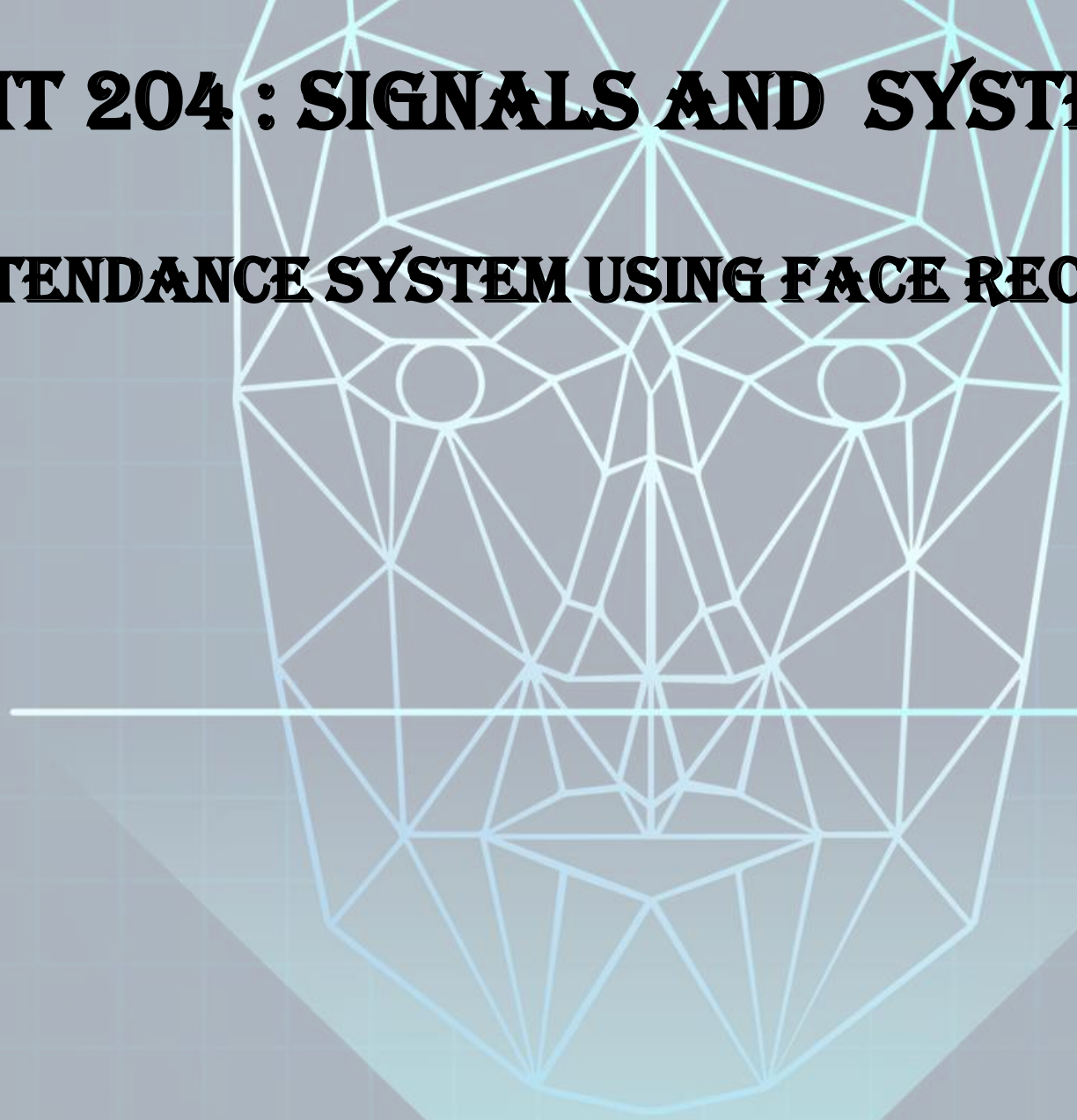
IT 204 : SIGNALS AND SYSTEMS

ATTENDANCE SYSTEM USING FACE RECOGNITION



CHARAN R - 191IT212
V N KARTHIK - 191IT154
RAHUL NAYAKA - 191IT243
RAKESH KALLIMANI - 191IT244

OBJECTIVE : In this project we aim to build an attendance management system with the help of face detection owing to the difficulty in manual as well as other traditional means of attendance system

FACE RECOGNITION is a biometric method of identifying an individual by comparing live capture or digital image data with the stored record of that person

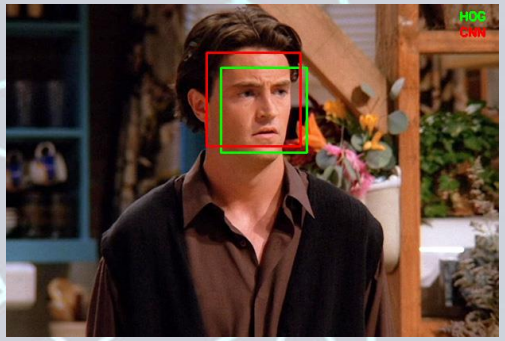
METHOD : Implementing face detection is offered by Dlib* library and uses a concept called Histogram of Oriented Gradients (HOG). This is an implementation of the original Paper by Dalal and Triggs.

The IDEA behind HOG is to extract features into a vector, and feed it into a classification algorithm like a Support Vector Machine for example that will assess whether a face is present in a region or not.

*Dlib is a general purpose cross-platform software library written in the programming language C++

The FIRST STEP in our pipeline is *face detection* .

We will be Encoding a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, we will find the part of the image that most looks like a generic HOG encoding of a face.



