A Project Report on

HANDWRITTEN CHARACTER RECOGNITION

Submitted in partial fulfillment of the requirements

in

Computer Department

By

Shweta Chauhan - 17102009
Nehal Barot - 17102060
Anjali Solanki - 18202003
Yash Awasthi - 15202013

Under the Guidance of

Prof. Amol Kalugade

Department of Computer Engineering
A.P. Shah Institute of Technology G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Approval Sheet

This Project Report entitled ***"Handwritten Character Recognition"*** Submitted by ***"Nehal Barot " (17102064), "Yash Awasthi"(15202013), "Shweta chauhan "(17102009), "Anjali Solanki"(18202003)***is approved for the partial fulfillment of the requirement in ***Computer Engineering*** from ***University of Mumbai*** .

Prof. Amol Kalugade

Guide

Prof. S.H.Malave

Head, Computer Engineering Department

Place: A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Handwritten Character Recognition"*** submitted by ***"Nehal Barot" (17102064),"Yash Awasthi" (15202013),"Anjali Solanki" (18202003),"Shweta Chauhan" (17102009)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer.***,to the University of Mumbai,is a bonafide work carried out during the academic year 2020-2021.

Prof. Amol Kalugade
Guide

Prof. S.H.Malave
Head,Computer Engineering Department

Dr. Uttam D.Kolekar
Principal

External Examiner

Place: A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

Shweta Chauhan - 17102009
Nehal Barot - 17102060
Anjali Solanki - 18202003
Yash Awasthi - 15202013

Date:

Contents

1. project conception and initiation

1.1 Abstract

Handwriting character recognition remains largely unsolved problems due to the presence of many handwritten characters present around the world. There are many existing advance methods which do not lead to a proper solution for handwriting character recognition.The handwriting character recognition will be done by using Keras, Convolutional Neural Network and Tensor Flow. Automatic identification of handwritten script facilitates many important applications such as automatic transcription of multilingual documents and search for documents on the Web containing different particular scripts. The increase in usage of handheld devices which accept handwritten input has created a growing demand for algorithms that can efficiently analyze and retrieve handwritten data. Handwritten character recognition is an area of research where many researchers have presented been done and is still an area under research to achieve more accuracy. This project is aimed at developing software which is helpful to recognize English characters. Our project is restricted to English characters only. It is still possible to further develop to recognize the characters of many languages present in the world. The most common use of neural network is pattern recognition. Neural network is used for solving problems such as the recognition of patterns, classifying the patterns into groups, data mining etc. Neural network contains a vector which have the pattern information as well as a target vector. Here the pattern information could be an image and handwritten data. Neural network attempts to determine the handwritten characters or image that the neural network has learned. Neural network is designed to take the input data and classify the data into groups. These groups can be fuzzy or not clear.

1.2 Objectives

1. To provide an easy user interface to input the source image.
2. User should be able to upload the image.
3. System should be able to pre-process the given input to suppress the Background.
4. System should detect text regions present in the image.
5. System should retrieve text present in the image and display them to the User.

1.3 Literature review

A..Neural Networks For Handwritten English Alphabet Recognition The paper makes use of Neural Networks to recognize the English handwritten alphabets. The alphabets are represented as binary values in the form of 0 and 1. These binary images are then used as an input to the feature extraction phase, and the output of this phase is fed as an input in Neural Network system. Similar appearing alphabets were seen to be misclassified in the experiments carried out, as the data set over which the tests were carried out was small. The larger the data set used the more accurate results the system would generate

B. Direction Based Feature Extraction Paper presents a feature extraction method referred as the direction feature to recognize the handwritten characters. This new direction feature extraction method proved to give better accuracy result on the segmented characters when compared to the transition feature extraction method and other methods as well. The recognition rate is above 80% proven experimentally. The characters are segmented automatically from the Cursive Digit and character Recognition (CEDAR) benchmark

C. Diagonal Based Feature Extraction System Using Neural Network The characters in this paper are recognized using a new technique called the diagonal feature extraction technique.There were two approaches used, firstly with 54 features and secondly with 69 features which constituted the Neural Network character recognition system. The diagonal feature extraction method was compared with the traditional horizontal and vertical feature extraction methods, by training the neural network with both the methods. Six recognition neural networks were built in process. The results experimentally revealed that 69 features gave a better accuracy rate than 54 features. The system presented in the paper seems ideal to convert handwritten text documents into structural format.

