Distributed Systems

Facial Recognition Project

Live stream

CA2

Milan Ples

20088120

Contents

[Introduction: 3](#_Toc122447762)

[Drawbacks 5](#_Toc122447763)

[Project: 6](#_Toc122447764)

[Introduction 6](#_Toc122447765)

[Implementation 6](#_Toc122447766)

[Functional and nonfunctional requirements 7](#_Toc122447767)

[Functional Requirements: 7](#_Toc122447768)

[Non Functional Requirements: 7](#_Toc122447769)

[Diagram 8](#_Toc122447770)

[Preprocessing 8](#_Toc122447771)

[Video Capture and Recognition 9](#_Toc122447772)

[Web Application 10](#_Toc122447773)

[Improvements to the code: 10](#_Toc122447774)

[Failed tries: 11](#_Toc122447775)

[Conclusion 12](#_Toc122447776)

[Youtube Link: 12](#_Toc122447777)

# Introduction:

Facial recognition technology has become increasingly prevalent in our daily lives, with applications ranging from unlocking our phones to improving security measures in various industries. This project aims to build a web service that utilizes a webcam to detect and recognize human faces, utilizing the power of neural networks and machine learning to accurately identify individuals.

To start with the process of facial recognition, we first need to locate the face within the image and crop it to focus on the facial features. This can be done using various image processing techniques such as edge detection and feature extraction. Once the face is isolated, specialized libraries are used to extract and map the specific features of the face, including the shape of the eyes, nose, and mouth, as well as the distance between different features.

These mapped features are then sent to a pre-trained neural network, which uses patterns and problem-solving abilities to mimic the behavior of the human brain in the fields of artificial intelligence, machine learning, and deep learning. In the case of facial recognition, the neural network receives the image and generates a feature vector of 128 values that represent the unique characteristics of the face. These values are generated by the model, which has been trained using a large dataset of facial images.

Once the face has been detected and the feature vector has been generated, the web service can then compare the feature vector to a database of known individuals to determine the identity of the person in the image. If a match is found, a rectangle will be drawn around the face on the image and the person's name will be displayed alongside additional text to confirm the match for the user. The video stream from the application will also be shared on the network, allowing others to view the live feed and observe the facial recognition in action.

In addition to the primary goal of accurately identifying individuals, this web service can also be used for a variety of other purposes. For example, it could be utilized in the automotive industry to grant access to vehicles through facial recognition, or in the entertainment industry for virtual video conferencing. The potential applications of this technology are vast and continue to expand as it becomes more widely adopted.

Overall, this project demonstrates the capabilities of facial recognition technology and the power of neural networks and machine learning to accurately identify individuals through a webcam. With the increasing prevalence of this technology in our daily lives, it is important to understand its capabilities and potential applications.Advantages:

Facial recognition technology has a number of potential advantages, including:

Convenience: Facial recognition can make certain tasks more convenient, such as unlocking a phone or granting access to a secure location.

Security: Facial recognition can be used to improve security measures in various industries, such as the automotive and entertainment industries. It can also be used to identify individuals in security footage and assist with law enforcement efforts.

Personalization: Facial recognition technology can be used to personalize experiences for individuals, such as recommending products or services based on their interests and preferences.

Efficiency: Facial recognition can be used to automate certain tasks and processes, improving efficiency and reducing the need for manual labor.

Accessibility: Facial recognition technology can also be used to improve accessibility for individuals with disabilities, such as those who are unable to use traditional methods of identification or input.

Overall, facial recognition technology has the potential to streamline certain processes, improve security, and enhance personalization and accessibility. It is important to consider the potential advantages and disadvantages when implementing this technology and to take steps to ensure that it is used ethically and responsibly.

# Drawbacks

Despite the many potential benefits of facial recognition technology, it is important to recognize that it also has its limitations and potential drawbacks. Some of the main criticisms of facial recognition include:

Privacy concerns: One of the main concerns surrounding facial recognition is the potential for invasion of privacy. This technology allows for the tracking and surveillance of individuals without their consent, raising ethical and legal concerns.

