A MINI-PROJECT REPORT ON

“Gender and Age Detection: predict if a person is a male or female and also their age”

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY IN THE PARTIAL FULFILLMENT OF THE REQUIRED AWARD OF THE DEGREE OF

BACHELOR OF COMPUTER ENGINEERING

SUBMITTED BY

Pradnya Gokhale - 41216
Nidhi Tikone - 41265
Rushabh Udani – 41268
Charvi Wankhede - 41271

Academic Year:
2024-2025

UNDER THE GUIDANCE OF

Mrs. Y. P. Narwadkar

Department of Computer Engineering
P.E.S Modern College of Engineering, Pune - 05

Department of Computer Engineering
PES's Modern College of Engineering,
Pune-05 2024-2025

CERTIFICATE

This is to certify that the project entitled

" Gender and Age Detection"

Submitted by

Pradnya Gokhale	[41216]
Nidhi Tikone	[41265]
Rushabh Udani	[41268]
Charvi Wankhede	[41271]

Are bonafide students of this institute and the work has been carried out by them under the supervision of **Mrs.Y.P.Narwadkar** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Mrs. Y.P. Narwadkar
Project Guide/ Internal Examiner
Computer Engineering

Prof. Dr. Mrs. S. A. Itkar
Head of Department
Computer Engineering

Place: Pune

Date:

CONTENTS

Abstract

Gender and age detection is a computer vision challenge that involves predicting an individual's gender and approximate age based on photos or videos. It is a difficult subject because of variances in facial characteristics, expressions, lighting circumstances, and image quality. This activity has applications in a variety of sectors, including demographic analysis, targeted advertising, and personalized user experiences. Accurate gender and age detection models necessitate strong machine learning algorithms, large datasets, and sophisticated image processing techniques.

Gender and age detection is a computer vision challenge that involves predicting an individual's gender and approximate age based on photos or videos. It is a difficult subject because of variances in facial characteristics, expressions, lighting circumstances, and image quality. Accurate gender and age detection models necessitate strong machine learning algorithms, large datasets, and sophisticated image processing techniques.

Keywords: Gender detection,Age estimation,Computer vision,Machine learning,Facial analysis, Deep learning,Convolutional neural networks (CNNs),Labeled datasets,Demographic inference, Targeted advertising,Pattern recognition,Image processing,Real-time applications,Transfer learning,Privacy-preserving algorithms

1. Introduction

1. History

Over the last decade, computer vision and machine learning researchers have paid close attention to gender and age detection. Early attempts used typical computer vision techniques including feature extraction and classification algorithms. However, the advent of deep learning transformed this process by allowing the creation of more accurate and resilient models.

Gender and age detection dates back to the mid-2000s, when academics began experimenting with automated facial analysis approaches. Initial efforts concentrated on gender classification using face traits such as the jawline, brows, and hair. These approaches frequently depended on handcrafted features and unsophisticated classifiers, which led to limited performance.

In the early 2010s, the emergence of convolutional neural networks (CNNs) paved the path for substantial advances in gender and age detection. Deep learning architectures enabled models to automatically learn hierarchical representations from raw image data, resulting in higher performance. Researchers began using large-scale datasets of annotated facial pictures, such as the Adience and IMDB-WIKI databases, to train deep learning models for gender and age estimate tasks.

.

2. Basic Theory

Computer vision and machine learning techniques are commonly used to predict gender and age by analyzing facial photos or video frames. The core premise behind this challenge is the extraction of discriminative elements from facial data that may be used to determine an individual's gender and age.

Feature Extraction: Facial photos include details like the form of the eyes, nose, and mouth, as well as skin texture and color. These features are extracted from images using techniques such as convolutional neural networks (CNNs) or manually created feature descriptors.

Model Training: Once the features have been extracted, machine learning models are trained on labeled datasets to discover how these features relate to the target variables (gender and age). Deep learning models, notably CNNs, have shown greater performance in extracting complicated patterns from raw image data.

