



Project Report Facial and Object/Entity Detection

Summary (TL;DR)

- This project leverages Deep Learning libraries to recognize and detect human faces and objects using live video
- One of the key inspirations for this project came from my idea that school shooters can be recognized in some cases based on:
 - If they are not existing school students
 - Carrying objects such as large bags or guns
- Future considerations for improving this Neural Network include:
 - Combining the identified person and objects in a single window
 - Currently, the face recognition and object/person recognition codes are separate
 - Identifying objects that indicate risk:
 - Large bags
 - Guns
- I am passionate about intelligent systems that can improve the life quality for people
- This project serves as a foundation for an ongoing project to build various computer vision and natural language processing functions that can then be used

- to create a risk assessment and security solution to secure school perimeters
- The goal is to help schools in USA and the world, better manage and address security risks

TABLE OF CONTENTS

Business Problem.....	2
Scope, Approach, Process, and Libraries.....	3
Solution Overview - Facial Recognition.....	4
Conceptual Solution Overview: Facial Recognition Training Code.....	5
Conceptual Solution Overview: Facial Recognition Live Video Code.....	6
Systems Screenshots.....	6
Facial Recognition Project Demo Video (Link).....	8
Facial Recognition Project Demo Screenshot.....	8
Solution Overview - Object/Entity Recognition.....	8
Object/Entity Recognition Project Demo Video (Link).....	9
Object/Entity Recognition Project Demo Screenshots.....	9
Key Challenges & Learnings For Future Exhibits:.....	9
Future Enhancements & Optimizations.....	10
Project Resource Links.....	11

Business Problem

- Campus security that primarily involves security guards is:

- Expensive
- Complex: schools are generally speaking fairly large open spaces
- In the post COVID world, hiring is a key issue
- Personnel are not well trained to monitor threats
- Threat identification is a key component of averting disastrous scenarios
- There are two mainstream offerings to manage school campus security in addition to other use cases, however, there is a need for more specialized solution tailored for schools [1]
- This project is a first step in the direction of creating specialized AI based solutions for schools

[1] [Actuate](#) and [Scylla](#)

Scope, Approach, Process, and Libraries

Scope:

- Define vision and approach
- Systems setup
- Model training (Facial Recognition only)
- Coding
- Report and presentation

Approach:

- Identify the best libraries for the applications
- Setup of systems
- Data collection and setup
- Troubleshooting and solution validation

Process:

- Systems setup:
 - Ubuntu Linux
 - Apache MyPHP
 - Apache MySQL
 - Webcam setup

- Installing the key libraries (See below)
- Data collection and setup
- Training
- Coding
- Validation

Libraries

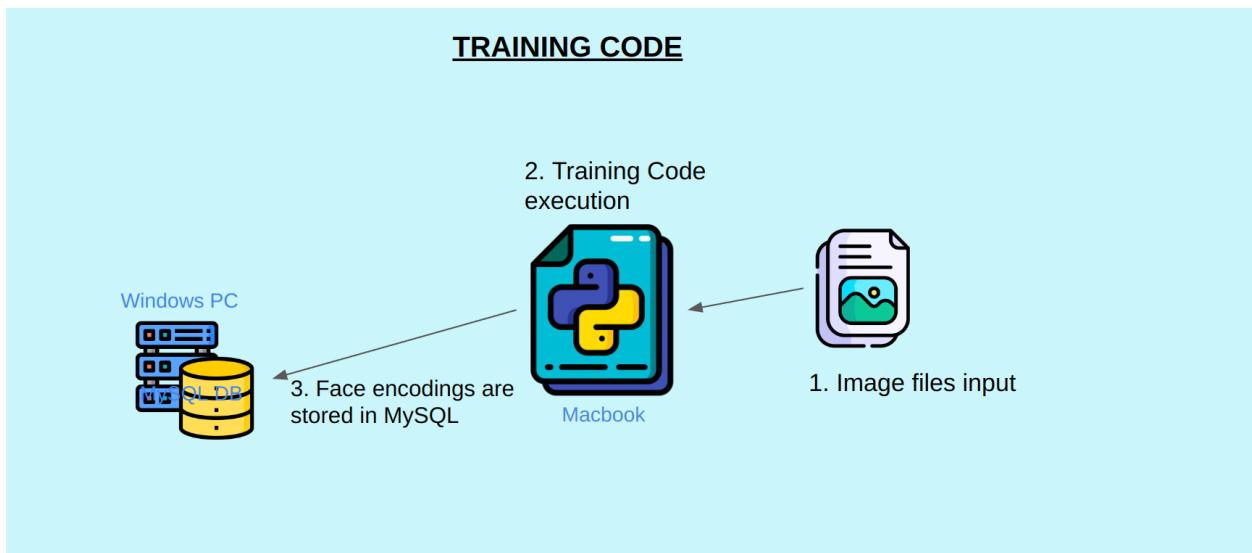
- Face Recognition (Dlib): It's a face recognition library that has a 99.38% accuracy and is built on top of the dlib library ([Link](#))
 - Dlib: DLib is a high performance C++ toolkit which includes many Deep Learning, image processing, and ([Link](#))
- OpenCV (CV2): The larger computer vision library ([Link 1](#), [Link 2](#))
- Numpy: library for arrays and mathematical functions ([Link](#))
- CVLIB: person and object detection library powered by OpenCV ([Link](#))
- This leverages YOLOv4 model trained on COCO dataset capable of detecting 80 common objects in context predicted with an accuracy of 61.7 percent

