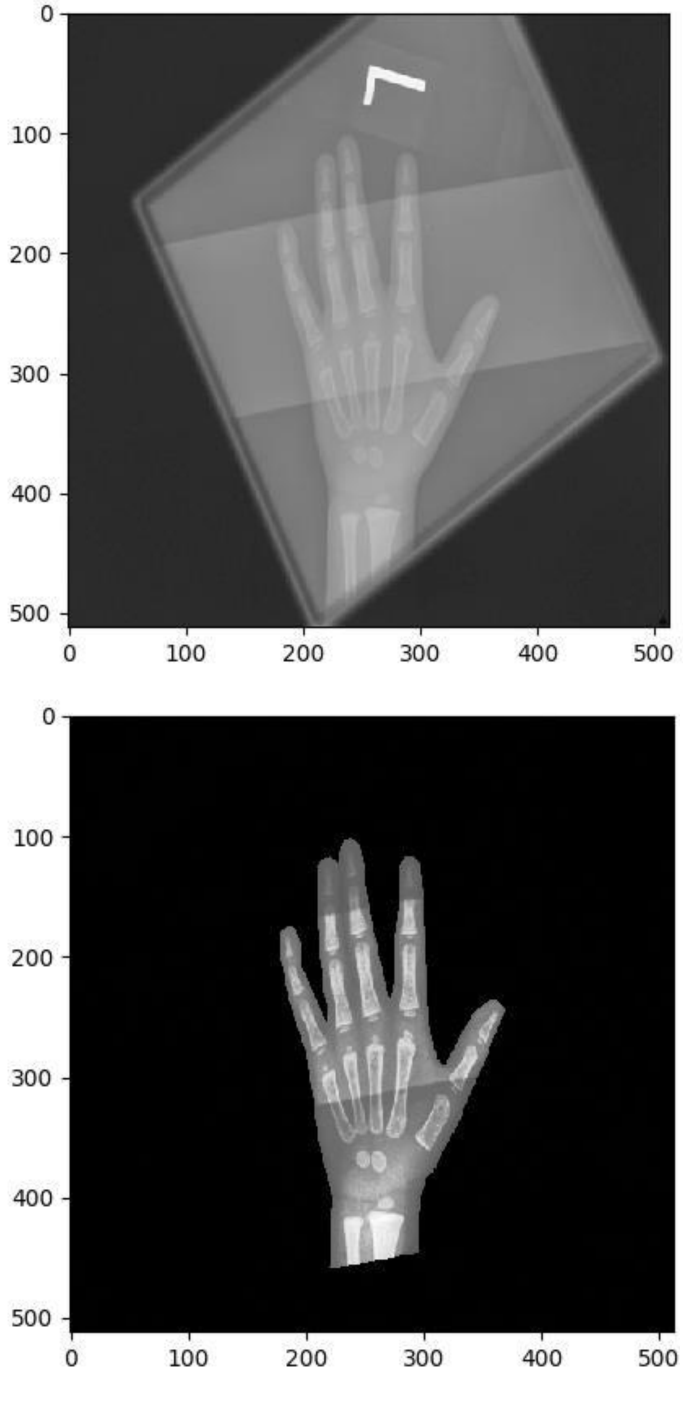


Figure 13: Segmented Images

4.2.3 Key point Detection:

The images in the dataset are not standardized. They have a varying degree of rotation, occupy a different location in the image and uneven length of the arm is present. To tackle these problems we detect 3 key points in the x-ray as shown. We created a dataset of 800 images with their labels being the 6 coordinates of the 3 points. ie. X-coordinate and the Y-coordinate.
We have used the VGG16 architecture and minimizing the mean squared error.

