Optical Character Recognition

CS 559: FUNDAMENTAL OF MACHINE LEARNING AND APPLICATION

Taru Tak
Amit Vadnere

Under guidance of Professor X. Wang

TABLE OF CONTENT

1.INTRODUCTION:

Optical character recognition is a progressive area of research in variety of computer science domains like computer vision, Machine Learning, Pattern Recognition and Artificial Intelligence.The basic idea is to recognize characters from scanned or digitalized media formats like documents and photos.

OCR system help in electronic conversion of images of typed, handwritten or printed text into machine-encoded text coming from stand document or photo of a document or subtitle text on an image.It allows us to convert a scanned paper document, PDF files or images captured by digital camera into editable and searchable data.The benefits of OCR are in increasing the efficiency and effectiveness of digitalization of printed data.It helps in avoiding retyping, quick digital searches, saving space and building digital knowledge base.

OCR can used in application like data entry automation, indexing document for the search engine, automated number plate recognition, blind and visually impaired assistance.The conversion of the printed text to digital form provides an advantage of storing data compactly, displaying online, and used in machine processes such as cognitive computing, machine translation, (extracted) text-to-speech, key data and text mining. The applicability can also be extended for routine activities like scanning paper records for further altering and sharing, extricating cites from books and magazines and utilize them for making course studies and papers without the need of retyping etc.

Each of these applications requires processing of datasets that contain hundred of thousands of documents which requires powerful algorithms that are highly scalable and accurate.

We propose a solution for OCR using state of the art Convolution Neural Network model which is extremely efficient, large scale model with high accuracy.

2. PROJECT SCOPE AND METHOD OVERVIEW

With a focus to move from manuscripts to digital world by digitizing the content of the images to digital characters so as to convert the data into computer readable form. In this project we are focusing on learning a classifier which is accurately able to classify handwritten or printed english alphabets. Our project compare the performance of commonly used machine learning algorithms like K-Nearest Neighbours, Artificial Neural Networks, Support Vector machines, Random forest classifiers , logistic regression classifier and Naive bayes Classifier with Convolution Neural Network on the Stanford handwritten word dataset collected by Rob Kassel at MIT spoken language system group.

3. DATASET AND TOOLS

Stanford dataset: http://ai.stanford.edu/~btaskar/ocr/

This dataset contains handwritten words dataset collected by Rob Kassel at MIT Spoken Language Systems Group. The selected set is a "clean" subset of the words and rasterized and normalized images of each letter.The tab delimited data file (letter.data.gz) contains a line for each letter, with its label, pixel values, and several additional.The data consist of images of letter with 128 dimensional vector which we reshaped into 16 X 8 pixel. The datasets has 52152 samples
The data set contains the following fields -
1. id: A unique integer id for each letter
2. letter: A-Z
3. next_id: Id for next letter in the word, -1 if the last letter
4. word_id: Each word is assigned a unique integer id
5. position: Position of the letter in the word
6. fold: 0-9 - Cross-validation fold
7. p_i_j: 0/1 - Value of pixel in row i, column j.

