

ABSTRACT

High Quality automatic face recognition System may be a web base application. A top quality face recognition System could be a technology capable of identifying or verifying an individual from a digital image or a video frame from a video source. There are multiple methods within which biometric authentication system work, but normally, they work by comparing selected facial expression from given image with faces within a database. It's also described a Biometric AI based application that may uniquely identify someone by analyzing patterns supported the person's facial, texture and shape. We are using some different features also in our project. Example. Eyes or lips, some people have eyes in different shape or some people have little different, also the lips and nose are different. Means distance between nose and lips are difference off every one. So such features we use. Then in our project two phases are there the primary one is implementation of detection and second is recognition. Face Recognition has become a pretty field in Computer based application developed within the previous few decades. This can be thanks to the big selection of areas during which it's used. Additionally, due to the wide various of faces, face recognition from database images, real data, capture Images.

CONTENTS

SNO:	TOPIC
1.	INTRODUCTION
2.	LITERATURE SURVEY
3.	THEORITICAL ANALYSIS
4.	EXPERIMENTAL INVESTIGATIONS
5.	FLOW CHART
6.	RESULT
7.	ADVANTAGES & DISADVANTAGES
8.	APPLICATIONS
9.	CONCLUSION&FUTURE SCOPE
10.	BIBLIOGRAPHY

APPENDIX

1. INTRODUCTION

High Quality Facial Recognition System is a Software Base and Web Base Application. A High Quality Facial Recognition System is a Technology Capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analyzing patterns based on the person's facial textures and shape. We are going to use some different features in our project. Over the most recent couple of decades, facial recognition has been considered the standout among the most imperative applications compared to other biometric-based systems. Face Recognition has become an attractive field in Computer-based application developed in the last few decades. This is because of the wide range of areas in which it is used. In addition, because of the wide several of faces, face recognition from database images, real data, capture images.

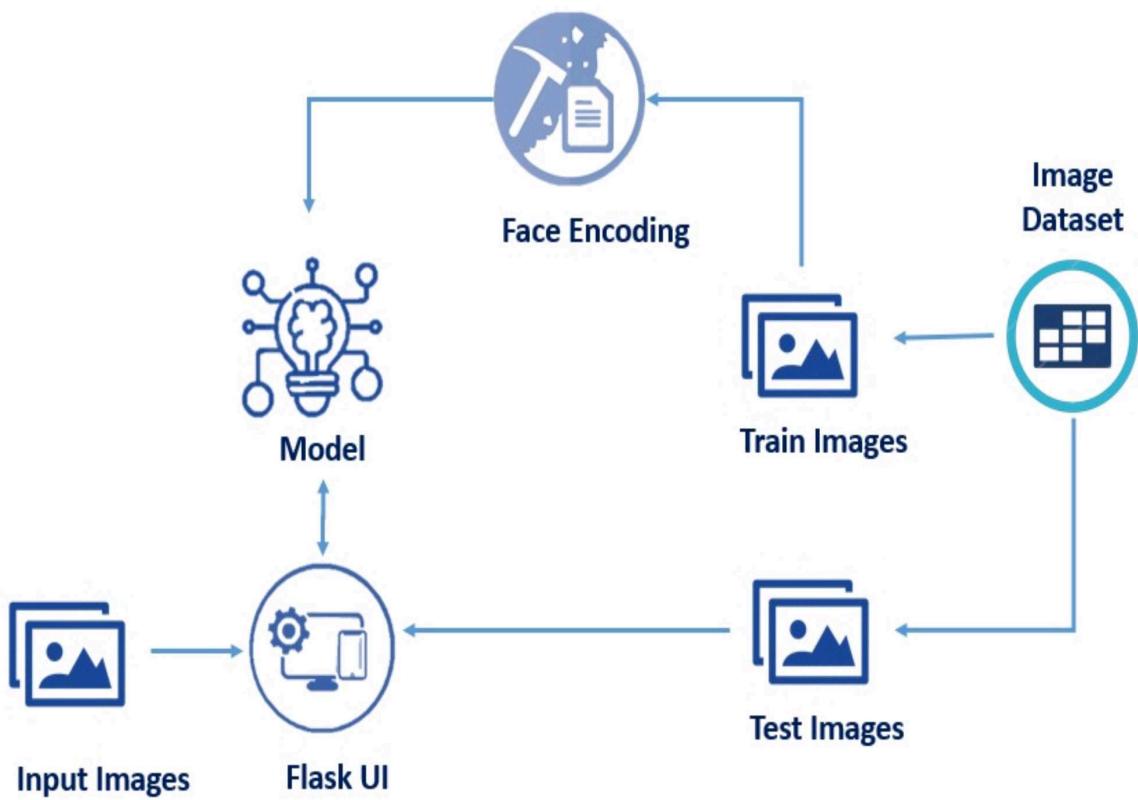
The task of face recognition has been actively researched in recent years. This paper provides an up-to-date review of major human face recognition research. We first present an overview of face recognition and its applications. Then, a literature review of the most recent face recognition is presented. Description and limitations of face databases which are used to test the performance of these face recognition algorithms are given. A brief summary of the face recognition vendor test, a large scale evaluation of automatic face recognition technology, and its conclusions are also given. Finally, we give a summary of the research results.