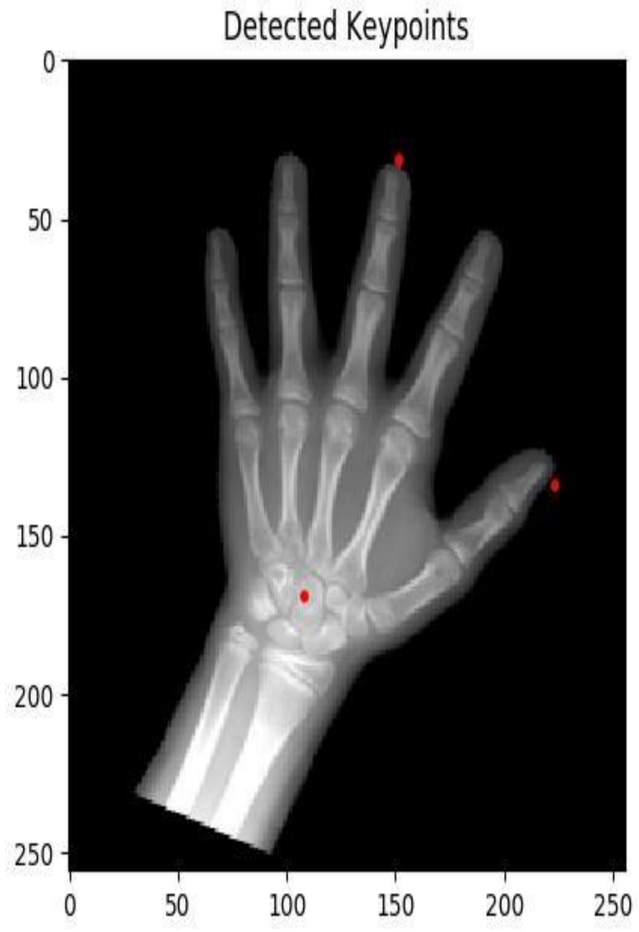


Figure 14: Key points detected by the network

Once we get the key points we rotate the image by calculating the inverse tangent of the line joining the tip of the middle finger and the wrist. The third keypoint of the thumb is used to make sure it is left hand. After rotation of the image, we crop the image by adding 20pixels above the middle finger and 50pixels below the wrist. We then resize the image back to the original image size.

4.2.2.1 VGG16:

VGG16 is a model made by the Visual Geometry Group at Oxford. It won the first and the second places in the localization and classification tracks respectively of ImageNet Challenge 2014. It is a simple yet effective model. The input to this model is our image and output is the 6 key-points.

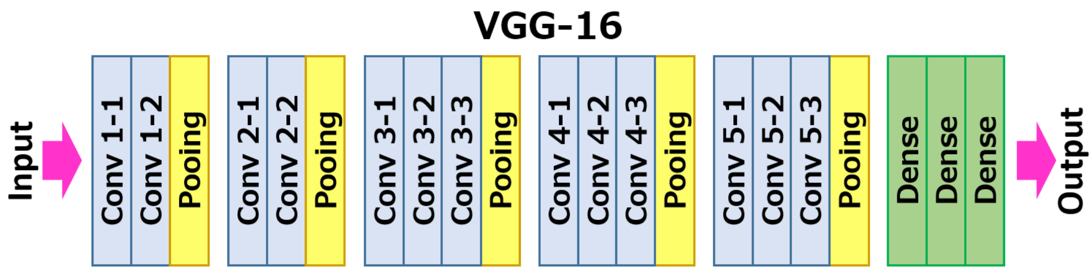


Figure 15: VGG Network Architecture

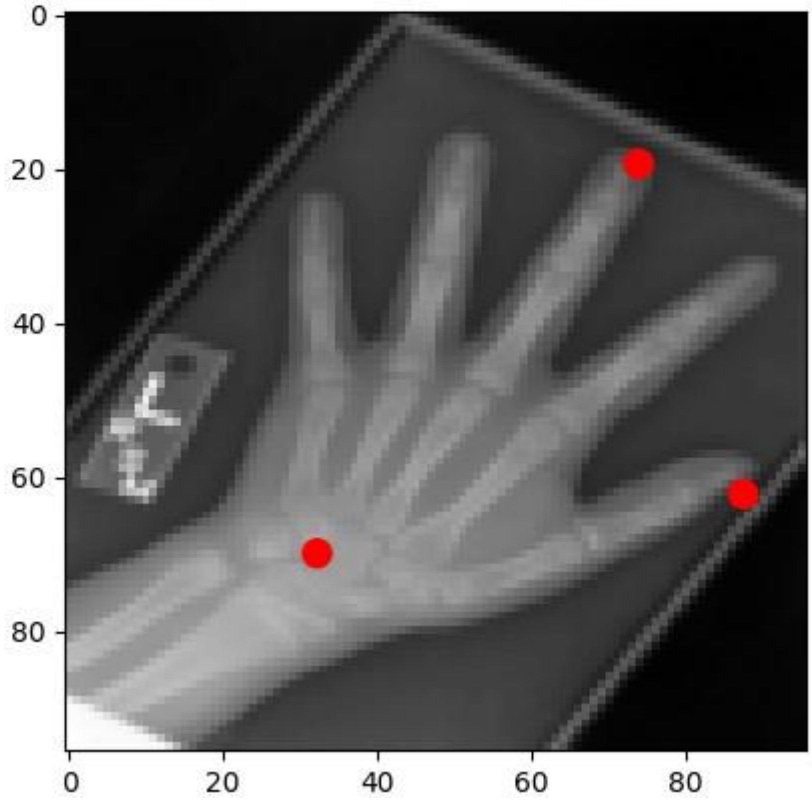


Figure 16: Key points Detected on a test image

4.3 Age Prediction:-

The final part of the system is actual age prediction. After the above preprocessing steps we get a standardized dataset of images. As per the literature, twelve thousand images is considered a small dataset, hence we have performed train time data augmentation to increase the number of images. Data augmentation is a method of making random transformations in the image to artificially increase the number of images. The random transformations include but not limited to brightness/contrast adjustment, random cropping, zooming, width shift, height shift, random rotation, horizontal/vertical flipping etc. The training is done using the state-of-the-art CNN architecture densenet. The network is trained from scratch as a pre-trained network on Imagenet dataset would not add any value since it did not contain any x-ray images. The densenet is followed by 2 fully connected layers and then a regression layer for predicting a continuous value. The network is trained using the mean absolute error, which is the difference between predicted age and target age.