Inaccuracy: Facial recognition algorithms can often be inaccurate, particularly when it comes to identifying people of color or those with certain facial features. This can lead to false positives and false negatives, which can have serious consequences in certain situations.

Bias: Facial recognition algorithms can also be biased, as they are often trained on datasets that do not accurately reflect the diversity of the population. This can lead to the algorithm performing poorly on certain groups of people, such as those with darker skin tones.

Misuse: There is also a risk that facial recognition technology could be misused by governments or private companies for nefarious purposes, such as mass surveillance or targeted advertising.

Dependence: The increasing reliance on facial recognition technology also raises concerns about its potential to replace human judgement and decision-making. This could lead to a loss of jobs and a reliance on the technology, which is not without its own flaws and limitations.

It is important to consider these potential drawbacks when implementing facial recognition technology and to take steps to mitigate any negative consequences. This may include measures such as regulating the use of the technology, ensuring that algorithms are tested for accuracy and bias, and respecting individuals' privacy rights.

# Project:

## Introduction

The Facial Recognition project is a software application that uses machine learning techniques to identify and track human faces in digital images and video. Implemented using Python, Flask, SocketIO, OpenCV, and face\_recognition, the project enables users to interact with the application through a web browser and perform actions such as uploading images, adjusting settings, and viewing the facial recognition results in real-time. It also includes features such as displaying the distance to the detected faces and the frame rate. The project demonstrates the capabilities of machine learning and real-time video processing, as well as the potential applications and implications of facial recognition technology.

## Implementation

The project is implemented using the Python programming language and several libraries, including Flask, SocketIO, OpenCV, and face\_recognition.

Flask is a web framework that enables the creation of web applications using Python. It provides a simple and lightweight way to set up a web server and handle HTTP requests and responses.

SocketIO is a library that allows real-time communication between the server and the client using WebSockets. It enables the server to send data to the client in real-time, without the need for the client to constantly poll the server for updates.

OpenCV (Open Source Computer Vision) is a library for image and video processing. It provides a wide range of functions for performing tasks such as object detection, image filtering, and image manipulation.

face\_recognition is a library for performing facial recognition tasks such as detecting and recognizing faces in images and video. It uses machine learning algorithms to calculate numerical representations, or "encodings", of the unique characteristics of a face.

## Functional and nonfunctional requirements

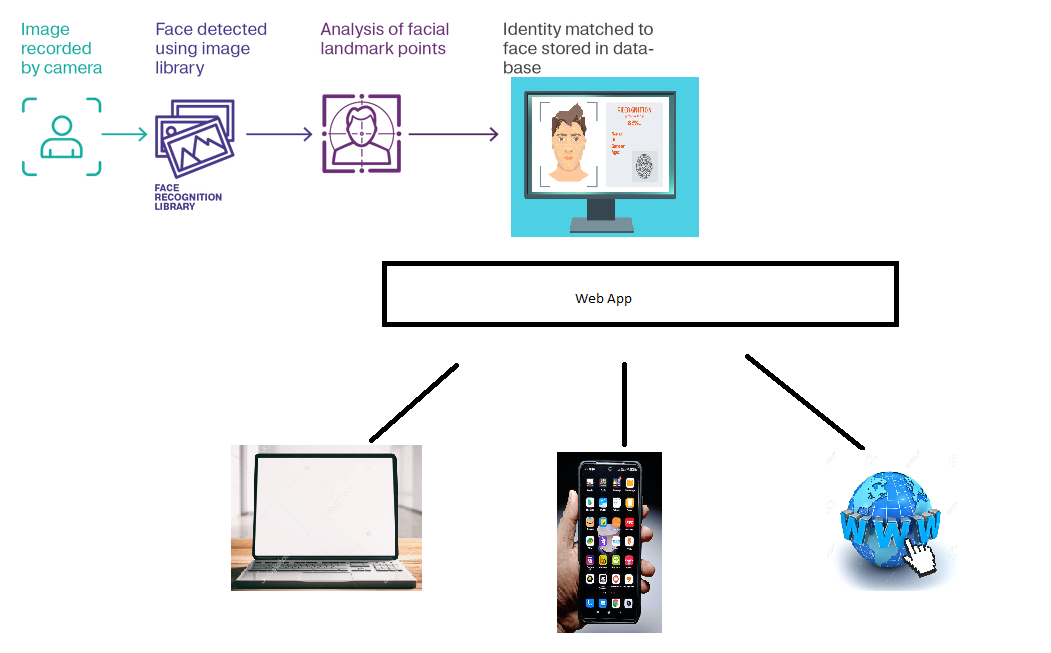
### Functional Requirements:

* Transform Original Video 
* Heterogeneous Delivery 
* The application must be able to detect at least 3 faces. 
* The application must be able to be a webserver accessible do devices over the network. 
* There is now authorisation requirements to access the stream. 
* The facial recognition must show the person’s name and the percentage match to the trained face. 

### Non Functional Requirements:

* The live video should be greater than 15 FPS. 
* The quality of the video should be on par with already available streaming qualities at least greater than 480p. 
* The faces should be of people in the class for demo purposes. 
* The server is hosted on local host machine. 
* The project is created with python and the web service provided by python flask library. 

## Diagram



Figure

Borrowed from <https://www.ellucian.com/blog/facial-recognition-campus-benefits-security-risks>

## Preprocessing

The project sets the path to the directory containing the images for facial recognition and creates empty lists for storing these images and their corresponding names.

It then reads each image in the directory, adds it to the images list, and adds the image name (in upper case) to the image\_names list. This allows the project to keep track of the images and their corresponding names for later use in the facial recognition process.

Encoding

The program defines a function called find\_encodings, which takes a list of images as input and returns a list of face encodings for these images.

A face encoding is a numerical representation of the unique characteristics of a face, such as the shape of the eyes, nose, and mouth. It is calculated using machine learning algorithms that analyze the features of a face and create a numerical representation that is unique to that specific face.

The find\_encodings function first converts the input images to the RGB color space using OpenCV. This is necessary because the face\_recognition library expects the images to be in this color space.

It then uses the face\_recognition library to calculate the face encoding for each image by analyzing the features of the face in the image.

The encodings are then added to the encoding\_list and the list is returned.

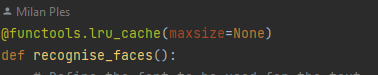
The project calls the find\_encodings function and stores the returned list of encodings in a variable called encode\_list\_known. This list will be used later to compare with the encodings of faces detected in real-time video frames.

## Video Capture and Recognition

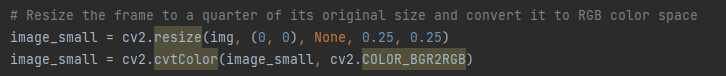
The project creates a VideoCapture object from the OpenCV library, which is used to capture video frames from the webcam.

It defines the recognise\_faces function, which is responsible for detecting and recognizing faces in the video frames using the face\_recognition library.

The function is decorated with the lru\_cache decorator from the functools library, which enables caching of the function's return value. This results in faster execution by avoiding the need to recalculate the same value multiple times.



A loop is used to continuously capture video frames from the webcam, resize them to a quarter of their original size, and convert them to the RGB color space using OpenCV.



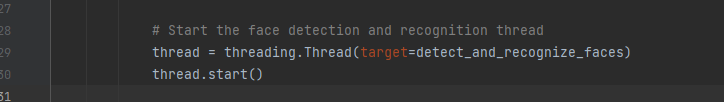
The face\_recognition library is then used to detect the coordinates of the faces in the frame, as well as their encodings.

The project compares the detected face encodings with the encode\_list\_known list using the face\_recognition.compare\_faces function. This function compares the input face encodings to a list of known face encodings and returns a list of True/False values indicating whether a match was found for each input encoding.

