

## SAVITRIBAI PHULE PUNE UNIVERSITY

The Mini Project Based On

## Gender and age Detection

### Submitted By:

Diwadkar Siddhi Sachin

### Under Guidance of:

Prof. J. A. Kalbhor In partial fulfillment of

Laboratory Practice-V (410255)



### DEPARTMENT OF COMPUTER ENGINEERING SAVITRIBAI PHULE PUNE UNIVERSITY 2024-25

CERTIFICATE

This is to certify that the Mini Project based on,

## Gender and age Detection

has been successfully completed by,

Name: Diwadkar Siddhi Sachin

Exam seat number:

Towards the partial fulfilment of the Final Year of Computer Engineering as awarded by the Savitribai Phule Pune University, at PDEA’s College of Engineering, Manjari Bk,” Hadapsar, Pune 412307, during the academic year 2024-25.

|  |  |
| --- | --- |
| Prof. J. A. Kalbhor | Dr. M. P. Borawake |
| **Guide Name** | **H. O. D** |

# Acknowledgement

My first and for most acknowledgment is to my supervisor and guide Prof. J. A. Kalbhor. During the long journey of this study, she supported me in every aspect. She was the one who helped and motivated me to proposer search in this field and inspired me with her enthusiasm on research, her experience, and her lively character.

I express true sense of gratitude to my guide Prof. J. A. Kalbhor. for her perfect valuable guidance, all the time support and encouragement that he gave me.

I would also like to thanks our head of department Dr. M. P. Borawake and Principal Dr. R. V. Patil and management inspiring me and providing all lab and other facilities, which made this mini project very convenient.

I thankful to all those who rendered their valuable help for successful completion on Internship presentation.

Name: Diwadkar Siddhi Sachin

# INDEX

|  |  |  |
| --- | --- | --- |
| Sr No. | Contents | Page No. |
| 1. | Abstract | 1 |
| 2. | Introduction | 2 |
| 3. | Objectives | 3 |
| 4. | System Specification | 4 |
| 5. | Methodology | 5 |
| 6. | Future Scope | 7 |
| 7. | Conclusion | 8 |
| 8. | Reference | 9 |

**Abstract**

This project presents the design and implementation of a real-time gender and age detection system using deep learning techniques and OpenCV. The system is capable of analyzing a facial image or video stream to predict the gender and approximate age group of a person. Gender and age prediction are integral parts of modern computer vision applications such as security systems, audience analytics, targeted advertising, and human-computer interaction. The implementation leverages pre-trained deep learning models provided by Caffe, a deep learning framework, integrated with OpenCV's DNN module. Using a webcam or a static image, the program detects faces, extracts facial features, and feeds the features into pre-trained convolutional neural networks to classify gender and estimate age. The system is efficient, user-friendly, and performs predictions with a high degree of accuracy. The solution demonstrates the power of transfer learning and classical computer vision tools in building real-time, practical AI applications.

# Introduction

In the modern era, the ability of machines to understand human features and emotions is becoming increasingly essential. Among various fields in computer vision, gender and age detection has gained significant attention due to its wide range of applications. For instance, in digital advertising, gender and age estimation can help tailor content to specific demographics. In security and surveillance, it aids in identifying individuals and profiling threats. This project aims to provide a real-time application that can recognize the gender and approximate age of a person from a single image or video input. By using deep learning and pre-trained models, the system reduces the computational overhead typically involved in training large neural networks from scratch. Instead, it utilizes trained models to produce fast and reliable results.

The development process uses OpenCV's deep learning module, which allows the integration of models from Caffe. Caffe models for age and gender recognition have been trained on large datasets, and their integration with OpenCV simplifies the implementation of computer vision tasks. This project is also an excellent introduction for students and researchers who want to explore practical deep learning applications using Python.

# Objectives

* To develop a real-time gender and age detection system using deep learning and OpenCV.
* To use pre-trained models to classify gender and estimate age from facial features.
* To implement an efficient and user-friendly interface that works with image and video input.
* To evaluate the model's accuracy and performance in different lighting and background conditions.
* To provide a base that can be extended for more complex demographic analysis or human feature recognition tasks.

# System Specification

**Hardware Requirements:**

**Processor:** Intel Core i5 or higher

**RAM:** Minimum 8GB

**GPU:** Optional, but recommended for faster performance (NVIDIA CUDA-enabled GPU)

**Camera:** Integrated or external webcam

**Storage:** Minimum 10GB of free disk space

**Software Requirements:**

**Operating System:** Windows, macOS, or Linux

**Programming Language:** Python 3.6 or higher

**Libraries:**

* OpenCV (cv2)
* argparse
* math

**Deep Learning Models:**

* **Gender:** gender\_net.caffemodel and gender\_deploy.prototxt
* **Age:** age\_net.caffemodel and age\_deploy.prototxt
* **Face Detection:** opencv\_face\_detector.pb and opencv\_face\_detector.pbtxt

**Data:**

The models are pre-trained on standard datasets, often including the Adience dataset which contains thousands of images labeled with gender and age group.

# Methodology

The methodology of the gender and age detection project is grounded in real-time image processing, feature extraction using deep learning, and robust inference from pre-trained models. This section elaborates on the detailed working of the code, the logical flow of operations, image handling, deep learning model integration, and output rendering.

