HBRP PUBLICATION

Advancement in Image Processing and Pattern Recognition
Volume 3 Issue 3

Improving the Recognition Accuracy of Tesseract-OCR Engine on Nepali Text Images via Preprocessing

***Umesh Hengaju*[1], *Dr Bal Krishna Bal*[2]**
[1,2]*Information and Language Processing Research Lab, Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Kavre, Nepal.*

***Corresponding Author**
***E-mail Id:-**hengajumesh@gmail.com*

ABSTRACT

Image Documents scanned or captured by digital cameras on mobile phones suffer from a number of limitations like geometric distortions, focus loss, uneven lightning conditions, low scanning resolution etc. Because of these limitations, the quality of image documents is often degraded and because of this, the recognition accuracy of OCR engines gets affected. This work focuses on improving the recognition of Tesseract-OCR engine for Nepali image documents via preprocessing. For this purpose, we developed an image preprocessing pipeline consisting of 8 steps and tested with several Nepali text images which were collected from different sources like Nepali news corpus, books, printed documents etc. Our test results showed that the recognition accuracy improved from 90.69%, 54.34% and 38.45 to 94.84%, 71.15% and 51.21% respectively for high, medium and low quality images.

***Keywords:-** Optical Character Recognition (OCR), improve recognition accuracy, tesseract ocr engine, nepali text image, image preprocessing*

INTRODUCTION

Optical Character Recognition (OCR)

OCR is one of the successful applications of computer technology in today's era. The OCR technology extracts the characters from the document image and converts them into machine recognizable form [1]. Furthermore, the technology has a wide number of applications in different day-to-day practical areas like electronic editing, searching, indexing, publication etc. It has become a common method of digitizing printed text documents and is being widely used in different tasks such as number plate recognition, cognitive computing, machine translation, text-to-speech conversion, text mining etc. [3]

OCR involves scanning or capturing image document, preprocessing, segmentation, feature extraction, classification and recognition [6]. Each of these steps passes its results to the next phase in a pipeline fashion. In the preprocessing phase, photographed document image is preprocessed via different image preprocessing techniques. Further sub-processes within preprocessing include Noise reduction, Gray Scale Conversion, Skew correction and Binarization [15]. In the segmentation phase, the preprocessed image is segmented using various segmentation approaches such as Histogram approach, Connected Component method etc. for easy extraction of characters, words, sentences or paragraphs present on the image document. Then, the necessary features such as counter line are selected using different classifiers like Neural Networks, Support Vector Machine, Random Forest, Naïve Bayes etc. so that the detected lines, characters, words can be classified. This yields a feature vector which is then inputted into the recognizer. Finally, the recognizer recognizes the characters in the text image and converts them into readable and editable text files [6]. Figure 1 shows an overview of the general OCR system.

 All Rights Reserved

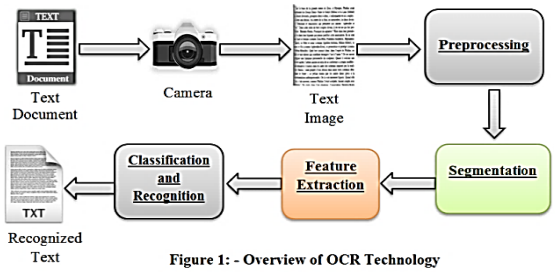

Figure 1: - Overview of OCR Technology

Fig.1:-*Overview of OCR Technology*

The research in the area of OCR technology has been going on for over half a century and the accuracy of these tools in terms of recognition has always been a topic of research. The recognition accuracy of OCR depends upon various factors and quality of an input image is one of them. The better the quality of input text image; the better the recognition accuracy of OCR engines. There is an increasing trend to use digital cameras available on mobile phones for capturing or scanning image documents. Unfortunately, these images suffer from a number of limitations like geometrical distortions, focus loss, uneven lightning conditions, low scanning resolution etc. As a result, the quality of document image is not of the expected quality leading to lower recognition accuracies of the OCR engines [11].