D. Handwritten English Character Recognition Using Neural Network The paper demonstrates the use of Feed Forward Algorithm along with Back Propagation Algorithm. We paper successfully recognizes characters using a multilayer perceptron with 1 hidden layer. Higher performance can be achieved in Back Propagation once the number of hidden nodes to be used is successfully determined. The recognition of characters is proved to be better and gives an accuracy of 70% and above for English handwritten characters.

E. Digital Image Processing Techniques In Character Recognition The paper explains the various phases of image processing being used in character recognition such as Image Restoration, Image Enhancement, Segmentation, Feature Extraction and Classification with Recognition. These techniques are used along with Neural Network due to its high tolerance to noise. This helps in removing all unwanted signals in images that are distorted over years. Successful character recognition becomes possible for such documents as well and the systems generate perfect results.

F. Character Recognition Using Neural Network In this paper, neural network is used to recognize characters. It improves the recognition rate as the system is developed for isolated English characters – A to Z. The paper makes use of feed forward back propagation and the Neural Network is trained using Back Propagation to classify and recognize characters. The English characters are represented in the binary form as is then fed to the Neural Network for further processing. The paper fails to recognize cursive handwritings.

G. Handwritten Digit Recognition The paper presents a comparison of the feature vectors, the feature extraction strategies are proven to perform better than their baseline counterparts. The gradient feature extraction technique works best for gray
scale images giving the most accuracy rate of characters and also the Normalization-Cooperated Feature Extraction (NCFE) yields a good performance result. The gradient feature extraction technique is applied on the gray scale images and other feature extraction techniques are applied on the binary images. The combination of feature extraction along with normalization has proven to yield higher accuracy rates of
character recognition.

H. Analysis Drawn
After successfully reviewing a number of papers, the following experimental results were analyzed and a comparison of the character recognition accuracy rates is given below along with the different algorithms and techniques used in each corresponding paper.

1.4 Problem Definition

Traditionally, the system exists only for the character recognition but our project extends this further and converts those characters into digital texts and store them in textual form. Moreover characters are only read through OCR (Optical Character Reader). These text are read by OCR are printed on product items, in our project we aim to recognize and convert the handwritten notes to their digital textual from. Traditional systems are not highly efficient but our project gives 85-90 per cent efficiency due to large datasets and precisely trained neural network model for character recognition and correction.

1.5 Scope

1. Converting handwritten notes to digital notes.
2. Converting old scriptures to digital library.
3. House Number Recognition through Google street images.
4. Physical written regional language to other.
5. Quick digitization of printed hand-filled form.
6. Converting postal address to digital texts.

1.6 Technology stack

1.Keras

2.Python OpenCV

3.Tensor Flow

4.Neural Networks

1.7 Benefits for environment and society

Healthcare and pharmaceuticals

Patient prescription digitization is a major pain point in healthcare/pharmaceutical industry. For example Roche is handling millions of petabytes of medical PDFs daily. Another area where handwritten text detection has key impact is patient enrollment and form digitization. By adding handwriting recognition to their toolkit of services, hospitals/pharmaceuticals can significantly improve user experience

Insurance

A large insurance industry receives more than 20 million documents a day and a delay in processing the claim can impact the company terribly. The claims document can contain various different handwriting styles and pure manual automation of processing claims is going to completely slow down the pipeline

Banking

People write cheques on a regular basis and cheques still play a major role in most non-cash transactions. In many developing countries, the present cheque processing procedure requires a bank employee to read and manually enter the information present on a cheque and also verify the entries like signature and date. As a large number of cheques have to be processed every day in a bank a handwriting text recognition system can save costs and hours of human work

Online Libraries

Huge amounts of historical knowledge is being digitized by uploading the image scans for access to the entire world. But this effort is not very useful until the text in the images can be identified which can be indexed, queried and browsed. Handwriting recognition plays a key role in bringing alive the medieval and 20th century documents, postcards, research studies etc.

