TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report
On
AI Based Bone Age Prediction

Submitted By:

Anish Raj Manandhar	(Exam Roll No.: 31458)
Nabin Shrestha	(Exam Roll No.: 31473)
Prayush Bhattarai	(Exam Roll No.: 31482)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

May, 2024

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report
On
AI Based Bone Age Prediction

Submitted By:

Anish Raj Manandhar	(THA077BCT010)
Nabin Shrestha	(THA077BCT026)
Prayush Bhattarai	(THA077BCT035)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

In partial fulfillment for the award of the Bachelor’s Degree in Computer Engineering

Under the Supervision of

Er. Ganesh Kumal

May, 2024

DECLARATION

We hereby declare that the report of the project entitled **“AI Based Bone Age Prediction”** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Anish Raj Manandhar (THA077BCT010) ________________________

Nabin Shrestha (THA077BCT026) ________________________

Prayush Bhattarai (THA077BCT035) ________________________

Date: May, 2024

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled **“AI Based Bone Age Prediction”** submitted by **Anish Raj Manandhar, Nabin Shrestha** and **Prayush Bhattarai** in partial fulfillment for the award of Bachelor’s Degree in Computer Engineering. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled, and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of bachelor’s degree of Computer Engineering.

____

Project Supervisor
Er. Ganesh Kumal
Department of Electronics and Computer Engineering, Thapathali Campus

____

External Examiner
Er. Birodh Rijal
Everest Engineering College Sanepa, Lalitpur

____

Project Co-Ordinator
Er. Umesh Kanta Ghimire
Department of Electronics and Computer Engineering, Thapathali Campus

____

Er. Kiran Chandra Dahal
Head of the Department,
Department of Electronics and Computer Engineering, Thapathali Campus

May, 2024

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards the Institute of Engineering, Tribhuvan University for the inclusion of the minor project during bachelor’s in computer engineering. We are especially thankful to the Department of Electronics and Computer Engineering, Thapathali Campus for providing us with such an auspicious opportunity to work on this project.

We would also like to express our respect towards the authors and various developers whose works we have referenced in building this project. The articles served as a great source of knowledge and inspiration for all of us. The entire project wouldn’t have been possible without their help.

Finally, we would like to express our thankfulness to our supervisor **Er. Ganesh Kumal** and our friends who have been involved with us in the project in one way or the other. We are extremely grateful for their support, cooperation, help, guidance, and encouragement in doing this project.

Anish Raj Manandhar	(THA077BCT010)
Nabin Shrestha	(THA077BCT026)
Prayush Bhattarai	(THA077BCT035)

ABSTRACT

The bone age prediction system utilizes a two-stage approach without manual annotations. In the first stage, an Improved InceptionV3 with Convolutional Block Attention Module (CBAM) extracts crucial bone regions, emphasizing areas of interest. The second stage employs specialized models (ResNet50) to learn features related to carpal, metacarpus, and phalanx bones. The ResNet50 Method combines these features, enhanced by gender information. The final Fully Connected Layer computes precise bone age estimation. While doing this project we found that when the features i.e. carpal and metacarpal are combined the overall accuracy of predicting the bone age increased. This innovative architecture seamlessly integrates advanced neural networks, CBAM, and strategic input considerations, promising accurate predictions for clinical applications.

Keywords: Bone Age, CBAM, Datasets, Deep Learning, InceptionV3, ResNet50, Radiological Society Of Northern America (RSNA)

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BAA	Bone Age Assessment
CBAM	Convolutional Block Attention Module
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CR	Computer RadioGraphs
CV	Computer Vision
GP	Greulich and Pyle
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
MAE	Mean Absolute Error
ML	Machine Learning
NumPy	Numerical Python
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
RHPE	Radiological Hand Pose Estimation
ROI	Region Of Interests
RSNA	Radiological Society Of Northern America
UI	User Interface
TW	Tanner Whitehouse

1 INTRODUCTION

An essential part of the diagnosis process for many pediatric growth, metabolic, and endocrine diseases is radiographic bone age evaluation. Generally, radiographs of the hands and wrists are used to assess bone age. A characteristic measure of skeletal maturation in children and adolescents is bone age. The goal of the Bone Age Estimation Program is to create an automated system that uses X-ray pictures to determine an individual's skeletal maturity. Conventional bone age expectation systems require labor and time-intensive annotation of each photo. However, we don't need to emphasize nearly that with annotation-free models. Thus, the dataset might be used without any annotations. To aid healthcare professionals, the program will use deep learning techniques to analyze non-annotated X-ray images of the left hand and produce an estimate of bone age.

1.1 Background

The term "chronological age" describes a person's true age as of the day of their birth. On the other hand, bone age is an evaluation of a person's skeletal maturity that is derived from comparing a person's bone development to standard reference data, typically obtained from X-rays of the hand and wrist. The distinction between a person's chronological age and bone age might reveal important details about their growth and development. When a person's chronological age is less than their bone age, it indicates that their skeletal growth is further along than their actual age. On the other hand, delayed skeletal development is indicated if bone age is smaller than chronological age.

Assessment of bone age is a common procedure for managing a variety of pediatric syndromes and endocrine diseases, as well as for evaluating growth in pediatric patients. For many years, the Greulich and Pyle atlas has been the most widely used visual assessment tool for determining bone maturity when determining the skeletal development of the hand and wrist.

In the realm of pediatric healthcare, the assessment of bone age plays a pivotal role in understanding an individual's growth and development. Bone age estimation involves evaluating the skeletal maturity of a person based on X-ray images, particularly those of the hand and wrist. This process is instrumental in diagnosing growth disorders, predicting final adult height, and monitoring the progression of puberty.

Traditionally, healthcare professionals have manually assessed bone age by comparing X-ray images to standard reference atlases. However, the advent of artificial intelligence and deep learning has opened new avenues for automating this intricate task, offering the potential for increased accuracy, efficiency, and objectivity.

The goal of the Bone Age Estimation Program is to offer an automated and accurate method for determining bone age by utilizing cutting-edge machine learning techniques. The tool uses cutting-edge deep learning models to analyze X-ray images, extract relevant features, and estimate bone age with an accuracy that is on par with or better than conventional methods.

Annotation Free Bone age Assessment

Annotation-free bone age prediction takes the hard work of physically labeling pictures and lets the model learn straightforwardly from crude picture information instead of depending on human annotations. In conventional bone age expectation strategies, each picture is annotated, which is time-consuming and labor-intensive. But with annotation-free models, you don’t have to stress almost that. So, we could use dataset without annotations.

Over time and with sufficient preparing, the model will learn to recognize designs and connections in the picture information. As it learns, it learns to extract; bone age-related highlights like bone thickness, bone structure, development designs, etc. Based on these highlights, the model will consequently anticipate the age of an person based on what’s watched in the X-ray. While the model performance may change from time to time but it is indeed superior to conventional strategies that utilize annotated datasets.

In summary, the annotation-less bone age prediction terminates the manual labeling step and moves forward the versatility and cost-effectiveness of bone age prediction.

1.2 Motivation

Every parent is deeply invested in the well-being and development of their child, often concerned about potential hindrances to their growth, such as weak bone structure or height retardation. The challenge lies in identifying these issues early on, facilitating timely intervention and personalized care. Traditional methods of assessing bone age from X-ray images, while informative, are fraught with challenges, including time-consuming processes and inter-observer variability.

The motivation is rooted in the desire to provide parents and healthcare professionals with a cutting-edge tool that not only enhances diagnostic accuracy but also contributes to the holistic well-being of every child by addressing concerns related to growth and development.

1.3 Problem Definition

The existing manual methods for bone age estimation of children are time-consuming and may introduce variability among different practitioners. To address these challenges, the project aims to develop an automated system using deep learning techniques to accurately estimate bone age from X-ray images. The system will provide a quantitative measure of skeletal maturity, aiding healthcare professionals in making informed decisions about a child's growth and development. This project aims to address this challenge by developing an artificial intelligence (AI) system that leverages transfer learning with ResNet50 for accurate and efficient pediatric bone age prediction.

1.4 Objective

The main objective of our project is listed below:

- To implement an AI-based Bone age prediction model based on Left hand X-ray Images.
- To automate the process of annotating the X-Ray images for training.

1.5 Scopes and Applications

The Scopes and applications of our projects are listed below:

- It can be applied to support remote healthcare consultations.
- It can be used as a primary care support for monitoring the growth and development of children.

1.6 Report Organization

INTRODUCTION section of our report presents the information regarding the background of the project along with the project objectives, motivations, project scope and its applications.

LITERATURE REVIEW section of our report includes research papers, journals and articles dealing with the field of concern of our project like segmentation, evaluation metrics, machine learning models, etc. are analysed and discussed thoroughly.

In the section REQUIREMENT ANALYSIS, functional requirements and non-functional requirements of the project are listed. Various technological tools used in our project are also specified in this section.

In the section SOFTWARE ARCHITECTURE AND METHODOLOGY, we have described the block diagram and working methodology of our project. We have described about the machine learning architectures used in this project along with algorithm and the project's sequential flow.

In the IMPLEMENTATION DETAILS section, we have described the software development model of our project.

In the RESULT AND ANALYSIS section, results from different machine learning models have been analyzed and presented. The overall result is compared with the initial objectives of the project.

In FUTURE ENHANCEMENTS section, future updates and enhancements that can be made in this project have been described.

In CONCLUSION section, the project is summarized, and overall analysis of the project outcomes were made.

2 LITERATURE REVIEW

The D&P method and the TW method are the two approaches now available for determining bone age. Both involve labor-intensive manual labor that takes time. Consequently, there is currently a growing interest in automated assessment systems for BAA (Bone Age Assessment). The deep learning techniques for BAA overcomes the issue of their time-consuming nature, the subjectivity of human interpretation, and variances between and among operators.

In 2009, the business Visiana introduced the BoneXpert method (Visiana, Hørsholm, Denmark) for automated bone age determination [1]. The purpose of the BoneXpert is to determine an individual's bone age in lieu of human experts. It was evaluated independently using no additional techniques. The image analysis makes use of a conventional technique to estimate an individual's bone age based on the appearance, brightness, and texture of their bones. With the exception of tiny bones, the approach attempts to identify nearly every bone in the hand and wrist and determines its age. Should a bone's appearance deviate from what the machine learning algorithm has learned, or should its age deviate significantly from the mean age of all the

In the “Residual attention based network for hand bone age assessment” [2], The framework is based on how doctors assess the age of a person's hand bones. It focuses on the important parts of the hand. The plan has two parts: 1. A Mask R-CNN to figure out the boundaries of the hand. 2. A residual attention network to estimate the age of the bones in the hand. The Mask R-CNN helps separate the hands in X-ray images from other objects (like X-ray tags) so they don't get in the way. The parts of the residual attention subnet help the network to pay attention to important things in the X-ray images and make predictions, like how doctors look at X-rays. It tests how well the new system works with the RSNA pediatric bone age dataset.

In the article by Iglovikov V., Rakhlin A., Kalinin A., Shvets A. titled, “Pediatric Bone Age Assessment Using Deep convolutional Neural Networks” [3], The paper talks about using a computer program to help doctors figure out a person's bone age. They used data from a challenge in 2017 to do this. This competition has 12,600 x-ray images as its dataset. Every x-ray picture in this group shows the bones of a left hand

and tells how old the person is and whether they are male or female. This method uses different types of deep neural network designs that are trained all at once. This method uses pictures of entire hands and certain hand parts for practicing and making predictions. This method helps us figure out which hand bones are most important for analyzing bone age automatically. This method worked well, with men having an average error of 6.3 months and women having an average error of 6.49.

The research in this paper by T. F. D. L. Escobar M., "Hand pose estimation for pediatric bone age assessment," [4] An innovative method of measuring bone age using hand x-rays by closely examining particular locations was presented at the International Conference on Medical Image Computing And Computer-Assisted Intervention. Using 6,288 x-ray pictures of hands, they created a novel dataset called Radiological Hand Pose Estimation (RHPE). This dataset differs from the ones that are already accessible. The BoNet model is superior to existing BAA techniques and makes use of local information.

The method in the research titled “Attention-based multiple-instance learning for pediatric bone age assessment with efficient and interpretable” [5] involves cutting pictures into small parts and finding out what is special about each part using a special computer program. This method can find important areas in a picture without needing extra labels and create maps that are easy to understand. The new model can see the small details in the picture better and train faster by cutting the picture into smaller parts and making it simpler. This method was checked using the RSNA dataset. This method had an average error of 4.17 months. By adding an attention mechanism, the model can see which parts are important for its decision, making it easy to understand. Additionally, the findings show that the suggested model pays attention to small areas like what humans already know, which boosts the model's trustworthiness.

This method titled, “Ridge Regression Neural Network for Pediatric Bone Age Assessment “[6] also extracts ROI features from pediatric hand images, followed by using a learning model to estimate the bone age these radiographs. This process is separated into two parts. In the first step, the x-ray images are edited using labels and cut them into smaller parts. In the next step, the new deep learning model is taught using separated X-ray images to see how well it works. The VGG image annotator helps mark and describe specific parts of a picture for image segmentation. Next, the objects in the picture are separated using the mask RCNN. This model uses a pretrained VGG-19 neural network with an additional output layer for making predictions. The ridge regression layer tries to find the best values for the regression coefficients without the problem of multicollinearity. Multicollinearity in a regression model happens when some predictor variables are related to each other, which can cause the regression coefficients to have more variation and become less reliable. This study was also done using the RSNA dataset.

A rectified linear unit (ReLU) activation function and a pre-trained VGG-19 convolutional neural network with an input image size of 512 x 512 x 3 are used for feature extraction. The mean square error was used as the loss function during the 160 epochs of training for the regression network using the Adam optimizer with a learning rate of 10-4 and batch size of 32. The lowest MAE of 6.38 months was attained with this strategy.

In 2001, a system was proposed titled, “Computer-Assisted Bone Age Assessment: Image Preprocessing and Epiphyseal/ Metaphyseal ROI Extraction“ [7]. The data for this study was obtained from the Los Angeles-based USC Children's Hospital. X-rays of normal, healthy people's left wrists were included in the data. Radiologists could view standard, prepared images thanks to the usage of preprocessing functions. Computer x-rays and digital images were the two types of images that were employed. It was necessary to align and fix the CR photos and to standardize their content. It was just necessary to convert the images to a new file format and make brightness and contrast adjustments; rotating them was not necessary. CNN or any other neural network is not used in this method of determining how old your bones are. A grid pattern is overlay on the image to identify key areas.

The dataset used in the study titled “Pediatric Bone Age Assessment using Deep Learning Models," [8] First used in the Radiological Society of North America (RSNA) 2017 competition was Manipal Institute of Technology. The hands of individuals aged 1 to 288 months are scanned with X-ray technology in this dataset. There are 12611 distinct hand scans in the training data set. The age and gender of each scan are matched. There are 2000 photos for validation and 6000 images for training in this model. This indicates that testing uses up 25% of the data. The initial step after obtaining a collection of images is pre-processing. Here, the images are created using the appropriate preparation parameters, and each trained model is given the appropriate image size. Using image augmentation, one can alter images by making adjustments, such as enlarging or contracting them. The reason behind this is that the majority of the images in the collection feature left hands. To further complicate the dataset, the training images are randomly rotated horizontally. Following data preparation, each training image's true age labels (measured in months) are gathered and supplied to the CNN model for training. The model will next need to be trained using the pre-treated images and their accurate labels. To assist in model training, this approach makes use of several pre-existing models, including VGG-16, InceptionV3, MobileNet, and XceptionNet. The model will next need to be trained using the pre-treated images and their accurate labels. To assist in model training, this approach makes use of several pre-existing models, including VGG-16, InceptionV3, MobileNet, and XceptionNet. Maintaining the same model structure while training the entire model from start to finish is one method of transferring learning. To avoid overfitting and minimize mistakes, they replaced the final SoftMax layer in all pre-trained models with layers of batch normalization, global average pooling, and dropout before the fully connected layer. In the end, the regression results were obtained by using a single neuron in the output layer.

This research, titled "A Fully Automated Deep Learning System for Bone Age Estimation," [9] achieved 90.38% accuracy at 1 year in women, and 2-year accuracy was 98.11% of the time. For men, an x-ray was ordered in 94.18% within 1 year and in 99.0% within 2 years. Using the input-occlusion method, it produced attention maps that show the properties that the trained model uses to produce the BAA. These are the things that human experts consider when doing BAA manually. Finally, the BAA system is now used in hospitals to help doctors make better and faster decisions about breast cancer. It works much faster than the old way, taking less than 2 seconds instead of more than 10 minutes. The inputs were images in DICOM format with a wide range of brightness, color and grayscale. It is important to standardize the images before using them in the model, so that the model performs well and excess noise is removed.