By enhancing the quality of image document, the text detection rate of OCR engines can be improved[4]. Based on this assumption, a pipeline of preprocessing activities like Illumination Adjustment, Orientation Correction, DPI Adjustment, Gray Scale Conversion, Binarization, Noise Reduction, Un-Sharp Masking and Normalization has been developed and its effect on the recognition accuracy of the OCR engine for Nepali has been analyzed in this work. The implemented preprocessing pipeline is applied to the image documents prior to passing the pre-processed images to the Tesseract-OCR for Nepali.

FEATURES OF NEPALI LANGUAGE

Nepali language is written in the Devanagari script. It has a set of basic alphabets which includes 11 vowels, 33 consonants, vowel modifiers, half-consonants, consonants modifiers, special diacritic marks & digits and has its own specific rule for combining these symbols to form word and sentences [14]. The basic symbols used in Nepali language text are provided in table 1 to 5.

Table 1:-*Vowels and Corresponding Modifiers*

Vowel	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ	अं	अः
Corresponding Vowel Modifier		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ	ं	ः

Table 2:-*Constants*

क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ
ब	भ	म	य	र	ल	व	श	ष	स	ह

Table 3:-*Half Constants*

क्	ख्	ग्	घ्		च्		ज्	झ्	ञ्	
			ण्	त्	थ्		ध्	न्	प्	फ्
ब्	भ्	म्	य्		ल्	व्	श्	ष्	स्	

Table 4:-*Diacritics, Consonant-modifiers and Special Symbols*

Diacritics and Special Symbols					Different forms of Consonant modifier ra (र्)
ँ	़	ॅ	०	ऽ	◌ / ^ ⌒

Table 5:-*Numerals*

०	१	२	३	४	५	६	७	८	९

In the Nepali language, more complex characters can be formed by combining these basic symbols [14]. Vowels can be used as a modifier above, below, before or after the consonant they belong to and characters formed by such methods are called conjuncts [2]. Also, by combining consonants and half-consonants, characters can be formed and characters formed in such a way are called compound characters [1]. In some situations, a consonant following (or proceeding) another consonant is represented by a modifier called consonant modifier.

Table 6:-*Character Formation in Nepali Text*

Conjuncts	Compound Characters	Consonant Modifier
क + ा → का	व + य → क्य	र् + म → म्र
क + ि → कि	न + म → न्म	र् + य → र्य

In Nepali texts, letter variations are also found in the writings. This is because fonts have different writing styles. Some characters having letter variants differ by the old and new writing styles. The old variants of some letters are not used these days but the old documents frequently contain these forms [14].

Table 7:-*Letter Variations*

Letter	Variants	Letter	Variants
Numeral Five	५ ५	Letter 'La'	ल ल
Letter 'A'	अ ॲ	Letter 'Sha'	श श
Letter 'Jha'	झ झ ऱ	Letter 'Ksha'	क्ष क्ष

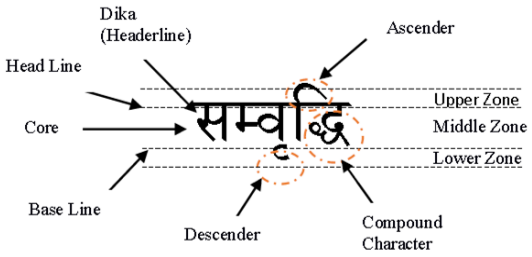

Fig.2:- *Structure of Nepali Text Word*

In the Nepali language, characters are written from left to right in a horizontal line and all the characters are placed under a horizontal line called "Dika" or "Shirorekha" or "Maatra" which runs at the top of core characters. Each word in the language spans in three zones: ascender zone, core zone and descender zone. The zone above the dika is ascender, zone just below dika is core and zone below the baseline of core zone is descender [14].

The rest of the paper is organized as follows: Section 2 states the problem statements of the work, Section 3 presents some related works. Section 4 states the objectives, Section 5 describes the methodology, Section 6 provides the results and finally Section 7 presents conclusion to this work.

PROBLEM STATEMENT

Digital camera and mobile document image acquisition process integrates complications in the image document. Some of those complications are explained below:

Complex Background

In a regular environment while capturing images of text documents, unlike in a

scanned document, many man-made objects such as paintings, buildings, and symbols are included in the image during the image acquisition process. Presence of these objects in the text image affects the recognition process [9].

Uneven Lightening Condition
The images captured in natural environments often comprise non-uniformly illuminated sub-region or shadows. This poses a challenge to OCR as it degrades the desired characteristics of the image and hence causes less accurate detection of text. Although using an on-camera flash may eliminate such problems with uneven lighting, it introduces new challenges [7].

Focus Loss
While capturing images of text documents at large apertures and short distances, uneven focus can be observed in the photographed image. Also, movement of the item at the time of image capturing may cause capturing of blurred images. These things affect the recognition efficiency of OCR engines. For the best accuracy of character recognition and character segmentation, character sharpness is required [4].

Incorrect Orientation
Since, the camera viewpoint of the text image is not the same as the viewpoint of the scanner, the photographed text image gets distorted. The shape of the photographed image does not have a constant parallel plane. Instead, they get skewed or may get a trapezoid shape such that text lines at distant seem less and text line near the camera seems greater. And because of this OCR exhibits lower recognition rate [11].

Noise
Noise in an image may appear due to dirt or random variation of brightness or color information. Due to the noise, the text in the document images becomes unclear which in turn affects the recognition efficiency of OCR engines [16].

Annotations
If the image of text documents includes notes and drawings by users, library stamps, watermarks or punch holes then text from them is very unlikely to be read correctly. As a result, recognition efficiency of OCR engines may get affected [12].

Yellowed Paper
Background color may change across the document due to age of material, causing fuzziness at edges of characters. The combination of humidity and age of material may change the background color of the document into yellow color. This may have a minor effect on OCR accuracy [12].

Show through of Ink
In documents printed on very thin paper, the ink from one side of the page will often show through to the other. This can have severe negative effects, comparable in all ways to bleed through [12].

RELATED WORKS
Several approaches have been proposed to enhance OCR accuracy and text detection. In [4], Wojciech Bieniecki et al. proposed a model for improving the recognition accuracy of OCR. The work showed that OCR recognition is most sensitive to geometric deformations. So it is more focused on correcting the orientation of text areas in the text images. For correcting orientation, the major steps considered in the work are Rotation, Perspective Correction and Non-Linear Transformation. With each successive step, the test result shows improvement in recognition accuracy keeping error rate less than 1%.

In [7], Badla investigated the principles of optical character recognition used in the Tesseract OCR engine and techniques to improve its efficiency and runtime. It has applied a preprocessing technique to the Tesseract OCR engine to improve the recognition of the characters keeping the runtime low. The work reports accuracy of 90.5% for recognizing text belonging to Hindi Language. But, the limitation of the work is that the accuracy of the Tesseract OCR engine decreases with the increase in average runtime of the system.

In [8], Gupta et al. worked for improving recognition accuracy of OCR using preprocessing techniques. The work reports binarization being done using morphological operator, border noise removal using connected component analysis, line segmentation using horizontal projection profile and word, character segmentation using active contour model. The experimental results show an accuracy of 80% which is significant considering the obscurities involved in the process of word and character segmentation. However, the problem in the work is, it could not deliver consistent performance on noisy documents containing bad characters like complement color patches (speckles), border artifacts, etc.

In [11], Harraj et al. presented a novel nonparametric and unsupervised approach of preprocessing to compensate for undesirable document image distortions aiming to optimally improve OCR accuracy. The approach presented in this work relies on a very efficient stack of document image enhancing techniques to recover deformation of the entire document image. The proposed approach significantly improves text detection rate and optical character recognition accuracy.

In [14], Pant et al. proposed a hybrid approach of segmentation using RF algorithm. It's main objective was to minimize the segmentation errors and to increase the accuracy of OCR in recognizing Devanagari script. In this, two approaches: Holistic approach and Character Level Recognition (CLR) approach has been incorporated. And, the recognition rate with character level recognition approach and the hybrid approach were 78.87% and 94.80% respectively. However, the work basically deals with over-segmentation and under-segmentation problems.

In [15], various image preprocessing steps have been identified for improving the recognition efficiency of Tesseract OCR engine. In the work, Luminous Gray Scaling, Deskewing, Linearization and Pixilation steps have been put together to form an image preprocessing pipeline. The experimental result shows significant improvement on the recognition accuracy of the OCR engine. But, the limitation of the work is that it only focuses on performing Multiframe Super resolution to produce a high resolution image and is computationally slow.

In this particular work, we focus on improving OCR accuracy by pre-processing the input documents images. Contrary to the method proposed in [7], we propose to use a slightly more enhanced non parametric approach for improving the quality of the input document, thus using a tested combination of pre-processing techniques. This gives us the possibility to improve OCR accuracy independently for recognizing Nepali text from Nepali text document images.

OBJECTIVES

We set the objectives of this work as follows:

1. To investigate the quality issues with camera captured image documents for Nepali texts.
2. To address those issues using image preprocessing techniques so that the

text recognition rate of Nepali OCR improves.

METHODOLOGY

This section discusses the methodology adopted for the given work. Essentially, we analyze the effect of a preprocessing pipeline in terms of recognition accuracies of the Tesseract OCR for Nepali image documents.

Developing Pipeline of Image Preprocessing Activities

An image preprocessing pipeline is developed as a conceptual framework of this work. The developed preprocessing pipeline is composed of eight steps. Figure 3 illustrates the pictorial representation of the preprocessing pipeline developed in the work.

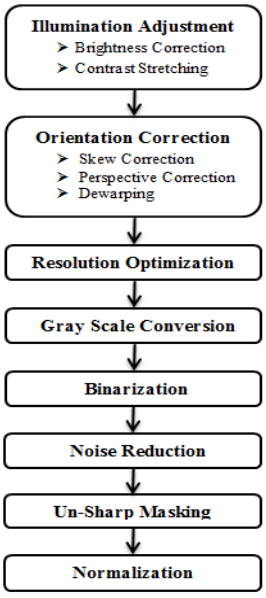

Fig.3:-*Image Preprocessing Pipeline*

Illumination Adjustment

The first step in the preprocessing pipeline involves correcting the brightness and stretching the contrast of the input text image. For that raison, we use Histogram Equalization Method and Recursive Adaptive Gamma Correction method as a powerful brightness and contrast enhancement approaches.

The algorithm for correcting brightness of an image using Histogram Equalization method involves following steps [18].

Step 1: The algorithm for correcting brightness of an image using Histogram Equalization method involves following steps [18]. Convert the image into gray scale form.

Step 2: Obtain Histogram for the gray image.
Gray histogram of an image is a one-dimensional discrete function, which can be represented as Equation 1.

$$h(k)= n_k \quad \ldots\ldots\ldots\ldots(1)$$

Here, n_k is the number of pixels with the gray value of k in image f (i, j).

Step 3: Compute the probability density function (PDF) using Equation 2.

$$p_s(s_k)= {}^{n_k}/_{n},$$

$0 \leq s_k \leq 1$, k = 1, 2, 3, ..., n-1……..…….. (2)

Here, s_k(k = $1, 2, 3, \ldots\ldots\ldots, n-1$) denotes the k-th gray-level of f(i, j) and n is the total number of pixels in the image.

Step 4: Now compute cumulative distribution function (CDF) using Equation 3:

$$t_k = E_h(s_k) = \sum_{i=0}^{k} {}^{n_k}/_{n} = \sum_{i=0}^{k} p_s(s_i) \quad \ldots(3)$$

According to Equation 3, the pixels with the intensity of s_k in the input image are mapped to the corresponding pixels with the intensity of t_k in output image. Theoretically, the PDF of the image obtained using Equation 2 produces histogram equalized image but it may not be uniform because of the discrete nature of the pixel intensities. Thus, to obtain uniformly histogram equalized image, CDF can be computed using Equation 3 [18].

