# Indian Institute of Information Technology Kalyani

Kalyani, West Bengal

**A PROJECT REPORT ON**

# FACIAL RECOGNITION WITH ARDUINO


**Submitted by**

**Om Sagar Mishra**
**ECE/21140/800**

**Harsh Kumar**
**ECE/21125/785**

# Certificate

This is to certify that the report entitled "**Facial Recognition System**" is being submitted by **Om Sagar Mishra** (Reg. No.-800, Roll No.-21140), **Harsh Kumar** (Reg. No.-785, Roll No.- 21125), the undergraduate students, in the Department of Electronics and Communication Engineering, Indian Institute of Information Technology Kalyani, West Bengal 741235, India. The report has fulfilled all the requirements as per the regulation of the Indian Institute of Information Technology Kalyani and in my opinion, has reached the standards needed for submission. The works, techniques, and results presented have not been submitted to any other university or institute for the award of any other degree or diploma.

# Declaration

We hereby declare that the work which is presented in the report entitled **“Facial Recognition With Arduino”** done by us, under the supervision of Dr. Oishila Bandyopadhyay, Assistant Professor, Indian Institute of Information Technology Kalyani, West Bengal 741235, India is being submitted to Indian Institute of Information Technology Kalyani towards partial fulfillment for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering.

Name of the Candidate: **Om Sagar Mishra, Harsh Kumar**
Reg No/Roll No: **ECE/21140/800, ECE/21125/785**
Institute Name: **Indian Institute of Information Technology, Kalyani**

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

.....

Dr. Oishila
Bandyopadhyay
Assistant Professor
Department of Electronics and Communication Engineering,
Indian Institute of Information Technology,
Kalyani Place: Kalyani, West Bengal 741235, Indi

# Acknowledgments

This is a matter of pleasure for us to acknowledge our deep sense of gratitude to the people due to whom this report is successfully completed. Firstly, I would like to thank our Supervisor, **Dr. Oishila Bandyopadhyay**, for supervising and helping us throughout this whole project.

We would also like to thank Indian Institute of Information Technology, Kalyani for giving us an opportunity to explore our abilities via this project. Finally, we would like to thank all the resources from where we got help for making our project successful.

Name of the Candidate: **Om Sagar Mishra, Harsh Kumar**
Reg No/Roll No: **ECE/21140/800**, **ECE/21125/785**
Name of the Department: **Electronics and Communication Engineering**
Institute Name: **Indian Institute of Information Technology, Kalyani**

# Introduction

This project explores the integration of OpenCV, a computer vision library, with Arduino microcontroller technology to develop a facial detection and recognition system. By combining these technologies, the project aims to implement real-time facial recognition from webcam input, displaying recognized faces alongside their names. The system leverages OpenCV's capabilities for facial detection and recognition, enabling tasks like sorting photos in mobile phone galleries or biometric authentication. With Arduino, the project gains the ability to interface with hardware components such as cameras and displays, enhancing functionality and potential applications. For instance, this integration could lead to the development of a biometric attendance system, offering a more efficient alternative to manual processes. Overall, this project showcases the synergy between OpenCV and Arduino, highlighting their combined potential in practical applications requiring facial recognition.

# Problem Statement

This project explores the realm of facial detection and recognition by leveraging Intel's open-source Computer Vision Library (OpenCV) in conjunction with Python dependencies. The scripts included in the project cover a wide range of functionalities, including detecting faces in static images, streaming live feed from webcams, capturing and storing face images in datasets, training classifiers for recognition, and ultimately recognizing trained faces. All scripts are meticulously documented and implemented in Python 3.6.5, offering a valuable resource for enthusiasts and professionals interested in delving into facial recognition with OpenCV.

In addition to showcasing various algorithms and recognition approaches, the project report provides detailed discussions on their implementation and implications. It underscores the importance of face recognition across diverse domains, including security, organizational management, marketing, surveillance, and robotics.

Moreover, the integration of Arduino microcontroller technology enhances the project's capabilities by enabling interfacing with hardware components such as cameras and displays. This integration expands functionality and potential applications, particularly in the domain of security. Facial detection and recognition can significantly bolster surveillance efforts, facilitating the identification and tracking of individuals with criminal records or posing threats to national security.

