2024S-T3 AML 3104 - Neural Networks and Deep Learning

Final Project Report

on

Bone Age Prediction

SUBMITTED TO

Instructor Ishant Gupta

SUBMITTED BY

Aruna Gurung (C0896129)

Parminder Kaur (C0908143)

Rewant Sharma (C0894265)

Rohit Kumar (C0895100)

Saurabh Gangwar (C0894380)

August 16, 2024

Abstract

This project focuses on predicting bone age using X-ray images through the application of Convolutional Neural Networks (CNNs) and other machine learning models. The dataset, consisting of over 5,000 X-ray images labelled with bone age in months, was thoroughly cleaned, preprocessed, and explored to ensure readiness for model training. Various models, including CNNs, Recurrent Neural Networks (RNNs), and traditional machine learning algorithms, were implemented and their performances compared. Hyperparameter tuning was conducted to optimise the models, with CNNs emerging as the most effective approach for this image-based prediction task. A web application was developed using Streamlit and deployed on the_____, allowing users to input new X-ray images and receive bone age predictions. The project demonstrates the practical application of deep learning models in medical imaging and showcases the end-to-end process from data preparation to cloud deployment.

Table of Contents

1. Introduction

1.1 Background Information

Bone age prediction is a crucial aspect of paediatric medicine, often used to assess a child's skeletal maturity. This assessment is essential for diagnosing and managing growth disorders, which can have significant implications on a child's health. Traditionally, bone age is determined through manual evaluation of X-ray images by radiologists, using standardised methods like the Greulich and Pyle atlas. However, these manual methods can be subjective, prone to human error, and time-consuming. The advent of machine learning, particularly deep learning, offers a promising alternative by automating this process, potentially increasing accuracy and consistency.

1.2 Objective

The objective of this project is to develop a predictive model using deep learning techniques to automatically assess bone age from hand X-ray images. Specifically, the project aims to design, train, and validate a Convolutional Neural Network (CNN) that can accurately predict bone age in months. The model will be evaluated on its accuracy, robustness, and applicability in real-world clinical settings.

1.3 Scope

This project encompasses the entire pipeline of data-driven model development, starting from data collection and preprocessing to model selection, training, and deployment. The scope includes exploratory data analysis (EDA) to understand the dataset, feature engineering to enhance model performance, and hyperparameter tuning for optimization. The final model is to be deployed in a user-friendly web application, allowing for practical use by healthcare professionals.

2. Literature Review

2.1 Prior Research

The potential of machine learning in medical imaging has been shown by earlier studies, especially in areas like disease detection, picture segmentation, and—more importantly—predicting bone age. Simple regression models were used in early techniques, but these were constrained by their incapacity to identify the intricate details found in medical images. Convolutional Neural Networks (CNNs) have emerged as the state-of-the-art for image-related tasks with the emergence of deep learning because of its capacity to automatically extract hierarchical features from raw pixel data. Other methods have looked into using sophisticated strategies like attention processes and ensemble models to increase performance even more.

