# Face Recognition App with Python

## 1 General Workflow

The workflow of the app is described as follow:

- First, we will take a stream of images from the webcam using OpenCV.
- Then, by using a *face detection model*, we locate faces in the images.
- The *face detection model* cannot recognize whose face in the image. To enable face recognition, we use a *face embedding model* to get an embedding or feature vector of the face.
- After we get the feature vector of the face, we compare this vector with the feature vector of the faces we have in the database. The database contains images of recognized faces that the *face embedding model* uses as the reference to identify faces from the webcam.
- By using a simple distance metric, we can find the closest match to feature vectors of reference images.

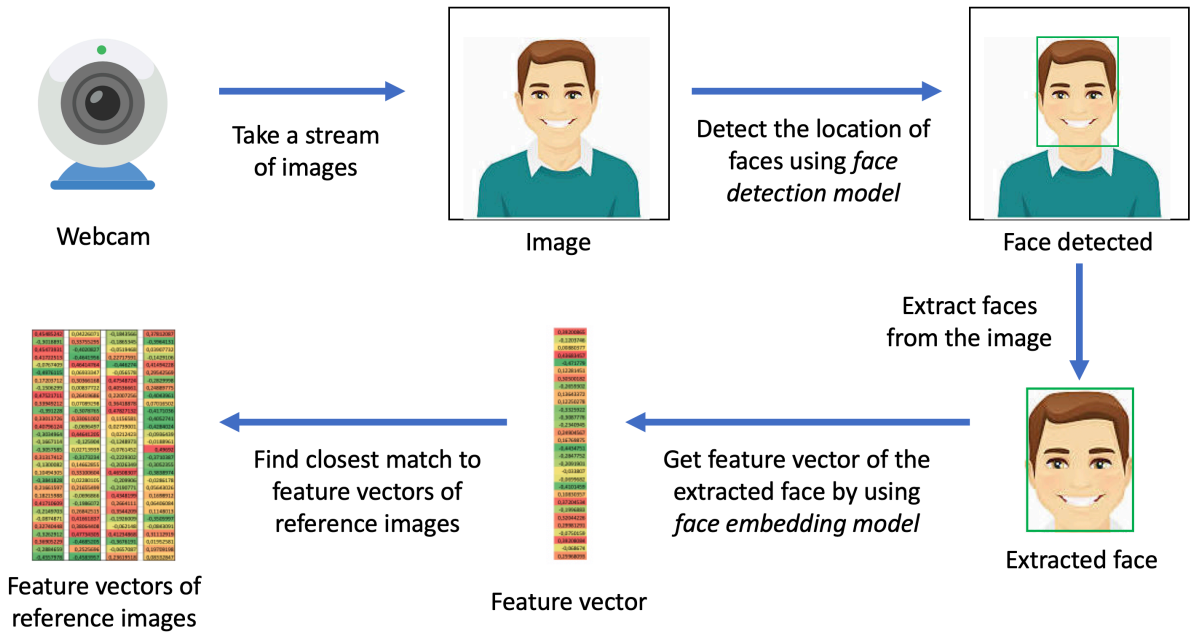


*Figure 1 The workflow of the face recognition app*

Based on the aforementioned workflow, two models are required: a face detection model and a face embedding model.

## 2 Face Detection Model

To build the face detection model, we use `face_recognition` library, which is built using dlib's state-of-the-art face recognition built with deep learning. There are some face detection methods, such as *Haar Cascade Classifiers*, *Histogram of Oriented Gradients* (HOG), and *Convolutional Neural Networks* (CNN). In our app, we will use HOG.

HOG works based on the fundamental notion of extracting feature vector of an image and feed it into a classification algorithm (e.g., support vector machine). The task of the classifier is to determine whether a face (or any object to recognize) is present in a region or not. The features extracted are

the distribution (histograms) of directions of gradients (oriented gradients) of the image. Gradients are typically large around edges and corners and allow us to detect those regions. HOG is originally introduced by Dalal and Triggs (2005). In the original paper, the process was implemented for human body detection, and the detection chain was the following:

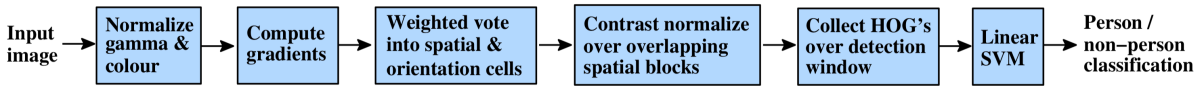


*Figure 2 Feature extraction and object detection chain by HOG (Dalal and Triggs, 2005)*

# 3 Face Embedding Model

The face embedding model uses a modified ResNet-34 classification model trained on 3 million faces. To make it into an embedding model, the last classifier layer(s) are removed. The model transforms the image of human faces into 128D vectors, where images of the same person are mapped near to each other and images of different people are mapped far apart.

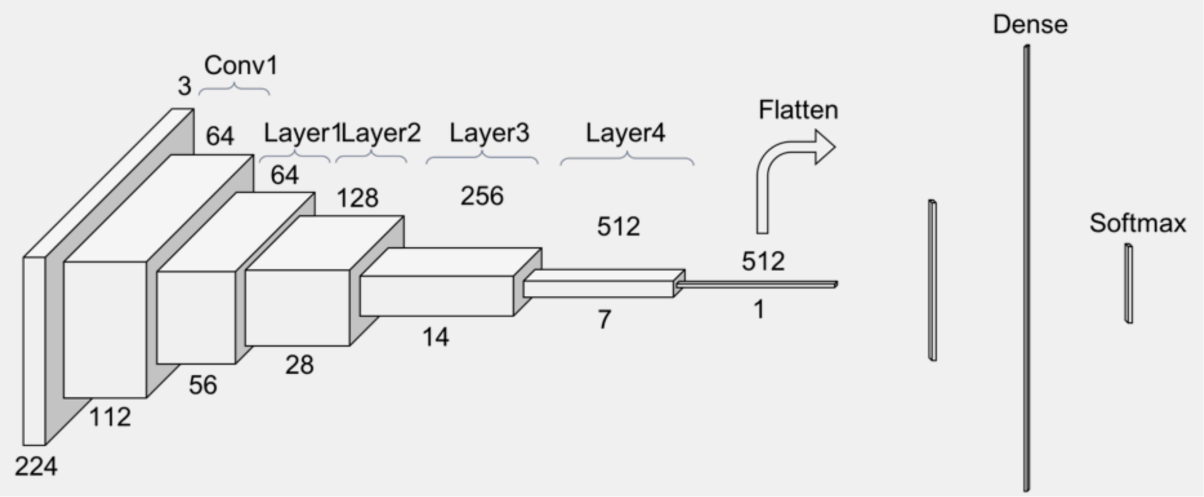


*Figure 3 The architecture of ResNet-34 (He et al., 2016)*

# References


Dalal, N., and Triggs, B. (2005, 20-25 June 2005). Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*:

He, K., Zhang, X., Ren, S., and Sun, J. (2016, 27-30 June 2016). Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*: