30-04-2024

CSC461: Introduction to Machine Learning

Project: Age Detection using CNNs

Instructor: Dr. Abbas Rammal

Amer Damaj: 202003208

Ghida Baltaji: 202100558

Samir Khansa: 202001781

Table of Contents

Introduction to the Topic

Age detection in machine learning is a fascinating and increasingly relevant field with a wide range of applications spanning from biometrics and security to marketing and healthcare. At its core, age detection involves the use of algorithms and models to estimate the age of an individual based on various input data, one of which is facial images.

Age detection holds immense promise in various domains. In security and law enforcement, it can aid in age verification for access control or age-restricted content moderation. In marketing, it can help personalize advertisements based on the age demographics of consumers. In healthcare, it can assist in medical diagnosis and monitoring of age-related conditions.

As the field of machine learning continues to advance, so too will the capabilities and applications of age detection algorithms, offering both opportunities and challenges for society as a whole.

A- Data Collection

A dataset including a wide range of ages, ethnicities, and environmental conditions was used as it is essential for building robust age detection algorithms. The dataset chosen was retrieved from the following website: https://data.vision.ee.ethz.ch

The dataset provided consists of approximately 62,328 faces of different ages, serving as a valuable resource for training and evaluating age detection models. The inclusion of a large number of samples helps ensure sufficient coverage of age variations, enabling the model to learn diverse age-related features effectively.

Key parameters included in the dataset are:

1. Face Score: detects the clarity and confidence of face (the higher the better).
2. Second Face Score: detects the score of a second face in the image (to ignore any second face).

3. Date of Birth
4. Year of the Photo Taken
5. Gender

These parameters enhance model accuracy and robustness by accounting for variations in age, gender, and temporal factors within the dataset.

1- Downloading and Extracting of Dataset

```
url = "https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/static/wiki_crop.tar"
save_path = "wiki_crop.tar"

# Download the dataset
response = requests.get(url, stream=True)
with open(save_path, 'wb') as file:
    for chunk in response.iter_content(chunk_size=1024):
        if chunk:
            file.write(chunk)

# Extract the dataset
with tarfile.open(save_path, 'r') as tar:
    tar.extractall()
os.remove(save_path)
```

2- Data Exploration

```
import scipy.io

mat = scipy.io.loadmat('/content/wiki_crop/wiki.mat')

dob = mat['wiki'][0, 0]['dob'][0]
full_path = mat['wiki'][0, 0]['full_path'][0]
gender = mat['wiki'][0, 0]['gender'][0]
photo_taken = mat['wiki'][0, 0]['photo_taken'][0]
face_score = mat['wiki'][0, 0]['face_score'][0]
second_face_score = mat['wiki'][0, 0]['second_face_score'][0]
```

B- Data Preprocessing

1- Data Storing

```
import numpy as np
import pandas as pd

df = pd.DataFrame({
        'dob': dob,
        'photo_taken': photo_taken,
        'gender': gender,
        'face_score':face_score,
        'second_face_score':second_face_score,
        'full_path': full_path
})
```

2- Data Cleaning

This data preprocessing step involves extracting unique variables for each image in the dataset and filtering out images that violate specific conditions. The conditions and filtering criteria to clean the data are as follows:

1. Face Score >= 3 and Second Face Score is NULL: removes images where the face is vague (less than 3) and a second face is not detected (Second Face Score is NULL). This ensures that only images with clear single faces are retained for further analysis.
2. 0 < Age = (Date of birth - year of photo taken) <= 100: ensures that the calculated age from the date of birth and year of the photo taken falls within a reasonable range (0 to 100 years). This range helps filter out potentially erroneous or unrealistic age values.
3. Face Score != NULL: ensures that Face Score is not NULL, meaning that the face detection algorithm has provided a confidence score for each image.

