Data Mining on Handwriting Recognition

PROJECT REPORT

Group-17

Eshita Shukla
171071059
epshukla_b17@ce.vjti.ac.in

Kartiki Sanap
171071029
kksanap_b17@ce.vjti.ac.in

Nidhee Kamble
171071037
ndkamble_b17@ce.vjti.ac.in

I. INTRODUCTION

A. Background

Data mining is a powerful method for mining useful patterns or data from image and textual data sets. Technique that performs handwriting recognition is able to detect and acquire characters from paper documents, images and other sources. There exist various data mining techniques which are used for Handwritten Character Recognition. Text detection is a process of detecting and finding those areas of the image that contain texts. Text detection is the first step in obtaining textual information. There are various factors which make text detection difficult are: some text variations related to style orientation, size and alignment, low contrast and complex backgrounds. Any computer that performs handwriting recognition is able to detect and acquire characters in paper documents, images and other sources. It can easily convert them into encrypted form i.e. machine-encoded form.

B. Motivation

The purpose of this project is to take English handwritten documents/word images as input and recognize the text contained in it.

Lots of work has been done in the field of character recognition but not much for analyzing a complete document. Recognizing the text of a document would be useful in many diverse applications like reading medical prescriptions, bank cheques and other official documents. It will also find uses in detective or police departments in applications like handwriting based person identification, identifying real from forged documents, etc.

C. Objective

The objective of this project is to identify handwritten characters with the use of neural networks. We have to construct a suitable neural network and train it properly. The program should be able to extract the characters one by one and map the target output for training purpose. After automatic processing of the image, the training dataset has to be used to train "classification engine" for recognition purpose.

D. Scope

1. System will be designed in way to ensure that offline Handwritten Recognition of English characters.
2. Use of Neural Network for classification.
3. Large number of training data set will improve the efficiency of the suggested approach.

E. Methodology

1) SDG Classifier

Stochastic Gradient Descent (SGD) is an approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. Even though SGD has been around in the machine learning community for a long time, it has received a considerable amount of attention just recently in the context of large-scale learning.

Strictly speaking, SGD is merely an optimization technique and does not correspond to a specific family of machine learning models. It is only a way to train a model.

For training on the above dataset, we used : classifier = SGDClassifier(random_state=42)

2) Handwritten Text Recognition using CNN, RNN, and CTC

We use a NN for our task. It consists of convolutional NN (CNN) layers, recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer.

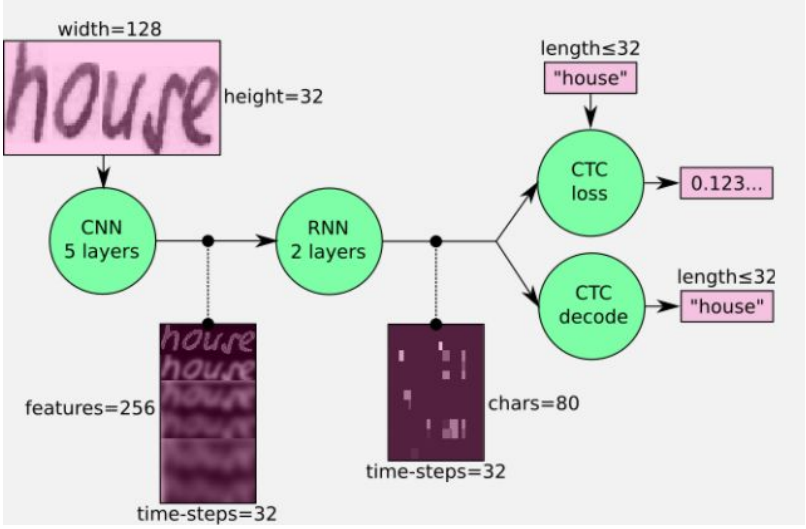


Fig. 1. Overview of the NN operations (green) and the data flow through the NN (pink)

Operations

1) **CNN**: The input image is fed into the CNN layers. These layers are trained to extract relevant features from the image. Each layer consists of three operations. First, the convolution operation, which applies a filter kernel of size 5×5 in the first two layers and 3×3 in the last three layers to the input. Then, the non-linear RELU function is applied. Finally, a pooling layer summarizes image regions and outputs a downsized version of the input. While the image height is downsized by 2 in each layer, feature maps (channels) are added, so that the output feature map (or sequence) has a size of 32×256.