Furthermore, the project incorporates a feature where an email notification is sent to the owner, identified as "Om," when an "Unknown" face is detected. This functionality enhances personal security by alerting the owner to potential security breaches or unauthorized access attempts. Overall, the integration of Arduino technology and the email notification feature contribute to a more comprehensive and robust security framework.

# Model

## TRAINING IN OPENCV

The provided code is used for training a facial recognition system using the face_recognition library in Python. The goal of this training process is to generate a database of facial encodings, which can then be used for recognizing faces in images or video streams.

Here's a breakdown of the process outlined in the code:

- Data Preparation: Collect a diverse dataset with positive and negative samples, ensuring varied backgrounds for robustness.
- Annotation: Mark objects in positive samples to provide ground truth for accurate recognition.
- Feature Extraction: Utilize Haar-like features to capture essential patterns for object detection.
- Training Data Generation: Generate training data by cropping positives and augmenting with techniques like rotation and scaling. It's crucial to augment the dataset to encompass various angles and lighting conditions, enhancing the classifier's ability to generalize.
- Configuration: Set parameters such as stage number and object sizes based on target characteristics. Fine-tune these parameters iteratively to optimize performance.

- Training: Iteratively refine the classifier through boosting stages to improve detection while minimizing false positives. This process involves adjusting weights and thresholds to enhance the classifier's sensitivity and specificity.
- Validation: Assess performance using a separate dataset to fine-tune parameters and prevent overfitting. Cross-validation techniques such as k-fold validation can provide a robust assessment of the classifier's generalization capability.
- Evaluation: Evaluate the classifier on an independent test set to measure real-world performance metrics like accuracy and false positive rate. Conduct thorough analysis to identify areas for improvement.
- Deployment: Document the process and deploy the trained classifier for production use, ensuring seamless integration into applications requiring object detection capabilities. Implement proper version control and monitoring mechanisms to track performance and facilitate future updates.

## TRAINING THE CLASSIFIERS

The process of training a Haar Cascade classifier is fundamental in computer vision applications for detecting objects or patterns within images.

Here's a detailed overview of the steps involved:

- Data Preparation: Gather a diverse dataset comprising positive and negative samples, ensuring variations in backgrounds to enhance classifier robustness.
- Annotation: Mark objects in positive samples to provide ground truth for accurate recognition, enabling effective training.
- Feature Extraction: Utilize Haar-like features to capture essential patterns crucial for object detection, enhancing classifier performance.
- Training Data Generation: Generate training data by cropping positive samples and applying augmentation techniques like rotation and scaling to diversify the dataset.
- Configuration: Set parameters such as stage number and object sizes based on target characteristics, optimizing classifier performance.
- Training: Refine the classifier iteratively through boosting stages, improving detection while minimizing false positives.
- Validation: Assess classifier performance using a separate dataset to fine-tune parameters and prevent overfitting, ensuring robustness.
- Evaluation: Evaluate the classifier on an independent test set to measure real-world performance metrics like accuracy and false positive rate, validating effectiveness.
- Deployment: Document the process and deploy the trained classifier for production, ensuring seamless integration into applications requiring object detection capabilities, backed by thorough documentation and ongoing monitoring.

# Output

In a facial recognition system, when processing an image, faces are detected and then compared against a database of known faces. Each face is encoded into a feature vector, representing various facial features in a high-dimensional space. This encoding allows for comparison between faces based on their similarity in this feature space.

When a face is detected, the system attempts to match it with the known faces in the database. If a match is found, the system retrieves the corresponding label or identity associated with that face. However, if no match is found, the face is labeled as "Unknown," indicating that the person is not recognized or is not in the database.