Prediction: During the prediction phase, the trained model accepts a new facial image as input and returns the projected gender and age. This prediction is based on previously learnt patterns and correlations between the retrieved characteristics and target variables.

2. Problem Statement/ Objective

2.1 Problem Statement

Gender and Age Detection: predict if a person is a male or female and also their age.

2.2 Objective

Accurate Gender Prediction: The primary objective is to develop models that can accurately predict the gender of individuals from facial images or video frames. This involves training models on diverse datasets to learn gender-specific facial features and patterns.

Precise Age Estimation: Another key objective is to estimate the age of individuals with high precision. Age estimation is often more challenging than gender prediction due to factors such as facial aging, variations in appearance, and the subjective nature of age perception.

Robustness to Variations: The models should be robust to variations in facial appearance such as lighting conditions, facial expressions, poses, and occlusions. Robustness ensures that the models can generalize well to unseen data and perform reliably in real-world scenarios.

Efficiency and Scalability: It's important to develop models that are computationally efficient and scalable, especially for real-time applications and large-scale deployments. This involves optimizing model architectures, reducing computational complexity, and exploring techniques like model compression and quantization.

2.3 Need

Defect Detection: One of the primary needs of software testing is to identify defects, bugs, and errors in the software. This ensures that issues are caught early in the development process and can be fixed before they reach end-users.

Quality Assurance: Testing is crucial for ensuring the overall quality of the software. It helps in verifying that the software meets the specified requirements and standards, and it functions correctly.

Risk Mitigation: Testing helps in identifying and mitigating risks associated with the software. By testing different scenarios, potential problems can be anticipated and addressed.

User Satisfaction: Ensuring that the software works as expected is vital for user satisfaction. Quality assurance efforts aim to provide a positive user experience by minimizing the occurrence of defects.

3. Methodology

The methodology for gender and age detection involves several steps, including data collection, preprocessing, model training, evaluation, and deployment. Below is a detailed explanation of each step:

Data Collection:Gather a diverse dataset of facial images or video frames labeled with gender and age information. The dataset should cover a wide range of demographics, ethnicities, and ages to ensure model generalization.Popular datasets for gender and age detection include IMDB-WIKI, CelebA, Adience, and UTKFace.

- Data Preprocessing: Perform preprocessing steps to standardize and enhance the quality of the data. This may include:
- Face detection: Use face detection algorithms to detect and extract faces from images or video frames.
- Face alignment: Align facial landmarks to a canonical pose to reduce variations in facial expression and pose.
- Image normalization: Normalize pixel values to a standard range (e.g., [0, 1]) and resize images to a fixed resolution.
- Data augmentation: Augment the dataset by applying transformations such as rotation, translation, scaling, and flipping to increase the diversity of training samples

Feature Extraction:Extract features from preprocessed facial images using convolutional neural networks (CNNs) or handcrafted feature descriptors.CNN-based approaches: Utilize pre-trained CNN architectures such as VGG, ResNet, or MobileNet to extract deep features from facial images. Alternatively, fine-tune these architectures on the target dataset if sufficient labeled data is available.

Handcrafted feature descriptors: Extract handcrafted features such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or Eigenfaces.

Model Training:Train machine learning models on the extracted features to predict gender and age labels.

Deep learning models: Train CNN-based architectures (e.g., convolutional neural networks) using labeled data in a supervised learning setting. Use loss functions such as cross-entropy loss for gender classification and mean absolute error or mean squared error for age regression.

Traditional machine learning models: Train classifiers or regressors (e.g., Support Vector Machines, Random Forests, Linear Regression) using handcrafted features or deep features as input.

Model Evaluation: Evaluate the trained models on a separate validation set or through cross-validation to assess their performance.Metrics for evaluation may include accuracy, precision, recall, F1-score, mean absolute error (MAE), and mean squared error (MSE) for gender classification and age estimation tasks, respectively.

