GENDER AND AGE PREDICTION

Submitted by

SINDHU KALEESWARAN [RA2011026010082]

APARNA SURESH[RA2011026010099]

CHEREDDY SOWMYA SRI [RA2011026010113]

Under the Guidance of

Dr.MAIVIZHI.R

Assistant Professor, Department of Computational Intelligence

In partial satisfaction of the requirements for the degree of

BACHELORS OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in Artificial Intelligence & Machine Learning

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

May 2023

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled **"GENDER AND AGE PREDICTION"** is the bonafide work done by **SINDHU KALLESWARAN [RA2011026010082], APARNA SURESH[RA2011026010099], CHEREDDY SOWMYA SRI[RA20110260100113]** who carried out under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Faculty In-Charge
Dr.MAIVIZHI.R
Assistant Professor
Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai

HEAD OF THE DEPARTMENT
Dr. R Annie Uthra
Professor and Head ,
Department of Computational Intelligence,
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai

ABSTRACT

Automatic prediction of age and gender from face images has drawn a lot of attention recently,due it is wide applications in various facial analysis problems. However, due to the large intra-class variation of face images (such as variation in lighting, pose, scale, occlusion), the existingmodels are still behind the desired accuracy level, which is necessary for the use of thesemodels in real-world applications. In this work, we propose a deep learning framework, basedon the ensemble of attentional and residual convolutional networks, to predict gender and agegroup of facial images with high accuracy rate. Using attention mechanism enables our model tofocus on the important and informative parts of the face, which can help it to make a moreaccurate prediction. We train our model in a multi-task learning fashion, and augment thefeature embedding of the age classifier, with the predicted gender, and show that doing so canfurther increase the accuracy of age prediction. Our model is trained on a popular face age andgender dataset, and achieved promising results. Through visualization of the attention maps ofthe train model, we show that our model has learned to become sensitive to the right regions ofthe face.

