

# Documentation

## Table of contents

[1. Overview of Functionality](#)

Results:

[2. Technological Stack and Dependencies](#)

[3. Deployment Requirements](#)

[Link to google colab:](#)

## 1. Overview of Functionality

- This face detection system is designed to recognize and label faces using image input. It leverages facial recognition technology to identify individuals from a database of stored images and corresponding labels. This system can process individual photos or scan images with multiple faces, accurately identifying and labelling known faces based on pre-trained encodings.

### Results:

#### 1 Label:

**Input:** A labelled image (reference). An unlabelled image we want to identify the people in.

**Output:** The unlabelled picture that has rectangles around people's faces and recognizes the person in the reference image and recognized unknowns.

#### Example:

Labelled image:

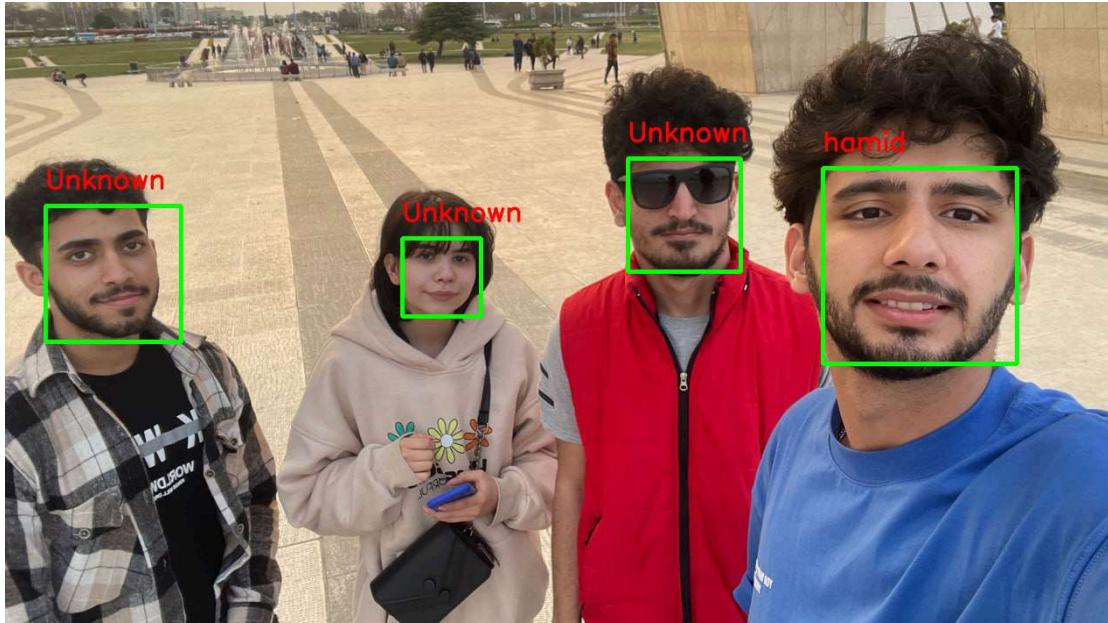
hamid



Prediction Image:



Predicted image:

**Multilabel:**

similar to the last part but gets multiple labelled images.