Model Optimization: Fine-tune model hyperparameters (e.g., learning rate, batch size, regularization strength) using techniques such as grid search or random search to optimize performance.Apply techniques like early stopping to prevent overfitting and regularization to improve generalization.

Deployment: Deploy the trained model in real-world applications, considering factors such as computational efficiency, memory footprint, and latency.Implement the model in a production environment using frameworks like TensorFlow Serving, ONNX Runtime, or deploying as a REST API.Continuously monitor model performance and update the model as necessary to adapt to changing data distributions and requirements.

By following this methodology, researchers and practitioners can develop accurate, efficient, and reliable gender and age detection systems capable of addressing various application needs across different domains.

4. Flow Diagram

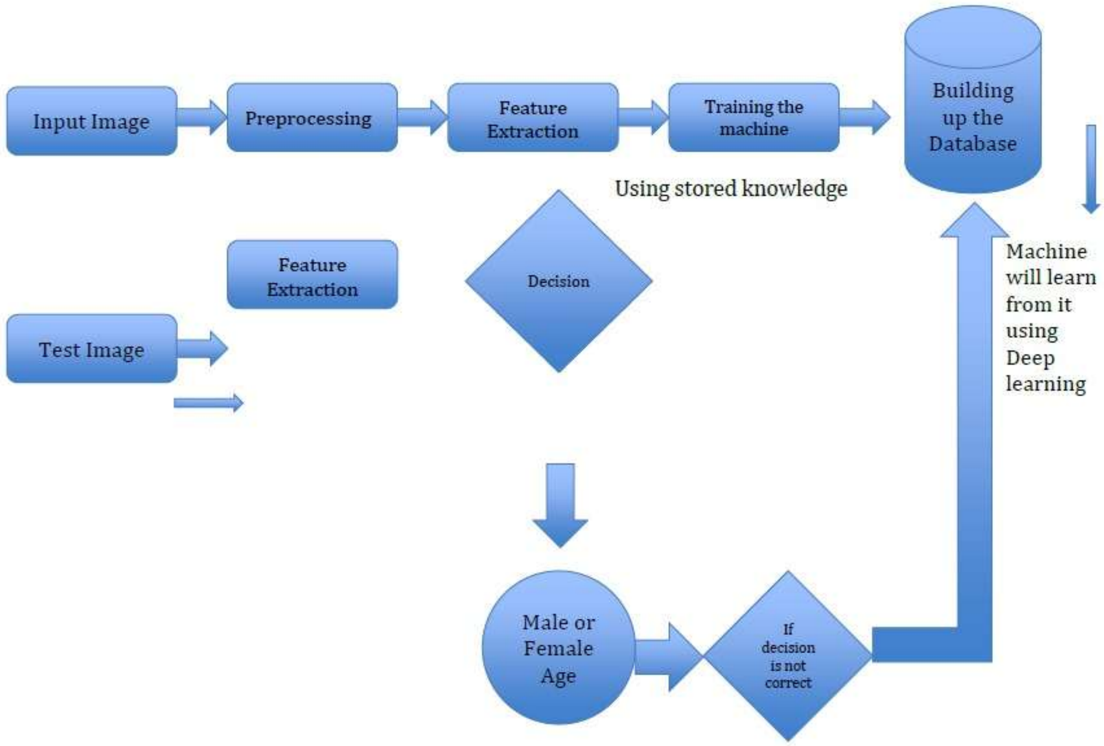


Figure 3: Flowchart for gender classification [7].

