IJCSN - International Journal of Computer Science and Network, Volume 9, Issue 2, April 2020
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Interpretation of English Documents in Kannada using Automated Cropping Technique

[1] Chirag Kamath; [2] Neha Prakash; [3] Dr. Radhika K R

[1] Department of Information Science Engineering, BMS College of Engineering
Bangalore, Karnataka 560019, India

[2] Department of Information Science Engineering, BMS College of Engineering
Bangalore, Karnataka 560019, India

[3] Professor, Department of Information Science Engineering, BMS College of Engineering
Bangalore, Karnataka 560019, India

Abstract – The Indian subcontinent houses 1,652 different languages. Communication between the central and state government occurs through the exchange of documents in English or Hindi. Kannada is the official language of Karnataka, a southern Indian state. State officials are often well versed in Kannada and prefer it over other languages. The motivation for this capstone project was to aid government offices by providing a medium for translating documents written in English to Kannada in a cost-effective way. Moreover, existing techniques are constrained by the orientation of input images, requiring users to capture input images at specific angles. This paper proposes a novel, semi-automated methodology that facilitates precise text extraction from document images captured via mobile phone and supports English to Kannada translation. The proposed system was evaluated on a database of document images and achieved precision of 0.9, recall of 0.92, F measure of 0.91 which is comparable to state-of-art techniques.

Keywords - *OCR, text extraction, language translation, skew correction.*

1. Introduction

The emerging technologies concerning deep learning, natural language processing, and image processing, translation is gaining greater importance. Most countries in the world are transitioning from unilingual and bi-lingual to multi-lingual cultures. The repercussions of this change are that either all public information must be available in all languages or one must be proficient in all languages. Government offices located locally also face a similar issue; officials may not understand documents in languages other than the native language. This paper is concerned with the translation of electronic documents from English to Kannada. Being a Dravidian language, it has been in use for about one and a half millennia and currently has around 56 million speakers.

Parameswarappa et al. [24] draws an apt conclusion regarding the semantics of Kannada stating that it is morphologically intricate and complex while having a relatively free order. Machine Translation (MT), a sub-field of computational linguistics, deals with the use of software to translate text and speech from one language to another. Recently, MT has gained a lot of attention, nevertheless, it still faces a lot of challenges. Firstly, the number of applications that provide ready to use applications for scanning a document, translating it to another language and printing the translated text, is very limited. Extracting text from a scanned document is challenging due to low font size, presence of watermarks, and other background noises. Additionally, if a word is misspelled in the original document language, then the translation of that word will either produce an incorrect translation or the word does not get translated at all. The most challenging aspect of translation is that Kannada does not have the concept of prepositions. The translation is provided by adding inflections to the noun of the prepositional phrase. Furthermore, every preposition in English can be translated to Kannada in more than one way. Ambiguities of this kind will have a detrimental impact on the accuracy of translation. There are two broad classifications of natural language processing for translation, Rule-based classifiers, and statistical methods, as indicated by Shivakumar et al [50]. A Rule-based approach relies on collecting information through the supplied training data regarding a subject and forming rules. For each translation, the classifier scans each of these rules. In contrast, a system based on statistical models acquires knowledge during the training phase and uses the acquired knowledge for translation. Both methods

have advantages. For languages with numerous resources, a Rule-Based approach may produce commendable accuracy and Statistical machines perform better when provided with a parallel corpus. Finding patterns in the generated rules along with knowledge acquired over time allows the proposed method to achieve a delicate balance between Rule-Based and Statistical approaches. ROI is filtered out by a document extraction module from the input image using a myriad of techniques which include Canny edge detection, contour detection, and thresholding. Cropping of a region containing text from the original image, essentially removing the background, while still accounting for image orientation by means of a skew correction module is of primary importance. Scanned image quality is enhanced by using capabilities like erosion-dilation, noise removal, and Gaussian blurring. The image is then fed into a Character Recognition module to facilitate text extraction. The extracted text is further translated into Kannada to generate a new document. The paper is organized as follows. The following section elucidates the important existing work done in fields concerning Image Processing, Optical Character Recognition(OCR) and Language Translation. Delineating the modular structure of the proposed architecture, section 3 discusses design methodologies and algorithms employed in this work. Section 4 describes an experimental setup, evaluation metrics and the results obtained. Sections 5 and 6 conclude the paper highlighting its usability, effectiveness and shed light on the scope for further research.