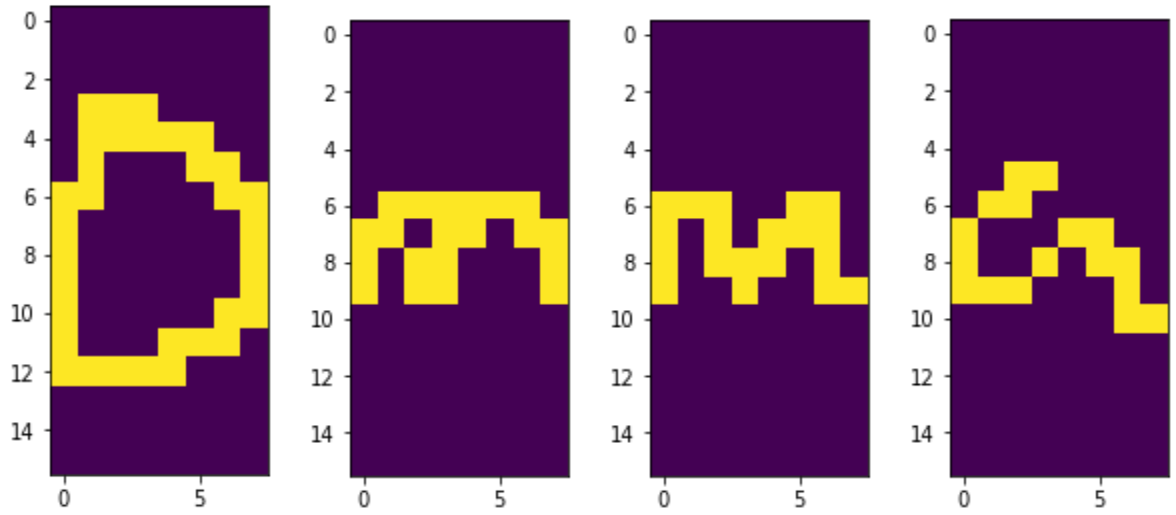
	0	1	2	3	4	5	...	123	124	125	126	127	label
0	-0.195347	-0.257120	-0.350951	-0.419458	-0.436893	-0.388364	...	-0.420185	-0.441738	-0.372311	-0.282047	-0.201558	o
1	-0.195347	-0.257120	-0.350951	-0.419458	-0.436893	-0.388364	...	-0.420185	-0.441738	-0.372311	-0.282047	-0.201558	m
2	-0.195347	-0.257120	-0.350951	-0.419458	-0.436893	-0.388364	...	-0.420185	-0.441738	-0.372311	-0.282047	-0.201558	m
...	...	...	...	...	...	...	...	...	...	...	...	...	...
52149	-0.195347	-0.257120	2.849399	2.384027	-0.436893	-0.388364	...	-0.420185	-0.441738	2.685928	-0.282047	-0.201558	i
52150	-0.195347	-0.257120	-0.350951	-0.419458	-0.436893	-0.388364	...	-0.420185	-0.441738	-0.372311	-0.282047	-0.201558	a
52151	-0.195347	3.889236	2.849399	-0.419458	-0.436893	-0.388364	...	-0.420185	-0.441738	2.685928	3.545511	-0.201558	l

52152 rows × 129 columns

Fig: Constructed Frame with 52,152 samples and 128 + 1 columns.

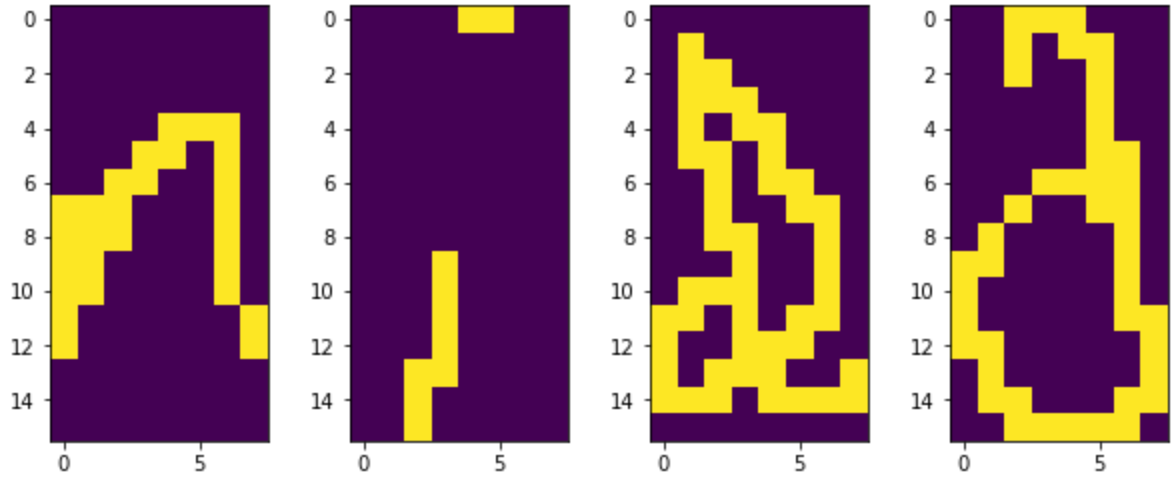
Fig: Number of Samples per alphabet in the dataset.