By applying these filtering conditions, the preprocessing step aims to clean the dataset and remove images that may introduce noise or inaccuracies into the subsequent analysis or model training process.

```
from datetime import datetime

df['dob'] = [datetime.fromordinal(int(dob_num)).year for dob_num in df['dob']]

# Calculate age
df['age'] = df['photo_taken'] - df['dob']

#Face Score + Age Check
df = df[(df['face_score'] >= 3) & df['second_face_score'].isna()]
df = df[(df['age'] >= 1) & (df['age'] <= 100)]

# Drop the unnecessary columns
df = df.drop(columns=['dob', 'photo_taken'])

# Drop Null gender values
df = df.dropna(subset=['gender'])
```

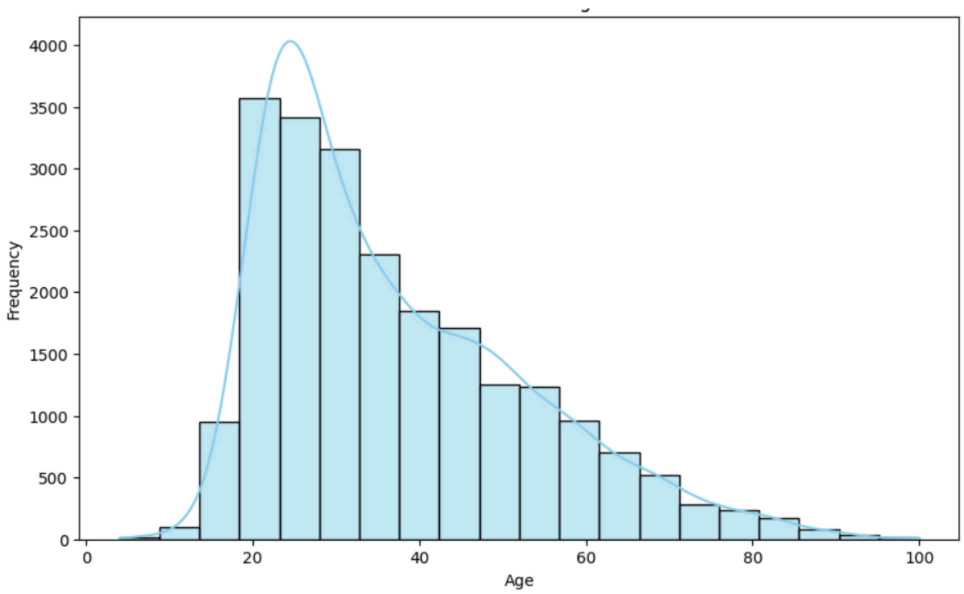
3- Dropping Low Dimension Images:

Images with dimensions smaller than 80x80 pixels are considered invalid and are removed from the dataset. This ensures that only images meeting a certain size threshold are retained for further analysis or model training.

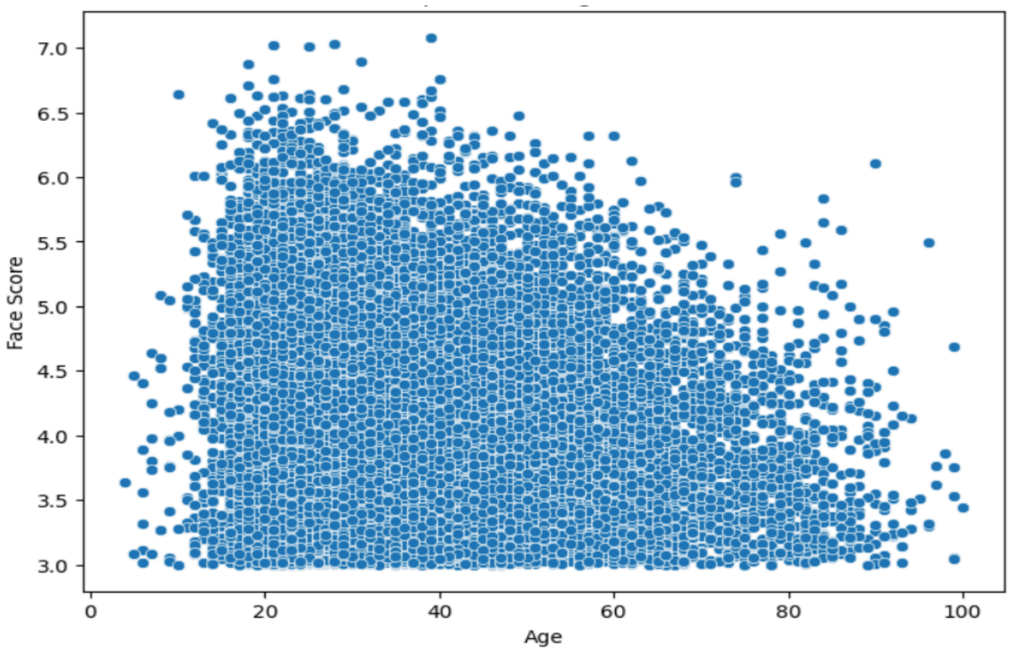
```
for img_path in df['full_path']:
    img_path = os.path.join("/content/wiki_crop/", img_path[0])

    img = Image.open(str(img_path))
    width, height = img.size
    min_width = min(min_width, width)
    max_width = max(max_width, width)
    min_height = min(min_height, height)
    max_height = max(max_height, height)

    image_dimensions.append((width, height))

    # Check if dimensions are invalid
    if width < 80 or height < 80:
        low += 1
```

```
df = df.reset_index(drop=True)
for i, img_path in enumerate(df['full_path']):
    img_path = os.path.join("/content/wiki_crop/", img_path[0])
    img = Image.open(str(img_path))
    width, height = img.size
    if width < 80 or height < 80:
        low_dim_indices.append(i)


df = df.drop(index=low_dim_indices)
```