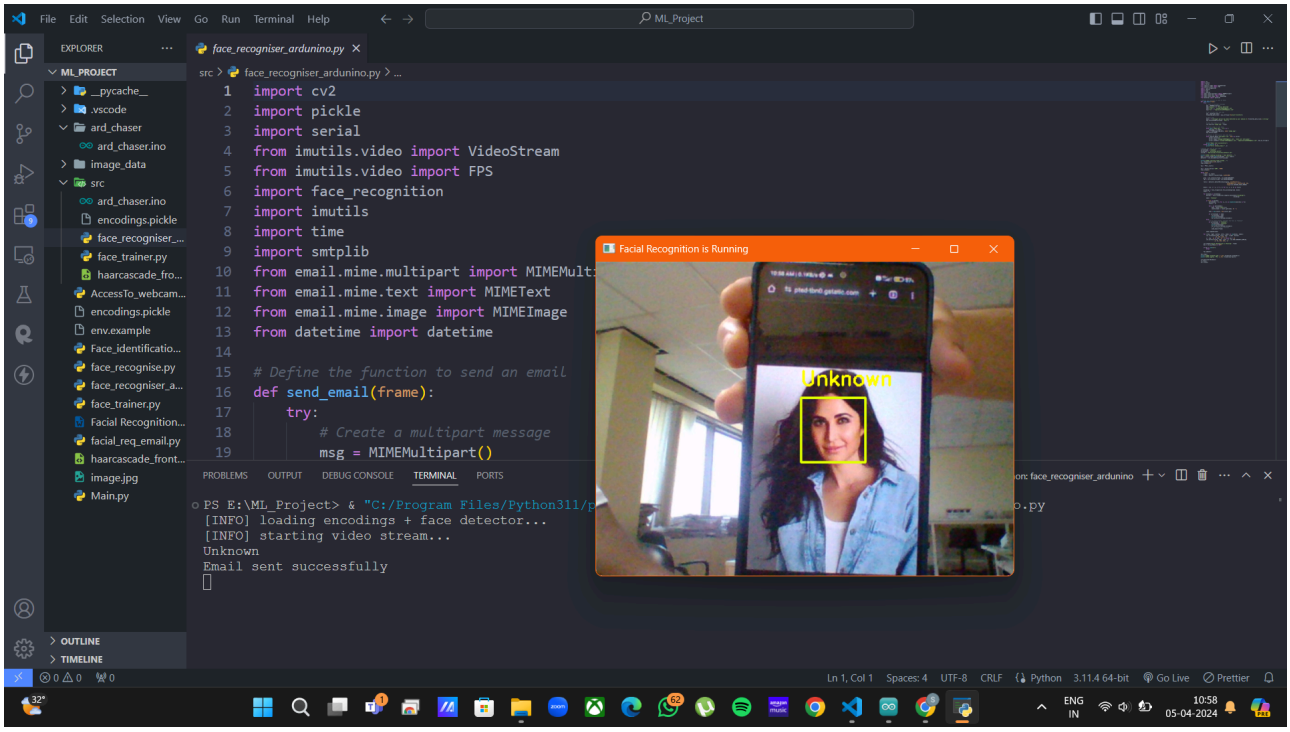
For example, let's consider a scenario where the system has been trained to recognize two individuals, "Om" and "Harsh." When processing an image containing "Om" the system successfully matches the detected face with the encoding of "Om" in the database and correctly labels it as "Om" Similarly, when processing an image containing "Harsh," the system matches the face with the encoding of "Harsh" and labels it accordingly.