Highlights of bone improvement studied in determining bone age include bone proximity (some bone continues to solidify), bone grade and shape, amount of mineralization (also called ossification), and the degree of fusion of the epiphyses and metaphyses. At birth, long bones have distinct coagulation centers that continue to multiply until the terminal or epiphyseal portion of the bone is completely fused with the diaphyseal portion. This handle is undoubtedly influenced by different variables, insulin-like growth factor-1. In addition, a lack of thyroid hormones or an abundance of corticosteroids causes a decrease in cell division in the reproductive zone, which causes developmental delay. Unlike hormones, gender can also affect this handle. In particular, the skeleton progressed more in women than in men of the same chronological age. In reality, the bone development process takes longer in men than in women, and young women typically close the epiphyseal region 2 years earlier than boys. Thus, the carpus does not ossify at birth, and this handle usually begins before the center of ossification. Usually, the ossification center appears in females around the head and teeth, and in males around the fourth month, and remains as a valuable noticeable point for another 6 months. . The remaining centers are then continuously displayed. Estimates of skeletal development in pubescent children are based primarily on measurements of the epiphyses of the phalanges as they relate to the adjacent metaphyses. During this stage of progression, the centers of consolidation of the epiphyses increase in width and thickness and are as wide as the metaphyses. During adolescence, the epiphyses begin to cover or cover the metaphysis. From there on, pisiform and sesamoid ended up recognizable. Amid late adolescence, the combination of the epiphyses to the metaphyses in the long bones of the hand tends to happen in a characteristic pattern:

- fusion of the distal phalanges,
- fusion of the metacarpals,
- fusion of the proximal phalanges,
- fusion of the middle phalanges. [11]

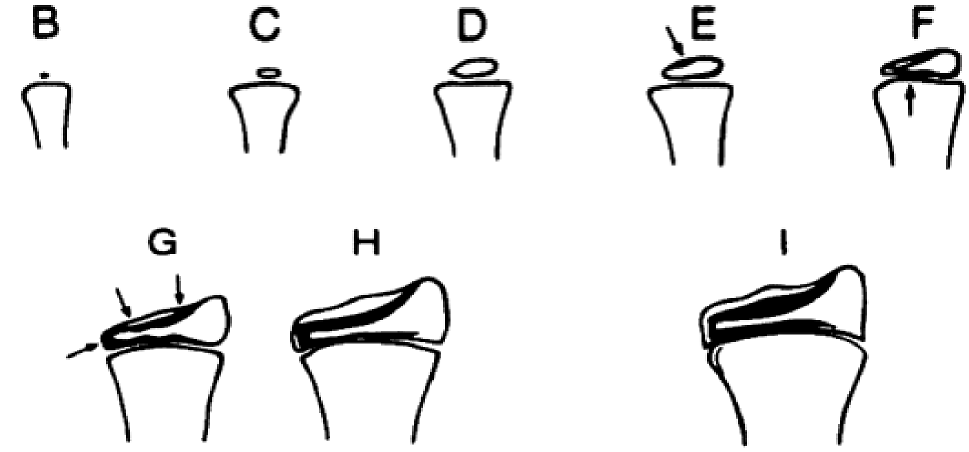


Figure 1: Stages of Bone Development

The GP atlas is the method used the most in the world. It was printed in 1950 and updated in 1988. The atlas has pictures of men's and women's left wrists taken every year. The person's bone age is determined by comparing their wrist x-rays with the pictures in the bone age atlas. The TW method [10] was made using x-rays of kids' left wrist from England with a middle socioeconomic status. In this method, it looks at the bones in the arm and hand to check if they are healthy. Some of these bones are sorted into groups A-I, and then it adds up how many there are in total. Based on this number, BA is calculated using the right value for the person's age and gender.

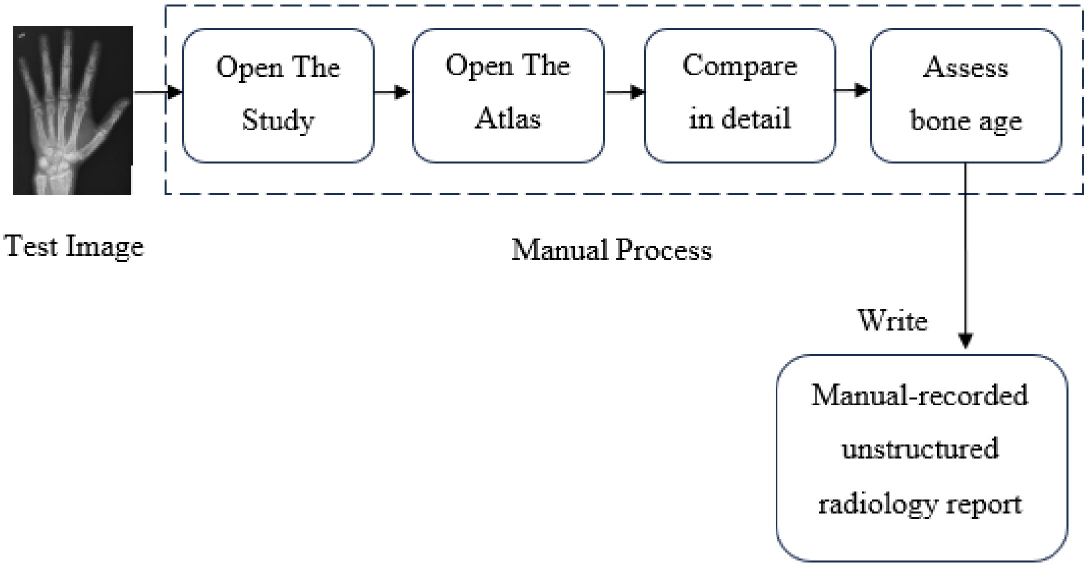


Figure 2-2: Block Diagram of Manual Bone Age Assessment

3 REQUIREMENT ANALYSIS

3.1 Functional Requirements

Design an intuitive dashboard with features for:

- Uploading left hand X-ray images.
- Implement drag-and-drop functionality.
- Validate image formats and sizes upon upload.
- Initiating bone age predictions.

3.2 Non-Functional Requirements

- Performance

Provide bone age predictions from left hand X-ray images within a maximum response time of 10 seconds.

- Accuracy

Achieve a minimum accuracy rate of +/- 1 year for bone age predictions from left hand X-ray images.

- Usability

The user interface is designed for ease of use and requires minimal training for medical professionals or the general population.

- Robustness

The system has been designed to be robust against variations in left hand image quality, different X-ray machine outputs, and variations in hand positioning.

3.3 Software Requirements

1. Python

Python is an advanced, general purpose, interpreted, dynamic programming language that supports a variety of programming paradigms, such as procedural, imperative, functional, and object-oriented programming.

2. Numpy

Numpy is a high-performance programming extension for python. It works excellently with the higher dimensions array.

3. Pandas

Pandas is an effective tool for analyzing and manipulating data. It is capable of loading, cleaning, transforming, and analyzing data from a variety of sources, including Excel spreadsheets and CSV files.

4. Tensorflow

Tensorflow provides a comprehensive ecosystem of tools and libraries for developing machine learning models, including high level APIs like Keras for building neural networks with ease.

5. Keras

It provides a simple interface for constructing neural networks by assembling building blocks like layers, activations, optimizers, and loss functions. Keras also supports both convolutional and recurrent neural networks and can be seamlessly integrated with other libraries and frameworks for comprehensive deep learning workflows.

6. OpenCV

It provides a wide range of functionalities for image and video processing, including reading and writing images and videos, performing basic and advanced image manipulation, object detection and recognition, feature extraction, and more.

7. Streamlit

It is an open-source Python library that simplifies the process of building web applications for machine learning and data science projects. With streamtlit, developers can create interactive and customizable web apps using only Python scripts, without needing to write HTML, CSS, or JavaScript. It provides a clean and intuitive interface for designing and deploying data-driven applications, allowing users to visualize data, explore models, and interact with machine learning algorithms in real-time.

8. Matplotlib

With Matplotlib, users can create various types of plots including line plots, scatter plots, bar charts, histograms, pie charts, and more. It supports both 2D and 3D plotting and can be seamlessly integrated with other Python libraries such as NumPy and Pandas for data manipulation and analysis.

3.4 Hardware Requirements

Since the project is based on using software applications, the only hardware required is a PC with at least 4 GB RAM and a dedicated GPU.

3.5 Feasibility Study

3.5.1 Economic Feasibility

The requirements of this project are software based. The costs involved with the project will be domain hosting, Google Colab, etc. Besides that, there will be no sophisticated hardware involved and hence the project is economically feasible.

3.5.2 Technical Feasibility

The libraries of python like pandas, Numpy, OpenCV, TensorFlow, Keras etc. had been used for programming. The libraries were sufficient for accomplishment of the project, so the system can be termed technically feasible.

3.5.3 Schedule Feasibility

The project can be accomplished within the allocated time and will be done according to the timestamps marked within the Gantt chart.

3.6 Dataset Analysis

The dataset used for this project was obtained from Kaggle which was hosted by Radiological Society of North America (RSNA) as a part of the research and competition. It constituted a comprehensive collection of 14,236 left handbone radiographs. This dataset was divided into three distinct categories: 12,611 samples allocated for training, 1425 for validation, and 200 for testing.

Within the training subset, a balanced distribution across genders with 5,778 images were of Female and 6,833 were of Male categories.

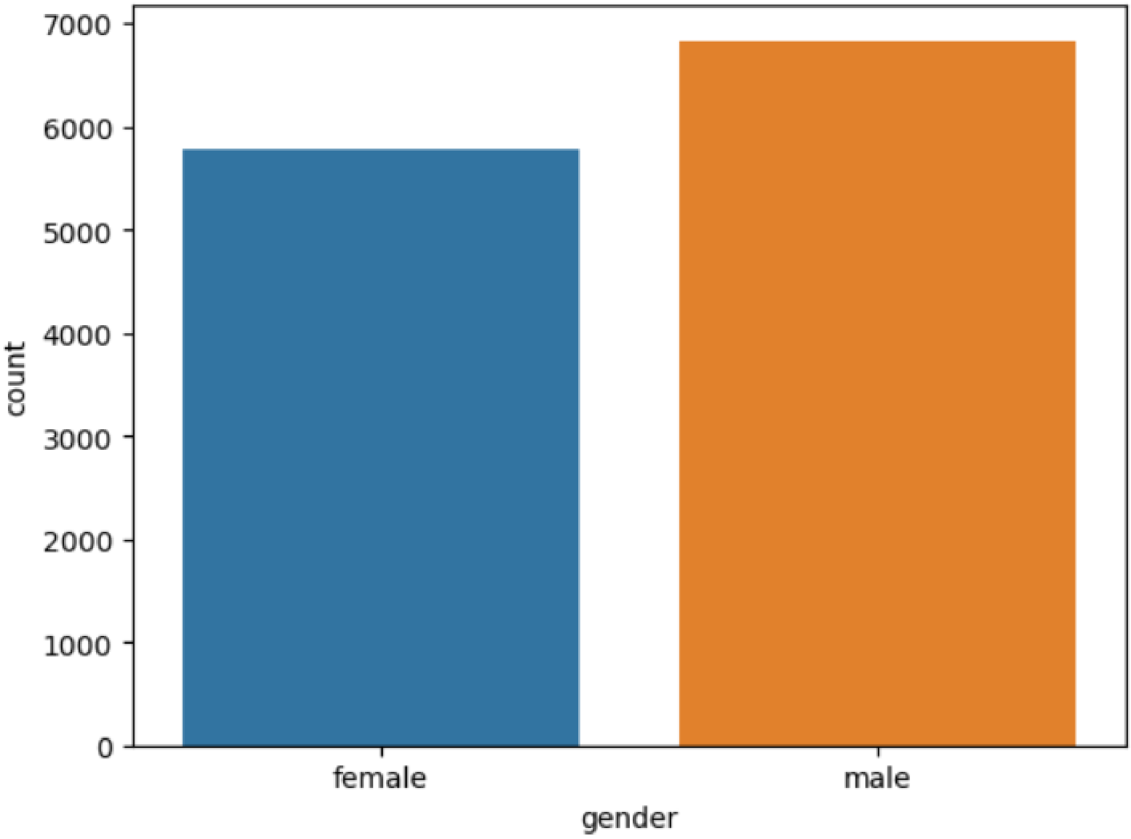


Figure 3-1: Gender Distribution of Datasets

This Gender balanced dataset ensured that our trained model will not be biased and can effectively learn from diverse demographic representations, thereby enhancing its robustness and generalization capabilities.

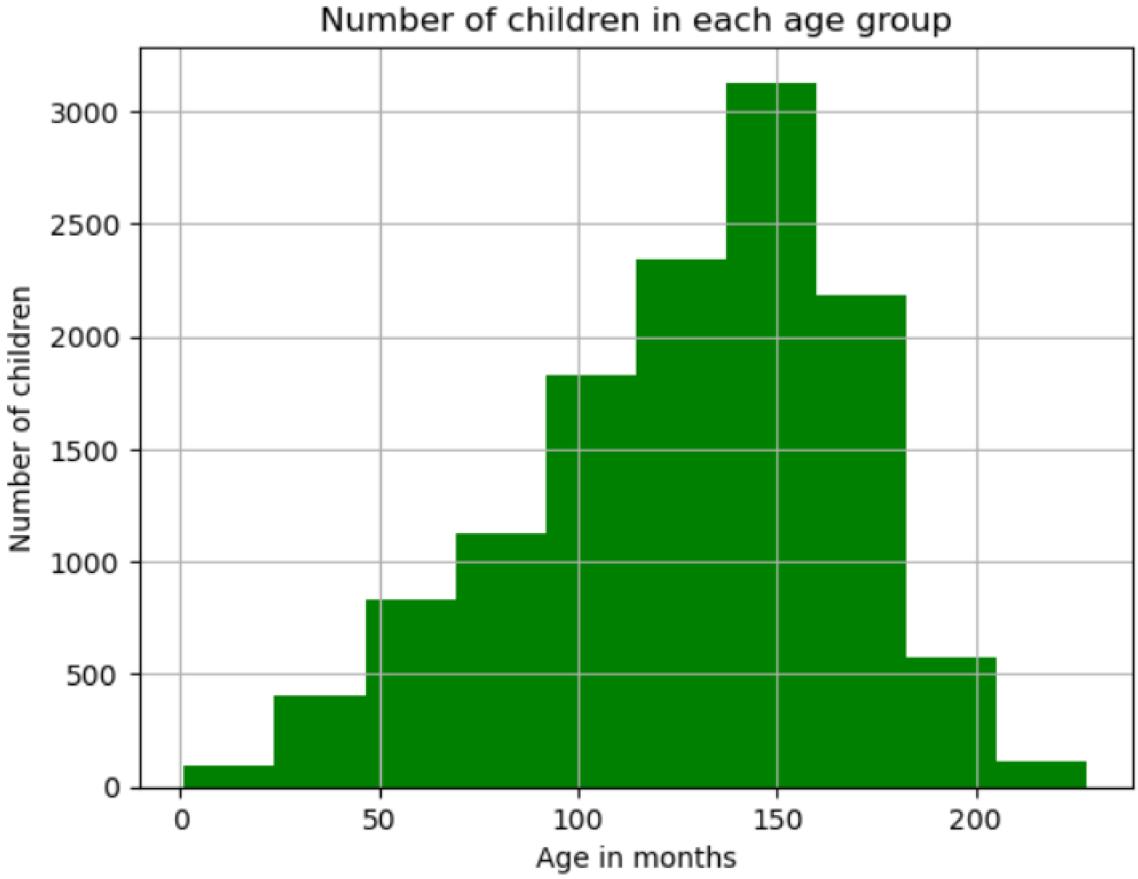


Figure 3-2: Age Distribution of Datasets

Table 3-1: Data Statistics

S.N.	Statistics Measures	values (months)
1	Mean	127.32
2	Median	132
3	Standard Deviation	41.28
4	Max age	228
5	Min age	1

$$\text{Z score} = \frac{x-u}{\sigma} \tag{3.1}$$

Where x = datapoint of bone age

u=Mean of bone age

σ=Standard Deviation of bone age

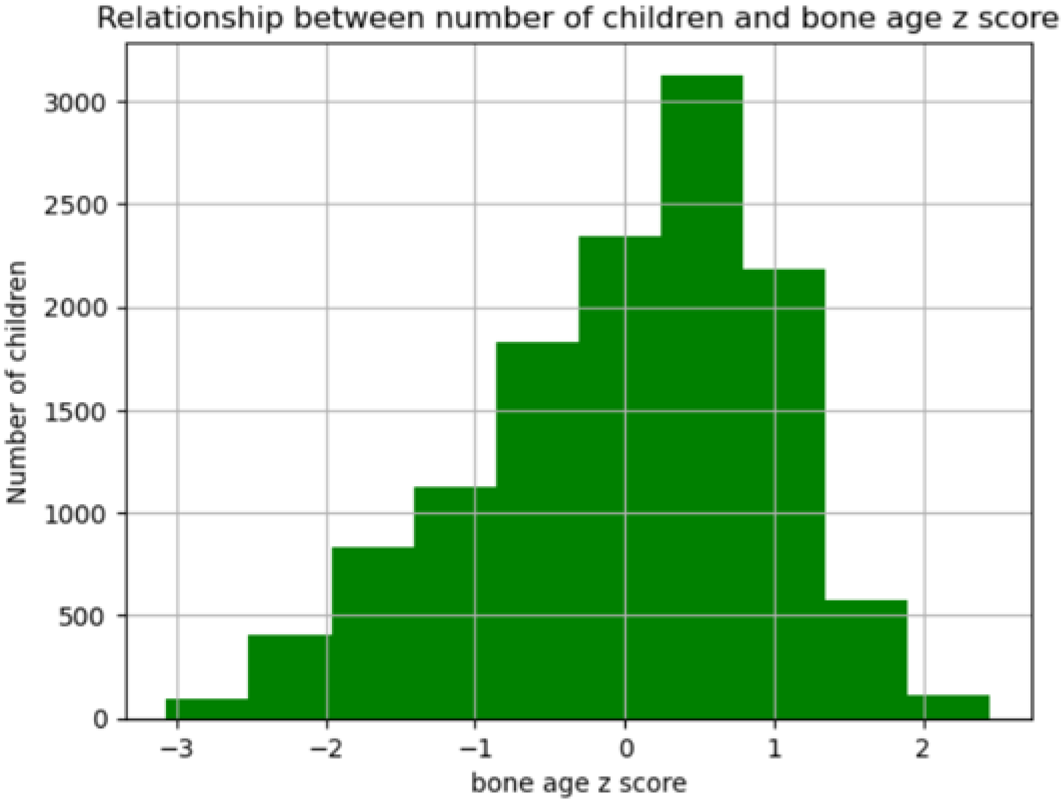


Figure 3-3: Plot of bone age Z-Score

From above histogram, it shows that the calculated z score of all the data points lies within [-3, 3]. This indicates that our data points are well within the standard deviations of the mean, suggesting that there are no extreme outliers. Hence, we can proceed with this dataset to feed into our model and train it to predict bone ages effectively.