TABLE OF CONTENTS

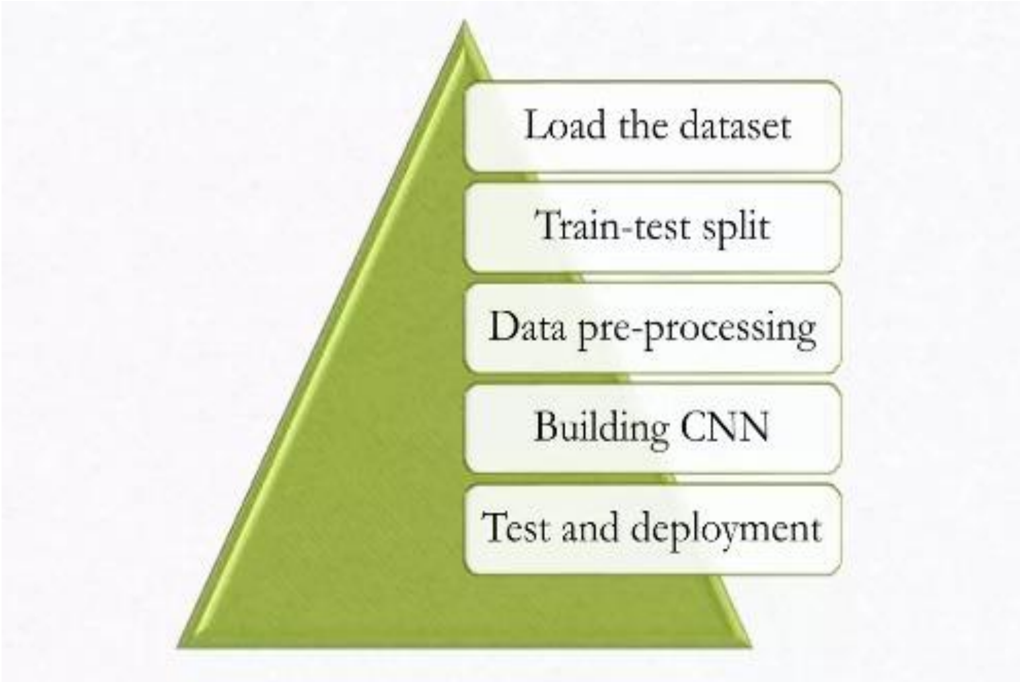
INTRODUCTION

Gender and age play a significant role in interpersonal interactions among people who live in communities. The use of smart gadgets has expanded as technology has progressed, and social media has begun to draw everyone's attention. Daily studies on gender and age prediction have grown in prominence, it increases the number of apps that use such techniques. In these applications, facial photographs are commonly employed since they contain useful information that may be used to extract human interaction.For gender detection and age prediction, Image processing, feature extraction, and classification steps are usually used. These steps may change based on the objective of the study and the characteristics to be used.The face images were processed using a variety of approaches, and calculations were performed based on the results of the investigations. For image processing, there are two basic and typical which we need to follow.Image enhancement is the process of improving an image so that the resultant image is of higher quality and can be used by other applications. The most popular technique for extracting information from an image is the other technique. The image is divided into a specified number of parts or objects in order to solve the challenge and this procedure is called Segmentation. Due to the accuracy of its classification technique, deep learning techniques are a variety of tasks such as classification, feature extraction, object recognition, and so on, it helps in gender and age prediction.

PROBLEM STATEMENT

Age and gender prediction from facial images is an important problem in computer vision with many real-world applications, such as human-computer interaction, marketing, and surveillance. However, accurate age and gender prediction is a challenging task due to variations in lighting, facial expression, and pose. Traditional methods for age and gender prediction rely on handcrafted features and shallow machine learning models, which may not capture the complex relationships between facial features and age/gender. Therefore, there is a need for a deep learning-based approach that can automatically learn discriminative features from large amounts of data and achieve high accuracy in age and gender prediction tasks. In this study, we propose a deep learning-based approach for age and gender prediction from facial images using convolutional neural networks (CNNs). Our goal is to develop a model that can accurately predict the age and gender of a person from their facial image under various real-world conditions. We will evaluate our model on a publicly available dataset and compare its performance with state-of-the-art methods. The results of this study can have practical implications for various domains, including human-computer interaction, marketing, and security.

METHOLODOGY



Grayscale conversation:
The conversion of a color image into a grayscale image is converting the RGB values (24 bit) into grayscale value (8 bit). Grayscale simplifies the algorithm and reduces computational requirements.

Resizing:
Deep learning models train faster on small images. Neural networks receive inputs Of the same size, all images need to be resized to a fixed size before inputting them to the CNN. Here we are resizing it into 128 x 128.

Storing in numpy:
Neural networks can handle only numpy array. NumPy arrays takes significantly less amount of memory. So we are appending all the images into a numpy array.

Normalizing:
Data normalization is used to standardize data which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network.

MODULE DESCRIPTION

Module name: Age and Gender Prediction using Deep Learning
Description: This module uses deep learning techniques to predict the age and gender of a person from their facial image. It takes an input image as input and produces two outputs: the predicted age and the predicted gender. The module is designed to be integrated into various applications such as security systems, advertising platforms, and healthcare systems, where age and gender information is required for decision-making purposes.

Functionality:
- Preprocessing: The module preprocesses the input image to normalize lighting, color, and facial expression variations.
- Feature extraction: The module extracts high-level features from the preprocessed image using a deep convolutional neural network (CNN).
- Age prediction: The module uses the extracted features to predict the age of the person from the image. It uses a regression model that outputs a continuous value for age.
- Gender prediction: The module also predicts the gender of the person from the image. It uses a classification model that outputs either "male" or "female" as the predicted gender.
- Output: The module outputs the predicted age and gender for the input image.

Inputs:
- Image: The input image must be a facial image of a person. It can be in any standard image format (e.g., JPG, PNG).

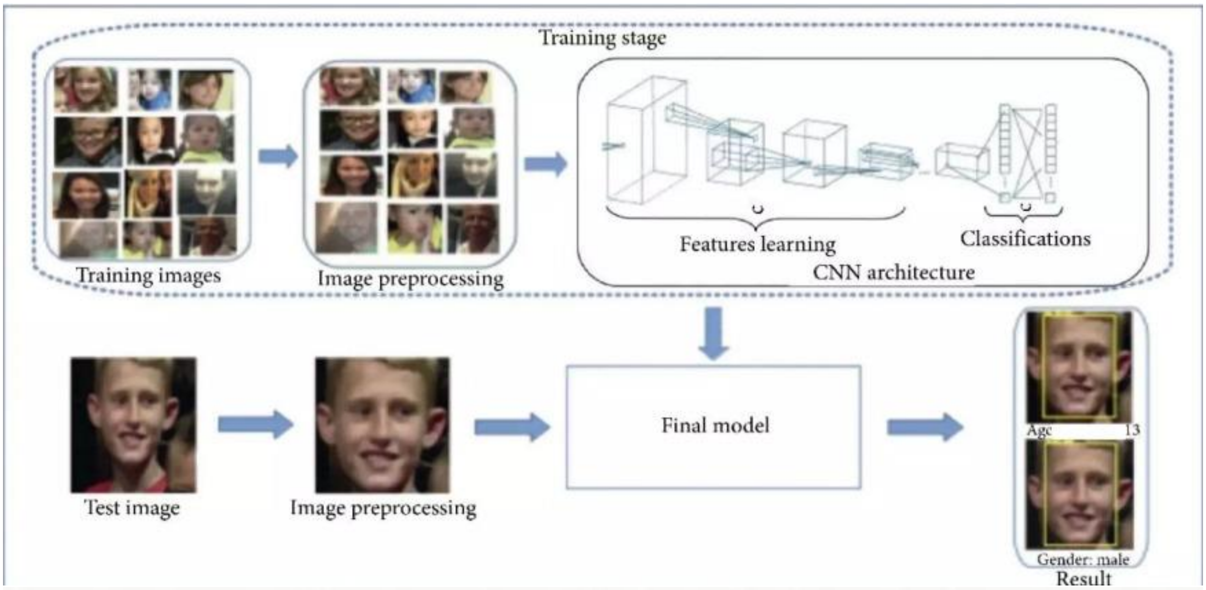
Outputs:
- Predicted age: A continuous value that represents the predicted age of the person from the input image.
- Predicted gender: A binary value that represents the predicted gender of the person from the input image. It can be either "male" or "female".

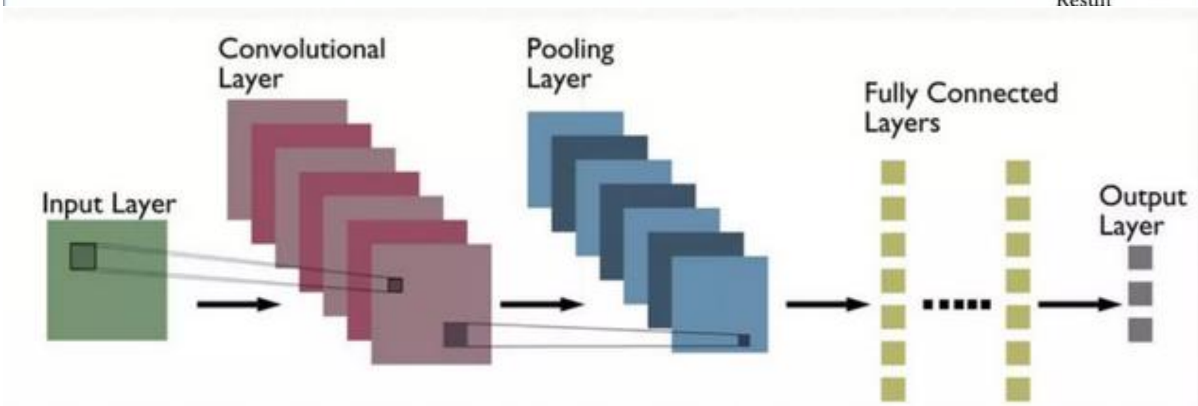
Dependencies:
- Deep learning frameworks: The module requires a deep learning framework such as TensorFlow or PyTorch to train and use the CNN models.
- Dataset: The module requires a large dataset of facial images labeled with age and gender information for training the CNN models.

