# A Comparative Analysis: Labeled Faces in the Wild Database


Divyanshu Kumar Singh
2017048
IIIT-Delhi, India
divyanshu17048@iiitd.ac.in

Sankalp Agrawal
2017363
IIIT-Delhi, India
sankalp17363@iiitd.ac.in

Harshit Rai
2017152
IIIT-Delhi, India
harshit17152@iiitd.ac.in



## 1. Abstract

This paper provide basic comparison on some simple machine learning techniques such as Logistic Regression, SVM, Neural Network and Convolution Neural Network to compare each of their performance over the famous de-facto dataset ***Labelled Faces in the Wild***. Since this is the de-facto dataset and is majorly used to test performance of the algorithms, our approach is not to compete with already implemented results.


## 2. Introduction

The quality of camera lenses has evolved over the past decade, starting from VGA to WideAngle, High-Resolution Cameras, etc and this has not only resulted in competition between different camera lens manufactures but also, the way images are now being processed has gained a lot of attention. In computer a huge amount of research is being done to analyze images, be it in the field of Computer Vision, Pattern Recognition, Image Analysis or Biometrics. One such problem that is being researched is Facial Recognition, and the results are also available for the same. But the question remains 'Why' and 'Which' algorithm is best for a particular dataset, being said that there is no universally accepted algorithm for Facial Recognition. In this particular study, we will try to answer these two 'W's' for the 'Labeled Faces in the Wild ' datasets by comparing the various algorithms and we will try to implement an algorithm to compare our results.

## 3. Related Work

A lot of work in computer vision has focused on extracting various feature from the dataset, and hence there are lot pre-defined models for the task of facial recognition, and this is one of the most challenging task. But one of the major concerns is the constrained datasets that those model use. A couple of studies were conducted using constrained datasets only, and then in the year 2007 Labelled Face in the Wild brought a new paradigm into research in facial recognition[1]. The data provided a fairly large dataset with sample of images from real life scenario condition.

A previous study which has compared the results of various kernel in SVM states that RBF and Polynomial performed better than Linear, but one drawback of this study was the size of the dataset, they used 400 image of 40 different individuals[2]. Another study which has only two classes for classification with 38 image for first class and 28 images from second class, reported that non linear model performed better than the linear model[3]. Gradually as neural network and deep learning enhanced the performance of various learning task, researchers started analyzing their neural network on this particular dataset, and some examples with their performance are as follows, Facenet[4] with accuracy of 99.63%, Cosface[5] with an accuracy of 99.73% and ArcFace[6] with an accuracy of 99.83%.

## 4. Methodology

We have used *Labelled Faces in the Wild(LFW)*[1] dataset for our study, which was published by University of Massachusetts[2]. This dataset contains 13,223 images of 5749 difference individual celebrity. There are total 1680 of the total people have two or more than two pictures in the data set. For our analysis we have, used only those image which have at least 40 or more face in the dataset. The only reason to do this was to attain reasonably well data set. The total number of samples in our dataset: 1867, total number of features: 2914, number of classes(number of people with more than 40 images): 19 A small sample of the dataset is shown in the Figure1

For our analysis, we took 80% of the data in the training set and 20% of the data in test set.

### 4.1. Evaluation metrics used

We use classification reports(precision and recall values) and visualisations like ROC curve for evaluating our models.

