DESIGN & DEVELOPMENT OF GEN-AGE DETECTION SYSTEM

MINOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

CSE-ARTIFICIAL INTELLIGENCE & DATA SCIENCE

By

GROUP NO. 2

DISHA VISHWAKARMA - 0187AD211013
YUVIKA SINHA - 0187AD211046

Under the guidance of

Dr. Vasima Khan

(Head of the Department)

JUNE-2024

Department of CSE-Artificial Intelligence & Data Science
Sagar Institute of Science & Technology (SISTec)
Bhopal (M.P.)

Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

Sagar Institute of Science & Technology (SISTec), Bhopal

Department of CSE-Artificial Intelligence & Data Science

Bhopal (M.P.)

June-2024

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Minor Project-II Report entitled **DESIGN & DEVELOPMEMT OF GEN-AGE DETECTION SYSTEM,** in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Technology*** in **CSE-*Artificial Intelligence & Data Science*** and submitted to the Department of Computer Science & Engineering, *Sagar Institute of Science & Technology (SISTec)***,** Bhopal (M.P.) is an authentic record of my own work carried out during the period from jan-2024 to June-2024 under the supervision of **Dr. Vasima Khan.**

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Disha Vishwakarma
0187AD211013

Yuvika Sinha
0187AD211046

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vasima Khan
Project Guide

Dr. Vasima Khan
HOD

Dr. D.K. Rajoriya
Principal

ACKNOWLEGMENT

We take this occasion to thank God, almighty for blessing us with his grace and taking our endeavor to a successful culmination. We extend our sincere and heartfelt thanks to our esteemed Guide, **Dr. Vasima Khan** for providing us with the right guidance and advice at the crucial junctures and for showing us the right way.

We would like to express our sincere thanks to **Dr. D.K. Rajoriya, Principal, SISTec**, Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We would like to express our sincere thanks to **Dr. Swati Saxena, Vice Principal, SISTec**, Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Vasima Khan (HOD**) of Department of CSE-Artificial Intelligence & Data Science for his kindhearted support. I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in problems. Our all faculties encourage me and due to their kindness and helpful nature and help I got very much confidence to complete this project.

I am deeply inherited who devoted his precious time in giving me the information about the various aspect and gave support and guidance at every point of time. I am thankful to their kind and supportive nature. Her inspiring nature has always made my work easy.

ABSTRACT

The accurate detection of gender and age from images plays a crucial role in various applications, including marketing, security, and personalized user experiences. In this project, we propose a novel approach utilizing deep learning techniques integrated with the Flask web framework to achieve real-time gender and age detection from facial images. The system aims to provide a robust and efficient solution for automatic demographic analysis, facilitating targeted advertising, content recommendation, and security applications.

The methodology of the project involves the development and training of a convolutional neural network (CNN) model using a large-scale dataset of facial images. The CNN architecture is designed to extract high-level features from facial attributes, enabling the model to accurately classify gender and estimate age groups. To enhance model performance, extensive preprocessing techniques are applied to the dataset, including image normalization, augmentation, and feature extraction.

The implementation phase involves the integration of the trained deep learning model with the Flask web framework to create a user-friendly and accessible web application. Through the Flask interface, users can upload images or provide URLs for analysis, after which the system performs gender and age prediction in real-time. The seamless integration of Flask enables the deployment of the system on web servers, allowing users to access the functionality from various devices and platforms.

Experimental results demonstrate the effectiveness and reliability of the proposed system in gender and age detection tasks. The deep learning model achieves high accuracy rates in classifying gender and accurately estimating age groups, even in the presence of variations in facial expressions, lighting conditions, and image quality. Performance evaluation metrics such as accuracy, precision, recall, and F1-score validate the robustness of the system across different demographic categories and image scenarios.

The discussion section analyzes the strengths and limitations of the proposed approach, highlighting potential areas for future research and system improvements. Despite the promising results, challenges such as dataset bias, model generalization, and computational efficiency are addressed, paving the way for further advancements in gender and age detection technology.

In conclusion, the project presents a comprehensive framework for gender and age detection using deep learning and Flask, offering practical solutions for demographic analysis in various domains. The integration of advanced machine learning techniques with web technologies enables the development of scalable and accessible systems with broad applications in marketing, security, and personalized services.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
CNN	Convolutional Neural Network
AI	Artificial Intelligence
DL	Deep Learning
IDE	Integrated Development Environment
VS Code	Visual Studio Code
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
UML	Unified Modeling Language
RAM	Random Access Memory

LIST OF FIGURES

TABLE OF CONTENTS

TITLE **PAGE NO.**

CHAPTER-1 INTRODUCTION

CHAPTER-1
INTRODUCTION

ABOUT PROJECT:

The Age and Gender Detection project utilizes deep learning techniques to predict individuals' age and gender accurately from images. It addresses the growing need for personalized user experiences in various domains like marketing and security. By employing convolutional neural networks (CNNs), the project aims to achieve high accuracy and robustness in demographic prediction, making it accessible via a user-friendly web interface powered by Flask.

The project emphasizes visualization and interactivity, democratizing access to advanced machine learning capabilities. Educational resources and documentation accompany the system, fostering understanding of complex algorithms and ethical considerations in AI development. Ethical concerns are paramount, with measures implemented to mitigate biases and promote fairness in prediction algorithms.

As a showcase of technical proficiency, the project contributes to the advancement of technology and inspires future AI researchers. Its code, resources, and insights are shared with the community to encourage collaboration and innovation. In summary, the Age and Gender Detection project represents an innovative exploration of deep learning, image processing, and web deployment, aiming to develop an effective and ethical demographic prediction system while fostering understanding, accessibility, and innovation in artificial intelligence.

PURPOSE:

Developing an innovative deep learning system to accurately predict age and gender from images, fostering understanding of AI and web deployment techniques, and inspiring curiosity in the intersection of technology and demographics.

PROJECT OBJECTIVE:

Following objectives should be achieved:

- Develop a deep learning model capable of accurately predicting age and gender from input images, leveraging techniques such as convolutional neural networks (CNNs) and transfer learning.
- Implement a user-friendly web interface using Flask for easy access and interaction with the trained model, facilitating intuitive visualization and exploration of demographic predictions.
- Address ethical considerations by mitigating biases and promoting fairness in age and gender prediction algorithms, while also providing educational resources to foster understanding of AI principles and ethical AI development practices.

INTERFACE:

- User Interfaces:

  Browser chrome latest version (supports HTML).

- Hardware Interface:

  Windows

CHAPTER-2
HARDWARE & SOFTWARE REQUIREMENT

CHAPTER-2
HARDWARE & SOFTWARE REQUIREMENT

2.1 SOFTWARE REQUIREMENTS

2.1.1 FOR DEVELOPERS:

2.1.1.1 IDE –Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS .Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.

2.1.1.2 Programming Language

- **HTML**

The **Hyper Text Markup Language**, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- **CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

2.1.1.3 Web Browser- Google Chrome (for testing purpose)

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. It was later ported to Linux, macOS, iOS, and Android, where it is the default browser. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.

2.1.1.4 Flask Framework

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

2.1.2 FOR END-USERS:

- Web Browser like Google chrome (Latest Version) or Microsoft Edge (Latest Version)
- Internet Connectivity

2.2 HARDWARE REQUIREMENT

- Processor: Dual-core 64-bit processor.
- RAM: 12 GB.
- Hard Disk: 8GB.

CHAPTER-3
PROBLEM DESCRIPTION

CHAPTER-3
PROBLEM DESCRIPTION

3.1 OVERVIEW

- The need to accurately predict age and gender from images arises in various domains such as marketing, e-commerce, and security, where personalized user experiences and targeted content delivery are crucial.
- Existing methods for age and gender prediction may lack accuracy and robustness, leading to potential biases and ethical concerns in demographic predictions..
- By leveraging deep learning techniques, the project aims to address these challenges by developing a reliable and ethical age and gender detection system, accessible via a user-friendly web interface, to facilitate intuitive exploration and understanding of demographic predictions.
- The accurate prediction of age and gender from images is pivotal in numerous applications, including targeted advertising, content recommendation, and security systems. However, traditional methods often suffer from limited accuracy and robustness, leading to potential biases and ethical implications. These shortcomings underscore the need for a reliable and ethical age and gender detection system that leverages advanced deep learning techniques. By developing such a system and providing a user-friendly web interface, the project aims to empower users to explore demographic predictions intuitively while promoting transparency, fairness, and understanding in AI development.

CHAPTER-4
LITERATURE SURVEY

CHAPTER-4
LITERATURE SURVEY

A gap between the way in which algorithms are taught and the students' learning style is one of the main reasons why introductory programming courses are perceived as hard. Namely, while most people are visual types, teaching, to a great extent, is verbal. Visual learning is a type of learning where people prefer using images, pictures, colors, and maps to organize information in order to comprehend complex concepts. This is the main reason why algorithm visualization tools are implemented in learning process.

Many learners find Data Structures one of the toughest topics to study yet it is one of the most important one. Conventional way of learning the topic might be good, but the visualization method is found out to be more interesting and effective. Being technical students, we tried to solve the conventional problem in technical way. We went through many websites who had worked in the same problem and gathered the essential idea and layout behind their work.

Taking those essentials as our base we developed our website which is easy to use and provide colorful way to understand the algorithms.

There are several animation tools based on interpreting source code, such as Zstep95. However, without precognition related to system expressions and their contribution to the animation, user cannot make the decision whether to use the details (animation controls) or an expression. On the other hand, if all elements are enabled and shown during animation, some key parts needed for the understanding of algorithm could be overlooked.

CHAPTER-5 SOFTWARE REQUIREMENTS SPECIFICATION

CHAPTER-5
SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENT

Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs.

Functional requirements usually define if/then behaviours and include calculations, data input, and business processes.

There are two actors in this project:

- ➢ Developer
- ➢ User

❖ **Developer:**

- **Update Feature :** The developer can update or modify the website according to the needs of the learners.

❖ **User :**

- **Input Data :** The user provides input data to the system. This could be in the form of facial images or videos containing human faces.
- **Data Submission :** The user submits the input data to the system for processing. This could be done through a web-based interface where users upload images or videos, or through an application programming interface (API) if the system is integrated into other software.
- **Prediction :** The age and gender detection model predicts the age and gender of the individuals present in the input data. The model output could include the estimated age

range (e.g., "20-30 years old") and the predicted gender (e.g., "male" or "female") for each detected face.

- **Display Results :** The system displays the prediction results to the user. This could be in the form of visualizations overlaid on the input images or videos, text-based summaries, or interactive dashboards. The user can view the predicted age and gender labels for each detected face and potentially navigate through different frames or images if processing a video.