Data preview: Reconstruction of 128 pixel values to Images for viewing.



Ground Truth

O M M A



N I D A

Libraries:

1. **Numpy:** Numpy is a python library will be majorly used for performing the complex mathematical function on multidimensional arrays.
2. **Pandas:** Pandas is a python library that will be used basically for data manipulation and analysis for the prediction.
3. **Sklearn:** Keras is an open source library, is written in python, is basically a scientific tool for machine learning.
4. **Matplotlib:** Matplotlib is a library written in python, used for visualization of data.
5. **Timeit:** Timeit is a open source library used to calculate running time of algorithms.
6. **Keras:** Keras is an open source library, is written in python, will be used for the CNN algorithm implementation.
7. **TensorFlow:** TensorFlow is an open-source software library for dataflow programming across a range of tasks.
8. **Seaborn**

Tools:

1. Anaconda Jupyter Notebook
2. Python

4. METHODOLOGY

Overview of the steps followed during the project.

Step 1: Exploratory Data Analysis

Step 2: Pre-Processing

Step 3: LDA vs PCA

Step 4: Divide the data into train and test with train set equal to 80% and test set 20%

Step 5: Evaluate models without LDA

Step 6: Evaluate models with LDA

Step7: Implemented CNN with 1-convolutional layer and 1-fully connected dense layer with 26 output nodes.

Step 8: Conclusion

5. CONVOLUTION NEURAL NETWORK

Convolution neural network is class of deep neural network is used for computer vision and image processing.

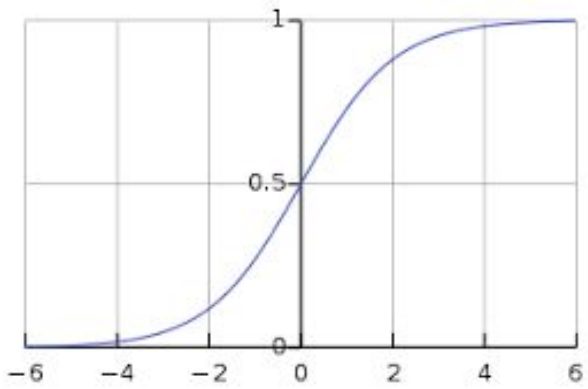
Convolutional Layers

CNN takes images as pixels and express them in a matrix of h X w X nc where height of the image is h, w is the width of the image and nc is the number of channels for example if the image is rgb image then the number channels equal to 3 and for a grayscale image the number of channels equal to 1. The fundamental part of convolution neural network is learnable filter which is used to identify the a particular feature in the original image.The filter is slided across the height and width of the input file and a dot product is calculated to produce an activation map.The different feature extracted by different filter when convolved on the input file generate an activation function maps which is passed to next layer in CNN.

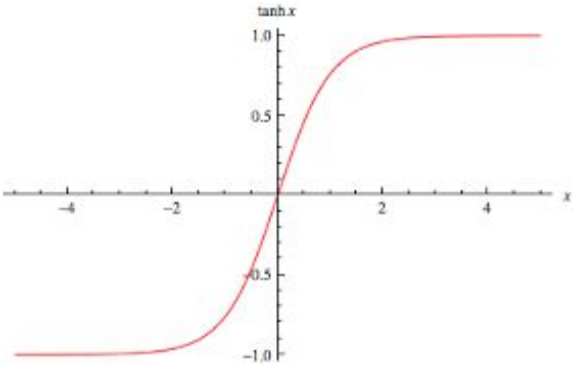
Activation Function

"The activation function is the non linear transformation that we do over the input signal.This transformed output is then sent to the next layer of neuron as input"- Analytics Vidhya

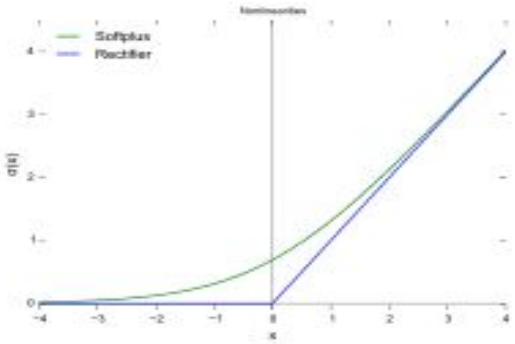
Activation function is also referred as threshold function which act as node either placed at the end or in between the Neural Networks. There are different type of Activation function however the most commonly used functions are listed below:



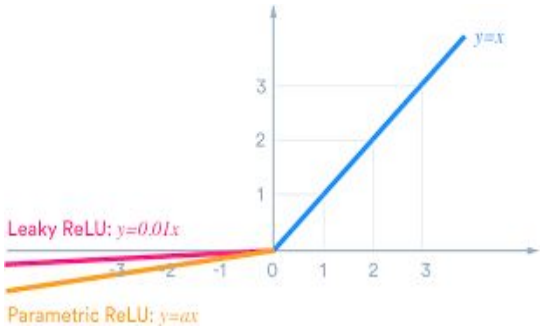
Sigmoid



Tanh



ReLU



Leaky ReLU

Fig: Activation Functions

CONVOLUTION LAYER OUTPUT FORMULA

$$n_h = \left[\frac{n_h prev - f + 2 \times pad}{Stride} \right] + 1$$

$$n_w = \left[\frac{n_w prev - f + 2 \times pad}{Stride} \right] + 1$$

$$n_c = No.\ of\ filters\ in\ Convolution\ layer$$

$$n_h = height\ of\ \ input\ image$$

$$n_h prev = height\ of\ \ image\ from\ previous\ layer$$

$$n_w = width\ of\ \ input\ image$$

$$f = Size\ of\ \ convolution\ filter$$

$$pad = Size\ of\ \ padding\ layer$$

Pooling Layer

CNN also involves a pooling layer which is present between the convolution layer which is used to reduce the amount of parameters and computation in the network and also control the overfitting by progressively reducing the spatial size of the network. Pooling layer involves two operations Max Pooling and Average Pooling.

The basic idea behind the max pooling is taking the maximum parameter while ignoring the rest in process of filtering the input at every strides and hence providing the maximum value from a pool.

Formula for the output after Max Pooling:

$$\frac{(N - F)}{S + 1}$$

$$N = dimension\ of\ \ input\ to\ pooling\ layer$$

$$F = Dimension\ of\ \ filter$$

$$S = Stride$$

Fully-Connected Layer

Fully connected layer is considered the last phase of the neural network and hence the activation for this layer can be calculated with a matrix multiplication followed by a bias offset.

Output Layer

Output layer corresponds to the number of outputs we have for or problem.

OCR CNN Model Implemented :

This is the model we designed using Keras.

