CHARACTER SEGMENTATION MACHINE-TYPED DOCUMENTS USING KNN ALGORITHM

Harjyot Singh Bagga[1], Vasudha Grover[2], Saumya Garg[3], Dr. Anisha M Lal[4]

[1]*Student, School of Computer Science and Engineering, VIT Vellore, Tamil Nadu, India*
[2]*Student, School of Computer Science and Engineering, VIT Vellore, Tamil Nadu, India*
[3]*Student, School of Computer Science and Engineering, VIT Vellore, Tamil Nadu, India*
[4]*Prof, Department of Information Security, VIT Vellore, Tamil Nadu, India*

Abstract - *In this paper, we have presented a very methodical proposition to segment characters of English Language from scanned documents of any form. This approach not only caters to scanned documents but also all kinds of machine-typed documents like the ones written on early typewriters. The suggested character segmentation strategy utilizes the altered projection profiles method that is dependent on utilizing the sliding window technique to basically get all the information you require about the image documented. The suggested approach for the character segmentation is not entirely automatic and for the segmentation procedure, threshold values are utilized. The preprocessing of the documented images follows a super rapid architecture for all the image transformations like the angle of the image is very important in order to do an efficient character recognition, hence we used skew correctness for this purpose. Threshold values are used so as to manage the process of word, line and character segmentation. This approach is very efficient for the major reason being that it can process large documents as well which is expected in real time scenarios.*

Key Words: **K-nearest neighbours, segmentation, character recognition, digit recognition**

1.INTRODUCTION

Character segmentation is a process of separating and extracting the individual characters from the digital image, either individually or as a part of a bigger mechanism. Diverse approaches for character segmentation have been presented in the past. Related works can be divided into two main categories, machine printed/written and hand written documents. The most frequent problem in character segmentation of handwritten documents are touching characters and the variety of works deal with that. For natural images, some segmentation approaches are based on tensor voting and the usage of the three color bar code. A character segmentation method for license plates based on Gaussian low- pass filter and innovation Laplace-like transform is also being used in the industry currently. The method spoken here utilizes adjusted projection profiles which depends on histogram preparation of recently got centralizations of dark pixels in territories of intrigue. The primary sliding window based technique for getting the centralization of dark pixels is utilized, with its varieties, on all degrees of division. The proposed approach itself is self-loader since it utilizes the edge esteems for fundamental parameters. The significant addition in picking this methodology lies in time multifaceted nature. Since the proposed approach utilizes a one-pass handling of the pixel force esteems for each degree of division, preparing time of the procedure is short of one second in any event, for enormous report pictures. It permits the utilization of this approach progressively. Results utilizing the new strategy are introduced concentrating on the part of division precision and the part of the time multifaceted nature, and are dissected and contrasted and best in class draws near.

2. RELATED WORK

[1]In this paper Vladan and boban talk about the segmentation of the individual characters from the machine typed-documents, which is based on different image processing techniques in spatial domain. The approach for general character segmentation for machine-typed documents uses the modified projection technique which uses the spatial features of a document structure in the segmentation methodology. First , Gray scale conversion of the image was done followed by noise reduction. Then the line segmentation which is the central task of the proposed approach was performed. The next step was word segmentation. Here, focus is on analysis of each detected document line and segmentation of each line into words. This is done through sliding window approach.Proposed technique proved to be accurate and efficient as compared to the other state-of-the-art methods

[2]Tyan and Neubauer's method for segmenting and recognizing license plates which may be implemented using a programmable device readable by a machine,

tangibly storing a program of executable instructions to perform the method steps, includes capturing an image of a license plate and preprocessing the image to prepare the image for recognition and segmentation. The image is forward sliced out into doubtful characters regions of the license plate image. The suspected character regions in the license plate image are recognized and provided with a confidence score based on a probability of a correct match to be judged and compared later. If the suspected characters have the score below a threshold value, backward merging adjacent character regions is conducted. The backward merged character regions in the license plate are recognized, and if the backward merged suspected character regions have a confidence criteria below the threshold value, the backward merging and recognizing stages are repeated.