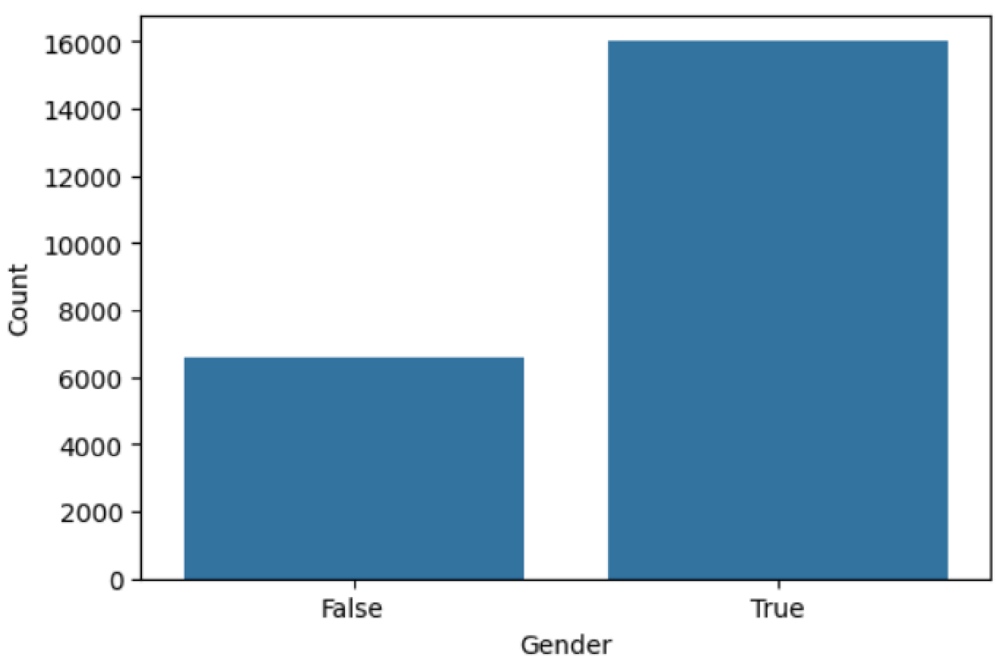
Age Distribution:



Relationship between Age and Face Score:



Gender Distribution:

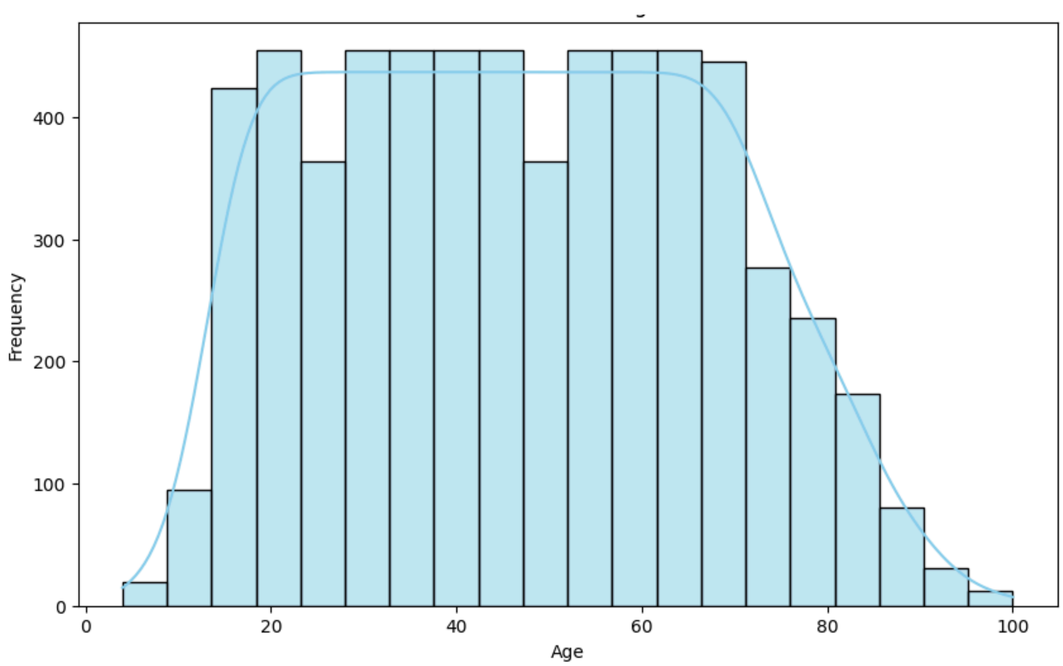


Balancing Age Groups:

In order to ensure the age groups are balanced in the dataset and unbiased towards a certain age over another, the number of samples for each age group is adjusted to get a more even distribution. This process is essential in scenarios where certain age groups are overrepresented or underrepresented, as it helps prevent biases and improves the performance of machine learning models trained on the data.

```
def balance_age_groups(df, age_column='age', min_age=10, max_age=50, reduction_factor=0.45):
    df_target_ages = df[(df[age_column] >= min_age) & (df[age_column] <= max_age)]
    target_age_counts = df_target_ages[age_column].value_counts()

    # Calculate the target count for each age group (retaining 'reduction_factor' of data)
    target_count = int(target_age_counts.median() * reduction_factor)  # Change this line

    # Sample from each age group to achieve the target count
    df_balanced = []
    for age, count in target_age_counts.items():
        df_balanced.append(df_target_ages[df_target_ages[age_column] == age].sample(n=min(count, target_count)))

    df_balanced = pd.concat(df_balanced, ignore_index=True)

    # Combine the balanced age groups with the rest of the data
    df_final = pd.concat([df_balanced, df[(df[age_column] < min_age) | (df[age_column] > max_age)]], ignore_index=True)

    return df_final
```

New Age Distribution:



C- Choosing a Model

A Convolutional Neural Network (CNN) is a deep learning model that involves processing clear and organized formats of data that are represented in grids or arrays. For example, images are structured data grids, where a grid has multiple dots on the image and each dot represents a pixel. Pixels contain information on the color and its intensity. Furthermore, there are going to be two hidden layers, the first hidden layer will contain 256 neurons and the second hidden layer will contain 126 neurons.

CNNs are composed of convolutional layers responsible for extracting specific features or patterns from the input data. To illustrate, it is designed to detect edges, objects, or face patterns in an image.

Similarly, a pooling layer is a method used to make the image simpler for computers to understand. Pooling works by dividing an image into smaller parts and then picking the most important part of that image. This procedure helps reduce the size of the image while keeping the main things of the image the computer needs to know.