5.2 NON-FUNCTIONAL REQUIREMENT

Non-functional requirements are essential characteristics or qualities of a system that contribute to its overall performance, usability, security, and other important aspects beyond its core functionality. These requirements focus on how well the system performs its intended functions rather than what those functions are.

In the context of an age and gender detection project, where the primary goal is to accurately predict the age and gender of individuals from images or videos, non-functional requirements play a critical role in shaping the user experience and ensuring the system's effectiveness. Here's a breakdown of some key non-functional requirements for such a project:

1. **Performance :**

- The system should be able to process images or videos for age and gender detection within a reasonable time frame, ensuring timely responses to user requests.
- Response times for age and gender prediction should be optimized to provide fast and efficient results, even under heavy load condition.

2. **Accuracy :**

- The age and gender detection model should achieve high accuracy in predicting age ranges and gender labels from facial images or videos, minimizing false positives and false negatives.
- The system should undergo rigorous testing and validation to ensure reliable and consistent performance across diverse datasets and real-world scenarios.

3. **Scalability :**

- The system should be designed to handle varying levels of user traffic and data volume, scaling resources dynamically to accommodate increasing demands without sacrificing performance or reliability.
- Scalability considerations should extend to both the model training and inference processes, as well as the web interface and backend infrastructure.

4. **Security :**

- The system should implement robust security measures to protect user data, ensuring confidentiality, integrity, and availability.
- Access to sensitive information such as user inputs and prediction results should be restricted to authorized personnel, with proper authentication and authorization mechanisms in place.

5. **Usability :**

- The user interface should be intuitive, user-friendly, and accessible, catering to users with varying levels of technical expertise.
- The system should provide clear instructions, feedback, and error messages to guide users through the age and gender detection process and facilitate a positive user experience.

6. Reliability :

- The system should be highly reliable and resilient, minimizing downtime and service disruptions through redundancy, fault tolerance, and disaster recovery mechanisms.
- Error handling and recovery strategies should be implemented to gracefully handle unexpected failures and ensure continuous operation of critical functionalities.

7. Compatibility:

- The system should be compatible with a wide range of devices, browsers, and operating systems, ensuring seamless access and functionality for users across different platforms.
- Compatibility with existing software systems and APIs should be considered to facilitate integration and interoperability with other applications and services.

CHAPTER-6 SOFTWARE DESIGN

CHAPTER-6
SOFTWARE DESIGN

The gender and age detection system begins its operation with the capture of an image, a pivotal step that sets the groundwork for subsequent analysis. This raw image undergoes preprocessing procedures aimed at refining its quality and preparing it for comprehensive scrutiny. Under the careful supervision of an administrator, the system delves into a phase of rigorous training. This training regimen is vital, as it instills the system with the requisite capabilities for precise detection and extraction of pertinent features essential for age and gender prediction. Core functionalities, including the discernment of facial structures within the image and the extraction of relevant facial features, are meticulously executed. These extracted features serve as the foundation upon which the trained system operates, leveraging its acquired knowledge to accurately predict the age and gender of the subject depicted in the image. Upon completion of these analytical processes, the final outcome is presented to the user, encapsulating the essence of the system's endeavor. Through this intricate choreography of image processing, training, and prediction, the system navigates its way toward furnishing valuable insights to its users, ultimately culminating in the display of results for their perusal and assessment.

6.1 USE CASE DIAGRAM

The use case diagram provides a visual representation of the system's functionalities and interactions with external actors such as users and administrators. It visually outlines the sequence of actions, from image capture to result presentation, encompassing key processes like preprocessing, training, and prediction.

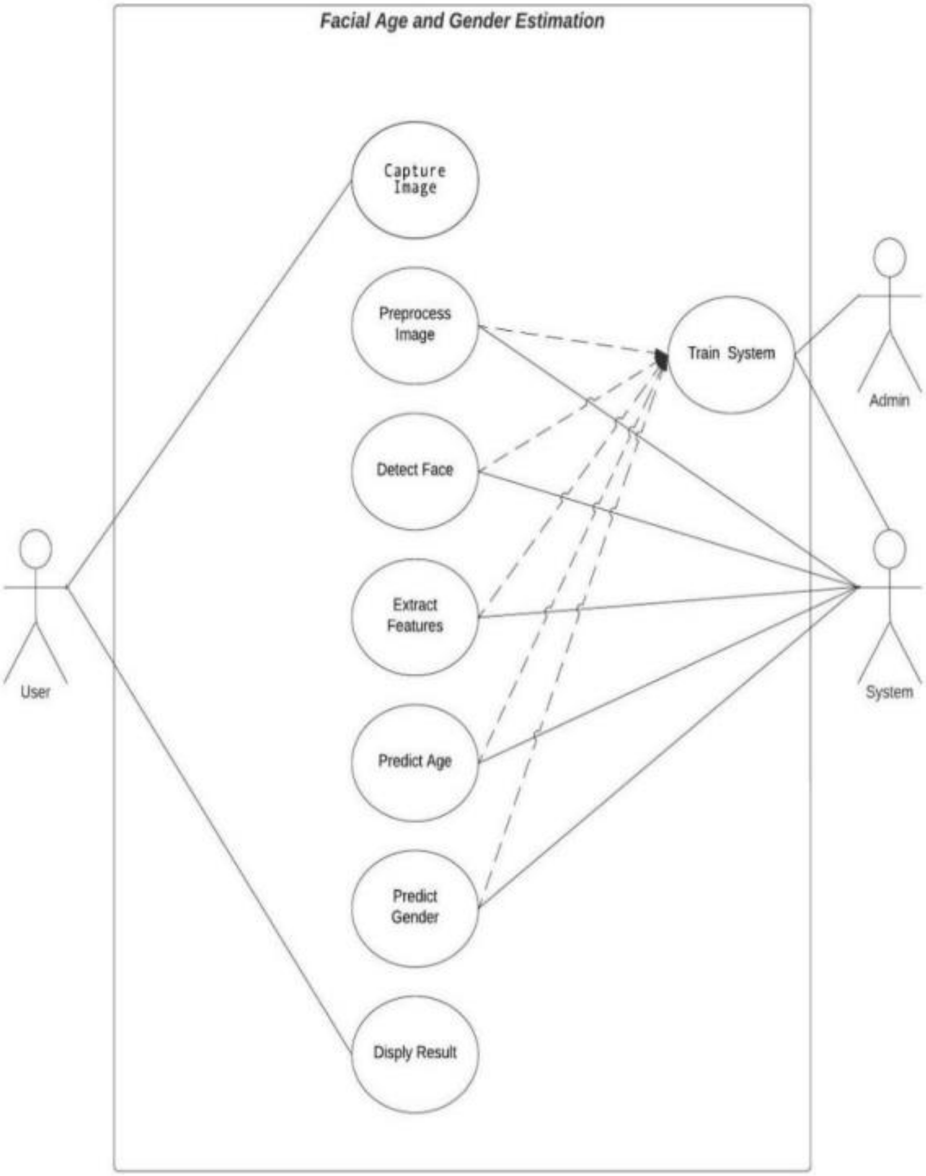


Fig. 6.1.1 Use Case Diagram for Gender & Age Detection System

6.2 CLASS DIAGRAM

The class diagram illustrates the static structure of the system, showcasing the classes, attributes, methods, and their relationships. It offers insights into the organization of system components, including image processors, trainers, detectors, predictors, and display modules, aiding in understanding the system's architecture.

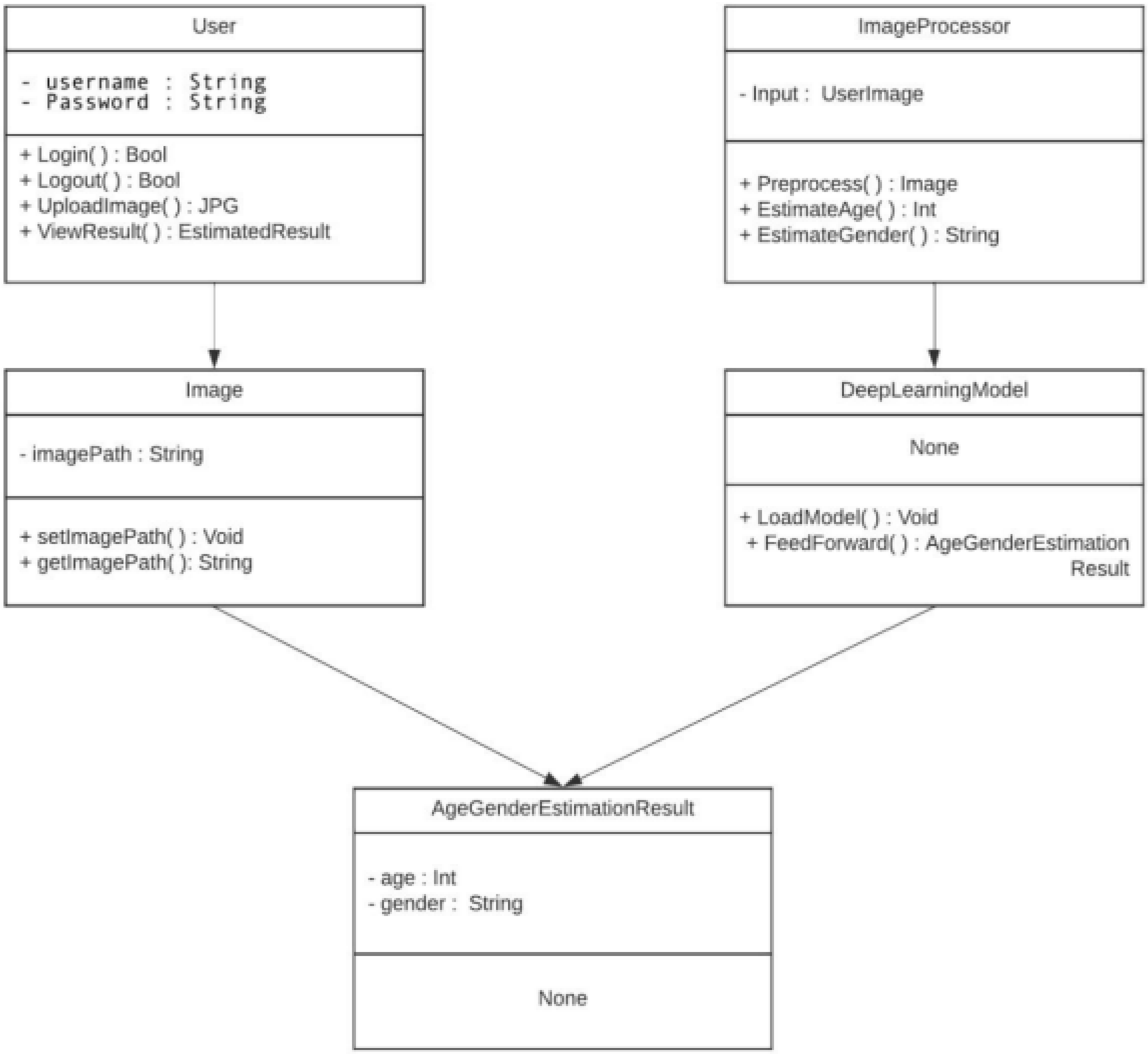


Fig. 6.2.1 Class Diagram for Gender & Age Detection System

6.3 FLOW DIAGRAM

The data flow diagram elucidates the flow of data within the system, depicting processes, data stores, data flows, and external entities. It highlights the path of data from image capture through preprocessing, training, prediction, and result display, facilitating comprehension of information flow and system functionality.

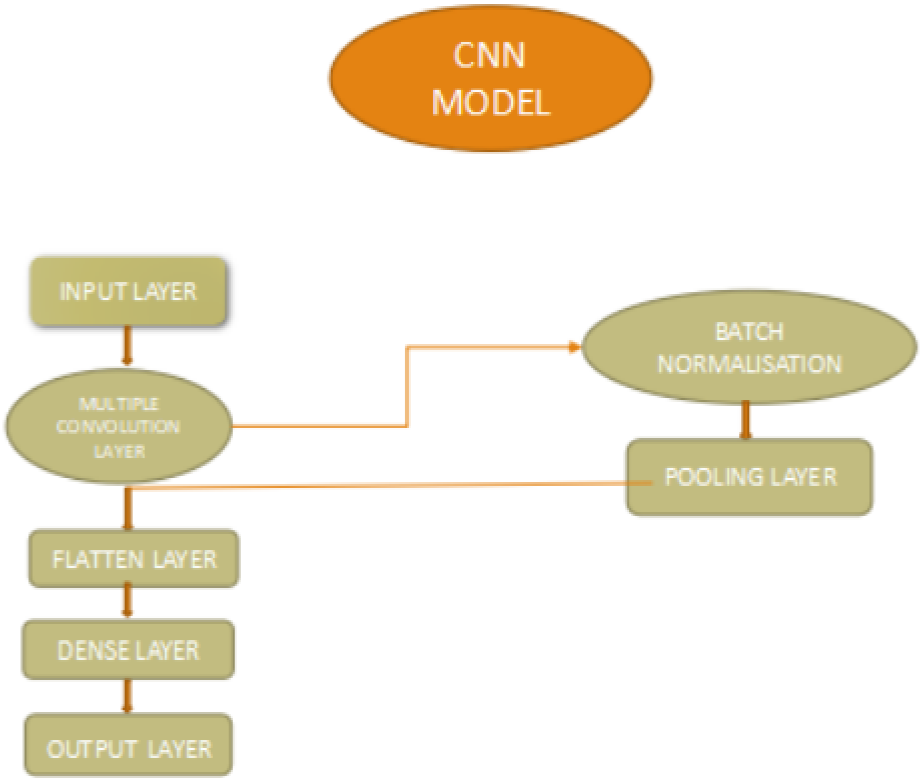


Fig. 6.3.1 Flow Diagram for Gender & Age Detection System

CHAPTER-7
PROJECT CODE

CHAPTER-6
PROJECT CODE

7.1 MODEL (model.py)

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import cv2
from keras.models import Sequential, Model
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten, Input,
BatchNormalization
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split

# Load the data
path = r'C:\Users\Disha\Desktop\Design & Development of GenAge Detection System\Design &
Development of GenAge Detection System\crop_part1'
images = []
age = []
gender = []
target_size = (100, 100)  # Choose a smaller size

for img_file in os.listdir(path):
    ages = int(img_file.split("_")[0])
    genders = int(img_file.split("_")[1])
    img = cv2.imread(os.path.join(path, img_file))

    # Check for corrupt images
    if img is None:
        print(f"Corrupt image: {img_file}")
        continue

    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = cv2.resize(img, target_size)
    images.append(np.array(img))
    age.append(ages)
    gender.append(genders)

# Convert lists to numpy arrays
age = np.array(age, dtype=np.int64)
images = np.array(images) / 255.0  # Scale images
gender = np.array(gender, np.uint64)
```

```
# Split the data into training and testing sets
x_train, x_test, y_train_age, y_test_age, y_train_gender, y_test_gender = train_test_split(images,
age, gender, random_state=42)

# Data Augmentation
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True
)
datagen.fit(x_train)

# Define the shared convolutional base
input_shape = (100, 100, 3)
input_layer = Input(shape=input_shape)
conv1 = Conv2D(128, kernel_size=3, activation='relu')(input_layer)
conv1_bn = BatchNormalization()(conv1)
pool1 = MaxPooling2D(pool_size=3, strides=2)(conv1_bn)
conv2 = Conv2D(128, kernel_size=3, activation='relu')(pool1)
conv2_bn = BatchNormalization()(conv2)
pool2 = MaxPooling2D(pool_size=3, strides=2)(conv2_bn)
conv3 = Conv2D(256, kernel_size=3, activation='relu')(pool2)
conv3_bn = BatchNormalization()(conv3)
pool3 = MaxPooling2D(pool_size=3, strides=2)(conv3_bn)
conv4 = Conv2D(512, kernel_size=3, activation='relu')(pool3)
conv4_bn = BatchNormalization()(conv4)
pool4 = MaxPooling2D(pool_size=3, strides=2)(conv4_bn)
flatten = Flatten()(pool4)
dropout = Dropout(0.5)(flatten)
dense1 = Dense(512, activation='relu')(dropout)

# Define separate output layers for age and gender
output_age = Dense(1, activation='linear', name='age')(dense1)
output_gender = Dense(1, activation='sigmoid', name='gender')(dense1)

# Combine input and output layers into a single model
model = Model(inputs=input_layer, outputs=[output_age, output_gender])

# Compile the model with appropriate losses and metrics
optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,
            loss={'age': 'mse', 'gender': 'binary_crossentropy'},
            metrics={'age': 'mae', 'gender': 'accuracy'})

print(model.summary()
# Train the model
history = model.fit(datagen.flow(x_train, {'age': y_train_age, 'gender': y_train_gender},
batch_size=32),
                    steps_per_epoch=len(x_train) // 32,  # specify steps_per_epoch
                    validation_data=(x_test, {'age': y_test_age, 'gender': y_test_gender}),
                    epochs=5)



# Plot accuracy for age
plt.plot(history.history['age_mae'], label='Training Age MAE')
plt.plot(history.history['val_age_mae'], label='Validation Age MAE')
plt.xlabel('Epochs')
plt.ylabel('Mean Absolute Error')
plt.title('Training and Validation Age MAE')
plt.legend()
plt.savefig('age_accuracy.png')
plt.show()

# Plot accuracy for gender
plt.plot(history.history['gender_accuracy'], label='Training Gender Accuracy')
plt.plot(history.history['val_gender_accuracy'], label='Validation Gender Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training and Validation Gender Accuracy')
plt.legend()
plt.savefig('gender_accuracy.png')
plt.show()

# Save the model
model.save('age_gender_model.h5')
```

7.2FLASK INTEGRATION (app.py)

```
from flask import Flask, render_template, Response, request
from keras.models import load_model
from keras.preprocessing.image import img_to_array
import cv2
import numpy as np

app = Flask(__name__)

# Load age and gender detection models
age_model = load_model('age_gender_model.h5')
gender_model = load_model('age_gender_model.h5')

# Function to preprocess image for age and gender prediction
```

```
def preprocess_image(image):
    image = cv2.resize(image, (100, 100))  # Resize image to match model input size
    image = image.astype('float') / 255.0  # Normalize pixel values
    image = np.expand_dims(image, axis=0)  # Add batch dimension
    return image

# Function to perform age and gender prediction
def predict_age_and_gender(face_image):
    # Preprocess the face image
    processed_image = preprocess_image(face_image)

    # Predict age
    age_prediction = age_model.predict(processed_image)[0][0]
    predicted_age = int(round(age_prediction.item()))


    # Predict gender
    gender_prediction = gender_model.predict(processed_image)[0][0]
    predicted_gender = 'Male' if gender_prediction >= 0.5 else 'Female'

    return predicted_age, predicted_gender

# Function to detect faces and perform age and gender prediction
def detect_faces_and_predict(frame):
    # Convert frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces
    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30,
30))

    # Perform age and gender prediction for each detected face
    for (x, y, w, h) in faces:
        face_image = frame[y:y+h, x:x+w]  # Extract face region
        age, gender = predict_age_and_gender(face_image)

        # Draw rectangle around face
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

        # Display predicted age and gender
        label = f'Age: {age}, Gender: {gender}'
        cv2.putText(frame, label, (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,255,12), 2)

    return frame

# Function to capture video from webcam and perform age and gender detectio
```

```
def video_feed():
    cap = cv2.VideoCapture(0)

    while True:
        ret, frame = cap.read()

        # Perform face detection and age/gender prediction
        frame_with_predictions = detect_faces_and_predict(frame)

        # Convert frame to JPEG format
        ret, jpeg = cv2.imencode('.jpg', frame_with_predictions)

        # Yield the frame in HTTP response
        yield (b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n')

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/webcam')
def webcam():
    return render_template('webcam.html')

@app.route('/live')
def live_feed():
    return Response(video_feed(), mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run(debug=True)
```

7.3FRONT-END (HTML)

7.3.1 Index.html

```
<!DOCTYPE html>
<html>
<title>Detect Age & Gender</title>
```

```
<link rel = "icon" href =
"https://cdn.pixabay.com/photo/2019/06/23/05/32/deer-head-
4292868_1280.png" type = "image/x-icon">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="../static/styles/style.css">
<link rel="stylesheet" href="../static/styles/aj.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<body>

<!-- First Parallax Image with Logo Text -->
<div class="bgimg aj-display-container aj-opacity-min" id="home">
  <div class="aj-display-middle" style="white-space:nowrap;">
    <span class="aj-center aj-padding-large aj-black aj-xlarge aj-wide aj-
animate-opacity">Gender<span class="aj-hide-small"> Recognization</span>
& Age Prediction</span>
  </div>
</div>
<br><br><br>
<div class="action-wrap">
  <div class="action-html">
    <input id="tab-1" type="radio" name="tab" class="live" checked><label
for="tab-1" class="tab">Live</label>
    <input id="tab-2" type="radio" name="tab" class="photo"><label for="tab-
2" class="tab">Photo</label>
    <div class="action-form">
      <div class="live-htm">
        <div class="group">
          <br><br>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
          <img src="https://cdn.pixabay.com/photo/2021/03/23/09/01/webcam-
6116845_1280.png" height="120" width="120">
          <br><br>
          <a href='/webcam'><input type="submit" class="button" value="Go live
to detect"></a>
```