4.3.1 DenseNet121:-

Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with L layers have L connections—one between each layer and its subsequent layer—our network has L(L+1) 2 direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

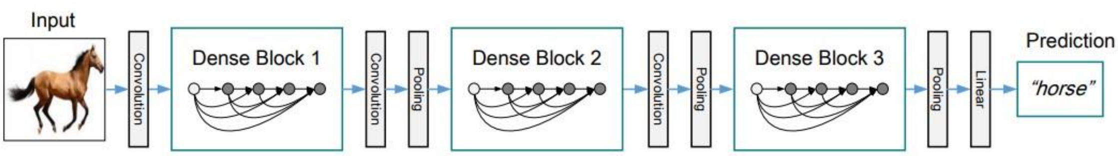


Figure 17: DenseNet Block Diagram

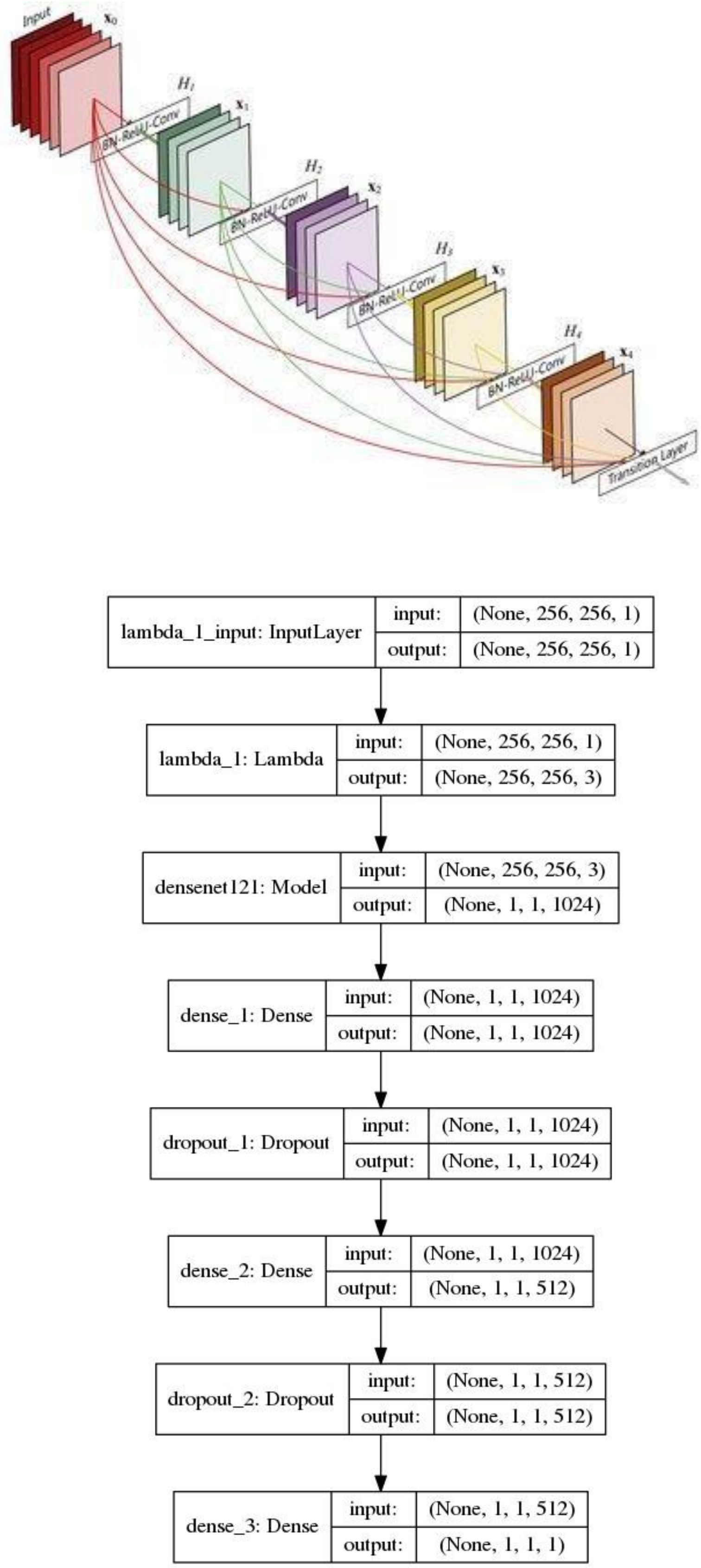


Figure 18: Our DenseNet Architecture

4.4 Front-End:

Django:

Django is a free and open source, high-level Python Web framework. In this project, we have used Python to support our machine learning application. This framework provided us with easy integration of our backend python script with the user interface. For designing our front end we used HTML, CSS and javascript.