1.6 Problem Definition

Handwritten Character difficult to access in an efficient manner

1.7 Scope

1. Converting handwritten notes to digital notes.
2. Converting old scriptures to digital library.
3. House Number Recognition through Google street images.
4. Physical written regional language to other.
5. Quick digitization of printed hand-filled form.
6. Converting postal address to digital texts.

1.6 Technology stack

1.Keras

2.Python OpenCV

3.Tensor Flow

1.7 Benefits for environment and society

Healthcare and pharmaceuticals

Patient prescription digitization is a major pain point in healthcare/pharmaceutical industry. For example Roche is handling millions of petabytes of medical PDFs daily. Another area where handwritten text detection has key impact is patient enrollment and form digitization. By adding handwriting recognition to their toolkit of services, hospitals/pharmaceuticals can significantly improve user experience

Insurance

A large insurance industry receives more than 20 million documents a day and a delay in processing the claim can impact the company terribly. The claims document can contain various different handwriting styles and pure manual automation of processing claims is going to completely slow down the pipeline

Banking

People write cheques on a regular basis and cheques still play a major role in most non-cash transactions. In many developing countries, the present cheque processing procedure requires a bank employee to read and manually enter the information present on a cheque and also verify the entries like signature and date. As a large

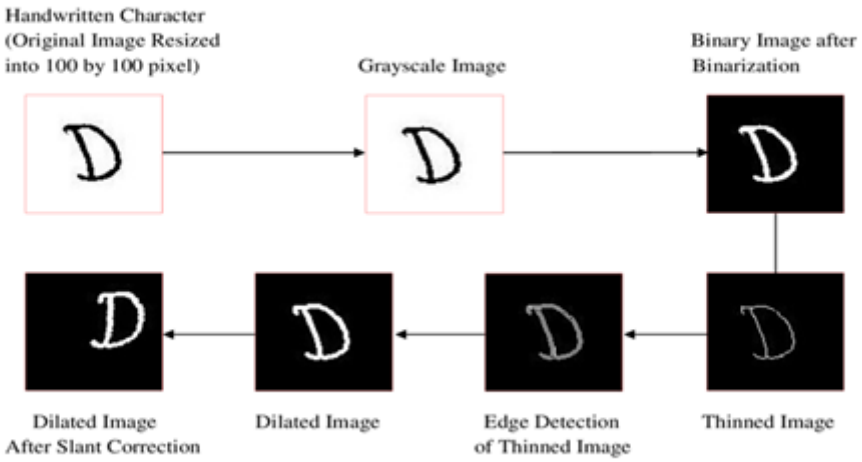
number of cheques have to be processed every day in a bank a handwriting text recognition system can save costs and hours of human work