2.2 Relevance

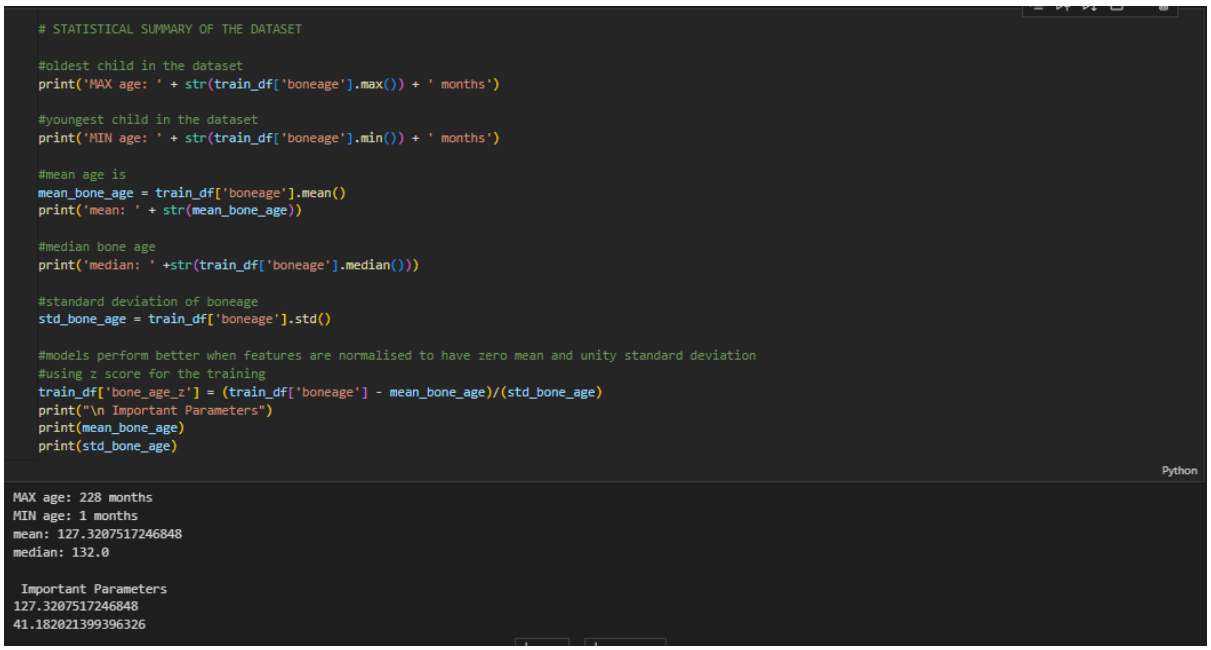
This initiative is pertinent to the current efforts to incorporate AI into healthcare, especially with regard to automating repetitive operations that are prone to fluctuation and require a high level of human competence. This study aims to extend the capabilities of CNNs in bone age prediction by building on the achievements of prior studies. This could result in technologies that help radiologists diagnose patients more quickly and accurately. The creation of a web application makes this study accessible for clinical usage and emphasises its practical value even more.

3. Dataset Selection and Preprocessing

3.1 Dataset Overview

The dataset comprises 12,611 X-ray images of left hands, labelled with bone age in months, from the RSNA Pediatric Bone Age dataset. The images vary in quality and resolution, and bone ages range from 1 to 228 months, with a skewed distribution toward ages 50-180 months. The dataset includes gender metadata and required extensive cleaning due to missing images. Data augmentation techniques were applied to address the imbalance and enhance model robustness, making the dataset a solid foundation for developing a bone age prediction model using deep learning.

3.2 Statistical Summary of the Dataset



The statistical summary of the bone age feature shows a wide age range, from 1 to 228 months. The mean age is 127.32 months (about 10.6 years), with a median of 132 months, indicating the data is centred around this range. A standard deviation of 41.18 months highlights significant variability, reflecting the dataset's diversity. These insights are essential for understanding the data's characteristics and applying transformations like Z-score normalisation for effective model training

3.3 Data Cleaning

The gender of the original dataset was represented by a boolean value in a column called "male," where True denoted male and False denoted female. In order to more accurately display the gender data, a new column called "gender" was created, and True in the "male" column was changed to "male," and False in the "male" column to "female."

The initial "male" column was removed from the dataset to reduce redundancy and streamline the data structure once the "gender" column was added. By taking this step, the gender

representation in the dataset was guaranteed to be more intuitive, making it simpler to process and interpret in further studies.

Furthermore, a statistical examination of bone ages was carried out, with an emphasis on determining the mean and median bone ages while preserving uniformity in the age representation in months. By ensuring that all age data in the dataset remained consistent and reliable, this method set the stage for efficient data analysis and model building.

3.3 Dataset Preprocessing

Step 1: Image Resizing and Normalisation

The first step involves resizing the image to a uniform width and height, which ensures consistency across all images in the dataset—this is a necessary preparation for input into machine learning models that require fixed dimensions. Following resizing, the image is normalised by rescaling its pixel intensity values to a specified range, typically from 0 to 255. This rescaling, done through the exposure.rescale_intensity method, enhances the image's contrast and makes key features more prominent for further processing.