For stretching contrast, the work makes use of Recursive Adaptive Gamma Correction method. The algorithm of recursive adaptive gamma correction method is taken from [19]

Orientation Correction

The camera captured images of printed documents or an open book page usually suffers from artifacts like skewness. Such artifacts reduce the quality of the images and cause many problems to the process of document image analysis. It is difficult to understand such documents by the Optical Character Recognizer (OCR). This step rectifies such problems. It involves detecting skew and correcting them as well as correcting perspectives, and dewarping of the image.

Hough lines transform method is used for detecting skew of text lines in document images and Rotation transformation equation is used to correct this skew. Interpolation method proposed by [4] is used for correcting perspectives of the images and for dewarping, coordinate transform model and document rectification method proposed by [20] is used.

Resolution Optimization

In this, the DPI of the image is adjusted. If the DPI of a text image is lower than 200 then it will give unclear and incomprehensible results while keeping the DPI above 600 will unnecessarily increase the size of the output file without improving the quality of the file. For OCR, if the text size is 8-10 points, then it is recommended to use 300 dpi for better recognition accuracy [4]. So this step involves optimizing the resolution of document images to 300DPI.

Gray Scale Conversion

Gray scale conversion is one of the simplest image enhancement techniques. The main reason why gray scale representations are often used for pattern recognition is that instead of operating on color image, gray scale images simplifies the algorithm and reduces computational requirements. Many algorithms have been proposed for gray scale conversion. It has been proven that not all color-to-gray scale algorithms work equally well [9], also it has been shown that the Luminance algorithm perform better than other variations for texture based image processing [9]. In our case we use Luminance algorithm which is designed to match human brightness perception by using a weighted combination of the RGB channels in component-wise manner.

Gray = (Red * 0.2126 + Green * 0.7152 + Blue * 0.0722)

Binarization

Binarization involves separating image pixel values as background and foreground pixel. It firstly convert the document image into gray scale image and then in to

binary image. To perform such a segmentation of document image, one intensity value called threshold is picked. Then pixel value below the threshold is turned into zero and pixels above that threshold to one [10].

This work used Otsu's Binarization method for binarizing the document images. Otsu's method for binarization is based on formation of two distinct classes of pixels, one which contains background pixel intensities and the other containing foreground pixel intensities and then calculating the variance between them. The calculated variance is then used as a threshold value for image binarization. The Otsu's binarization algorithm consists of the following steps [21]

Step 1: Read a gray scale image.
Step 2: Calculate image histogram.
Step 3: Select a threshold and referred as t,
- Calculate foreground variance.
- Calculate background variance.

Step 4: Calculate Within-Class variance.
Step 5: Repeat steps 3 and 4 for all possible threshold value.
Step 6: Final global threshold, T = threshold in MIN(Within-class variance)
Step 7: Binarized Image = gray scale image > T

Noise Reduction

Generally, the printed document images contain salt and pepper noise, marginal noise or clutter noise [5]. These kinds of noises can be reduced by filtering. Thus, this step involves using median filter for reducing noise in the text images.

In median filtering, the input pixel is replace by the median of the pixel contain in windows. The algorithm for median filter requires arranging the pixel value of window in increasing order. According to [5], if g(x, y) is the input image then the median is given by following Equation 4.

$$R = \text{median}\{g(s, t)\} \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Then, the central pixel intensity value is replaced by median value.

Un-sharp masking

This step involves enhancing the appearance of text details by using un-sharp masking filter. Un-sharp masking filter enhances the textual details in an image by increasing small-scale acutance without creating additional detail. Generally, the noise reduction steps outputs unclear images having blurriness effect. Un-sharp masking is a very powerful method to sharpen images. The basic concept used in un-sharp masking is subtracting the blurred version of the image from the original one [11].