4 SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Block Diagram

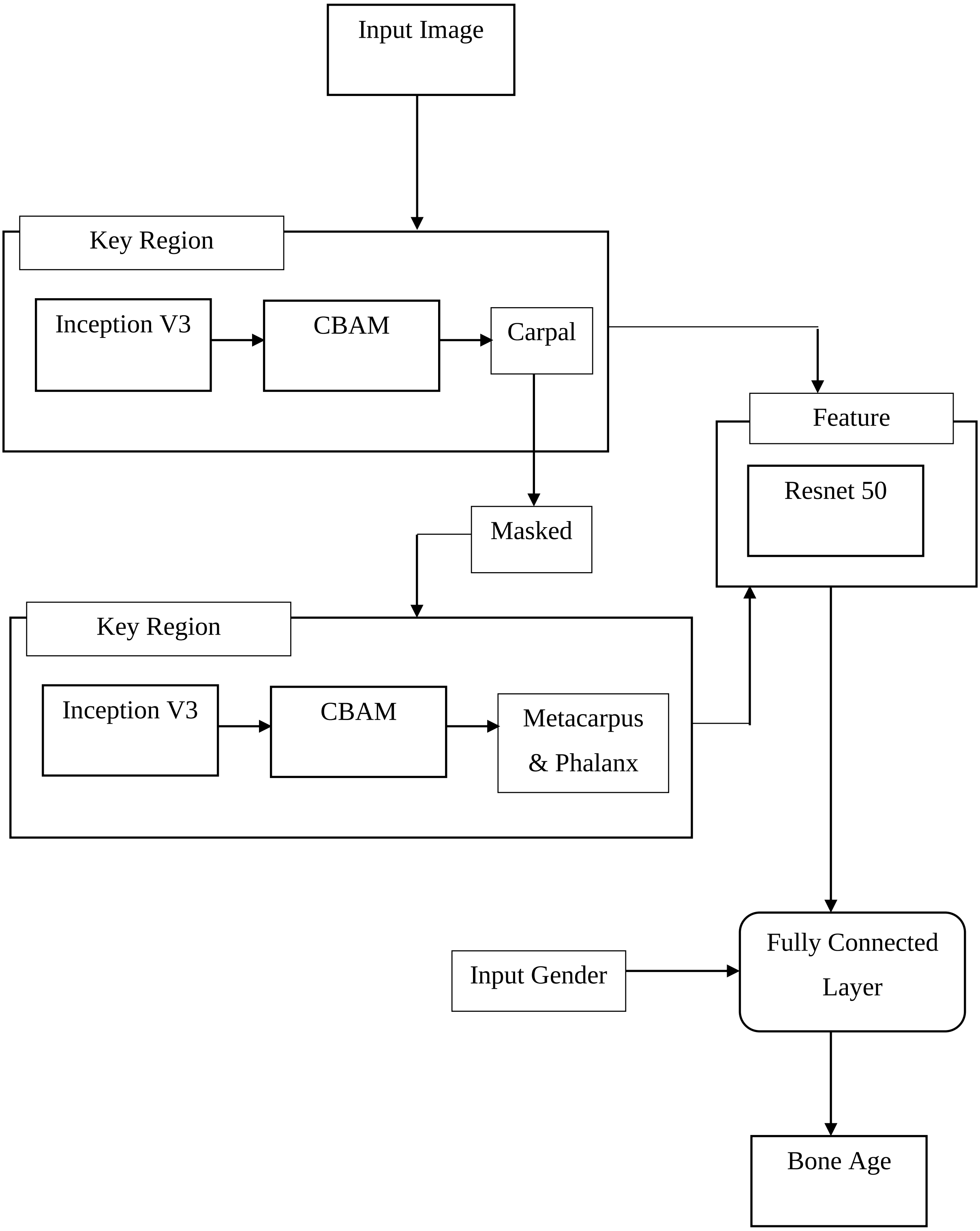


Figure 4-1: System Block Diagram

4.2 System Architecture Description

4.2.1 Critical Bone Region Extraction

First, the image is given as input. The shape of the input image is 299*299*3. This shape is needed because the image is passed to the Inception V3 model which performs well for this image size. This part of the system is responsible for the localization of region of interest in the X-Ray images of the left hand. Here, in this part the image given as input is passed to Inception V3 model. Then, the output of the last convolutional layer of the Inception V3 model is given as input to the Channel Block Attention Module (CBAM). CBAM then outputs a feature map of size 8*8*2048. This output is then used to produce the heat-map of the image. Then, by using the heat-map we crop the image where the attention is high. This step is expected to localize the area of the carpal in the X-Ray as the most features the model will learn will be from this area. After that the cropped part of the original image is replaced with the black mask. And again, the masked image is passed through the network of Inception V3 + CBAM. As black mask is applied to the area of the carpal another area from which the Inception V3 learns the most is the metacarpal region. So, a heat-map for the masked image is produced. Now, the heat-map has most attention in the metacarpal region of the masked image. Then, again by using the heat-map we crop the original image where the attention is high. At this moment of time, we have the cropped images of the carpal region and metacarpal region of X-Ray images. Hence, by using the attention mechanism we got the region of interest in the X-ray images which are crucial for the prediction of the bone age. Various modules in this part are explained below:

4.2.2 Inception V3

The InceptionV3 is a sophisticated convolutional neural network (CNN) model. This model is distinguished by a complex architecture that includes 350 connections and 316 layers in total. 94 convolution layers with different filter widths are included, and the first input layer has dimensions of 299 × 299 × 3. Using both symmetrical and asymmetrical construction components, InceptionV3's design achieves a well-optimized local topology. A variety of convolution operations, average pooling, max pooling, concatenations, and fully connected operations are all included in each block of the architecture.

4.2.2.1 Inception V3 Architecture

The network begins with a series of convolutional and pooling layers for initial feature extraction, gradually reducing the spatial dimensions of the input image while increasing the number of channels. The core of InceptionV3 consists of multiple Inception modules, each comprising a stack of convolutional layers with different kernel sizes (including 1x1, 3x3, and 5x5) and pooling operations. These modules are designed to capture both local and global features effectively by utilizing parallel convolutional pathways. InceptionV3 incorporates dimensionality reduction techniques such as 1x1 convolutions and pooling layers to reduce the computational complexity of the model while preserving important features. These techniques help prevent over-fitting and improve model efficiency. InceptionV3 includes auxiliary classifiers at intermediate layers of the network, which are used during training to encourage the propagation of gradients and improve gradient flow. These classifiers aid in addressing the vanishing gradient problem and contribute to more stable training.

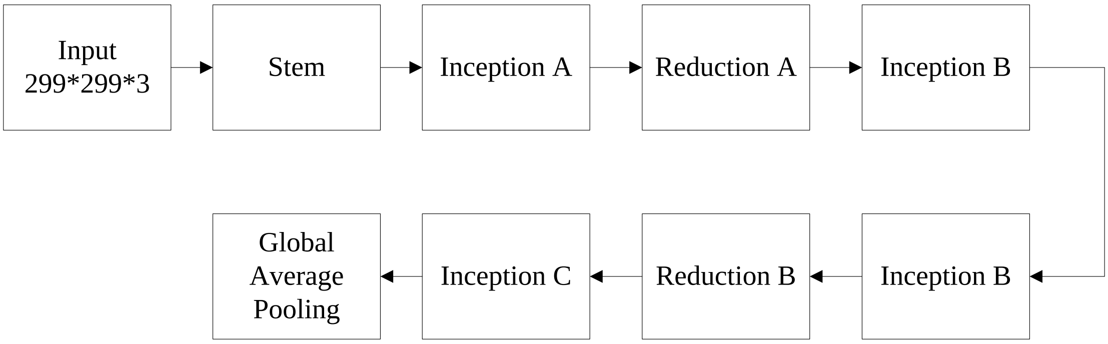


Figure 4-2: Block Diagram of Inception V3

The stem typically consists of a series of convolutional and pooling layers designed to extract basic features from the input image while reducing its spatial dimensions. The stem acts as the entry point for the network, where the raw input image undergoes initial processing to extract fundamental features before being passed through the deeper layers of the network for further analysis and feature.

The Inception module is designed to capture features at different scales by incorporating parallel convolutional pathways with varying kernel sizes. Within the Inception module, multiple convolutional layers with different filter sizes (e.g., 1x1, 3x3, and 5x5) are applied in parallel to the input. By using these different filter sizes simultaneously, the Inception module can capture both local and global features effectively, enhancing the network's ability to understand complex patterns in the input data.

The Reduction module is responsible for reducing the spatial dimensions of the feature maps while increasing their depth. This reduction in spatial dimensions helps manage computational complexity and improve the efficiency of the network. The Reduction module typically employs a combination of convolutional layers with larger filter sizes (e.g., 3x3) and pooling operations to down sample the feature maps. Additionally, dimensionality reduction techniques such as 1x1 convolutions may be incorporated within the Reduction module to reduce the number of parameters and enhance computational efficiency.

4.2.3 Channel Block Attention Module

Convolutional Block Attention Module (CBAM) is a method that focuses on pertinent spatial and channel-wise features to improve the representational capability of convolutional neural networks (CNNs). The Channel Attention Module (CAM) and the Spatial Attention Module (SAM) are its two sub-modules.Channel Attention Module (CAM).

4.2.3.1 Spatial Attention Module (SAM)

The SAM focuses on spatial dependencies by learning to attend to important spatial regions within each feature map. It computes spatial attention weights by first extracting channel-wise statistics (e.g., mean, and standard deviation) across each spatial location. The channel-wise statistics are then used to generate spatial attention maps through a set of convolutional layers and activation functions. These spatial attention maps are multiplied elementwise with the original feature maps to highlight informative spatial regions while suppressing irrelevant ones.

By integrating both the CAM and SAM, CBAM enables Inception V3 to adaptively attend to both channel-wise and spatial-wise features, facilitating better feature representation and discrimination. This attention mechanism helps improve the network's ability to capture fine-grained details and contextually relevant information, leading to enhanced performance in various computer vision tasks such as image classification, object detection, and semantic segmentation. In our case the output feature will be of size 8*8*2048.

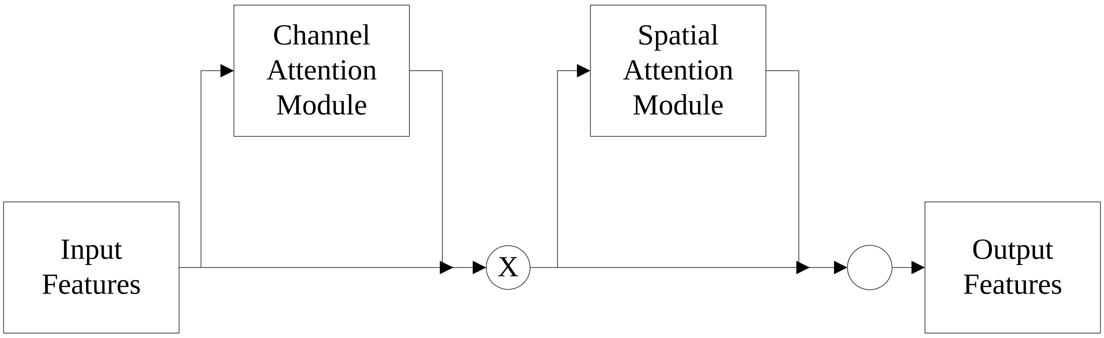


Figure 4-3: Working Mechanism of CBAM

Channel attention weights Mc(F) and Spatial attention weights Ms(F) are represented as follows:

$$Mc(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F)))$$

$$= \sigma((W1(W0(F_{avg}^{c})) + W1(W0(F_{max}^{c}))) \quad (4.1)$$

$$Ms(F) = \sigma(f^{7 \times 7} ([AvgPool(F); MaxPool(F)]))$$

$$= \sigma(f^{7 \times 7} ([F_{avg}^{s}; F_{max}^{s}])) \quad (4.2)$$

Where, F represents the output feature map for each layer of the model,

MLP is the fully connected layer,

AvG&Pool is the global average pooling layer,

MaxPool is the global maximum pooling layer,

σ is the sigmoid activation function,

f (7 × 7) represents 7 × 7 convolution layer.

4.2.4 Feature Extraction

The cropped images of the Region of Interests (ROIs) from the above module are given as input to this module. Then the Deep Convolutional Neural Network Resnet 50 extracts feature from each cropped image. For this purpose, we need two Resnet 50 networks. One for the extraction of feature from the carpal image and another one for the extraction of feature from the metacarpal image. Then, the output from the last convolutional layer is taken from each network. Then, the outputs are flattened and then concatenated. Since we will be providing the gender input to the program, this will be concatenated to the above concatenated vector. This flattened vector is then given as input to the fully connected layer. The value of the gender input is either 1 or 0:1 for male and 0 for female.

4.2.5 ResNet50

Using a short circuit method, ResNet50 is a representative network of the ResNet residual network series. Identity mapping is incorporated into the architecture of the residual learning unit in order to establish a direct correlation channel between the input and output. This improves the network's capacity to collect intricate characteristics by allowing the reference layer to concentrate on acquiring the remaining data. One of the most important ResNet design principles is that the number of feature maps doubles when the feature map size is halved, guaranteeing that the network's complexity is maintained at every layer.

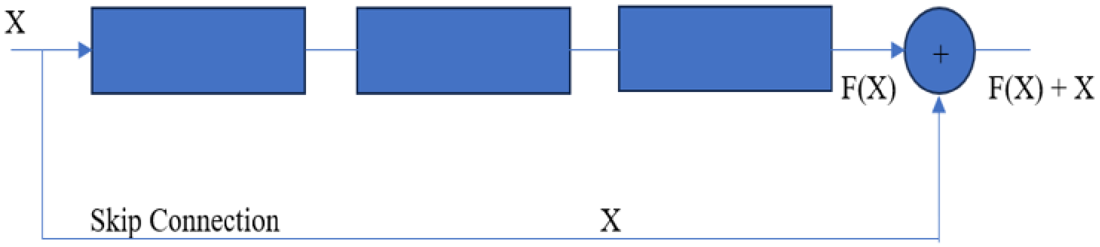


Figure 4-4: Residual Unit Structure

4.2.5.1 Identity Block

When the dimensions of the input activation and the output activation are the same, the identity block is utilized. It has several convolutional layers, batch normalization, and ReLU activation functions in order of sequence. The output of the last layer is added to the input (hence the name "identity" block), which is then processed by a second ReLU activation. The identity block's job is to discover residual mapping that approaches an identity function in order to efficiently maintain network information flow. By giving the gradients during backpropagation a shortcut path (identity connection), this block aids in avoiding the vanishing gradient problem.

4.2.5.2 Convolutional Block

When the dimensions of the input and output activations differ, the convolutional block is employed. Firstly, a convolutional layer with a greater stride is used to minimize the input activation's spatial dimensions. To recover the original dimensions, it then does batch normalization and ReLU activation. This is followed by a further set of convolutional layers with smaller filters. Lastly, batch normalization and ReLU activation are applied once more. A projection shortcut, which matches the dimensions of the output activation by performing a linear projection of the input activation, may also be included in the block. This shortcut adds the input activation to the processed output. The network can learn more intricate mappings by means of the convolutional block, which assists in adjusting the dimensions of the input activation to correspond with those of the output activation.

