CSP525 – Computer Vision

Assignment -1

Face Recognition using HOG-SIFT-CNN

Kartavya Bhatt (201501009), Heet Gorakhiya (201501034), Dipkumar Patel (201501070),
Harshkumar Patel (201501071), Shreejal Trivedi (201501112)

***Abstract*—Here we attempt to solve the basic and well-known face recognition problem in the Computer Vision paradigm. We tackled this problem in three different ways, namely, using Euclidean Distance approach, the statistical learning methods – like HOG, SIFT for feature extraction and SVM and Neural Network as multiclass classifiers. The final approach deals with Deep Learning approach, which constitutes Siamese Neural Networks and Convolutional Neural Network (CNN) The data set consists of 588 images, each with one of the 43 people's faces and on that we apply the learning algorithms to classify and identify each person uniquely. To test the results and accuracy of face detection, we give a different picture of a person's face as an input and find out how many times it identifies the person correctly.**

Keywords—Convolutional Neural Network (CNN), Deep Learning, Histogram of Oriented Gradients (HOG), Siamese Neural Network, Scale Invariant Feature Transform (SIFT).

I. Introduction

Face Recognition is one of the most relevant applications in image analysis. It's a true challenge to build an automated system which equals human ability to recognize faces. The fundamental difference in human and computer based face recognition is that humans can identify a small number of faces to 100% accuracy, but don't do so well with a large number of faces, say, a thousand or so. Computers, on the other hand, have huge computing abilities. That, coupled with a large amount of memory, helps a well-trained computing system to recognize a large number of faces with lesser accuracy than human brain. Due to the recent advancements in the various fields like pattern recognition, bio-metric authentication and computer vision, this problem has recently gained a lot of popularity and is now studied as a principal component of the Machine Learning paradigm.

As this document focuses on the face recognition from pixel based images, we first need to identify the problems that are likely to be encountered while solving the problem. One of the problems is the problem of illumination. The images which have low illumination will have more greyscale pixel values while the ones with high illumination will low greyscale values. This will affect the accuracy of face recognition. The other problem is the pose problem; i.e., the orientation of the face in the image. Different images of the same face with different poses require higher intelligence level to identify as one and the same. The third problem is the background. This problem occurs when the face occupies less part of the image than the background, because then the different background's RGB pixel values will have a great impact on the image identification.

Here, we have compared the three approaches:

- Euclidean Distance Approach
- Statistical Learning Approach:
  - HOG-SIFT features with SVM classifier
  - HOG-SIFT features with ANN classifier
- Deep Learning approach using CNN and Siamese NN

We will compare the performance of these approaches and conclude to find the best working approach in such a small dataset.

II. Euclidean Distance Approach

Introduction:

The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points. It is essentially the root mean square distance between the co-ordinates of a point.

Jiali Yu and Chisheng Li[3] provide a great insight for its implementation.

Implementation:

In our version of Euclidean distance approach for face recognition,

1. We first calculate the SIFT features of the images and keep a record.
2. Then we take the test image and compare it with the SIFT-feature map
3. Then, we try to determine the class of the test image by calculating the Euclidean distance of the SIFT feature map with the actual test image.
4. The feature map with lowest distance represents the feature map for the class of the image.

III. Statistical Learning Approaches

In the statistical learning paradigm for face recognition, we have used Histogram of Oriented Gradients (HOG) and Scale Invariant Feature Transform (SIFT) for feature extraction, and then we use Support Vector Machine and Artificial Neural Network as multi-class classifiers. Before defining the *modus operandi* of the methods, we introduce the feature extraction methods briefly:

A. Histogram of Oriented Gradients (HOG)

Introduction:

HOG is a dense feature extraction method for images. Dense means that it extracts features for all locations in the image (or a region of interest in the image) as opposed to only the local neighborhood of keypoints like SIFT. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms.

Implementation:

1. Extract the HoG features for all the images with following parameters:
    a. 9 bins per cell (360/9 angle for gradient directions).
    b. Pixels / cell = (16*16)
    c. Cells / block = (2*2)

2. After doing above steps on training data we will get a (386, 34596) dimension matrix where 34596 signifies total HoG features of a single image.