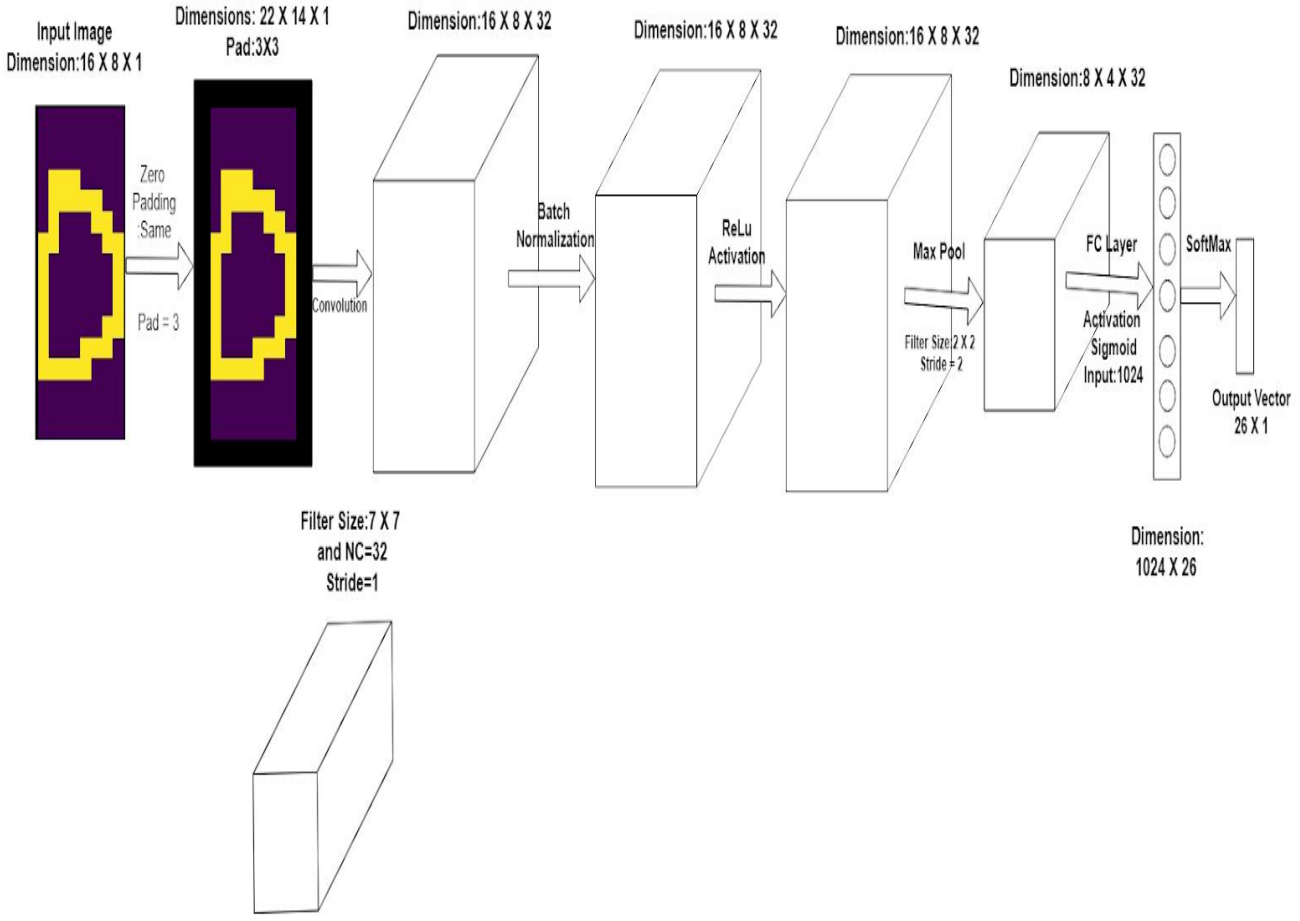


Fig: CNN model with 1-convolution layer and 1-fully connected layer

Implementing the Convolution Neural Network

- Convert labels into 26 dimensional one hot encoded matrix.
- Reshape the train data into M X 16 X 8 X 1 dimensional tensor
- Split. the train test data.
- Use Keras to design sequential Keras Model
    - **Step 1:** Pad with 3 X 3 filter
    - **Step 2:** Run 32, 2-D Convolution filter of size 7 X 7 X 1 with stride equal to 1.
    - **Step 3:** Batch normalization
    - **Step 4:** Apply ReLu activation function
    - **Step 5:** Apply Max Pooling with filter size 2 X 2 and stride equal to 2.
    - **Step 6:** Flatten the matrix
    - **Step 7:** Make a fully connected dense neural network with activation function equal to sigmoid consisting of 26 output nodes.
    - **Step 8:** Initiate the design model with binary cross entropy loss function and adam optimizer.
    - **Step 9:** Fit the model on train data with batch size equal to 50 and number of epoch 40.
- Evaluate the model with our test set and record the accuracy.

