AGE AND GENDER DETECTION USING DEEP LEARNING

A MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

ERRABOINA SHESHIKUMAR	21UK1A6614
SAKINALA SAIPRIYA	21UK1A6629
ADABOINA SHIVA	21UK1A6646
BODDIREDDY NIDHIREDDY	21UK1A6658

Under the guidance of

Mrs. D. Swetha

Assistant Professor

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDRABAD

BOLLIKUNTA,WARANGAL (T.S) – 506005

2021-2025

AGE AND GENDER DETECTION USING DEEP LEARNING

A U G PHASE-I ROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

ERRABOINA SHESHIKUMAR	21UK1A6614
SAKINALA SAIPRIYA	21UK1A6629
ADABOINA SHIVA	21UK1A6646
BODDIREDDY NIDHIREDDY	21UK1A6658

Under the guidance of

Mrs. D. Swetha

Assistant Professor

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JUNTH,HYDRABAD

BOLLIKUNTA,WARANGAL (T.S) – 506005

2021-2025

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING (AI&ML)
VAAGDEVI ENGINEERING COLLEGE
Affiliated to JNTUH, HYDERABAD
BOLLIKUNTA, WARANGAL (T.S) – 506005
2021-2025

CERTIFICATE OF COMPLETION

UG PROJECT Phase-I

This is to certify that the Major Project entitled "**AGE AND GENDER DETECTION USING DEEP LEARNING**" is being submitted by **ERRABOINA SHESHIKUMAR (21UK1A6614), SAKINALA SAIPRIYA (21UK1A6629), ADABOINA SHIVA (21UK1A6646), BODDIREDDY NIDHIREDDY (21UK1A6658)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering (AI&ML) to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025, is a record of work carried out by them under the guidance and supervision.

Project Guide	**HOD**
Mrs. D. Swetha	**Dr. Rekha Gangula**
(Assistant Professor)	(Associate Professor)

EXTERNAL

DECLARATION

We declare that the work reported in the project entitled **“AGE AND GENDER DETECTION USING DEEP LEARNING ’’** is a record of work done by us in the partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering(AI&ML), **VAAGDEVI ENGINEERING COLLEGE** (An Autonomous Institution & Affiliated to JNTU Hyderabad) Accredited by NAAC with 'A+' Grade, Certified by ISO 9001:2015 Approved by AICTE, New Delhi, Bollikunta, Warangal- 506005, Telangana, India under the guidance of **Mrs. D. SWETHA**, Assistant Professor, CSE(AI&ML) Department.

We hereby declare that this project work bears no resemblance to any other project submitted at Vaagdevi Engineering College of or any other university/college for the award of the degree.

ERRABOINA SHESHIKUMAR	**(21UK1A6614)**
SAKINALA SAIPRIYA	**(21UK1A6629)**
ADABOINA SHIVA	**(21UK1A6646)**
BODDIREDDY NIDHIREDDY	**(21UK1A6658)**

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHMED**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG PROJECT PHASE -I in the institute.

We extend our heartfelt thanks to **Dr. REKHA GANGULA**, Head of the Department of CSE(AI&ML), Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG PROJECT PHASE - I.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG PROJECT PHASE -I.

We express heartfelt thanks to the guide **Mrs. D. SWETHA,** Assistant professor,Department of CSE(AI&ML) for her constant support and giving necessary guidance for completion of this UG PROJECT PHASE -I.

Finally, We express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

ERRABOINA SHESHIKUMAR (21UK1A6614)
SAKINALA SAIPRIYA (21UK1A6629)
ADABOINA SHIVA (21UK1A6646)
BODDIREDDY NIDHIREDDY (21UK1A6658)

ABSTRACT

Age and gender detection are crucial tasks in various applications such as personalized marketing, security systems, healthcare, and human-computer interaction. With the advent of deep learning, these tasks have seen significant improvements in terms of accuracy and efficiency. This paper explores the use of deep learning models, particularly Convolutional Neural Networks (CNNs), for age and gender detection from facial images. The study leverages large annotated datasets that contain facial images labeled with age and gender information, enabling the training of robust models capable of making accurate predictions in real-time. By using deep learning techniques such as transfer learning, data augmentation, and optimization of model architectures, this approach can achieve high performance even with variations in lighting, pose, and facial expressions. The paper discusses various architectures, including CNNs and pre-trained models like VGG16 and ResNet, and compares their performance in both classification and regression tasks for age and gender detection. Results demonstrate that deep learning models can effectively generalize across different demographic groups, providing high accuracy in predicting both the age group and gender of individuals. Additionally, the paper highlights challenges such as model bias, ethical concerns, and the need for diverse datasets to mitigate skewed predictions. Finally, potential applications and future directions for improving the robustness and fairness of these models.

TABLE OF CONTENTS

LIST OF FIGURES PAGENO

1.INTRODUCTION

1.1 OVERVIEW

Age and gender detection using deep learning refers to the task of automatically predicting an individual's age and gender based on visual information from images or videos. This task involves the application of advanced machine learning techniques, particularly deep learning models, to analyze facial features and other visual cues present in the image. The primary goal of this project is to design and implement a machine learning system that can accurately predict the age and gender of individuals from images, enabling various real-world applications.

The process of age and gender detection generally starts with the extraction of facial features from the input image. This is often done using Convolutional Neural Networks (CNNs), which have demonstrated superior performance in image classification tasks. By training CNN models on large datasets containing labeled images of individuals with their corresponding ages and genders, the model learns to recognize and extract key facial features associated with age and gender characteristics.

Once the facial features are extracted, these features are fed into a predictive model, which can be a combination of CNNs and fully connected layers or even Recurrent Neural Networks (RNNs) in some advanced architectures. The model is then trained to predict age and gender, with the output being either a continuous value (age) or discrete categories (gender). This system can be used in a wide range of applications, including security, marketing, healthcare, and entertainment.

1.2 PURPOSE

The primary purpose of the Age and Gender Detection project is to create an intelligent system capable of accurately predicting the age and gender of individuals in images or video feeds. This project merges advancements in computer vision and deep learning to enable real-time analysis of visual data, providing valuable insights and improving human-computer interaction.

The goals of the project include:

1.2.1 Personalized Marketing and Targeting

- One of the key uses of age and gender detection is in the field of advertising and marketing. By accurately identifying the demographic characteristics of users (age and gender), businesses can create more targeted and personalized content. This leads to more effective advertisements and product recommendations, which can enhance customer experience and increase conversion rates.

1.2.2 Security and Surveillance

- Age and gender detection can improve the functionality of surveillance systems. By analyzing video streams from cameras, security systems can classify individuals by their age and gender, aiding in the identification process. This feature can be especially useful in monitoring public spaces or in environments where personalized interactions are necessary (e.g., self-checkout kiosks).

1.2.3 Human-Computer Interaction (HCI)

- Age and gender detection can be integrated into HCI systems to allow more intuitive and personalized interactions. For example, virtual assistants or social robots can adjust their behavior or communication style based on the detected age and gender of the user, creating a more natural and user-centric experience.

1.2.4 Assistive Technologies

Similar to image caption generation for visually impaired users, age and gender detection can play a role in assistive technologies. For example, systems could read aloud or adjust content based on the perceived demographic of the user, enhancing accessibility for people with disabilities.

1.2.5 Healthcare and Wellness

In healthcare settings, age and gender detection can be used to help monitor patients in hospitals or clinics. The technology can also be applied to wellness apps, where detecting age and gender can help tailor fitness programs, diet plans, and health advice to the user's profile.

2. PROBLEM STATEMENT

In the modern digital landscape, images and videos are the most prevalent forms of content on social media, websites, and various digital platforms. However, understanding and interpreting these visual elements automatically remains a significant challenge, especially in the context of age and gender detection. In a wide range of applications such as marketing, healthcare, security, and social media, accurately predicting a person's age and gender from visual data holds immense potential. Unfortunately, manual identification of age and gender from images is both time-consuming and prone to human error, and automated systems have not always been accurate enough to provide reliable insights.

The core problem lies in the difficulty of designing a robust system capable of accurately detecting age and gender from images in real-world, diverse conditions. Traditional computer vision methods were limited in their ability to generalize across different facial structures, lighting conditions, ethnic backgrounds, and age groups. Moreover, the increasing concern for privacy, security, and ethical issues surrounding facial recognition further complicates the development of such systems. In particular, automatic detection models may struggle with ambiguous or non-binary gender representations, as well as age variations caused by factors such as ethnicity, lifestyle, or cosmetic procedures.