[1]http://vis-www.cs.umass.edu/lfw/index.html
[2]http://vis-www.cs.umass.edu/


Figure 1. Sample Dataset

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_547 (Conv2D)          (None, 48, 35, 64)        640
_________________________________________________________________
max_pooling2d_408 (MaxPoolin (None, 24, 17, 64)        0
_________________________________________________________________
conv2d_548 (Conv2D)          (None, 22, 15, 128)       73856
_________________________________________________________________
max_pooling2d_409 (MaxPoolin (None, 11, 7, 128)        0
_________________________________________________________________
conv2d_549 (Conv2D)          (None, 7, 3, 256)         819456
_________________________________________________________________
max_pooling2d_410 (MaxPoolin (None, 3, 1, 256)         0
_________________________________________________________________
dropout_21 (Dropout)         (None, 3, 1, 256)         0
_________________________________________________________________
flatten_118 (Flatten)        (None, 768)               0
_________________________________________________________________
dense_116 (Dense)            (None, 19)                14611
=================================================================
Total params: 908,563
Trainable params: 908,563
Non-trainable params: 0
_________________________________________________________________
```

Figure 2. CNN Model

#### 4.1.1 Logistic Regression

For Logistic Regression the hyperparameters were selected by using GridSearchCV method, by taking 5 fold cross validation, and a range of parameter value for maximum iterations and C(inverse regularization parameter). Best classifier was obtained for Logistic regression:

```
LogisticRegression(C=100000000.0, class_weight=None, dual=False,
                   fit_intercept=False, intercept_scaling=1, l1_ratio=None,
                   max_iter=500, multi_class='ovr', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

#### 4.1.2 Linear SVM

The Linear SVM is same as polynomial kernel of degree=1. It is widely used in support vector machines for classification tasks.

#### 4.1.3 RBF SVM

The Radial basis function kernel is a widely used kernel function. It is commonly used in support vector machines. The result is reported for the same in the Result section. The major challenge we faced was how to select the features. We used the optimal parameters with the help of GridSearchCV and the hyperparameters came out to be: C=10 and for gamma we selected 'scale' parameter.

#### 4.1.4 Quadratic SVM

The Quadratic polynomial kernel is a kernel function of degree=2. By hit and trail method we selected gamma='scale'. The result is reported for the same in the Result section.

#### 4.1.5 Neural Network(MLP)

We used hit and trial to decide the number of hidden layers for the network. We found number of hidden layer equal to 7 as the best fit.

#### 4.1.6 Convolution Neural Network

We deployed our own CNN network in keras. The model summary for is show in the Figure 2. We started with using a single layer and then we using hit and trial, we noticed an increase in the accuracy but after 4th layer the accuracy started decreasing. Therefore, we decided to use only 4 layers and and then later on we tried different parameter to achieve best accuracy.

#### 4.1.7 PCA

Principal Component Analysis (PCA) is a method for feature extraction, i.e it combines our input feature variables in a specific way while dropping the "less" significant variables. The new variables formed are all independent of each other. This process is also called dimensionality reduction of the feature space. There can be two ways of reducing the dimensionality. One way is by mentioning the number of components/features which we want in the transformed dataset. The other way can be mentioning the total amount(%) of variation explained. The features which retain the amount mentioned by us are taken. For our model, we took variation explained = 90%

## 5. Results

The result for the various models are as follows:

### 5.1. Logistic Regression

Classification using the Logistic Regression report an accuracy of 74%. The ROC Curve for the same is shown in the Figure3.

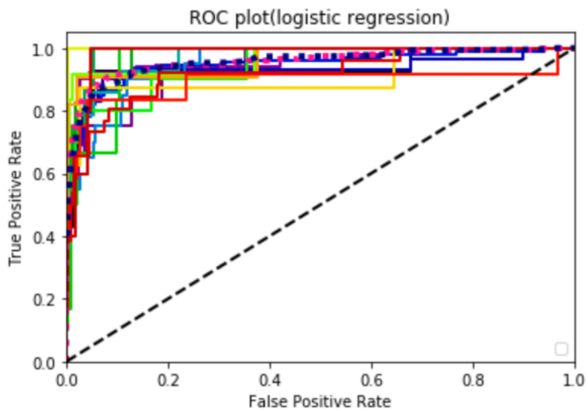


Figure 3. ROC Curve for Logistic Regression.