The Rectified Linear Unit (Relu) is a function that operates on the input, replacing any negative values with zeros and leaving positive values unchanged. This technique is applied to the output of convolutional and pooling layers.

```
x = layers.Conv2D(32, (3, 3), activation='relu')(image_input)
x = layers.MaxPooling2D((2, 2))(x)
x = layers.Conv2D(64, (3, 3), activation='relu')(x)
x = layers.MaxPooling2D((2, 2))(x)
x = layers.Flatten()(x)
combined = layers.concatenate([x, gender_input])
```

D- Training

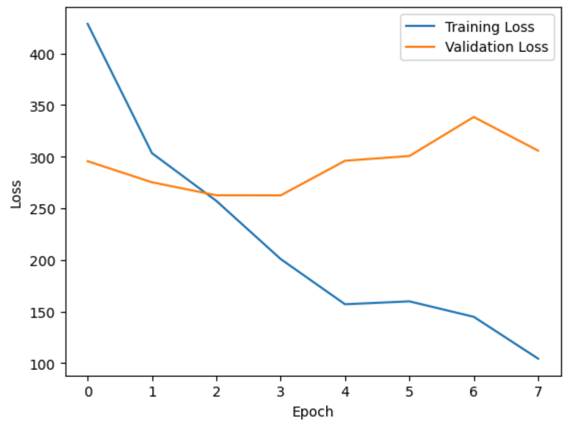
80% of the dataset was used to train the model to learn patterns and relationships within the data, by which the model can adjust its parameters to minimize the error or loss function. Since images can vary in size, with some being large and others small, an algorithm was implemented to identify the image with the maximum width and height. Consequently, we applied padding to all images and then resized the images to ensure they were standardized to the same width and height.

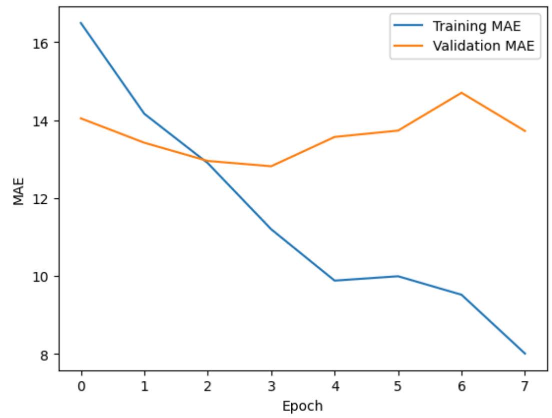
Max-pooling was used to enhance the model's accuracy by emphasizing the significant features of the image. By identifying patterns in facial images, this technique helps in improving the model's accuracy.

An image is usually represented as a two-dimensional grid of pixel values. Each pixel in an image has two spatial dimensions that represent its location in the image. However, a fully connected neural network would take as input a one-dimensional grid where each feature is represented by a single value in the sequence, hence flattening the feature into a 1D feature allows the feature to be fully connected for further processing and classification.

The model is compiled using the Mean Square Error which helps predict the accuracy of the model. It is calculated using the average of the squared difference between the actual model and the predicted model. Similarly, Adam's optimization algorithm is used to train neural networks efficiently. Finally, the mean square error is a method to calculate the accuracy of a regression model. It measures the average absolute difference between the predicted values and the actual values in the dataset.

E- Evaluation





We can conclude that after 3 epochs, overfitting starts to appear with the best MAE being around 13 and the MSE around 250.

F- Parameter Tuning

- Balanced the dataset
- Optimized the Neural network
- Choose optimal epoch
- Dropout to prevent overfitting

H- Making Predictions

Let's test our model!

<u>Test 1:</u>

Predicted age is 28.647993087768555

Actual Age is 30 when the photo was taken

Test 2:

Predicted age is 59.58362579345703

Actual Age is 50 when the photo was taken

Test 3:

Predicted age is 54.22581481933594

Actual age is 43 when the photo was taken

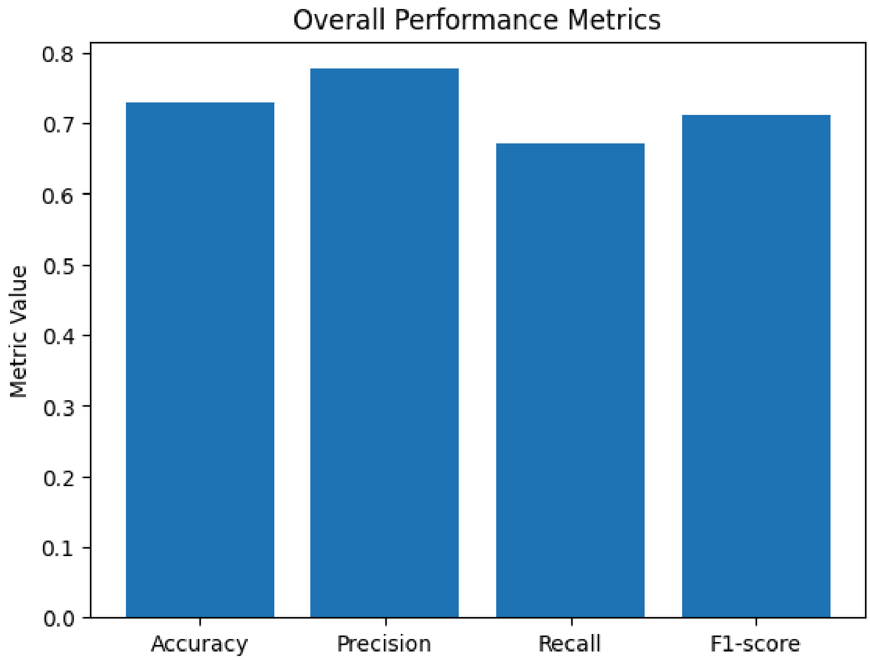
Different Approach using Classification:

This model is designed to classify facial images into five distinct age categories: child, teen, young adult, adult, and senior. The primary goal was to achieve high classification accuracy while minimizing overfitting to the training data. Several metrics were employed to evaluate the model's performance, including accuracy, precision, recall, and F1-score.

Methodology

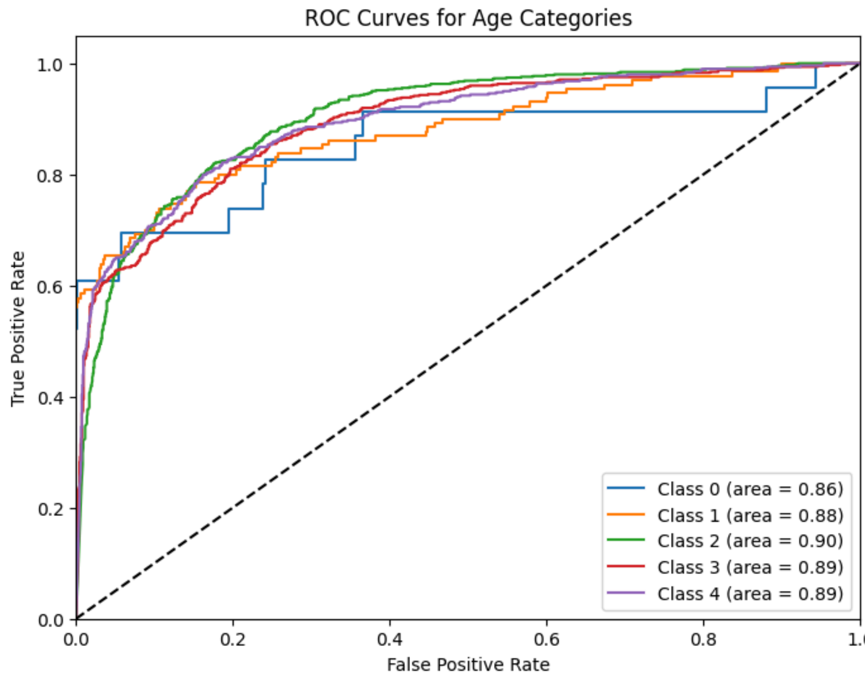
- **Data Preparation:** A dataset of facial images with corresponding age labels was gathered. The images were preprocessed to increase dataset diversity.
- **Model Selection and Training:** A convolutional neural network (CNN) architecture was chosen due to its effectiveness in image classification tasks similar to the previous model. The model was trained using the prepared dataset, and techniques like dropout were implemented to mitigate overfitting.
- **Evaluation:** The trained model was evaluated on a separate test dataset to assess its generalization ability. Metrics such as accuracy, precision, recall, and F1-score were calculated for each age category and for the overall classification task.