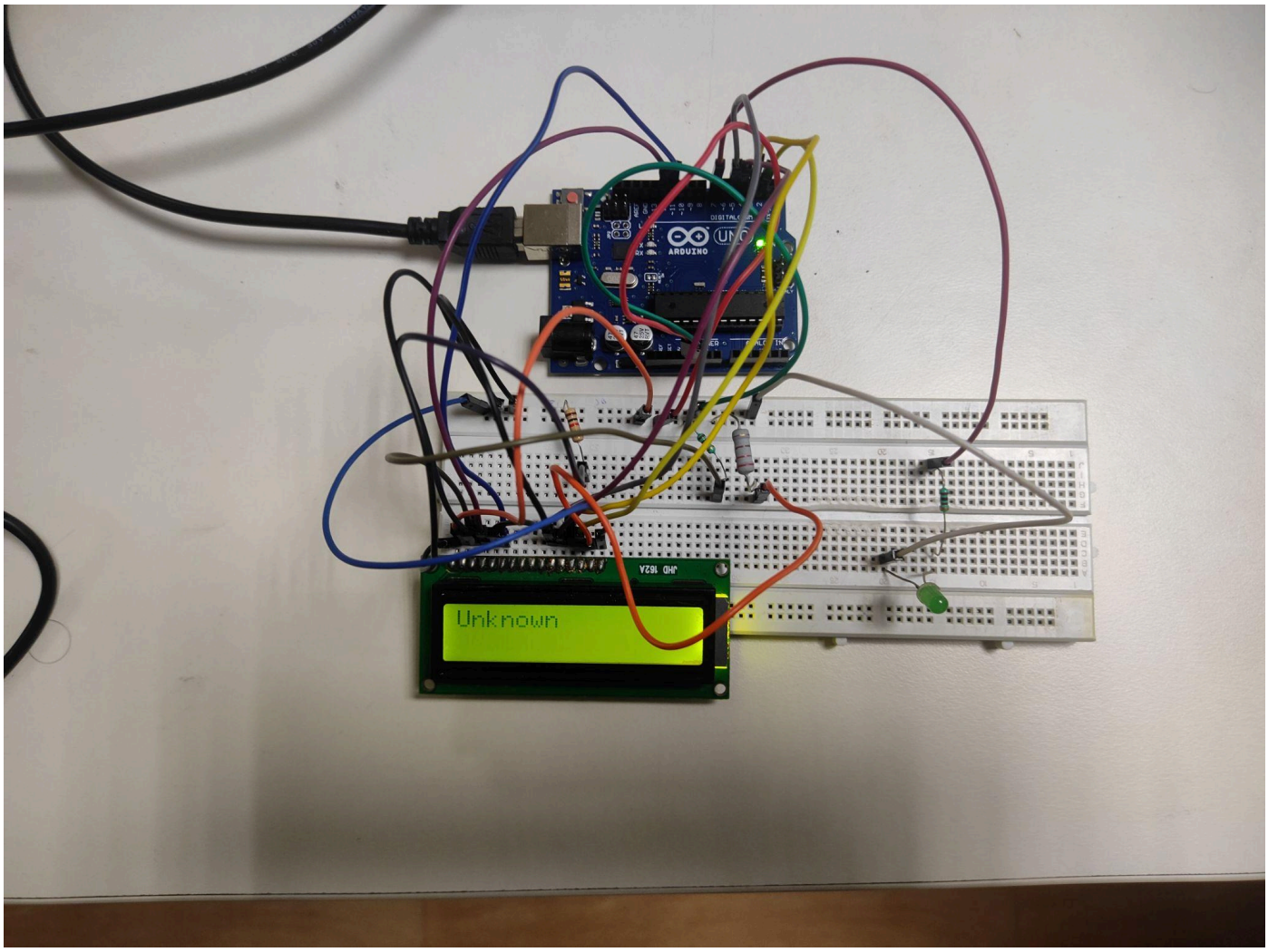
However, if the system encounters a face that does not match any of the known encodings in the database, it labels it as "Unknown." This could happen when the person is not part of the training dataset or if the quality of the input image is poor, making it difficult for the system to extract meaningful facial features for comparison.