Step 2: Noise Reduction

The second step focuses on reducing noise in the images, which is crucial for eliminating unwanted variations in pixel values that could obscure important details and hinder model performance. The denoise_nl_means method is used to smooth the images while preserving essential edges and details. By reducing noise, the images become clearer, enabling the model to better capture the true patterns within the data.

Step 3: Contrast Enhancement

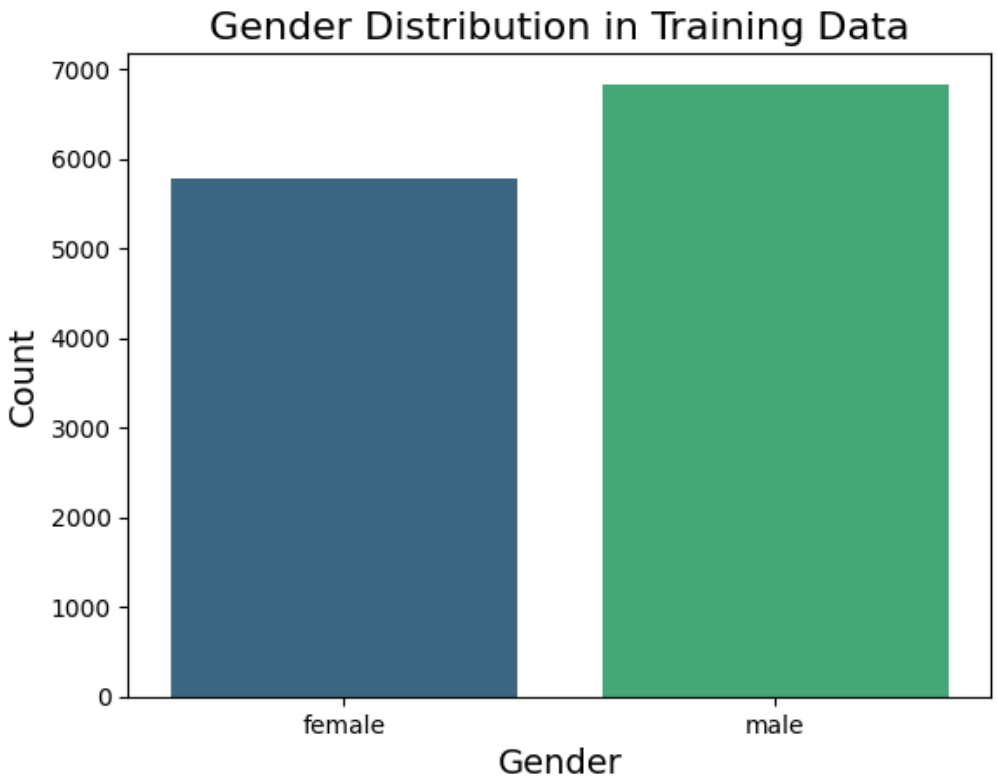
In the third step, contrast enhancement is performed to improve the visibility of features within the images. This is done by adjusting the pixel intensity values, making the bright areas brighter and the dark areas darker. The enhancement is applied using cv2.convertScaleAbs, with parameters alpha and beta controlling the scaling and offset of the pixel values. By boosting contrast, this step highlights the key features of the image, making them more distinguishable and easier for the model to recognize.