5. Input Details

[12]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
```

+ Code + Markdown

[13]:

```python
# Load the CSV
df = pd.read_csv("/kaggle/input/age-gender-csv/age_gender.csv")
df.head()
```

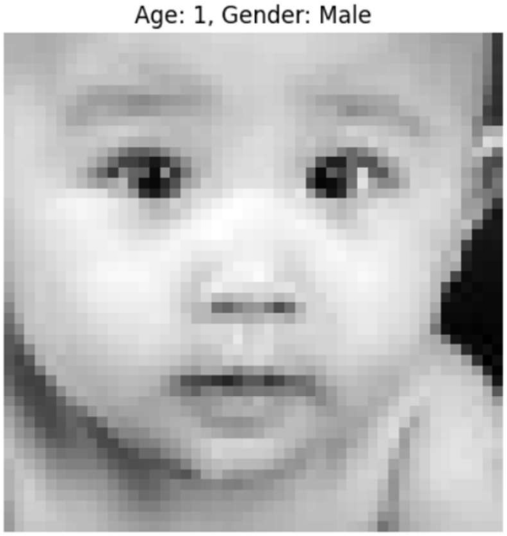
[13]:

	age	ethnicity	gender	img_name	pixels
0	1	2	0	20161219203650636.jpg.chip.jpg	129 128 128 126 127 130 133 135 139 142 145 14...
1	1	2	0	20161219222752047.jpg.chip.jpg	164 74 111 168 169 171 175 182 184 188 193 199...
2	1	2	0	20161219222832191.jpg.chip.jpg	67 70 71 70 69 67 70 79 90 103 116 132 145 155...
3	1	2	0	20161220144911423.jpg.chip.jpg	193 197 198 200 199 200 202 203 204 205 208 21...
4	1	2	0	20161220144914327.jpg.chip.jpg	202 205 209 210 209 209 210 211 212 214 218 21...

[14]:

```python
# View one sample as image
pixels = np.array([int(p) for p in df['pixels'][0].split()], dtype=np.uint8).reshape(48, 48)
plt.imshow(pixels, cmap='gray')
plt.title(f"Age: {df['age'][0]}, Gender: {'Male' if df['gender'][0]==0 else 'Female'}")
plt.axis('off')
plt.show()
```

Age: 1, Gender: Male



[15]:

```python
# Convert pixels to image array
def process_pixels(pixel_str):
    pixels = np.array([int(p) for p in pixel_str.split()], dtype=np.uint8)
    return pixels.reshape(48, 48, 1)

X = np.array([process_pixels(pix) for pix in df['pixels']])
X = X / 255.0  # Normalize

# Prepare labels
y_age = df['age'].values
y_gender = to_categorical(df['gender'].values, 2)
```

[16]:

```python
X_train, X_test, y_age_train, y_age_test, y_gender_train, y_gender_test = train_test_split(
    X, y_age, y_gender, test_size=0.2, random_state=42
)
```

[17]:

```python
input_layer = Input(shape=(48, 48, 1))

x = Conv2D(32, (3, 3), activation='relu')(input_layer)
x = MaxPooling2D()(x)
x = Conv2D(64, (3, 3), activation='relu')(x)
x = MaxPooling2D()(x)
x = Flatten()(x)
x = Dropout(0.5)(x)

# Output for Age (regression)
age_output = Dense(1, name='age_output')(x)

# Output for Gender (classification)
gender_output = Dense(2, activation='softmax', name='gender_output')(x)

model = Model(inputs=input_layer, outputs=[age_output, gender_output])
```

[18]:

```python
model.compile(
    optimizer='adam',
    loss={'age_output': 'mse', 'gender_output': 'categorical_crossentropy'},
    metrics={'age_output': 'mae', 'gender_output': 'accuracy'}
)
```

[19]:
```
history = model.fit(
    X_train,
    {'age_output': y_age_train, 'gender_output': y_gender_train},
    validation_data=(X_test, {'age_output': y_age_test, 'gender_output': y_gender_test}),
    epochs=10,
    batch_size=64
)
model.save('age_gender_model.h5')
```

