ABSTRACT

In this report, a recognition system, to recognize handwritten address text is described. The handwriting recognition system takes English handwritten text address as input, recognize the text and convert it into digital text which can be stored for future use and various operations can be performed it.

Handwritten text address recognition is the one of the most interesting and difficult task in field of Deep learning and Image processing nowadays**.** However, Due to the individual differences in handwriting styles, a 100% accurate handwriting recognition software has not yet been developed. Human reading ability is dependent on various factors including the knowledge of grammar, context, etc. Hence, it is difficult to build a machine learning model to recognize handwritten address text as accurately as humans. Here, we have developed a machine learning model to recognize the hand written text address which gives us an accuracy of around 84%.

The development of machine learning model is divided into three non-overlapping but dependent phases which are pre-processing, feature extraction and model training. The machine learning algorithm used for model building is convolutional neural network.

The recognition of handwritten address text is divided into four phases which area pre-processing, segmentation, feature extraction and character recognition. Last phase gives us predicted characters and each predicted character is then brought back into words and address.

INTRODUCTION

Unlike humans which have the high capability to easily understand the texts from a text image, machines are not intelligent enough to recognise the text available in image. Therefore, a large number of efforts have been put forward that try to transform a handwritten text image to machine understandable form.
Handwritten Recognition System is a software which converts handwritten and printed text and images into digitized form such that it can be stored in the database and can be manipulated by machine whenever required. This involves scanning of the text envelope horizontally and vertically to get individual words and then character, extraction of features to form feature vector, and then recognition of character using the classifier.

On-line vs. Off-line

Online text recognition-

Online handwriting text recognition means that the machine recognizes the writing while the user is writing. Online handwriting recognition requires a software that captures the writing as it is written.

Offline text recognition-

Offline handwriting text recognition is performed after the writing is complete. It can be performed later on. Offline handwriting text recognition is a subset of *OCR*.

Literature Review

In the past, lots of paper work is done by the handwriting text. Nowadays, a lot of documents are produced in paper form. But now, we are moving toward a paperless world, digitalized files greatly replace the paper ones, which means scanned copies dominate our workplaces. Working with files is cheaper than processing traditional handwritten documents, because there is no space required for document storage. Moreover each handwritten document exists in one copy, but if we digitize, it visible for all users without any increase in cost or computation time.
As we are moving towards a digitized world, we cannot leave our past data in handwritten or printed papers, we need to digitized this data and store it somewhere so that it can be kept safe and no information is lost. For such application, handwriting text recognitions systems are being developed.

[aidya et al :1999] use a feature-based approach for numeral recognition. They have used a statistical method by assigning weights to each feature and assessing the numerals using these weights. We also use the feature-based approach to recognize the handwritten words. Our approach is different from theirs as words may be written in cursive writing whereas numerals aren't. It is not always possible to break words into characters. Hence we have to use a continuous process of matching the set of features to a database while accounting for the permutations as new features come into view and old ones are discarded. Also, the set of features in the case of alphabets is larger than that in the case of numerals.

[Nicchiotti/Scagliola:1999] have shown some good examples of normalization by removing unwanted variations in cursive handwriting using estimation and compensation of reference lines and slant angle. In the preprocessing of documents for the purpose of recognition and beautification, normalization is an important step to facilitate feature extraction.

[Spitz:1998] use character shape coding process for typed word recognition. They have a small dictionary to which all the words in the document belong. After scanning the words, they are classified on the basis of the regions that they occupy (extending above middle-line, extending below bottom-line or completely between the two). This narrows down the range of possibilities for the word which is then matched against all these possibilities. We had considered this approach but it would have been highly inefficient in our case which is more general as ours is not restricted to a small fraction of a dictionary nor is it restricted to typed documents where the characters are easily distinguishable.