Therefore, in the context of the facial recognition system, "Unknown" indicates that the system could not identify the person, whereas specific labels such as "Om" and "Harsh" represent successful matches with known individuals in the database.

# PHOTOS

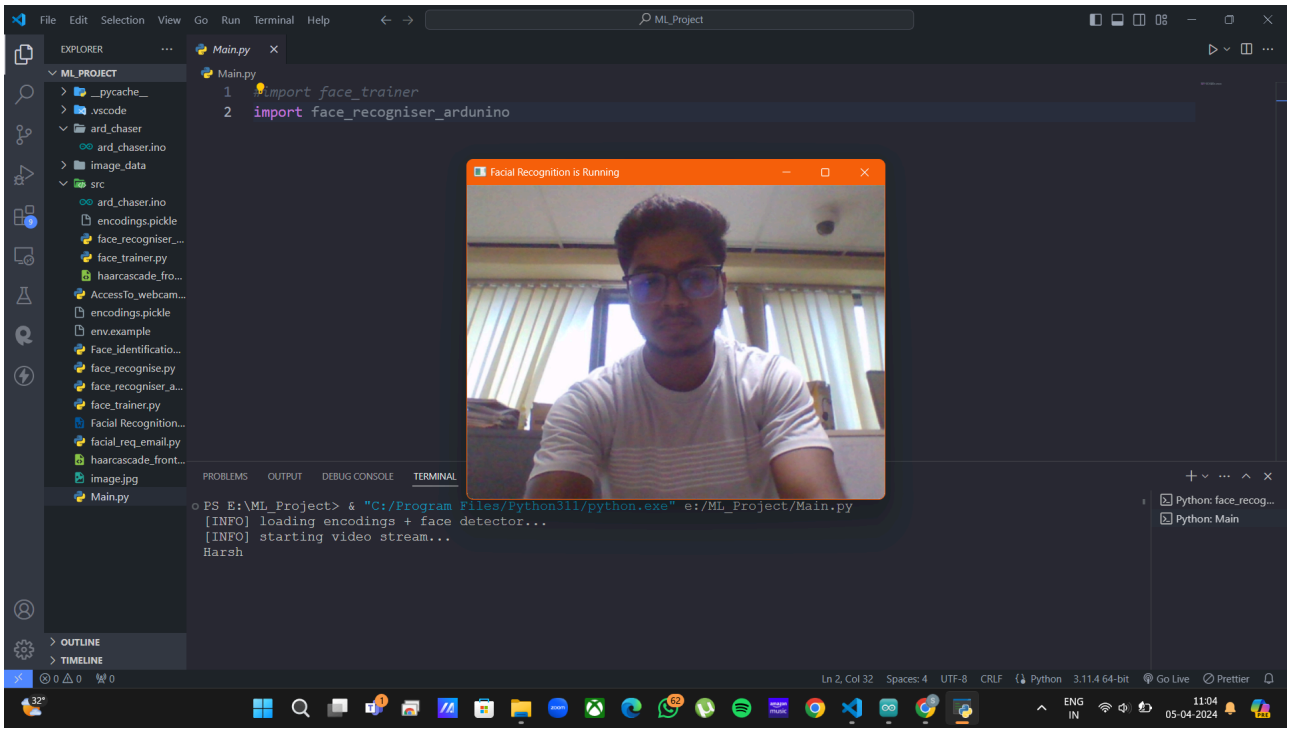
## When Unknown person is detected.