4.2.5.3 ResNet50 Architecture

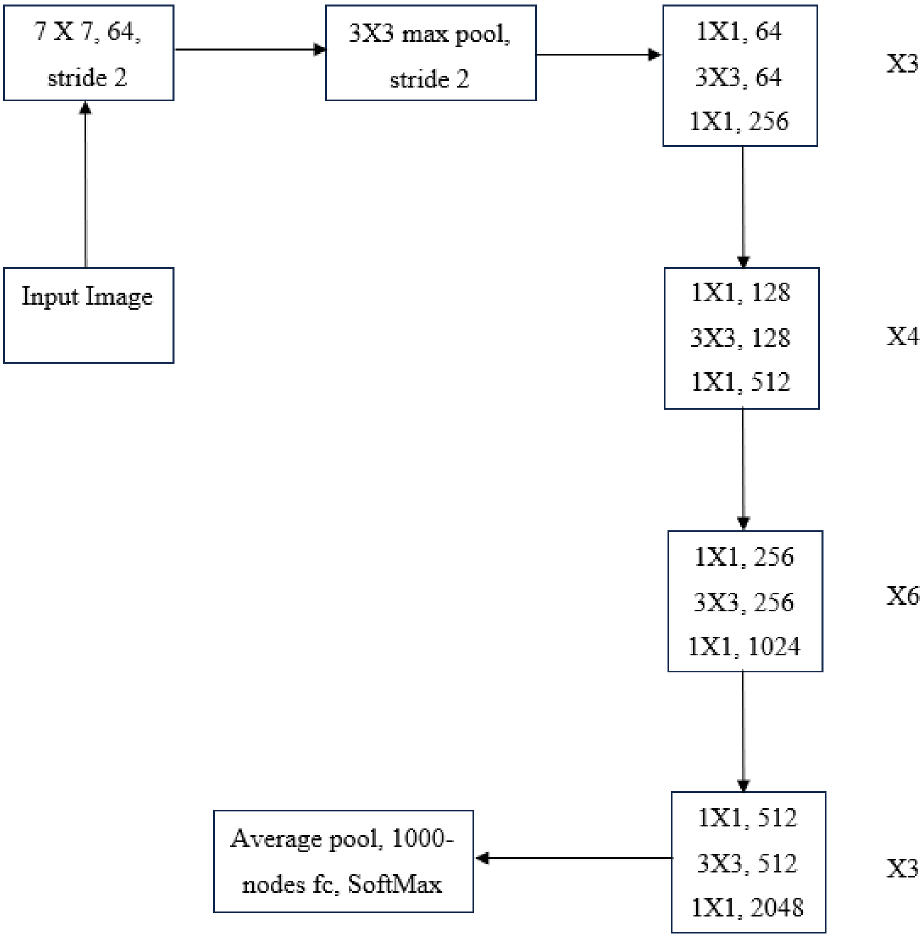


Figure 4-5: Block Diagram of ResNet50

The following components are part of the 50-layer ResNet architecture, as illustrated below:

• A 64-kernel convolution with a 2-sized stride that includes a 7x7 kernel.

• A maximum pooling layer with a stride size of two.

• Nine additional layers: one with 1×1,64 kernels, another with 3×3,64 kernels, and a third with 1×1,256 kernels. There are three repetitions of these levels.

• An additional 12 layers that contain 4 iterations of 1×1,128, 3×3,128, and 1×1,512 kernels.

• 18 additional layers with 2 cores (3×3,256 and 1×1,1024) and 1×1,256 cores, iterated six times.

• Nine additional layers, each having three iterations of 1×1,2048, 3×3,512, and 1×1,512 cores.

• The softmax activation function is used to create a fully linked layer with 1000 nodes after average pooling.

4.3 Regression Head:

- Add a custom regression head on top of the fine-tuned ResNet50.
- This head consists of fully connected layers that take the extracted features from ResNet50 and predict the bone age as a continuous value.

4.4 Optimizer

Adam is an optimization algorithm that can be used to update network weights iteratively based on training in place of the traditional stochastic gradient descent process.

The adaptive moment estimation optimizer, or ADAM, is what we had employed. It determines a different learning rate for every parameter. It is ideal for our model since it is highly memory-efficient and computationally productive.

4.5 Validation Metrics

Since the suggested model views the evaluation of bone age as a regression task, mean square error (MSE) can be used as the task's loss function.

$$\mathrm{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|^2 \tag{4.3}$$

where n represents the number of the training sets, y is the ground-truth age and $\hat{y}$ is the predicted value of bone age.

The evaluation metric for network performance in this study's treatment of bone age assessment as a regression task is the mean absolute error (MAE) between the model's output and the ground-truth age, which may be expressed as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{4.4}$$

where N represents the number of input samples and y is the true value of bone age, ŷ and is the predicted bone age.

4.6 Use Case Diagram

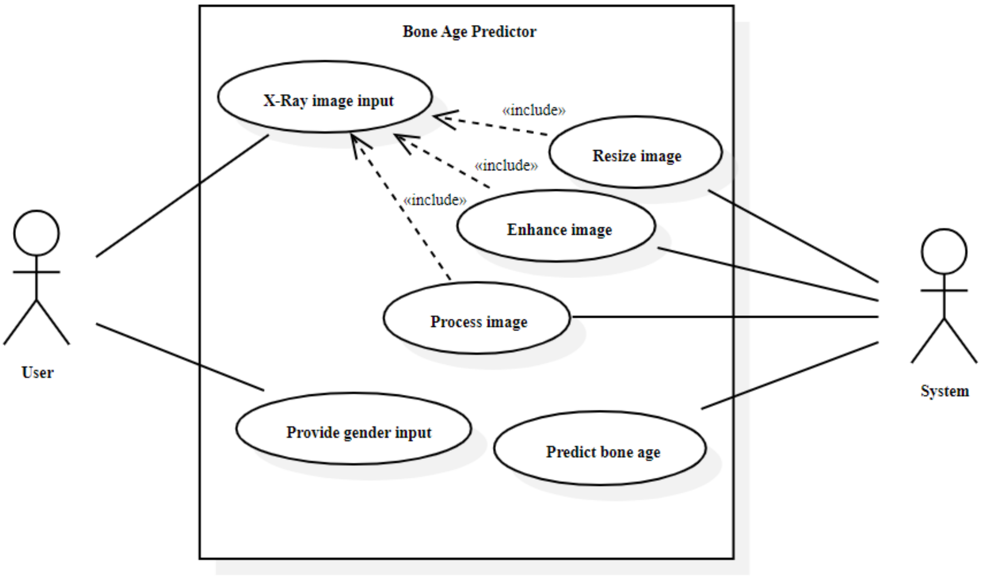


Figure 4-6: Use Case Diagram

This is the overall use case of our project. Here the external actors are user and the system. The user interacts with the system by providing the X-Ray image of the left hand and corresponsing gender. Then, the system pre-processes the images. The system performs the task such as resizing, enhancing, etc on the image. Then, the user is provides with the bone age of the provide image X-Ray.

4.7 Algorithm

Step 1: Input left hand X-ray image with dimensions 299 x 299 x 3.

Step 2: Pass the input image through inception V3 model.

Step 3: Extract output feature map from the last convolutional layer.

Step 4: Apply CBAM to the output feature map.

Step 5: Generate feature map of size 8 x 8 x 2048.

Step 6: Produce a heat-map from the CBAM output to highlight areas of high attention.

Step 7: Crop the original image based on the heat-map to localize the carpal region.

Step 8: Replace the cropped region with a black mask.

Step 9: Pass the masked image through Inception V3 and CBAM again.

Step 10: Produce a heat-map for the masked image to localize the metacarpal region.

Step 11: Crop the original image based on the new heat-map to extract the metacarpal region.

Step 12: Feed the cropped images of the carpal and metacarpal regions into separate ResNet50 networks.

Step 13: Extract features from the last convolutional layer of each ResNet50 network.

Step 14: Flatten and concatenate the extracted features.

Step 15: Concatenate the gender input (1 for male, 0 for female) to the concatenated vector.

Step 16: Implement regression head on the top of concatenated vector.

Step 17: Use fully connected layers to predict the bone age as a continuous value.

Step 18: Use the Adam optimizer for updating network weights.

Step 19: Train the model using mean square error (MSE) as the loss function.

Step 20: Evaluate model performance using mean absolute error (MAE) between predicted and ground-truth bone ages.

5 IMPLEMENTATION DETAILS

Our goal in this project is to create and apply a state-of-the-art model that can both identify possible growth problems and predict bone age from X-ray pictures. By utilizing the capabilities of deep learning, specifically Convolutional Neural Networks (CNNs), our goal is to transform the crucial duty of determining a child's age through bone age assessment. Using cutting-edge methods like the InceptionV3 architecture enhanced with the Convolutional Block Attention Module (CBAM), our model attempts to concurrently identify anomalies or growth problems from X-ray images and accurately estimate bone age.

5.1 Preprocessing

5.1.1 Resizing

The size of the image required by our Inception V3 model is (299,299,3). To resize the image required by our model, we had applied the Bi-cubic interpolation. It calculates the new value of pixel by considering the 4 x 4 neighborhood pixel surrounding the data. As this technique used the 16 pixels to calculate new pixel value, this show the high accuracy over the bi-linear and nearest neighbor interpolation. To ensure the smoothness of the images, it used the cubic function to estimate the weight of each neighborhood pixel contributions. New pixel get influenced higher by the nearer pixels. Also, for the edge pixels, they lack the neighborhood pixel. So, this extends the image boundary by padding.

$$h(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \le |x| < 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \le |x| < 2 \\ 0 & 2 \le |x| \end{cases} \tag{5.1}$$

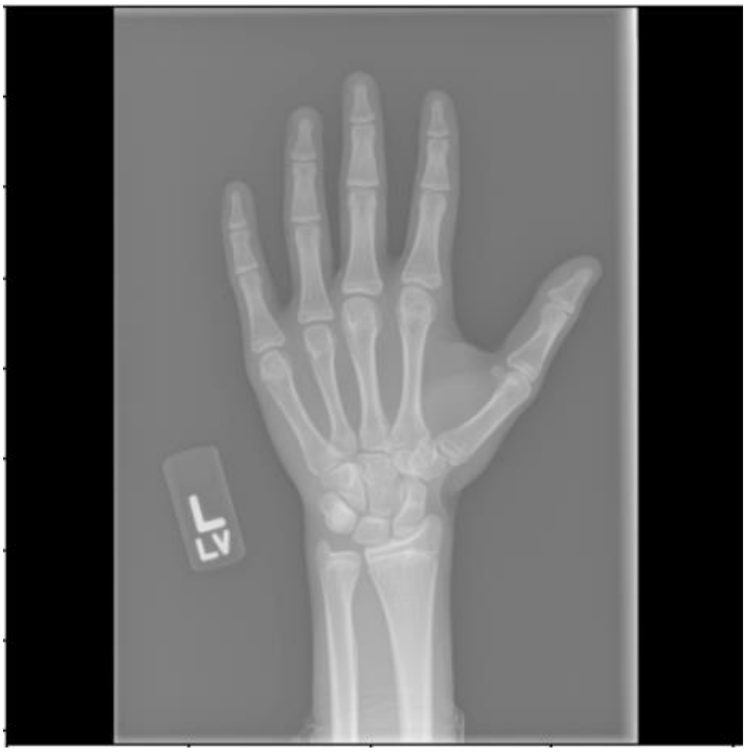


Figure 5-1: Original Image

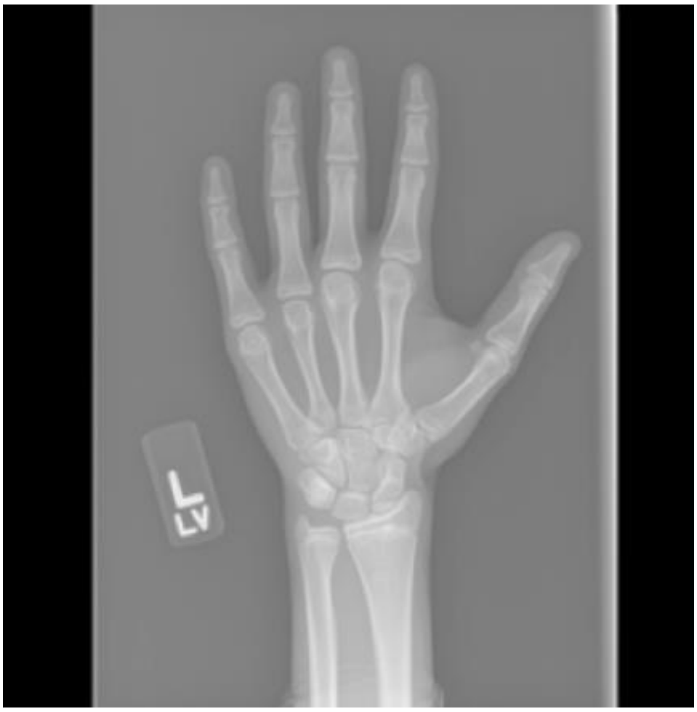


Figure 5-2: Resized Image

5.1.2 Enhancing

5.1.2.1 Histogram Equalization (HE):

For a grayscale image, each pixel has a value representing its brightness, ranging from 0 (black) to 255 (white). Histogram equalization aims to spread out these pixel values evenly across the entire range, making the image look clearer. Mathematically, it involves computing the cumulative distribution function (CDF) of the histogram and then transforming pixel values according to this CDF.

5.1.2.2 Adaptive Partitioning:

Instead of applying histogram equalization to the entire image at once, CLAHE divides the image into smaller blocks or tiles. For each tile, it computes a local histogram and performs histogram equalization. This step helps to enhance local contrast while preserving details.

5.1.2.3 Contrast Limiting:

To prevent over-amplification of noise, CLAHE applies a contrast limiting mechanism. After histogram equalization for each tile, it checks if any pixel values exceed a certain limit. If they do, it scales down the pixel values to ensure they stay within the limit. This step helps to maintain the overall image quality.

Table 5-1: CLAHE Parameters

Parameters	Value
Clip Limit	2.0
Title Grid Size	(64, 64)

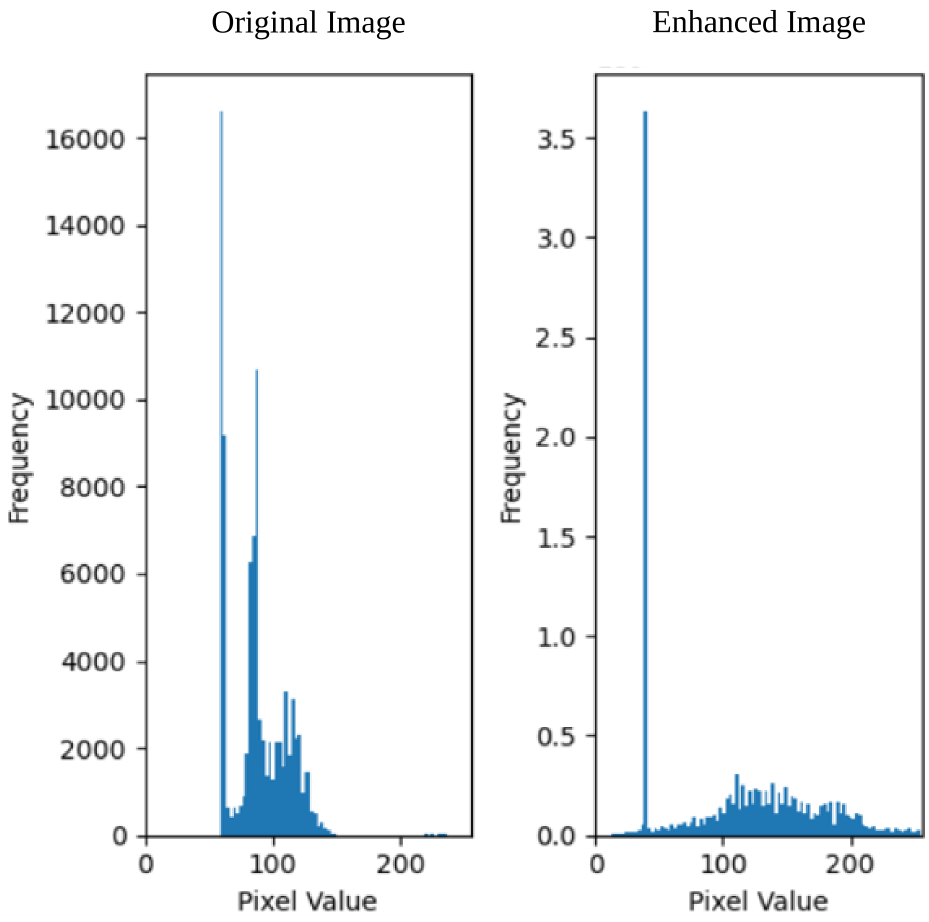


Figure 5-3: Histogram Graph of original and enhanced image

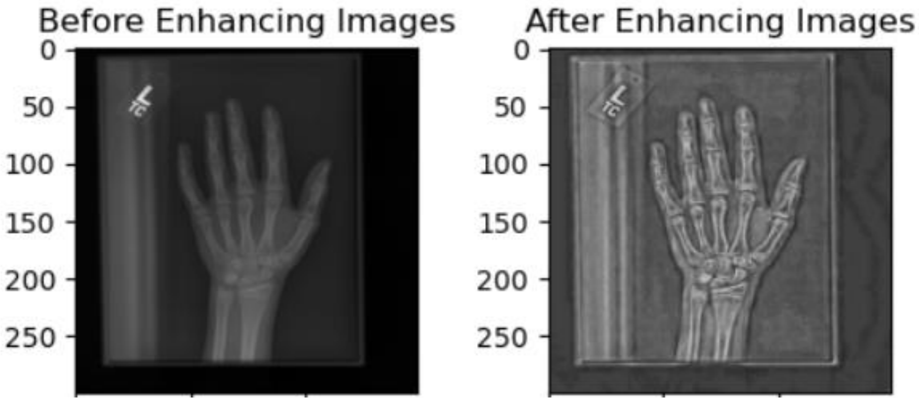


Figure 5-4: Original and Enhanced Images

5.1.3 Augmentation

We had applied the following augmentation techniques to diversify our datasets:

- Rotation

We had used rotation_range=40 degree which means it can rotate the image within any angle in the range [-40, 40] degree randomly.

- Width and Height shift range

These parameters determine the horizontal and vertical shift of the image as a fraction of the total width and height of the image respectively.

- Zoom Range

This parameters control percentage of random zoom in or zoom out of the original image.

- Horizontal flip

This flips the images i.e. mirror images.