[3]They used the results of a study on a first sight decision on tree algorithms for cursive script recognition based on the use of histogram as a projection file technique. They used the advantage of the ability to learn of neural networks which helped itself adjust upon the recognition of patterns based on a set of multiple inputs. Cursive character segmentation is a very difficult task due to the unsolved problems like slanted character segmentation, underlined and connected characters. The segmentation processes was based on decision tree construction and on the use of projection profile histograms.Projection profile is a data structure used to store the number of non-background pixels.In the first step segmentation is applied to the info stored in the first level of histogram.Spurious pixels and lines can be removed using horizontal histogram data. Then refining is done on the second level of the decision tree. The most common type of error was observed to be an improper separation of a digit as if there was a connection. The use of projection histograms proved to be able to solve more than 80% cases.

[4]A new algorithm for character segmentation was presented by Zhang and Zhang using Hough transformation and the prior knowledge in horizontal and vertical segmentation. The LPR system has three main steps : license plate locating, character segmentation and character recognition. This paper used a new algorithm to combat the drawbacks of image binarization.This algorithm used the information of intensity and avoided the abruption and conglutination of characters. The algorithm used three steps: preprocessing, horizontal segmentation and vertical segmentation. Horizontal segmentation using Hough transformation is done.to analyze the horizontal projection of a single character and find the horizontal segment lines was an easier task to do. Taking the algorithm further, Vertical segmentation algorithm is done based projection analysis, constrained by prior knowledge. They used a database containing 697 license plate images. It was noted that the algorithm has good performance on character segmentation, and can deal with images with disturbance of noise, plate frame, rivet, rotation and illumination variance.

[5] Rashad, M., & Semary, N. A had written a paper on Isolated Printed Arabic Character Recognition in which they have used KNN and random forest classifiers for the classification process which is the most important step, in order to determine efficient and accurate classifiers. The extracted features from this classifier should be efficient and appropriate. The methodology used for the character images is binarization which is later used to extract statistical features on the basis of characters and also shape. The classifiers have been used to draw a comparison between them and RFT gave better results than KNN with an approximation of 11% more accuracy than KNN recognition rate. The dataset they used had 1160 sample characters of sizes 18, 24, 32 and 64 from 6 different Arabic fonts. The disadvantage here is that they trained and tested the same dataset due to which it isn't certain if they would get the same recognition rate on a new set of characters in different languages. The paper could use a better integrated approach like rough sets-based feature extraction, rule generation and classification and on refining the algorithms may provide with better accuracy of the character recognition

[6]Hazra, T. K., Singh, D. P., & Daga, N aims to recognize handwritten or printed text which could be an efficient method that makes use of a customized dataset of images to train the classifier. This OCR extracts distinct features from the input image for classifying its contents as characters, specifically letters and digits. Input to the system is digital images containing the patterns to be classified. Procedure done here is initially the basic preprocessing of image like thresholding, blurring etc and then image features are extracted from the false contours to differentiate between the unwanted background in the image from the characters needed. The research gap here was that It makes use of each feature.This can cause classification errors, especially because there are only limited features useful for classification. Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results. It has a higher cost of computation since every query instance's distance needs to be computed for all the training samples.

[7] Yuchun Lee in this paper aims to provide an algorithm for handwritten digit recognition that showcases less training time, low memory usage, and low classification time. The backpropagation network used here seemed to be successful in terms of memory usage and lowest time for classification, but when the

input given isn't a digit, it could provide "false positive" as the classification. But this network is deemed to have found taking the longest training time. The Radial-Basis Function has less training time but higher classification time and also seems to expect a lot of memory. The kNN classifier was found to use a huge amount of memory and longer classification time even though it can carry-out the recognition of digits. The digit database contains 30,600 training and 5,060 testing patterns. In the end, RBF and KNN proved to be better suited for such problems.

[8] O. Matei · P.C. Pop · H. Valean proposed a novel procedure to perform OCR used in conditions like electric meters. They have used a combination of two algorithms; that is ANN and kNN as a validation algorithm. What's different in this approach to other approaches is that it's based on pixels rather than angles. The benefit of using this approach is the fact that you get the chance of working on various exposure and light situations, the possibility of utilising heuristics for the recognition of characters. The results show that their approach with a decent number of epochs produced a 99.3% accuracy in the digits recognition.

[9] Michael Reynaldo Phangtriastu in his paper made an evaluation using 2 classifiers Support Vector Machine (SVM) and Artificial Neural Network (ANN). This experiment achieves the highest accuracy of 94.43%. The paper used techniques for extracting the features such as: zoning algorithm, projection profile, Histogram of Oriented Gradi adients (HOG) and combination of those feature extractions. The dataset used was the Chars74K dataset for this experiment. ANN gave higher accuracy compared to SVM, except for Projection and Projection + HOG feature extraction. 500 training cycles were used during the training of ANN models.

[10] They used a fuzzy c-means clustering method as a pre-processing method for the basic region growing segmentation method. Basic difference from other approaches is extension of feature space, which results in better segmentation.Extension of feature space is based on simple idea, that neighbouring pixels have approximately same values of lightness and chroma.first, Convert image into feature space of clustering method.Then they ran the fuzzy c-means method on converted image. Finally,some defuzzification rule or rules were used to classify each pixel to segment. Though it was found that in real images, noise is corrupting the image data or image that usually consisted of textured segments.

[11]Parveen Kumar put forward an attempt to do handwritten character recognition using the feed forward and backpropagation neural system without highlighting extraction utilizing the neural system for transcribed character acknowledgment framework. This methodology shaped the information character with 70x50 pixels. It tends to be taken as a contribution to the preparation framework. 25 features were extracted during training from single characters which were then utilised for SVM model training. The accuracy of 80.96% was achieved.

[12]In the algorithm,first, a pre-processing step that includes image binarization and enhancement takes place. At the second step a top - down segmentation approach was used by them in order to detect the characters,text lines and words.A clustering scheme is then adopted in order to group characters of similar shape.word images are grouped into clusters of similar words by using image matching to find similarity. Their goal was to produce reasonable recognition accuracies which enable performing retrieval of handwritten pages from a user-supplied ASCII query OCR systems. This methodology could either be applied to machine printed or handwritten documents.

[13]A typical License Plate Recognition system consists of two parts:license plate detection (LPD) and license plate character segmentation (LPS). LPS aims at extracting the actual boundary and segmenting the license plate into characters which are then sent to the Optical Character Recognition. To fulfil the purpose, Gabor filters are used as a tool to analyse the texture in an unlimited number of directions and scales.To segment , first , the thresholding algorithm was utilised and a binary image was obtained. Then the morphological dilation operator is applied to the binary image. Finally plate regions are extracted by applying eight connected blob coloring.Though several factors like weather conditions and lighting had a negative influence on the results of the LPR system.

[14] Ms.M.Shalini , Dr.B.Indiramany in their research paper aim at researching not for english language characters but for Indian languages since the research in this field is very limited and constrained. India being a multilingual country has so many languages so for character recognition of all these languages needs to be done. Since this area isn't very explored, there's a lot of scope to further do character and handwritten digit recognition for various indian languages. Hence, there is a need to build a model for this and achieve a higher accuracy rate in this field.

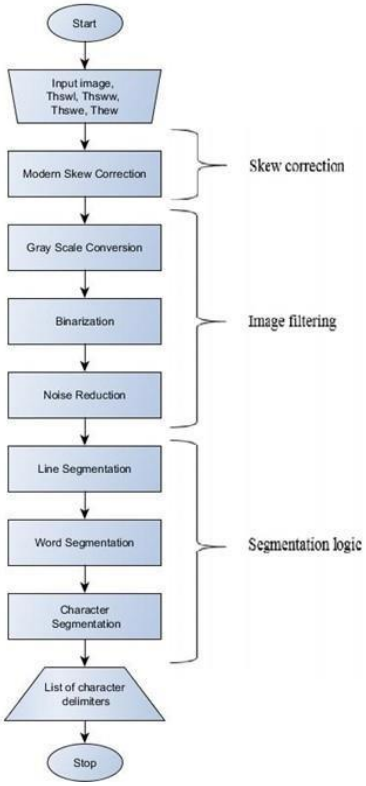
3. DATASETS AND METHODOLOGY

The algorithm consists of three main stages:

1. Manual skew correction, using a new general, ultra-fast architecture for geometrical image transformations.
2. Image filtering, which represents a pre-processing stage and uses literature algo- rithms for image binarization, and noise reduction.

3. Segmentation logic, which uses histogram processing in the process of determining character positions in the given document image.

The inputs are a document image and threshold values which are used in order to control the segmentation process, since the proposed character segmentation approach is semi- automatic.



In order to develop an Object Character Recognition, the image was preprocessed and a sliding window was created to scan through the document for the testing data. For training the characters of the document, a small dataset of 26 English alphabet letters and 10 digits were taken and preprocessed and trained to the KNN model.

3.1 Image processing

For the purpose of preprocessing the characters, the characters were converted into grayscale, so that a good chunk of information could be extracted from the image and the features could be easily recognizable. Dealing with the colored image may lead us to the problem of identifying fake contours due to numerous reasons like shade or an abrupt change in the pixel intensity. Then the image was blurred in order to bring smoothness in the image and ignore or remove any unwanted outlier if present in the image. It also helped to reduce the unwanted noise from the image again making its features more prominent. Using the Gaussian Blur filter here. Post blurring the image, the image was thresholded to highlight any prominent feature or edge that could be used to identify the character while training and differentiate it from among the rest of the characters.

3.2 Grayscale Conversion

First task is image filtering and a pre-task for Binarization. Grayscale conversion is performed on the original image, f(x,y) to obtain a grayscale image, g(x, y) with 256 Gray color levels.

0: Complete Black

255: Complete White

'f 'is the starting 24-bit image of size M ×N, and that 'g' is the 8-bit image of the same size.

For all values of x and y, g(x, y) is given as:

g(x, y) = max { f(x, y)red, f(x, y)green, f(x, y)blue }

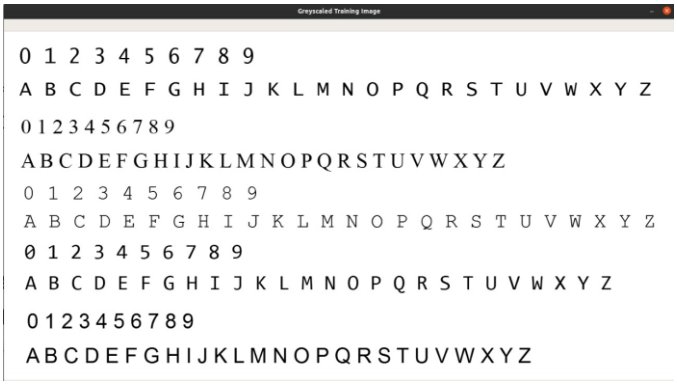


Chart-2 grayscale image training

3.3 Binarization

Next step is binarization or in other terms thresholding. Image Binarization is performed on grayscale images using a threshold function, which is also known as gray-level transformation function. Input is an 8-bit grayscale image and output is a binary image in which

0 : White Pixels(Foreground)

1 : Black Pixels(Background)

If Threshold value is equal to t. Each pixel in image g will take the intensity value:

g(x, y) = 1 , 0 ≤f(x, y)≤t or g(x,y) = 0 , t <f(x, y)≤255

The threshold value that gives the best results depends primarily on the quality of the document image. It is mostly constant.