2) **RNN**: The feature sequence contains 256 features per time-step, the RNN propagates relevant information through this sequence. The popular Long Short-Term Memory (LSTM) implementation of RNNs is used, as it is able to propagate information through longer distances and provides more robust training-characteristics than vanilla RNN. The RNN output sequence is mapped to a matrix of size 32×80. The IAM dataset consists of 79 different characters, further one additional character is needed for the CTC operation (CTC blank label), therefore there are 80 entries for each of the 32 time-steps.

3) **CTC**: While training the NN, the CTC is given the RNN output matrix and the ground truth text and it computes the loss value. While inferring, the CTC is only given the matrix and it decodes it into the final text. Both the ground truth text and the recognized text can be at most 32 characters long.

We can also view the NN in a more formal way as a function which maps an image (or matrix) M of size W×H to a character sequence (c1, c2, …) with a length between 0 and L.
The text is recognized on character-level, therefore words or texts not contained in the training data can be recognized too (as long as the individual characters get correctly classified).

$$\underset{W\times H}{M} \rightarrow \underset{0\le n\le L}{(c_1, c_2, \ldots, c_n)}$$

Fig.2. NN as a formal function

II. RELATED WORK

A. *Offline Handwritten English Numerals Recognition using Correlation Method*

In this paper the author has proposed a system to efficiently recognize the offline handwritten digits with a higher accuracy than previous works done. Also previous handwritten number recognition systems are based on only recognizing single digits and they are not capable of recognizing multiple numbers at one time.So the author has focused on efficiently performing segmentation for isolating the digits.[1]

B. Intelligent Systems for Off-Line Handwritten Character Recognition: A Review

Handwritten character recognition is always a frontier area of research in the field of pattern recognition and image processing and there is a large demand for Optical Character Recognition on handwritten documents. This paper provides a comprehensive review of existing works in handwritten character recognition based on soft computing technique during the past decade.[2]

C. An Overview of Character Recognition Focused on Off-Line Handwriting

Character recognition (CR) has been extensively studied in the last half century and progressed to a level sufficient to produce technology driven applications. Now, the rapidly growing computational power enables the implementation of the present CRmethodologies and creates an increasing demand on many emerging application domains, which require more advanced methodologies.[3]

D.Image preprocessing for optical character recognition using neural networks

Primary task of this master's thesis is to create a theoretical and practical basis of preprocessing of printed text for optical character recognition using forward-feed neural networks. Demonstration application was created and its parameters were set according to results of realized experiments.[4]

E. Recognition for Handwritten English Letters: A Review

Character recognition is one of the most interesting and challenging research areas in the field of Image processing. English character recognition has been extensively studied in the last half century. Nowadays different methodologies are in widespread use for character recognition. Document verification, digital library, reading bank deposit slips, reading postal addresses, extracting information from cheques, data entry, applications for credit cards, health insurance, loans, tax forms etc. are application areas of digital document processing. This paper gives an overview of research work carried out for recognition of handwritten English letters. In Hand written text there is no constraint on the writing style. Hand written letters are difficult to recognize due to diverse human handwriting style, variation in angle,

size and shape of letters. Various approaches of hand written character recognition are discussed here along with their performance.[5]

F. Diagonal Based Feature Extraction For Handwritten Alphabets Recognition System Using Neural Network

An off-line handwritten alphabetical character recognition system using multi layer feed forward neural network is described in the paper. A new method, called, diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets. Fifty data sets, each containing 26 alphabets written by various people, are used for training the neural network and 570 different handwritten alphabetical characters are used for testing. The proposed recognition system performs quite well, yielding higher levels of recognition 6 accuracy compared to the systems employing the conventional horizontal and vertical methods of feature extraction. This system will be suitable for converting handwritten documents into structural text form and recognizing handwritten names.[6]

III. IMPLEMENTATION

A. Dataset Information

1) Kaggle: A-Z Handwritten Alphabets (.csv)

This dataset contains 26 folders (A-Z) containing handwritten images in size 2828 pixels, each alphabet in the image is centre fitted to the 2020 pixel box. Each image is stored as Gray-Level. This dataset is already pre-processed.