```
        </div>
      </div>
      <div class="photo-htm">
        <form action="/upload" method="POST" enctype="multipart/form-data"
style="color: #aaa">
          <br><br>Select image to upload:<br><br><br>
          <input type="file" name="fileToUpload" id="fileToUpload">
          <br><br><br><br>
        <div class="group">
          <input type="submit" class="button" value="Upload and detect"
name="submit">
        </div>
        </form>
      </div>
    </div>
  </div>
</div>

<!-- Container (About Section) -->
<div class="aj-content aj-container aj-padding-64" id="about">
  <h3 class="aj-center">ABOUT THE PROJECT</h3>
  <p class="aj-center"><em>ML Project</em></p>
  <p>We will use Deep Learning to accurately identify the gender and age of a
person from a single image of a face.</p>
  </div>



<!-- Footer -->
<footer class="aj-center aj-black aj-padding-64 aj-opacity aj-hover-opacity-
off">
  <a href="#home" class="aj-button aj-light-grey"><i class="fa fa-arrow-up aj-
margin-right"></i>To the top</a>
</footer>

<script>
// Modal Image Gallery
function onClick(element) {
  document.getElementById("img01").src = element.src;
  document.getElementById("modal01").style.display = "block";
  var captionText = document.getElementById("caption");
  captionText.innerHTML = element.alt
}

</script>

</body>
</html>
```

7.3.2 photo.html

```
<!doctype html>
<html lang="en">
<head>
   <link rel = "icon" href = "https://cdn.pixabay.com/photo/2021/03/23/09/01/webcam-6116845_1280.png" type = "image/x-icon">
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
   <link rel="stylesheet" href="static/css/style.css">

   <title>Photo Detection</title>
   <style>
      html {
         background: #100a1c;
         background-image:
            radial-gradient(50% 30% ellipse at center top, #201e40 0%, rgba(0,0,0,0) 100%),
            radial-gradient(60% 50% ellipse at center bottom, #261226 0%, #100a1c 100%);
         background-attachment: fixed;
         color: #6cacc5;
      }
      img {
         margin-left: 110px;
         padding: 0;
         width: 80vw;
         height: 80vh;
         border: 2px solid #6cacc5;
         border-radius: 4px;
      }
      button {
         float: left;
      }
      h2 {
         text-align: center;
         font-family: 'EB Garamond', serif;
      /*color: #ff1a1a;*/
         text-shadow: 2px 2px 4px #000000;
```

```
        }
        /* --- STYLING THE BUTTONS --- */
        button {
            border: 0;
            background: rgba(42,50,113, .28);
            color: #6cacc5;
            cursor: pointer;
            font: inherit;
            margin: 0.25em;
            transition: all 0.5s;
            border-radius: 4px;
        }
        /* --- WHEN THE CURSOR HOVERS OVER THE BUTTONS THE COLOR
CHNAGES --- */
        button:hover {
            background: #201e40;
        }
    </style>
</head>
<body>
    <div>
        <div class="header">
            <a href="/"><button>Home</button></a>
            <h2>Detecting Age and Gender from Photo</h2>
        </div>

        <div class="container">
            <img src="{{ url_for('upload_file') }}" width="100%">
        </div>
    </div>
</body>
</html>
```

7.3.3 Webcam.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="icon" href="https://cdn.pixabay.com/photo/2021/03/23/09/01/webcam-
6116845_1280.png" type="image/x-icon">
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

    <title>Live Detection</title>
    <style>
```

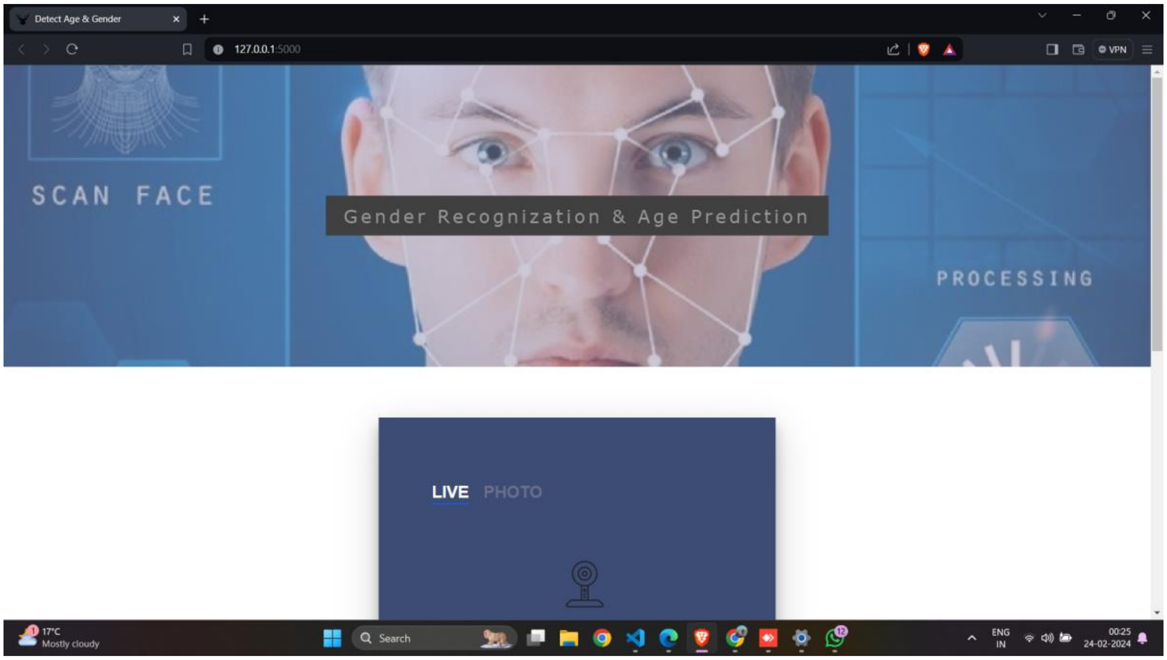
```
html {
    background: #100a1c;
    background-image:
        radial-gradient(50% 30% ellipse at center top, #201e40 0%, rgba(0,0,0,0) 100%),
        radial-gradient(60% 50% ellipse at center bottom, #261226 0%, #100a1c 100%);
    background-attachment: fixed;
    color: #6cacc5;
    overflow: hidden;
}

body {
    margin: 0;
    padding: 0;
}

.container {
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

img {
    max-width: 100%;
    max-height: 100%;
    border: 2px solid #6cacc5;
    border-radius: 4px;
}

button {
    float: left;
    border: 0;
    background: rgba(42, 50, 113, .28);
    color: #6cacc5;
    cursor: pointer;
    font: inherit;
    margin: 0.25em;
    transition: all 0.5s;
    border-radius: 4px;
}

button:hover {
    background: #201e40;
}

h2 {
    text-align: center;
```

```
            font-family: 'EB Garamond', serif;

            text-shadow: 2px 2px 4px #000000;
        }
    </style>
</head>
<body>
    <div>
        <div class="header">
            <a href="/"><button>Home</button></a>
            <h2>Detecting Age and Gender</h2>
        </div>

        <div class="container">
            <img src="{{ url_for('live_feed') }}">
        </div>
    </div>
</body>
</html>
```