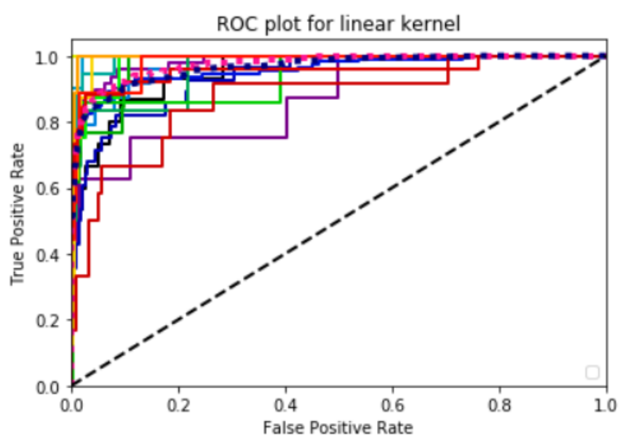


Figure 4. ROC Curve Linear SVM

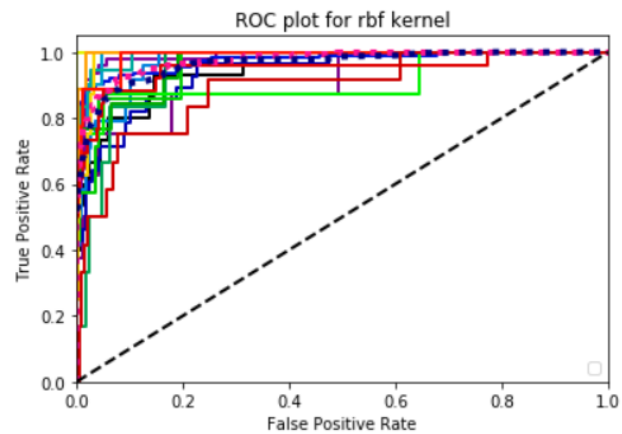


Figure 5. ROC Curve for SVM with RBF/Gaussian Kernel.

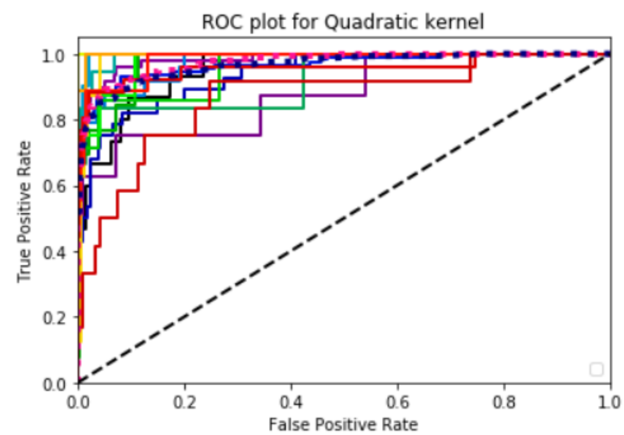


Figure 6. ROC Curve for SVM with Quadratic Kernel.

### 5.2. Logistic Regression (PCA)

We obtained 68.44% accuracy(test set) for Logistic Regression Classification. Due to PCA transformation, we observed the time taken for training was drastically reduced as compared to dataset without dimensionality reduction. The accuracy obtained by PCA may or may not be better than the accuracy without PCA.

### 5.3. SVM with Linear Kernel

SVM classification using the Linear Kernel reported an accuracy of 78%. The ROC Curve for Linear Kernel is shown in Figure4.

### 5.4. SVM with Gaussian/RBF Kernel

SVM classification using the Gaussian/RBF Kernel reported an accuracy of 76%. The ROC Curve for the same is shown in Figure5.

### 5.5. SVM with Quadratic Kernel

SVM classification using the Quadratic Kernel reported an accuracy of 78%. The ROC Curve for the same is shown in Figure6.

### 5.6. SVM with PCA

For RBF kernel, we observed 71.12% accuracy on test set. For linear kernel, we observed 70.32% accuracy on test set. For quadratic kernel, we observed 58.82% accuracy on test set.