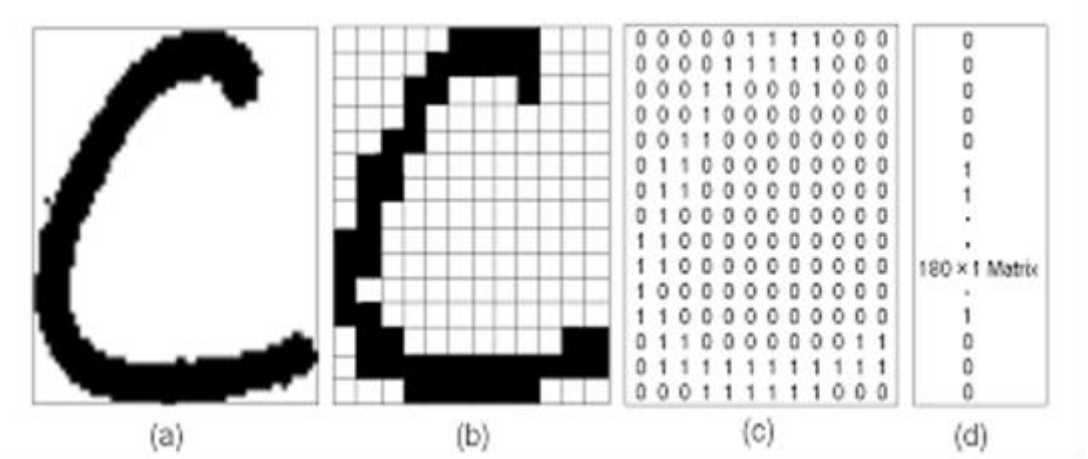
Online Libraries

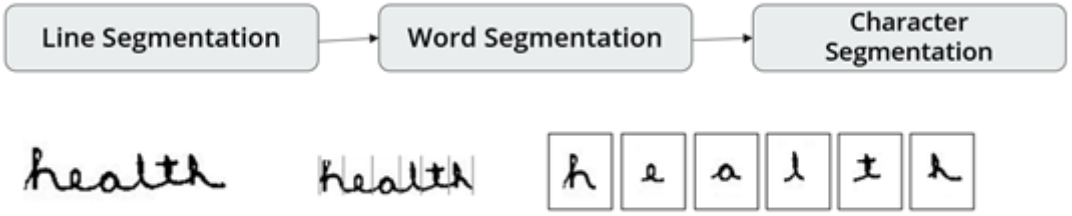
Huge amounts of historical knowledge is being digitized by uploading the image scans for access to the entire world. But this effort is not very useful until the text in the images can be identified which can be indexed, queried and browsed. Handwriting recognition plays a key role in bringing alive the medieval and 20th century documents, postcards, research studies etc.

2.Project Design

2.2 Design(flow of modules)







3. Impelementation

3.1 proposed system

In this section, the proposed recognition system is described. A typical handwriting recognition system consists of pre-processing, segmentation, classification and post processing stages. Image acquisition, the recognition system acquires a scanned image as an input image. The image should have a specific format such as JPEG, BMT etc. This image is acquired through a scanner, digital camera or any other suitable digital input device.

Pre-processing: The pre-processing is a series of operations performed on the scanned input image. It essentially enhances the image rendering it suitable for segmentation. The various tasks performed on the image in pre-processing stage are shown in Fig.2. Binarization: process converts a gray scale image into a binary image using global thresholding technique. Dilation of edges in the binarized image is done using sobel technique, dilation the image and filling the holes present in it are the operations performed in the last two stages to produce the pre-processed image suitable for segmentation.

Segmentation: In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character In the proposed system, the pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. This labeling provides information about number of characters in the image. Each individual character is uniformly resized into 30X20 pixels.

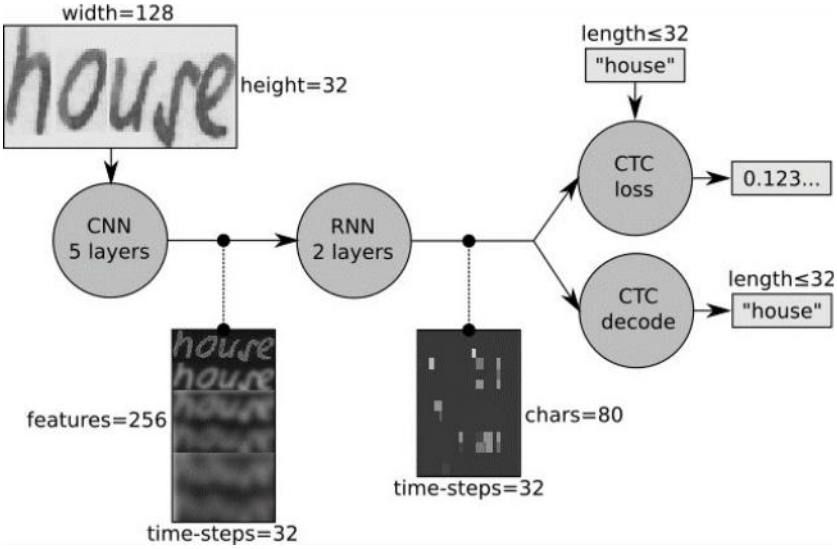
Classification and Recognition: The classification stage is the decision making part of the recognition system .A feed forward back propagation neural network is used in this work for classifying and recognizing the handwritten .characters. The 600 pixel derived from the resized character in the segmentation stage form the input to the classifier. The neural classifier consists of two hidden layers besides an input layer and an output layer as shown in. The hidden layers use log sigmoid activation function and the output layer is a competitive layer as one of the characters is required to be identified at any point in time. The total number of neurons in the output layer is 26 as the proposed system is designed to recognize English alphabets