2. Literature Review

Image cropping is a fundamental task in image editing to enhance the aesthetic quality of images. In the past, image cropping has transformed into an automated task that is targeted at improving efficiency and reducing time spent on preprocessing. Kao et al. proposed to utilize an aesthetic map and a gradient energy map to learn a composition model from a large professional dataset. However, previous works considered saliency maps and various handcrafted features to learn models from extensive datasets [11]. Kao et al. evaluated their automatic image cropping model against the Atomic Visual Action, AVA dataset and it consistently showed better results as compared to existing works while using the maximum overlap metric. Guo et al. further extended this work by replacing the aesthetic map-based approach using deep learning models, such as Convolutional Neural Network (CNN) and Cascaded Regression [12]. A pre-trained CNN model was used for the extraction of CNN features. In the next stage, a primitive regressor was used to estimate an improved cropping region by means of CNN features, spread across T stages of regression. The neural network-based model was evaluated against CHUKPQ dataset containing 950 images and performed nearly 3 times better than the aesthetic based method proposed by [11]. Both models performed the automatic cropping functionality relatively well on image datasets. The neural network model was more efficient and involved less implementation complexity, making it a more attractive and novel approach as compared to the aesthetic map model in [11]. The extraction of ROI from the input document image, proposed in this paper uses a combination of image processing based mathematical models to achieve the required functionality and is inspired by the mathematical models used in the neural network-based approaches.

An essential aspect that is taken into consideration when automatic image cropping is applied to document images is the presence of a skew correction module. Skew correction must always be employed to obtain a birds-eye or a top-down view of an image regardless of its input image orientation. The most significant stage of skew correction for scanned document images is to estimate the skew angle. Conventional methods are mostly based on Hough transformation. The downside of Hough Transformation is that it is often affected by text structure or other noise [19]. To tackle this issue, a cogent solution was proposed by Wang et al. which is a robust skew correction algorithm based on low-rank matrix decomposition theory. The low-rank matrix decomposition model was an extension of the Transform Invariant Low-rank Textures, TILT, model. Furthermore, successful noise removal in addition to skew correction while achieving marginally better results as compared to Hough transformation in terms of efficiency and processing speed is a compelling aspect of the Low-Rank matrix decomposition. Li et al. addresses skew correction in terms of line and text extraction using OCR, proposing a novel sub-region-based approach applicable for generic printed text images with no parameter tuning required. Guided by the spacing between text lines, the detection of a skew angle between ±90° was feasible [13]. Relevance of the subregion-based approach is immense as it addresses skew correction relative to text extraction using OCR which is the basis of the model being proposed in this paper. Nevertheless, state-of-the-art approaches that are aimed at analyzing document images continue to suffer from limited detection range and application-specific parameter tuning. Extending the approach proposed in [13], in this proposed work, skew correction is coupled with the automatic cropping module that utilizes a four-point detection algorithm. Perspective transform and local

thresholding follow, in order to carry out skew correction and cropping in tandem. Input image quality is of the utmost importance when the preprocessed image is fed into the OCR for text extraction. Pandey et al. put forth a nonlinear fusion of multiple interpolations (NFMI) which is a CNN based model to generate a high-resolution document image from a low-resolution image, achieved by improving the spatial resolution of document images [30]. Word recognition accuracy substantially improved by NFMI over traditional interpolation techniques. Evaluation of NFMI against a Tamil document dataset resulted in an improvement of 54% for upscaling the resolution by a factor of 2 and 33% for upscaling by a factor of 4. The proposed model adopts multiple noise removal techniques to facilitate efficient text extraction using OCR. Noise removal entails removing salt and pepper noise by means of a Median Filter followed by a Gaussian smoothing module to eradicate high-frequency noise. The two-stage noise removal technique improves image quality and the overall OCR efficiency.