2.LITERATURE SURVEY

As one of the most successful application of image analysis and understanding, face recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend: the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. Even though current machine recognition systems have reached a certain level of maturity, their success is limited by the conditions imposed by many real applications. For example, recognition of face images acquired in an outdoor environment with changes in illumination and/or pose remains a largely unsolved problem. In other words, current systems are still far away from the capability of the human perception system. As a primary modality in biometrics, human face recognition has been employed widely in the computer vision domain because of its performance in a wide range of applications such as surveillance systems and forensics. Recently, near infrared (NIR) imagery has been used in many face recognition systems because of the high robustness to illumination changes in the acquired images. Even though some surveys have been conducted in this infrared domain, they have focused on thermal infrared methods rather than NIR methods. Face recognition has become an attractive field in computer-based application development in the last few decades. This is because of the wide range of areas in which it is used. In addition, because of the wide variations of faces, face recognition from database images, real data, capture images, and sensor images are a challenging problem and limitation. Image processing, pattern recognition, and computer vision are relevant subjects to the face recognition field. The innovation of new approaches of face authentication technologies is a continuous subject to building much stronger face recognition algorithms.

3.THEORETICAL ANALYSIS

3.1 Block Diagram



Prerequisites:

Software Requirements:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics . It was created by Guido van Rossum, and first released on February 20,1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio and Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Spyder

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third party plugins includes support for interactive tools for data inspection and embeds Python-specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

Flask

Webframework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

Imutils

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Python 3.

NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Build Tools

A build tool is a script, framework, or any kind of software created specifically for a given programming language to compile code, run tests on it, or do other operations needed for a build.

Cmake

CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice.

Dlib

dlib is a toolkit for making real world machine learning and data analysis applications in C++. While the library is originally written in C++, it has good, easy to use Python bindings.