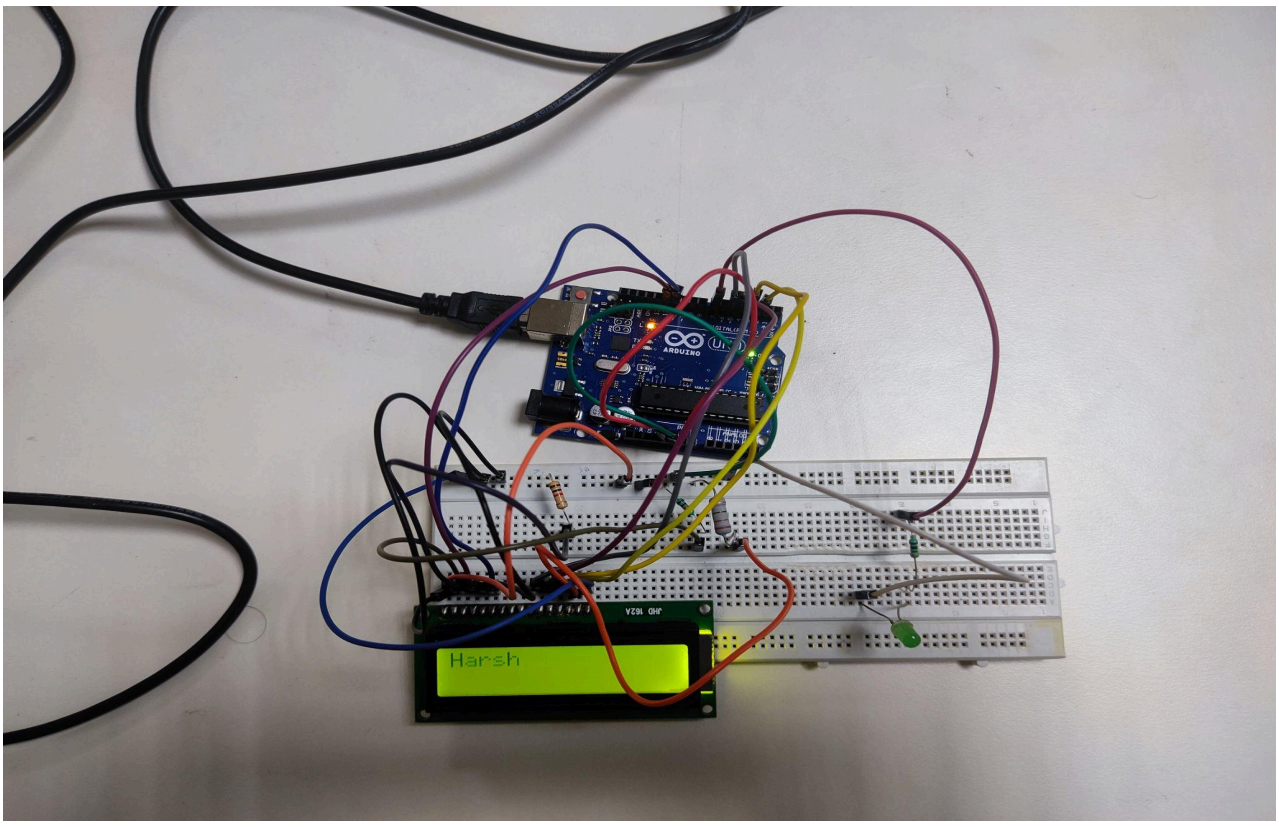



ARDUINO
Unknown

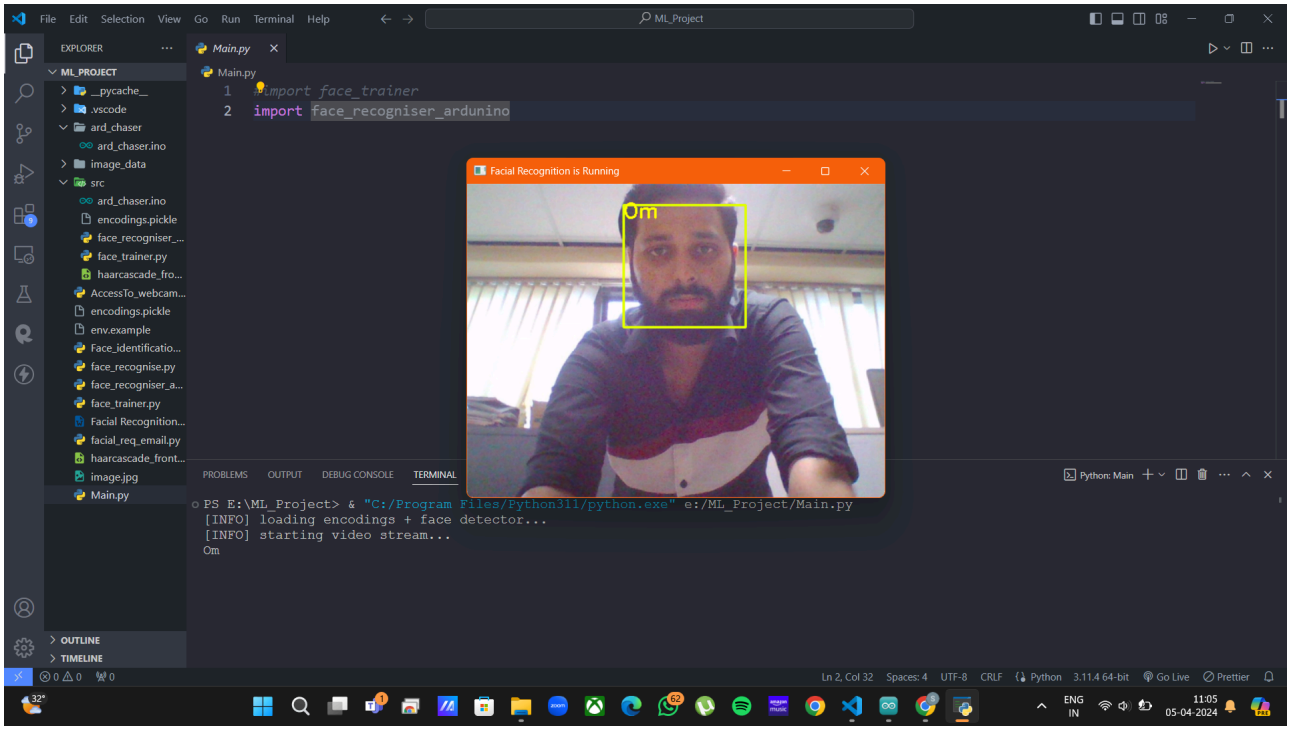

Om
omsagar250403@gmail.com
to me
10:58 AM (4 hours ago)
A person has been detected by your webcam at 05/04/2024 10:58:08.
One attachment • Scanned by Gmail