### 5.7. Neural Network (Multi Layer Perceptron)

#### 5.7.1 MLP Using ReLu Activation Function

Neural Network with 7 hidden layers reported an accuracy of 71.92%.

### 5.8. Convolution Neural Network

CNN classification using the network defined in the Methodology section, reported an training accuracy of

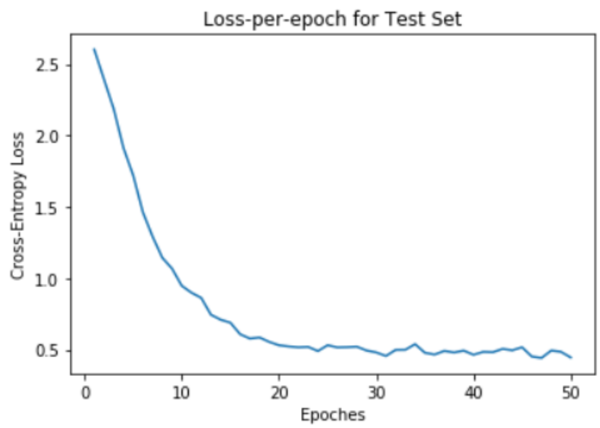


Figure 7. ROC Curve for SVM with Quadratic Kernel.

89.57%. The Epoch vs Error curve for the same is shown in Figure 7.

## 6. Conclusion

We started with default hyperparameters of Support vector classifier. But the overall accuracy came out to be very less. Then we used the optimal parameters with the help of GridSearchCV and the hyperparameters came out to be: C=10 and for gamma we selected 'scale' parameter. Without feature selection the data was overfitting as training accuracy came out to be 100%. Best accuracy for the SVM was found in the RBF kernel.

| Model | Testing Accuracy |
|---|---|
| Logistic Regression | 74% |
| Logistic Regression (with PCA) | 68.4% |
| Linear SVM | 78% |
| RBF SVM | 76% |
| Quadratic SVM | 78% |
| Linear SVM(with PCA) | 70.32% |
| RBF SVM(with PCA) | 71.12% |
| Quadratic SVM(with PCA) | 58.82% |
| Neural Network(MLP) | 71.92% |
| Convolution Neural Network(CNN) | 89.57% |

Due to PCA transformation, we observed the time taken for training was drastically reduced as compared to dataset without dimensionality reduction. The accuracy obtained by PCA was found better than the accuracy without PCA, and only reason that we could think is that PCA is not supposed to boost the accuracy of the model instead it act a catalyst for the training and testing. This could also been seen in the table provide in the results section.

For the neural network, the accuracy reported is less than that of SVM without PCA, though we cannot compare this results since the Neural network work as a black bock but a general fact about the Neural Network does apply here, i.e, whenever the dataset is small Neural Network could be out performed by the SVM since to attain good prediction Neural Network require large datasets.

The best results were found the in the CNN, and it does not surprise us at all. The previous too have found CNN's and deep network to work really well with such datasets.

Statistical analysis of the results clearly state the fact that non-linear model is better than linear model and this fact fact has been well established by various other research but only few have tested their model on this particular dataset. The possible reason for low accuracy in our models can be attributed to higher dimension of the feature space. This dataset is a state of the art dataset , and current state of the art Deep Learning models have proved some outstanding performance over this dataset.

Our approach is also different in the way we have used the datasets, since most of the published papers have used 'minimum faces attribute as 70 and reported their accuracy, therefore in order to complicate the model a bit we used minimum faces = 40

## References

[1] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. 2008.

[2] Md Omar Faruqe and Md Al Mehedi Hasan. Face recognition using pca and svm. In *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, pages 97–101. IEEE, 2009.

[3] Shaoyan Zhang and Hong Qiao. Face recognition with support vector machine. In *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003*, volume 2, pages 726–730. IEEE, 2003.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[5] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.