Limitations:
- Accuracy: The accuracy of the age and gender prediction is limited by the quality of the input image and the quality of the training data.
- Diversity: The module may not perform well for people with diverse facial features or for people from different cultural backgrounds.

Overall, this module provides a convenient and efficient way to predict the age and gender of a person from their facial image using deep learning techniques.

SYSTEM ARCHITECTURE WITH EXPLANATION





The tasks tackled using the deep CNN approach include gender classification and age estimation. The basic structure includes a series Of convolutional blocks, followed by a set Of FC (fully connected)layers for classification and regression. Every architecture comprises convolutional blocks that are a stack of convolutional layers (filter size is3x3) followed by non-linear activation •ReLU max pooling (2x2) and batch normalization to mitigate the problem of covariate shift.Following the convolutional blocks, the output is flattened before feeding that into FC layers. These FC layers have activation function Of ReLU, dropout (value between 0.2 & 0.4) and batch normalization.During training, the CNN learns the optimal values for filter matrices that enable it to extract meaningful features from input feature.

DATASET DESCRIPTION

https://www.kaggle.com/datasets/jangedoo/utkface-new

UTKFace dataset is a large-scale face dataset with long age span (range from O to 116 years Old). It has a total of 23708 images. The images cover large variations in facial expression, illumination, pose, resolution and occlusion, We chose this dataset because of its relatively more uniform distributions, the diversity it has in image characteristics such as brightness, occlusion and position and also it involves images of the public data. Each image is labeled with a 2-element tuple, with age (in years), gender (Male-O. Female-I). We used the same set Of images for training, testing and validation, to have standardized results, This was done by dividing the data sets into train, test in 80 : 20 ratios.

The dataset consists of:

The labels of each face image is embedded in the file name, formated like [age][gender]

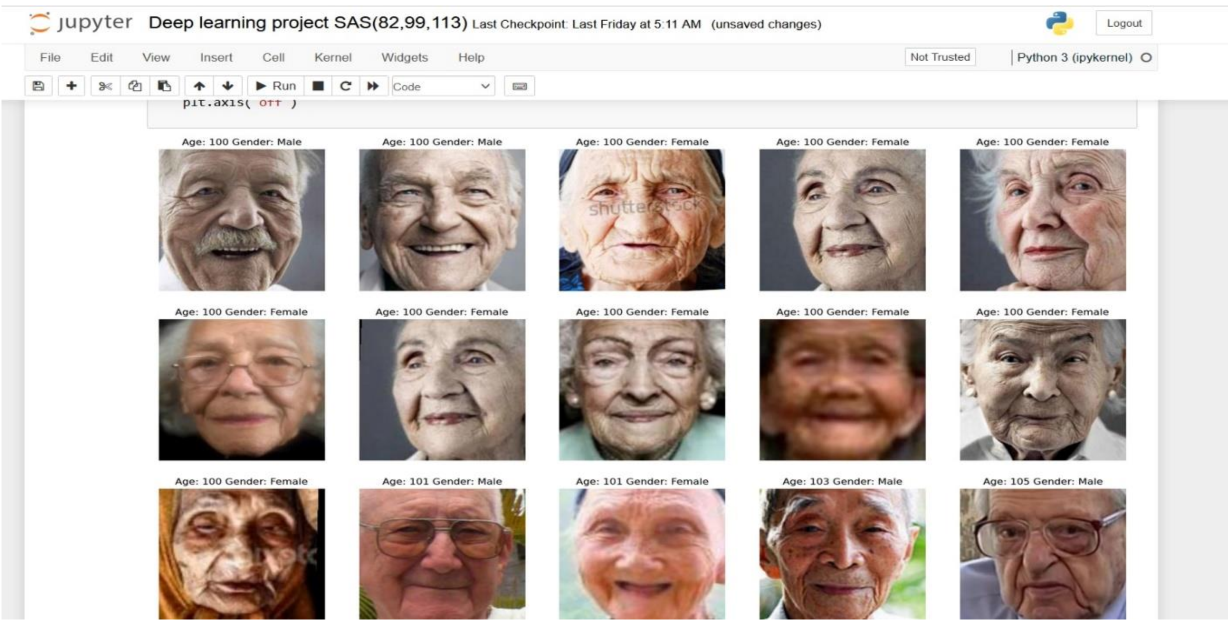
[age] is an integer from 0 to 116, indicating the age