Chart-3 Threshold image after grayscaling

3.4 Noise Reduction

As the binarization is a thresholding based process, there is a possibility of noise being introduced in the image.
Noise reduction is the last step in the filtering process in which a simple noise reduction algorithm eliminates isolated black pixels which do not have other black pixels as 8- neighbors. The image is padded with 0's on all borders. w is a filter mask of odd size. Each pixel g (x, y) will be given as:

g(x, y) = f(x, y)
If Σ(s=-a)^(s=a) Σ(t=-b)^(t=b) , "w(s, t) f(x+s, y+t) " > 0

g(x, y) = 0
If Σ(s=-a)^(s=a) Σ(t=-b)^(t=b), "w(s, t) f(x+s, y+t)" = 0

If a pixel has all of its 8-neighbours as white pixels(0-value) then a given pixel will be set as 0 irrespective of its value. If a pixel has even one of its 8-neighbours as black pixel (1-value) then the given pixel will remain unchanged.

3.5 Feature Detection

Once the basic image was preprocessed, we tried to abstract the important information from the image like edge or various other contours. For that, we applied edge detection on the training, to detect and recognize all the edges of the image specifically the character set.

3.6 Introducing the Sliding-Window

Now according to the font-size of the image, a sliding window was chosen. The sliding window is basically a rectangular box-like structure that can completely fit each character and can incorporate all its information with minimum noise or interference of any other character. Once we have decided on the size of the sliding-window, we can go over and extract the characters based on the information of their contours. All of this is part of segmentation.

3.7 Segmentation Logic

This is the third stage for a proposed approach for detecting text in image. This part is based on histogram processing. The histogram is computed using the values obtained after performing the projection profiles technique using the sliding window for calculation of the concentration of black pixels. The windows determine the whitespaces in between characters, words and lines by calculating image areas with low concentration. The window has two types of motions:

1. Horizontal motion
- For words: Words segmentation
- For characters: Characters segmentation
2. Vertical motion
- For lines: Lines segmentation

The sliding window moves pixel by pixel in either of two above mentioned motions in one iteration.

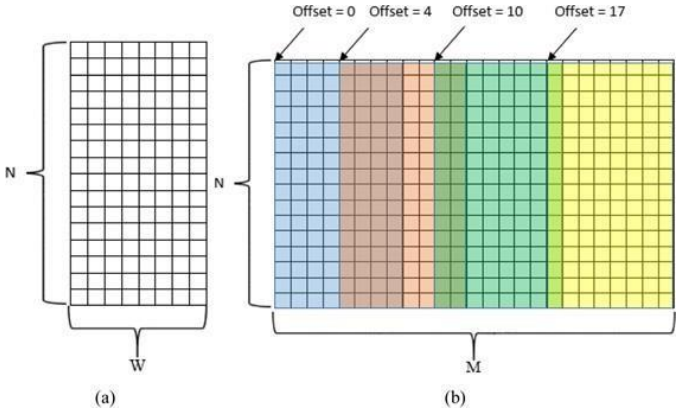


Chart-4 Segmentation

The image above shown as:

a) Sliding window without offsets

b) Sliding window with different offsets

3.8 Line Segmentation

Vertical sliding window is used to segment the image

Equation 1

$$d_x = \begin{cases} x + \left[\frac{H}{2}\right], & s_n < T_{hswl} \\ -1, & \text{otherwise} \end{cases}$$

Equation 2

$$s_n = \sum_{s=0}^{H-1} \sum_{y=0}^{N-1} f(x+s,\ y)$$

Let N X H be size of image as well as window In the equations 1 and 2:

- S_n = represent the values in the array of all sliding window concentrations of black pixels of S
- Thswl -> Threshold
- dx -> offset relative to top of image

Only the offsets which belong to the sliding window with concentration of black pixels lower than the threshold value Thswl are taken into consideration. Sometimes a line segmentation can cut document characters due to a low concentration of black pixels.

