Home Assignment 2

Yash Arvindkumar Patel
School of Engineering and Applied Science
Ahmedabad University
Ahmedabad, India
AU1841141

Bhavesh Oza
School of Enginnering and Applied Science
Ahmedabad University
Ahmedabad, India
AU1949003

***Abstract*—In this Assignment we need to create a Convolution Neural Network for facial recognition with the help of feature descriptor HOG(Histogram of Oriented Gradient). To get model trained we have used 'fetch lfw people' dataset from sklearn library.**

***Index Terms*—HOG(Histogram of Oriented gradient), CNN(Convolution Neural Networks), Feature descriptor, Face, Recognition**

I. Introduction/ Motivation and Background

In this assignment, we have been asked to create a facial recognition. For that we need to train a model of Neural Networks with the help of fetch lfw people dataset from sklearn library. To get the input of this model we need to carry out the features from the images and to do that HOG (Histogram of Oriented Gradient) is used in this assignment.

II. Detailed Mathematical Analysis

A. Mathematics behind HOG

At first we need to understand what are the features.
Features are the combination of Gradient(Magnitude) and Orientation(Direction). To get these gradients and orientations we need to first reshape the image in 128x64 or somewhat combinations of that. After that we can follow below method:

- At first divide the image into 8x8 pixels (can be called as cells).
- Then get the gradients of each pixel by subtracting pixels in up-down and left-right manner:
  consider x (x-direction) and y(y-direction) as pixel, i as current location

$$G(x) = x_{i+1} - x_{i-1}$$
$$G(y) = y_{i+1} - y_{i-1}$$
$$Gradient = \sqrt{G(x)^2 + G(y)^2}$$

- Also we can get orientation from G(x) and G(y) values

$$\phi = atan(\frac{G(y)}{G(x)})$$

- After getting both the values we can now go for Histogram of values and to do that we need to make bins of 20-20 slots, (ex.: 0, 20, 40, ..).

$$value = \frac{\phi}{bin} * Gradient$$

  After getting value we can decide that in which bin does the value can be put in and after getting that we can add that value to that particular bin bucket of Histogram.
- Now here we are considering bins with 20-20 gap so there will be 9 bins (called orientations) and hence the output of HOG, vector will be of 1x9.
- After that we need to normalize gradients to decrease the effect of brightness because we cannot completely neglect that. This normalization can be done with two cells at a time (called blocks). Therefore now 16x16 pixels at a time.
- And the above steps are carried out with each and every pixel and then combine them all in one.

B. Mathematics behind Training models

Basically in this section we can understand what is data we need to train and how we are going to train and also which models we are going to use.

- When we use the hog image library we got two returned values, first is the feature 1-D matrix and second is the hog image.
- We are going to use the feature values to train and test model.
- Now if we go with $64 * 32$ image then the feature matrix will be of $1 * 756$ i.e. there will be 765 feature columns for about 9000 data rows.
- Other possibility is we can go with $128 * 64$ image then feature matrix will be of $1 * 3780$.

III. Experiments and Results

There has been experiments with the HOG descriptor as well as with the classifiers.

- Properties of dataset that has been used:
  In this assignment lfw people dataset is used from sklearn library, that contains about 13233 images of $62 * 47$ dimension/resolution in grayscale mode.
- There has been two classifiers are used, one is SVM classifier, second is KNN classifier and almost all the classifiers results into 20-30 percent of accuracy.

A. Results of HOG classifier

Here when we converted the image to 128x64 many features got changes and a obvious observation here is the