Normalization

In this, an image is mapped onto a standard plane for removing all types of pixel variations. It involves transforming the text image in some way to make it consistent. Moment-based normalization technique derived from Prasad et al., (2014) is used for normalizing the document images.

Developing Tesseract Based Nepali OCR

The Tesseract based Nepali OCR is developed by integrating a preprocess pipeline with Tesseract OCR engine version 4.0. For developing this OCR, python version 2.7.15 is used as a programming tool. The Open Computer Vision (OpenCV) is used as the implementation framework for preprocessing activities and for language support, Leptonica version 1.7 is used. Figure 4 illustrates the overview of the Nepali OCR that has developed in this work.

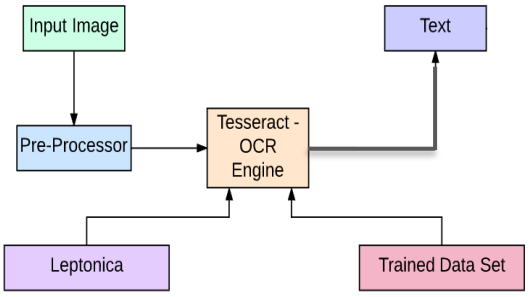

Figure 4: Overview of developed Nepali OCR

Experiment

It first involves comparing the performances of various methods of a preprocessing technique in terms of OCR character recognition accuracy for deciding which one is to be used in the pipeline.

To demonstrate the improvement in the recognition accuracy of OCR with the proposed image preprocessing pipeline, the Tesseract based Nepali OCR is experimented with a dataset of Nepali text images. For the purpose, the dataset of Nepali text images is generated first.

Image Data-set Generation

The Nepali text images used in the experiment are collected from various sources. Nepali text documents like Nepali news corpus, magazines, books, printed documents are collected and their images are captured using digital cameras available on Android Mobile Phone. Also, images of Nepali text documents are collected from different internet sources like Facebook or websites like pinterest etc. Besides this, snapshots of the Nepali printed document in MS-WORD or any sort of PDF files are taken for generating image data set. About 1200 Nepali text images are collected for conducting the test experiment.

Performing Test Experiment

The quality of Nepali text image collected from different sources varies in quality from high to low. Thus to keep the work as much as simple, the collected images are classified into 3 categories, namely, high quality image, average quality image and low quality image.

High Quality Images:

They do not contain noise and their contents are not blurred. The lighting conditions in these images are good. Also, they are not distorted and are not composed of complex backgrounds.

Average Quality Images:

These images are distorted and contain randomized mixture of noise and sharp text. Their background contains non-textual objects.

Low Quality Images:

These images contain black spots in them and there appear black lines in the areas where documents are folded. Also their contents are blurred and they have very complex backgrounds such as presence of shadow regions, yellow colored background etc.

The test experiment involves forming three test cases, namely, Best Test Case, Average Test Case and Worst as for analyzing the effect of preprocessing pipeline in the recognition efficiency of OCR engines.

Best Test Case involves analyzing the effect of image preprocessing pipeline on the recognition efficiency of Tesseract OCR engine with high quality images. The number of Nepali text images used in this test case is 400.

Average Test Case involves analyzing the effect of image preprocessing pipeline on

the recognition efficiency of Tesseract OCR engine with average quality images. The number of Nepali text images used in this test case is 400.

Worst Test Case involves analyzing the effect of image preprocessing pipeline on the recognition efficiency of Tesseract OCR engine with very low quality images. The number of Nepali text images used in this test case is 400.

In each of the test cases, first the original photographed image of the text document is fed into the Tesseract for recognition without carrying any type of preprocessing activities. Then the next thing done is, the photographed text image is passed into the preprocessing pipeline and the preprocessed image is fed into Tesseract for recognition of characters.