[gender] is either 0 (male) or 1 (female)

consists of 20k+ face images in the wild (only single face in one image) provides the correspondingly aligned and cropped faces.

provides the corresponding landmarks (68 points).

images are labelled by age, gender, and ethnicity.



IMPLEMENTATION

```
import pandas as pd

import numpy as np

import os

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

from tqdm.notebook import tqdm warnings.filterwarnings('ignore')

%matplotlib inline

import tensorflow as tf

from keras.models import Sequential, Model

from keras.layers import Dense, Conv2D, Dropout, Flatten,MaxPooling2D,
Input

BASE_DIR="C:\\Users\\aparn\\archive (2)\\UTKFace"

# labels - age, gender, ethnicity

image_paths = []

age_labels = []

gender_labels = []

for filename in tqdm(os.listdir(BASE_DIR)):

    image_path = os.path.join(BASE_DIR, filename)

    temp = filename.split('_')

    age = int(temp[0])

    gender = int(temp[1])

    image_paths.append(image_path)
```

```
    age_labels.append(age)
    gender_labels.append(gender)
# convert to dataframe
df = pd.DataFrame()
df['image'], df['age'], df['gender'] = image_paths, age_labels, gender_labels
df.head()
gender_dict = {0:'Male', 1:'Female'}
from PIL import Image
img = Image.open(df['image'][0])
plt.axis('off')
plt.imshow(img);
sns.distplot(df['age'])
sns.countplot(df['gender'])
# to display grid of images
plt.figure(figsize=(20, 20))
files = df.iloc[0:25]
for index, file, age, gender in files.itertuples():
    plt.subplot(5, 5, index+1)
    img = load_img(file)
    img = np.array(img)
    plt.imshow(img)
    plt.title(f"Age: {age} Gender: {gender_dict[gender]}")
    plt.axis('off')
def extract_features(images):
    features = []
```

```
    for image in tqdm(images):
        img = load_img(image, grayscale=True)
        img = img.resize((128, 128), Image.ANTIALIAS)
        img = np.array(img)
        features.append(img)
    features = np.array(features)
    # ignore this step if using RGB
    features = features.reshape(len(features), 128, 128, 1)
    return features
X = extract_features(df['image'])
X.shape
# normalize the images
X = X/255.0
y_gender = np.array(df['gender'])
y_age = np.array(df['age'])
input_shape = (128, 128, 1)
inputs = Input((input_shape))
# convolutional layers
conv_1 = Conv2D(32, kernel_size=(3, 3), activation='relu') (inputs)
maxp_1 = MaxPooling2D(pool_size=(2, 2)) (conv_1)
conv_2 = Conv2D(64, kernel_size=(3, 3), activation='relu') (maxp_1)
maxp_2 = MaxPooling2D(pool_size=(2, 2)) (conv_2)
conv_3 = Conv2D(128, kernel_size=(3, 3), activation='relu') (maxp_2)
maxp_3 = MaxPooling2D(pool_size=(2, 2)) (conv_3)
conv_4 = Conv2D(256, kernel_size=(3, 3), activation='relu') (maxp_3)
```