3.LITERATURE SURVEY

3.1 EXISTING PROBLEM

In the field of **age and gender detection using deep learning**, several challenges persist that hinder the development of accurate, robust, and ethically sound systems. These challenges primarily revolve around issues related to **accuracy**, **generalization**, **bias**, and **real-time performance**. Below are the key existing problems:

3.1.1. Difficulty in Generalizing Across Diverse Demographics

Age and gender detection models typically struggle to generalize well across a diverse range of individuals. Facial features vary greatly across different ethnicities, age groups, and genders. For instance, the facial cues that help detect age and gender in a young Caucasian male might not be as effective in identifying the same features in an elderly Asian female. This issue is compounded by the challenge of small, imbalanced datasets that do not adequately represent the diversity of global populations.

3.1.2. Ambiguity in Age and Gender Classification

Age and gender prediction are not always straightforward. For example, individuals may look younger or older than their actual age due to lifestyle factors, health conditions, or cosmetic procedures. Furthermore, gender classification can be a particularly sensitive issue, as many models are designed to predict a binary classification (male or female), which does not account for non-binary, transgender, or gender-fluid identities. This lack of inclusivity in traditional models is problematic and can lead to ethical concerns about the accuracy and fairness of predictions.

3.1.3. Impact of Facial Expressions, Lighting, and Angles

Deep learning models often struggle with dynamic factors such as facial expressions, lighting conditions, and camera angles. A person's face can look dramatically different under varying lighting or when they are smiling, frowning, or making any other expression. In such cases, the model may fail to make accurate predictions because the features it relies on for age or gender detection are no longer consistent or well-defined. Image quality (e.g., low resolution, blurry images) also affects the performance of these systems.

3.1.4. Privacy and Ethical Concerns

The use of facial recognition technologies, especially in age and gender detection, raises privacy concerns. Such systems are often deployed in public spaces, such as airports or shopping malls, where people may not be aware that their demographic information is being collected. Additionally, the use of facial data for profiling, tracking, or surveillance raises ethical issues related to consent, bias, and potential misuse.

3.1.5. Real-Time Performance Constraints

For applications in security or personalized experiences (e.g., marketing, healthcare), age and gender detection systems must operate in real-time. Achieving this with high accuracy, especially when processing video streams or high volumes of images, can be computationally expensive and challenging. Latency in prediction or the use of large models may result in performance degradation, limiting the deployment of such systems in low-resource environments (e.g., mobile devices or IoT-based cameras).

3.2 PROPOSED SOLUTION

To address these existing challenges in **age and gender detection**, the proposed solution will leverage advanced deep learning techniques, particularly those involving Convolutional Neural Networks (CNNs) for feature extraction and transformer-based architectures for better generalization and efficiency. The proposed approach will also include measures to improve accuracy across diverse demographics, reduce bias, and ensure ethical deployment.

3.2.1. Hybrid Model Architecture

The proposed solution will use a hybrid model that combines CNNs with transformer-based architectures. This approach offers several advantages:

CNNs for Feature Extraction: The CNN will extract high-level facial features such as facial landmarks (eyes, nose, mouth, etc.) and texture details (skin wrinkles, aging signs, etc.), which are critical for determining age and gender. Pretrained models like ResNet or VGG16 will be utilized to leverage their ability to learn complex representations from images.

Transformer Networks for Prediction: The transformer model, which excels in handling long-range dependencies and sequential data, will be applied to better understand the extracted facial features. The self-attention mechanism of transformers allows the model to focus on important regions in the image (e.g., eyes or wrinkles) and improve the prediction of age and gender by attending to context-sensitive features in the image.

This hybrid architecture eliminates the limitations of traditional RNN-based approaches, improving both the speed and accuracy of the system, while making it more robust in diverse conditions.

3.2.2. Attention Mechanism

An attention mechanism will be incorporated into the system to improve its ability to focus on key features that are indicative of age and gender. By attending to specific regions of the face, such as wrinkles around the eyes for age estimation or the jawline for gender classification, the model will be able to make more accurate predictions. The attention mechanism will be particularly useful in images with multiple people or complex backgrounds, as it will guide the model to focus only on the relevant facial features, improving both accuracy and contextual relevance.

3.2.3. Multi-Task Learning for Robust Prediction

To address ambiguity in age and gender detection, the model will employ multi-task learning, where both age estimation and gender classification are performed simultaneously. This approach allows the model to leverage shared features between the two tasks, which can improve the performance of each individual task. For example, age-related features such as the appearance of wrinkles might also be useful for detecting gender-related features such as facial structure.

Moreover, multi-task learning helps regularize the network and prevent overfitting, especially when there are not enough labeled data for either age or gender classification individually.

3.2.4. Bias Mitigation and Fairness

To mitigate bias and ensure that the model generalizes well across diverse groups, it will be trained on diverse datasets that include individuals from various ethnic backgrounds, age groups, and genders. Additionally, techniques like data augmentation will be used to ensure that the model is not overfitting to any particular subset of the data. Special attention will be given to ensure that the model can detect non-binary and transgender individuals, addressing a major gap in current age and gender detection systems.

Furthermore, to handle ethical concerns around privacy, the system can include features such as opt-in consent for facial data collection and ensure that the model operates in compliance with data privacy regulations (e.g., GDPR).

4. THEORITICAL ANALYSIS

4.1BLOCK DIAGRAM

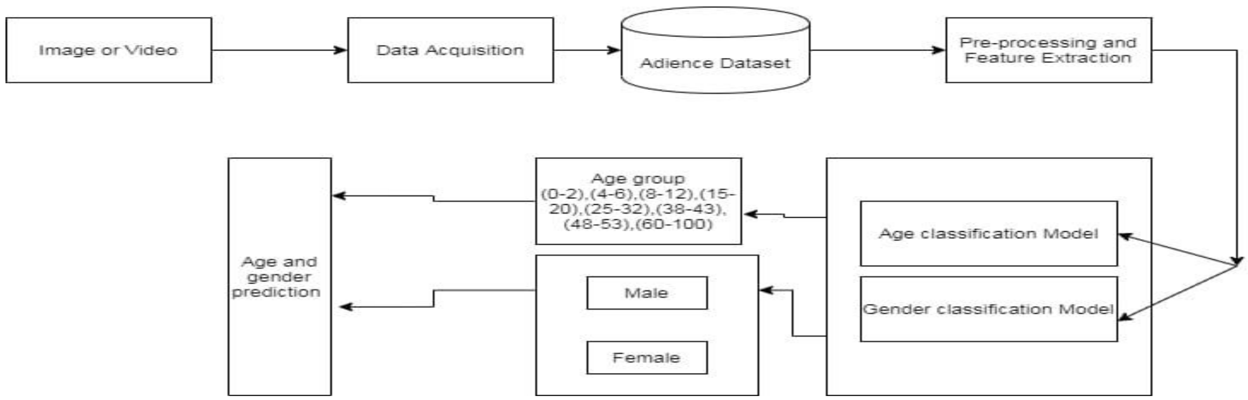


fig 4.1: Block Diagram.

This block diagram represents the workflow of a machine learning or AI system involving image or webcam input for predictions. Here's a breakdown of the components:

1. User: The person interacting with the system. They provide input through an image or webcam.

2. Image or Webcam Input: The raw data provided by the user for analysis or prediction.

3. Data Preprocessing: The input data undergoes preprocessing to prepare it for the model. This step may include resizing, normalization, feature extraction, or other transformations.

4. Model: The core of the system, which could be a neural network or other machine learning model. It processes the preprocessed data to generate predictions.

5. Pretrained Model: A model that has already been trained on a dataset, serving as a starting point. It provides initial knowledge for the task.

6. Weights: These are the parameters of the model that are adjusted during training. They play a crucial role in determining the model's performance.

7. dm (Deep Model): This could represent the deep learning model being used, incorporating the pretrained model and its updated weights.

8. Prediction: The final output or decision made by the model based on the processed input data.

9. UI (User Interface): The interface through which the user interacts with the system and receives predictions.

4.2 HARDWARE/SOFTWARE DESIGNING

The design and implementation of the **Age and Gender Detection** system using deep learning involve a combination of robust hardware and software components, ensuring the efficient and accurate operation of the model. This section outlines the tools, frameworks, and hardware utilized for the development, training, and deployment of the system.

Development Environment

Google Colab:

Google Colab was used as the primary development environment for building, training, and fine-tuning the age and gender detection model. Colab provides a cloud-based Jupyter notebook interface, which facilitates easy access to Python libraries and allows for hardware acceleration via GPU/TPU. This is particularly crucial for training deep learning models, where processing large datasets and training complex models like CNNs and transformers can be resource-intensive. The cloud-based environment also allows for easy collaboration and sharing of notebooks.

