# Face Recognition Using Machine Learning

Shivam Kumar (Team 6)
160010010

October 3, 2018

## 1 Introduction and Overview

Face recognition has become an essential part of the unlocking system of mobiles and computers. Easiness to use is accelerating its ubiquity. Facebook uses this technology to tag our friends automatically. Baidu is using it instead of ID cards to allow their employees to enter their offices. This report explains how does this technology work. The method section contains explanations of Histogram of Oriented Gradients (HOG), Face Landmark Estimation, Siamese Neural Network, and Triplet Loss. In the discussion section, we will explain how these methods are combined to create a face recognition system. This section also contains advantages, limitations, and suggested improvements of this system.

## 2 Methods

### 2.1 Histogram of Oriented Gradients (HOG)

HOG is a feature descriptor used for object detection. To detect faces in an image, we follow these steps:

1. Make the image black and white because we don't need color data to find faces.
2. For every pixel, look at the pixels directly surrounding it. Then draw an arrow showing in which direction the image is getting darker. These arrows will show the flow from light to dark across the entire image.
3. Break up the image into small squares of 16x16 pixels each. Then replace each square with the vector sum of all arrows present inside that square.
4. Find the part of the image that looks the most similar to a known HOG pattern extracted from a bunch of other training faces.

### 2.2 Face Landmark Estimation

We will come up with 68 specific points (called landmarks) existing on every face like the top of the chin, the outside edge of each eye, etc. Then we will train a machine learning model to find these 68 specific points on any face. After locating these 68 points, we'll rotate, scale and shear the image to center align the face.

## 2.3 Siamese Neural Network (SNN)

SNN is a class of neural network architectures that contain two or more identical (same configurations with the same parameters and weights) sub-networks. Generally, two identical sub-networks are used to process the two inputs, and another module will take their outputs and produce the final output. In this case, we take two images and find their encodings. Then we compare these encodings to determine the similarity between images. Images of the same person have similar encodings. Using this property, we can compare and tell if the two images are of the same person or not.

## 2.4 Triplet Loss

We can train our SNN by taking an anchor image and comparing it with both a positive and negative image. The dissimilarity between the anchor image and a positive image must be lower than that of between the anchor image and a negative image. So, our loss function will be:

- $L(A, P, N) = max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0)$
- $loss\ function,\ J = \sum_{1im} L\left(A^i, P^i, N^i\right)$

The variable A represents the anchor image, P represents a positive (image of same person) image and N represents a negative image (image of different person). $\alpha$ defines the minimum difference should be present between $||f(A) - f(N)||$ and $||f(A) - f(P)||$.

# 3 Discussion

These steps are followed to create a face recognition system:

1. Find all the faces in the picture using the HOG method
2. For each face, figure out if it is turned in a weird direction and correct them if necessary using Face Landmark Estimation method
3. Train our SNN on labeled data using Triplet loss function
4. Use our trained SNN to pick out encoding of the face and compare this encoding to all the encodings you already have in the database to determine the person's name

We are storing encodings of images and associated IDs in the database. Generally, the dimension of encoding is substantially less than that of the image. So, it will take comparatively less space to store encodings than storing images. It will find the name of the person quickly given image because once we get encoding, the problem will be reduced into a binary search problem of finding this encoding among all the encodings stored in the database.

We need a large number of labeled images and tremendous processing power to train our SNN to get good accuracy. And it is always expensive to collect and label a huge amount of data. We can improve our system by doing the following things:

1. Find a better way to train our SNN which requires less number of labeled images and less processing power.
2. Find a way to collect more labeled data cheaply or find a way to use unlabeled data.
3. Get high-resolution image or images from multiple cameras to improve accuracy.

# 4 References


- Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning - https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

- Convolutional Neural Networks - Coursera - https://www.coursera.org/learn/convolutional-neural-networks