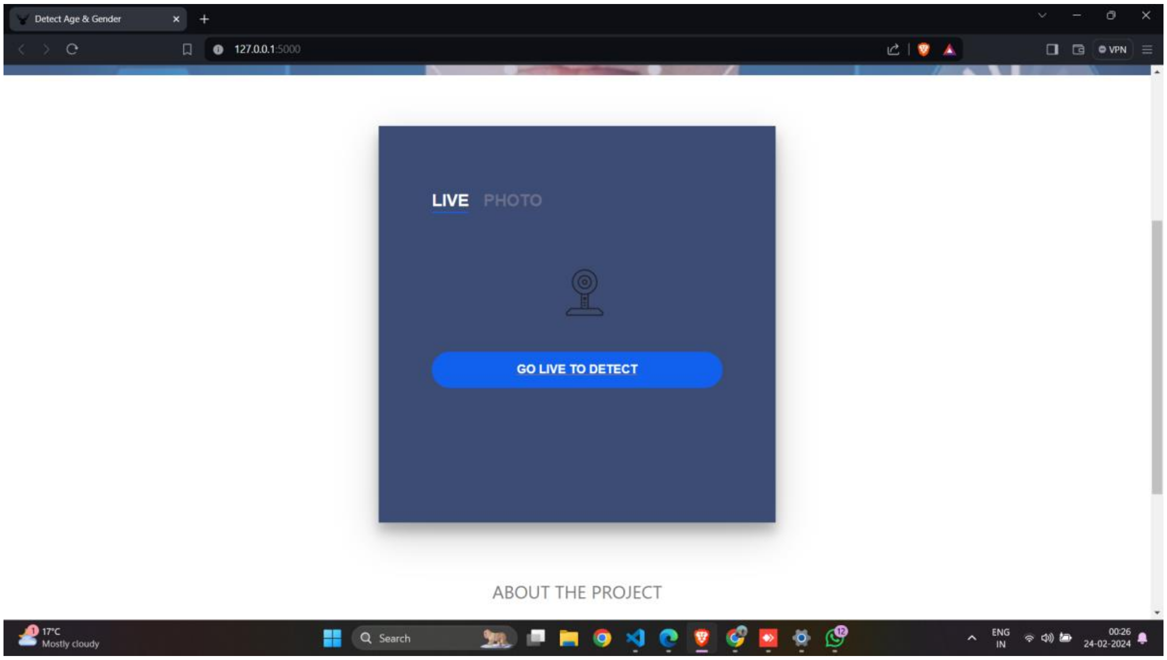
CHAPTER-8
OUTPUT
SCREEN

CHAPTER-8
OUTPUT SCREEN

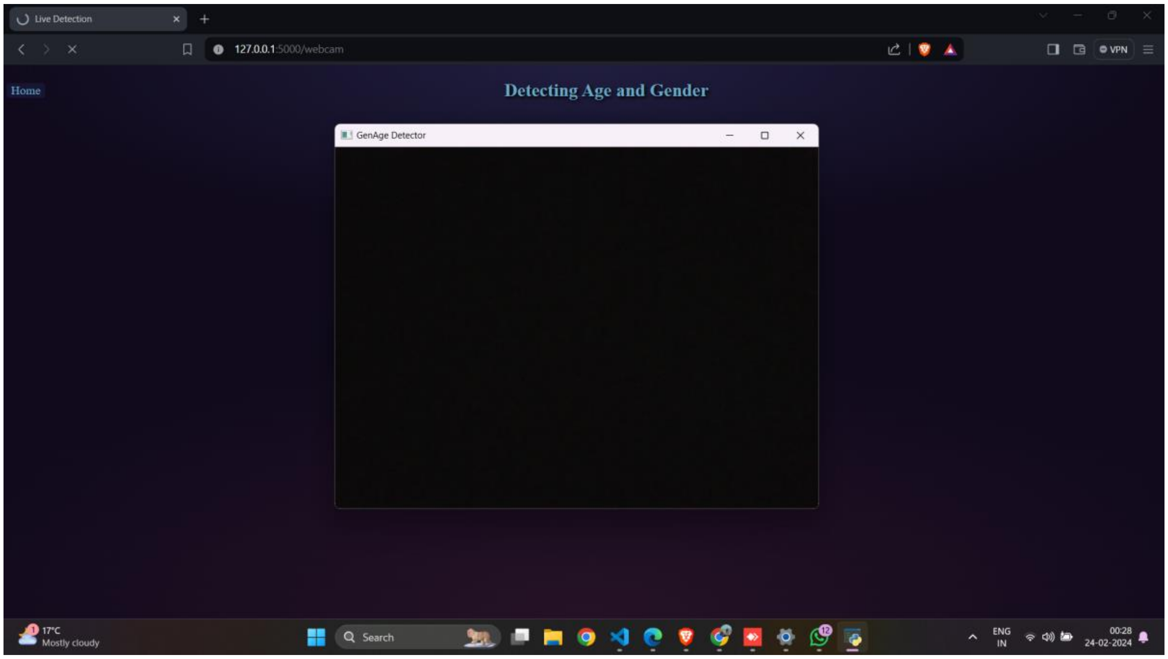
8.1 HOME PAGE:



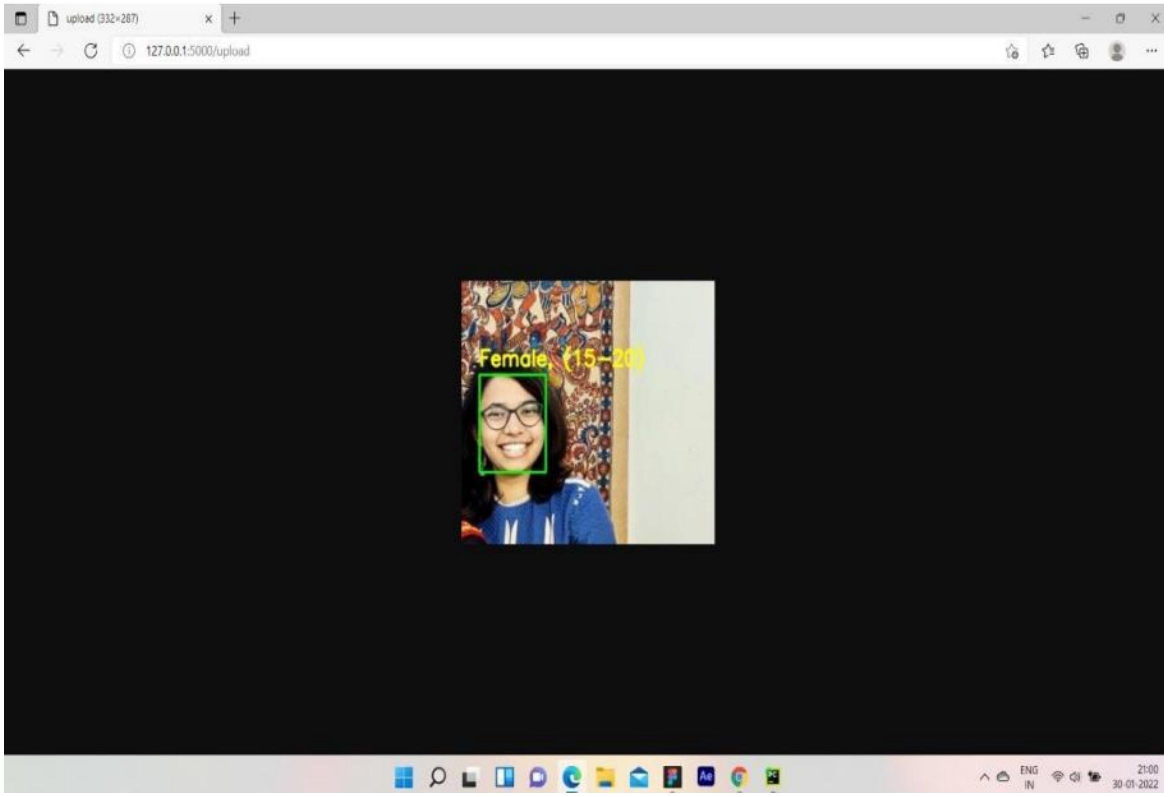
8.2 SELECTING PAGE:



8.3 DETECTING PAGE:



8.4 RESULT PAGE:



CHAPTER-9
DEPLOYMENT

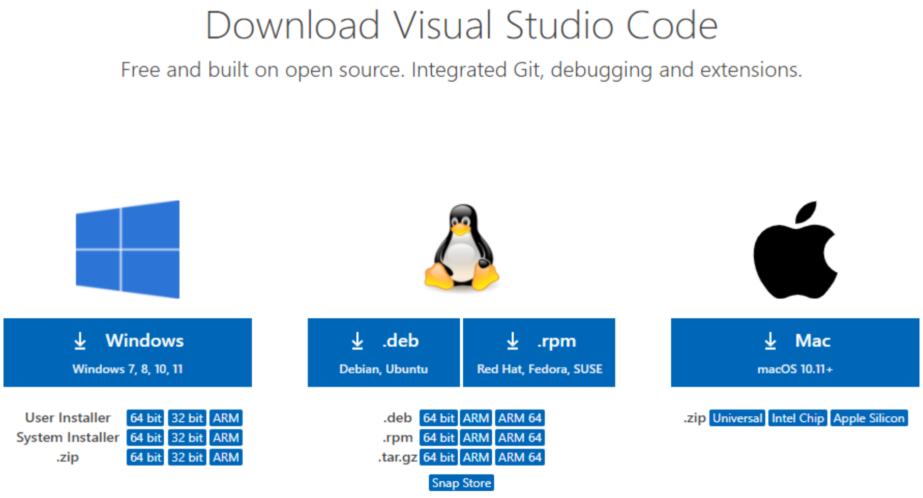
CHAPTER-9
DEPLOYMENT

10.1 INTRODUCTION

This chapter describes how to deploy the project on a fresh machine. It includes Installation steps & snapshots of pre-required software like Visual Studio Code & application developed under the major project**.** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

10.2 INSTALLATION OF VS CODE

You will need to install VS Code on your Windows computer.

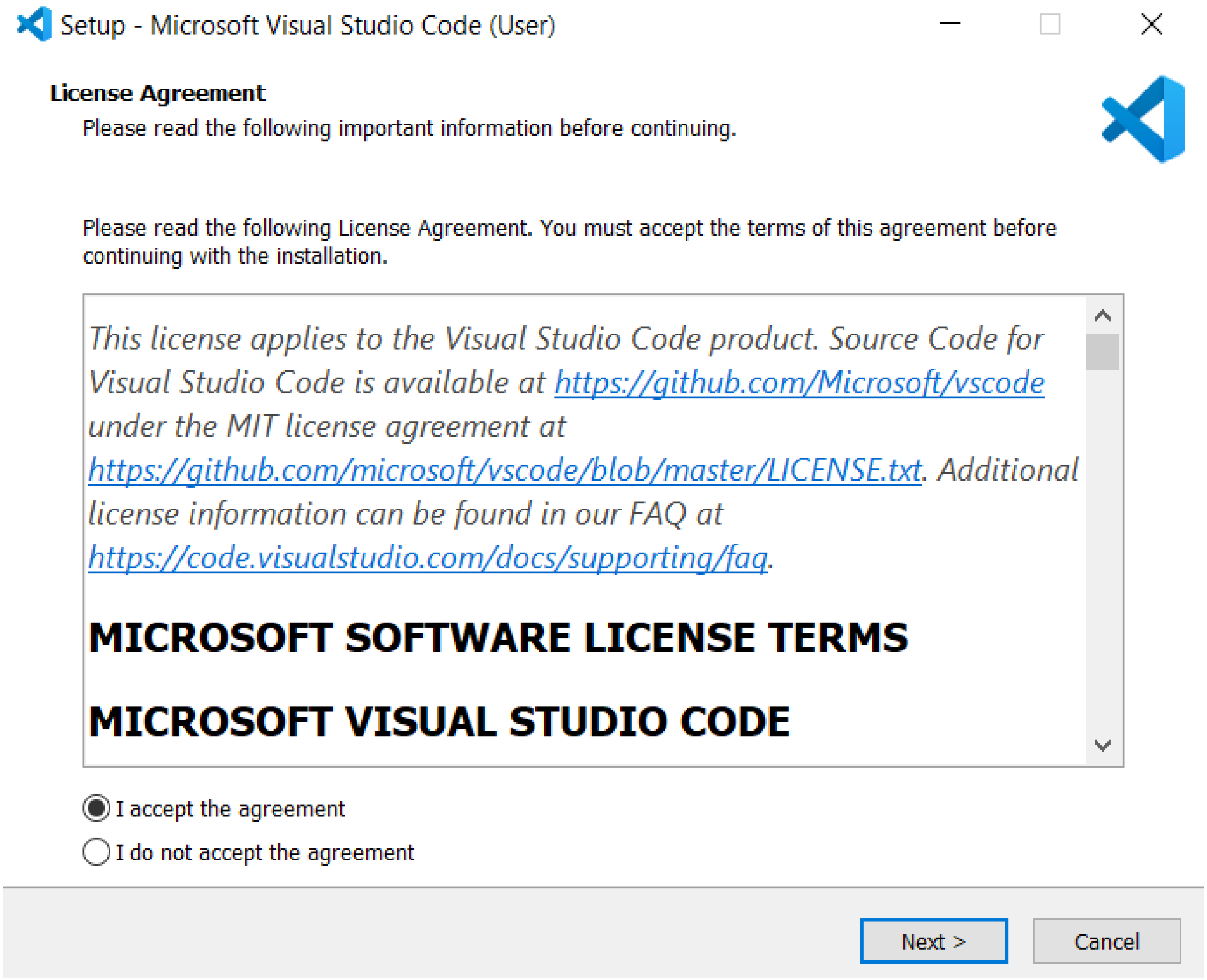


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Fig. 9.2.1 Download VS Code

- Click the option Download.
- Double Click the Downloaded file
- Now a dialog box appears
- Select I accept the agreement
- Then select Next



License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/master/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.

MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL STUDIO CODE

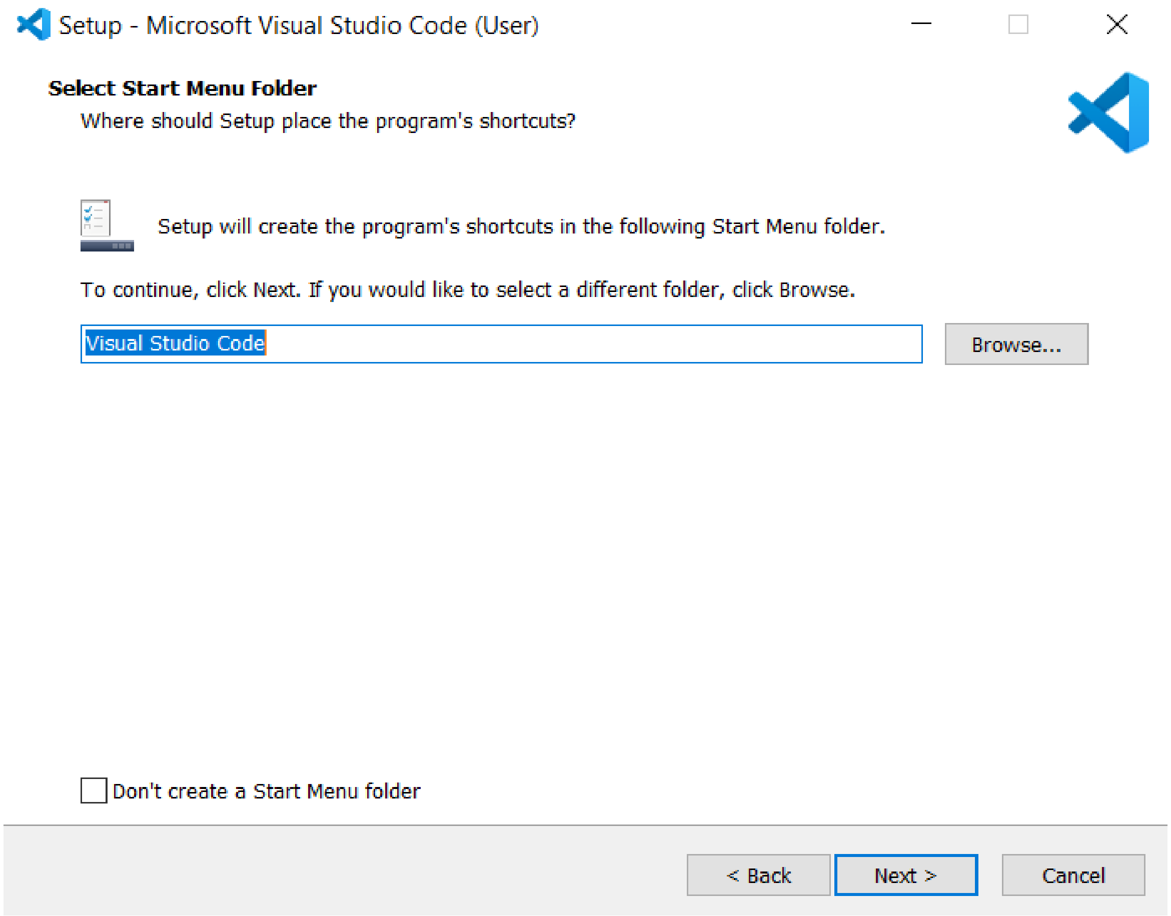
● I accept the agreement

○ I do not accept the agreement

Fig. 9.2.2 License Agreement of VS Code

- Select a folder by clicking Browse or just follow the default path.

- Then select Next



Setup - Microsoft Visual Studio Code (User)

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

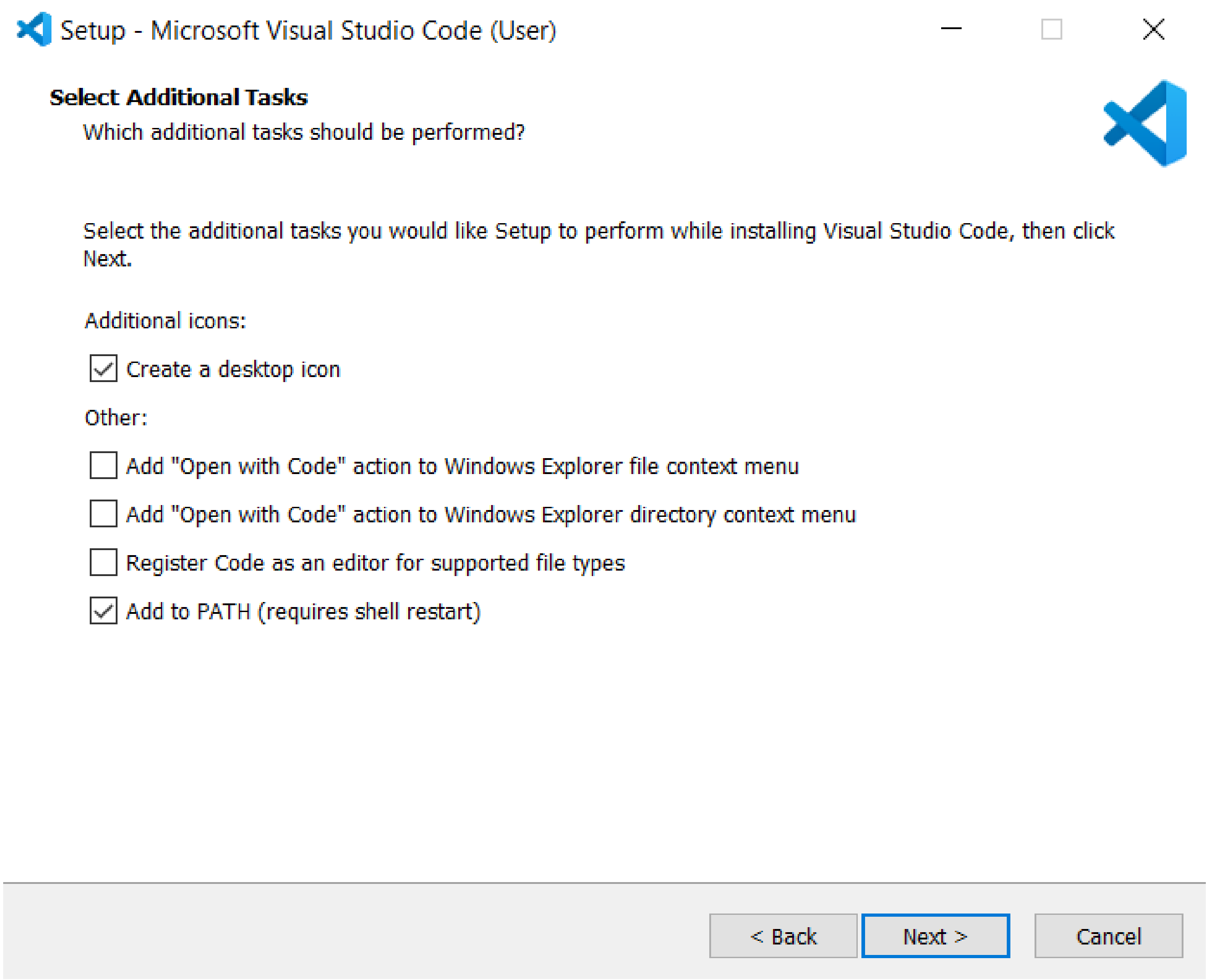
To continue, click Next. If you would like to select a different folder, click Browse.

Visual Studio Code

□ Don't create a Start Menu folder

Fig. 9.2.3 Select Start Menu Folder of VS Code

- Select the required options as per your need by clicking in the checkbox.
- Then select Next



Setup - Microsoft Visual Studio Code (User)

Select Additional Tasks

Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☑ Create a desktop icon

Other:

□ Add "Open with Code" action to Windows Explorer file context menu

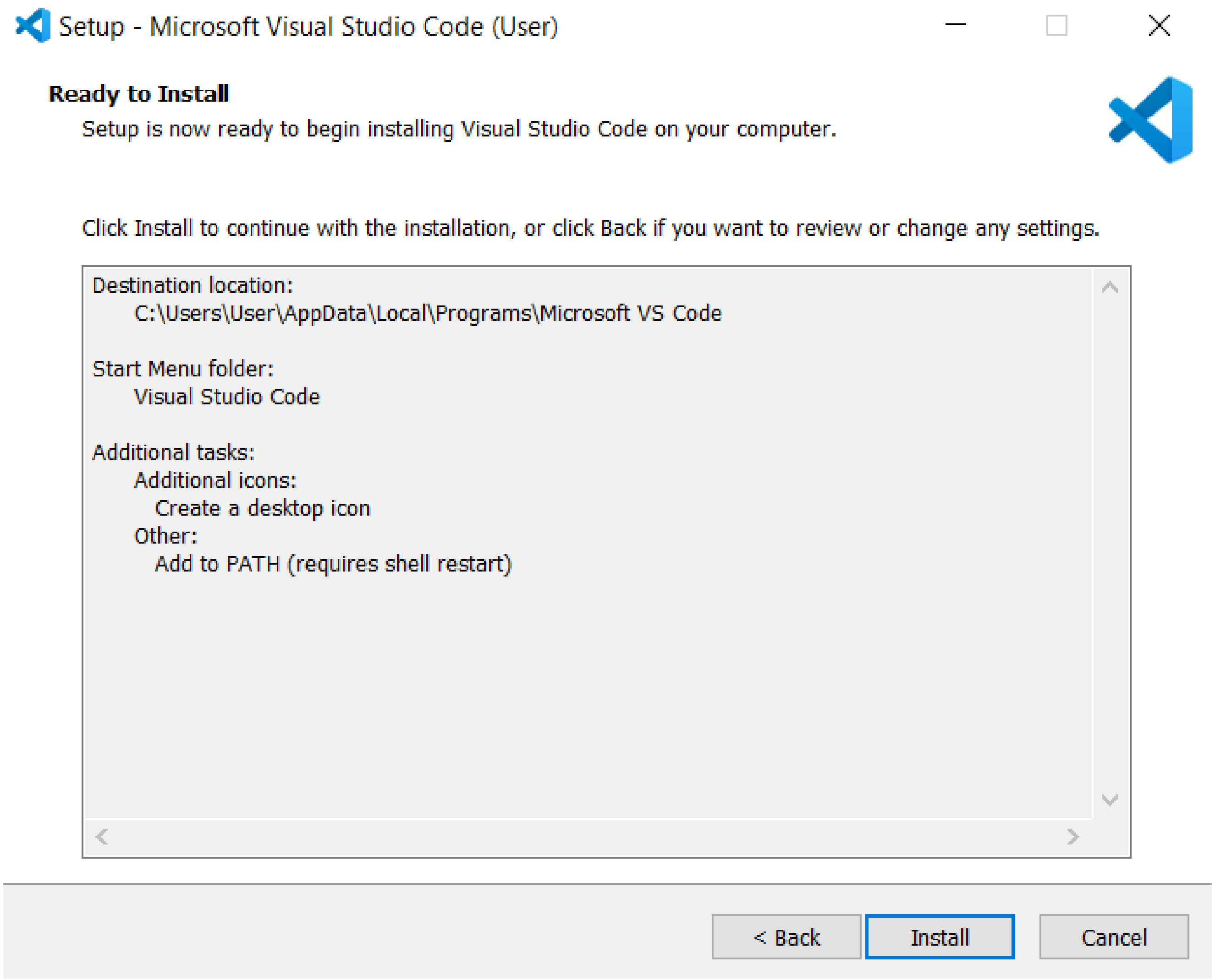
□ Add "Open with Code" action to Windows Explorer directory context menu

□ Register Code as an editor for supported file types

☑ Add to PATH (requires shell restart)

Fig. 9.2.4 Select Additional Tasks of VS Code

- Select Next.