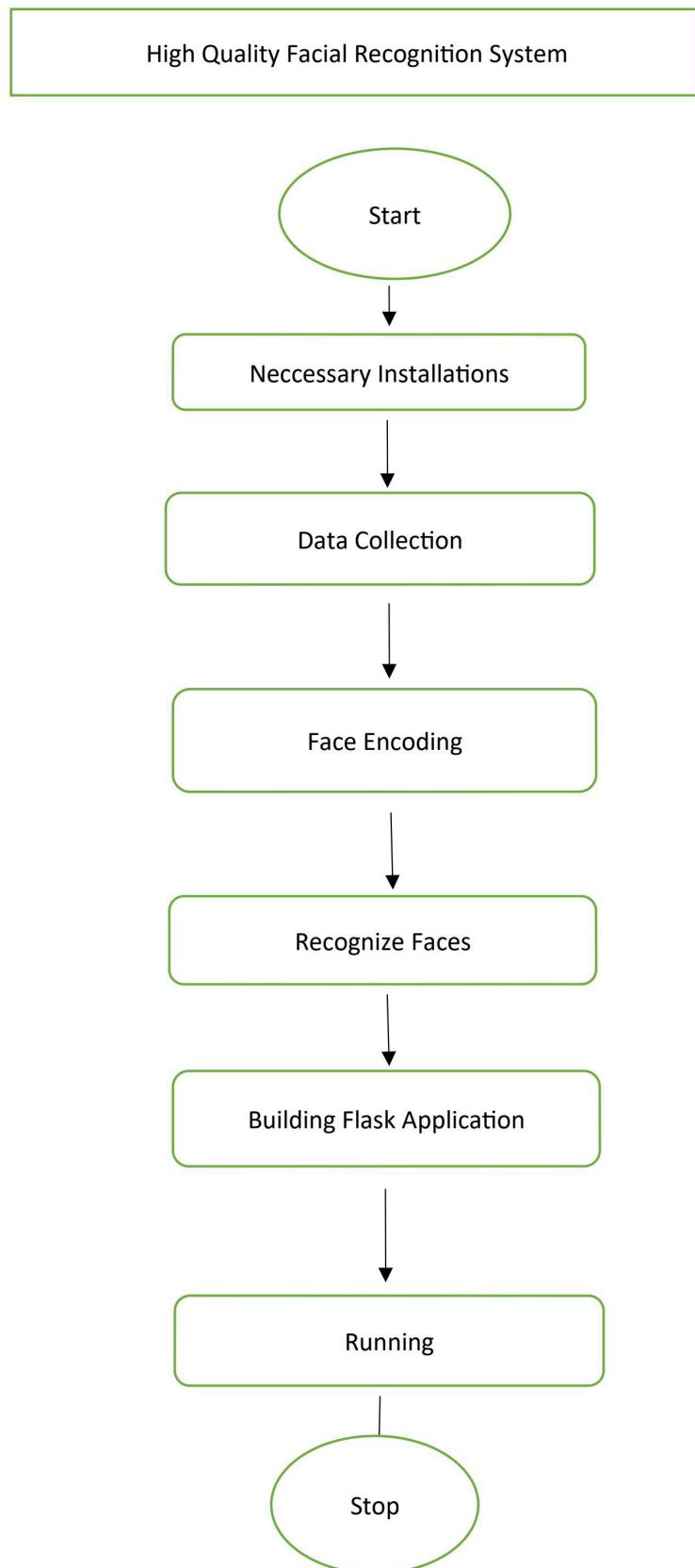
Hardware Requirements:

- Operating system: window 7 and above with 64bit
- Processor Type -Intel Core i3-3220
- RAM: 4Gb and above
- Hard disk: min 100GB