Post-processing: Post-processing stage is the final stage of the proposed recognition system. It prints the corresponding recognized characters in the structured text form by calculating equivalent ASCII value using recognition index of the test samples.

3.1.2 algorithms

In our project we are using

- CNN
- RNN
- CTC



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 operation. 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.

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), and therefore there are 80 entries for each of the 32 time-steps

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.

3.1.3 pseudo code

```
def ctc_lambda_func(args):
    y_pred, labels, input_length, label_length = args
    # the 2 is critical here since the first couple outputs of the RNN
    # tend to be garbage
    y_pred = y_pred[:, 2:, :]
    return K.ctc_batch_cost(labels, y_pred, input_length, label_length)

```

```
labels = Input(name='gtruth_labels', shape=[max_str_len], dtype='float32')
input_length = Input(name='input_length', shape=[1], dtype='int64')
label_length = Input(name='label_length', shape=[1], dtype='int64')

ctc_loss = Lambda(ctc_lambda_func, output_shape=(1,), name='ctc')([y_pred, labels, input_length, label_length])
model_final = Model(inputs=[input_data, labels, input_length, label_length], outputs=ctc_loss)
```

```
# the loss calculation occurs elsewhere, so we use a dummy lambda function for the loss
model_final.compile(loss={'ctc': lambda y_true, y_pred: y_pred}, optimizer=Adam(lr = 0.0001))

model_final.fit(x=[train_x, train_y, train_input_len, train_label_len], y=train_output,
                validation_data=([valid_x, valid_y, valid_input_len, valid_label_len], valid_output),
                epochs=15, batch_size=128)
```

```

# CNN to RNN
inner = Reshape(target_shape=((64, 1024)), name='reshape')(inner)
inner = Dense(64, activation='relu', kernel_initializer='he_normal', name='dense1')(inner)

## RNN
inner = Bidirectional(LSTM(256, return_sequences=True), name = 'lstm1')(inner)
inner = Bidirectional(LSTM(256, return_sequences=True), name = 'lstm2')(inner)

## OUTPUT
inner = Dense(num_of_characters, kernel_initializer='he_normal',name='dense2')(inner)
y_pred = Activation('softmax', name='softmax')(inner)

model = Model(inputs=input_data, outputs=y_pred)
model.summary()

```

```
input_data = Input(shape=(256, 64, 1), name='input')
#Padding is a term relevant to convolutional neural networks as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN.
#For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero.
inner = Conv2D(32, (3, 3), padding='same', name='conv1', kernel_initializer='he_normal')(input_data)
inner = BatchNormalization()(inner)
inner = Activation('relu')(inner)
inner = MaxPooling2D(pool_size=(2, 2), name='max1')(inner)

inner = Conv2D(64, (3, 3), padding='same', name='conv2', kernel_initializer='he_normal')(inner)
inner = BatchNormalization()(inner)
inner = Activation('relu')(inner)
inner = MaxPooling2D(pool_size=(2, 2), name='max2')(inner)
inner = Dropout(0.3)(inner)

inner = Conv2D(128, (3, 3), padding='same', name='conv3', kernel_initializer='he_normal')(inner)
inner = BatchNormalization()(inner)
inner = Activation('relu')(inner)
inner = MaxPooling2D(pool_size=(1, 2), name='max3')(inner)
inner = Dropout(0.3)(inner)