Equation 3

$$U = U - \left[\frac{U-L}{2}\right]$$

$$L = L + \left[\frac{U-L}{2}\right] + (U-L) \bmod 2 - 1$$

So, the equation 3 has its correction:

- U represents the offset of the upper border of the given line.
- L represents the offset of the lower border of the previous line.

3.8 Word Segmentation

Horizontal sliding window is used for this type of segmentation. It detects the punctuation characters, some of which may be even recognized by the concentration of black pixels in specific areas of the line. Let W X H be the size of the window where

- W -> width of window
- H -> height of window
- hn is number of black pixels in the window

$$h_n = \sum_{x=H1}^{H2} \sum_{t=0}^{W-1} f(x,\ y+t)$$

$$V = \{h_i | h_{i-1} > h_i \wedge h_{i+1} > h_i\}$$

Successive repeating values are merged into one value at the position of the rightmost value in that group,the histogram is computed for each document image line. Histogram is constructed and the minima of the graph is calculated. Obtained delimiters which are located between the words are moved to the left and to the right to the closest characters.

3.9 Character Segmentation

Horizontal sliding window is used to do this segmentation. Equations given beside are used to determine word length after the computation of local minimal of histogram, where

- Cn -> assumed word length
- Cwavg -> average word length
- Thcw -> threshold value of characters space
- We -> Word width in pixels

$$C_n = \left[\frac{W_w}{T_{hcw}}\right] \qquad d_i = h_i + \left[\frac{W}{2}\right]$$

$$C_{wavg} = \left[\frac{W_w}{C_n}\right]$$

$$j = \underset{i}{argmin}\left(\left|d_i - d_{ref}\right|\right)$$

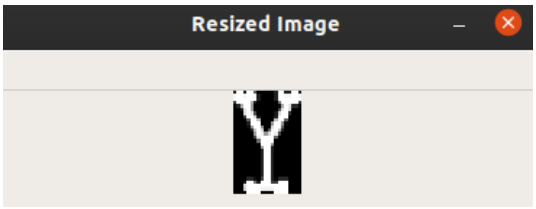


Chart-5 Resized image

3.10 Flattening of the Images

Once the images are preprocessed, they are flattened and converted into a vector so that they can be easily fed to the model for further training and testing.

3.11 Labeling

Once the individual characters from the character set have been extracted, we can start labeling the characters on as to what the characters represent, so that the features of the character can be correspondingly mapped to the proper character.

Chart-6 Labelling

4. Training

Once the characters are preprocessed and labeled, we can feed it to the model where the model can categorize the characters according to their features appropriately.

4.1 Choice of Algorithm

KNN (K-nearest neighbor clustering)
The tests were made on 2 widely used algorithms i.e. SVM and KNN, and the results were recorded. It was found out that on testing for images having a small amount of noise, the SVM algorithm failed to identify among close characters like 'S' and '8'. It was concluded that further steps of preprocessing were required in order to make a working model on SVM that gives a workable accuracy. Whereas, KNN, on the other hand, gave a much better accuracy for images that had some amount of noise and distortion. Hence, KNN was taken ahead to build the OCR.

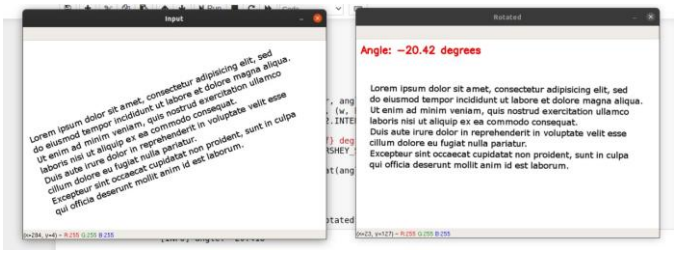
5. Testing

Before applying the basic preprocessing steps required for testing, an additional skewing was taken into consideration. As observed, images scanned may have some amount of skewness in them that may decrease the accuracy of the model significantly. Skewing the images and the text, helped boost the accuracy of our model significantly on the testing dataset.
Image Transformation using mapping offsets which represent the transformation matrix for chosen transformation. Using the mapping offsets each input position is mapped to the specific output position. Mapping Offsets for a transformation with particular values is performed at the start and at the time of transformation, offsets are used. Mapping Offsets are calculated using standard rotation transformation pair:

$$Sx = x\cos(\theta) + y\sin(\theta)$$
$$Sy = -x\sin(\theta) + y\cos(\theta)$$

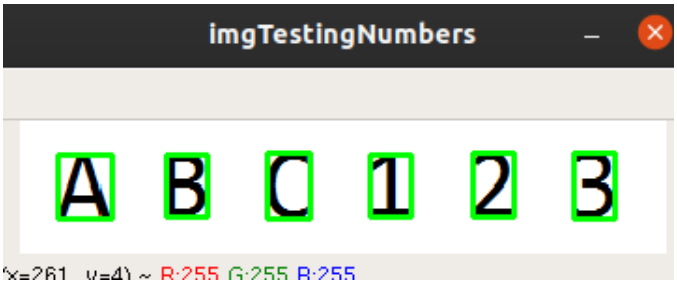
We must correct the text skew in an image containing a rotated block of text at an unknown angle by:

1. Detecting a text block in a picture.
2. Calculating the rotational text's angle.
3. To compensate for the skew, rotate the image.



■ **Chart-7** On the left is the skewed image and the right one is De-skewed image.

After skewing, the images were preprocessed similar to the training data, the images were flattened into a vector and then were fed to the model. Here, with the help of the information of the contours, a sliding window was placed on the borders of the character and was predicted with the help of the training model.



■ **Chart-8** output

6. Performance Evaluation

While building the OCR, we experimented over multiple models and found the best results using SVM and KNN. But when the model was trained on distorted and noisy images, KNN came out with better results with 92% accuracy on training data and 82% on the test data. The following is the comparison between the models in terms of their performance.

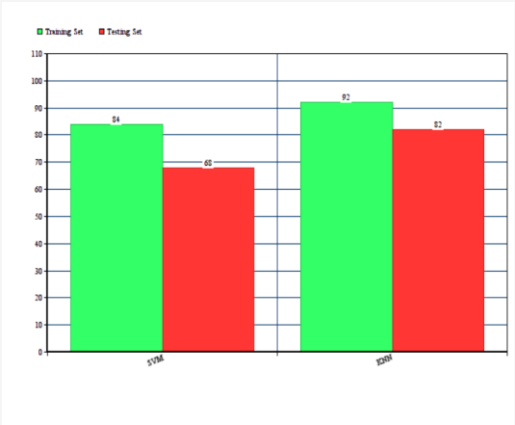


Chart -9 Comparison of model performance

Also a major boost in the performance was a result of incorporating the process of skewness before training and image processing. Removing the skewness of the image and text lead to an approximate 25% enhancement of the models performance.

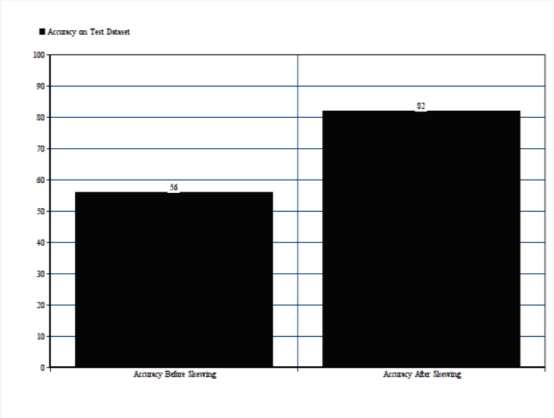


Chart -10 Performance results before and after skewing

7. Experimental Results

The OCR was successfully able to extract characters and recognize it successfully, with an average test accuracy of about 82%. The main motive of training the data over a very small sample and effectively extracting the text of documents that maintain a standard of font and writing throughout was successful. However, there is quite a scope of improvement and enhancement in its application. This whole process is currently trained over characters of the English Alphabet but can be expanded over various other languages by increasing the training data, as contour-based image segmentation techniques have been used here. Also, more efficient methods can be bought into practice to decide the sliding window size. All the training and testing have been done on various journal papers that have used Fonts 'Arial' and 'Times New Roman' majorly in font size 8 to 14, and all the results are with respect to those fonts only.