The recognition of optical character plays an important role in finding information for machine-editable text formats from pixel-based images [25]. A breakthrough OCR algorithm was posited by Chidiac et al. which overcomes the constraint of image orientation by means of a Maximally Stable Extremal Regions (MSER) model. The MSER method was evaluated against ICDAR dataset along with the KAIST Scene Text Database. The results of the experiment showed a performance gain of 8% in terms of Precision rate, 2% in terms of Recall rate, and 1% in terms of F-measure as compared to traditional OCR [3]. Wei et al. proposed a deep learning-based solution for text extraction using OCR through the Inception v3 model which used a pre-trained model with transfer learning. Inception v3 had fast training time with high accuracy as compared to the traditional CNN based OCR, gaining high significance due to reliability. Deep learning-based OCR's in general, tend to outperform traditional OCR's and are considered state of art when it comes to text extraction in the fields of image processing and computer vision. The proposed application uses a similar Deep learning-based open-source OCR called Tesseract 4.1.0. Tesseract is an ensemble of traditional OCR algorithms and additionally implements a Long Short-Term Memory (LSTM) based recognition engine. LSTM is a kind of Recurrent Neural Network. Tesseract OCR is one of the most accurate OCR's today as it combines several text extraction and classification algorithms such as line recognition and character pattern recognition.

3. Proposed Method

The proposed work employs a five-module application as illustrated by Fig. 1. The first module is the input acquisition module. It deals with capture of image on a mobile device, remote transfer to the computer where the proposed application is executed. This work is tailored to work for document images, consequently the input for the scope of this work is restricted to images of documents.

The Document extraction module follows the Image Acquisition module. This module consists of extracting the ROI, elimination of any background if present along with other pre-processing steps, thereby filtering out the ROI. OCR engine is used to carry out the extraction stage.

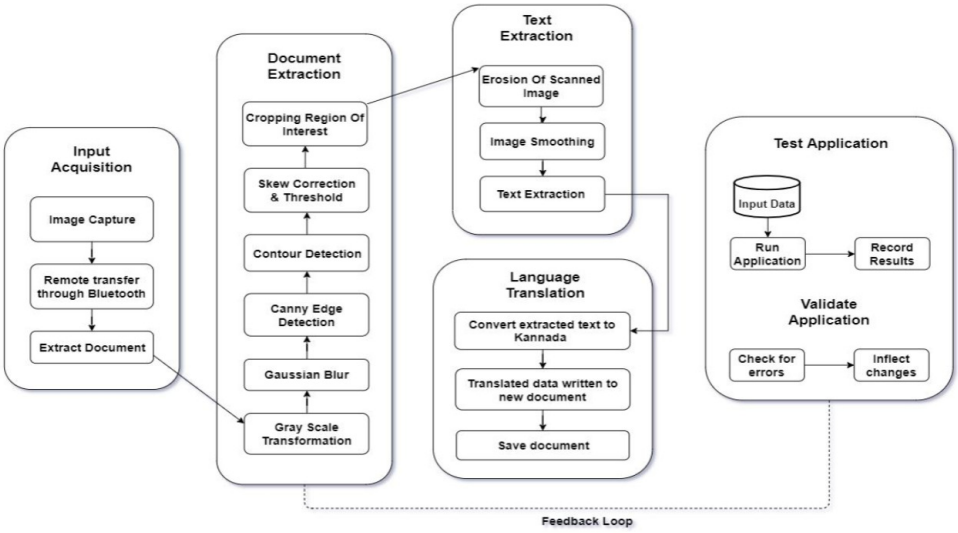

Fig. 1 Proposed architecture

Language Translation module uses Google Translate API to convert text from English to Kannada. The last module is the testing and validation module, dealing with running the application recording metrics such as precision, recall and F-measure.