The flow of data is as follows:

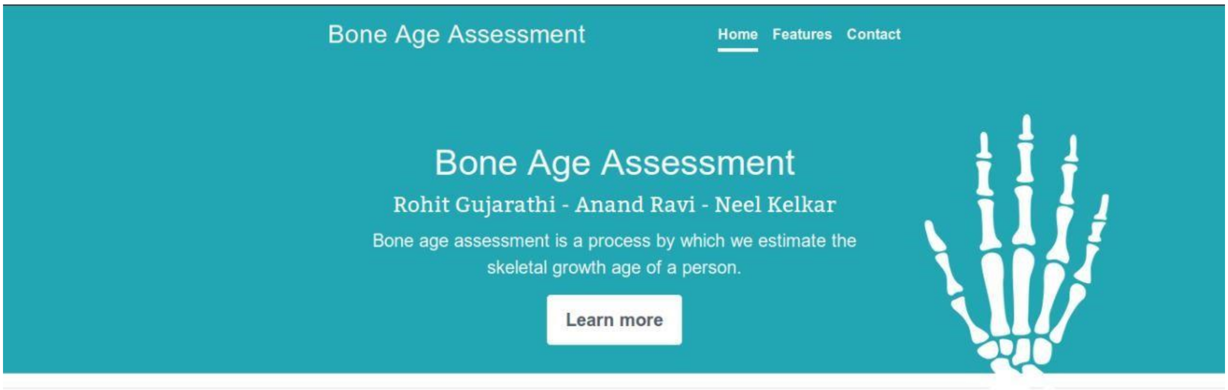
Front-end:

- Choose the image file located at the terminal by clicking the _Browse‘ button.
- Select the appropriate X-Ray image and click _Predict‘ button.

Backend:

- The file gets uploaded to the server and triggers the prediction function.
- The image goes through all the pre-processing stages of U-Net and VGG.
- The pre-processed image is sent to a DenseNet function which predicts the Bone Age.
- The result is sent back to the user terminal.

The whole process takes roughly 10 seconds on Dell Intel i3 CPU which is acting as the server. The processing time can be reduced significantly by using a computer powered by a GPU.



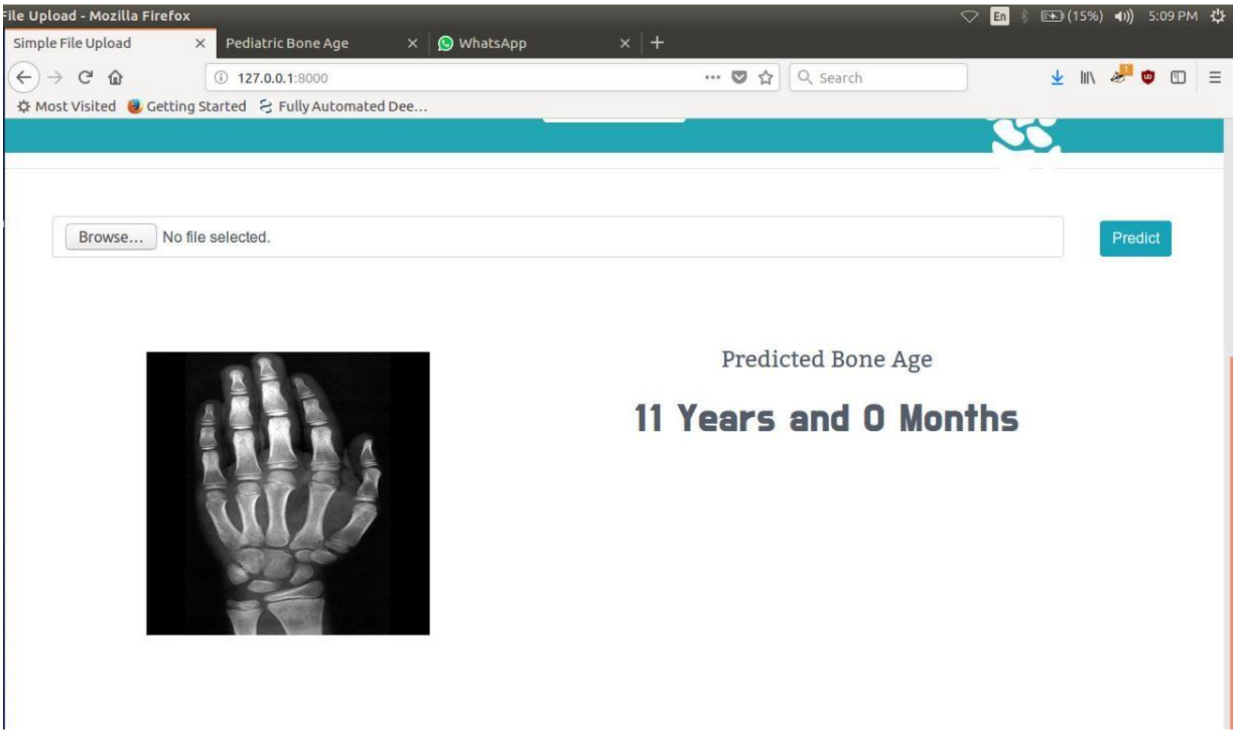


Figure 19: Front End

CHAPTER 5 SYSTEM IMPLEMENTATION

We have implemented the system in python using the popular deep learning framework keras. Since deep learning requires heavy mathematical computations the GPU is essential. We have trained the model on Nvidia Titan X GPU.