Solution Overview - Facial Recognition

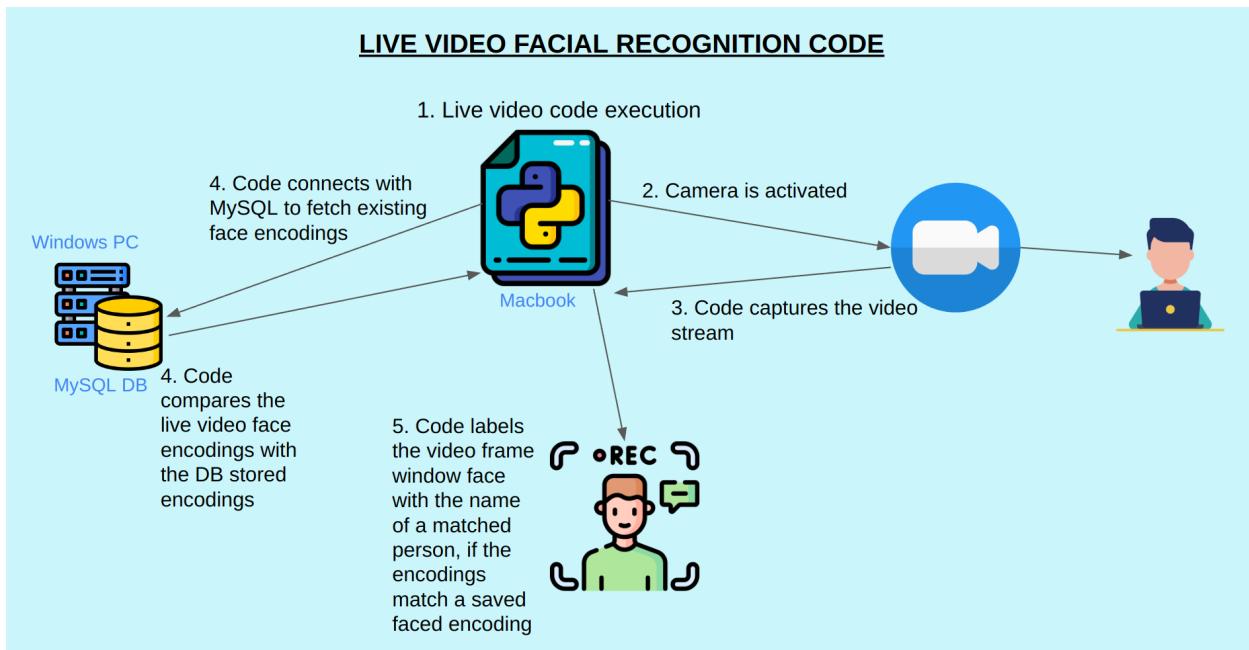
- The key components of the face identification solution consist of:
 - MySQL database
 - Photos with person names as labels
 - Webcam
 - Training code
 - Live video facial identification
- Below are the details of the training solution:
 - MyPHP and MySQL servers were installed on a windows machine
 - The code development and image files are worked on from a Macbook
 - 100+ images of celebs that were downloaded from the internet
 - One of the photos is of me (Author of this presentation: Scott Hameed)
 - Majority of these images were accepted by the algorithm but some generate errors and were removed
 - Usually the errors are due to the frontal face not being clearly visible
 - The code then generates face encodings for the photos
 - The face encodings are then saved in the MySQL database in the windows machine
- Below are the details of the live video solution face detection solution:

- The code enables the webcam
- The code looks for human face(s) in the live video stream generated from the webcam
- It then computes the face encodings for any faces in the live video stream
- It then makes a call into the MySQL server to compare the face encodings from the live video stream
- If it finds a match, it labels it with the name of the file matched with (which in this case is the first and last name of the person matched)
- I then test the code by running and the code and verifying that it recognizes my face and labels it
 - Also, that it does not recognize other people whose face encodings are not in the DB

Conceptual Solution Overview: Facial Recognition Training Code

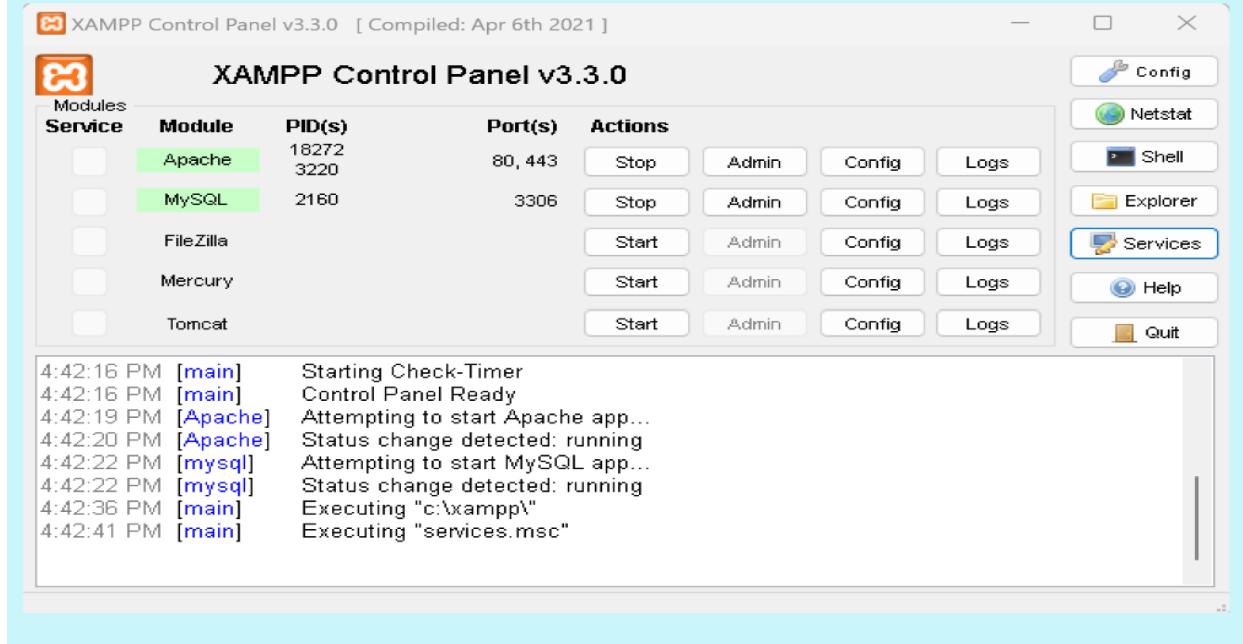


Conceptual Solution Overview: Facial Recognition Live Video Code



Systems Screenshots

XAMPP Control Panel for Apache & MySQL Servers



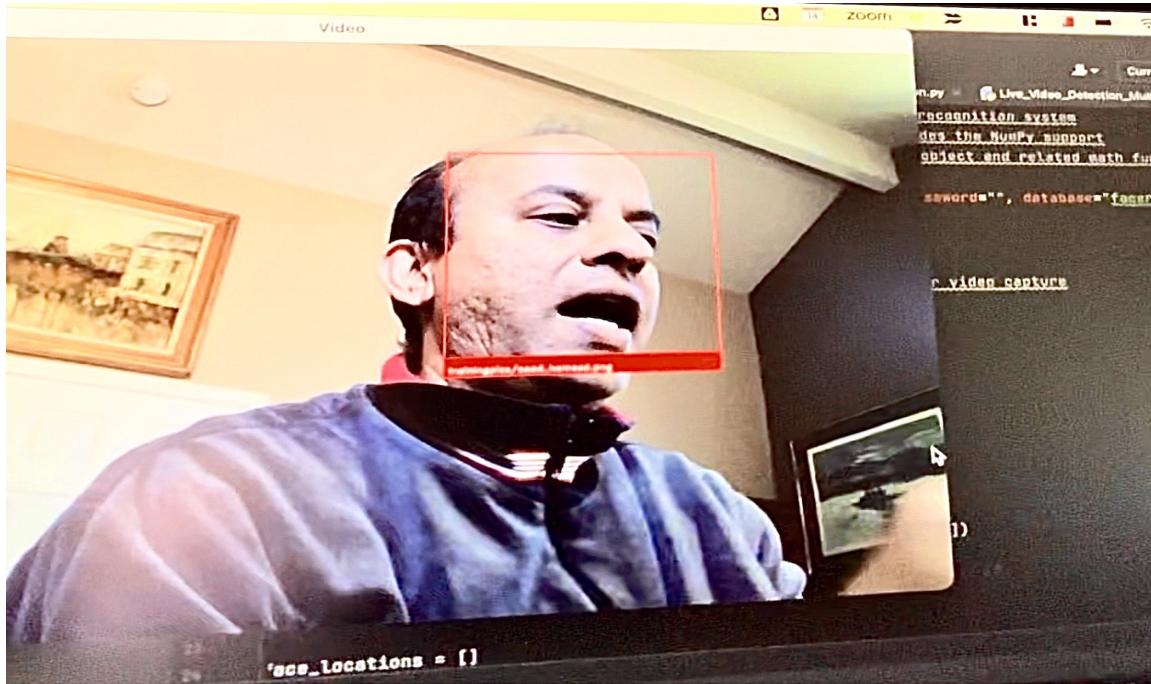
Face Encodings MySQL Table

The screenshot shows the phpMyAdmin interface connected to the MySQL server at 127.0.0.1. The current database is 'facerecon'. The 'encodings' table is selected, displaying 89 total rows. The table has columns: id, filename, and imageencoding.

	filename	imageencoding