6. K NEAREST NEIGHBOUR

K nearest neighbour is a nonparametric approach that is based on feature similarity techniques that consider the out of samples and training sets for comparision to make the prediction by examining the majority vote from K nearest neighbour selected on the basis of similarity feature.

KNN Algorithm

- A positive integer k is specified along with new sample
- We find the K - nearest samples to the new sample which by considering the distance.
- We do majority polls among the K nearest samples to select the class of new sample
- This gives us the class of the new sample

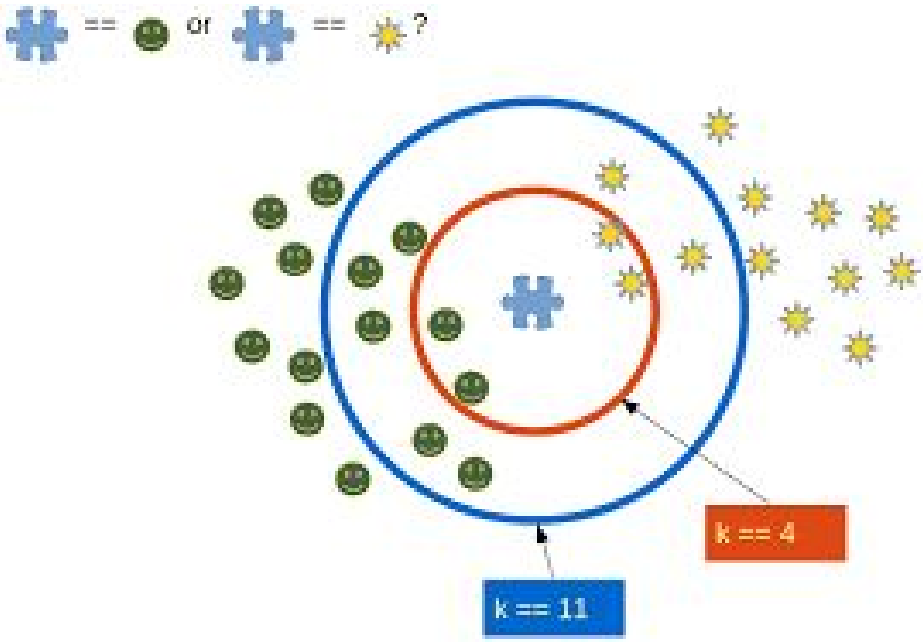


Fig:KNN

7. PCA

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. If there are n observations with p variables, then the number of distinct principal components is min(n-1,p) This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

We can observe that PCA is not doing a good job in preserving variance. To preserve 90% variance we will have to keep athletes 100 columns, so we will not be utilizing PCA.