```
Epoch 1/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 27s 84ms/step - age_output_loss: 624.9348 - age_output_mae: 19.1542 - gender_output_accuracy: 0.57
77 - gender_output_loss: 0.7101 - loss: 625.6465 - val_age_output_loss: 331.4856 - val_age_output_mae: 13.8869 - val_gender_out
put_accuracy: 0.7030 - val_gender_output_loss: 0.5948 - val_loss: 331.3983
Epoch 2/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 24s 82ms/step - age_output_loss: 308.9274 - age_output_mae: 13.5060 - gender_output_accuracy: 0.70
21 - gender_output_loss: 0.6253 - loss: 309.5535 - val_age_output_loss: 241.9082 - val_age_output_mae: 12.1753 - val_gender_out
put_accuracy: 0.7583 - val_gender_output_loss: 0.5403 - val_loss: 242.2881
Epoch 3/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 24s 82ms/step - age_output_loss: 263.2468 - age_output_mae: 12.5758 - gender_output_accuracy: 0.74
35 - gender_output_loss: 0.5639 - loss: 263.8095 - val_age_output_loss: 223.2188 - val_age_output_mae: 11.5921 - val_gender_out
put_accuracy: 0.7927 - val_gender_output_loss: 0.4446 - val_loss: 223.6790
Epoch 4/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 23s 77ms/step - age_output_loss: 249.6264 - age_output_mae: 12.1919 - gender_output_accuracy: 0.75
43 - gender_output_loss: 0.5419 - loss: 250.1677 - val_age_output_loss: 231.5335 - val_age_output_mae: 11.4003 - val_gender_out
put_accuracy: 0.8106 - val_gender_output_loss: 0.4138 - val_loss: 232.3844
Epoch 5/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 24s 79ms/step - age_output_loss: 245.7486 - age_output_mae: 12.1448 - gender_output_accuracy: 0.76
08 - gender_output_loss: 0.5185 - loss: 246.2681 - val_age_output_loss: 216.6930 - val_age_output_mae: 11.1107 - val_gender_out
put_accuracy: 0.7207 - val_gender_output_loss: 0.5933 - val_loss: 217.5236
Epoch 6/10
297/297 ━━━━━━━━━━━━━━━━━━━━ 24s 79ms/step - age_output_loss: 237.2641 - age_output_mae: 11.8825 - gender_output_accuracy: 0.74
62 - gender_output_loss: 0.5750 - loss: 237.8381 - val_age_output_loss: 214.6123 - val_age_output_mae: 11.5360 - val_gender_out
```

]:
```
results = model.evaluate(
    X_test,
    {'age_output': y_age_test, 'gender_output': y_gender_test}
)
print(f"Test Loss: {results[0]}")
print(f"Age MAE: {results[3]}")
print(f"Gender Accuracy: {results[4]}")
```