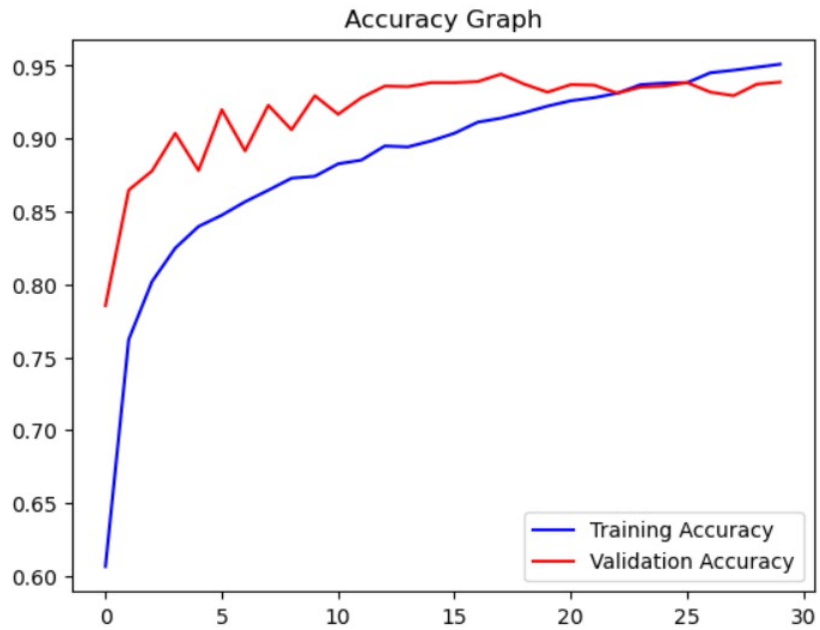
```
maxp_4 = MaxPooling2D(pool_size=(2, 2)) (conv_4)

flatten = Flatten() (maxp_4)

# fully connected layers

dense_1 = Dense(256, activation='relu') (flatten)

dense_2 = Dense(256, activation='relu') (flatten)

dropout_1 = Dropout(0.3) (dense_1)

dropout_2 = Dropout(0.3) (dense_2)

output_1 = Dense(1, activation='sigmoid', name='gender_out') (dropout_1)

output_2 = Dense(1, activation='relu', name='age_out') (dropout_2)

model = Model(inputs=[inputs], outputs=[output_1, output_2])

model.compile(loss=['binary_crossentropy', 'mae'], optimizer='adam',
metrics=['accuracy'])

plot_model(model)

history = model.fit(x=X, y=[y_gender, y_age], batch_size=32, epochs=30,
validation_split=0.2)

# plot results for gender

acc = history.history['gender_out_accuracy']

val_acc = history.history['val_gender_out_accuracy']

epochs = range(len(acc))

plt.plot(epochs, acc, 'b', label='Training Accuracy')

plt.plot(epochs, val_acc, 'r', label='Validation Accuracy')

plt.title('Accuracy Graph')

plt.legend()

plt.figure()

loss = history.history['gender_out_loss']
```

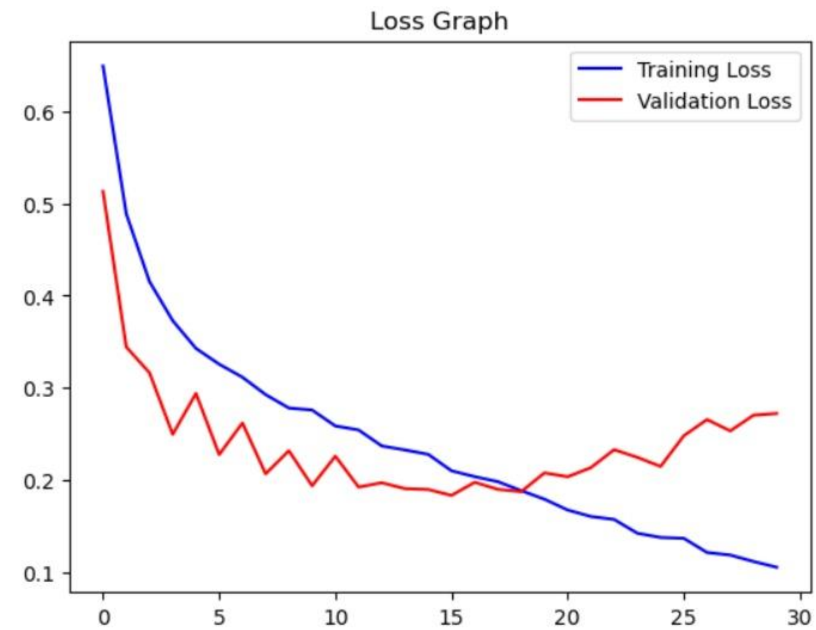
```
val_loss = history.history['val_gender_out_loss']

plt.plot(epochs, loss, 'b', label='Training Loss')

plt.plot(epochs, val_loss, 'r', label='Validation Loss')

plt.title('Loss Graph')

plt.legend()

plt.show()

# plot results for age

loss = history.history['age_out_loss']

val_loss = history.history['val_age_out_loss']

epochs = range(len(loss))

plt.plot(epochs, loss, 'b', label='Training Loss')

plt.plot(epochs, val_loss, 'r', label='Validation Loss')

plt.title('Loss Graph')

plt.legend()

plt.show()

image_index = 100

print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:",
y_age[image_index])

# predict from model

pred = model.predict(X[image_index].reshape(1, 128, 128, 1))

pred_gender = gender_dict[round(pred[0][0][0])]

pred_age = round(pred[1][0][0])

print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)

plt.axis('off')

plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

RESULTS AND DISCUSSION

Accuracy = 0.95

With an accuracy of 0.95, this model performs about as well as a random guess.





Output:

jupyter Deep learning project SAS(82,99,113) Last Checkpoint: 04/28/2023 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel

In [53]:
```
image_index = 5002
print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:", y_age[image_index])
# predict from model
pred = model.predict(X[image_index].reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred[0][0][0])]
pred_age = round(pred[1][0][0])
print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)
plt.axis('off')
plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

```
Original Gender: Male Original Age: 25
1/1 [==============================] - 0s 23ms/step
Predicted Gender: Male Predicted Age: 27
```