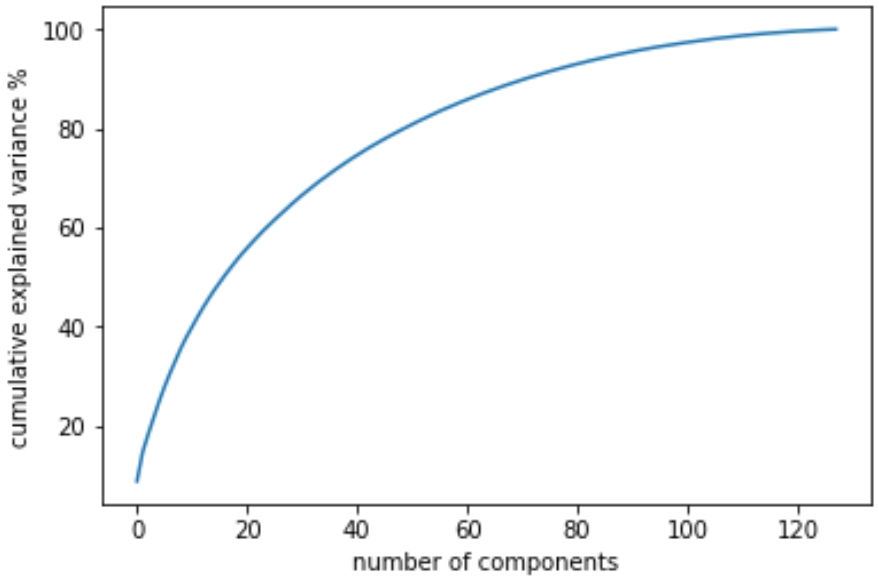


Fig: PCA on OCR dataset

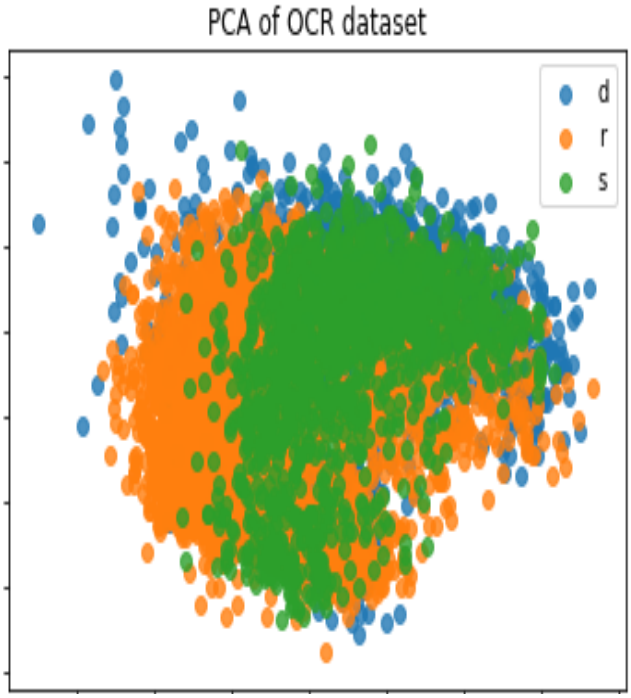
8. PCA and LDA

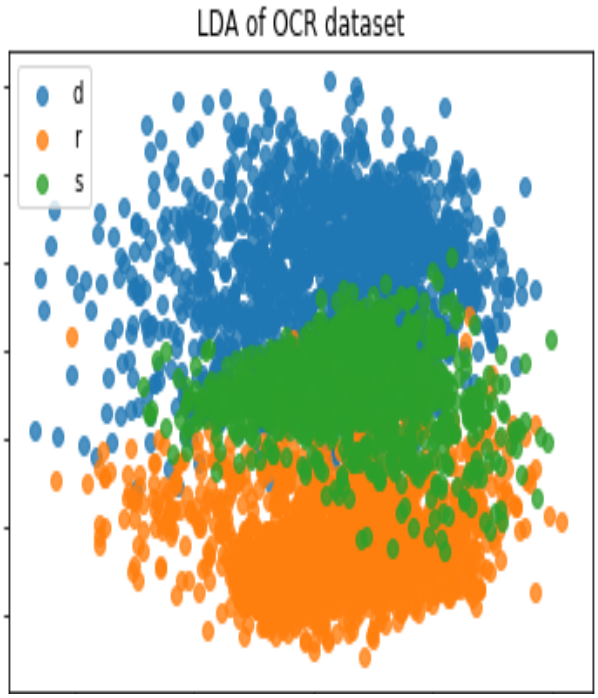
LDA

Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification.