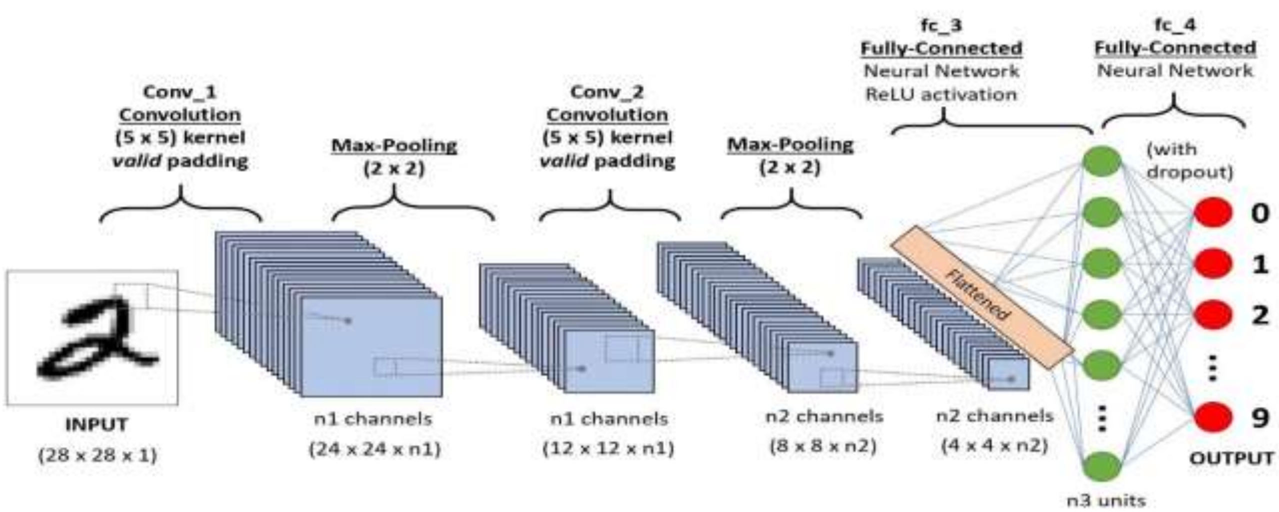
PROPOSED METHODOLOGY AND WORK

Hand written Address reading is very similar and closely related to Hand written text Recognition that can be applied to recognize text. This kind of multimedia recognition is performed on images based on the computer vision and pattern recognition application. We can use image processing, character positioning, character Segmentation, convolutional neural network to solve the problem of image to text recognition.

Convolutional Neural Network Approach

Character Recognition in the handwritten text recognition has important role in optical recognition system which is related directly with success or failure of the system. We used Back Propogation Convolutional Neural Network to optically recognise the image. The basic idea of BP algorithm is the learning process is divided into two phases :

PHASE I: Forward Propogation
PHASE II: Back Propogation



The method that we have used for training and learning of our model is based on Convolutional Neural network. convolutional neural network consists of an input layer,an output layer, multiple number of hidden layers. The starting layers of a Convolutional Neural network consist of convolutional layers, activation function, MaxPooling layers, fully connected layers.

The Convolution layer -It is always the first layer. The image entered into it reading of the input matrix begins at the top left of image. Next we select a smaller matrix there, which is called a **filter** .Then, the filter produces convolution, i.e. moves along the input image. The filter's task is to multiply its values by the original pixel values. All these multplications are summed up. One number is obtained in the end. The filter has read the image in the upper left corner, it moves to right by 1 unit and perform the same operation. After passing the filter through whole image a matrix is obtained, which is smaller than an input matrix.

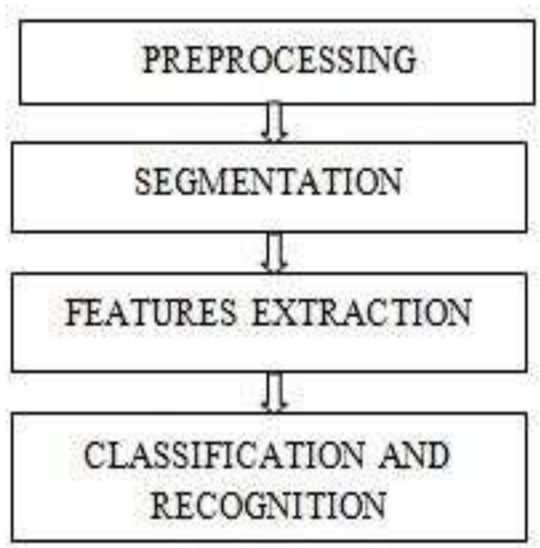
Pooling-Convolutional networks may include local or global pooling layers. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, of 2 x 2. Global pooling acts on all the neurons of the convolutional layer.

Fully Connected Layer - Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the multi-layer perceptron neural network (MLP).

The image of an envelope with a hand written address on a particular form is taken as an input. This image is then preprocessed and broken down into lines, each line into words and each word into character. These characters are then predicted using the pre trained model for hand written character recognition. Each predicted character is then brought back into words and address. Hence the complete address is extracted.