3. Apply PCA decomposition on this (386, 34596) vector with 98% of variance which will produce (386,350) matrix. This will be an input as a training data which will be fed into ANN and SVM

B. Scale Invariant Feature Transform (SIFT)

Introduction:

The scale-invariant feature transform (SIFT) is an algorithm used to detect and describe local features in digital images. It locates certain *key points* and then furnishes them with quantitative information (descriptors) which can for example be used for object recognition. The descriptors are supposed to be invariant against various transformations which might make images look different although they represent the same object(s). Scale Invariant Feature Transform is an advanced version of Harris Corner detector which is helpful in detecting the corners and abrupt changes in the images at different scales. SIFT descriptor describes the local features of image patches, so it can be helpful in recognition tasks.

Implementation:

Broadly, the SIFT feature extraction is implemented in 5 steps:

1. Scale Space Extrema Detection
2. Keypoint Localization
3. Orientation Assignment
4. Keypoint Descriptor
5. Keypoint Matching

Procedure for generating fix basis vector:

1. Extract the local features/descriptors of each and every image from our dataset.
2. Stack all the feature/descriptors in one single matrix. In total there are (91129, 128) dimensional matrix generated from our 376 training images.
3. Apply K-means algorithm on this stack which we generated with number of clusters equals to 55 clusters.
4. Allocate specific cluster to each and every descriptor from the stack
5. Generate the histogram of an image counting the number of times particular cluster appeared in the image.
6. Normalize the histogram generated using StandardScaler.
7. Above generated histogram will be of dimension (386, 55), where each image is represented by the number of clusters and their counts.
8. Y_Test_new is used a label vector of each image of dimension (386, 1).

C. Support Vector Machine

Introduction:

Support Vector Machine is a classification algorithm based on a supervised learning structure, which means the input training data needs to be labelled for it to work. The SVM learns a hyperplane based model, which can also be said to be a linear classification model in two dimensions[2]. There are four types of SVMs, namely – The Maximal Margin Classifier, Kernelized MMC, Soft-margin MMC and the Kernelised Soft-margin MMC. The most commonly and widely used is the Kernelized Soft-margin Maximal Margin Classifier, which is also used in the keras library for SVM implementation[5].

Implementation:

1. As discussed in the HOG section, the generated histogram of dimension (386, 55) is used as a training data which is fed into SVM classifier.
2. Parameters of SVM are linear kernel with regularization parameter set to C=3.
3. Results were calculated with different kernel such as rbf, polynomial and linear, and their F1 score and model score were calculated for analysis

IV. Deep Learning based Approach

In the Deep Learning paradigm, we have implemented two methods for face recognition:

A. Convolutional Neural Network

Introduction:

Convolutional Neural Networks use a similar method to Statistical learning approaches, that is, it consists of two parts:

1. Feature Extraction: For this convolutional and maxpooling layers are there. The convolution layers extract the significant features of the image and the MaxPooling layers reduce the dimensions of the image to retain the most important information in the images
2. Multiclass Classification: For this purpose, the convolution and MaxPooling layers are followed by the normal Dense layers (accompanied by dropout layers for increasing the accuracy), which is just a normal Neural Network. This helps in multiclass classification.

Implementation:

Using CNN for Face Recognition, we have image 40 different people in our dataset with different expressions:

1. First we have cut the faces from the image using face detection on the given dataset. Input size of images is: 100x100x3
2. Next, for feature extraction and dimensionality reduction, we introduce three layers of Convolution and MaxPooling2D. After this, the dimensions of the image reduce to: 12x12x3
3. After this, we flatten these image vectors to introduce them as input to the Dense layers for classification.
4. The flatten layer is followed by a dense layer of 512 neurons.
5. The output layer is of 41 nodes, because there are 41 classes in our dataset.

