Image losing clarity

Image collected from Pinterest

How to store the data

Irregulates in algorithm

Post melone tatoo not working]

# The solution

## Data collection:

One method would have been to ask 10 people to write the English alphabet and numbers. Which we then had to take a photo and find a way to upload it and organise it and use those images for our model

But we went the easy way and took the data from Kaggle which had its data well organised and an easy way to access it

When we played with the no of components, we found that when we increase beyond 30-40 the accuracy became low and when we decreased it from 30 we found it to be less accurate as well

When the k value was increased the accuracy got low k value of 1 gave the maximum accuracy

# How we got to the solution from the failures and how we tuned it

Failure to center the image(as we thought it would increase the accuracy)

How we thought of doing it

# Presentation points

Recognising hand written English alphabets and number is the other project

Applications of this are numerous like reading postal addresses, or any forms which of this type where you need to fill up the boxes individually with characters one by one.

The end goal of this project is to develop a fairly accurate model which Recognisees hand written English alphabets both capital and small letter and the numbers which can be applied in many scenarios like the ones mentioned above.

The method that we used has five steps.

Data Collection

Data analysis

Feature extraction from the images and reducing it using Eigenfaces that is by applying PCA

Modelling of different classifiers which will classify a given image into whether it a Small letter w or capital letter W or whether it is a number, there are 62 such classes 26 small letter and 26 capital letters and 10 numbers

Performance analysis under different scenarios like different k values in the knn classifer or different no of components in pca.

Data collection

One way to collected data would have been to ask 10 people to write the characters and the take a photo of it and organise for easy accessing, but we found a better data set from Kaggle which had 55 different handwritings of each character so we had around 3410 sample images for pre-processing and training.

The dataset had an excel file with all the file names of the sample images along with there labels which one of the main reasons we went with this dataset.

So we read this excel sheet and store all the file name and labels in a numpy array.

Data analysis

we are taking each pixel in the image to be a feature. in order to have same number of features for each image we resize the grey scale image as we are working only with grey scale values of the image to a size of 64\*64 using OpenCV. Next we transform this 64\*64 image to an ndarray of size (64\*64) in order the apply pca.

So now the images variable is (3410 \* 4096) 2d array where 3410 is number of sample images and 4096 is the number of pixels in the grey scale 64\*64 image.

Visualization

The show function creates n images with their corresponding titles

Here we have displayed 12 random images from the dataset

The data is spitted into 75 Train and 25 test data using the train\_test\_split from sklearn library

Now we apply pca from the sklearn library to the training data and reduce the dimension to 30 that is we get 30 eigenfaces and transform the train and test data into a weight vector of size 30. Each weight in the vector is the corresponding weight of the eigenface in representing the images with eigen faces.

When we apply Knn classifier from the sklearn library with different k values and find that the accuracy for k value equal to 1 is the maximum

Now we visualize how the images looks like with 30 dimensions representation and how the knn predicts the labels.

For this we first have to convert the images with weight vector representation to the actual dimension that is 4096 we do this by matrix multiplying eigenfaces with the weight vectors.

Here is a set of random images which are represented using 30 eigenfaces with the predicted label and actual label. We can see that around half of the predicted labels are corrects

Now we try to classify the test data using different classifiers such svm and logistic regression

SVM has a accuracy of about 0.5 where as logistic regression has an accuracy of 0.32