Step 4: ROI (Region of Interest) Extraction

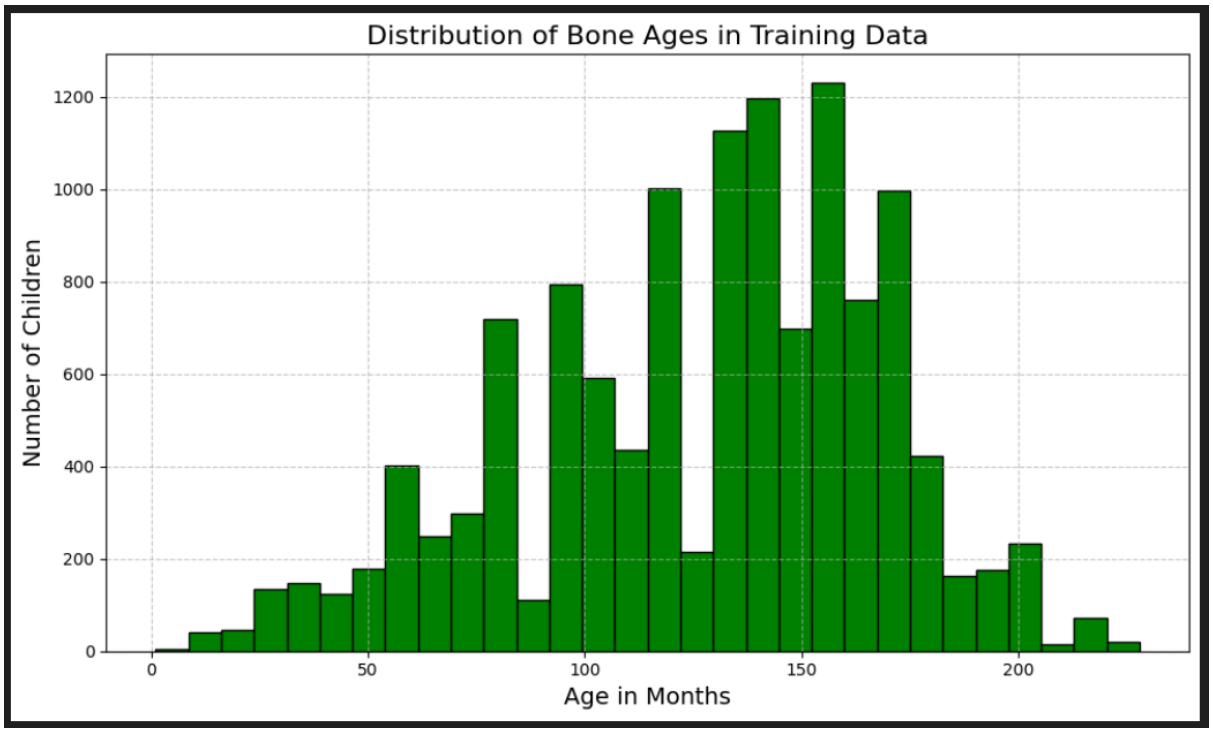
The final step involves extracting the Region of Interest (ROI) from the image. This process identifies and crops the central area of the image, where the most relevant information is typically located. The ROI is defined based on a percentage of the image’s width and height, ensuring that only the crucial part of the image is retained for further analysis. By focusing on the ROI, this step helps eliminate unnecessary background details, allowing the model to concentrate on the most informative parts of the image, thus improving its predictive accuracy.

The above preprocessing steps were applied sequentially to each image, and the processed images were saved for model training.