5.1 Training Stage

Step 1: Get images from provided path

Step 2: Resize the image to 512x512

Step 3: Train U-net

Step 4: Train VGG16

Step 6: Train Densenet121

5.2 Prediction Stage

Step 1: Get images from provided path

Step 2: Resize the image to 512x512

Step 3: Forward pass image through U-net

Step 4: Mask the image using predicted mask from U-net

Step 5: Forward pass output of U-net through VGG16

Step 6: Perform affine transformation

Step 7: Forward pass transformed image through densenet121

Step 8: Show predicted age

CHAPTER 6

Results

U-NET Results:

Number of Training images	300*
Number of Epochs	1000
Training Accuracy	96.48%
Validation Accuracy	94.26%

*initial training set before positive mining

Table 1: U-NET results

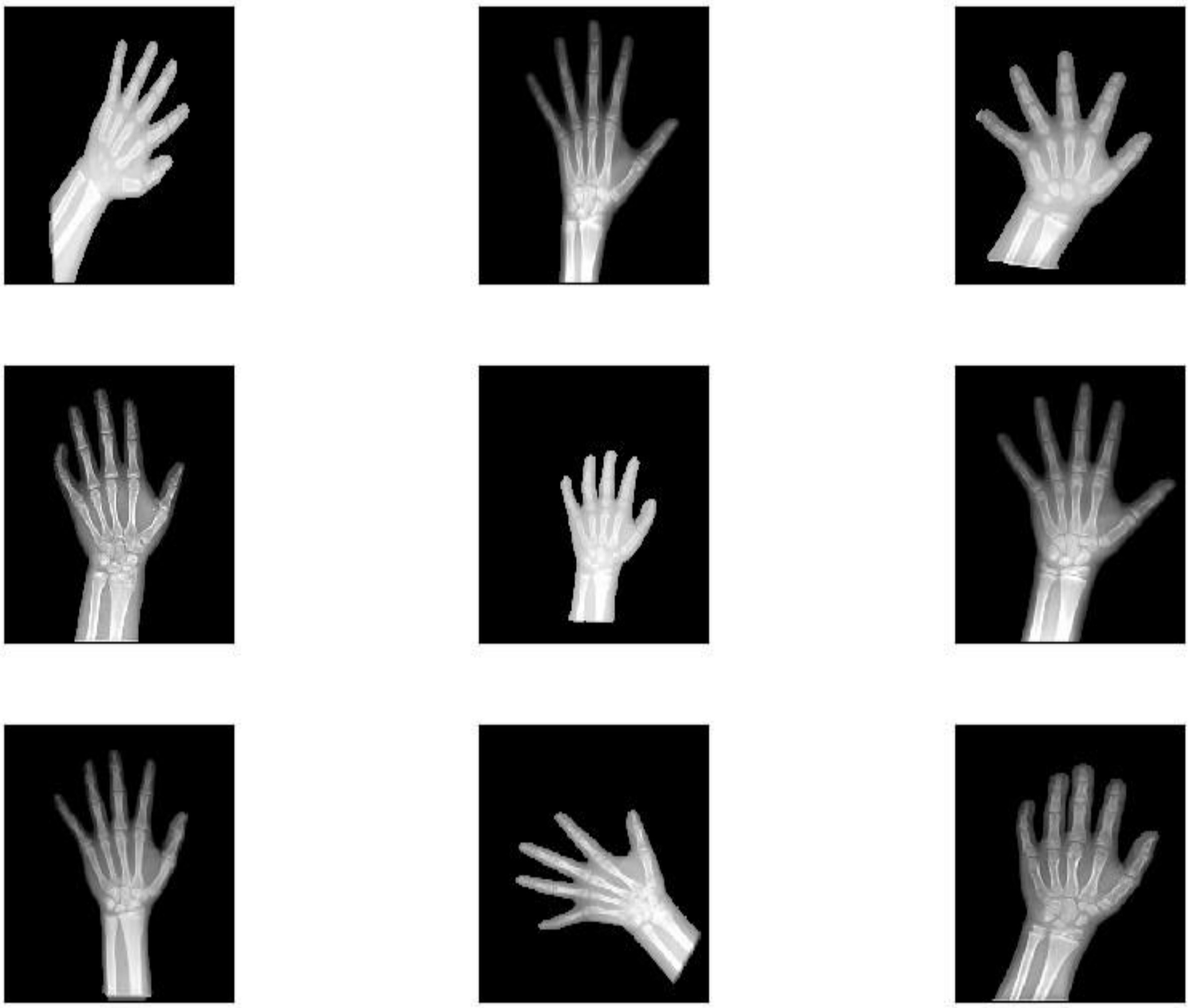


Figure 20: Sample images after Segmentation

VGG Results:

Number of Training images	600
Number of Epochs	1000
Training Accuracy	99.54%
Validation Accuracy	87.24%