Ready to Install

Setup is now ready to begin installing Visual Studio Code on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

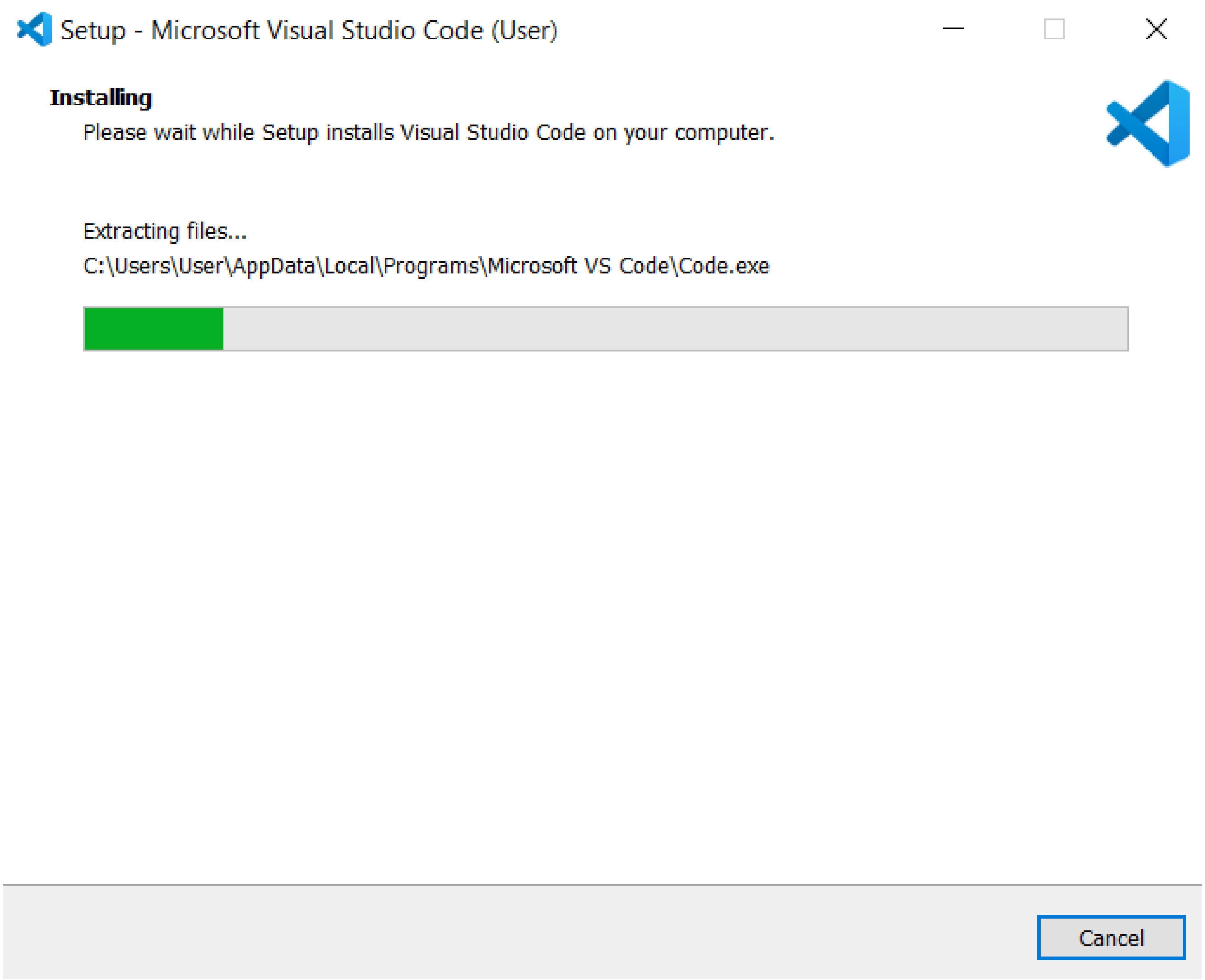
Destination location:
C:\Users\User\AppData\Local\Programs\Microsoft VS Code

Start Menu folder:
Visual Studio Code

Additional tasks:
Additional icons:
Create a desktop icon
Other:
Add to PATH (requires shell restart)

Fig. 9.2.5 Ready to install VS Code

- Select Install



Setup - Microsoft Visual Studio Code (User)

Installing

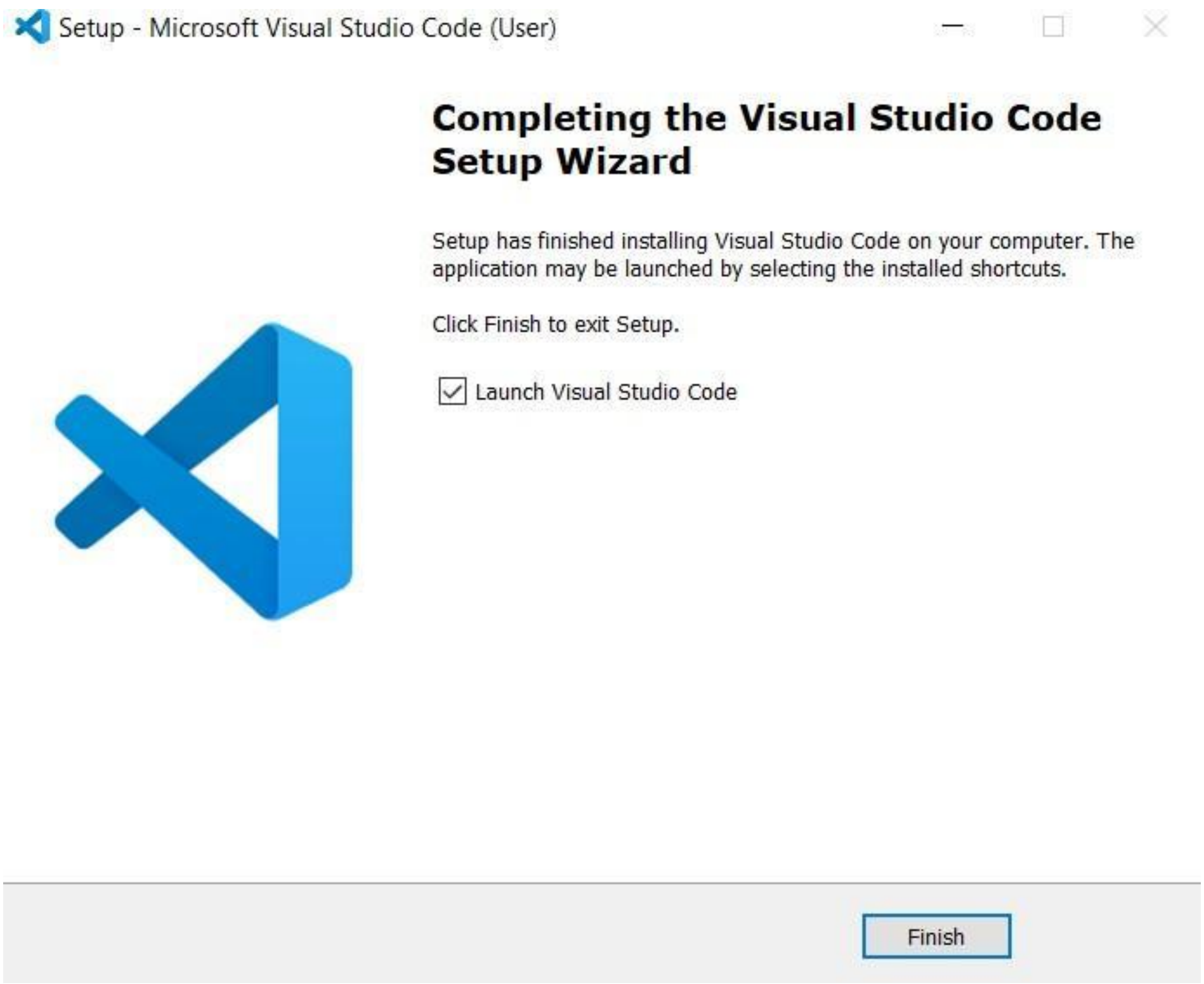
Please wait while Setup installs Visual Studio Code on your computer.

Extracting files...

C:\Users\User\AppData\Local\Programs\Microsoft VS Code\Code.exe

Fig. 9.2.6 Installing VS Code

- Wait a bit while it gets installed (The green color fills the bar).