**Example:**

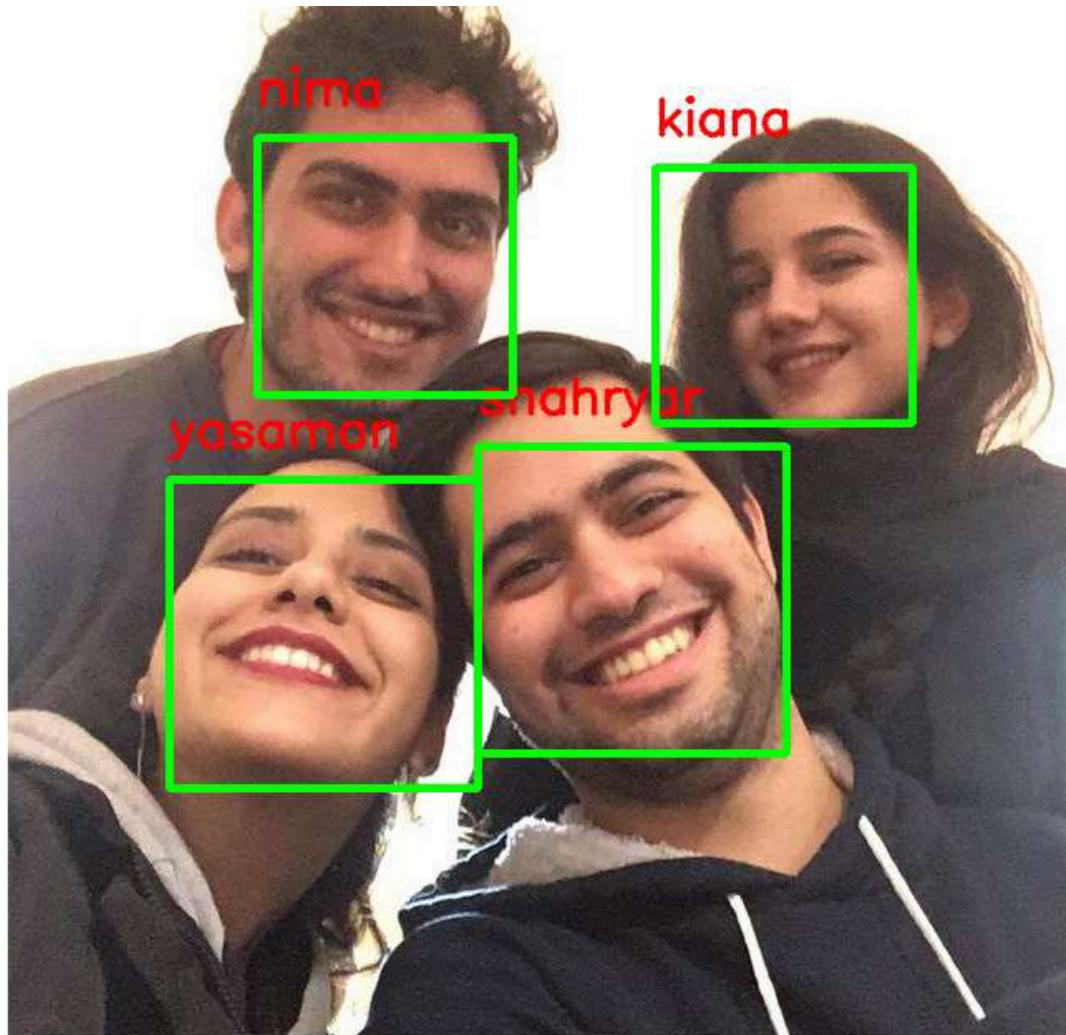
Labelled Images:



Prediction Image:



Predicted Image:

**From Webcam:**

It can also get the prediction or reference image from the webcam.

**Example:****Labelled Image:**



## ▼ Label Reference Image

```
[ ] user_label = "neda"
```

Predicted Image:



## 2. Technological Stack and Dependencies

The project utilizes several key technologies and libraries:

- **face\_recognition:** Utilized for its powerful facial recognition capabilities that allow for easy face detection and comparison.
- **OpenCV (cv2):** Employed for image processing tasks, including reading, displaying, and manipulating images in the notebook.
- **NumPy:** Used for handling data operations, especially for manipulating arrays that store face encodings.
- **Google Colab Patches:** `cv2_imshow` from Google Colab patches is specifically used for displaying images inline within Colab notebooks.

## 3. Deployment Requirements

### Hardware Requirements

- **Camera:** A standard webcam or an advanced camera setup depending on the required accuracy and performance.
- **Computational Device:** A computer with at least an Intel i5 processor (or equivalent) and 8 GB of RAM to handle the computation load.

### Software Requirements

**Operating System:** Compatible with Windows, macOS, or Linux operating systems.

**Programming Environment:** Google Colab or any local Python environment that supports Jupyter Notebooks.

### Dependencies Installation:

Ensure Python 3.6 or later is installed.

Install the necessary libraries using pip:

```
pip install face_recognition numpy opencv-python
```

If you're running the code locally, you need to install the libraries "dlib" and "cmake" in order to install face\_recognition. There might be some issues with installing dlib, such as the python version(probably doesn't work with newer versions of python). You might need to use conda or virtual environment for this to work. Also, you might need to install additional packages for C, C++ programming. If you run into any problems, this website might help:

 [How to install dlib library for Python in Windows 10](#)

If running locally, ensure you have Jupyter Notebook installed:

```
pip install notebook
```

## Installation Guide

- **Google Colab:**

- a. Upload the notebook to your Google Drive.
- b. Open the notebook in Google Colab.
- c. Run the first cell to install the `face_recognition` library.
- d. Make sure to edit the Notebook Settings and change the Hardware accelerator to T4 GPU.

- **Local Setup:**

- a. Clone the repository containing the notebook.
- b. Install the required dependencies as listed above.
- c. Open the Jupyter Notebook to run the project locally.

## Link to google colab:

 [Google Colaboratory](#)

<https://colab.research.google.com/drive/165Db22MuOzG1hiM6jowIH...>





Upload images to the Files section of google colab.