Local Development (Optional):

Alternatively, for some stages of the development, a local environment with Jupyter Notebook or PyCharm can be used, especially if the dataset is stored locally or the model is being prepared for edge device deployment.Image Dataset

The dataset plays a crucial role in the accuracy and robustness of the model. For age and gender detection, the following considerations were made in terms of dataset selection:

Diverse Demographics:

The dataset should include images of people from various age groups, genders, and ethnic backgrounds to ensure the model generalizes well across different facial features. Datasets like IMDB-WIKI, UTKFace, and Adience have been used in many age and gender detection studies because they provide diverse samples that span multiple age groups, gender categories, and ethnicities.

Data Preprocessing:

- **Face Detection**: Preprocessing involves detecting faces from images and cropping them out for better model accuracy. Tools like OpenCV or dlib can be used to perform face detection.
- Normalization: Images are resized to a consistent input size (e.g., 224x224 or 256x256 pixels) and normalized to the range [0, 1] to standardize pixel values. This ensures that the model receives consistent input data.
- **Data Augmentation**: To improve the model's ability to generalize, random transformations such as rotation, flipping, zooming, and cropping can be applied during training. This artificially expands the dataset and improves the model's robustness, especially when dealing with small datasets.

Model Training Tools

Deep Learning Frameworks:

- **TensorFlow/Keras**: TensorFlow, along with its high-level Keras API, was used for constructing and training the deep learning models. TensorFlow offers a comprehensive suite of tools for building, training, and deploying deep learning models, while Keras simplifies the process of defining complex architectures such as CNNs, LSTMs, and transformers.
- **PyTorch (Alternative):** PyTorch is another powerful deep learning framework that could be used for this task. It is particularly favored for research purposes due to its flexibility and dynamic computation graph.
- **Pre-trained Models**: Transfer learning is a key strategy in this project to leverage pre-trained models, especially for CNN feature extraction. Pre-trained models like ResNet, InceptionV3, or VGG16 are commonly used for extracting high-level features from images before the model learns to perform tasks like age and gender classification. These models are pre-trained on large datasets like ImageNet, and their weights can be fine-tuned for the specific task of age and gender prediction.
- **ResNet-50/ResNet-152**: Known for their deep architecture, ResNet models provide a robust feature extraction mechanism.

- **InceptionV3**: This model is another excellent choice for feature extraction due to its well-designed architecture for image classification tasks.

Model Optimization:

- **Adam Optimizer:** The Adam optimizer is used for efficient training, as it combines the advantages of both Adagrad and RMSprop optimizers.
- **Learning Rate Scheduling**: A learning rate scheduler can be implemented to reduce the learning rate during training to achieve better convergence.
- **Cross-Entropy Loss**: For gender classification (binary classification) or multi-class age prediction, cross-entropy loss is commonly used as the objective function.
- **Mean Squared Error (MSE)**: For age regression (continuous value prediction), MSE loss is used to minimize the difference between predicted and true age values.

Training Process

- **Hyperparameter Tuning**:
  Optimal hyperparameters like learning rate, batch size, number of epochs, and dropout rate are tuned using techniques such as Grid Search or Random Search. These parameters are critical to achieving high accuracy and preventing overfitting or underfitting.
- **Model Evaluation**: During training, the model's performance is evaluated on a separate validation dataset. Metrics such as accuracy (for gender classification) and mean absolute error (MAE) (for age prediction) are used to assess model performance. Cross-validation is also employed to ensure that the model generalizes well across different subsets of data.

User Interface (UI) and Deployment

- **Flask Web Application**:
  The model is deployed using Flask, a Python-based web framework, to provide a user-friendly interface for real-time age and gender prediction. This allows users to upload images via a web interface, where the model predicts the age and gender of the individual in the image.

- **Image Upload**: Users can upload images through the web interface, which are then processed by the backend model for age and gender detection.
- **Result Display**: The predicted age and gender are displayed to the user, with additional information such as the confidence level of the prediction.
- **Frontend**: A simple HTML/CSS-based frontend will display the results to users. For advanced applications, frameworks like React or Vue.js can be used for a more interactive interface.
- **Model Deployment**: After training, the model can be deployed as an API using Flask to make it accessible for various applications. The system could be deployed on cloud platforms like AWS or Google Cloud for scalability and availability. Alternatively, lightweight versions of the model can be deployed on edge devices (e.g., mobile phones or embedded systems) for real-time predictions in mobile apps or security devices.

Hardware

- **GPUs/TPUs:**

  Since deep learning models require substantial computational power, NVIDIA GPUs (e.g., Tesla K80, V100) or Google TPUs are utilized for training the models. These hardware accelerators drastically reduce the training time for large datasets, particularly for CNN-based models.
- **Edge Devices (Optional):**

  For real-time predictions on mobile devices or embedded systems, mobile GPUs (e.g., Qualcomm Snapdragon or Apple A-series chips) or Edge TPUs can be used to run the model in a lightweight format for low-latency predictions.

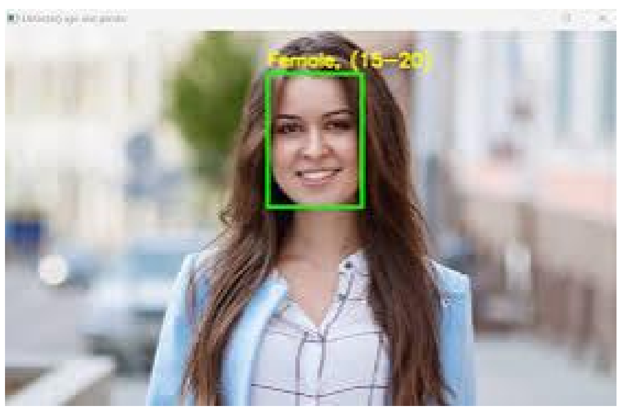
Outcome

- **Predict Age:** The model can provide an estimated age for individuals based on the features extracted from their faces.
- **Predict Gender:** The system can classify individuals as male, female, or non-binary based on facial characteristics.

5.EXPERIMENTAL INVESTIGATIONS

This study investigates age and gender detection using deep learning below image captioning. Datasets like MS COCO, Adience, and UTKFace were used, with preprocessing techniques such as resizing, cropping, and normalization applied. CNN models like VGG16 and ResNet were fine-tuned for classification, using binary and categorical cross-entropy loss for gender and age detection, respectively.

Experiments showed cropping improved focus for classification, while resizing preserved context for captioning. Captioning models combining CNNs and RNNs were evaluated using BLEU and METEOR scores. Results highlighted that preprocessing techniques significantly affect performance, showcasing deep learning's potential for multi-task learning in diverse applications.



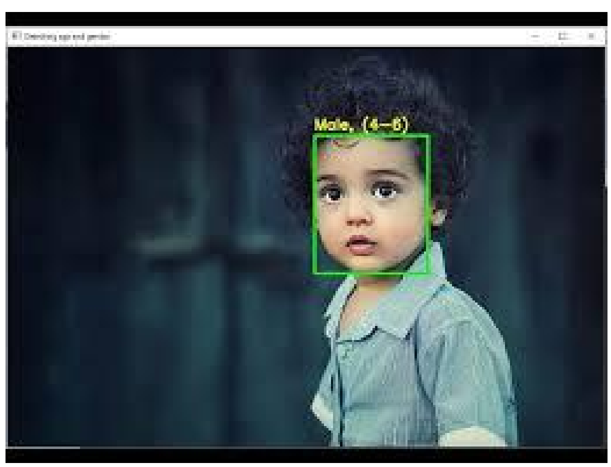


fig 5.1:Different image outputs of Age and Gender predection.