Performance Evaluation

For evaluating the performance, first ground truth is acquired by using NeOCR. The output of NeOCR is manually corrected if they contain error and is saved as ground truth. Then, the recognized text with and without preprocessed images are compared with the ground truth for determining the accuracy of recognition. Mathematically, the rate of recognition accuracy is calculated by using the formula given by [17] which is as follows:

Accuracy = $\frac{n-e}{n}$ ……..... (6)

Here, n = total no. of characters in original document and e = total no. of error characters in recognized text.

RESULTS

Table 8 and 9 shows the result of comparative study between various approaches of brightness correction and contrast stretching algorithms

Table 8:-*Recognition Accuracy with different Brightness Correction Methods*

Methods	**Accuracy**
HSV Transformation	60.94%
Histogram Equalization	62.71%
Adaptive Histogram Equalization	62.29%

Table 9:-*Recognition Accuracy with different Contrast Stretching Methods*

Methods	**Accuracy**
Min-Max Contrast Stretch	61.21%
Gamma Correction Method	63.05%
Recursive Adaptive Gamma Correction Method	63.82%

Table 8 shows that among various methods of brightness correction, Histogram Equalization method provides greater recognition accuracy and Table 9 shows Recursive Adaptive Gamma Correction method among others provides greater recognition accuracy for Tesseract OCR engine.

The result of comparison between various skew detection approaches in terms of character recognition accuracy is shown in table 10. This table depicts that the Hough Line Transform method contributes to a higher rate of character recognition among other approaches.

Table 10:-*Recognition Accuracy with different Skew Detection Methods*

Methods	**Accuracy**
Hough Lines Transform	65.15%
PCA	65.04%
Minimum Area Rectangle	64.85%

Table 11 depicts recognition accuracy of OCR engine with 300DPI is higher as compared to others.

Table 11:-*Recognition Accuracy with Different DPI Images*

DPI	Accuracy
150	61.12%
200	61.07%
300	61.64%
400	61.31%
500	61.15%
600	61.13%

Table 12 shows that Otsu method of Binarization provides better recognition accuracy for OCR among others.

Table 12:-*Recognition Accuracy with Different Binarization Approaches*

Binarization Method	Accuracy
Kittler Thresholding	62.10%
Otsu Binarization	63.23%
Niblack Binarization	62.83%
Bernse Binarization	62.59%
Sauvola Binarization	63.08%

Table 13 shows that Moment-based Normalization technique among others contributes greater recognition accuracy in OCR.

Table 13:-*Recognition Accuracy with Different Normalization techniques*

Normalization Techniques	Accuracy
Linear Normalization	59.32%
Non-linear Normalization	60.78%
Moment-based Normalization	62.15%

The results from the test experiment conducted for analyzing the effect of image preprocessing pipeline in the recognition accuracy of OCR engine is shown in the Figure 5.

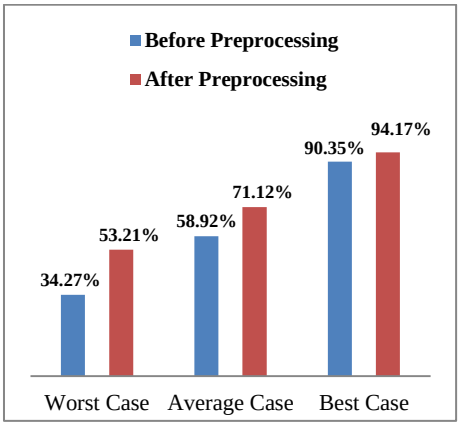

Fig.5:-*Graph Illustrating Overall Experimental Result*

Figure 5 shows that the proposed preprocessing pipeline contributes to a positive effect in the text detection rate and optical character recognition accuracy of Tesseract OCR engines.

CONCLUSION

This work involves addressing the issues

of photographed text image via image preprocessing technique so that the quality of image gets improved and thus the recognition rate of OCR engine gets increased. For this, a pipeline of preprocessing steps has been developed and test experiments were conducted to check the impact of the preprocessing pipeline on the recognition accuracy of OCR engine. The results of the test experiment shows significant improvement in the recognition efficiency of OCR with the developed image preprocess pipeline. However, the limitation of this work is that the presence of non-textual objects in a text image affects the recognition efficiency of OCR engines. Also, if the text image is composed of multiple columns of text areas then the performance of the OCR engine gets affected.