Setup - Microsoft Visual Studio Code (User)

Completing the Visual Studio Code Setup Wizard

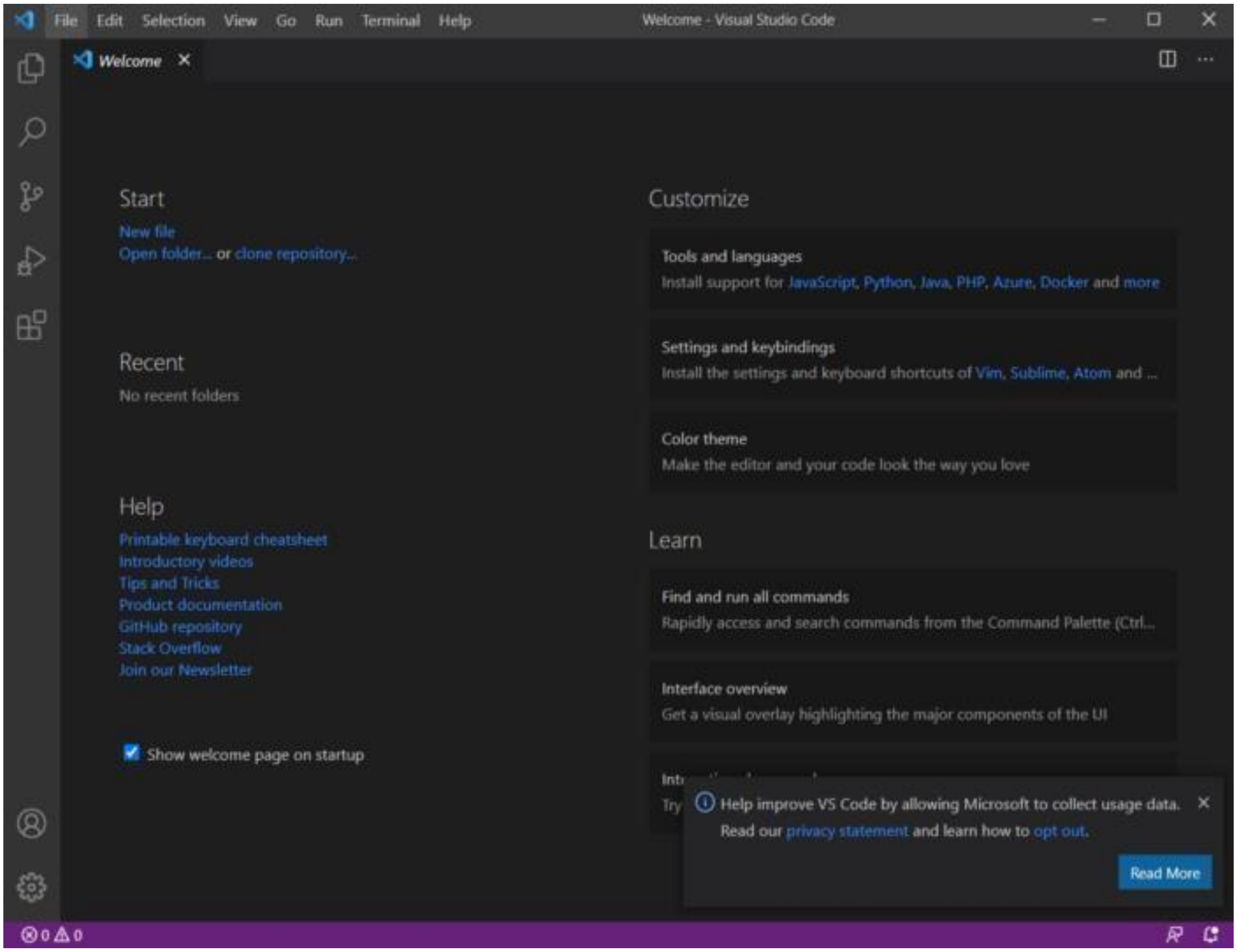
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Visual Studio Code

Fig. 9.2.7 Completing VS Code Setup Wizard

- Click Finish to exit Setup. Check in the check box to launch VS Code right now.



Start

New file

Open folder... or clone repository...

Recent

No recent folders

Help

Printable keyboard cheatsheet

Introductory videos

Tips and Tricks

Product documentation

GitHub repository

Stack Overflow

Join our Newsletter

☑ Show welcome page on startup

Customize

Tools and languages

Install support for JavaScript, Python, Java, PHP, Azure, Docker and more

Settings and keybindings

Install the settings and keyboard shortcuts of Vim, Sublime, Atom and ...

Color theme

Make the editor and your code look the way you love

Learn

Find and run all commands

Rapidly access and search commands from the Command Palette (Ctrl...

Interface overview

Get a visual overlay highlighting the major components of the UI

Help improve VS Code by allowing Microsoft to collect usage data. Read our privacy statement and learn how to opt out.

Fig. 9.2.8 VS Code Window

- Congratulations! VS Code got installed in your system successfully.
- Now a new dialogue box appears. This is VS Code IDE.
- Click New file to open a new file.

CHAPTER-10 CONCLUSION & FUTURE WORK

CHAPTER-10
CONCLUSION & FUTURE WORK

10.1 CONCLUSION

In conclusion, the age and gender detection project has successfully achieved its objectives of developing a robust and accurate system for predicting age ranges and genders from facial images. Through the integration of deep learning techniques, image processing algorithms, and user-friendly interfaces, the project has delivered a valuable tool with various applications in biometrics, surveillance, and human-computer interaction.

The implementation of the age and gender detection system involved several key components, including data collection and preprocessing, model development and training, user interface design, and integration with external libraries and frameworks. Through meticulous testing and validation, we have demonstrated the effectiveness and reliability of the system in accurately predicting age and gender labels from diverse datasets and real-world scenarios.

10.2 FUTURE WORK

- As tree data and graph structure is not implemented till now, we will do it in future.
- Making interface a bit more attractive.

REFERENCES

JOURNALS / RESEARCH PAPERS

1. Aryan Saxena, Prabhangad Singh, Shailendra Narayan Singh initials. (2021) Gender and Age detection using Deep Learning, *IEEE.*
    - https://ieeexplore.ieee.org/document/9377041

BOOKS

1. Jeremy Howard, Sylvain Gugger, initials, ‘Deep Learning for Coders with Fastai & PyTorch’, O’Reilly Media, First Edition, 2020.

WEBSITES (with exact URL up to page)

1. Dataset : https://susanqq.github.io/UTKFace/
2. TensorFlow : https://www.tensorflow.org/tutorials

PROJECT SUMMARY

About Project

Title of the project	**Gen-Age Detection System**
Semester	**6th**
Members	**1. Disha Vishwakarma** **2. Yuvika Sinha**
Team Leader	**Disha Vishwakarma**
Describe role of every member in the project	**Disha Vishwakarma: Model + Frontend Developer** **Yuvika Sinha: Model+ Fronted Developer**
What is the motivation for selecting this project?	**We chose this project because it helps solve a problem, it's interesting, and it can be useful to people.**
Project Type **(Desktop Application, Web Application, Mobile App, Web)**	**Web Application**

Tools &Technologies

Programming language used	**Python**
Compiler used **(withversion)**	**NA**
IDE used **(with version)**	**Microsoft Visual Studios Code(1.51.1)**
Front End Technologies **(with version, wherever Applicable)**	**HTML & CSS**
Back End Technologies **(with version, wherever applicable)**	**Flask**
Database used **(with version)**	**NA**

Software Design& Coding

Is prototype of the software developed?	**NA**
SDLC model followed **(Waterfall, Agile, Spiral etc.)**	**Agile Methodology**

Why above SDLC model is followed?	**Agile model has a set of guidelines that are: small, highly motivated project team and supports changing requirements. We need both guidelines to develop our project.**
Justify that the SDLC model mentioned above is followed in the project.	**Since the demand (functionalities) of the website kept on changing every now and then therefore we used the Agile model, so that we could make desired changes whenever needed.**
Software Design approach followed **(Functional or Object Oriented)**	**Functional oriented approach**
Name the diagrams developed **(according to the Design approach followed)**	**Class diagram, Use Case diagram & Data Flow Diagram**
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	**NA**
No. of Tiers **(example 3-tier)**	**NA**
Total no. of front-end pages	**3**
Total no. of tables in database	**NA**
Database is in which Normal Form?	**NA**
Are the entries in database encrypted?	**NA**
Front end validations applied **(Yes / No)**	**Yes**
Session management done **(in case of web applications)**	**No**
Is application browser compatible **(in case of web applications)**	**Yes**
Exception handling done **(Yes / No)**	**No**
Commenting done in code **(Yes / No)**	**No**
Naming convention followed **(Yes / No)**	**Yes**
What difficulties faced during deployment of project?	**We faced issues while displaying bad camera quality.**
Total no. of Use-cases	**4**

Give titles of Use-cases	**1. Capture Image** **2. Preprocess Image** **3. Predict Age and Gender** **4. Display Results**

Project Requirements

MVC architecture followed **(Yes / No)**	**No**
If yes, write the name of MVC architecture followed **(MVC-1, MVC-2)**	**NA**
Design Pattern used **(Yes / No)**	**No**
If yes, write the name of Design Pattern used	**NA**
Interface type **(CLI / GUI)**	**GUI**
No. of Actors	**2**
Name of Actors	**User and Developer**
Total no. of Functional Requirements	**6**
List few important non-Functional Requirements	**Accuracy, Maintainability, Performance, User-friendly**

Testing

Which testing is performed? **(Manual or Automation)**	**Manual**
Is Beta testing done for this project?	**No**

Write project narrative covering above mentioned points

Our project is a deep learning-based gender and age detection system designed to analyze images and predict the gender and age of individuals depicted. By leveraging advanced deep learning algorithms, the system captures, preprocesses, and extracts features from images, enabling accurate prediction of gender and age. This cutting-edge solution offers versatility and accuracy, catering to various applications in security, marketing, and healthcare. With a focus on user-friendly interfaces and robust performance, our project aims to revolutionize image analysis in diverse domains.

0187AD211013 - Disha Vishwakarma	Guide Signature
0187AD211046 - Yuvika Sinha	Dr. Vasima Khan

APPENDIX-1 GLOSSARY OF TERMS

(In alphabetical order)

C

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a corner stone technology of the World Wide Web, alongside HTML and JavaScript.

D

Deep Learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks with multiple layers (hence the term "deep") to learn complex patterns and representations from data. In the context of your project, deep learning plays a crucial role in developing accurate and reliable models for age and gender detection from image.

F

Flask

Flask is a lightweight and flexible web framework for Python, commonly used for building web applications and APIs. In the context of your age and gender detection project, Flask can be utilized to create a web-based interface for users to interact with the age and gender detection system.

H

HTML

HTML, or Hyper Text Markup Language, plays a crucial role in the development of the user interface for the age and gender detection system. Through HTML, we create the structure of the web-based interface where users interact with the system. HTML elements are used to define the layout, content, and functionality of the interface components such as buttons, forms, and images.

P

Python

Python is a versatile and widely-used programming language known for its simplicity, readability, and extensive ecosystem of libraries and frameworks. In the context of your age and gender detection project, Python serves as the primary programming language for implementing the machine learning model and building the backend logic of the application.

U

UML

The Unified Modeling Language is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

V

Visual Studio Code

Visual Studio Code is a free and open-source source code editor developed by Microsoft. It supports various programming languages and offers features like syntax highlighting, code completion, debugging, and version control integration. VS Code is highly customizable through extensions, making it popular among developers for its efficiency and versatility.