Table 2: VGG Results

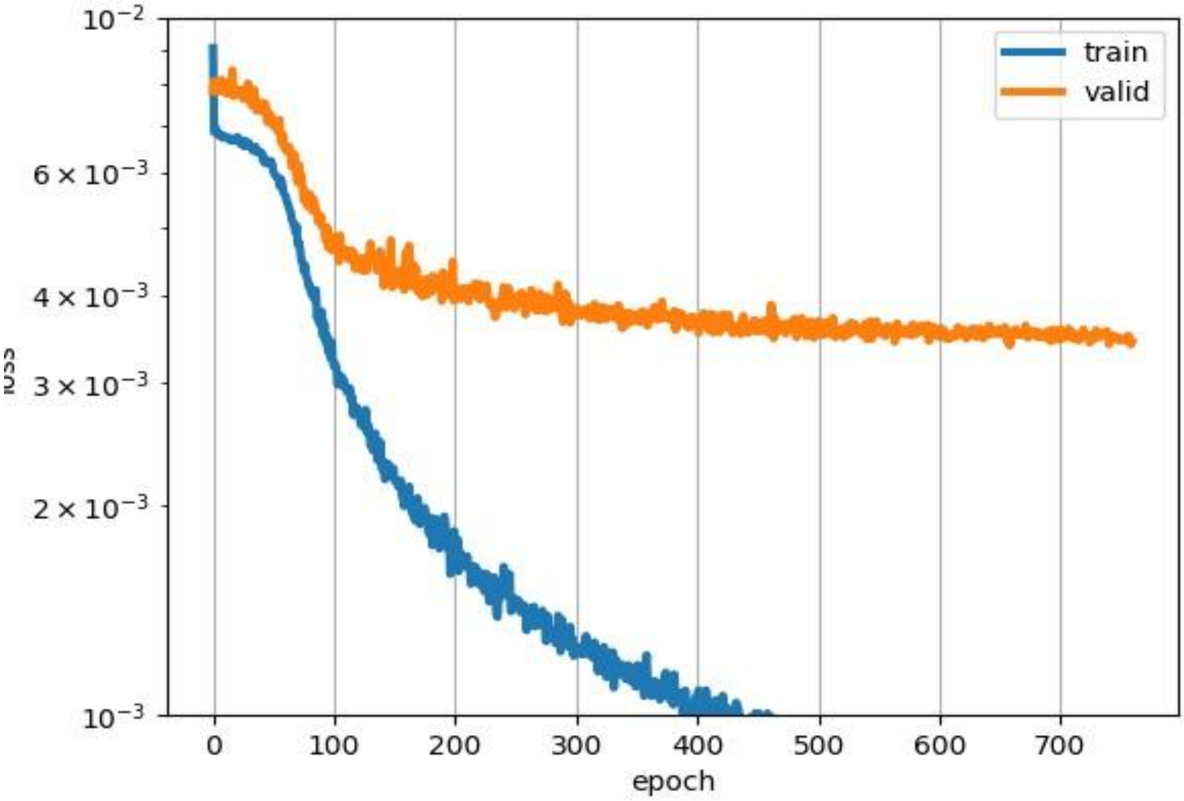


Figure 21: Loss function of VGG network

After segmenting the images and finding the key points we perform few operations on images to standardize them. They are:

Rotating the images

Cropping and zooming the images

Contrast Adjustment / matching

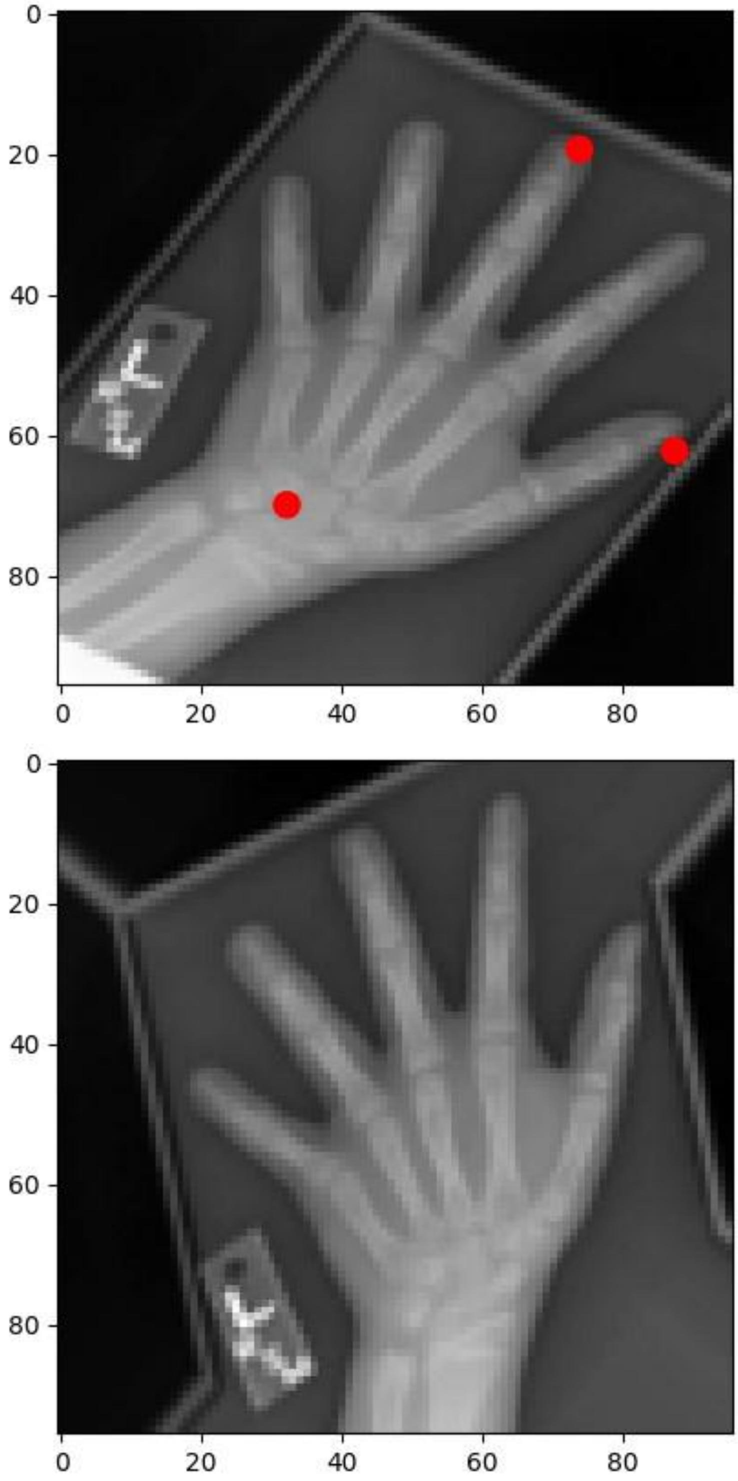


Figure 22: Example of Rotation.

Results after pre-processing engine:

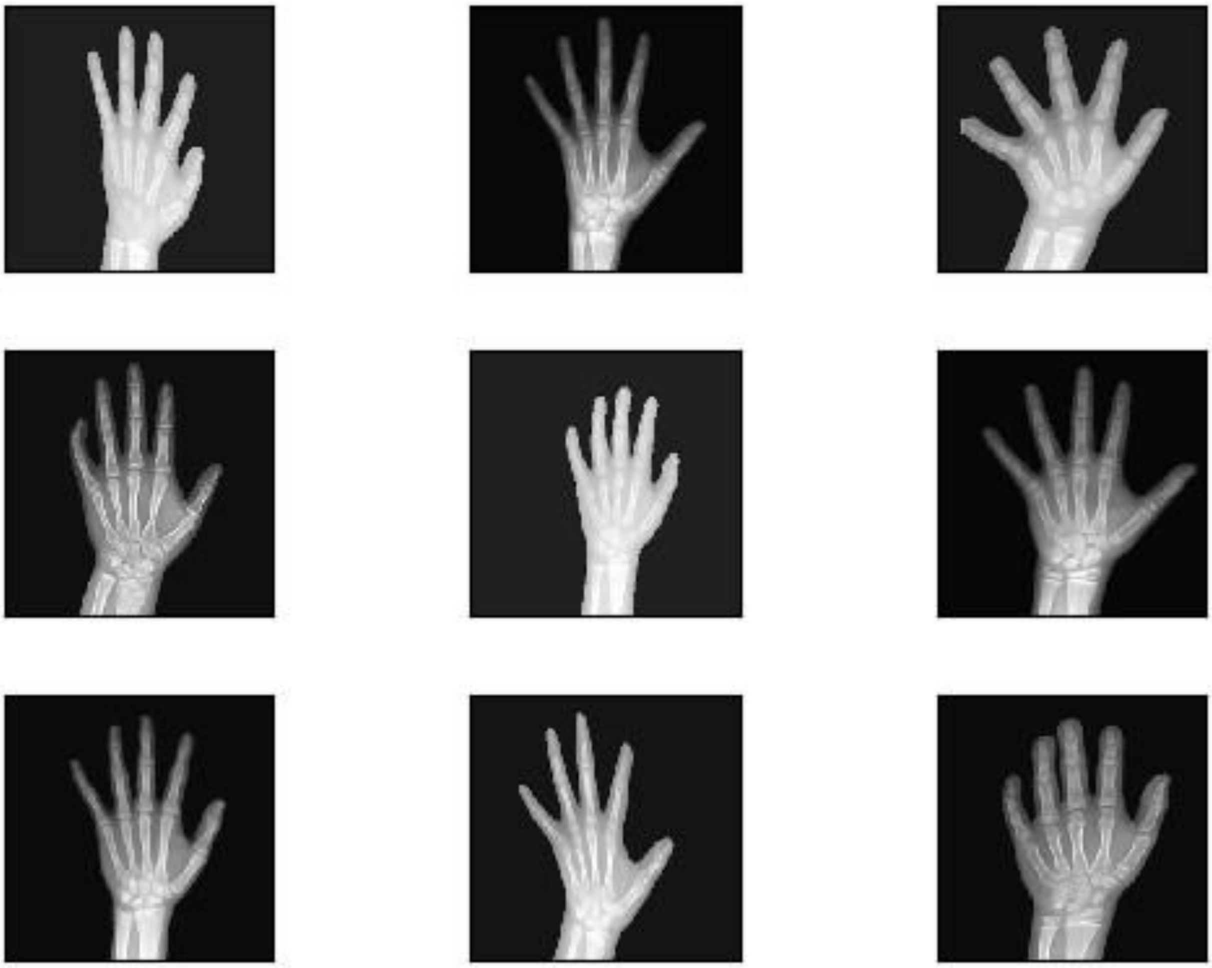


Figure 23: Sample Images after Pre-processing

DenseNet Results:

Training Data: 12k images

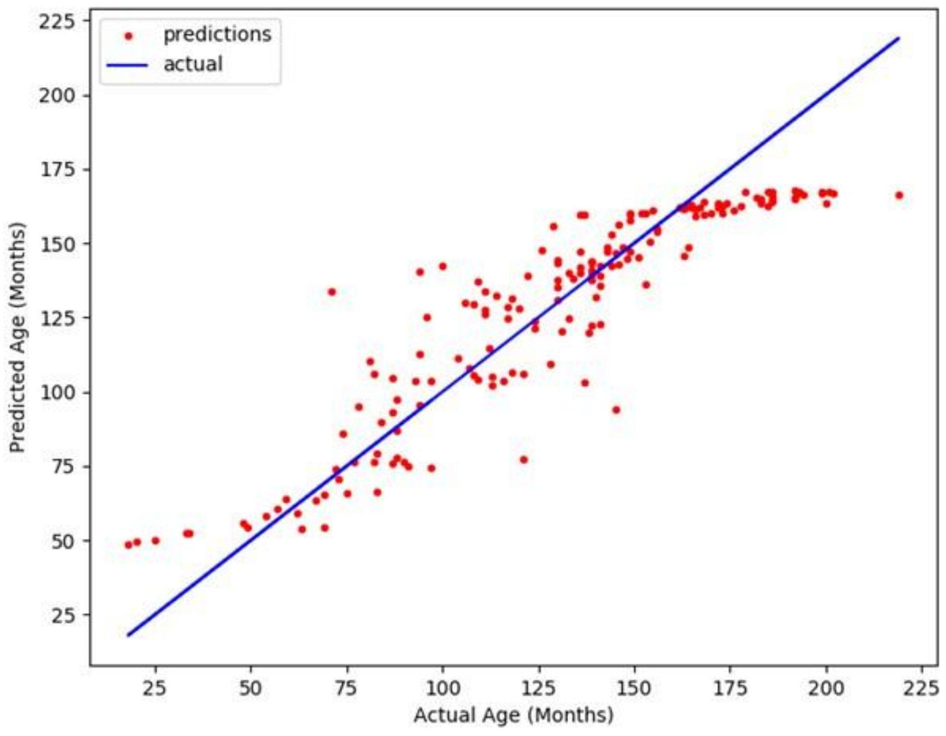
Testing Data: 200 images

CNN Architecture	DenseNet121
Mean Age Deviation on the training set	7.9 months
Mean Age Deviation on Test Set	12.2 months
Training Time Per Epoch	640 Seconds*

*Time Calculated on Nvidia Titan GPU

Table 3: Results of DenseNet

Scatter Plot:



CHAPTER 7 CONCLUSION AND FUTURE SCOPE

.

It is easier to develop a Bone Age Assessment system by leveraging the power of deep learning using CNN's rather than traditional image processing methods. CNN's have the potential to pick up on possible patterns or distinctive trends that may have been missed in traditional methods due to them being formulated through human observation. These methods do, however, form the foundation of the deep learning method because they confirm that a pattern does exist in skeletal development.

The system will be developed specifically for Indian people since there is no such existing system on the market. An Android App to provide Bone age can be developed for people with no domain knowledge using a user-friendly interface.

We have developed an automated Bone age assessment system using deep convolutional neural networks. Using this system the paediatricians can get results within seconds and with a degree of accuracy of up to 1 year. Further fine-tuning the system should provide us with an even higher accuracy.

CHAPTER 8
APPLICATIONS

This project aims to provide an easy and non-invasive approach to provide an automated Bone Age Assessment System which can be accessed freely by everyone across the world. The project will be running on a server which will be connected to the internet. By making this test mandatory, parents can get a rough estimate of the prevailing issue.

This framework is also compatible with other biomedical systems and by changing the training data we can use this model for a wide variety of applications.

REFERENCES

[1] A. M. Mughal, N. Hassan, and A. Ahmed, ―Bone Age Assessment Methods: A Critical Review,‖ Pakistan Journal of Medical Sciences, vol. 30, no. 1, Dec. 1969.

[2] C. Spampinato, S. Palazzo, D. Giordano, M. Aldinucci, and R. Leonardi, ―Deep learning for automated skeletal bone age assessment in X-ray images,‖ Medical Image Analysis, vol. 36, pp. 41–51, Feb. 2017.

[3] H. Lee et al., ―Fully Automated Deep Learning System for Bone Age Assessment,‖ Journal of Digital Imaging, vol. 30, no. 4, pp. 427–441, Mar. 2017.

[4] Michael Bowles, ―Machine Learning in python‖ , Wiley, 2016

[5] Olaf Ronneberger, Philipp Fischer, Thomas Brox -----U-Net: Convolutional Networks for Biomedical Image Segmentation May 2015.

[6] Karen Simonyan, Andrew Zisserman------- Very Deep Convolutional Networks for Large-Scale Image Recognition Sep 2014.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger----Densely Connected Convolutional Networks Aug 2016