REFERENCES

1. Khedekar, S., Ramanaprasad, V., Setlur, S., & Govindaraju, V. (2003, August). Text-image separation in devanagari documents. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 1265-1269). IEEE.
2. Kompalli, S., Nayak, S., Setlur, S., & Govindaraju, V. (2005, August). Challenges in OCR of Devanagari documents. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 327-331). IEEE.
3. Smith, R. (2007). An Overview of the Tesseract OCR Engine. In proceedings of Document analysis and Recognition. ICDAR.
4. Bieniecki, W., Grabowski, S., & Rozenberg, W. (2007, May). Image preprocessing for improving ocr accuracy. In *2007 International Conference on Perspective Technologies and Methods in MEMS Design* (pp. 75-80). IEEE.
5. Alginahi, Y. (2010). Preprocessing Techniques in Character Recognition, Character Recognition, Minoru Mori (Ed.), ISBN: 978-953-307-105-3, InTech.
6. Bansal, V., & Sinha, M. K. (2001, September). A complete OCR for printed Hindi text in Devanagari script. In *Proceedings of Sixth International Conference on Document Analysis and Recognition* (pp. 0800-0800). IEEE Computer Society.
7. Yadav, D., Sánchez-Cuadrado, S., & Morato, J. (2013). Optical character recognition for Hindi language using a neural-network approach. *JIPS*, *9*(1), 117-140.
8. Gupta, D., & Nair, L. (2013). Improving OCR By Effective Pre-Processing and Segmentation for Devanagiri Script: A Quantified Study. *Journal of Theoretical & Applied Information Technology*, *52*(2).
9. Badla, S. (2014). Improving the efficiency of Tesseract OCR Engine.
10. Bawa, R. K., & Sethi, G. K. (2014). A binarization technique for extraction of devanagari text from camera based images. *Signal & Image Processing*, *5*(2), 29.
11. Harraj, A. E., & Raissouni, N. (2015). OCR accuracy improvement on document images through a novel pre-processing approach. *arXiv preprint arXiv:1509.03456*.
12. Kale, P., Phade, G. M., Gandhe, S. T., & Dhulekar, P. A. (2015, January). Enhancement of old images and documents by digital image processing techniques. In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)* (pp. 1-5). IEEE.
13. Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

14. Pant, N. (2016). Nepali OCR Using Hybrid Approach of Recognition. Master's Thesis.
15. Shoumorup M. (2017). Improving the Efficiency of Tesseract OCR through Super Resolution, Project under the Summer Research Fellowship Programme 2017,
16. Hambal, A. M., Pei, Z., & Ishabailu, F. L. (2017). Image noise reduction and filtering techniques. *International Journal of Science and Research (IJSR)*, *6*(3), 2033-2038.
17. Tong, X., & Evans, D. A. (1996). A statistical approach to automatic OCR error correction in context. In *Fourth Workshop on Very Large Corpora*.
18. Cheng, H. D., & Shi, X. J. (2004). A simple and effective histogram equalization approach to image enhancement. *Digital signal processing*, *14*(2), 158-170.
19. Singh, G., & Singh, S. (2015). An Enhancement of Images Using Recursive Adaptive Gamma Correction. *Gagandeep Singh et al,/(IJCSIT) International Journal of Computer Science and Information Technologies*, *6*(4), 3904-3909.
20. Fu, B., Li, W., Wu, M., Li, R., & Xu, Z. (2012). A document rectification approach dealing with both perspective distortion and warping based on text flow curve fitting. *International Journal of Image and Graphics*, *12*(01), 1250002.
21. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.
22. https://github.com/tesseract-ocr
23. https://github.com/tesseract-ocr/tessdata
24. https://pypi.python.org/pypi/pytesseract