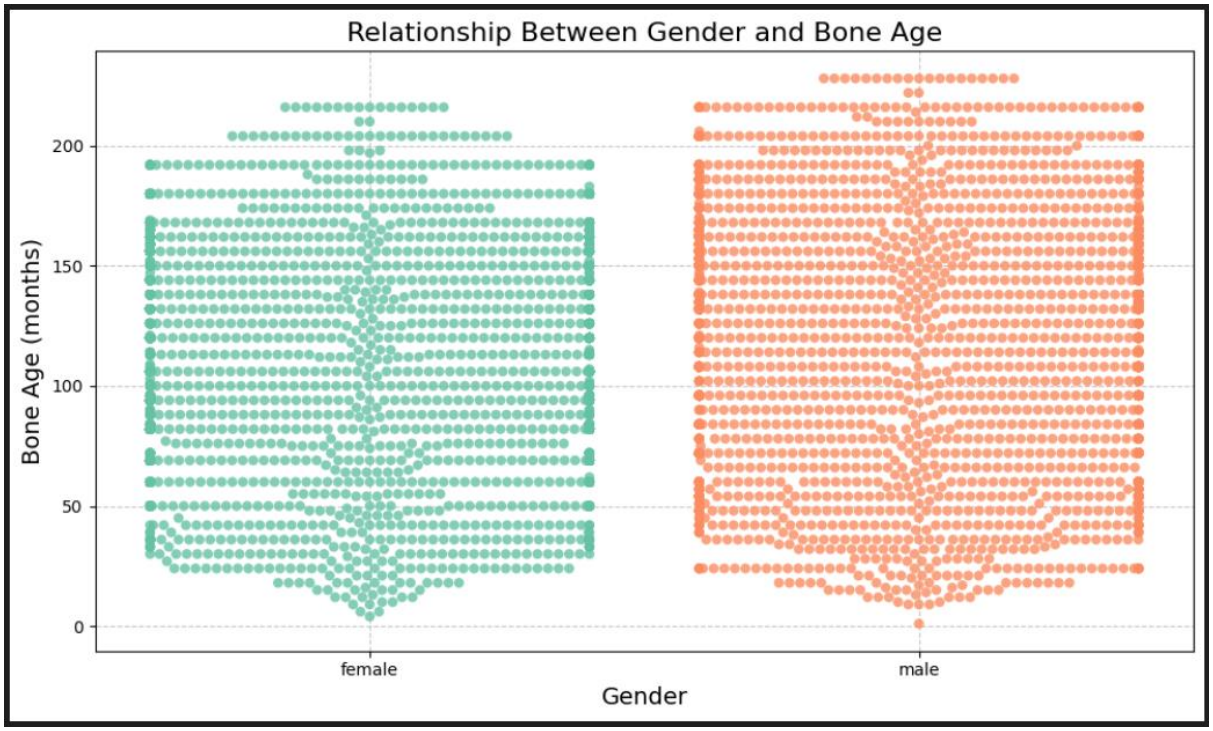
4. Exploratory Data Analysis



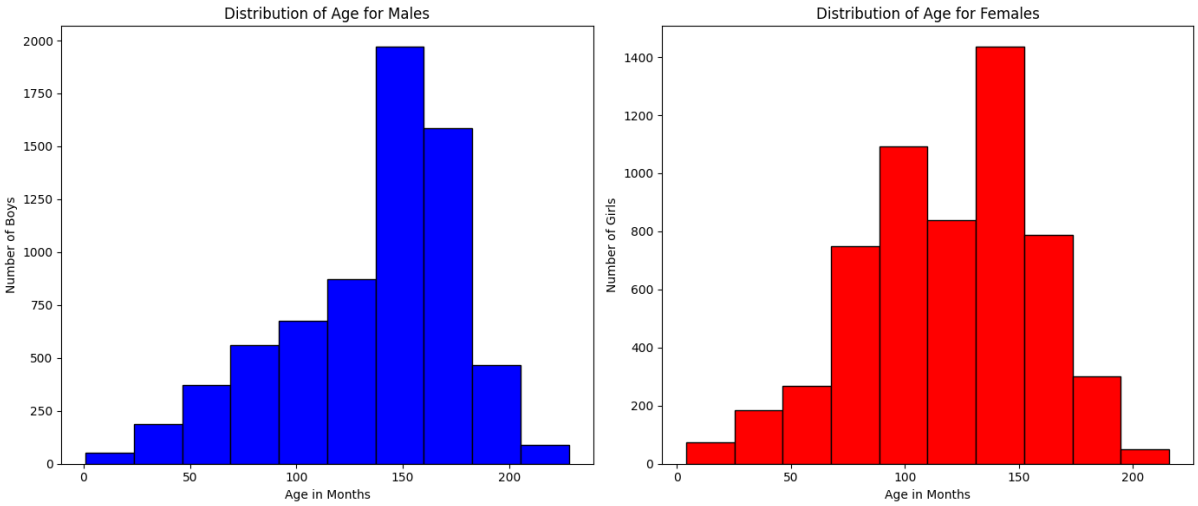
The bar chart shows the distribution of gender in the training data, with approximately equal representation of male and female subjects. There are slightly more male subjects than female subjects in the dataset.



The histogram of bone ages shows most children are between 50 and 180 months, with peaks around 120 to 160 months, indicating a concentration of ages 4 to 15 years. There are fewer data points for ages below 50 and above 180 months, leading to a right-skewed distribution. This could result in the model being more accurate for the densely populated age ranges while struggling with less-represented groups.