```
149/149 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - age_output_loss: 190.4501 - age_output_mae: 10.3822 - gender_output_accuracy: 0.747
9 - gender_output_loss: 0.5601 - loss: 191.0154
Test Loss: 193.9940643310547
Age MAE: 10.50826358795166
Gender Accuracy: 0.7336004972457886
```

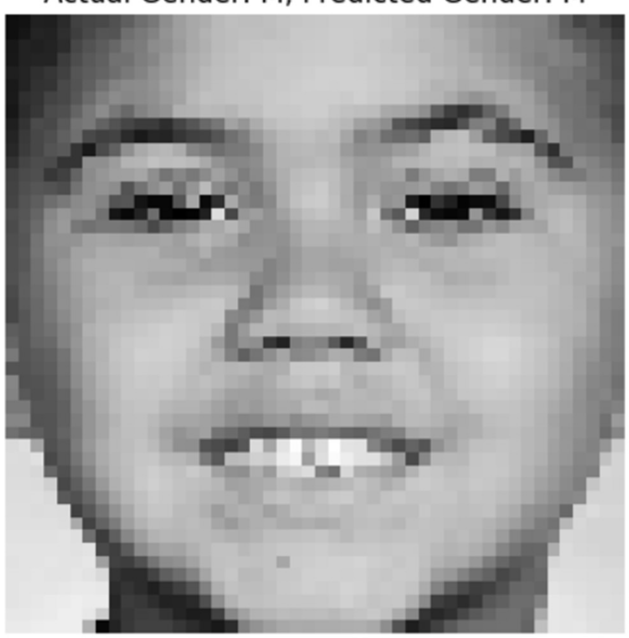
]:
```
pred_age, pred_gender = model.predict(X_test[:5])
for i in range(5):
    plt.imshow(X_test[i].reshape(48, 48), cmap='gray')
    plt.title(f"Actual Age: {y_age_test[i]}, Predicted Age: {int(pred_age[i])}\n"
              f"Actual Gender: {'M' if np.argmax(y_gender_test[i])==0 else 'F'}, "
              f"Predicted Gender: {'M' if np.argmax(pred_gender[i])==0 else 'F'}")
    plt.axis('off')
    plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step
```

6. Results

Actual Age: 27, Predicted Age: 28
Actual Gender: M, Predicted Gender: M

Actual Age: 8, Predicted Age: 9
Actual Gender: M, Predicted Gender: M



7. Conclusion

In conclusion, gender and age detection are critical jobs in computer vision and machine learning, with countless real-world applications. This study examined the approaches, challenges, and achievements in the field, with the goal of properly and efficiently inferring demographic information from facial photographs or video frames.

Achievements and Progress:In recent years, significant progress has been made due to advances in deep learning, notably convolutional neural networks (CNNs). These models have shown greater ability in learning discriminative features from raw pixel data, resulting in significant increases in gender and age detection accuracy.
Large-scale labeled datasets, along with powerful computational resources, have made it easier to construct and train more complicated models capable of capturing detailed patterns in facial photos.

Challenges and Opportunities: Despite advancements, there are still significant challenges in gender and age detection, including as robustness to variations in facial appearance, dataset bias, and privacy concerns. Addressing these difficulties will necessitate interdisciplinary collaboration in computer vision, machine learning, psychology, and ethics.
Furthermore, there are potential for future study and innovation, such as experimenting with multimodal techniques that combine facial photos with other modalities like voice or text to improve demographic inference. Furthermore, research into explainable AI strategies can improve model transparency and interpretability, increasing trust and comprehension of model predictions.

Applications and Implications: Gender and age detection have a wide range of applications, including marketing, healthcare, security, entertainment, and human-computer interaction.
These applications include targeted advertising and personalized content recommendations, as well as age estimation for age-restricted content and healthcare services.
However, it is critical to evaluate the ethical consequences of using gender and age detection systems, particularly in terms of privacy, fairness, and bias. Responsible AI approaches, such as fairness-aware algorithms, bias mitigation strategies, and transparent decision-making processes, are critical to ensuring equitable outcomes and minimizing potential downsides.

8. References

1) Gerald M. Weinberg. The Psychology of Computer Programming: Silver Anniversary Edition. Dorset House; 2nd edition (September 1998).
2) Robert N. Britcher. The Limits of Software: People, Projects, and Perspectives. Addison-Wesley Pub Co; 1st edition (June 25, 1999).

3) Hans Sassenburg. Software Engineering van ambacht naar professie. Den Bosch: Uitgeverij Tutein Nolthenius, 2002.

4) Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Addison-Wesley Pub Co; 1st edition (August 2, 1995).

5) Tom Demarco, Timothy Lister. Peopleware : Productive Projects and Teams. Dorset House; 2nd edition (February 1, 1999).

6) Tells about the failure of the Advanced Automation System project, whose goal was to develop a new National Air Traffic Control System.

7) Hans Sassenburg. Software Engineering van ambacht naar professie. Den Bosch: Uitgeverij Tutein Nolthenius, 2002.

8) 1st Edition: 1971. Describes computer programming as human performance, a social activity, and an individual activity.