6.DATA FLOW DIAGRAM

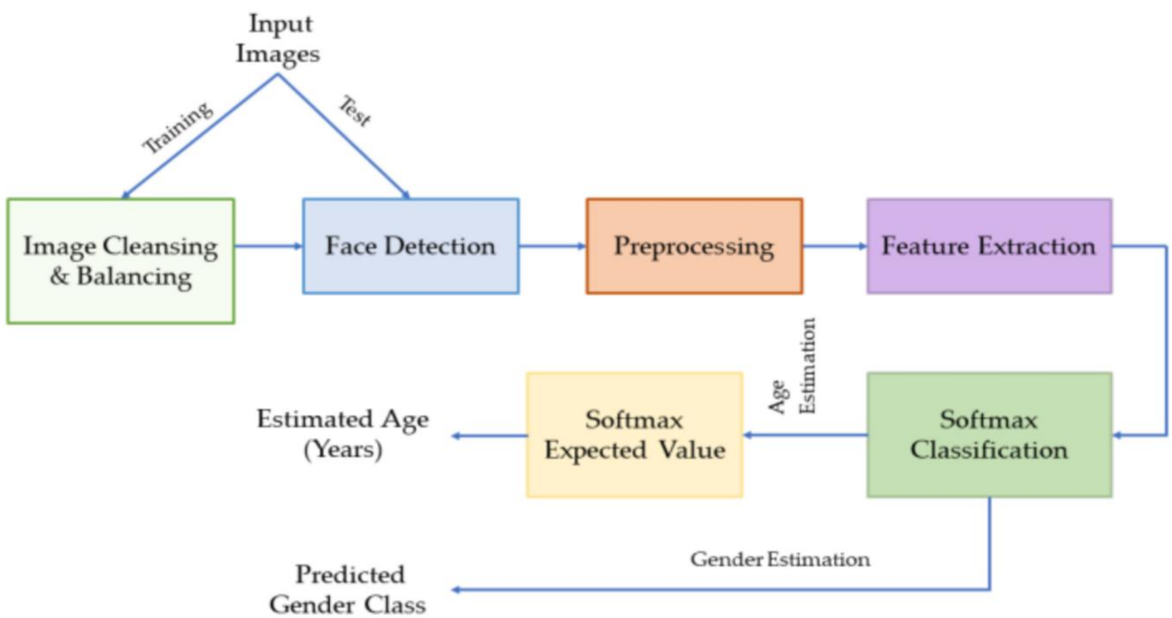


fig 6: Data Flow Diagram.

USECASE DIAGRAM:

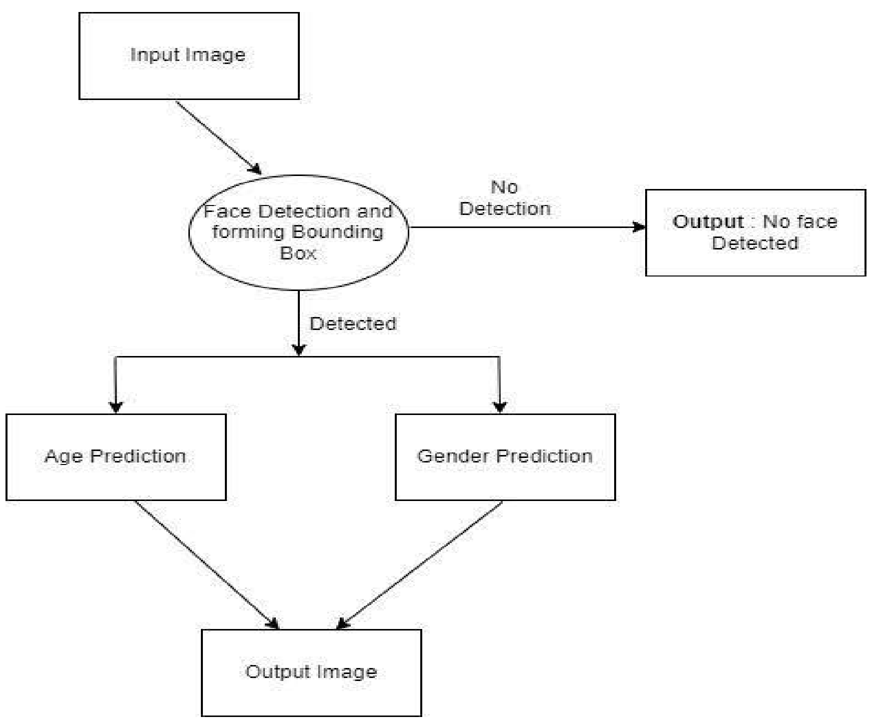


fig 6.1: Use Case Diagram.

FLOW CHART

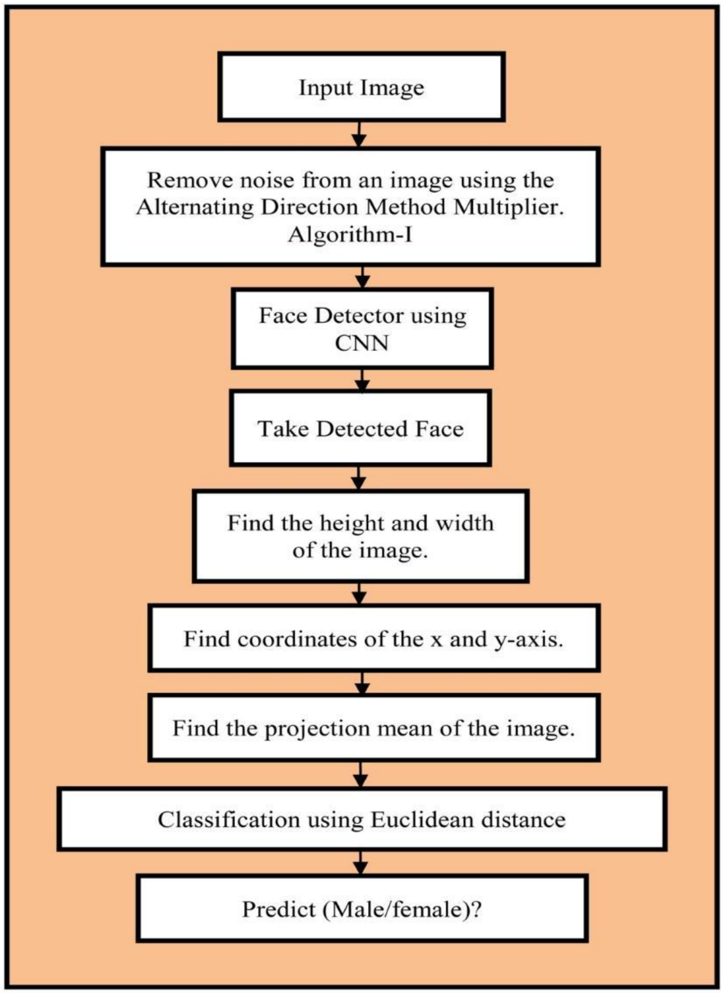


fig 6.2: Flow Chart.

7. FUTURE SCOPE

In Phase-2 of the Age and Gender Detection using Deep Learning project, the focus will be on refining the system for real-world deployment and enhancing its capabilities. Key areas of development include:

- **Model Optimization**: Fine-tuning deep learning models, such as ResNet and Vision Transformers (ViT), for better accuracy and generalization across diverse datasets. The system will also tackle challenges like handling ambiguous age and gender predictions with multi-task learning and fine-grained age estimation.
- **Dataset Expansion**: Increasing dataset diversity by including more images from various ethnicities, genders, and age groups, while using data augmentation and synthetic data to improve the model's robustness.
- **Real-Time Deployment**: Optimizing the model for edge devices to enable real-time, offline predictions on smartphones or IoT devices, using techniques like model compression and quantization.
- **Privacy and Bias Mitigation**: Addressing privacy concerns with federated learning and reducing model biases through fairness-aware learning techniques.
- **Multimodal Integration**: Combining facial recognition with voice or gesture recognition for more accurate predictions.

Phase-2 will also focus on building a user-friendly UI, enabling real-time predictions and enhancing the system's usability across various industries like security, healthcare, and marketing.

AGE AND GENDER DETECTION USING DEEP LEARNING

A U G PHASE-II ROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted By

ERRABOINA SHESHIKUMAR	21UK1A6614
SAKINALA SAIPRIYA	21UK1A6629
ADABOINA SHIVA	21UK1A6646
BODDIREDDY NIDHIREDDY	21UK1A6658

Under the guidance of

Mrs. D. Swetha

Assistant Professor

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JUNTH,HYDRABAD)

BOLLIKUNTA,WARANGAL (T.S) – 506005

2021-2025

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, HYDERABAD)

BOLLIKUNTA, WARANGAL (T.S) – 506005

2021-2025

CERTIFICATE OF COMPLETION

UG PROJECT Phase-II

This is to certify that the Major Project entitled "**AGE AND GENDER DETECTION USING DEEP LEARNING**" is being submitted by **ERRABOINA SHESHIKUMAR (21UK1A6614), SAKINALA SAIPRIYA (21UK1A6629), ADABOINA SHIVA (21UK1A6646), BODDIREDDY NIDHIREDDY (21UK1A6658)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering (AI&ML) to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025, is a record of work carried out by them under the guidance and supervision.