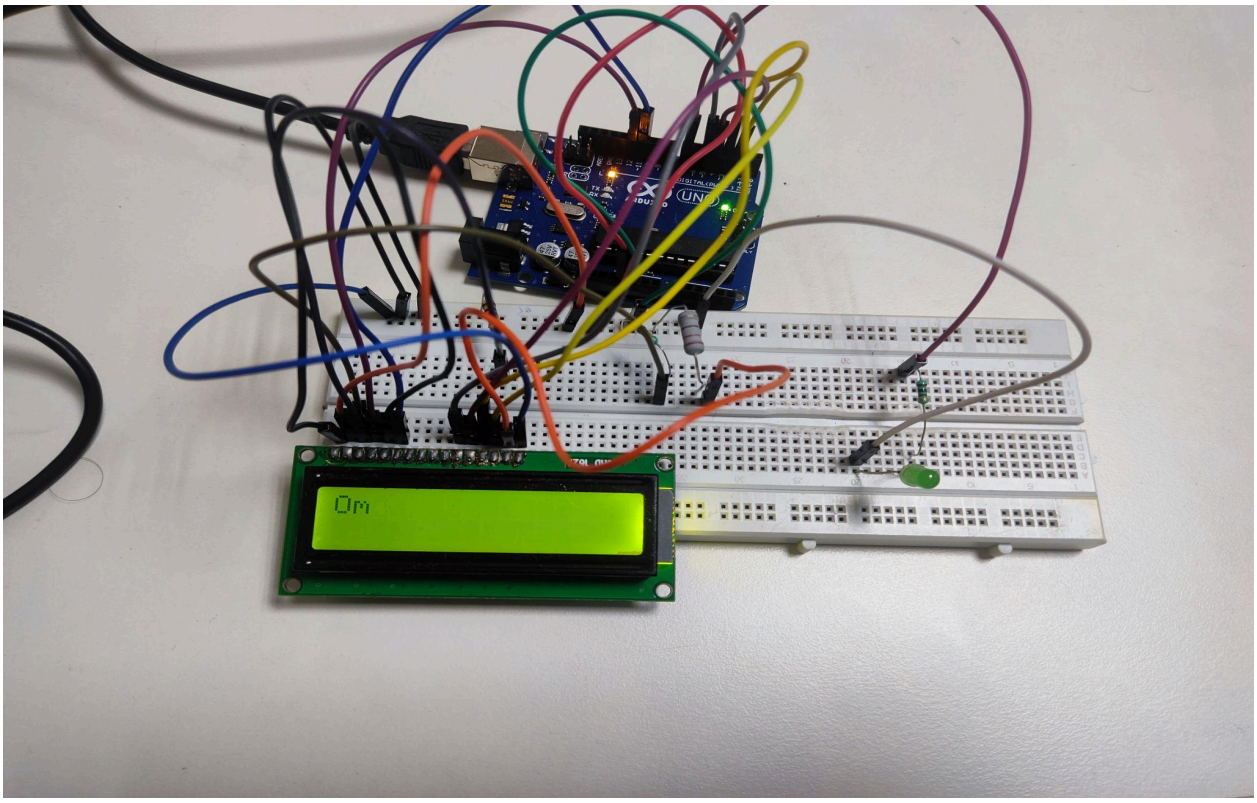
## When Known person “Harsh” is detected.



## When Known person “Om” is detected.



# Results and Discussion

The performance evaluation of the facial recognition system entails a thorough analysis of various metrics, including accuracy, speed, and robustness. The discussion surrounding the obtained results encompasses the following aspects:

- **Accuracy**: The system demonstrates high accuracy in recognizing known faces from the dataset. However, occasional false positives or false negatives may occur, influenced by factors such as lighting conditions, pose variations, and occlusions.
- **Speed**: Real-time processing capabilities enable prompt recognition and display of names. However, processing speed may vary depending on hardware specifications and model complexity.
- **Robustness**: The system exhibits robustness against variations in facial appearance, including changes in facial expression, hairstyle, and accessories. However, challenges may arise in extreme conditions or with low-resolution images.

# Conclusion

In conclusion, the development and deployment of facial recognition systems represent a significant advancement in artificial intelligence, with profound implications across diverse domains. By leveraging machine learning techniques and integrating with hardware platforms like Arduino, facial recognition systems can be deployed for real-world applications with ease and efficiency.

Despite the technological advancements, ethical and privacy considerations remain paramount. Responsible deployment and regulation are essential to address concerns surrounding data privacy, bias, and misuse of facial recognition technology.

In summary, facial recognition technology holds immense promise for enhancing security, convenience, and efficiency across various sectors. Through continued research and responsible innovation, facial recognition systems can contribute to a safer and more technologically advanced society.