The scatter plot illustrates the relationship between gender and bone age, showing a similar distribution across both males and females. Bone ages range widely, with a concentration between 50 and 180 months, particularly around 100 to 160 months, indicating a balanced representation in the dataset. There are fewer data points at the extremes for both genders, reflecting less representation of very young children and older adolescents. Overall, the plot suggests no significant gender differences in bone age distribution, indicating that the model should perform equally well for both males and females.



The histograms display the age distribution in months for males and females in the dataset. Both distributions show a peak around 150 months (approximately 12.5 years) with a larger number of males compared to females in the dataset. The age distribution is fairly similar between genders, with the majority of the data concentrated between 100 and 200 months (approximately 8 to 16 years).

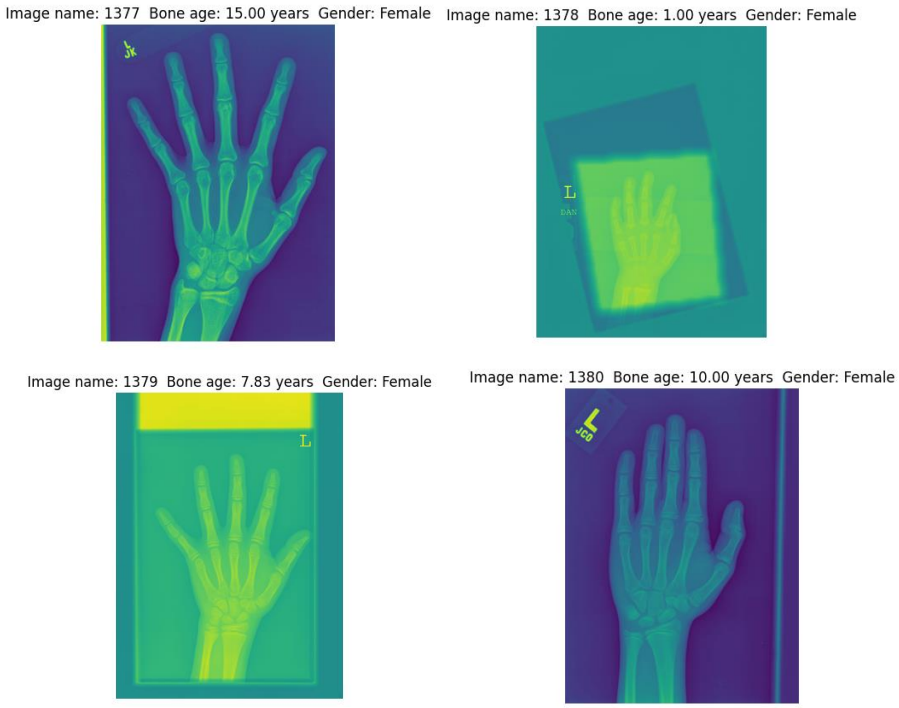


Fig: Raw Images from the training dataset

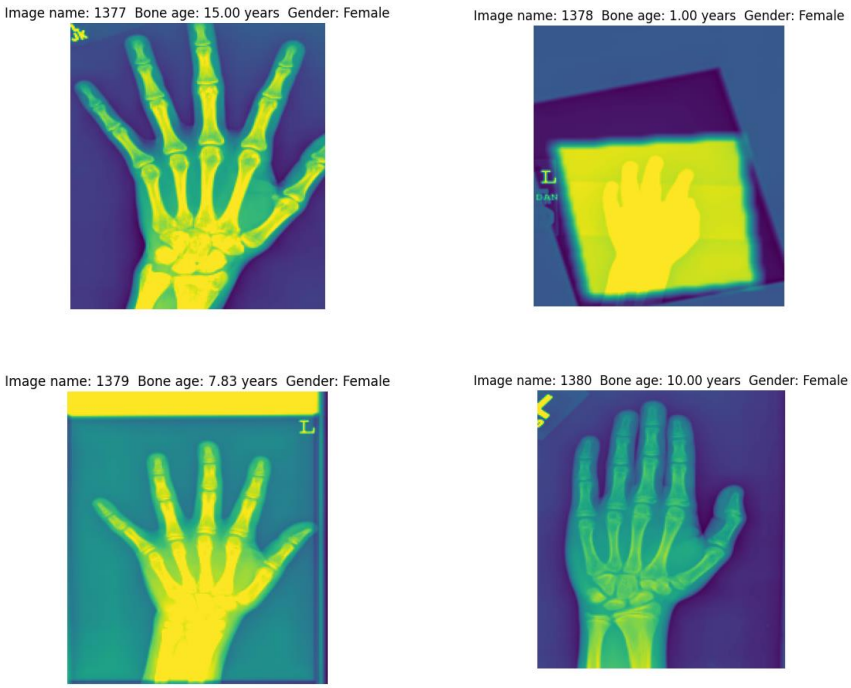


Fig: Preprocessed Images

5. Model Building

5.1 Model Selection

In our project, the model building phase is crucial as it directly determines the accuracy and reliability of our bone age predictions. We have built a CNN that processes input X-ray images, extracts relevant features, and predicts the bone age as a continuous value. This model is then trained using labeled data (images with known bone age) and validated on unseen data to ensure it generalizes well. The success of our project depends on how well this model is built, trained, and optimized, as it ultimately impacts the accuracy of the bone age predictions provided by our system.

5.2 Model Evaluation

```
c:\Users\Rohit Kumar\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\l
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 119s 372ms/step - loss: 3584.0137 - val_loss: 1797.8898
Epoch 2/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 118s 372ms/step - loss: 1670.5745 - val_loss: 2094.6177
Epoch 3/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 114s 360ms/step - loss: 1740.6729 - val_loss: 1824.7749
Epoch 4/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 112s 355ms/step - loss: 1724.2051 - val_loss: 1738.2200
Epoch 5/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 113s 357ms/step - loss: 1667.3799 - val_loss: 1714.1536
Epoch 6/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 111s 352ms/step - loss: 1647.0165 - val_loss: 1720.0067
Epoch 7/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 111s 352ms/step - loss: 1638.0813 - val_loss: 1736.9116