- Fill Mode

Due to shifting and rotation, there can be newly created pixels. The fill mode determines how this newly created pixels will fill.

Table 5-2: Augmentation Parameters

Parameters	Rotation range	width_shift_ range	height_shift range	zoom_ range	horizontal_flip	fill_ mode
Values	40	0.2	0.2	0.2	True	nearest

5.2 Training Inception V3 with CBAM Network

Then, instead of utilizing a prefabricated and pre-trained network, we deployed our own Inception V3 network. The top layers of the Inception V3 were removed, and the output of the last convolutional layer was then fed into the Convolutional Block Attention Module's (CBAM) Channel Attention Module. Subsequently, global average pooling was applied to the CBAM output, which had the dimensions 8*8*2048. After that, it was fed into the dense layer, which had a single output, the projected age and a sigmoid activation function. During roughly 40 epochs, the network was trained using 12611 pictures from the RSNA dataset.

5.2.1 Sigmoid Activation Function

The sigmoid activation function, also known as the logistic function, is a type of activation function commonly used in artificial neural networks and machine learning models. It is especially prevalent in binary classification problems.

The sigmoid function is defined mathematically as:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{5.1}$$

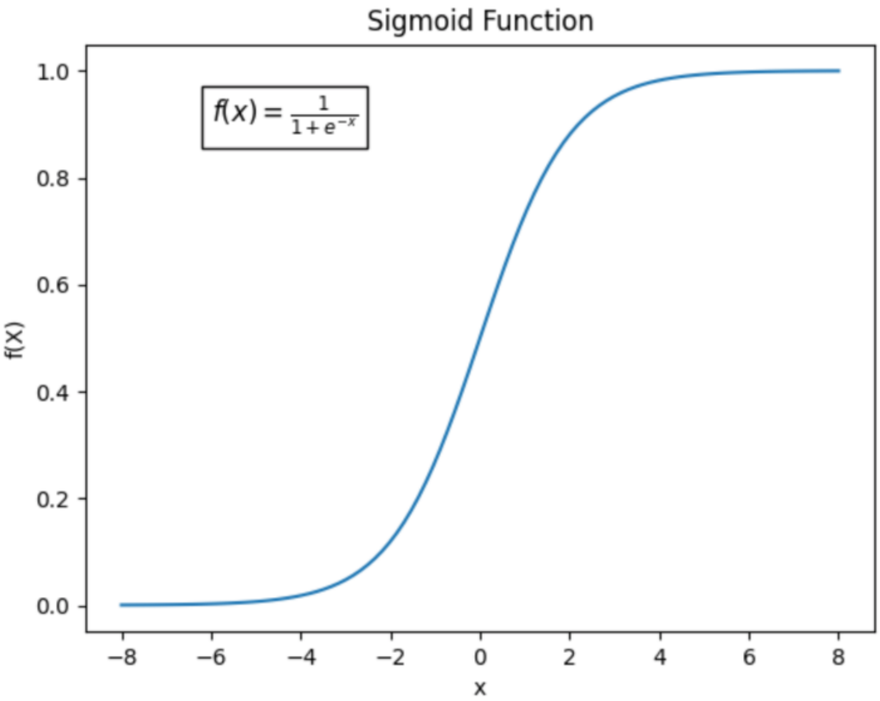


Figure 5-5: Sigmoid Activation Function

5.3 Generating Heat Map

5.3.1 Preprocessing:

First, we take the input image and preprocess it to prepare it for analysis. This may involve resizing the image and applying normalization or other transformations to make it suitable for input into the model.

5.3.2 Prediction:

Then, we pass the preprocessed image through a CBAM (Channel-wise Attention Module) model. This model is designed to identify and highlight important regions or features within the image.

5.3.3 Heat-map Generation:

The CBAM model generates an output that represents the attention or importance of different parts of the image. This output contains information about which regions the model considers most relevant for the task at hand. Aggregate the attention information across channels to create a single heat-map. This involves combining the attention weights from all channels to produce a unified representation of the overall saliency of each pixel in the image.

5.3.4 Normalization:

Normalize the values in the heat-map to ensure they fall within a standardized range, typically [0, 1]. This step ensures consistency and facilitates interpretation of the heat-map by scaling the values appropriately.

5.3.5 Up sampling:

Resize or up sample the heat-map to match the dimensions of the original input image. This ensures that the heat-map aligns properly with the original image for visualization purposes, allowing you to overlay the heat-map on top of the image to visualize the areas of high attention or importance.

5.4 Masking and cropping image

For generating and applying mask, various steps are involved. First, we get the coordinate of the bounding box that encloses the more focused area in the heatmap. So, to get the bounding box coordinate we followed these steps:

5.4.1 Thresholding

First, we begin with a heat-map generated from an image using the model. Then, apply a threshold to the heat-map, creating a binary mask where values above the threshold indicate regions of interest and values below the threshold are ignored.

5.4.2 Identifying Regions of Interest

Then, we find the indices of non-zero elements in the binary mask. These indices correspond to the locations where the heat-map intensity exceeds the threshold, indicating regions of interest.

5.4.3 Bounding Box Calculation

If there are no non-zero indices found, it suggests that there are no significant regions of interest in the heat-map, and therefore no bounding box can be defined. In such cases, there is no attention in the image. Otherwise, we compute the minimum and maximum row and column indices of the non-zero elements. These indices represent the corners of the bounding box enclosing the regions of interest in the heat-map.

5.4.4 Bounding Box Representation

Then, we store the computed minimum and maximum row and column indices as coordinates in a bounding box data structure. This structure typically includes attributes such as 'min_row', 'min_col', 'max_row', and 'max_col' to represent the bounding box coordinates.

After getting the bounding box coordinate of the ROIs we cropped the original images using the python library OpenCV. First, we got the image of the carpal region of the X-Ray. After getting the cropped image of the carpal we applied black mask to the cropped part in the original image and again we generated heat-map of the masked image by passing it to the Inception V3 + CBAM model. Now, the attention was shifted to metacarpal part of the X-ray images and heat-map was generated accordingly. Now, by using the new heat-map we got the bounding box coordinate of the metacarpal portion in the image, and we cropped that portion from the original X-Ray image.

5.5 Training ResNet50 and predicting Bone age

The preserved image i.e. Carpal and Metacarpal region was passed into the respective ResNet50 architecture. The additional input of Gender was also added to the architecture to maintain the unbiasedness of the datasets. The various operations performed under this architecture are explained below:

5.5.1 Convolution Operation

Given an input feature map X of size H*W*C_{in} (height, width, input channels) and convolutional filter F of size f*f*C_{in} *C_{out} (Filter height, filter width, input channels, output channels), the output feature map Y is computed as:

$$Y_{i,j,k} = \sum_{l=0}^{C_{\text{in}}-1} \sum_{m=0}^{F-1} \sum_{n=0}^{F-1} X_{i+m,j+n,l} \times W_{m,n,l,k} + b_k \tag{5.2}$$

Where b_k is the bias term for the kth filter.

5.5.2 Batch Normalization

Given an input X to a batch normalization layer, the output Y is computed as:

$$Y = \frac{X-\mu}{\sqrt{\sigma^2+\epsilon}} \times \gamma + \beta \tag{5.3}$$

where, μ is the mean of the batch,

σ is the standard deviation of the batch,

ϵ is a small constant for numerical stability,

γ is the scale parameter, and

β is the shift parameter.

5.5.3 ReLU Activation

The Rectified Linear Unit (ReLU) activation function was used after each convolutional and batch normalization layer which is defined as below:

$$f(x) = \max(0, x) \qquad (5.4)$$

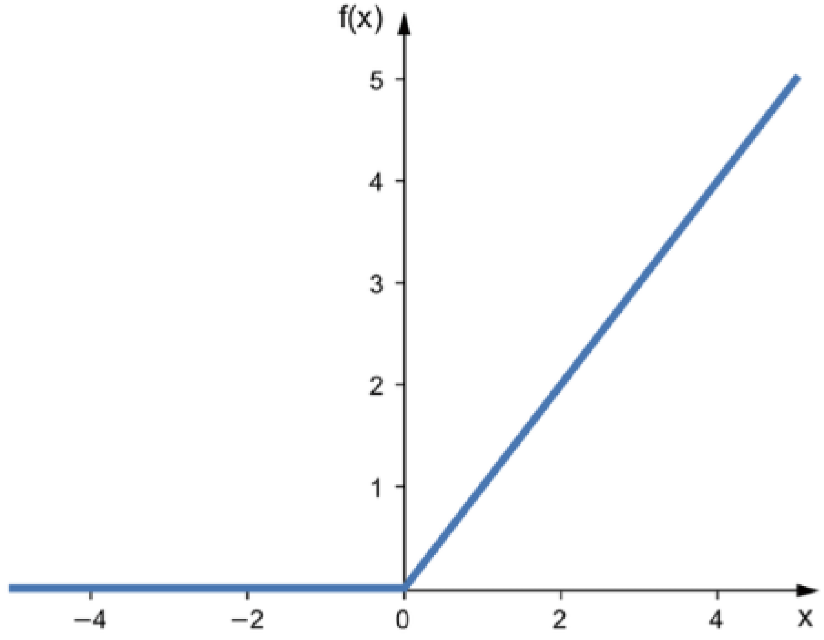


Figure 5-6: ReLU activation function graphical visualization

5.5.4 Residual Connection

The input of a residual block (or residual unit) was added to its output, which allowed for the training of very deep networks without the vanishing gradient problem.

If f(x) represents the output of the residual block and X represents its input, the output Y can be computed as:

$$Y = f(x) + X \qquad (5.5)$$

5.5.5 Pooling Operation

Max pooling and average pooling were often used. So that, we retained the most important information by reducing the spatial dimensions of the feature maps.

Given an input feature map of size H*W*C (height, width, channels), and pooling window size S*S, the output feature map Y is computed as follows:

For Max pooling:

$$Y_{i,j,k} = \max_{m=0}^{S-1} \max_{n=0}^{S-1} X_{Si+m,Sj+n,k} \tag{5.6}$$

For Average Pooling:

$$Y_{i,j,k} = \frac{1}{S^2} \sum_{m=0}^{S-1} \sum_{n=0}^{S-1} X_{Si+m,Sj+n,k} \tag{5.7}$$

Where, $Y_{i,j,k}$ is the value of the kth channel of the output feature map at position (i, j).

$X_{si+m,\ sj+n,\ k}$ represents the values within the pooling window centered at position (S_i, S_j) in the kth channel of the input feature map.

6 RESULT AND ANALYSIS

Rather than utilizing the pre-trained weights of the ImageNet dataset, we trained the Inception V3 + CBAM model manually. We employed binary cross entropy as the loss function and a sigmoid activation function at the output to train the model. And we were able to produce an accurate heat-map for the X-Ray images after training the model over more than 60 epochs. The images of the heat-map are shown below:

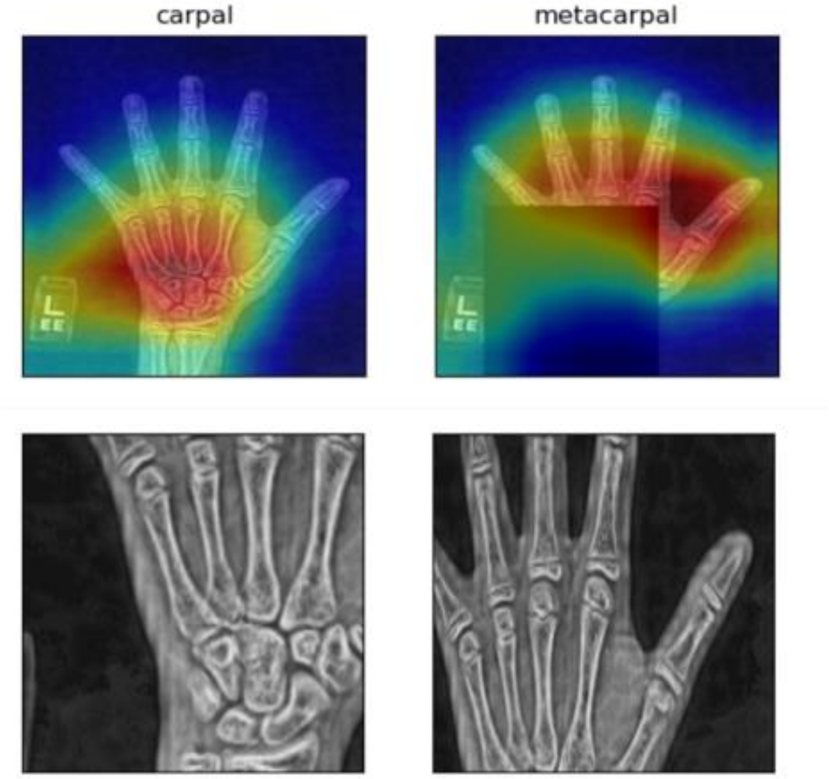


Figure 6-1: Sample 1

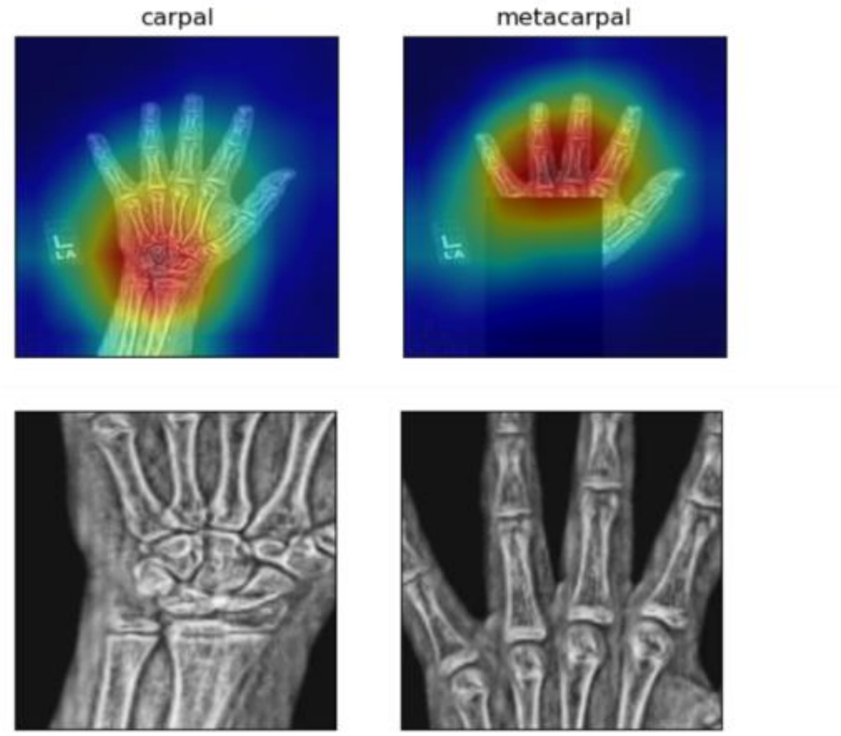


Figure 6-2: Sample 2

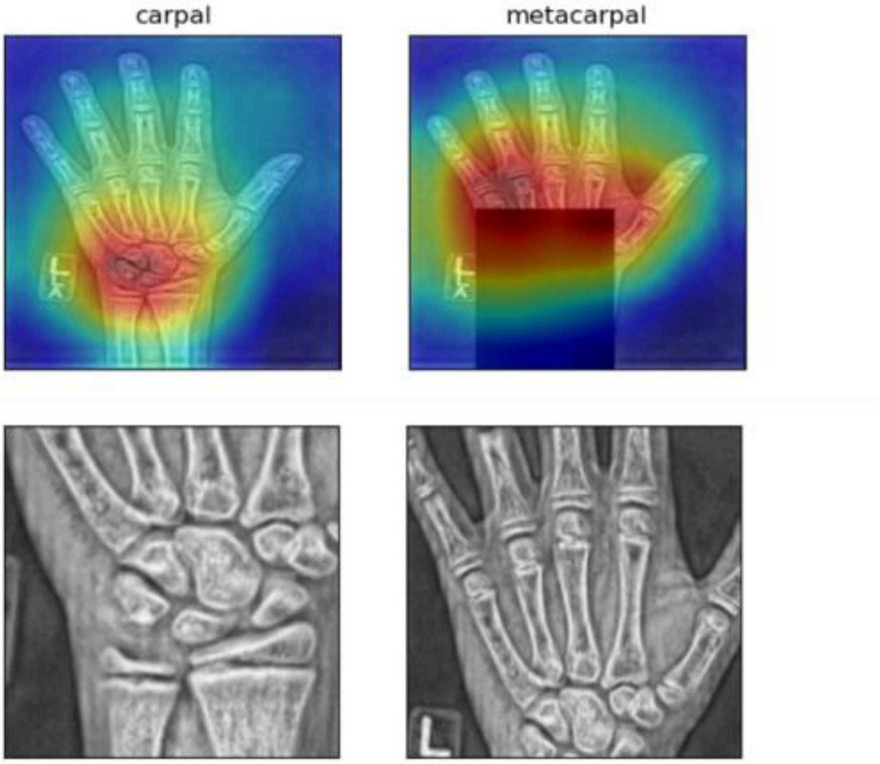


Figure 6-3: Sample 3

Here, we can see that the heat-map first tends to focus on the carpal region. So, by leveraging this fact we then get the bounding box coordinate of the carpal region. Using these coordinates, we crop the image and save it. Then, we apply a black mask as shown in the figure. Then, again the masked picture is passed through the model and now the focus is shifted to the metacarpal region of the image. Now, we get the bounding box coordinate for the metacarpal region based on the heat-map. After that we crop the image and save the image of the metacarpal.