2) IAM Handwriting Dataset

Contains forms of handwritten English text which can be used to train and test handwritten text. The database contains a form of unconstrained handwritten text, which was scanned at a resolution of 300 dpi and saved as PNG images with 256 gray levels.

This dataset is structured as follows:

a. 657 writers contributed samples of their handwriting
b. 1539 pages of scanned text
c. 5685 isolated and labeled sentences
d. 13353 isolated and labeled text lines
e. 15320 isolated and labeled words

This dataset is already pre-processed.

3) Custom (self-generated) dataset

1. Photos of handwritten text (raw)
2. Digital text generated from available handwritten letter dataset

Fig. 3.Word images generated from custom dataset of hand-written letters (cleaned and binarised)

B. Data set preprocessing

1) Gaussian Blur

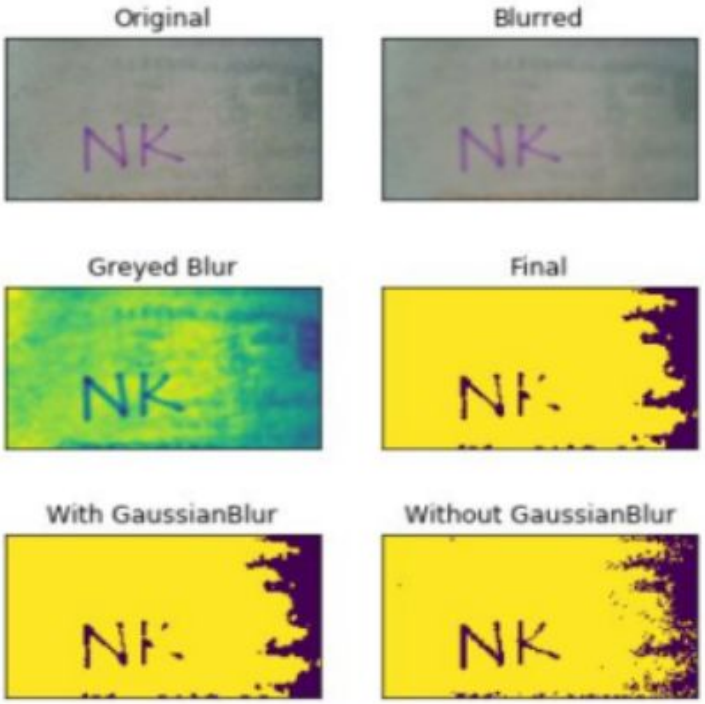
In image processing, a **Gaussian blur** (also known as **Gaussian** smoothing) is the result of **blurring** an image by a **Gaussian** function (named after mathematician and scientist Carl Friedrich **Gauss**). It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

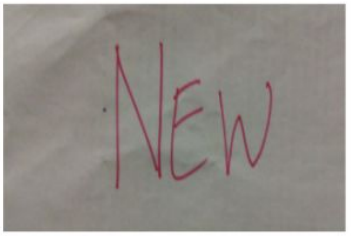
We have used Gaussian blur for noise reduction, with a sigma value of 21 on the X and Y axes both.

2) Thresholding

An image processing method that creates a bitonal (aka binary) image based on setting a threshold value on the pixel intensity of the original image. While most commonly applied to grayscale images, it can also be applied to color images.

Using **thresholding**, we have converted **images** from colored or grayscale into a binary **image**, i.e., one that is simply black and white (each pixel is either 0 or 1). Each image is perfectly binary due to threshold being 127.