Now this extracted address is automatically entered into the csv file into the correct columns according to the sender’s name, receiver’s name, address, Landmark, City, State and Pin code.

Handwriting recognition can be broken into a number of relatively non-overlapping but dependent phases which are-



1. Pre-processing-

Image pre-processing is the initial step of handwritten address recognition that comprises obtaining a digital image and converting it into suitable form that can be easily processed by computer. Hence pre-processing aims to enhance the quality of image. Some of the image pre-processing techniques are image thresholding, image dilation, image erotion. It applies a number of operations on grey and binary images for making them more readable for the software. The major role of the pre-processing is to filter out the impurities from the image and also to perform smoothing and normalization.

2. Segmentation-

Character recognition involves scanning address envelope written sometime in the past. the individual characters present in the scanned image will need to be extracted.
In image segmentation step, the characters in the image are separated such that they can be passed to recognition system. The techniques for image segmentation are connected component analysis and projection profiles methods.

3. Feature extraction-

In feature extraction step, segmented characters are processes to extract different features. These features are in the form of pixel value. Based on these features, the characters are recognized.

4. Recognition-

In character recognition step, features of segmented image are given to the classifier which maps these features to different categories or classes. these classes are preloaded into the system during the training step.

The process has been completely automated and drafted into an easy User Interface application to demonstrate and to use the Address Reading System efficiently for a bulk number of images.

Tools and Technology Used

We used **Python** syntax for this project. As a framework we used **Keras**, which is a high-level neural network API written in Python. it keras needs a backend for low-level operations. For that, we installed a dedicated software library—Google's **TensorFlow**.

For scientific computation, we installed **Anaconda Navigator .**

As a development environment we used the **PyCharm and Spyder**.

For image pre-processing and segmentation task we used **Opencv** and **Pillow** library which are python's powerful imaging laibrary.

We used **Matplotlib** for data visualization, **Numpy** for various array operations and **Pandas** for data analysis.

For GUI designing, we used python's **tKinter** and **Pygame** library.

For convolutional neural network training and testing we used a dataset of photos of characters and digits from ***https://www.nist.gov***

IMPLEMENTATION & CODING

The entire project can be divided into Training and Testing and Predicting phases:

Training and Testing phase:

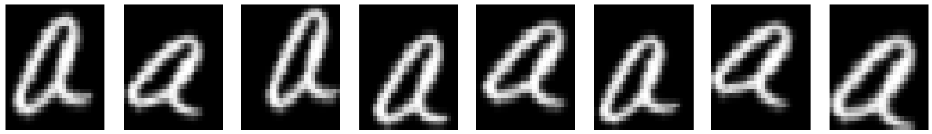
The data set consisted of 28 x 28 Emnist format images in the form of a one dimensional array (784 x 1). These images are then split into two sets, one for training and the other one for testing the accuracy of our model. These images are then converted to grey scale and normalized by dividing the value of each pixel by 255 and labels are converted to one hot encoding format for implementing Machine Learning Algorithms over it and provide better prediction results.

One-hot encoding convert an integer class label to a binary matrix where the array contains single one '1' value and the rest element of the array are '0'.

For example, suppose output is 2 so according to one-hot coding its binary matrix will be [0,0,1,0,0,0,0,0,0,0]

These images have then been augmented by providing slight deviations for better learning of the model.

SINGLE IMAGE



IMAGES AFTER AUGMENTATION

After this preprocessing of the images for training we now use a Convolutional Neural Network based on sequential model for training. Sequential model allows to create a model layer by layer.
Sequential model can be broken down into following layers:

1. The first layer of our model is a convolutional layer called a Convolution2D. The layer has 32 filters,size of each filter is 5×5 and an activation function. This is the input layer and the imput to this layer is a 28*28 image.

2. The Second layer is the MaxPooling layer. MaxPooling layer is used to down-sample the input to enable the model to make assumptions about the features so as to reduce over-fitting. It only considers relevant features and reduces the number of parameters and hence reducing the training time.

3. One more hidden layer with 32 filters/output channels with the size of 3×3 and an activation function.

4. One more MaxPooling layer.

5. The next layer is a regularization layer using dropout called Dropout. It is configured to randomly exclude 10% of neurons in the layer in order to reduce overfitting.

6. Next layer converts the 2D matrix data to a vector called Flatten. It allows the output to be processed by standard fully connected layers.