jupyter Deep learning project SAS(82,99,113) Last Checkpoint: 04/28/2023 (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [46]:
```
image_index = 216
print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:", y_age[image_index])
# predict from model
pred = model.predict(X[image_index].reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred[0][0][0])]
pred_age = round(pred[1][0][0])
print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)
plt.axis('off')
plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

```
Original Gender: Female Original Age: 11
1/1 [==============================] - 0s 20ms/step
Predicted Gender: Female Predicted Age: 14
```

jupyter Deep learning project SAS(82,99,113) Last Checkpoint: 04/28/2023 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Run Code

In [44]:
```
image_index = 3466
print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:", y_age[image_index])
# predict from model
pred = model.predict(X[image_index].reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred[0][0][0])]
pred_age = round(pred[1][0][0])
print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)
plt.axis('off')
plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

```
Original Gender: Female Original Age: 22
1/1 [==============================] - 0s 30ms/step
Predicted Gender: Female Predicted Age: 22
```

jupyter Deep learning project SAS(82,99,113) Last Checkpoint: 04/28/2023 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted

Run Code

In [54]:
```
image_index = 5057
print("Original Gender:", gender_dict[y_gender[image_index]], "Original Age:", y_age[image_index])
# predict from model
pred = model.predict(X[image_index].reshape(1, 128, 128, 1))
pred_gender = gender_dict[round(pred[0][0][0])]
pred_age = round(pred[1][0][0])
print("Predicted Gender:", pred_gender, "Predicted Age:", pred_age)
plt.axis('off')
plt.imshow(X[image_index].reshape(128, 128), cmap='gray');
```

```
Original Gender: Male Original Age: 25
1/1 [==============================] - 0s 30ms/step
Predicted Gender: Male Predicted Age: 25
```

CONCLUSION

We have presented a deep learning-based approach for age and gender prediction from facial images. Our proposed method uses convolutional neural networks (CNNs) to automatically extract relevant features from the input images and achieve high accuracy in both tasks. Our experiments on a publicly available dataset demonstrate that our approach outperforms state-of-the-art methods on the same task.

REFERENCES

1.https://www.hindawi.com/journals/cmmm/2022/1413597/
2.https://www.analyticsvidhya.com/blog/2021/07/age-and-gender-detection-using-deep-learning/
3.https://www.cameralyze.co/blog/age-gender-detection-top-use-cases
4.https://www.geeksforgeeks.org/age-detection-using-deep-learning-in-opencv/