Original
Blurred
Greyed Blur
Final
With GaussianBlur
Without GaussianBlur



C. Model Performance and Comparison

We see a huge increase in accuracy with every new Epoch. After each Epoch of training on the training set, we tested the created model on the test set too. We only saved the model if there was an increase in accuracy after every epoch (compared to the previous one). If not, we stop the training in order to prevent overfitting.

There are 115 images in the **test set**.

First 8 Epochs (1-14):

```
Character error rate: 26.071636%. Word accuracy: 48.695652%.
Character error rate improved, save model
```

```
Character error rate: 22.378340%. Word accuracy: 54.278261%.
Character error rate improved, save model
W1121 20:48:04.467337 139645412550464 deprecation.py:323] From /
```

```
[ERR:3] "wish" -> "wol"
[ERR:1] "I" -> ","
[OK] "went" -> "went"
[OK] "to" -> "to"
[OK] "that" -> "that"
Character error rate: 18.002587%. Word accuracy: 60.765217%.
Character error rate improved, save model
```

```
[ERR:1] "I" -> ","
[OK] "went" -> "went"
[OK] "to" -> "to"
[OK] "that" -> "that"
Character error rate: 17.141710%. Word accuracy: 62.104348%.
Character error rate improved, save model
```

```
[ERR:1] "I" -> ","
[OK] "went" -> "went"
[OK] "to" -> "to"
[OK] "that" -> "that"
Character error rate: 16.294215%. Word accuracy: 63.547826%.
Character error rate not improved
```

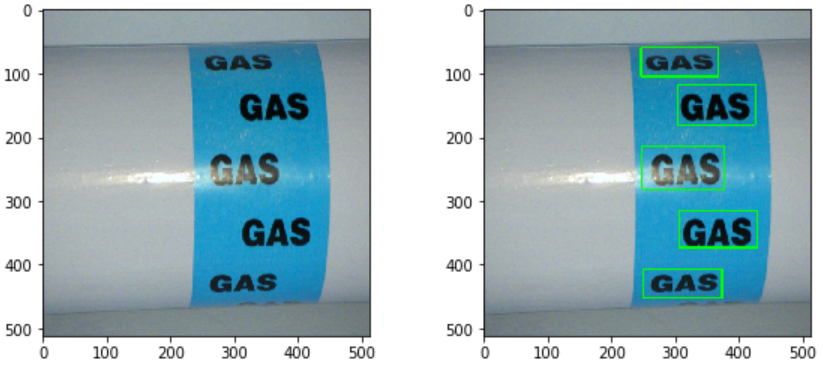
Our model gives an accuracy of 62% after being trained for just 14 epoch on 500 batches. We intend to increase the Epochs which will probably result in a better accuracy. Since we trained the model locally, the computational power for running a Convolutional Neural Network wasn't enough. We have tried to use an online GPU. This accuracy increases to 82%.

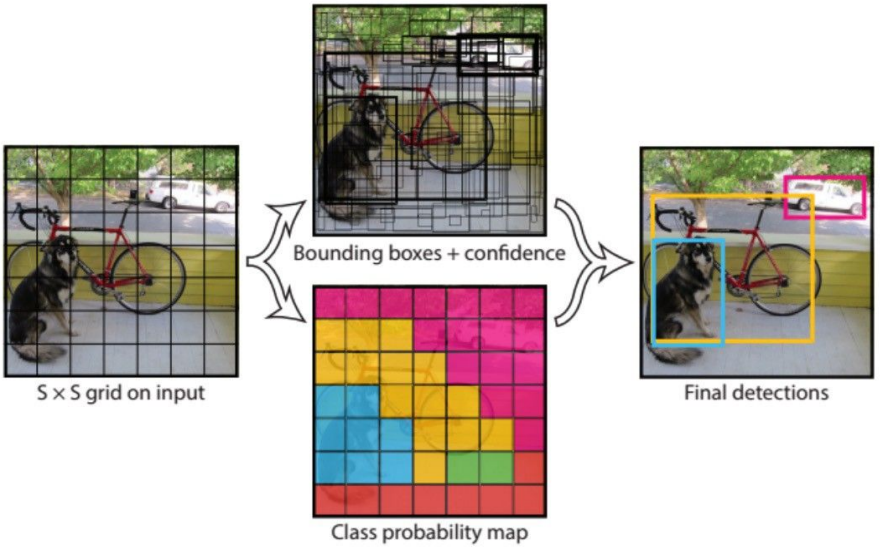
```
W1105 16:24:30.915715 140351396333376 module_wrapper.py:139] From /home/thepe
nstead.

Init with stored values from ../model/snapshot-1
Recognized: "little"
Probability: 0.28522897
(base) theperson@eshitapshukla:~/Downloads/SimpleHTR-master/src$
```

