Handwriting Recognition

Pallab Chakarborty **Sudeep Vig** **Imankalyan Sarkar**
{pallab20063,sudeep20097,imankalyan20010}@iiitd.ac.in

Abstract

Handwriting recognition is a technique to convert the handwritten documents data from different input files to a Format that computer can understand. Handwriting recognition solves the problem of keeping records at multiple files at various locations or warehouses, which may get damaged due to various reasons when captured on paper and takes a huge amount of time to access past records. This system can be applied to different languages, including English and many other languages, for which we will be going to train our model. This system is very beneficial for the banking system for verifying signatures. For developing this system, we are going to use various ML techniques with some neural networks.

1 Project Introduction

Handwriting recognition is a technique used by the computer to convert the handwritten documents data into the format that the computer understands. The aim of our project is to recognize the text from all the images given in the data set, having entries about 3,00,000. We will make a model that will take these many images of the data set and give us the required labels or textual data from the images.

2 Importance of project

In the past, we have to keep the records that we have to store at multiple files at various locations or warehouses, which may get damaged due to various reasons when captured on paper and takes a huge amount of time to access past records. This system solves the problem by storing all the records on the computer memory so that we can access the past records in an instant. This also solves the issues of data protection, which may get damaged in the paper format. This system also doesn't require any labour to update any entry in the record.

3 Literature Survey

The first research on character recognition is done by Grimsdale in 1959. In the early sixties, they found some results after so many attempts. K. Gaurav, after used various preprocessing techniques like skew detection and correction,noise removal, etc on different kinds of images. but after applying all the techniques, he is not able to get accurate results.
Recent research was done by the M. Hanmandlu, O.V. Ramana Murthy presented their paper in both Hindi and English language.He gave a fuzzy model that we derived from the features using the box approach. His model has given 95% accuracy for Hindi Numerals and 98.5 % for English Numerals. http://ijcsit.com/docs/Volume
Problems faced in handwriting recognition today are:
Poor handwriting and sometimes cursive handwriting is also even more difficult to interpret by the models.

4 Dataset

The dataset used is open source and available to the public. The source of the dataset is given in the references. The data set provides Transcriptions of about 400,000 handwritten names for Optical Character Recognition (OCR) which was obtained through different charity projects to support disadvantaged children all around the world. The data is quite large in size as it contains images. We use only a certain portion of the data set due to limited computing power at our end. The data set is split into columns containing the links to images, and the transcription on them. There are unique ID's in the data set which contain first names and last names. Overall in the data set there are 206,799 first names and 207,024 surnames in total which have been preset into train, test and validation sets. So there are each of the first names and last names are present in the data set and they are uniquely identified with the help of a tag. Despite the clean nature of the datset it is dirty in the sense that not all images are clean are there are certain computer generated text in most of the images which were spotted during initial viewing of the dataset.

5 Data Analysis

We begin by obtaining a clean dataframe containing the images of the first and last names of the individuals who are identified with the help of a tag. To perform this operation we need to fetch the actual images which can be done in the form of url or downloads. The download size was too large to be put up on the cloud for furthur processing thus we chose to donwload the data and save only a portion of the total images to work with further. More data is expected to be added further on in the project. Some of the images are then visualized with the help of python libraries. Analysis reveals that the data is contaminated with computer generated text in almost all of the images. Some of the words appear in all of the images which is not at all useful to us for our machine learning model or preprocessing. This part of the image will need to be cropped off somehow so as to obtain clean data. On researching about the same we found techniques to clean the images so that only the desired word remained in the image. Cropping can be used but without human intervention it won't perform a very clean job. We employ a small cropping algorithm taking an estimate of whee the characters of the name are contained in the images. An advancead technique can be employed here which uses the opencv library and uses template matching to delete that part of the image. We try to combine the two methods in order to obtain a reasonly good image.

6 Methodology

Our first objective is to draw the bounding boxes around each character in the name so that we can later crop it off to make an isolated character recognition. So we need to crop off single characters and give them a label. So we know that a processed picture is a binarized one and will contain only black and white pixels. so let us consider an image as an grid of 0's and 1's. So we can find the connected components in the image by finding the custers of either 0's and 1's. So each major cluster will give us the each character and we can return the end points of each clusters.
Some characters in the English alphabet system do not solely rely on a single large connected component. Like 'i' has a dot over a straight line. So if we find a small cluster over a large cluster then we can consider it as a part of the large cluster.
After we are done with the detection of clusters we need to draw the bounding boxes. For drawing the bounding boxes we will fit a largest square/rectangle that completely encloses the cluster and return the points which are at its corner.
After the bounding boxes are drawn we need to crop each bounding box into separate parts. Now we know the characters in the string from the labels present in the training data. So we will map each cropped cluster to its label . So this will provide us a good resource for the MLP to predict these single character later on when we test the model.

7 Preprocessing

For preprocessing the data we employ the above methods discussed above. In detail the steps that we perform are the following :-

- Remove the portion of the images which are common to all the images i.e the computer generated text having words like "NOM" and "PRENOM".