4.EXPERIMENTAL INVESTIGATIONS

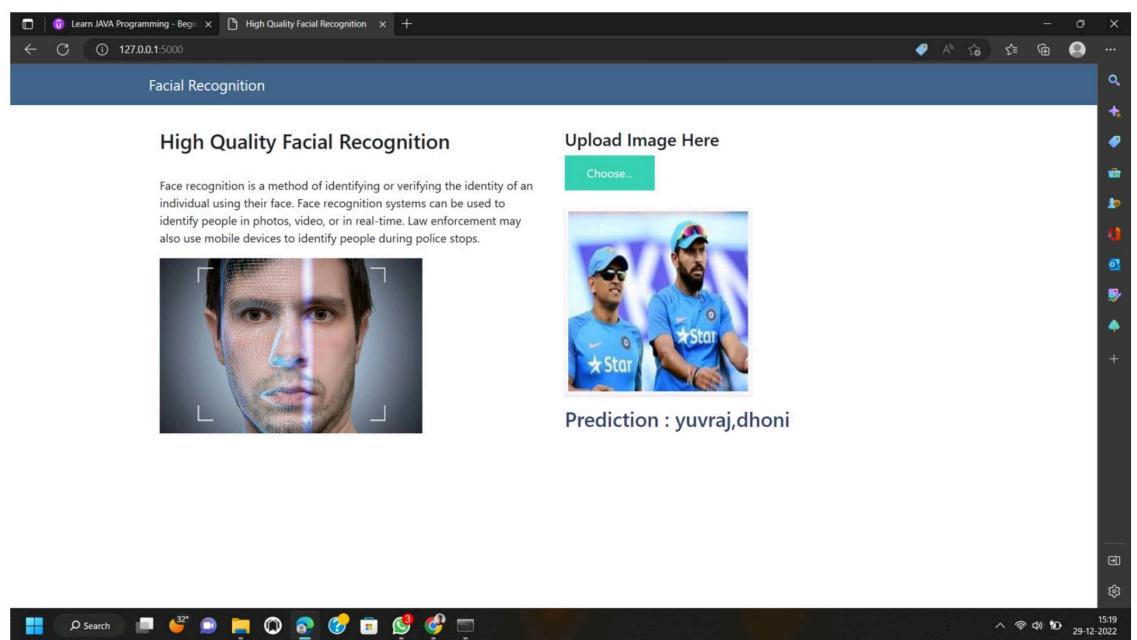
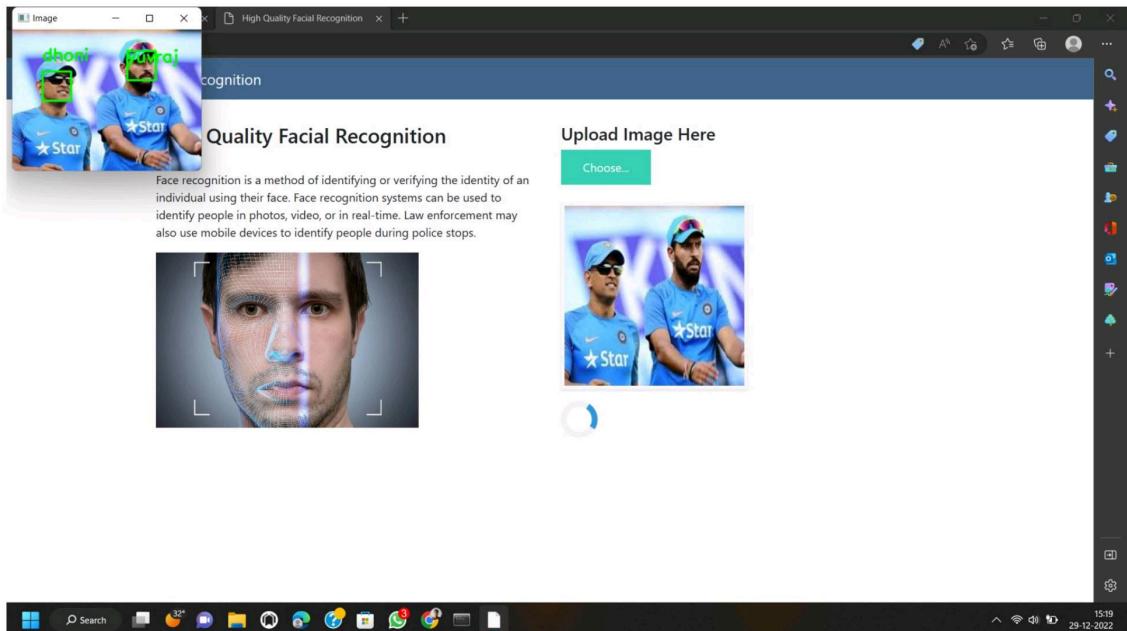
The face images which are used in this project experiment is based on different varieties including changes in pose and facial details and variability in expressions also in images .This end is achieved using the OpenCV identification using the dataset which consists 4 different types of 20 images for each person. All images in the dataset is considered for experiment purpose. The performance measure is based on accuracy of face recognition.