Metacarpal

We have trained the ResNet 50 model ourselves rather than using the pre-trained weights of ImageNet dataset. For training the model we had used the metacarpal images that we cropped with the help of bounding box coordinate obtained from the heatmap. The activation function used was ReLU activation function at the output and the loss function used was mean squared error. And, we have used ADAM as the optimizer with learning rate of 0.0001, Beta1 value as 0.9 and Beta2 value as 0.999.And after training the model more than 30 epochs the graph for training loss was obtained as

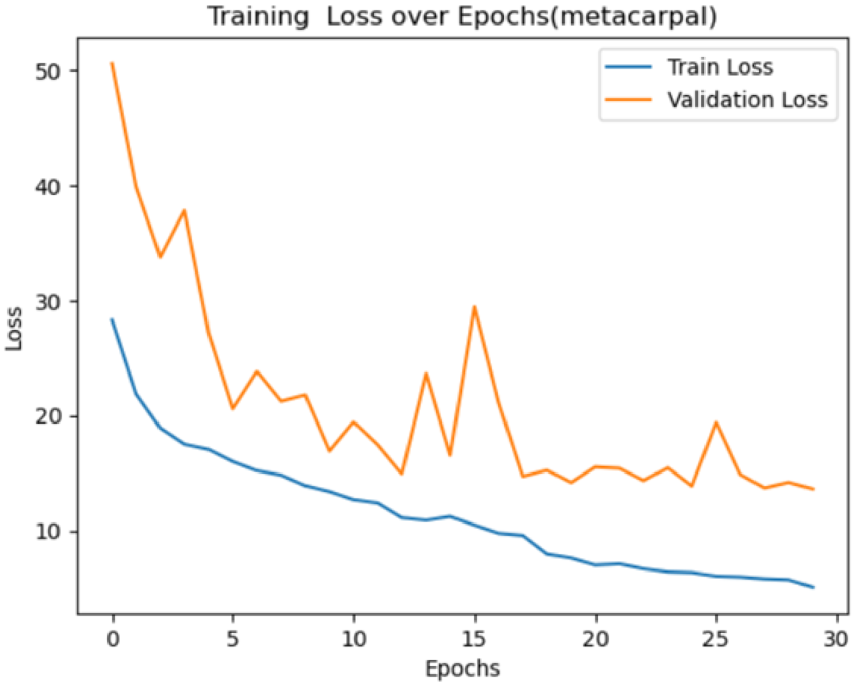


Figure 6-4: Training and validation loss graph for Metacarpal

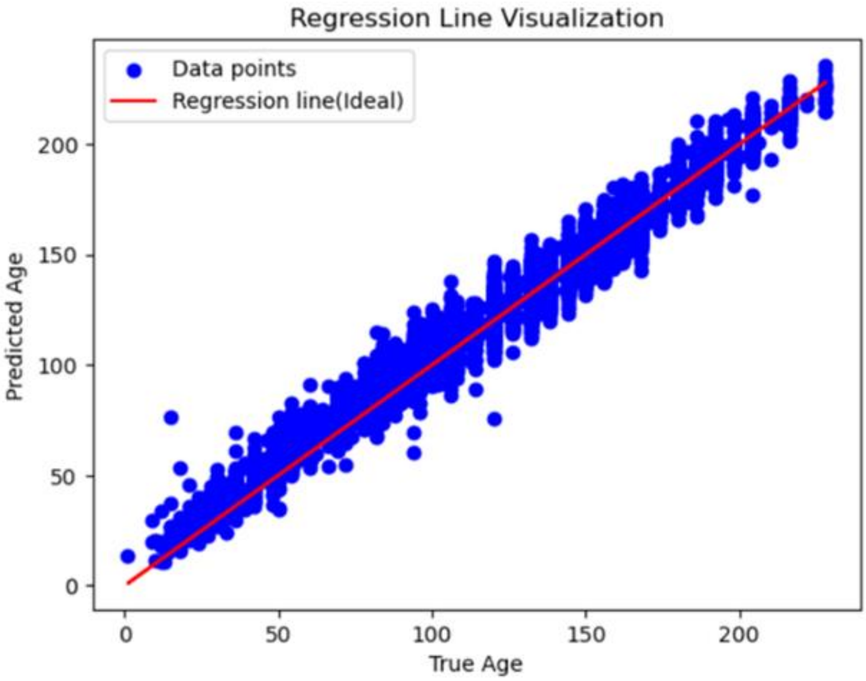


Figure 6-5: Regression Graph of Metacarpal Training

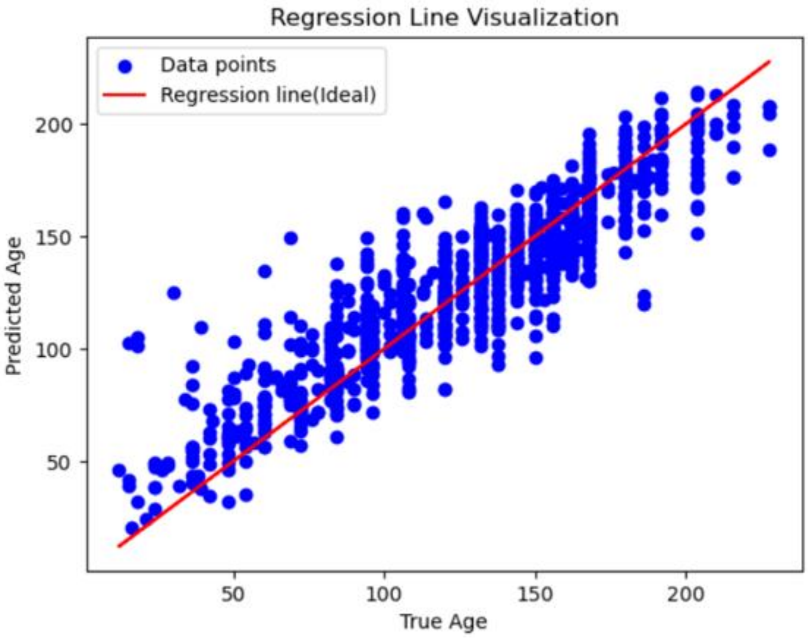


Figure 6-6: Regression Graph for Metacarpel Validation

Again, we have followed the same task for training the ResNet 50 model for carpal images. For training the model we had used the carpal images that we cropped with the help of bounding box coordinate obtained from the heatmap. The activation function used was ReLU activation function at the output and the loss function used was mean quared error. And, we have used ADAM as the optimizer with learning rate of 0.0001, Beta1 value as 0.9 and Beta2 value as 0.999.And after training the model more than 30 epochs the graph for training loss was obtained.

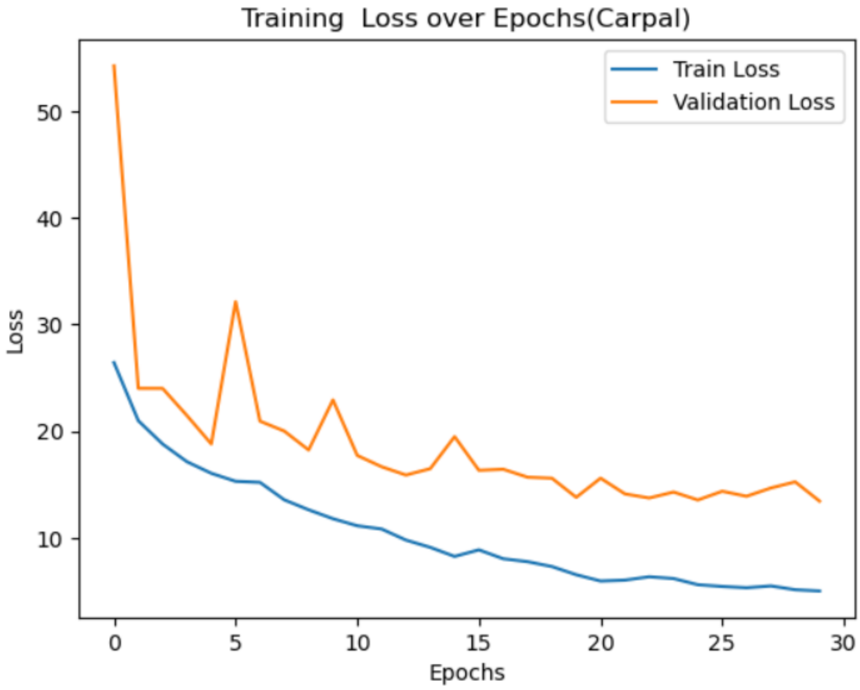


Figure 6-7: Training and validation loss graph for Carpal

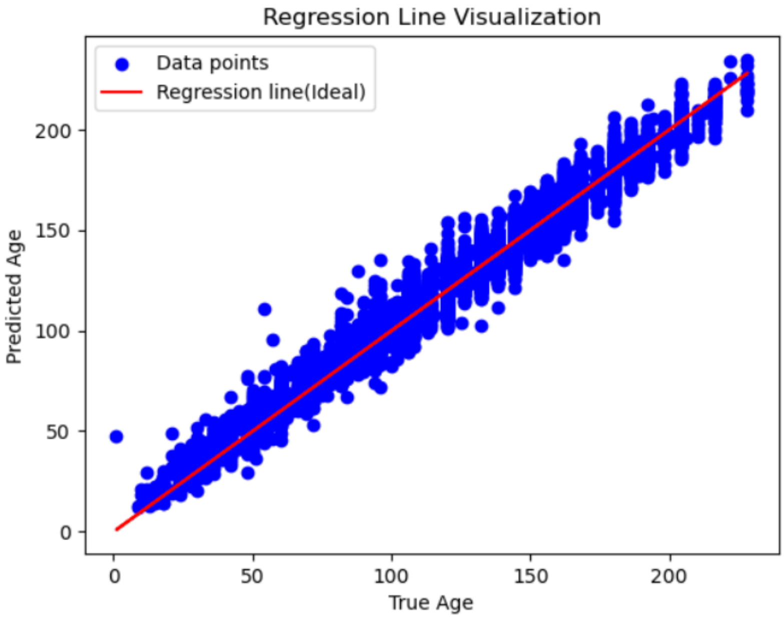


Figure 6-8: Regression Graph for Training Carpal

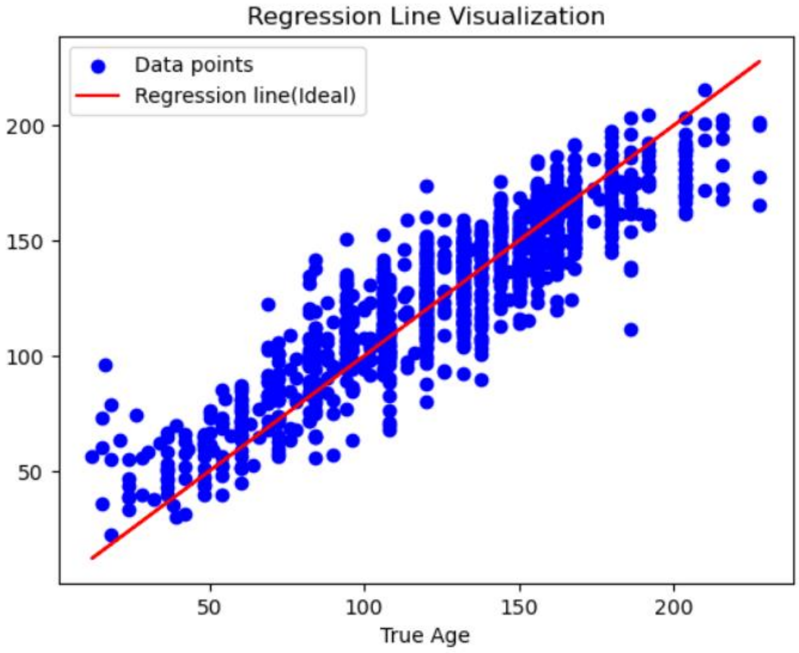


Figure 6-9: Regression Graph for Validation of Carpal

We had trained the model for 30 epochs each for metacarpal and carpal regions separately. During training, the mean absolute error (MAE) for carpal and metacarpal regions was found to be 5.0046 months and 5.070 months respectively. For the validation sets the MAE was found to be 13.429 months and 13.603 months respectively.

While plotting the regression line, it seemed that the predicted age varies sharply from the actual age. This was due to the small frequency of dataset of that age, particularly below 4 years and above 15 years. This difference causes our model to increase the mean absolute error to 13 months.

Combined

After concatenating the model and taking gender into account, the model seemed to be improved largely. During the training loss decreased to 1.296 and the mean absolute error was 7.4922. For the validation set the loss was 3.213 and MAE was found to be 10.34 months.

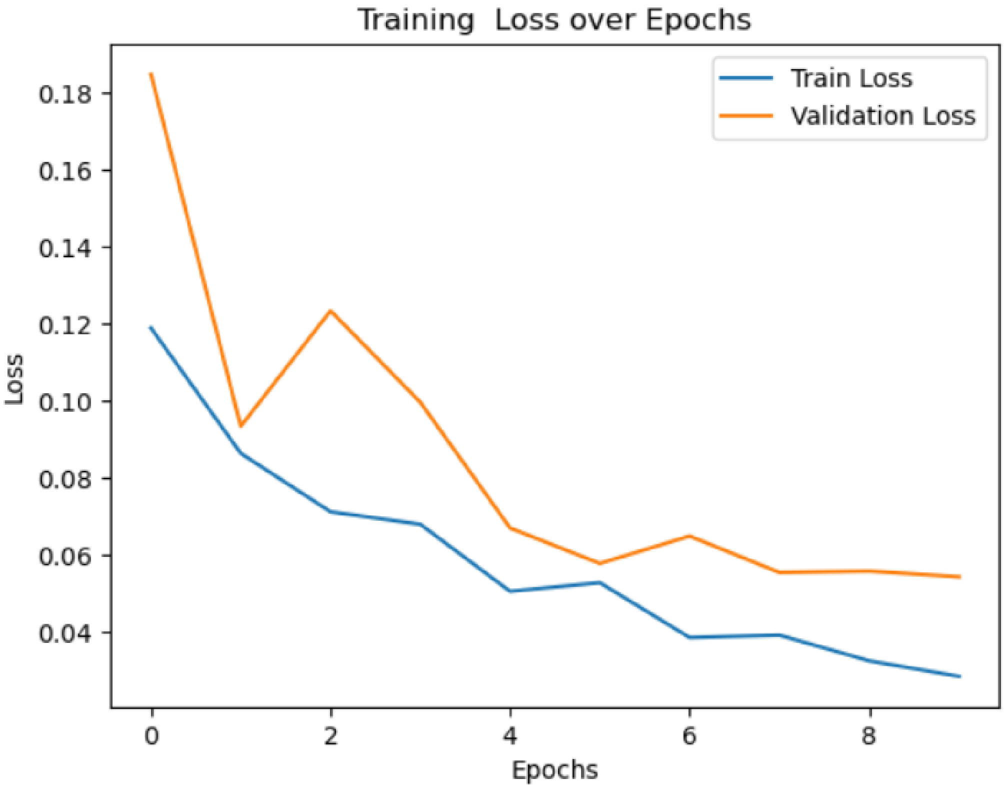


Figure 6-10: Training and Validation Graph for Combined Data

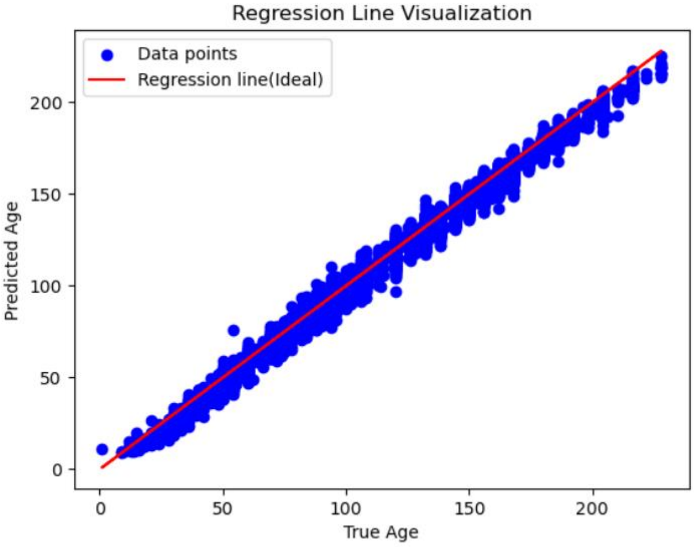


Figure 6-11: Regression Graph of Combined Training

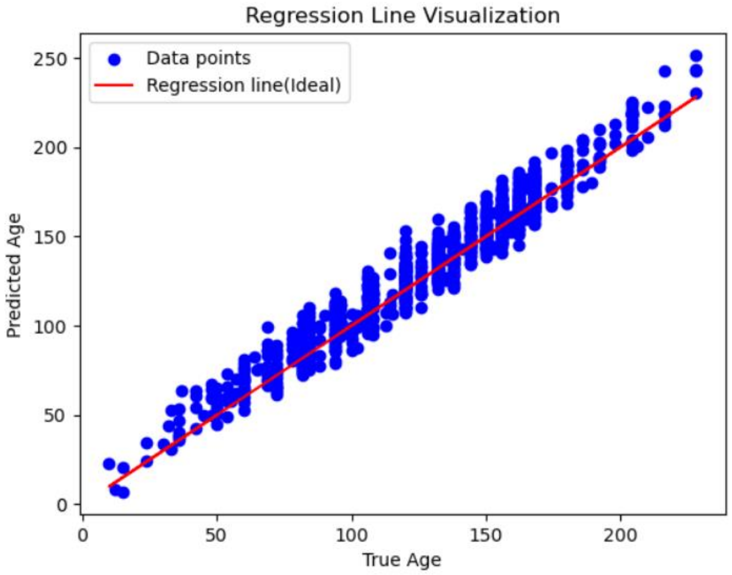


Figure 6-12: Regression Graph for Combined Validation

When we trained the resnet50 architecture for each feature seperately we can see that the error was large and there was more randomness in the scatterplot. But when we combine these features we can see that the overall error of predicting the boneage on the validation set was decreased. We can reason this as more detail about the x-ray image provide more information about the bone age.