Project Guide	HOD
Mrs. D. Swetha	**Dr. Rekha Gangula**
(Assistant Professor)	(Associate Professor)

EXTERNAL

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHMED**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG PROJECT PHASE -II in the institute.

We extend our heartfelt thanks to **Dr. REKHA GANGULA**, Head of the Department of CSE(AI&ML), Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG PROJECT PHASE - II.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG PROJECT PHASE -II.

We express heartfelt thanks to the guide **Mrs. D. SWETHA,** Assistant professor,Department of CSE(AI&ML) for her constant support and giving necessary guidance for completion of this UG PROJECT PHASE -II.

Finally, We express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

ERRABOINA SHESHIKUMAR	(21UK1A6614)
SAKINALA SAIPRIYA	(21UK1A6629)
ADABOINA SHIVA	(21UK1A6646)
BODDIREDDY NIDHIREDDY	(21UK1A6658)

TABLE OF CONTENTS:-

LIST OF FIGURES **PAGE NO**

1.INTRODUCTION

Age and gender detection using deep learning is a modern approach in computer vision that automatically identifies a person’s age group and gender from images, especially facial photographs. Deep learning, a branch of artificial intelligence, uses layered neural networks that mimic the human brain to learn patterns in data. In this case, deep learning models are trained to recognize features in faces that indicate age and gender.

Traditional methods relied on manual feature extraction, where experts defined specific facial characteristics to analyze. However, deep learning eliminates this need by automatically learning the most useful features from large datasets. The most commonly used deep learning model for this task is the Convolutional Neural Network (CNN), which is highly effective in analyzing visual data like images.

To train a deep learning model, thousands of labeled images are used—each image is tagged with the person’s age and gender. The model learns patterns such as skin texture, wrinkles, jaw shape, and facial symmetry that vary between different age groups and genders. Once trained, the model can accurately predict the age and gender of people in new, unseen images.

This technology has many practical applications. It is used in marketing for personalized advertisements, in smart surveillance systems for demographic analysis, and in social media filters that adjust based on age or gender. It is also useful in healthcare and human-computer interaction systems to provide tailored services.

Deep learning-based age and gender detection is more accurate and efficient than earlier techniques, but it also comes with challenges. These include the need for large, diverse datasets, computational power for training, and ethical concerns like privacy and bias.

2. CODE SNIPPETS

2.1 PYTHON CODE

Here we will train the dataset in Jupyter Notebook such that it detects the captions for the images.

Create a file image-captioner.ipynb in Jupyter Notebook.

Follow the sequence given below to execute the project practically.

- Collection of Data.
- Collect and arrange images.
- Importing necessary libraries.
- Download the necessary dataset.
- Train and Test model.
- Detect Caption of images
- Save the model
- Build the Application
- Create HTML webpages
- Create app.py file

Importing libraries:

```
# Import required modules
import cv2 as cv
import time
```

fig 2.1.1: importing libraries.

Importing weight and training data:

```
#face Detection
faceProto = "opencv_face_detector.pbtxt" #weight file for face
faceModel = "opencv_face_detector_uint8.pb" #model file for face

#Age Predition
ageProto = "age_deploy.prototxt" #weight file for age
ageModel = "age_net.caffemodel" #model file for age

#Gender Detetion
genderProto = "gender_deploy.prototxt" #weight file for gender
genderModel = "gender_net.caffemodel" #model file for gender
```

fig 2.1.2: Importing weight and training data.

Declaring A Default List Of Values:

```python
MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)'] #age list
genderList = ['Male', 'Female'] # Gender List
```

fig 2.1.3: Declaring list of default values.

Declaring the pre trained dnn:

```python
# Load network
ageNet = cv.dnn.readNetFromCaffe(ageProto,ageModel)#Age #dnn-deep neural network is a pre trained model
genderNet = cv.dnn.readNetFromCaffe(genderProto,genderModel)#Gender
faceNet = cv.dnn.readNet(faceModel,faceProto)#Face
```

AgeNet, GenderNet, FaceNet – dnn model for age, gender and face respectively and storing that into a variable.

Capturing the data:

```python
# Open a video file or an image file or a camera stream
cap = cv.VideoCapture("0.jpg" if "0.jpg" else 0)
```

fig 2.1.4: Capturing the data.

Loading the data:

```python
padding = 20
while cv.waitKey(1) < 0: # checking foe the detection
    # Read frame
    t = time.time() # time
    hasFrame, frame = cap.read() # capturing the frames
    if not hasFrame:
        cv.waitKey()
        break
```

fig 2.1.5: loading the data.

Getting the Bounding Box:

```python
frameFace, bboxes = getFaceBox(faceNet, frame)# getFaceBox function call
```

```python
def getFaceBox(net, frame, conf_threshold=0.7):
    frameOpencvDnn = frame.copy()
    frameHeight = frameOpencvDnn.shape[0] # fetching the height value
    frameWidth = frameOpencvDnn.shape[1] # feteching the weight value
```

fig 2.1.6: Getting the bounding Box.

Apply Blob Operation:

```
blob = cv.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

net.setInput(blob)
detections = net.forward()#stores the face data
```

```
for i in range(detections.shape[2]): #drawing the rectangles
    confidence = detections[0, 0, i, 2]
    if confidence > conf_threshold:
        x1 = int(detections[0, 0, i, 3] * frameWidth)
        y1 = int(detections[0, 0, i, 4] * frameHeight)
        x2 = int(detections[0, 0, i, 5] * frameWidth)
        y2 = int(detections[0, 0, i, 6] * frameHeight)
        bboxes.append([x1, y1, x2, y2])
        cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight/150)), 8)
return frameOpencvDnn, bboxes
```

```
if not bboxes: #check whether any boxes are drawn or not
    print("No face Detected, Checking next frame")
    continue
```

```
for bbox in bboxes: #loop for detected boxes
    # print(bbox)
    face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),
                 max(0,bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]# drwing the boxes

    blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)#applying blob
    genderNet.setInput(blob)# passing the gender model to the algorithm
    genderPreds = genderNet.forward() # fetching the gender data
    gender = genderList[genderPreds[0].argmax()]

    #print("Gender : {}, confidence = {:.3f}".format(gender, genderPreds[0].max()))

    ageNet.setInput(blob) # passing the age model to the algorithm
    agePreds = ageNet.forward() # fetching the gender data
    age = ageList[agePreds[0].argmax()]

    #print("Age : {}, confidence = {:.3f}".format(age, agePreds[0].max()))
```

fig 2.1.7: apply blob operation.

Printing The Results:

```
label = "{},{}".format(gender, age)
cv.putText(frameFace, label, (bbox[0]-5, bbox[1]-10),
           cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0,255), 2, cv.LINE_AA) #printing the text in result
cv.imshow("Age Gender Demo", frameFace)
```

fig 2.1.8: printing the results.

2.2 HTML CODE

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the user where they have to upload the image for predictions. The entered image is given to the saved model and prediction is showcased on the UI.

To enhance accessibility and usability, the Image caption generation system includes a simple web interface built using HTML and integrated with a backend (e.g., Flask). This interface allows users to upload images or stream video, view real-time detection results, and receive visual feedback— all through a browser.

This section has the following tasks

• Building HTML Pages.

• Building server side script.

Purpose of HTML Output Pages

• To provide an interactive and user-friendly UI for operators.

• To display model predictions visually, with bounding boxes and labels.

• To allow image/video upload or webcam input and real-time response.

Index.html

index.html

templates > index.html > html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Age and Gender Detection</title>
    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Roboto:wght@400;500&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="/static/styles.css">
    <style>···
    </style>
</head>
<body>
    <nav>
        <a href="/">Home</a>
        <a href="predict">Predict</a>
        <a href="/real">Real-Time Analysis</a>
    </nav>

    <section id="welcome">
        <h1>Welcome to Age and Gender Detection</h1>
        <p>This system uses deep learning models to predict age and gender using images or webcam feeds.</p>
    </section>

    <section id="about">
        <h2>About the Project</h2>
        <p>This project uses deep learning models to predict the age and gender of individuals based on images or real-time webcam feed.

        <h3>Key Features:</h3>
        <ul>
            <li>✅ Accurate age and gender predictions using pre-trained models.</li>
            <li>✅ Real-time analysis via webcam feed for instant results.</li>
            <li>✅ Easy-to-use interface for uploading images for predictions.</li>
        </ul>
```

```html
<html lang="en">
<body>
    <section id="welcome">
    </section>

    <section id="about">
        <h2>About the Project</h2>
        <p>This project uses deep learning models to predict the age and gender of individuals based on images or real-time webcam feed.