5.FLOWCHART

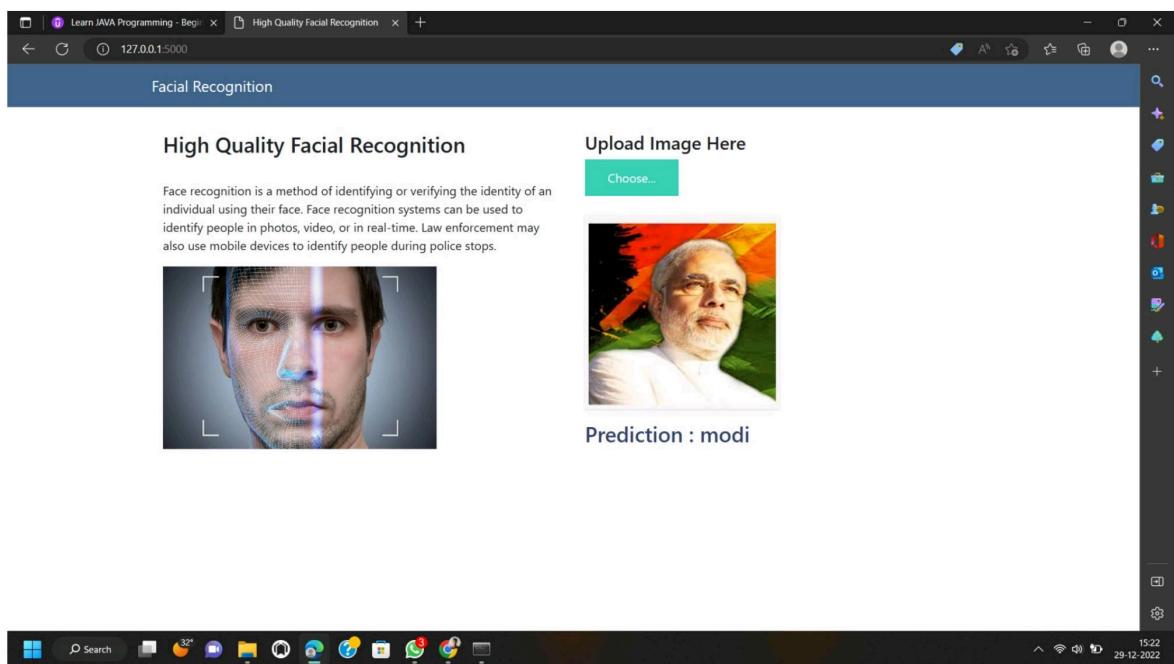
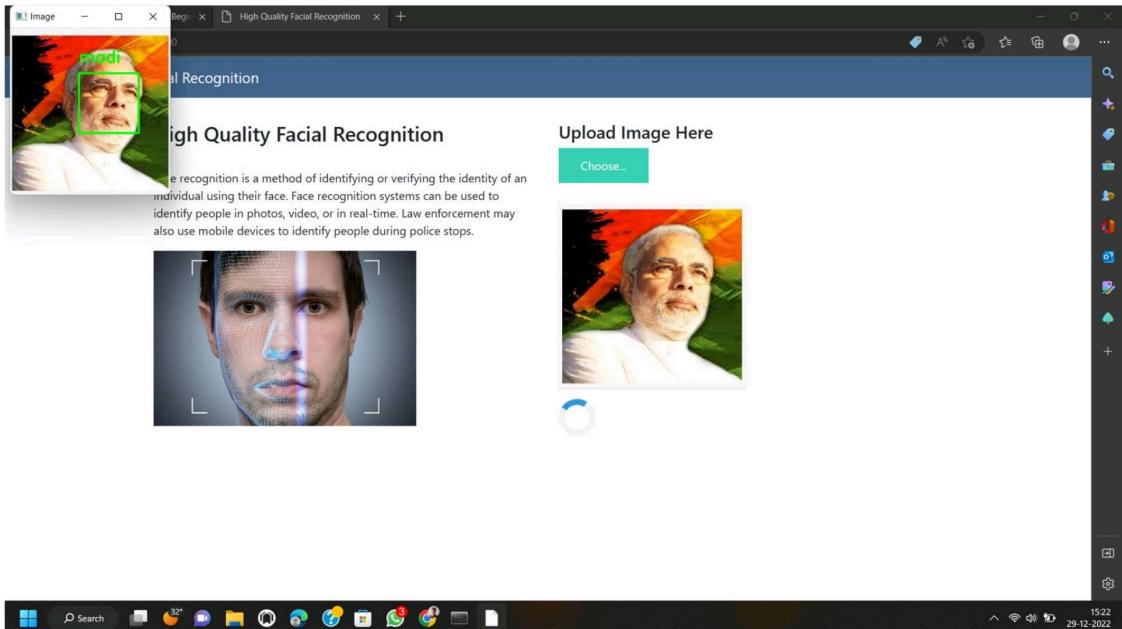