Fig. 2 depicts the stages of processing the input image goes through, following which we obtain our skew corrected image which has been cropped. The first set of images show the cropping process on a piece of paper without any text and the second set of images show the cropping process on a document image with a background.

3.1 Detecting Region of Interest and Skew Correction

The first step in any document analysis system is to define the ROI. The ROI refers to the part of the image (or frame in the case of video multimedia) that is of interest for further processing. Detection and separation of ROI often go hand in hand. Extracting the ROI from the input image includes a skew correction module which always makes sure that the detected region is always extracted such that it appears in a top down position during the cropping stage. The inclusion of skew correction capabilities is highly significant as Tesseract OCR performs well only when a top down view of the image is available.

3.1.1 Four Point Detection

The Four Point Detection algorithm, as depicted in Algorithm 1 is used to detect and derive a list of coordinates for the four corners of the ROI. Since the ROI in all cases are pictures of documents, it is valid to consider the ROI to be a rectangle. Algorithm 1 takes a single argument, *co-ords*, which is a list of four points specifying the x and y coordinates of each point of the rectangle. It is crucial to have a consistent ordering of points in the rectangle. The four corner points are ordered as top-left, top-right, bottom-right, and bottom-left. Memory is then allocated for the four ordered points.

The top left point has the smallest $x + y$ value, while the bottom right point has the largest $x + y$ value. The next step is to take the difference (i.e. $x - y$) between the points using the *np.diff()* function. Coordinates associated with the smallest difference will be the top right points, whereas coordinates with the largest difference will be the bottom left points. Finally, a list consisting of the four co-ordinates that denote the four corners of the rectangle is returned.

Algorithm 1 Finds 4 corner points of the ROI

Input: List of coordinates of the ROI
Output: Array containing 4 corner points of the ROI
Procedure: get_points(co-ords) :

```
corners = np.zeros((4, 2), dtype = "float32")
s = co-ords.sum(axis = 1)
corners[0] = co-ords[np.argmin(s)]
corners[2] = co-ords[np.argmax(s)]
diff = np.diff(co-ords, axis = 1)
corners[1] = co-ords[np.argmin(diff)]
corners[3] = co-ords[np.argmax(diff)]
return corners
```

3.1.2 Skew Correction

Skew correction module coupled with the four-point detection algorithm, provides an important functionality which greatly benefits the accuracy of the OCR. The *skew_correct()* function defined in Algorithm 2 takes in an input image and detects the four corners of the ROI while simultaneously performing a perspective transform of the image to obtain a top down view of the image, thereby facilitating better text extraction. The *skew_correct()* function requires two arguments, *img* and *co-ords*. *img* variable is the input image for which the perspective transform is applied. *co-ords* is a list of four points that contain the ROI of the image to be transformed. *Corners* variable represents the output when the *get_points()* function defined in Algorithm 1 is invoked. A function call to *get_points()* function, places the *corners* variable in a consistent order for processing. Four corner points of the ROI which are stored in *corners* list is unpacked and assigned to individual variables.

The next stage involves determination of the dimensions of the newly warped image. The width of the new image is calculated since the width is the largest distance between the bottom right and bottom left x-coordinates or the top right and top left x-coordinates. The image height is determined using a similar methodology, where the height is the maximum distance between the top right and bottom right y-coordinates or the top-left and bottom left y-coordinates.

The main objective is to obtain a "birds eye view" of the ROI in the original image. Four points are defined to represent the 'top-down' view of the image where the first entry in the list is (0, 0) indicating the top-left corner. The second entry is (*maxWidth - 1, 0*) which signifies the top-right corner. Similarly, (*maxWidth - 1, maxHeight - 1*) and

(*0, maxHeight - 1*) represent bottom right and bottom left corners respectively.