B. Siamese Neural Networks

Introduction:

In our case, given very few examples per class, It’s extremely hard to train neural network without overfitting to train dataset. For preventing this, state-of-art Siamese Neural Network can be used for finding similarities between two classes. Here, we refer classes as facial image of people. Siamese Neural Network loss function encourages to find similarity between same classes.Siamese Neural Network learn function f : D→R where is high dimensional and D is low dimensional vector. If input R1 and R2 are from same class then resulting output D1 and D2 are close in terms of euclidean distance. Siamese Network trained using contrastive loss[1] function.

Implementation:

$$(1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

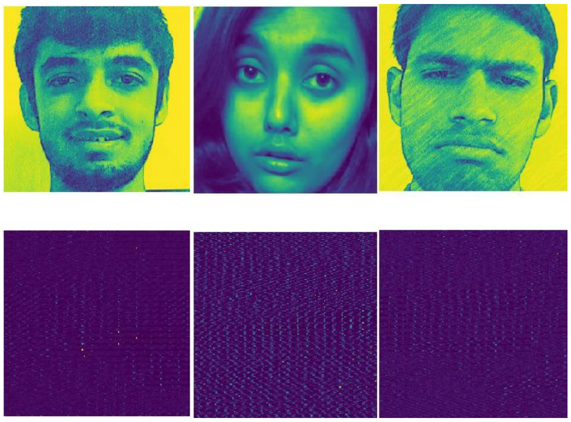
1. Here, Dw is Euclidean distance between two output dimensions. Y is 0 if both of inputs are from different class and 1 if both of inputs are from same class.
2. Implementation was done using Pytorch[2] and we used Google Colab for training.
3. While training Siamese Neural Network, loss rapidly goes down and is unable to reduce further more.
4. The architecture comprise of 3 convolution layer having filter size 3x3 and filters 32x32x16 respectively followed by 3 Feed forward Layer.
5. Last Feed forward Layer has output of 128 dim.

V. Results.

A. *Statistical Learning – HOG/SIFT + SVM/ANN*

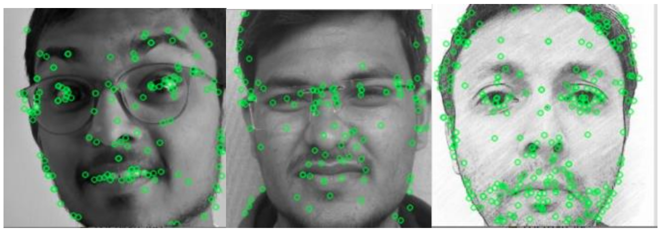
1) HOG Features:

As we follow the steps given in the section IIIA, we get the following HOG feature maps of respective images:



2) SIFT Features:

As we follow the Implementation steps in section IIIB, we get the results as shown below – these are SIFT features extracted from the respective images:

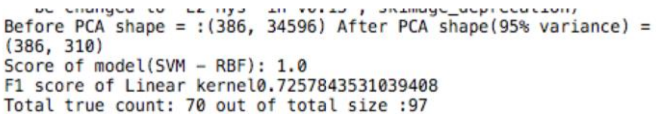


These Extracted features from HOG and SIFT are inserted into the SVM and ANN classifiers:

3) Using SVM classifier with HOG features:

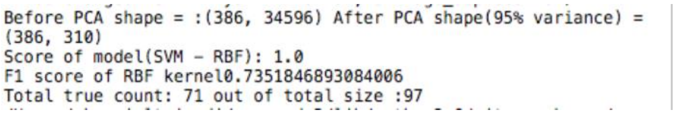
1. SVM parameters: Training data :- (386,350) optimized PCA vector, Label data:- Y_train, kernel:- linear, Regularization parameter:- C = 9
2. Analysis were done by implementing the different kernels and finding their f1_score.