6.RESULT

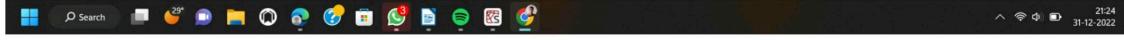
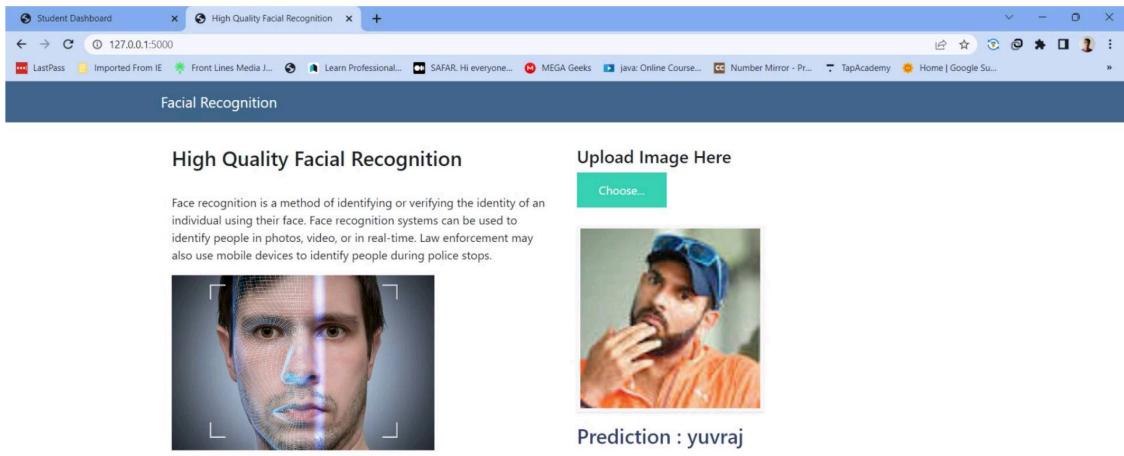
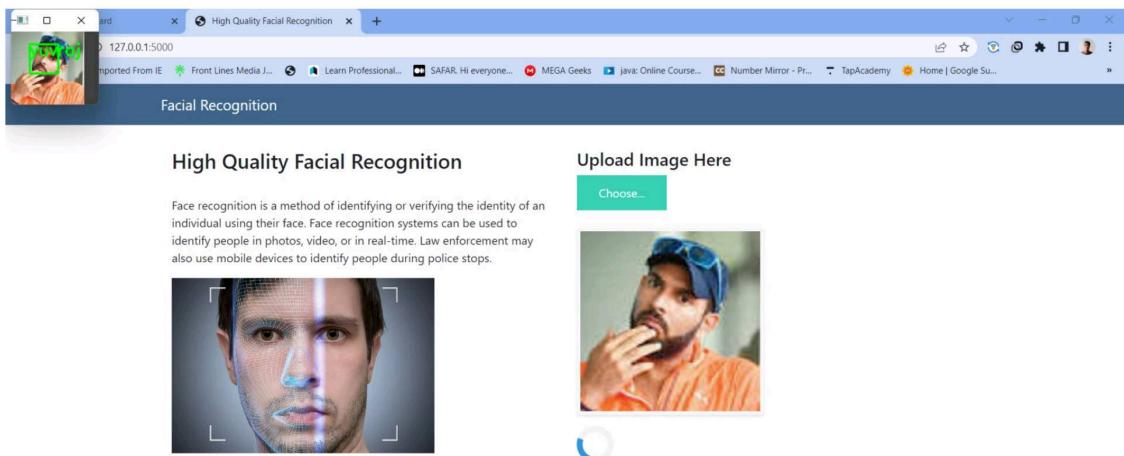
Output Image 1:-