7. Next layer is a fully connected layer with 128 neurons.
8. Next (last) layer is output layer with 10 neurons(number of output classes) and it uses softmax activation function. Each neuron will give the probability of that class. It’s a multi-class classification that’s why softmax activation function if it was a binary classification we use sigmoid activation function.

```
# create model
model = Sequential()
model.add(Conv2D(32, (5, 5), input_shape=(X_train.shape[1],
X_train.shape[2], 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(number_of_classes, activation='softmax'))
```

Thus after creating model, we compile the code. We used **categorical_crossentropy** as a loss function because it is a multi-class classification problem.

```
# Compile model
model.compile(loss='categorical_crossentropy', optimizer=Adam(),
metrics=['accuracy'])
```

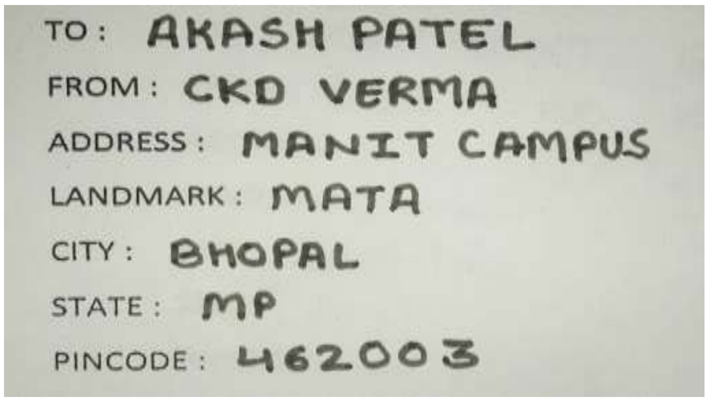
After compiling we train the model on 10 epochs and in batch sizes of 200. On checking for our testing data set, we obtain an accuracy of 87%.This entire model is saved to be used for handwritten address recognition system.

```
# Fit the model
model.fit(X_train, y_train, validation_data=(X_test, y_test),
epochs=10, batch_size=200)
```

Address Prediction Phase:

Pre-processing of image:

First of all the image of envelope,on which the text address is written by hand is required. Below is the example of one case in which an image sample of an address form is taken.



This image is then converted to grey scale and then converted into binary image.

Binary Image

Binary Inverted Image

Segmentation of image:

After preprocessing of this image, it is then segmented into lines first, each of these lines are then segmented into words individually.

Line segmented image

Word segmented image

After being segmented into each word, the images is then segmented into individual letters.

Letter segmentation

Recognition of image:

Each of these letter mages are then converted into Emnist form and the similar kind of pre-processing is performed on these images as done previously while training the data set.

Once the images of each letter are in the required form they are predicted by the pre-trained Convolutional Neural Network based model for optical character recognition. The final prediction for this address form is as follows:

TO: AKASH PATEL
FROM: CKD VERMA
ADDRESS: MAN LT CAMPUS
LANDMARK: MATA
CITY: BHOPAL
STATE: MP
PINCODE: WEBOOS

RESULT ANALYSIS

The handwritten Address Recognition system predicts on the basis of Convolutional Neural Network based Machine Learning Algorithm. The system has been trained independently for English Alphabets and for numeric since a common training on a limited data greatly affected the efficiency of the predicting system.

TRAINING ALPHABETS:

The data set used for this purpose consisted of 3000 images of each alphabet. Since the number of classes to be divided were 26, 3000 images of each were not enough to provide an acceptable accuracy hence data augmentation was performed. After augmenting the image data, each image produce 4 augmented images, thus equaling to 12000 new images of each alphabets. Thus a total of 390,000 images were trained on 10 epochs. On performing for more epochs, the accuracy hardly showed an improvement as it had already reached the saturation. The maximum accuracy hence achieved was 89%.

TRAINING NUMERICS:

The data set used consisted of 10000 image of each number, since the number of classes were only 10, data augmentation was not performed as it would not have improved the accuracy much. A total of 100,000 images were trained on 10 epochs and the final accuracy achieved was 92%.

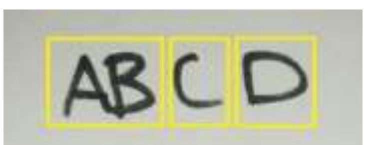
TESTING ON ADDRESS FORMS:

The extraction of handwritten address form image worked fairly well, predicting more than 84% of the characters correctly after segmentation and prediction. The data was distributed in correct attributes in most of the cases.