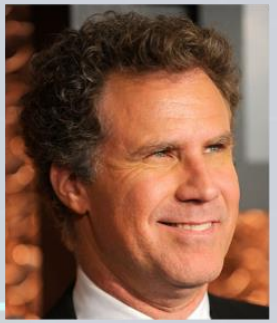
Face detection

Image after using HOG method

SECOND STEP :

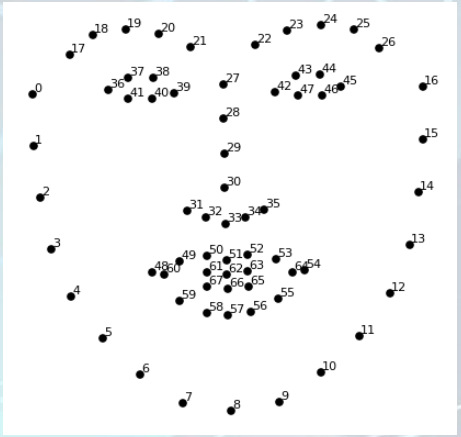
After we isolate the faces in our image ,we now have to deal with the problem that faces turned different directions look totally different to a computer.

To account for this, we will try to warp each picture so that the eyes and lips are always in the sample place in the image. To do this, we are going to use an algorithm called **Face landmark estimation.**
we are going to use the approach invented in 2014 by Vahid Kazemi and Josephine Sullivan.



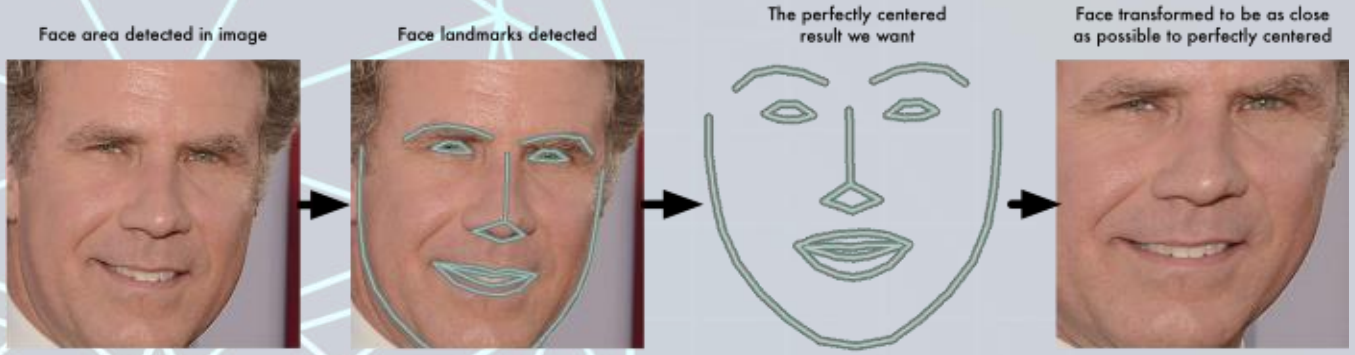
Humans can easily recognize that both images are of Will Ferrell, but computers would see these pictures as two completely different people.

The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face:



The 68 landmarks we will locate on every face. This image was created by Brandon Amos of CMU who works on OpenFace.

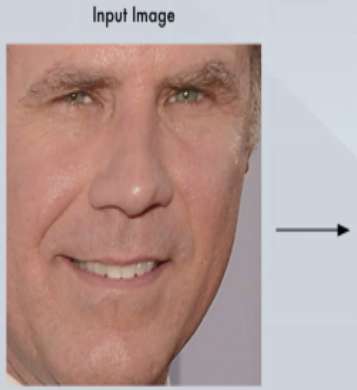
Now no matter how the face is turned, we are able to center the eyes and mouth are in roughly the same position in the image.



THIRD STEP : ENCODING FACES

We will be Passing the centered face image obtained in second step through a neural network that knows how to measure features of the face. We are going to use a pre trained Deep Convolutional Neural Network to generate 128 measurements for each face like the size of each ear, the spacing between the eyes, the length of the nose, etc.

Example :



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.00727777555558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

LAST STEP : FINDING THE PERSON'S NAME FROM THE DATABASE

We will find the person in our database of known people who has the closest measurements to our test image. We will be using a Basic machine learning classification algorithm known as linear SVM classifier.
All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match.

After creating the face recognition program, we will use this to create an attendance project that will use webcam to detect faces and then record the attendance live in an excel sheet.

TIME LINE

STAGE 1

IMPLEMENTING HOG ALGORITHM.

Sep 4- Sep 13

STAGE 2

IMPLEMENTING FACE LANDMARK ESTIMATION ALGORITHM.

Sep 15-Sep 20

STAGE 3

ENCODING FACES TO GENERATE 128 MESURMENTS FOR EACH FACE.

Sep 28 – Oct 3

STAGE 4

Finding the person's name from the DATABASE

Oct 5 – Oct 12

LAST STAGE:

Implementation of face recognition project to create an attendance project

Oct 12 – Oct 27

REFERENCES :

1. Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning by Adam Geitgey

2. Histograms of Oriented Gradients for Human Detection by Navneet Dalal and Bill Triggs

3. One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan KTH, Royal Institute of Technology Computer Vision and Active Perception Lab Teknikringen 14, Stockholm, Sweden

4. FaceNet: A Unified Embedding for Face Recognition and Clustering by Florian Schroff Google Inc ,Dmitry Kalenichenko Google Inc, James Philbin Google Inc.

5. Murtaza's Workshop - Robotics and AI

THANK YOU