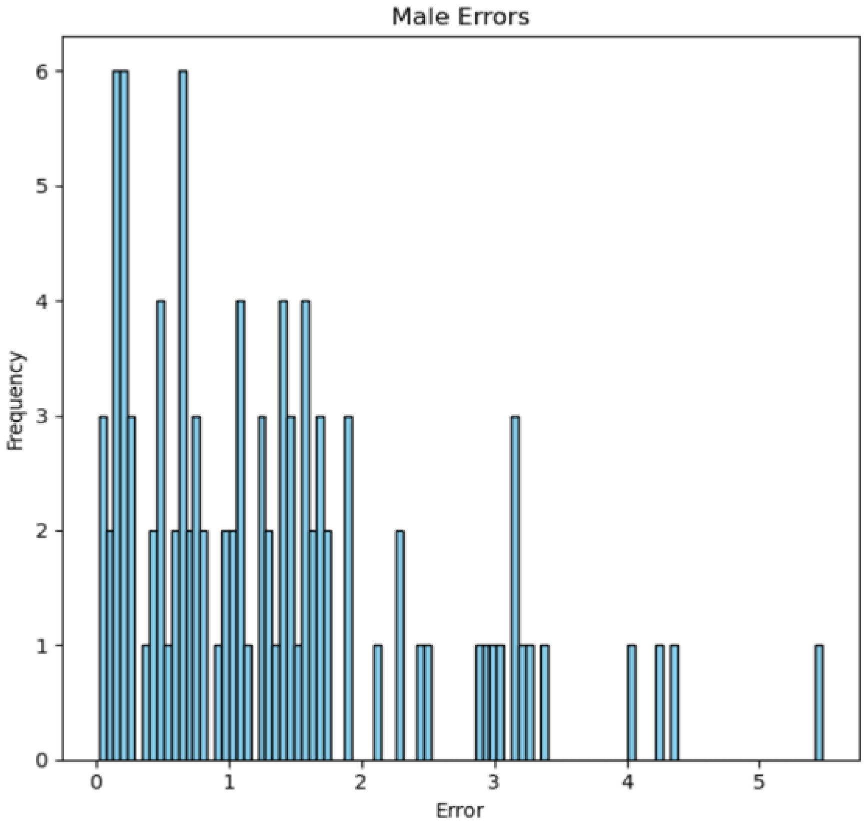


Figure 6-13: Error Frequency(male)

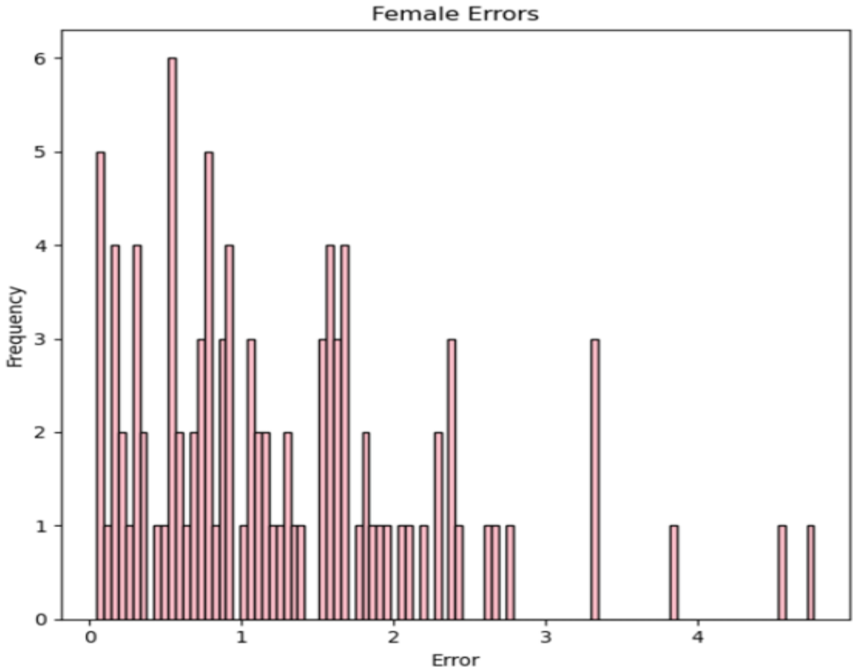


Figure 6-14: Error Frequency(Female)

Majority of errors are under 1.5 years but some large error in the histogram is due to the vary of predicted age below 5 years and above 15 years as mentioned earlier. Hence our model works well to with the test dataset as well.

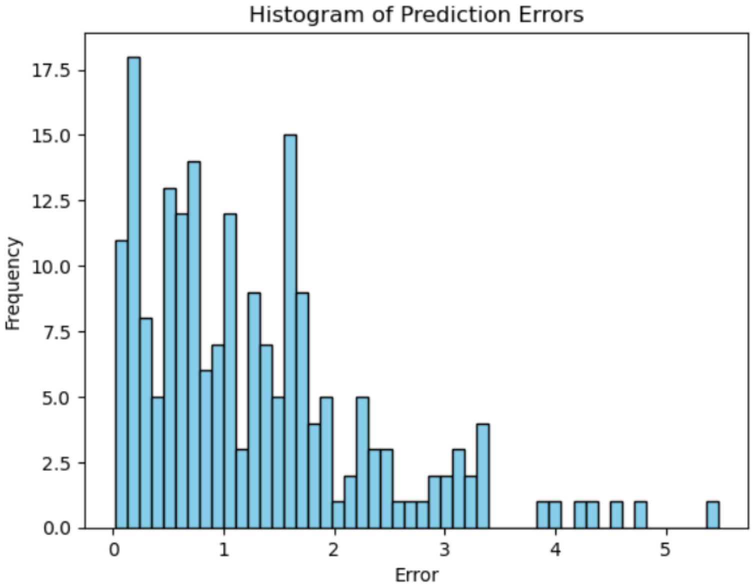


Figure 6-15: Histogram of Prediction Errors

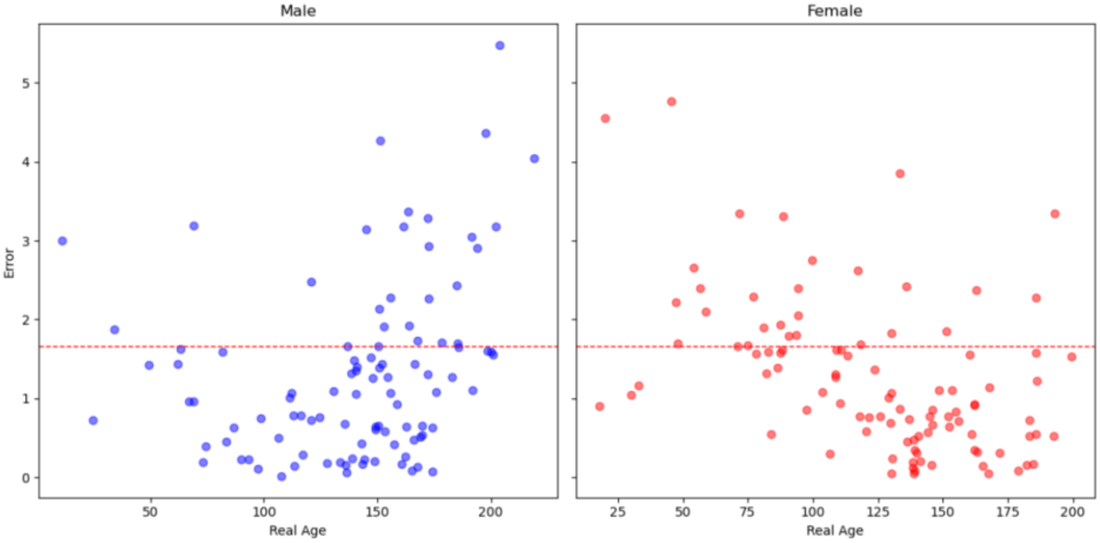


Figure 6-16: Frequency of Error

While plotting a scatter plot between the real age and error while predicting, we had drawn a horizontal line at error equal to 1.67 years to know the which age group had the most error. Majority of ages group are below that horizontal line on both male and female. It means, model is predicting with lower error. Some scattered points are above the that line for the age group above 15 years and below 4 years. Since we had taken gender into the considerations of our model, the predicting error was too same which was verified by our scatter plots.

Comparison with Dr. Albert Model.

After predicting the bone age of 200 test dataset, we calculated the coefficients of determination (R^2) which was found to be 0.901. Our R squared values found to be nearer to 1 which means our prediction almost match the actual values or references. The calculation involved are shown below:

$R^2 = 1 – (SS_{residual}) / (SS_{total})$

Where,

$SS_{residual}$ = sum of squares of the residuals, which are the difference between the actual values and predicted values= 36578.11

SS_{total} = total sum of squares, which is a measure of the total variability of in the actual values= 370099.12

The high R^2 implied that our model generalizes well to new data as well as effectively captures the relationships between the predictors and the bone age in the testing dataset. Overall, R^2 value of 0.901 from testing dataset signifies strong predictive performance and suggests that our model is robust and reliable for estimating bone age.

On Contrary, we had tested the same dataset on *freeboneage.com* provided by Dr. Pradeep Albert as cross check of our model predictions. We had found that the coefficient of determination by our model (0.901) and this model(0.8973) was approximately equal which verified that our model is robust as well.

6.1 Bone age prediction model

Table 6-1: Parameter specification

Component	**Hyperparameters**
Inception V3 Model	
Input shape	Dimensions: (299, 299, 3)
activation	function: sigmoid
loss function	binary_cross_entropy
optimizers	adam, learning rate :0.001
ResNet50 Model	
Input shape	Dimensions:(224, 224, 3)
activation	function: relu
loss function	mean squared error
optimizers	adam, learning rate: 0.0003
Combined Model	
Dense	units: 128 , Activation:'relu'
Dense	units: 64 , Activation:'relu'
Dense	units: 32 , Activation:'relu'
Dense(output)	units: 1
Training Parameters	
Number of Epochs	Inception V3 Model: 60
	Resnet50 Model: 30(Carpal), 30(metacarpal)
	Combined Model: 10
Batch Size	16

7 FUTURE ENHANCEMENTS

- Since we trained and validated our model solely on Dataset provided by RSNA, collection of Regional specific dataset for training the model could be more appropriate.
- Newer Models like Vision Transformers can be implemented to further improve the model.
- Can be integrated in hospital service applications.

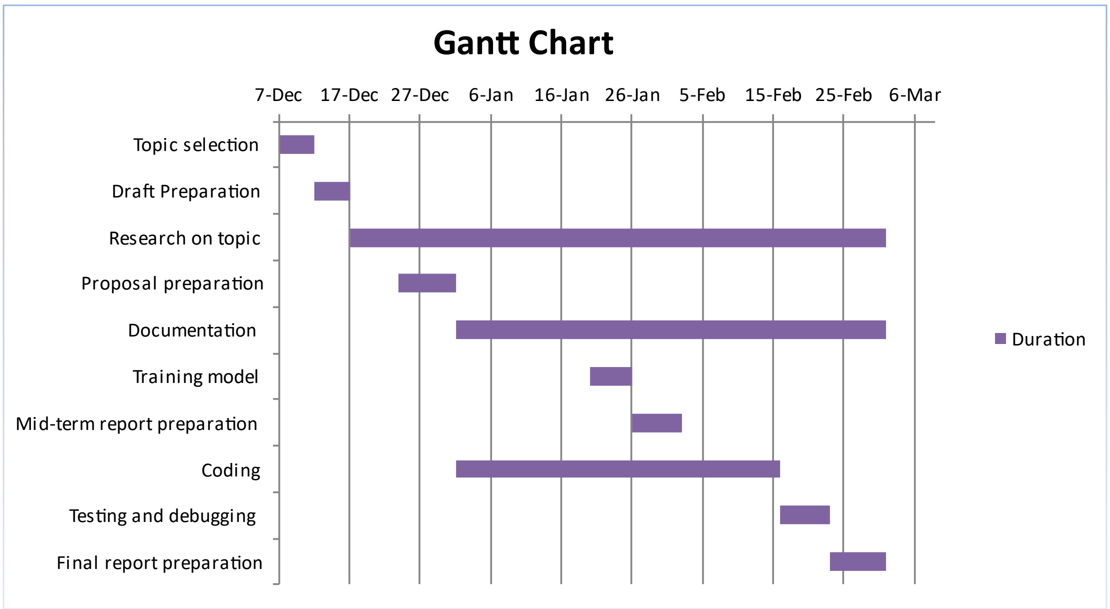
8 CONCLUSION

This system effectively deals with the task of picking out important features of left-hand bones without relying too much on expensive or subjective manual inputs. Based on our bone age prediction, we accomplished a Mean Absolute Error (MAE) of 0.62 years during the training phase and 0.98 years during validation. These results indicate that our performs moderately well in predicting bone age, with a marginally higher error rate observed during validation than training. This suggests that our model may generalize somewhat less effectively unseen data but still gives reasonably accurate prediction. This system shows a lot of potential for being used in clinics and hospitals in the future, which is a big step forward in using technology to improve medical care.

9 APPENDICES

Appendix A: Project Schedule

Table 9-1: Gantt Chart

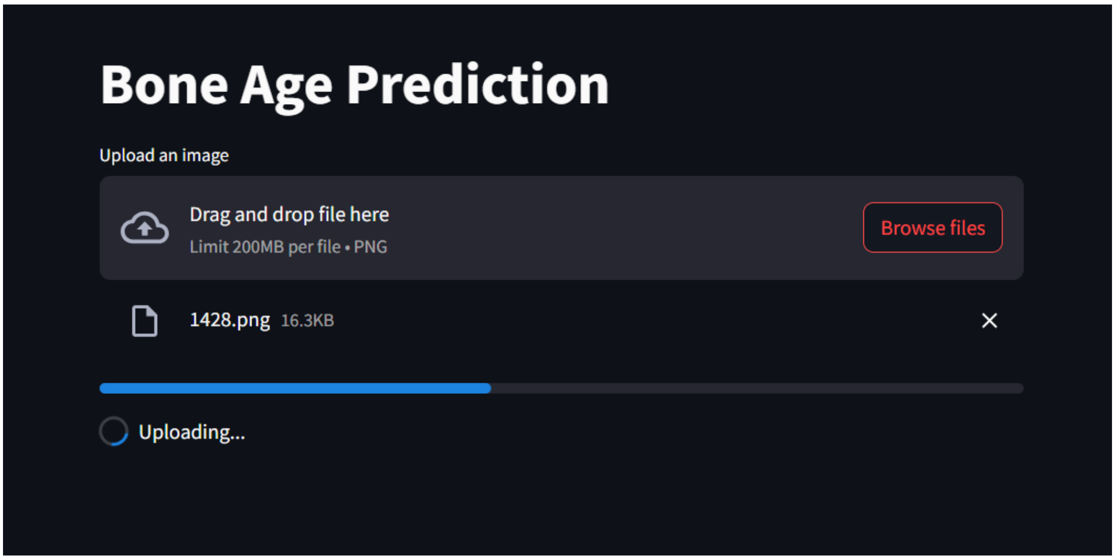


Appendix B: Project Budget

Table 9-2: Project Budget

Tasks	**Cost**
Printing	Rs. 2000
Miscellaneous	Rs. 500
Total	Rs. 2500

Appendix C: Snapshots



Bone Age Prediction

Upload an image

Drag and drop file here

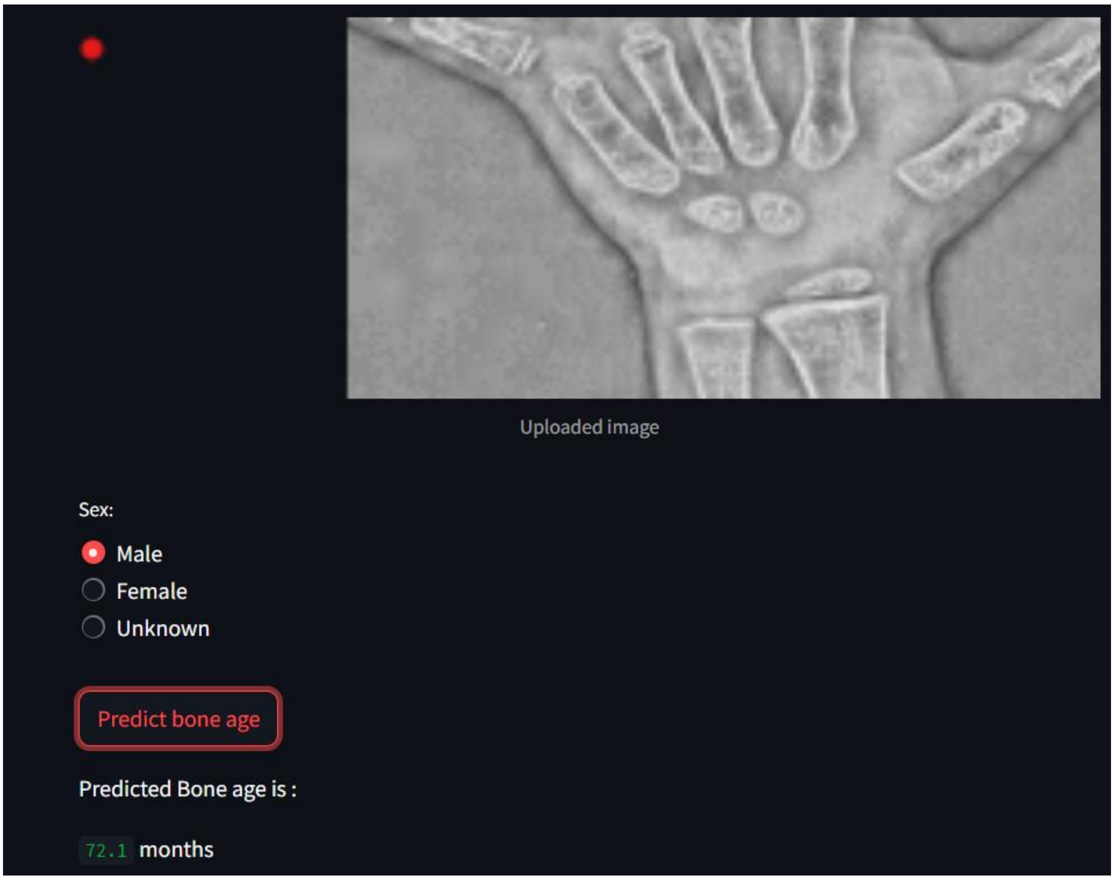
Limit 200MB per file • PNG

Browse files

1428.png 16.3KB

Uploading...

Figure 9-1: Image Upload Section



Uploaded image

Sex:

● Male

○ Female

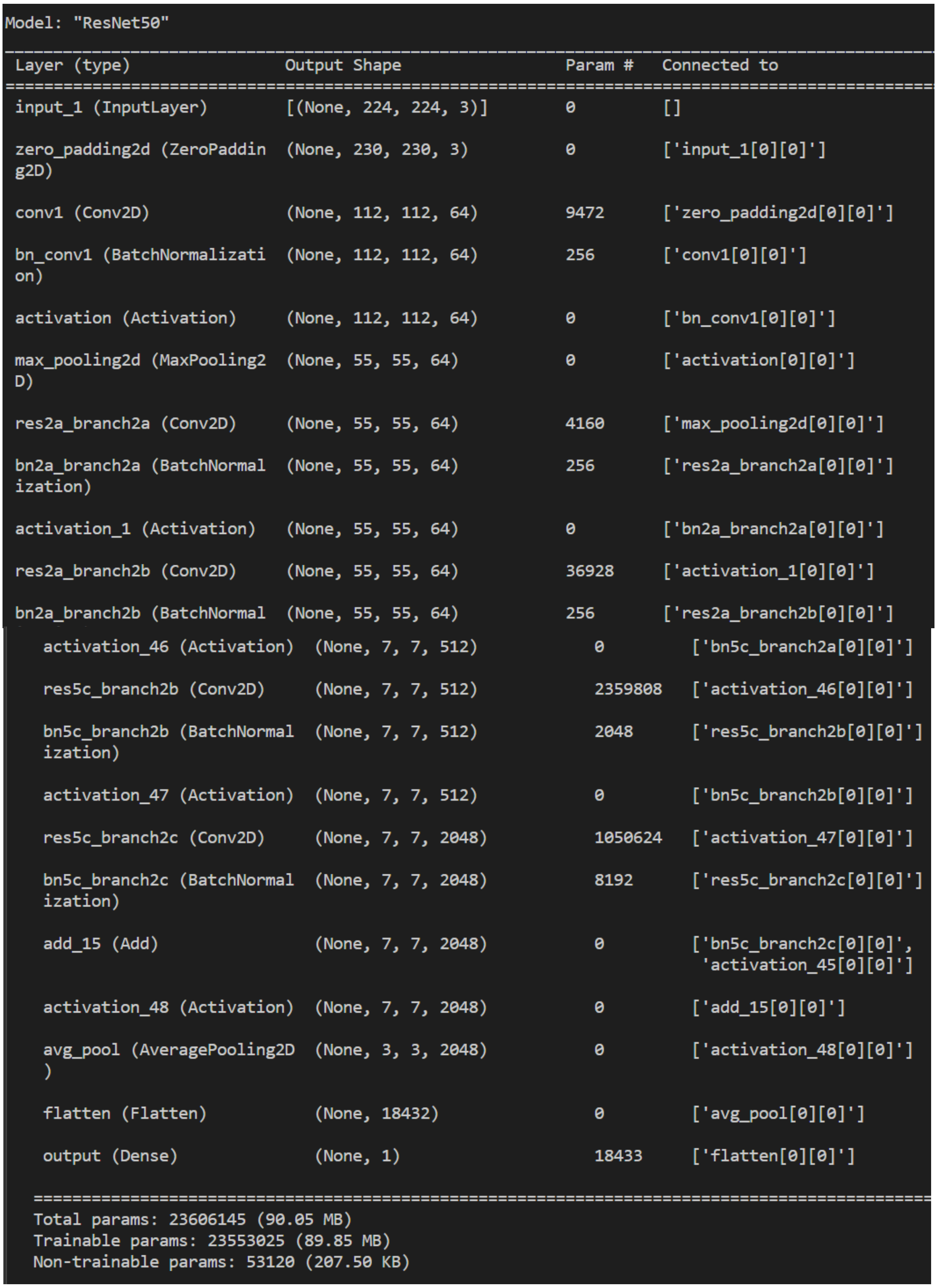
○ Unknown

Predict bone age

Predicted Bone age is :

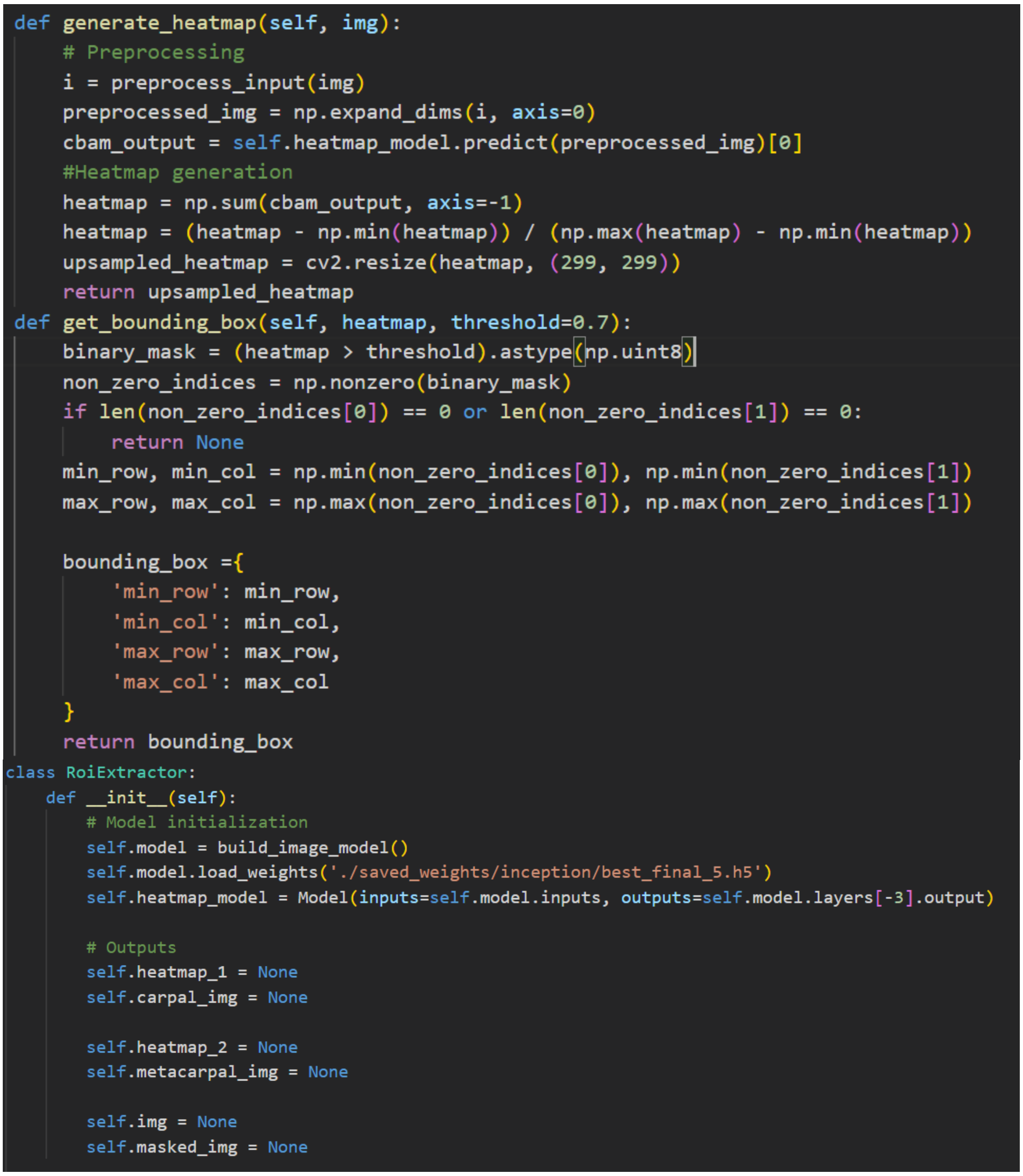
72.1 months

Figure 9-2: Prediction Section



```
Model: "ResNet50"
__________________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
==================================================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]        0         []

 zero_padding2d (ZeroPaddin  (None, 230, 230, 3)          0         ['input_1[0][0]']
 g2D)

 conv1 (Conv2D)              (None, 112, 112, 64)         9472      ['zero_padding2d[0][0]']

 bn_conv1 (BatchNormalizati  (None, 112, 112, 64)         256       ['conv1[0][0]']
 on)

 activation (Activation)     (None, 112, 112, 64)         0         ['bn_conv1[0][0]']

 max_pooling2d (MaxPooling2  (None, 55, 55, 64)           0         ['activation[0][0]']
 D)

 res2a_branch2a (Conv2D)     (None, 55, 55, 64)           4160      ['max_pooling2d[0][0]']

 bn2a_branch2a (BatchNormal  (None, 55, 55, 64)           256       ['res2a_branch2a[0][0]']
 ization)

 activation_1 (Activation)   (None, 55, 55, 64)           0         ['bn2a_branch2a[0][0]']

 res2a_branch2b (Conv2D)     (None, 55, 55, 64)           36928     ['activation_1[0][0]']

 bn2a_branch2b (BatchNormal  (None, 55, 55, 64)           256       ['res2a_branch2b[0][0]']
 activation_46 (Activation)  (None, 7, 7, 512)            0         ['bn5c_branch2a[0][0]']

 res5c_branch2b (Conv2D)     (None, 7, 7, 512)            2359808   ['activation_46[0][0]']

 bn5c_branch2b (BatchNormal  (None, 7, 7, 512)            2048      ['res5c_branch2b[0][0]']
 ization)

 activation_47 (Activation)  (None, 7, 7, 512)            0         ['bn5c_branch2b[0][0]']

 res5c_branch2c (Conv2D)     (None, 7, 7, 2048)           1050624   ['activation_47[0][0]']

 bn5c_branch2c (BatchNormal  (None, 7, 7, 2048)           8192      ['res5c_branch2c[0][0]']
 ization)

 add_15 (Add)                (None, 7, 7, 2048)           0         ['bn5c_branch2c[0][0]',
                                                                     'activation_45[0][0]']

 activation_48 (Activation)  (None, 7, 7, 2048)           0         ['add_15[0][0]']

 avg_pool (AveragePooling2D  (None, 3, 3, 2048)           0         ['activation_48[0][0]']
 )

 flatten (Flatten)           (None, 18432)                0         ['avg_pool[0][0]']

 output (Dense)              (None, 1)                    18433     ['flatten[0][0]']

==================================================================================================
Total params: 23606145 (90.05 MB)
Trainable params: 23553025 (89.85 MB)
Non-trainable params: 53120 (207.50 KB)
```

Figure 9-3: ResNet50 Model Summary



```
def generate_heatmap(self, img):
    # Preprocessing
    i = preprocess_input(img)
    preprocessed_img = np.expand_dims(i, axis=0)
    cbam_output = self.heatmap_model.predict(preprocessed_img)[0]
    #Heatmap generation
    heatmap = np.sum(cbam_output, axis=-1)
    heatmap = (heatmap - np.min(heatmap)) / (np.max(heatmap) - np.min(heatmap))
    upsampled_heatmap = cv2.resize(heatmap, (299, 299))
    return upsampled_heatmap
def get_bounding_box(self, heatmap, threshold=0.7):
    binary_mask = (heatmap > threshold).astype(np.uint8)
    non_zero_indices = np.nonzero(binary_mask)
    if len(non_zero_indices[0]) == 0 or len(non_zero_indices[1]) == 0:
        return None
    min_row, min_col = np.min(non_zero_indices[0]), np.min(non_zero_indices[1])
    max_row, max_col = np.max(non_zero_indices[0]), np.max(non_zero_indices[1])

    bounding_box ={
        'min_row': min_row,
        'min_col': min_col,
        'max_row': max_row,
        'max_col': max_col
    }
    return bounding_box
class RoiExtractor:
    def __init__(self):
        # Model initialization
        self.model = build_image_model()
        self.model.load_weights('./saved_weights/inception/best_final_5.h5')
        self.heatmap_model = Model(inputs=self.model.inputs, outputs=self.model.layers[-3].output)

        # Outputs
        self.heatmap_1 = None
        self.carpal_img = None

        self.heatmap_2 = None
        self.metacarpal_img = None

        self.img = None
        self.masked_img = None
```

Figure 9-4: ROI Extractor

Appendix D: Prediction of testing dataset

Table 9-3: Testing dataset Prediction

Case ID	Sex	Ground truth bone age (months)	Predicted (months)	Error (months)
4360	M	168.93	162.84	6.094248893
4361	M	169.65	161.88	7.772677566
4362	M	73.26	71.04	2.216112365
4364	M	135.46	127.32	8.136953604
4365	M	62.07	44.76	17.30957714
4376	M	191.97	178.68	13.29491556
4378	M	138.74	135.84	2.895783492
4538	F	97.37	107.4	10.03114956
4551	F	184.72	186.48	1.756330985
4557	F	167.69	181.2	13.51496678
4370	M	165.16	164.04	1.123287381
4371	M	120.73	109.08	11.64840285
4372	M	161.64	155.9	5.736397547

References

[1] H. H. Thodberg and S. Kreiborg, "The BoneXpert Method for Automated Determination of Skeletal Maturity," 2008.

[2] W. Eric, K. Bin and W. Xin, "RESIDUAL ATTENTION BASED NETWORK FOR HAND BONE AGE ASSESSMENT," 2017.

[3] V. Iglovikov, A. Rakhlin, K. A. and S. A., "Pediatric Bone Age Assessment Using Deep convolutional Newral Newworks," *International Workshop on Deep Learning in Medical Image Analysis,* 2017.

[4] M. Escobar, F. Torresl and D. L., "Hand pose estimation for pediatric bone age assessment," *Internation Conference on Medical Image Computing And Computer-Assisted Intervention,* pp. 531-539, 2017.

[5] W. Chong and W. Yang, "Attention-based multiple-instance learning for pediatric bone age assessment with efficient and interpretable," *Frontiers,* 2023.

[6] S. Ibrahim and H. A.Ben, "Ridge Regression Neural Network for Pediatric Bone Age Assessment," *Concordia Institure for Information Systems Engineering,* pp. 4-5, 2021.

[7] P. Ewa, G. Arkadiusz and P. Sylwia, "Computer-Assisted Bone Age Assessment: Image Preprocessing and Epiphyseal/ Metaphyseal ROI Extraction," *IEEE Transactions on medical imaging,* pp. 3-6, 2001.

[8] R. Aravinda, P. Sameena and A. Tanweer, "Pediatric Bone Age Assesment using Deep Learning Models," *Manipal Institute of Technology,* 2017.

[9] L. Hyunkwang, T. Shahein, L. Jenny and Z. Maurice, "Fully Automated Deep Learning System for Bone Age Assessment," *Massachusetts General Hospital and Harvard Medical School,* 2017.

[10] M. Satoh and K. Xin, "J-stage," 2015. [Online]. Available: https://www.jstage.jst.go.jp/article/cpe/24/4/24_2015-0011/_article#:~:text=The%20main%20bone%20age%20assessment,resonance%20(MR)%20imaging%20methods..

[11] H. K. Pyeong, M. Y. Hee and R. K. Jeong, "Bone Age Assessmet Using Artificial Intelligence in Korean Pediatric Population," *Korean Journal,* 2023.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. and Wojna, "“Rethinking the Inception Architecture for Computer Vision,”," in *CVPR*, Los Vegas, 2016.