- Crop the image to only have the portion of the image which contains the names written by a human and nothing else.

- Noise in the form of random half text in some of the images in the sides, bottom or top.

The better we preprocess and work in the data the better results could be expected from out model. This is highly laborious task to figure out which of the images and their labels could be finally usable for the machine learning model.
One of the ways this has been tackled in the later section is during our character recognition phase. If the obtained characters aren't equal to the number of characters in the labels we can reject the data as it wasn't properly processed or it might have been too noisy to take it into consideration for the model.

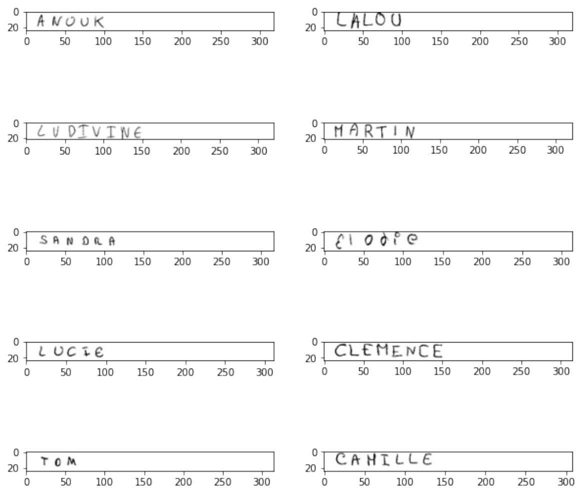
Figure 1: Sample images from the dataset

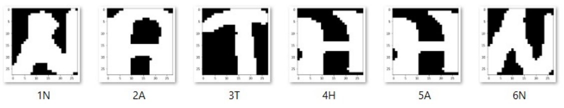

Figure 2: Cropped Images

8 Visualization

We have 4991 sample images . Each of varying size. Each image is categorized into two parts. First name and last Name. So if a persons name is ”Ram Kumar” then there will be two images for him 1: Ram 2: kumar. The images consists of handwriting written in black over a white sheet of paper.
A few sample images are shown .
One thing to notice is that we are inverting the image before doing any such processing of bounding boxes.

Now our goal is to draw the bounding boxes around each character , rather the large clusters and then crop them off. A sample example of the cropped character is given .

9 Character Separation

The process of seperation of character falls into three broad steps.

- **Identifying the clusters**:
  First we need to find the points which are connected in the image. That would break the image into clusters. By clusters we mean the large groups of connected points. An example of cluster is given in Figure 3.
- **Drawing the Bounding Box**:

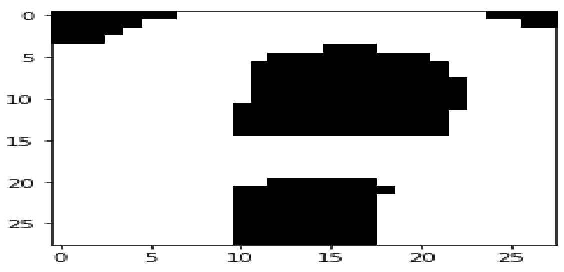
Figure 3: Cropped Images

  After we have figured out the clusters we are going to find the largest square that would fit the cluster. We call that as bounding box. we return the end co ordniate of the bounding box.
- **cropping the image**:
  after we got the coordinate of the bounding box we need to crop out the part present inside the box.
- **resizing each cluster**:
  We need to resize the clusters into some standard size (28x28) so that every image is of same size.
- **Attaching label**:
  We got the label in out training set. So we need to give label to each cluster according to each character [except space] in the string.

10 Feature Extraction

We have used two different strategies to extract feature from the images.

- **Principal Component Analysis(PCA):**
  Principal Component Analysis is the process of reducing the dimension of a dataset and maintaining information from the images. With the help of PCA, we are reducing the chances of overfitting. PCA will also going to improve the visualization of data and improves our model performance.
- **Histogram of Oriented Gradients(HOG):**
  Histogram of oriented gradients is the combination of gradient magnitude and direction.HOG measures the contrast in the different image regions as we move from one area to another. There is a change in pixel

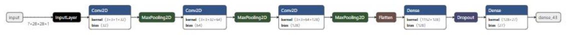

Figure 4: CNN model Architecture

intensity. This transition from one region's pixel intensity to another region is known as the gradient magnitude. Whenever we move from the black to the white region, we will get the maximum difference in the pixel intensity values.HOG also measures the direction, i.e., in which direction these gradient magnitude is changing, giving us the shape of an image because this gradient magnitude will give maximum values at the edges and corners.

11 Architecture:

We have used different models for the prediction of handwritten characters:-

- **CNN:**
  we have used a CNN architecture that we trained from scratch using 20000 images belonging to 26 classes of characters. The architecture of CNN is given in Figure 4. We have used 3 Layers of convolution and maxpool layer each having a kernel size of (3,3) [for convolution] and we have used padding because the image was a bit small in size and we didn't want to loose valuable information in the images. The CNN were self sufficient for feature extraction and we didn't had to do any external feature extraction. After the convolution and max pool layer we used lwo fullly connected layers with first one activated by relu and second by softmax as it is a case of multiclass classification.