Fig. 1. Grayscale image

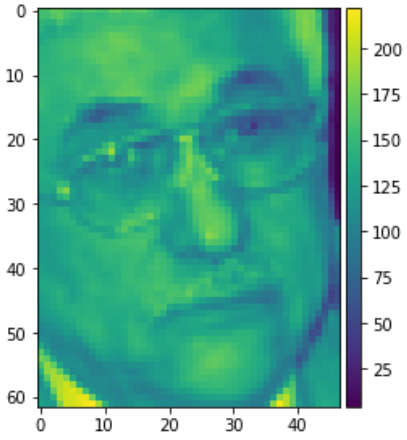


Fig. 2. HOG of grayscale image

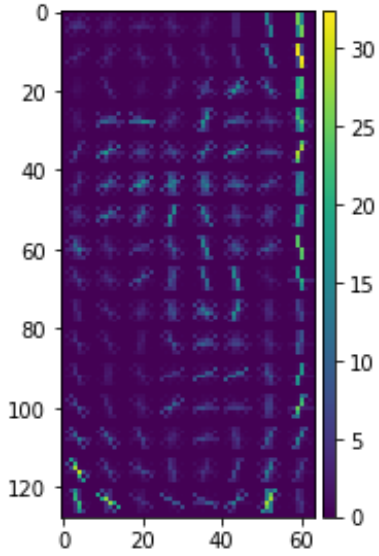


image gets stretched out vertically which may not be in favour of good features.
But when we converted image into lower dimensions then the features got affected badly and there can be loss of around $3780 - 756 = 3004$ per image.

B. Experiments on classifier

At first we use SVM classifier with both the dimensions $64 * 32$ and $128 * 64$. For first model we got some 19.5 percent accuracy.
In second model we got some 25 percent accuracy.
For the third model we used KNN model and we got some 15 percent accuracy.

classifier	dimension	training time	accuracy
SVM classifier	64*32	35 min	0.195
SVM classifier	128*64	45 min	0.25
KNN classifier	128*64	20 min	0.155

C. Performance Metrics

Since there has been around 2500 faces, there will be 2500 classes and to get ROC plot of False Positive Rate vs True Positive Rate we need to find all the values of Confusion metrics (i.e. True Positive, True Negative, False Positive, False Negative).

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

So we can do one thing here, we can take 1 for first class and rest classes as 0 and get the values. But after applying this idea on the class with highest frequency we got FP value as zero.

Fig. 3. Results of confusion matrix

```
TP_1871- 140  |  TP_1047- 70
FN_1871- 20   |  FN_1047- 12
FP_1871- 0    |  FP_1047- 0
TN_1871- 877  |  TN_1047- 947
```

Here we can see that False Positive values for both the classes are zero and thus False Positive Rate will be zero and we can't plot the ROC-AUC.

IV. Discussions and Conclusion

- From the above results we can conclude that for the classifier problem of multi-classes performance metrics should consider confusion matrix values for each needed class rather than plotting ROC plot for every class.
- Also from the results for accuracy of the classifiers for different dimensions draws a discussion of more the number of features more is the accuracy. But here at certain point we were afraid of overfitting the data as number of features are roughly $1/4_{th}$ of dataset.
- Here we can also get hints of background light intensity or cluttering background, HOG will carry out features of all the objects in background and hense the feature matrix seemed out confusing for model.
- Light intensity carried out some difficulties on training as there will be difficulty in getting the magnitude of change for feature matrix. From the below images we can see that in second image with more illumination we are not able to see the parts of faces properly and thus model will be able to get the boundaries of face but is not able to get the boundaries of nose.

Fig. 4. Image with less brightness

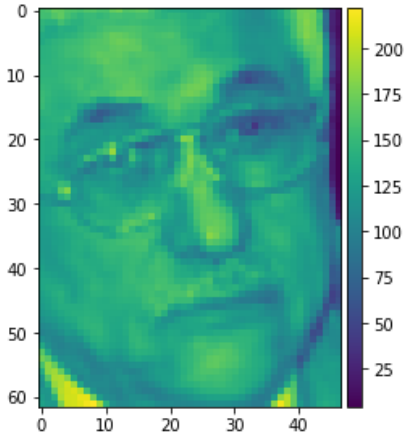
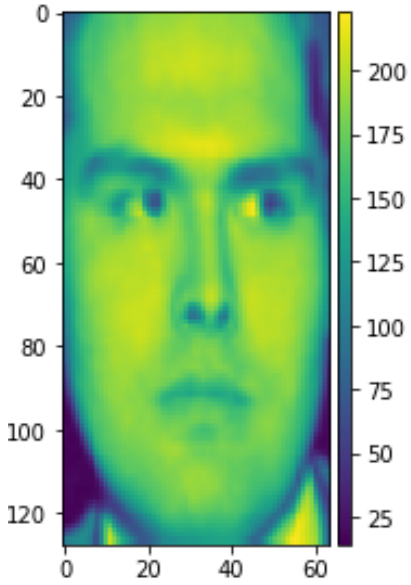


Fig. 5. Image with more brightness



REFERENCES

[1] https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/.
[2] https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/
[3] https://scikit-learn.org/stable/modules/svm.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html