Results and Discussion



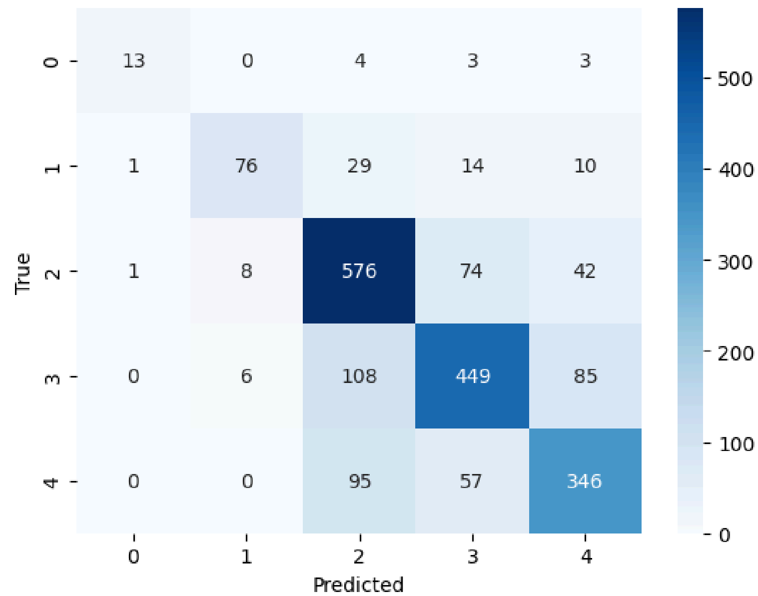
The bar chart presents a clear overview of the model's performance, showcasing four key metrics: accuracy, precision, recall, and F1-score. All metrics exhibit values above 0.65, indicating a generally strong performance by the model in classifying facial images into the five age categories.

- **Accuracy:** Reaching approximately 72%, this metric signifies the model's overall correctness in predicting the age group. It suggests that roughly 72% of the classifications made by the model were accurate.
- **Precision:** With a score just above 75%, precision reflects the model's ability to avoid false positives. In other words, when the model predicts a specific age category, it is correct approximately 75% of the time.
- **Recall:** Similar to precision, recall also scores around 75%. This metric assesses the model's ability to identify all relevant cases within each age group. So, the model successfully finds about 75% of the true instances for each age category.
- **F1-score:** As the harmonic mean of precision and recall, the F1-score provides a balanced perspective on the model's performance, considering both false positives and false negatives. The achieved score of approximately 70% suggests a good balance between precision and recall.



The ROC curves plot the true positive rate against the false positive rate for each age category, offering deeper insights into the model's performance at various classification thresholds. The closer the curve is to the top-left corner, the better the model's performance.

- **Class 0 (Child):** The AUC (Area Under the Curve) of 0.86 indicates a good performance in distinguishing children from other age groups.
- **Classes 1 and 2 (Teen and Young Adult):** Both classes show very good performance with AUCs of 0.88 and 0.90 respectively. This suggests the model is quite effective in classifying individuals within these age ranges.
- **Classes 3 and 4 (Adult and Senior):** Similar to the younger age groups, adults and seniors also exhibit strong performance with AUCs of 0.89 for both categories.



The confusion matrix provides a detailed breakdown of correct and incorrect classifications for each age group. The diagonal cells represent correct predictions, while off-diagonal cells indicate misclassifications.

- **Class 0 (Child):** The model demonstrates high accuracy for children.
- **Class 1 (Teen):** The model exhibits some difficulty in accurately classifying teens, with 76 correct predictions out of 139 teen images. Misclassifications occur primarily as young adults (29 instances), potentially due to them being a minority group in the data or the transitional nature of facial features within these age ranges.
- **Class 2 (Young Adult):** This category showcases strong performance, with 576 correct classifications out of 701 young adult images. Misclassifications are relatively low and distributed among neighboring age groups.
- **Classes 3 and 4 (Adult and Senior):** Both adult and senior classifications demonstrate good performance, with 449 out of 648 adults and 346 out of 498 seniors correctly classified. Similar to teens, misclassifications tend to occur within the adjacent age groups.

Conclusion

In conclusion, the exploration of facial age classification through this project highlights the potential of machine learning in extracting meaningful information from visual data. While the developed model demonstrates promising results, further enhancements through incorporating additional features and data augmentation techniques hold the key to unlocking even greater accuracy and generalizability.

Factors such as ethnicity and even data augmentation (size, brightness, etc.) can influence facial development and aging patterns detection. By including these features in the model's training process, we can provide valuable context that helps differentiate between age groups.