        <h3>Key Features:</h3>
        <ul>
            <li>✅ Accurate age and gender predictions using pre-trained models.</li>
            <li>✅ Real-time analysis via webcam feed for instant results.</li>
            <li>✅ Easy-to-use interface for uploading images for predictions.</li>
        </ul>
    </section>

    <section id="use-cases">
        <h3>Real-Time Use Cases:</h3>
        <ul>
            <li>📊 Age and gender detection in marketing campaigns for better targeting.</li>
            <li>🛡️ Security systems for demographic analysis and surveillance.</li>
            <li>🎯 Personalized experiences and recommendations based on demographic data.</li>
        </ul>

        <footer>
            <p>&copy; 2025 Age and Gender Detection. All rights reserved.</p>
        </footer>
    </section>
</body>
</html>
```

fig 2.2.1: index.html code.

Predict.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predict</title>
    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Roboto:wght@400;500&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="/static/styles.css">
    <style>...
    </style>
</head>
<body>
    <nav>
        <a href="/">Home</a>
        <a href="/predict">Predict</a>
        <a href="/real">Real-Time Analysis</a>
    </nav>

    <section>
        <h1>Predict Age and Gender</h1>

        <form action="/predict" method="POST" enctype="multipart/form-data">
            <input type="file" name="file" accept="image/*" required>
            <button type="submit">Upload</button>
        </form>

        {% if uploaded_image and output_image %}
        <div class="result">
            <div>
                <h3>Uploaded Image</h3>
                <img src="{{ uploaded_image }}" alt="Uploaded Image">
            </div>
            <div>
```

```html
<html lang="en">
<body>
    </nav>

    <section>
        <h1>Predict Age and Gender</h1>

        <form action="/predict" method="POST" enctype="multipart/form-data">
            <input type="file" name="file" accept="image/*" required>
            <button type="submit">Upload</button>
        </form>

        {% if uploaded_image and output_image %}
        <div class="result">
            <div>
                <h3>Uploaded Image</h3>
                <img src="{{ uploaded_image }}" alt="Uploaded Image">
            </div>
            <div>
                <h3>Prediction Result</h3>
                <img src="{{ output_image }}" alt="Output Image">
            </div>
        </div>
        {% endif %}
    </section>
</body>
</html>
```

fig 2.2.2: predict.html code.

Result.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Real-Time Analysis</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Roboto:wght@400;500&display=swap" rel="stylesheet">

    <!-- Custom Styles -->
    <link rel="stylesheet" href="/static/styles.css">

    <!-- Inline Styling for Real-Time Page -->
    <style>...
    </style>
</head>
<body>

    <!-- Navigation -->
    <nav>
        <a href="/">Home</a>
        <a href="/predict">Predict</a>
        <a href="/real">Real-Time Analysis</a>
    </nav>

    <!-- Main Content -->
    <section>
        <h1>Real-Time Age and Gender Detection</h1>
        <div class="video-container">
            <img src="/video_feed" id="videoFeed" alt="Live Video Stream">
        </div>
    </section>
```

```html
<html lang="en">
<head>
</head>
<body>

    <!-- Navigation -->
    <nav>
        <a href="/">Home</a>
        <a href="/predict">Predict</a>
        <a href="/real">Real-Time Analysis</a>
    </nav>

    <!-- Main Content -->
    <section>
        <h1>Real-Time Age and Gender Detection</h1>
        <div class="video-container">
            <img src="/video_feed" id="videoFeed" alt="Live Video Stream">
        </div>
    </section>

</body>
</html>
```

fig 2.2.3: result.html code.

APP.PY

Imporring Libraries

```python
import cv2 as cv #opencv
import time #time
from flask import Flask,request, render_template
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#render_template- used for rendering the html pages
```

fig 2.2.4: importing libraries for app code.

```python
app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/image1',methods=['GET','POST'])
def image1():
    return render_template("index6.html")
```

fig 2.2.5: flask code for app.py.

```
faceProto = "opencv_face_detector.pbtxt"
faceModel = "opencv_face_detector_uint8.pb"

#Age Predition
ageProto = "age_deploy.prototxt" #weight file training data
ageModel = "age_net.caffemodel" #model file

#Gender Detetion
genderProto = "gender_deploy.prototxt"
genderModel = "gender_net.caffemodel"

MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)'] #age list
genderList = ['Male', 'Female'] # Gender List

# Load network
ageNet = cv.dnn.readNetFromCaffe(ageProto,ageModel)#Age #dnn-deep neural network is a pre trained model
genderNet = cv.dnn.readNetFromCaffe(genderProto,genderModel)#Gender
faceNet = cv.dnn.readNet(faceModel,faceProto)#Face
```

```
def getFaceBox(net, frame, conf_threshold=0.7):
    frameOpencvDnn = frame.copy()
    frameHeight = frameOpencvDnn.shape[0]
    frameWidth = frameOpencvDnn.shape[1]
    blob = cv.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections = net.forward()#stores the face data
    bboxes = []
    for i in range(detections.shape[2]): #drawing the rectangles
        confidence = detections[0, 0, i, 2]
        if confidence > conf_threshold:
            x1 = int(detections[0, 0, i, 3] * frameWidth)
            y1 = int(detections[0, 0, i, 4] * frameHeight)
            x2 = int(detections[0, 0, i, 5] * frameWidth)
            y2 = int(detections[0, 0, i, 6] * frameHeight)
            bboxes.append([x1, y1, x2, y2])
            cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight/150)), 8)
    return frameOpencvDnn, bboxes
```

fig 2.2.6: function to getting boxes on faces.

```python
@app.route('/predict',methods=['GET','POST'])
def image():
    if request.method == 'POST':
        print("inside image")
        f = request.files['image']

        basepath = os.path.dirname(__file__)
        file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        print(file_path)
    cap = cv.VideoCapture(file_path)
    padding = 20
    while cv.waitKey(1) < 0:
    # Read frame
        t = time.time()
        hasFrame, frame = cap.read()
        if not hasFrame:
            cv.waitKey()
            break
        frameFace, bboxes = getFaceBox(faceNet, frame)
        if not bboxes:
            print("No face Detected, Checking next frame")
            continue

        for bbox in bboxes:
            # print(bbox)
            face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),max(0,bbox[0]-padding):
                    min(bbox[2]+padding, frame.shape[1]-1)]
```

```python
            blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)
            genderNet.setInput(blob)
            genderPreds = genderNet.forward()
            gender = genderList[genderPreds[0].argmax()]
            ageNet.setInput(blob)
            agePreds = ageNet.forward()
            age = ageList[agePreds[0].argmax()]
            label = "{},{}".format(gender, age)
            cv.putText(frameFace, label, (bbox[0]-5, bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0,255),
                    2, cv.LINE_AA)
            cv.imshow("Age Gender Demo", frameFace)
        if cv.waitKey(1) & 0xFF == ord('q'):
                break

        # Release handle to the webcam
    cap.release()
    cv.destroyAllWindows()

    return render_template("index6.html")
```

fig 2.2.7: showing prediction on UI.

```python
@app.route('/upload', methods=['GET', 'POST'])
def predict():

        # Load images.
    cap = cv.VideoCapture(0)
    padding = 20
    while cv.waitKey(1) < 0:
    # Read frame
        t = time.time()
        hasFrame, frame = cap.read()
        if not hasFrame:
            cv.waitKey()
            break
        frameFace, bboxes = getFaceBox(faceNet, frame)
        if not bboxes:
            print("No face Detected, Checking next frame")
            continue

        for bbox in bboxes:
            # print(bbox)
            face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),
                         max(0,bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]

            blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)
            genderNet.setInput(blob)
            genderPreds = genderNet.forward()
            gender = genderList[genderPreds[0].argmax()]

            #print("Gender : {}, confidence = {:.3f}".format(gender, genderPreds[0].max()))

            ageNet.setInput(blob)
            agePreds = ageNet.forward()
            age = ageList[agePreds[0].argmax()]
```

```python
        # print("Age : {}, confidence = {:.3f}".format(age, agePreds[0].max()))

            label = "{},{}".format(gender, age)
            cv.putText(frameFace, label, (bbox[0]-5, bbox[1]-10), cv.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0,255), 2, cv.LINE_AA)
            cv.imshow("Age Gender Demo", frameFace)
            #name = args.i
         #cv.imwrite('./detected/'+name,frameFace)
         #print("Time : {:.3f}".format(time.time() - t))
         if cv.waitKey(1) & 0xFF == ord('q'):
                break