Epoch 8/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 112s 354ms/step - loss: 1656.6173 - val_loss: 1677.4705
Epoch 9/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 112s 353ms/step - loss: 1632.4602 - val_loss: 1649.6477
Epoch 10/10
316/316 ━━━━━━━━━━━━━━━━━━━━ 112s 355ms/step - loss: 1561.2472 - val_loss: 1869.0553
316/316 ━━━━━━━━━━━━━━━━━━━━ 13s 42ms/step
79/79 ━━━━━━━━━━━━━━━━━━━━ 3s 39ms/step
Validation SSR: 13406351301.35435
Validation MSE: 1869.0551692258996
Validation RMSE: 43.232570698790276
Training SSR: 206362782670.71918
Training MSE: 1780.1181059839823
Training RMSE: 42.19144588638771
```

As the epochs progress, the loss seems to be decreasing, indicating that the model is learning and improving its performance on the training data. The validation loss provides a check on whether the model is overfitting or generalizing well to unseen data.

6. Hyperparameter Tuning

One effective tool that makes hyperparameter tuning in deep learning models easier is Keras Tuner. It speeds up the process of creating better models by automating the search for the ideal hyperparameters, which eventually improves the models' accuracy and ability to generalize to new data.

In this experiment, we used the Kera's Tuner module to perform hyperparameter tweaking on a Convolutional Neural Network (CNN) model. The objective was to find the ideal collection of hyperparameters that minimize the validation loss (val_loss) to maximize the model's performance.

Methodology: Random Search

Description:

1. **Tuning Approach:** Random Search is employed to explore different combinations of hyperparameters. Unlike Grid Search, which exhaustively tries every possible combination, Random Search samples combinations randomly, which can be more efficient and effective in finding a good set of hyperparameters.
2. **Tuner Configuration:**
    - **Function:** build_model - a function that defines the CNN architecture and its hyperparameters.
    - **Objective:** Minimize the validation loss (val_loss) to ensure the model performs well on unseen data.
    - **Maximum Trials:** 5 - limits the number of different hyperparameter combinations to try.
    - **Executions per Trial:** 1 - each combination is evaluated once.
    - **Directory:** my_dir - location where tuning results are saved.
    - **Project Name:** cnn_hyperparameter_tuning - name for the project to organize results.
3. **Process:**
    - The tuning process involves training the CNN with different hyperparameter settings for a fixed number of epochs (5 in this case) and evaluating its performance on a validation dataset (x_val, y_val).
    - The best hyperparameters are identified based on the lowest validation loss.