- **SVM:**
  We also use Support Vector Machine in order to make predictions on the data. The image first needs to be flattened in order to process it in SVM. The 28x28 images is flattened to an array of 784 pixels. Therefore, we have a 784 features to handle. So we apply feature extraction methods to reduce the number of features. Two of the methods used here for getting the features are Histogram of Gradients(HOG) and PCA(Principal Component Analysis).In SVM we used a radial basis function (rbf) kernel with c = 1.0 and gamma to scale. we have used One vs One for multiclass classification.

- **MLP:**

  We have also tried multilayer perceptron or MLP classifier with hidden layer of [300,400,150] and we have used hog and pca for feature extraction . Hog gave us a better accuracy both classwise and overall in case of MLP classifier. we used the implementation of MLP in sklearn.

12 Different model accuracy:

CNN	MLP +HOG	MLP +PCA	SVM +HOG	SVM +PCA
0.9325	0.88175	0.86275	0.89725	0.898

Character wise accuracy

	CNN	MLP +HOG	MLP +PCA	SVM +HOG	SVM +PCA
A	0.9628	0.9341	0.9138	0.9594	0.9543
B	0.8684	0.6842	0.5263	0.6578	0.5526
C	0.9531	0.875	0.875	0.9296	0.9218
D	0.7571	0.7285	0.6428	0.8	0.7428
E	0.9518	0.9129	0.9074	0.9351	0.9370
F	0.714	0.3333	0.5714	0.0952	0.3333
G	0.9047	0.8571	0.7619	0.7857	0.8095
H	0.7758	0.7327	0.6637	0.7413	0.7413
I	0.9387	0.8830	0.8662	0.9359	0.9108
J	0.7608	0.7826	0.6956	0.6956	0.6521
K	0.8888	0.7777	0.8333	0.8333	0.8333
L	0.9697	0.9368	0.9505	0.9423	0.9587
M	0.9492	0.8223	0.8172	0.8223	0.8680
N	0.9509	0.8882	0.8474	0.9182	0.9100
O	0.9553	0.9241	0.9330	0.9419	0.9598
P	0.8888	0.7777	0.7777	0.7777	0.7777
Q	0.6	0.4	0.4	0.2	0.2
R	0.9129	0.8878	0.8146	0.8634	0.8487
S	0.9687	0.925	0.9187	0.95	0.9562
T	0.9222	0.9111	0.9055	0.9222	0.9166
U	0.9323	0.8421	0.9022	0.8872	0.9097
V	0.875	0.75	0.6562	0.6875	0.7187
W	0.3333	0.2222	0.4444	0.1111	0.4444
X	0.8571	0.8	0.6285	0.8	0.7142
Y	0.9344	0.9344	0.8032	0.8852	0.9180
Z	1.0	0.8	0.8666	0.8	0.8

Word accuracy using CNN : 0.7586
Word accuracy using SVM : 0.6074

13 Inferences

The different accuracy obtained from each of the classifiers have been shown in section 12. From the results that we found we can see that the classical machine learning models and the deep learning models were performing closely in case of overall character wise accuracy , but CNN performed much better in case of word wise accuracy overall. So there is a clear-cut victory of deep learning models to achieve the target which we wanted. But the classical machine learning models were not far behind. The reason behind the closeness might be the small size of the images , thereby less features to handle.

14 Conclusion

Handling image data is much more labour intensive than working on a regression/classification problem where the data is almost ready to be fitted into a model. It requires a lot of preprocessing for it to be ready for use. In our work till date we've spend most of our time on perfecting the dataset inorder for it to be usable in different ML models. We had to spend quite a bit of time handling large number of images and preprocessing them to be ready for the models to train on. The steps of the preprocessing have been extensively shown above.

Once we have the images ready for processing in the form of 28x28 images we experimented with different ML models and see their performances. We compared and contrasted a classical machine learning model SVM with a CNN. The CNN comes out on top. MLP and SVM perform at par.

References

https://appen.com/datasets/handwritten-name-transcription-from-an-image/

https://scikit-learn.org/

https://docs.scipy.org/doc/scipy/reference/ndimage.html

https://opencv-python-tutroals.readthedocs.io/en/latest/

https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5

https://docs.python.org/2/library/urllib.html

https://pandas.pydata.org/docs/

https://matplotlib.org/api/pyplot_api.html

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.htr

https://scikit-learn.org/stable/modules/generated/sklearn.decompositic

https://scikit-learn.org/stable/modules/generated/sklearn.decompositic

https://keras.io/

https://www.tensorflow.org/

A Appendices

Convolutional neural networks(CNN)
CNN is a deep learning technique which is similar to the biological neural system which is composed of interconnected artificial neural network units called as neurons.

Support Vector Machines(SVM)
SVM is a supervised technique which can be used for classification and regression. The goal of SVM is to identify a line that maximizes the minimum margin.We want a line such that distance between the data points and the line is minimum.