In order to obtain the required top-down view of the image, *cv2.getPerspectiveTransform()* method is utilized which requires two arguments, *corners*, which is a list of four corner co-ordinates of the ROI and *final*, which is a list of transformed points. *cv2.getPerspectiveTransform()* function returns '*transformed_matrix*', which is the actual transformation matrix.

Algorithm 2 Image Skew Correction

Input: The input image to be transformed and the four corners of the ROI
Output: Skew corrected image
Procedure: skew_correct(img,co-ords) :

```
corners = get_points(co-ords)
(tl, tr, br, bl) = corners
width1 = np.sqrt(((br[0]-bl[0])**2)+((br[1]-bl[1])**2))
width2 = np.sqrt(((tr[0]-tl[0])**2)+((tr[1]-tl[1])**2))
maxWidth = max(int(width1), int(width2))
height1 = np.sqrt(((tr[0]-br[0])**2)+((tr[1]-br[1])**2))
height2 = np.sqrt(((tl[0]-bl[0])**2)+((tl[1]-bl[1])**2))
maxHeight = max(int(height1), int(height2))
final = np.array([ [0, 0], [maxWidth - 1, 0],
      [maxWidth - 1, maxHeight - 1],
      [0, maxHeight - 1]], dtype = "float32")
transformed_matrix = cv2.getPerspectiveTransform(corners,
final)
res=cv2.warpPerspective(img,transformed_matrix,
(maxWidth,maxHeight))
return res
```

(a) Original Image (b) Edge Detection (c) Detection of ROI (d) Final Image

Fig. 2 Image pre-processing for text extraction

In the final stage, transformation matrix is applied using the *cv2.warpPerspective()* function which takes parameters *image*, *transformed-matrix*, along with the *width* and *height* of the output image. The result generated by *cv2.warpPerspective()* is the top down view of the original image.

3.2 Automatic Image Cropping

Image cropping is the process of removal of an unwanted background from an image. In the context of text extraction via OCR, the image cropping step is of high significance as it reduces the possibility of misinterpreting characters that don't exist and hence reduce the false positive rate. To achieve this cropping a semi-automated algorithm is used as depicted in Algorithm 3, that automatically detects and crops the ROI without any action from the user.

In order to speedup image processing, as well as make the edge detection step more accurate, the scanned image is resized to obtain a height of 500 pixels.

The image is converted from RGB to grayscale and Gaussian blurring is performed to remove high frequency noise which aids in contour detection, performed in a later stage.

The cropping module must be able to scan the document from the input image while cropping out the rest. A piece of paper is assumed to be a rectangle. A rectangle has four edges. A simple heuristic can aid in further enhancement of document cropping and scanning. It is justifiable to assume that the largest contour in the image with exactly four corner points is our piece of paper to be scanned. The algorithm assumes that the document to be scanned is always the predominant focus of the input image. Furthermore, the proposed work assumes that the piece of paper has four edges. The steps involved in Automatic Image Cropping is illustrated by Fig. 2.

3.2.1 Edge Detection

The first stage of the cropping process involves detecting the edges in the input image, thereby extensively reducing the amount of data to be processed while simultaneously maintaining the structural integrity of the image. Fig. 2b shows how the edges are detected in the input image. OpenCV's Canny edge detection algorithm is used in the proposed method as it produces highly reliable results and detects maximum edges in the input image [69].
The *cv2.Canny()* method is a four-stage algorithm that starts off by carrying out noise removal by means of a 5x5 Gaussian filter. The smoothed image is further filtered along the x and y axis to obtain the horizontal (G_x) and vertical (G_y) derivatives of the first order.

The Edge Gradient and pixel direction can be calculated as:

$$Edge\_Gradient\ (G) = \sqrt{G_x^2 + G_y^2} \tag{1}$$

$$Angle\ (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2}$$

The Edge Gradient calculated in Eq. (1) is used to categorize edges as strong and weak edges based on whether it is within a specified threshold. The Gradient direction in Eq. (2) is always perpendicular to edges. The Gradient angle is approximated as one of the four angles representing vertical, horizontal and two diagonal directions (0°, 45°, 90° and 135°).