Output image 2 :-



Output Image 3:-



7.ADVANTAGES and DISADVANTAGES

ADVANTAGES:

- Helps find missing people
- Protects businesses against theft
- Improves medical treatment
- Strengthens security measures
- Makes shopping more efficient
- Reduces the number of touchpoints
- Improves photo organization

DISADVANTAGES

- Threatens privacy
- Imposes on personal freedom
- Violates personal rights
- Data vulnerabilities
- Misuse causing fraud and other crimes
- Technology is still new
- Errors can implicate innocent people
- Technology can be manipulated

8.APPLICATIONS

From business solutions, commercial solutions and even solutions for the home—we take a look at how facial recognition is improving the way different industries work today—and where applicable, the exciting opportunities it presents for the future. Beyond just unlocking phones or laptops, the biometric software behind facial recognition applications can accurately identify faces today, better than other people can. While this makes the technology an obvious choice for security and identification purposes, it can also be used creatively and repurposed to serve different industries.

9. CONCLUSION&FUTURE SCOPE

The High Quality Facial Recognition System is a Software base and Web base application. This technology is used to identifying or verifying a person from digital image or video frame from video source. The High Quality Facial Recognition System is one of the newer developments of biometric Artificial Intelligence based application that can be identifying person. Other biometric identifiers such as a fingerprint scanner and voice recognition require many different pieces in order to function. The High Quality Facial Recognition System is highly effective biometric technology that holds a lot of potential. The High Quality Facial Recognition System can save resources and time and even generate new income streams for companies that implement it right.

The future of High Quality Face Recognition System technology is bright. Forecasters opine that this technology is predicted to grow at a formidable rate and may generate huge revenues within the approaching years. Security and surveillances are the foremost segments which may be deeply influenced. Other

areas that are now welcoming it with open arms are private industries, public buildings, and schools. It's estimated that it will even be adopted by retailers and industry in coming years to remain fraud in debit/credit card purchases and payment especially people who are online.