```

3.1.4 platforms for execution

WINDOWS OS
GOOGLE COLAB

4. Results


```
Epoch 1/15
235/235 [==============================] - 155s 394ms/step - loss: 36.7745 - val_loss: 20.7340
Epoch 2/15
235/235 [==============================] - 87s 371ms/step - loss: 20.2480 - val_loss: 20.0894
Epoch 3/15
235/235 [==============================] - 87s 369ms/step - loss: 19.9253 - val_loss: 19.7133
Epoch 4/15
235/235 [==============================] - 86s 366ms/step - loss: 19.6370 - val_loss: 19.2615
Epoch 5/15
235/235 [==============================] - 86s 366ms/step - loss: 19.0341 - val_loss: 18.2470
Epoch 6/15
235/235 [==============================] - 86s 366ms/step - loss: 17.8713 - val_loss: 17.0223
Epoch 7/15
235/235 [==============================] - 87s 369ms/step - loss: 16.0723 - val_loss: 15.2691
Epoch 8/15
235/235 [==============================] - 87s 369ms/step - loss: 13.8572 - val_loss: 12.6894
Epoch 9/15
235/235 [==============================] - 87s 369ms/step - loss: 11.5040 - val_loss: 9.1899
Epoch 10/15
235/235 [==============================] - 87s 370ms/step - loss: 8.9940 - val_loss: 8.0182
Epoch 11/15
235/235 [==============================] - 87s 370ms/step - loss: 7.1354 - val_loss: 8.4384
Epoch 12/15
235/235 [==============================] - 87s 370ms/step - loss: 5.8725 - val_loss: 5.5627
Epoch 13/15
235/235 [==============================] - 87s 371ms/step - loss: 5.2044 - val_loss: 5.1504
Epoch 14/15
235/235 [==============================] - 87s 368ms/step - loss: 4.6044 - val_loss: 4.3415
Epoch 15/15
235/235 [==============================] - 87s 369ms/step - loss: 4.2699 - val_loss: 3.7286
<keras.callbacks.History at 0x7f004051c490>
```

5.Conclusion

Our project has produced good results for image containing handwritten text in different styles, different size and alignment with varying background. Also we have tried to increase the efficiency of the English character recognition and conversion to its digital form using our trained neural network model. The method of training neural network with extracted features from sample images of each character has detection accuracy to greater extent. In spite of the great number of algorithms that have been developed for character recognition and conversion, the problem is not yet solved satisfactory, especially not in the cases when there are no strict limitations on the handwriting or quality of print. Up to now, no recognition algorithm may compete with man in quality.
Users can have a varying handwritten text that can be of different sizes, styles, fonts etc. sometimes it may be possible that their alignment can be different with a varying background. Our project focuses on producing the best results from all by providing a better image to users. Also we have tried to increase the efficiency of the English character recognition and conversion to its digital form using our trained neural network model. There is a great extent in the detection accuracy for methods of training the neural network having features that are

extracted from the images of a particular character. A lot of algorithms have been created for character recognition and their conversation so far but still this problem is not solved yet. If we talk about a specific case where there is no strict action on the users writing, many system fails in this, thus no algorithm can beat humans when it comes to the efficiency.

Future Scope:

1. We can develop an android application which will directly use the camera to take picture of documents instead of taking the images from other sources.

2. This can be further developed into a system which will recognize devnagri scripts.

Bibliography and References

[1] L. Spitz 'Shape-based word recognition' in International Journal on Document Analysis and recognition October 1998, pages 178-190Optical Character Recognition by Line Eikvil,Issue December 1993

[2] Steps Involved in Text Recognition and Recent Research in OCR ;A Study K.Karthick, K.B.Ravindrakumar, R.Francis, S.Ilankannan - International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019

[3] Srikanta Pal Nallapareddy Priyanka and Ranju Manda. Article:line and word segmentation approach for printed documents. IJCA,Special Issue on RTIPPR, pages 30–36, 2010.