1	05571452155709267.jpg	0.03547859609590838, 0.067752...
2	0.14054104985783386.jpg	-0.207166535008011, 0.12140517...
3	3_trainingpics/willem_dafloz.jpg	-0.151171982293606, 0.0812196726922951, 0.09962247...
4	4_trainingpics/stanley_tucci.jpg	-0.03258374705910683, 0.08724205613138292, 0.0404131...
5	5_trainingpics/steve_buscemi.jpg	-0.02370915180385414, 0.0022175791673362255, 0.09621...
6	6_trainingpics/james_jones.jpg	-0.12028404525246811, 0.10391490161418915, 0.0936896...
7	7_trainingpics/robert_downeyjr.jpg	-0.0634055170416832, 0.037374574691057205, 0.1474663...
8	8_trainingpics/samuel_jackson.jpg	-0.14129793849951416, 0.04052489629629135, 0.100137...
9	9_trainingpics/cary_grant.jpg	-0.0714027285578867, 0.053817637264728548, 0.064159...
10	10_trainingpics/leonardo_dicaprio.jpg	-0.062469787895679474, 0.11339661478996277, 0.094752...
11	11_trainingpics/kevin_spacey.jpg	-0.08152744819061661, 0.10583332926034927, 0.0707996...
12	12_trainingpics/al_pacino.jpg	-0.042015016078948975, 0.08296416999886322, 0.120985...
13	13_trainingpics/harrison_ford.jpg	-0.06851059198379517, 0.18414439260959625, 0.0100477...
14	14_trainingpics/alan_rufus.jpg	0.0173657220892617, 0.01425783172565958, 0.037818...

Facial Recognition Project Demo Video ([Link](#))