PCA VS LDA ON OCR DATASET

We are using are 3 commonly occurring characters D,R,S to visualize the transformations of LDA and PCA on our dataset. The plot clearly demonstrate LDA is capable of reducing number of features while providing maximum separability.





9. RESULT AND ANALYSIS

Shown are comparison of performances of various models.

Classifier Name	Runtime (in sec) Without LDA Features = 128	Runtime (in sec) With LDA Features=25	
KNN	260.6	68.097	***
Linear SVM	314.4	101.63	***
Random Forest	4.2	7.59	*
AdaBoost	111.18	260.42	***
Neural Nets	70.39	72.53	~
Naive Bayes	3.52	1.21	*

Run Time Comparison Table

Classifier Name	Accuracy(%) Without LDA Features = 128	Accuracy(%) With LDA Features=25	
KNN	81.93	85.18	+
Linear SVM	83.15	80.17	-
Random Forest	86.04	83.75	-
AdaBoost	27.22	29.69	+
Neural Nets	86.51	82.28	-
Naive Bayes	35.40	71.54	+

Accuracy Comparison Table

CNN

The CNN reaches 77.63 % accuracy in its first epoch which is followed by 39 more epochs to reach **train accuracy of 97.04%.**
Training time for 40 Epochs was 1788 seconds.

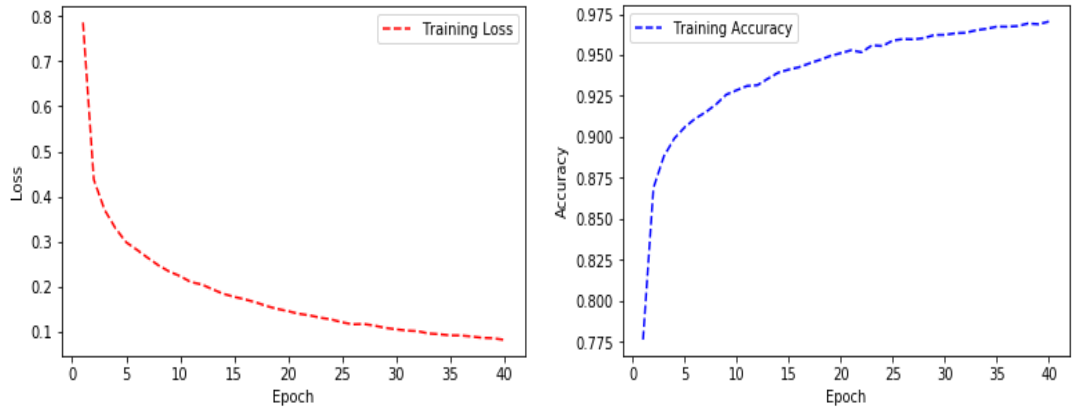


Fig: Training loss and Accuracy by Epochs

Final Test accuracy is 89.72% that lead us to consider CNN as the best algorithm among all other considered algorithm.

10. CONCLUSION

The CNN model used in implementation of OCR can be efficiently used to speed up large amount of image based documents into structured documents that are easy to discover, search, edit and process.OCR provides fast and automated data capture which can save considerable time and labour cost . CNN implementation on OCR easily surpasses human level accuracy in performing character recognition.

11. FUTURE WORK

We will try to extend this project by combining it with object detection model so that we are able to localize text in a given image and perform optical character recognition such an application can be used in entry automation, indexing document for the search engine, automated number plate recognition, blind and visually impaired assistance.

12. REFERENCES

- https://medium.com/@udemeudofia01/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17
- https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7
- https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5
- Dataset: http://ai.stanford.edu/~btaskar/ocr/
- https://en.wikipedia.org/wiki/Principal_component_analysis
- https://en.wikipedia.org/wiki/Linear_discriminant_analysis
- https://keras.io/
- https://scikit-learn.org/stable/index.html
- https://www.deeplearning.ai/