8. Application

This technique further can be used to implement various other scenarios in real life. This OCR can be used in various tools like PDF to Word Conversions and text extraction tools from images. The whole tool can be used to conduct text-based operations where the text isn't directly available like images, pdfs, and epubs.

9. CONCLUSIONS

In this project, an efficient approach for character segmentation of old machine-typed documents is presented. Although the main targets of the technique are machine-typed documents, the proposed approach can also be used for machine-printed documents. The method is semi-automatic, since threshold values are necessary to define some important parameters. This approach uses the image processing methods in the spatial domain and exploits the features of the document structure in the process of character segmentation. Three main stages of the proposed approach are manual skew correction, image filtering, and segmentation logic, which represents the core stage of the algorithm.

10. REFERENCES

[1] Vučković, V., & Arizanović, B. General Character Segmentation Approach for Machine-Typed Documents.

[2] Tyan, J. K., & Neubauer, C. (2002). U.S. Patent No. 6,473,517. Washington, DC: U.S. Patent and Trademark Office.

[3] Rodrigues, R. J., & Thomé, A. C. G. (2000, August). Cursive character recognition-a character segmentation method using projection profile-based technique. In The 4th World Multiconference on Systemics, Cybernetics and Informatics SCI 2000 and The 6th International Conference on Information Systems, Analysis and Synthesis ISAS.

[4] Zhang, Y., & Zhang, C. (2003, June). A new algorithm for character segmentation of license plate. In IEEE IV2003 Intelligent Vehicles Symposium. Proceedings (Cat. No. 03TH8683) (pp. 106-109). IEEE.

[5] Rashad, M., & Semary, N. A. (2014, November). Isolated printed Arabic character recognition using KNN and random forest tree classifiers. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 11-17). Springer, Cham.

[6] Hazra, T. K., Singh, D. P., & Daga, N. (2017, August). Optical character recognition using KNN on custom image dataset. In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 110-114). IEEE.

[7] Lee, Y. (1991). Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. Neural computation, 3(3), 440-449.

[8] Matei, O., Pop, P. C., & Vălean, H. (2013). Optical character recognition in real environments using neural networks and k-nearest neighbor. Applied intelligence, 39(4), 739-748.

[9] Phangtriastu, M. R., Harefa, J., & Tanoto, D. F. (2017). Comparison between neural network and support vector machine in optical character recognition. Procedia computer science, 116, 351-357.

[10] Horvath, J. (2006, January). Image segmentation using fuzzy c-means. In Symposium on Applied Machine Intelligence.

[11] Parveen Kumar, Nitin Sharma and Arun Rana, "Handwritten Character Recognition using Different Kernel based SVM Classifier and MLP Neural Network", International Journal of Computer Science, Vol. 53, NO.I I, Sep.2012

[12] Vamvakas, G., Gatos, B., Stamatopoulos, N., & Perantonis, S. J. (2008, September). A complete optical character recognition methodology for historical documents. In 2008 The Eighth IAPR International Workshop on Document Analysis Systems (pp. 525-532). IEEE.

[13] Kahraman, F., Kurt, B., & Gökmen, M. (2003, November). License plate character segmentation based on the gabor transform and vector quantization. In International Symposium on Computer and Information Sciences (pp. 381-388). Springer, Berlin, Heidelberg.

[14] Shalini, M. M., & Indira, B. (2014). Automatic Character Recognition of Indian Languages–A brief Survey. International Journal of Innovative Science, Engineering & Technology, 1(2), 131-138.