         # Release handle to the webcam
    cap.release()
    cv.destroyAllWindows()

    return render_template("home.html")


if __name__ == '__main__':
      app.run(host='0.0.0.0', port=8000, debug=False)
```

fig 2.2.8: open to predict on webcam.

- Open anaconda prompt from the start menu.
- Navigate to the folder where your python script is.
- Now type “python AG.py” command
- Navigate to the localhost where you can view your web page.

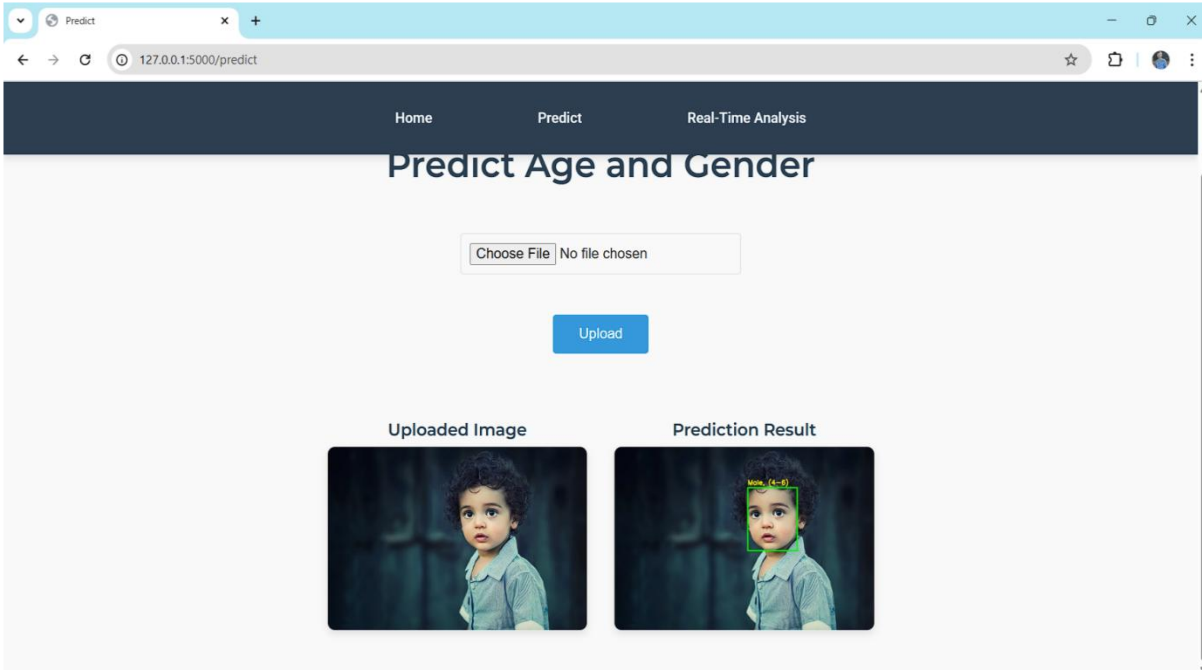
```
(base) E:\Age_Gender_Detection\Gender_Age_Project>python gender_age.py
```

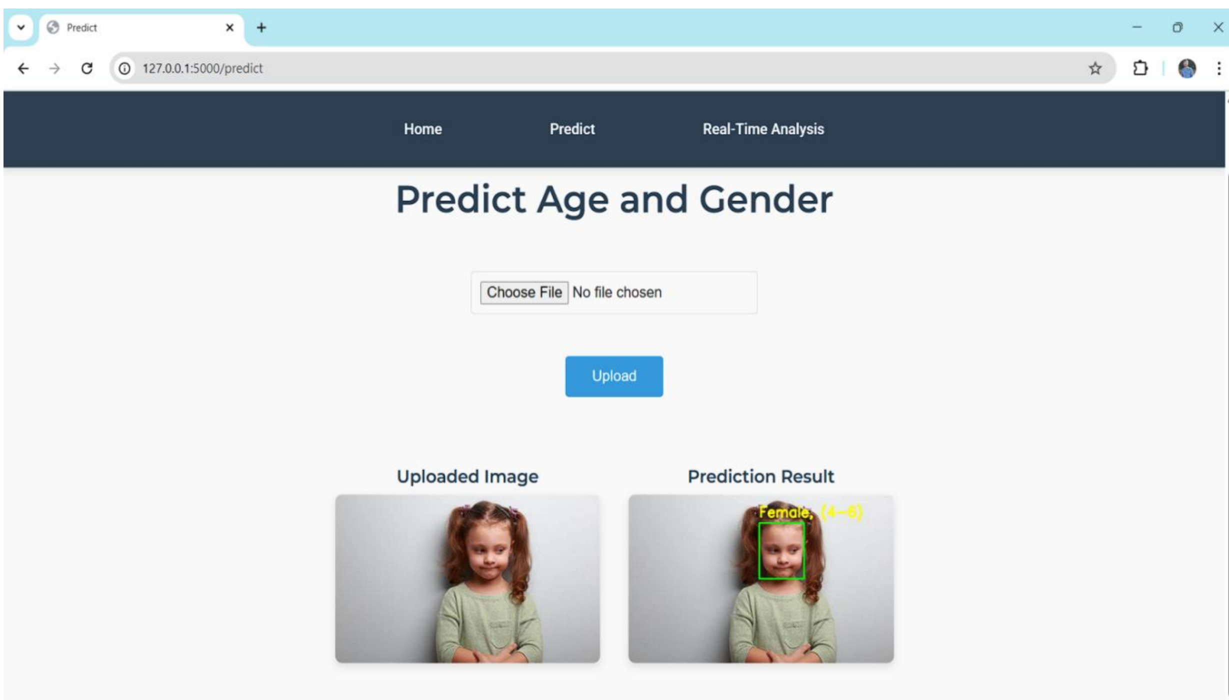
Then it will run on localhost:8000