10.BIBLIOGRAPHY

We referred some books and surfed the internet for the better outcome of the project:

- [1] N. Dhib, H. Ghazzai, H. Besbes, and Y. Massoud, "Extreme gradient boosting machine learning algorithm for safe auto insurance operations," in IEEE International Conference on Vehicular Electronics and Safety (ICVES'19), Cairo, Egypt, Sept. 2019.
- [2] M. Wassel, "Property Casualty: Deterring Claims Leakage in the Digital Age," Cognizant Insurance Practice, Tech. Rep., 2018.
- [3] N. Dhib, H. Ghazzai, H. Besbes, and Y. Massoud, "A Very Deep Transfer Learning Model for Vehicle Damage Detection and Localization," in IEEE International Conference on Vehicular Electronics and Safety (ICVES'19), Cairo, Egypt, Sept. 2019.
- [4] K. Patil, M. Kulkarni, A. Sriraman and S. Kerande, "Deep Learning Based Car Damage Classification," in IEEE Int. Conf. Machine Learning App. (ICMLA'17), Cancun, Mexico, Dec. 2017.
- [5] j. d. Deijin, "Automatic Car Damage Recognition using Convolutional Neural Networks," March 2018

11.APPENDIX

FLASK CODE:-

```
# USAGE
# python recognize_faces_image.py --encodings encodings.pickle --image examples/example_01.png
# import the necessary packages
import face_recognition
import pickle
import cv2
from flask import Flask, request, render_template
#from werkzeug import secure_filename
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
import sys
import os.path
import glob
app = Flask(__name__, static_url_path='')
@app.route('/', methods=['GET'])
def index():
    return render_template('base.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        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("[INFO] loading encodings...")
        data = pickle.loads(open('encodings.pickle', "rb").read())
        # load the input image, convert it from BGR to RGB channel ordering,
        # and use Tesseract to localize each area of text in the input image
        image = cv2.imread(file_path)
        rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        # load the known faces and embeddings

        # detect the (x, y)-coordinates of the bounding boxes corresponding
        # to each face in the input image, then compute the facial embeddings
        # for each face
        print("[INFO] recognizing faces...")
        boxes = face_recognition.face_locations(rgb,
            model="cnn")
        encodings = face_recognition.face_encodings(rgb, boxes)

        # initialize the list of names for each face detected
        names = []
```

```

# loop over the facial embeddings
for encoding in encodings:
    # attempt to match each face in the input image to our known
    # encodings
    matches = face_recognition.compare_faces(data["encodings"],
        encoding)
    name = "Unknown"

    # check to see if we have found a match
    if True in matches:
        # find the indexes of all matched faces then initialize a
        # dictionary to count the total number of times each face
        # was matched
        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
        counts = {}

        # loop over the matched indexes and maintain a count for
        # each recognized face face
        for i in matchedIdxs:
            name = data["names"][i]
            counts[name] = counts.get(name, 0) + 1

        # determine the recognized face with the largest number of
        # votes (note: in the event of an unlikely tie Python will
        # select first entry in the dictionary)
        name = max(counts, key=counts.get)

        # update the list of names
        names.append(name)

    # loop over the recognized faces
    for ((top, right, bottom, left), name) in zip(boxes, names):
        # draw the predicted face name on the image
        cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)
        y = top - 15 if top - 15 > 15 else top + 15
        cv2.putText(image, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
            0.75, (0, 255, 0), 2)

    # show the output image
    cv2.imshow("Image", image)
    cv2.waitKey(0)

return ','.join(map(str, names))
if __name__ == '__main__':
    app.run(debug=False)

```

HTML CODE :-

index.html:

```
</div>
<div class="col-sm-6">
    <div>
        <h4>Upload Image Here</h4>
    <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
            Choose...
        </label>
        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg,.pdf">
    </form>

    <div class="image-section" style="display:none;">
        <div class="img-preview">
            <div id="imagePreview">
            </div>
        </div>
        <div>
            <button type="button" class="btn btn-info btn-lg " id="btn-predict">Faces</button>
        </div>
    </div>
```

```
28
29      <div class="loader" style="display:none;"></div>
30
31      <h3>
32          <span id="result"> </span>
33      </h3>
34
35
36      </div>
37      </div>
38
39      </div>
40      </div>
41      </div>
42  </div>
43 </body>
44
45 <footer>
46     <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
47 </footer>
48
49 </html>
```