CONCLUSIONS, LIMITATIONS AND FUTURE SCOPE

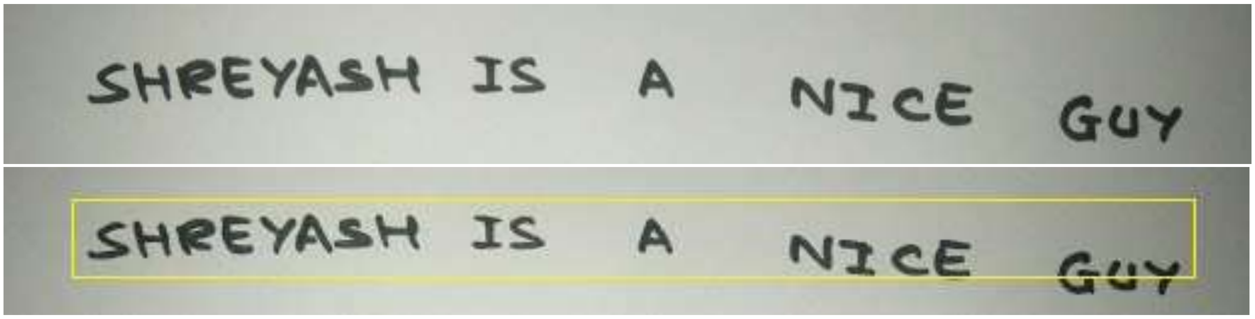
In this system, a machine learning classifier is designed for the recognition of handwritten address from the envelope. At first plate location of text address is extracted. then separated the characters individually by segmentation. Finally recognition of characters is done.

The address extraction system based on segmentation of the input image and prediction on the numeric and alphabetic trained model performs fairly well. But this model does have certain restriction and fails to perform when the following cases occur:

1) Overlapping of alphabets:
   The handwritten characters in the text need to be written clearly and a little spaciously. If a character overlaps with the other, the segmentation of the image is not done properly and as the system is based on recognition of each individual character, the system fails to work in this case and performs poor producing absurd result.
   

2) Style of writing:
   The model being trained only on the simple alphabets doesn't work on cursive hand written texts.
3) Change of Font size:
   The segmentation of the image is greatly affected if the font of the handwriting is changed in line. Since the image is scanned from left to right, an intermediate change in font would be ignored and hence the segmentation of the image would be faulty.

4) Skew lines:
   The skew lines will also be ignored as the image is scanned from left to right. The segmentation performed would not be proper in this case as well.



5) Broken text:
   If the handwritten alphabet is broken in between the prediction of the image can be absurd and inaccurate.

Clear image and will be predicted correctly

Broken image and will not be predicted correctly

6) Low resolution of the characters-If the image is of low resolution, model will face difficulties in recognition the character as well as in finding the Inter-word and Intra-word gap.

Methods can be developed in the future to segment the images more efficiently and correctly as this is the first and major step in recognition of handwritten characters.

With slight modifications the same system can be applied for other applications such as reading from bank cheques and various kinds of forms.

Though work on recognition of handwritten text is being worked on for many years now but still languages like Hindi, Spanish and Mandarin and other large scale spoken languages have been unexplored.

REFERENCES

INTERNATIONAL PAPERS REFFERED:

[1] N. Arica and F. Yarman-Vural, "An Overview of Character Recognition Focused on Off-line Handwriting", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol.31 (2), pp. 216 - 233. 2001.

[2] Mahmoud, S.A., & Al-Badr, B., 1995, Survey and bibliography of Arabic optical text recognition. Signal processing, 41(1), 49-77.

[3] V.K. Govindan and A.P. Shivaprasad, "Character Recognition – A review," Pattern Recognition, Vol. 23, no. 7, pp. 671- 683, 1990.

[4] Qadri, M.T., & Asif, M, 2009, Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition presented at International Conference on Education Technology and Computer, Singapore, 2009. Singapore: IEEE.

BOOKS:

1. Character and Handwriting Recognition: Expanding Frontiers (Series in Computer Science) by P. S. P. Wang.
2. Deeplearning and Neural Networks by Michael Nielsen.

WEBSITES:

1.https://towardsdatascience.com/image-pre-processing-c1aec0be3edf

2. https://towardsdatascience.com/a-comprehensive-guide to convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3. https://becominghuman.ai/towards-forms-text-recognition-using-deep-learning-f8b39840b984