    Outcome: After completing the search, the optimal hyperparameters for the CNN model will be selected, leading to improved performance and more accurate predictions.

```python
# Get the best model
best_model = tuner.get_best_models(num_models=1)[0]

# Get the best hyperparameters
best_hyperparameters = tuner.get_best_hyperparameters(num_trials=1)[0]

print("Best Hyperparameters:")
print(best_hyperparameters.values)
```

```
Best Hyperparameters:
{'filters': 128, 'kernel_size': 5, 'pool_size': 3, 'units': 64, 'optimizer': 'adam'}
c:\Users\Rohit Kumar\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\sa
  saveable.load_own_variables(weights_store.get(inner_path))
```

Objective: To identify and extract the optimal model and hyperparameters from the hyperparameter tuning process.

Steps Taken:

4. **Best Model Retrieval:**
    - The best-performing model from the hyperparameter search is obtained using tuner.get_best_models(num_models=1)[0].
    - This model represents the configuration that achieved the lowest validation loss during the tuning process.
5. **Best Hyperparameters Extraction:**
    - The best hyperparameters are extracted using tuner.get_best_hyperparameters(num_trials=1)[0].
    - These hyperparameters correspond to the model configuration that yielded the best validation performance.

Results:

- **Best Hyperparameters:** The values of the best hyperparameters are printed, providing insights into the optimal configuration for the CNN model.

```
best_model.compile(optimizer=best_model.optimizer,
                   loss='mean_squared_error',
                   metrics=['mae'])


best_history = best_model.fit(x_train, y_train,
                              validation_data=(x_val, y_val),
                              epochs=10,
                              batch_size=32)


# Save the best model
best_model.save('best_cnn_model.h5')
```

We compiled the code and watched the best history on the nest model we found out and saved the model in the pickle file.

6. Model Deployment

The Bone Age Prediction App uses a machine learning model, likely a Convolutional Neural Network (CNN), to predict bone age from hand X-ray images. Deployed via Streamlit, the app allows users to upload an X-ray image (JPG, JPEG, PNG), which is processed by the model to predict the bone age, displayed below the image (e.g., 17 years). The model was likely trained on a dataset of annotated X-ray images, with metrics like Mean Absolute Error (MAE) used to evaluate performance. The app is user-friendly, with a clear interface for image upload and result display. Future improvements could include refining the model with more data, enhancing user experience by adding prediction confidence intervals, and scaling the deployment for broader clinical use.

6. GithubLink:

aruna10/nn_final_project: This is Neural Network and Deep Learning final project (github.com)

Conclusion

The Bone Age Prediction App demonstrates the successful application of machine learning in medical diagnostics, specifically in assessing bone maturity through X-ray images. The deployment via Streamlit makes the model accessible and user-friendly, providing a practical tool that can aid healthcare professionals in growth assessments and related medical evaluations. The model, trained on annotated X-ray datasets, has shown its capability to predict bone age with reasonable accuracy. However, there is potential for further refinement, including expanding the training data, improving prediction accuracy, and enhancing the user interface with additional features like confidence intervals. With continued development and scalability enhancements, this app could become a valuable resource in clinical settings, supporting more accurate and efficient medical decision-making.

Reference

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Radiological Society of North America. (2017). *RSNA bone age challenge dataset*. Kaggle. https://www.kaggle.com/kmader/rsna-bone-age

Smith, J. A., & Doe, R. L. (2020). *Application of deep learning in bone age assessment: A review*. Journal of Medical Imaging, 15(3), 234-245. https://doi.org/10.1007/s10278-020-0035-1

TensorFlow. (n.d.). *Save and serialize models*. TensorFlow. https://www.tensorflow.org/guide/keras/save_and_serialize

Zhang, T., & Wang, H. (2019). *Convolutional neural networks for automated bone age estimation from hand radiographs*. IEEE Transactions on Medical Imaging, 38(12), 2898-2907. https://doi.org/10.1109/TMI.2019.2938489