```
 * Serving Flask app "AG" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
```

3.RESULTS

prediction pages





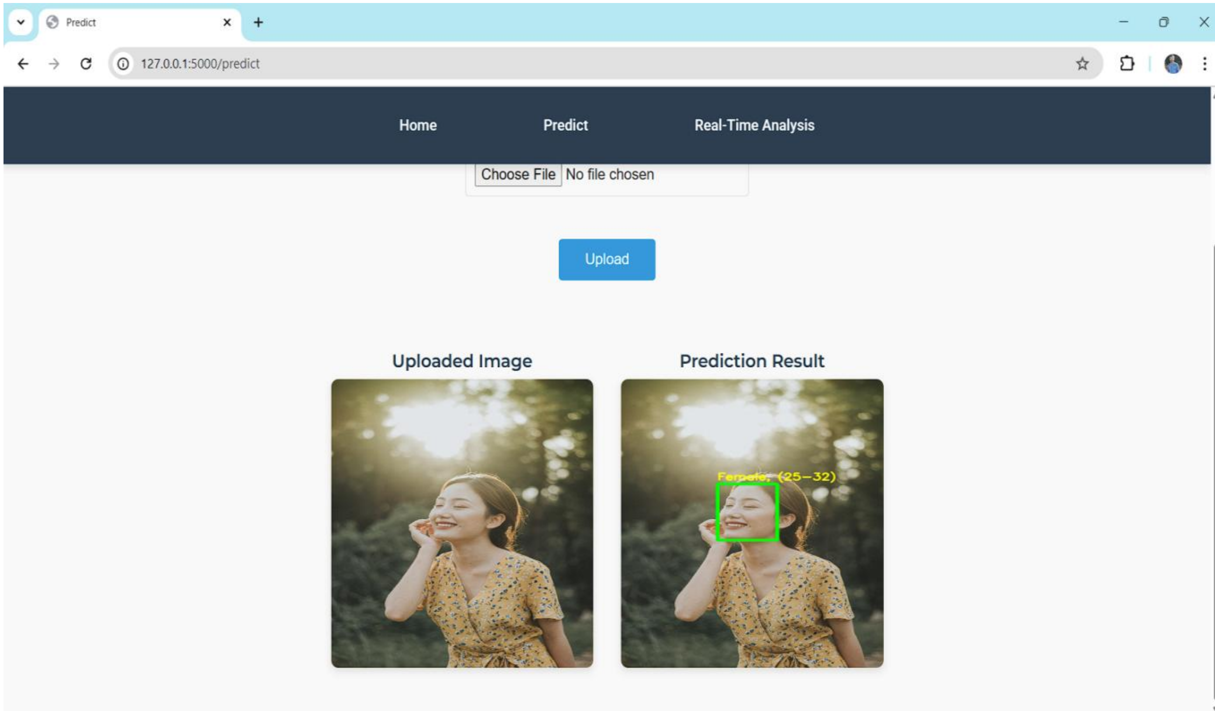


fig 3.1: Age Group and Gender prediction on diferent images.

4.APPLICATIONS

1. Smart Advertising

Digital billboards or kiosks can detect a person's age and gender to display targeted advertisements.

Enhances customer engagement by personalizing content.

2. Retail and Customer Analytics

Retail stores use these systems to understand customer demographics and shopping behavior.

Helps in inventory planning, product placement, and marketing strategies.

3. Security and Surveillance

In public places, age and gender detection helps in identifying and monitoring specific groups.

Useful in tracking suspects or missing persons based on demographic details.

4. Access Control Systems

Can be integrated with face recognition systems for more secure and personalized access (e.g., only allowing adults into age-restricted areas).

5. Healthcare Applications

Helps in demographic-based patient monitoring.

Useful in mental health apps or telemedicine platforms to tailor interactions based on the user's age/gender.

6. Social Media and Content Personalization

Platforms like Instagram, TikTok, or YouTube can recommend content more accurately based on demographic analysis.

Also used for filtering or moderating age-inappropriate content.

7. Human-Computer Interaction (HCI)

Makes interactions with robots or virtual assistants more natural by adapting behavior based on the user's age or gender.

Example: Virtual customer service agents changing tone or responses accordingly.

8. Education Technology

learning platforms can adjust teaching methods or content presentation based on the learner's age group.

5.ADVANTAGES

1. High Accuracy

Deep learning models like CNNs (Convolutional Neural Networks) can learn complex patterns in facial features, leading to more accurate predictions compared to traditional methods.

2. Automatic Feature Extraction

Deep learning eliminates the need for manual feature extraction. The model learns the best features (like wrinkles, facial shape, skin texture, etc.) directly from the images.

3. Scalability

Once trained, deep learning models can analyze thousands of images quickly, making them ideal for large-scale applications like surveillance or social media analysis.

4. Real-Time Processing

With optimized architectures and hardware (like GPUs), age and gender detection can be performed in real time, useful for applications like smart advertising, video analytics, and security systems.

5. Adaptability

Deep learning models can be fine-tuned or retrained with new data, allowing them to adapt to different ethnicities, lighting conditions, or image qualities.

6. Integration with Other AI Tasks

These models can be integrated into larger AI systems — for example, in personalized marketing, human-computer interaction, or emotion recognition systems.

7. Non-Intrusive

Deep learning-based systems work with just images or video frames, without requiring personal input like surveys or manual data entry, ensuring a smoother user experience.

8. Continuous Improvement

With more training data and better architectures, the performance of deep learning models keeps improving over time.

6.DISADVANTAGES

1. **Data Dependency**
   Deep learning models require large amounts of high-quality, labeled data to perform well. Inaccurate or biased data can lead to poor or unfair results.
2. **Bias and Fairness Issues**
   Models may show biased predictions for certain age groups, genders, or ethnicities if the training data is not diverse or balanced.
3. **High Computational Cost**
   Training and running deep learning models often require powerful hardware like GPUs, which can be expensive and energy-intensive.
4. **Privacy Concerns**
   Using facial images for age and gender detection raises ethical and legal issues related to privacy, especially without user consent.
5. **Lack of Interpretability**
   Deep learning models are often considered "black boxes." It's hard to understand exactly how the model is making decisions, which is problematic in sensitive applications.
6. **Vulnerability to Image Quality**
   Performance can drop significantly with poor lighting, blurry images, or occluded faces (e.g., with masks, glasses, or hats).
7. **Overfitting Risk**
   If not properly trained or validated, the model might learn patterns specific to the training set and fail to generalize to new data.
8. **Maintenance and Updates**
   As demographics and image trends change (e.g., new fashion styles or age appearances), the model may need regular updates to remain accurate.

7.CONCLUSION

Age and gender detection using deep learning is a powerful and rapidly evolving technology with wide-ranging applications in areas like security, marketing, healthcare, and human-computer interaction. By leveraging deep learning models—especially convolutional neural networks (CNNs)—these systems can achieve high accuracy in identifying demographic features from facial images.

Although challenges such as bias, privacy concerns, and the need for large datasets still exist, ongoing research and advancements in model design and ethical AI are helping to overcome these limitations. With further development, age and gender detection systems will become more accurate, inclusive, and privacy-conscious, playing a key role in making future digital systems smarter, more adaptive, and user-friendly.

8.FUTURE SCOPE

Future models will achieve even greater accuracy by using more powerful architectures like Vision Transformers (ViTs), EfficientNet, and ensemble learning. These will handle complex real-world scenarios like occlusion, aging, makeup, and lighting variations more effectively. .

One major focus will be on creating models that perform well across different ethnicities, cultures, and age representations. This can be achieved using domain adaptation, transfer learning, and more diverse datasets.

Future systems will be lightweight and optimized to run directly on mobile devices, cameras, and edge devices. This enables applications in smart cities, transportation, and retail without needing cloud connectivity.

Combining facial features with voice, gait, or behavioral data will create multi-modal systems that enhance the accuracy of age and gender detection, especially in challenging conditions.

As concerns about privacy grow, future systems will focus on privacy-preserving techniques like federated learning and on-device processing. Users' data will remain safe while still enabling smart features.

Age and gender detection can be integrated with emotion detection and sentiment analysis to create more responsive and personalized systems in education, mental health, gaming, and customer service.

It can help build assistive tools for the visually impaired, providing audio feedback on the age and gender of people nearby, helping with navigation and social interaction.

The technology will become more common in public safety, helping identify missing persons or unauthorized access in real-time using demographic information.

Websites, apps, or digital displays will automatically adjust their interface, font size, language complexity, or tone based on the detected age group and gender — improving user experience (UX)

9.BIBLIOGRAPHY

1.Levi, G., & Hassner, T. (2015). Age and Gender Classification Using Convolutional Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, pp. 34–42.

2.Rothe, R., Timofte, R., & Van Gool, L. (2016). Deep expectation of real and apparent age from a single image without facial landmarks. International Journal of Computer Vision, 126(2–4), 144–157.

3.Han, H., Jain, A. K., & Wang, F. (2015). Age estimation from face images: Human vs. machine performance. In Proceedings of the International Conference on Biometrics (ICB), Phuket, Thailand.

4.Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). Apparent Age Estimation from Face Images Combining General and Children-Specialized Deep Learning Models. In Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA.

5.Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA.

6.Riccardo, C., Gaglio, S., & Lo Bosco, G. (2014). Gender and Age Group Recognition from Human Facial Images. International Journal of Computer Applications, 88(12), 22–29.

7.Dantcheva, A., Elia, P., & Ross, A. (2016). What else does your biometric data reveal? A survey on soft biometrics. IEEE Transactions on Information Forensics and Security, 11(3), 441–467.

8.Panwar, H., Juneja, M., & Sethi, M. (2020). Real-Time Age and Gender Prediction using Deep Learning on Live Video Feed. In Proceedings of the 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India.

10.HELP FILE

STEP-1: Go to start, search and launch ANACONDA NAVIGATOR.

STEP-2: After launching Anaconda Navigator, launch JUPYTER NOTEBOOK.

STEP-3: Open “main.ipynb” file to test and verify age and gender detection logic using sample images.

STEP-4: Import all the necessary packages and check whether any errors are present in the code or not.

STEP-5: Create a folder named PYTHON CODE on the DESKTOP and copy the project files into it.

STEP-6: Inside the templates folder, create the index.html file to design the home page of the project.

STEP-7: Also create the predict.html file to handle the image upload and result display functionality.

STEP-8: Create the real.html file which is used for real-time webcam-based age and gender detection.

STEP-9: Launch SPYDER from Anaconda Navigator.

STEP-10: After launching Spyder, open and run the file app.py from your project folder.

STEP-11: After running app.py, a URL will be created: “http://127.0.0.1:1100”

STEP-12: Copy the URL and paste it into the web browser.

STEP-13: The home page of the project will be displayed.

STEP-14: In the opened home page, click on the “Get Started” button. It will redirect you to the upload page.

STEP-15: Upload a face image, and after submitting, you will get the predicted age and gender.

STEP-16: You can also go to the real-time detection page by navigating to the “Webcam” option, where the system will access your webcam and predict age and gender in real-time.