If a match is found, the project calculates the distance between the camera and the detected face using a formula based on the focal length of the camera and the coordinates of the detected face. The distance is then displayed on the video frame along with the name of the recognized person.

Another improvement added was through the use of threading which increased my FPS by over 50%.

It still runs kind of slow but this has speed it up considerably.



## Web Application

The project uses the Flask and SocketIO libraries to serve the video frames to a web browser in real-time using WebSockets.

It includes several Flask routes for handling image file uploads, displaying the video frames and facial recognition results, and handling user interactions with the web application.

Flask routes are URL patterns that can be used to trigger specific actions in the web application. For example, the '/' route is used to render the main page of the web application, which contains the video frames and the facial recognition results. The '/upload' route is used to handle image file uploads from the client, allowing users to add their own images to the database for facial recognition.

SocketIO is used to send the video frames from the server to the client in real-time, using a technology called WebSockets. This enables the web browser to display the video frames as they are being processed by the server, without the need for the client to constantly poll the server for updates.

## Improvements to the code:

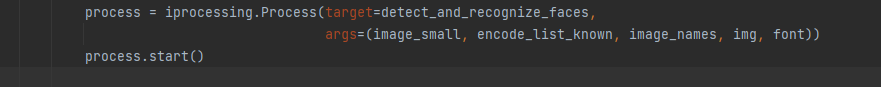
Threading: 50% increase

Caching: About 10% to performance and less overhead computation.

FPS counter: For monitoring

## Failed tries:

I tested the use of multiprocessing which did not result in any gain except in increase in overhead processing on my machine as it started multiple instances of the program. Similar to threading but no gains achieved.



Tensor flow and MCNN:

I wanted to use my GPU which would without the doubt increase performance but it would not work.

Had so many issues and complaints from the program about geometry and such so I binned it.

Tried to do Distance calculation of a face from camera, didn’t work.

Challenges:

One of my main challenges has been getting the libraries for facial recognition to work properly in my Windows environment.

This has required a significant amount of time and effort, as I have had to install and configure various software packages, drivers, and other dependencies. Additionally, I have had to troubleshoot any compatibility issues or errors that have arisen during the setup process.

Another challenge has been finding and integrating the right hardware, and ensuring that my system has sufficient processing power and storage to handle the demands of the facial recognition algorithms. Overall, setting up and maintaining a facial recognition system in my Windows environment has proven to be a complex and time-consuming process.

## Conclusion

The Facial Recognition project is a complex software application that demonstrates the capabilities of machine learning and real-time video processing. It uses a range of libraries and technologies to perform facial recognition and display the results in a user-friendly interface.

The project has allowed me to gain a deeper understanding of how technology works behind the scenes, particularly in the areas of web development, image and video processing, and machine learning. By implementing and interacting with the various components of the project, I have been able to learn more about the capabilities and limitations of each technology, as well as how they can be combined to create powerful and sophisticated software applications.

Overall, the Facial Recognition project has been a valuable learning experience, providing me with the opportunity to apply my knowledge and skills to a practical, real-world problem, and to gain a deeper understanding of the technologies and techniques involved.

In addition to the technical skills and knowledge that I have gained from working on the Facial Recognition project, I have also developed a greater appreciation for the importance of facial recognition technology in today's society. Facial recognition has the potential to revolutionize a wide range of industries and applications, from security and surveillance to healthcare and entertainment.

However, it is also important to consider the potential ethical and privacy implications of facial recognition technology. As with any powerful tool, it is important to use it responsibly and with consideration for the impact it may have on individuals and society as a whole.

In conclusion, the Facial Recognition project has been a valuable learning experience that has helped me to better understand the technology behind facial recognition and the potential applications and implications of this technology. It has also provided me with the opportunity to apply my knowledge and skills to a real-world problem, and to gain a deeper understanding of the technologies and techniques involved.

# Youtube Link:

<https://youtu.be/G9GP7ypOCI8>

# Github Link:

<https://github.com/Kasdal/recognizeFaces-CA2>