Fig. 5. The image and the detected text

Compared to other region proposal classification networks (fast RCNN) which perform detection on various region proposals and thus end up performing prediction multiple times for various regions in a image, Yolo architecture is more like FCNN (fully convolutional neural network) and passes the image (nxn) once through the FCNN and output is (mxm) prediction. This the architecture is splitting the input image in mxm grid and for each grid generation 2 bounding boxes and class probabilities for those bounding boxes.

We trained a YOLO model to detect word chunks from an image.





Metrics Used for model comparison

```
Clasification report:
             precision    recall  f1-score   support

          1       1.00      0.76      0.86        71
          2       1.00      0.84      0.91        43
          3       1.00      0.74      0.85        89
          4       0.98      0.95      0.96       288
          5       0.87      1.00      0.93       367

avg / total       0.94      0.93      0.93       858


Confussion matrix:
[[ 54   0   0   0  17]
 [  0  36   0   1   6]
 [  0   0  66   5  18]
 [  0   0   0 273  15]
 [  0   0   0   0 367]]
```

The confusion matrix is shown as a heat map; with the colour of each cell corresponding to its value in the matrix. The rows represent actual features, and the columns represent predicted features. The value of a cell at index pair (i, j) represents weight of feature i being recognised as feature j by the model.

IV. CONCLUSION AND FUTURE SCOPE

Conclusion

Many regional languages throughout the world have different writing styles which can be recognized with HCR systems using proper algorithms and strategies. We are learning to recognize English characters. It has been found that recognition of handwritten characters becomes difficult due to the presence of odd characters or similarity in shapes for multiple characters. Scanned image is pre-processed to get a cleaned image and the characters are isolated into individual characters. Preprocessing work is done in which normalization, filtration is performed using processing steps which produce noise free and clean output. Managing our evolution algorithm with proper training, evaluation and other stepwise processes will lead to successful output of the system with better efficiency. Use of some statistical features and geometric features through neural networks will provide better recognition results of English characters. This work will be helpful to the researchers for the work towards other scripts.

Future Scope

This work further extended to the character recognition for other languages. It can be used to convert the fax and newspapers into text format.In order to recognize words,sentences or paragraphs we can use multiple ANN for classification.It can be used in post offices for reading postal addresses.

REFERENCES

[1] Isha Vats, Shamandeep Singh, "**Offline Handwritten English Numerals Recognition using Correlation Method**",InternationalJournal of Engineering Research and Technology (IJERT): ISSN: 2278-0181 Vol. 3 Issue 6, June 2014.

[2] Shabana Mehfuz,Gauri katiyar, '**Intelligent Systems for Off-Line Handwritten Character Recognition: A Review**" ,International Journal of Emerging Technology and Advanced Engineering Volume 2, Issue 4, April 2012.

[3] Rahul KALA, Harsh VAZIRANI, Anupam SHUKLA and Ritu TIWARI, "**An Overview of Character Recognition Focused on Off-Line Handwriting**",IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART APPLICATIONS AND REVIEWS, VOL. 31, NO. 2, MAY 2001.

[4] Miroslav NOHAJ, Rudolf JAKA, "**Image preprocessing for optical character recognition using neural networks**"Journal of Patter Recognition Research, 2011.

[5]] Nisha Sharma et al, "**Recognition for handwritten English letters: A Review**"International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 7, January 2013.

[6] J.Pradeep et al,, "**Diagonal based feature extraction for handwritten alphabets recognition System using neural network**"International Journal of Computer Science and Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.