Facial Recognition Project Demo Screenshot



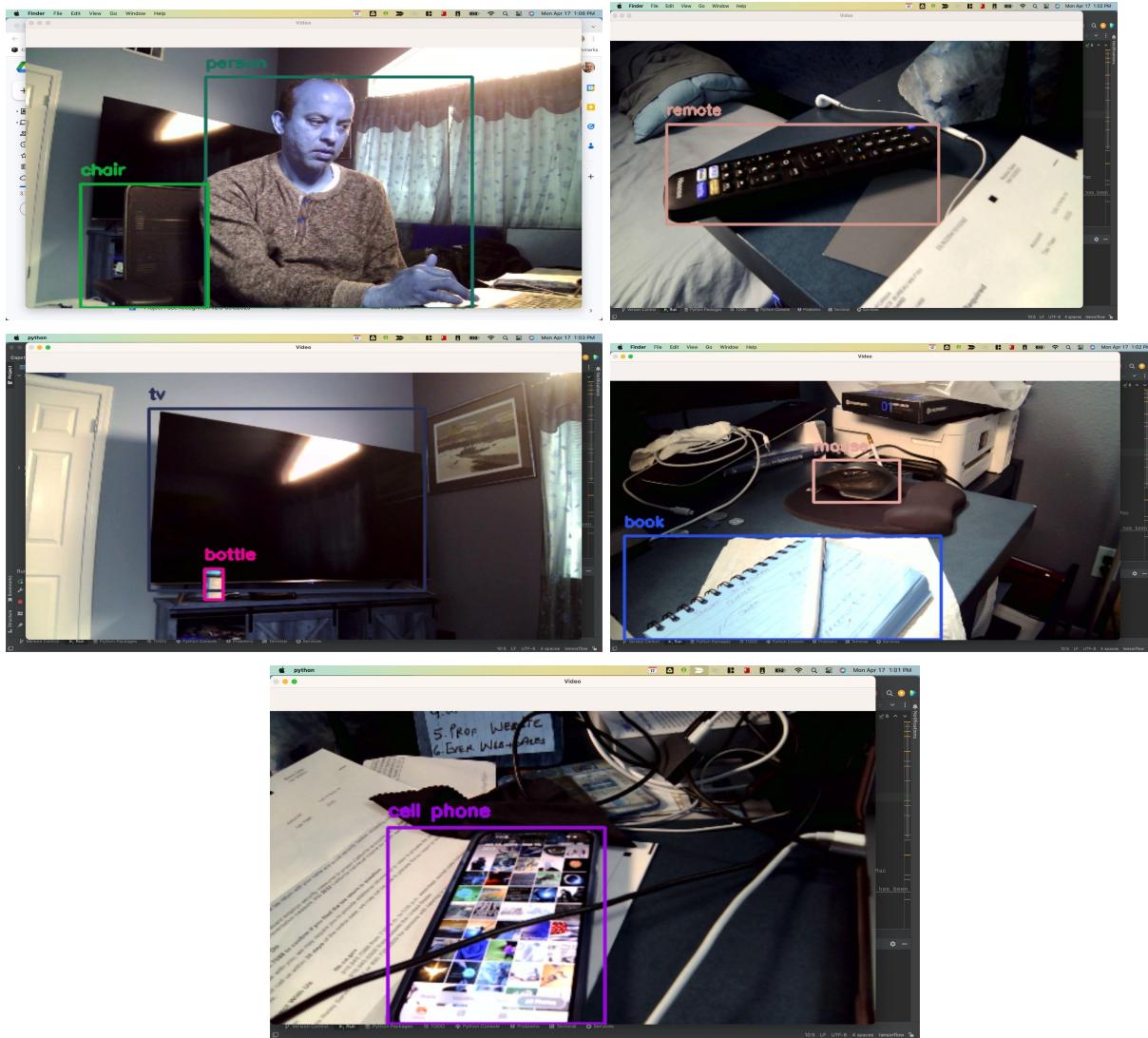
Solution Overview - Object/Entity Recognition

- The key components of the face identification solution consist of:
 - Webcam
 - Live video object identification
- Below are the details of the Live Video solution:
 - The code enables the webcam
 - The code looks for human and other objects(s) in the live video stream generated from the webcam based on a pre-compiled model that has encodings for a person and specific objects
 - If it finds a match, it labels it with the name of the entity (for example: person, cell phone, mouse, keyboard, tv, bottle, etc.)

- I then test the code by running and the code and verifying that it recognizes my person and other objects by labels them

Object/Entity Recognition Project Demo Video ([Link](#))

Object/Entity Recognition Project Demo Screenshots



Key Challenges & Learnings For Future Exhibits:

Key Challenges:

- Dlib library installation ran into several issues; below is the sequence of the

events:

- Installation on Macbook failed
- I then installed a VM using UTM and also Parallels but both had issues with installation of Dlib
- I then changed to a new Ubuntu Linux system but the same issue kept coming up
- I was able to resolve this issue through a set of other app installs and then using Conda for package management

Key Learnings For Future :

- It's critical to not continue investing time into a systems problem that continues to worsen over time (cut losses)
- Always be willing to change an OS/system in sticky situations including using online compute (Colab et al)

Future Enhancements & Optimizations

- Combining the code sets of known person identification with entity/object identification
 - This will ensure we can create a solution where the system recognizes unknown persons for example a person who is not a student of a school
- Update object/entity identification code from CVLib library to Ultralytics YOLOv8 library
 - This will result in improved performance and also identification of larger set of objects
 - The v8 model is able to recognize backpacks ([link](#))
- Leverage YOLOv8 to identify guns or create a new neural network to identify guns as this ensures we can create a school security and risk

identification solution

- Create programmatic solution which identifies variations in sizes of bags to identify risks
- Leverage larger datasets for facial detection training

Project Resource Links

- Github repository homepage ([Link](#))
- Jupyter Notebook ([Link](#))
- Python code file ([Link](#))
- Project Presentation ([Link](#))