Linear Kernel:



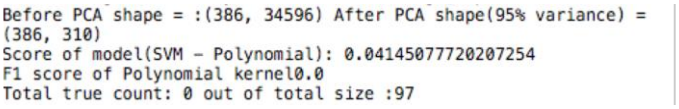
```
  be changed to 'L2-Hys' in v0.15', skimage_deprecation)
Before PCA shape = :(386, 34596) After PCA shape(95% variance) =
(386, 310)
Score of model(SVM - RBF): 1.0
F1 score of Linear kernel0.7257843531039408
Total true count: 70 out of total size :97
```

RBF Kernel:



```
Before PCA shape = :(386, 34596) After PCA shape(95% variance) =
(386, 310)
Score of model(SVM - RBF): 1.0
F1 score of RBF kernel0.7351846893084006
Total true count: 71 out of total size :97
```

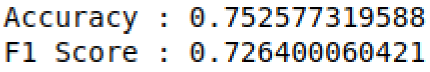
Polynomial Kernel:



```
Before PCA shape = :(386, 34596) After PCA shape(95% variance) =
(386, 310)
Score of model(SVM - Polynomial): 0.04145077720207254
F1 score of Polynomial kernel0.0
Total true count: 0 out of total size :97
```

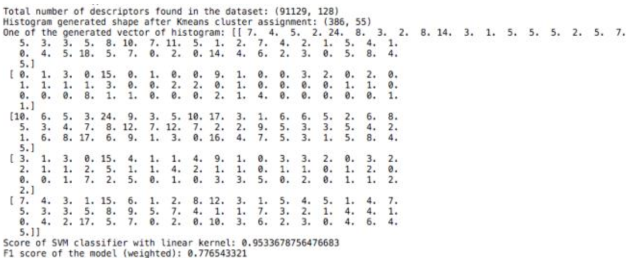
4) Using ANN classifier with HOG features:

Using the Feed Forward ANN as a classifier for the HOG features gives the following accuracy:



```
Accuracy : 0.752577319588
F1 Score : 0.726400060421
```

5) Using SVM classifier with SIFT features:



```
Total number of descriptors found in the dataset: (91129, 128)
Histogram generated shape after Kmeans cluster assignment: (386, 55)
One of the generated vector of histogram: [[ 7.  4.  5.  2. 24.  8.  3.  2.  8. 14.  3.  1.  5.  5.  5.  2.  5.  7.
   5.  3.  3.  5.  8. 10.  7. 11.  5.  1.  2.  7.  4.  2.  1.  5.  4.  1.
   0.  4.  5. 18.  5.  7.  0.  2.  0. 14.  4.  6.  2.  3.  0.  5.  8.  4.
   5.]
 [ 0.  1.  3.  0. 15.  0.  1.  0.  0.  9.  1.  0.  0.  3.  2.  0.  2.  0.
   1.  1.  1.  1.  3.  0.  0.  2.  2.  0.  1.  0.  0.  0.  0.  1.  1.  0.
   0.  0.  0.  8.  1.  1.  0.  0.  0.  2.  1.  4.  0.  0.  0.  0.  0.  1.
   1.]
 [10.  6.  5.  3. 24.  9.  3.  5. 10. 17.  3.  1.  6.  6.  5.  2.  6.  8.
   5.  3.  4.  7.  8. 12.  7. 12.  7.  2.  2.  9.  5.  3.  3.  5.  4.  2.
   1.  6.  8. 17.  6.  9.  1.  3.  0. 16.  4.  7.  5.  3.  1.  5.  8.  4.
   5.]
 [ 3.  1.  3.  0. 15.  4.  1.  1.  4.  9.  1.  0.  3.  3.  2.  0.  3.  2.
   2.  1.  1.  2.  5.  1.  1.  4.  2.  1.  1.  0.  1.  1.  0.  1.  2.  0.
   0.  0.  1.  7.  2.  5.  0.  1.  0.  3.  3.  5.  0.  2.  0.  1.  1.  2.
   2.]
 [ 7.  4.  3.  1. 15.  6.  1.  2.  8. 12.  3.  1.  5.  4.  5.  1.  4.  7.
   5.  3.  3.  5.  8.  9.  5.  7.  4.  1.  1.  7.  3.  2.  1.  4.  4.  1.
   0.  4.  2. 17.  5.  7.  0.  2.  0. 10.  3.  6.  2.  3.  0.  4.  6.  4.
   5.]]
Score of SVM classifier with linear kernel: 0.9533678756476683
F1 score of the model (weighted): 0.776543321
```

6) Using ANN classifier with SIFT features:

1. RBF kernel:
   a. Model score: 0.28763
   b. F1_score: 0.13342
2. Linear kernel:
   a. Model score: 0.95336
   b. F1_score: 0.77654
3. Polynomial kernel:
   a. Model score: 0.67753
   b. F1_score: 0.56432

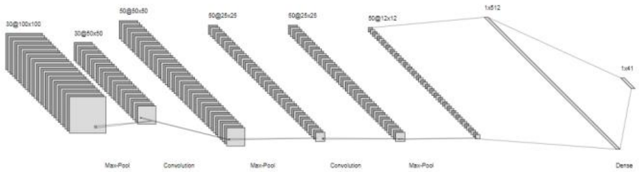
B. *Deep Learning – CNN and Siamese Neural Network*

1) Convolutional Neural Network:

Classification Report

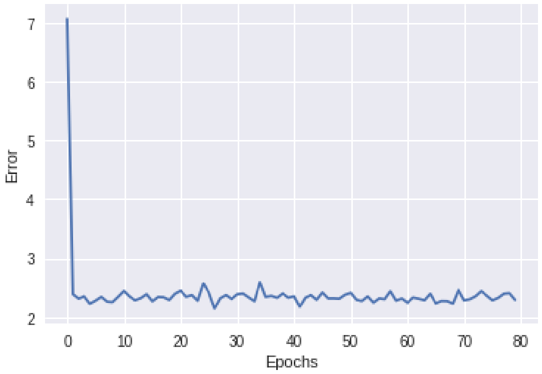
	precision	recall	f1-score
avg / total	0.85	0.82	0.81

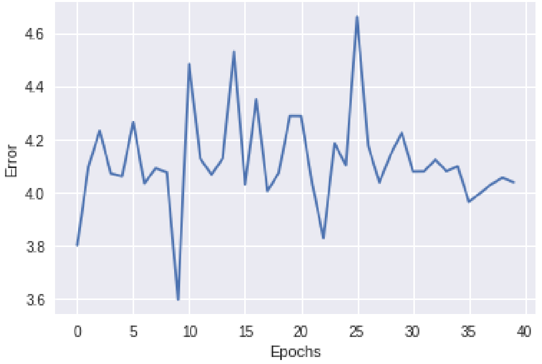
Architecture:



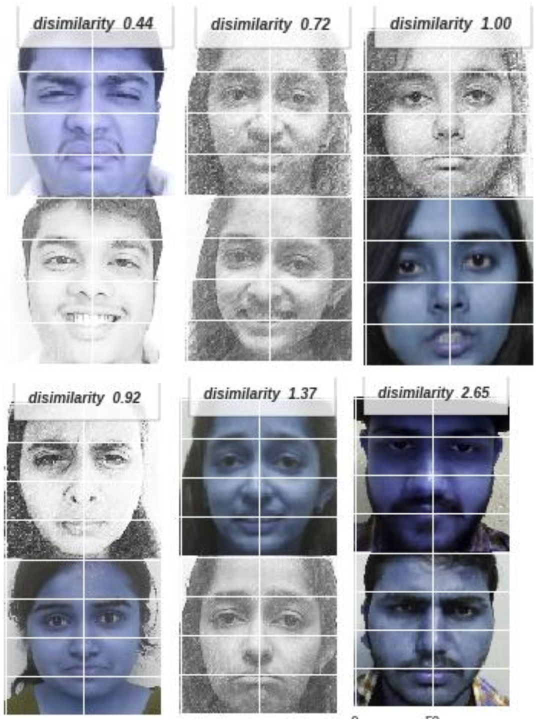
2) *Siamese Neural Network:*

Training – Epochs vs Learning





Dissimilarity Indices:



REFERENCES

[1] R. Hadsell, S. Chopra and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, 2006, pp. 1735-1742.

[2] https://pytorch.org/

[3] Yu, Jiali, and Chisheng Li. "Face recognition based on euclidean distance and texture features." *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on*. IEEE, 2013.

[4] Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. "Deep Face Recognition." *BMVC*. Vol. 1. No. 3. 2015.

[5] OpenCV Documentation: https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html