**1. Code Initialization and Argument Parsing:**

The Python script begins by importing the required libraries, including OpenCV for computer vision tasks, math for basic mathematical operations, and argparse for handling command-line arguments. The argument parser allows the user to optionally pass an image path using --image. If no image is provided, the system uses the webcam by default for real-time detection.

**2. Loading Pre-trained Models:**

The system uses three pre-trained models:

* A face detector model using TensorFlow or Caffe, loaded using .pb and .pbtxt files.
* A gender classification model, defined by gender\_net.caffemodel and gender\_deploy.prototxt.
* An age classification model using age\_net.caffemodel and age\_deploy.prototxt.

These models are loaded using OpenCV’s cv2.dnn.readNet() method, which prepares them for inference. The models are trained on large datasets and can generalize well across diverse facial inputs.

**3. Face Detection Logic:**

A critical function, highlightFace(), is defined to detect faces from the input frame. It works by:

* Creating a blob from the image with a specified mean value normalization.
* Feeding the blob to the face detection network.
* Extracting bounding boxes of faces with confidence above a threshold (e.g., 0.7).
* Drawing green rectangles around detected faces.

This logic ensures that only the most relevant face regions are forwarded to subsequent networks.

**4. Image Preprocessing:**

Each detected face region is extracted using the bounding box coordinates and resized to (227, 227), which is the expected input size for both gender and age models. The pixel values are normalized by subtracting mean values specific to the model’s training configuration.

**5. Gender Prediction Flow:**

* The normalized face is converted into a blob.
* The blob is fed into the gender network using genderNet.setInput(blob).
* A forward pass through the network gives the prediction vector.
* The gender is classified as either "Male" or "Female" based on the maximum confidence index.

**6. Age Prediction Flow:**

The same blob is reused and passed into the age network. The forward pass results in a probability distribution across predefined age ranges like:

* (0-2), (4-6), (8-12), (15-20), (25-32), (38-43), (48-53), and (60-100).
* The highest confidence score determines the predicted age group.

**7. Output Rendering and User Interface:**

The final gender and age prediction are:

* Printed on the console.
* Annotated on the video feed using cv2.putText() to display text just above the detected face.
* Displayed with bounding boxes via OpenCV’s GUI window.

This makes the interface both informative and interactive. The real-time updates make it ideal for live surveillance or interactive booths.

**8. Performance Considerations:**

The implementation is lightweight enough to run on modern laptops without requiring a GPU. However, GPU acceleration can drastically reduce inference time. The DNN module of OpenCV is highly optimized, making the frame rate sufficient for real-time use cases.

**9. Code Execution and Command-Line Usage:**

* Run the script directly in a terminal with or without the --image flag.
* Example: python gender\_age\_detect.py --image sample.jpg
* If the flag is omitted, the webcam is used.

**10. Error Handling and Edge Cases:**

* If no face is detected, a message is printed.
* The system checks frame availability from the webcam or video file to avoid crashes.

**11. Logic Design Summary:**

* **Input Source:** Webcam or image file.
* **Face Detection:** Deep learning-based bounding box extraction.
* **Image Preprocessing:** Blob creation with resizing and normalization.
* **Feature Inference:** Forward propagation through gender and age networks.
* **Output Display:** Real-time display with OpenCV GUI.

By leveraging OpenCV's modular design and pre-trained Caffe models, the project combines simplicity and power in its implementation. The logic ensures modularity, allowing for easy upgrades or integration with other vision-based systems such as face recognition, emotion detection, or pose estimation.

# Future Scope

The current implementation provides a foundational model for real-time gender and age detection. Future enhancements may include:

**Accuracy Improvement:** Incorporate more robust models trained on larger and more diverse datasets for increased prediction accuracy.

**Expanded Categories:** Refine age groups to include narrower age ranges and potentially add emotion recognition.

**Model Optimization:** Convert the model to lightweight formats (e.g., TensorFlow Lite or ONNX) for deployment on mobile and embedded devices.

**Web and Mobile Integration:** Build a web or mobile application interface using Flask, React, or Android frameworks.

**Multiple Face Detection:** Enhance the system to handle multiple faces simultaneously with individualized predictions.

**Privacy and Ethics:** Implement privacy safeguards and consider ethical implications of deploying such technologies in public or sensitive environments.

# Conclusion

# This project demonstrates the feasibility and practicality of implementing a real-time gender and age detection system using Python, OpenCV, and pre-trained deep learning models. The system accurately detects faces, classifies gender, and estimates age within a few milliseconds, making it ideal for real-world applications that require quick and reliable results. The simplicity and effectiveness of the implementation make it an excellent learning project for students, as well as a base for further research in demographic analytics and computer vision. The program serves as a working prototype that can be extended and improved based on application needs.

# Reference

1. **OpenCV Documentation: https://docs.opencv.org/**
2. **Caffe Model Zoo: https://github.com/BVLC/caffe**
3. **UTKFace Dataset: https://susanqq.github.io/UTKFace/**
4. **Adience Dataset: https://www.openu.ac.il/home/hassner/Adience/data.html**
5. **Python Official Website: https://www.python.org/**
6. **GitHub Repositories and Tutorials on Age & Gender Detection using OpenCV**