The last two stages of the *cv2.Canny()* method involve Non-Maximum suppression and Hysteresis thresholding [69]. These methods deal with removal of unwanted pixels that may not constitute edges by using a local maximum and a thresholding range respectively. The final output image consists of all the edges detected as "sure edges" and unwanted noise is removed.

3.2.2 Contour Detection

The Contour detection stage as the name suggests, is used for finding the contours of the ROI in the image. A contour refers to a closed curve or line segment of which encloses the boundaries of an object. The objective is to find the contours in the edged image. *cv2.findContours()* method is employed for contour detection. It returns a list of contours detected in our pre-processed image. The contours are then sorted by area and only the largest ones are of significance to enhance the performance. The algorithm stores the largest contours while discarding the rest. The contours are looped over to approximate the number of points. If the approximated contour has four points, then the algorithm assumes that document in the image is found. The result of contour detection successfully detects all contours in the image and approximates the resultant contour which encloses the ROI. Fig. 2c shows the largest contour detected in the image with the green bounding box depicting the ROI.

Algorithm 3 Automatic Image Scanning and Cropping

```
Input: The input image to be cropped
Output: Cropped and Skew corrected image
Procedure: scan_doc(img) :
ratio = image.shape[0] / 500.0
copy = image.copy()
img = imutils.resize(img, height = 500)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (5, 5), 0)
edges = cv2.Canny(gray, 75, 200)
contours=cv2.findContours(edges.copy(),
cv2.RETR_LIST)
contours = imutils.grab_contours(contours)
contours = sorted(contours, key = cv2.contourArea,reverse
= True)[:4]
for i in contours do
    perimeter = cv2.arcLength(i, True)
    approx = cv2.approxPolyDP(i, 0.02 * perimeter, True)
    if len(approx) = 4 then
        finalCnt = approx
        break
    end if
end for
transformed = skew_correct(copy, finalCnt.reshape(4, 2) *
ratio)
transformed=cv2.cvtColor(transformed,
cv2.COLOR_BGR2GRAY)
T = threshold_local(transformed, 11, limit = 10, method
= "gaussian")
transformed = (transformed ≥ T).astype("uint8") * 255
cropped = imutils.resize(transformed, height = 700))
return cropped
```

3.2.3 Image Resizing

The last step in building automatic cropping module is to take the four points representing the outline of the document and apply a perspective transform to obtain a top-down, "birds eye view" of the image.

First step in obtaining the required view is to perform a warping transformation. The *skew_correct()* function from Algorithm 2 is called which was defined earlier to obtain a warped image with upright orientation. The *skew_correct()* method is passed two arguments, the first is our original input image and the second argument is the contour representing the document, multiplied by the resized ratio.

The contour detected in the contour detection stage is multiplied by the resized ratio. The resizing is carried out because the edge and contour detection were performed on the resized image and not the original input image. Since the scanning is required to be performed on the original image and not the resized image, the contour points are multiplied by the resized ratio.

Fig. 2d shows the final image which crops out the ROI detected in Fig. 2c and applies the skew correction to obtain the upright view of the input image.

To obtain a black and white feel to the image, the warped image is converted to grayscale, following which adaptive thresholding is applied. This step is essential as the Tesseract OCR employed in the following stages performs best on black and white images. Lastly, image resizing is carried out to correct the aspect ratio of the resultant skew corrected image which is then fed into the OCR after undergoing further image quality enhancement.

3.3 Upscaling Image Quality

The cropped image obtained from the previous stage may have small amount of noise associated with it. Scanned images often have a component of noise associated with it, either due to cropping or due to digitization. Noise is generally considered to be a random variable with zero mean. A pixel p_0 is said to be noisy if its value is altered as shown in Eq. (3).

$$p = p_0 + n \tag{3}$$

where p is the new pixel value, p_0 is the initial value of the pixel and n is the amount of noise. In a noise free image, $p=p_0$, as the mean noise is zero over all the pixels. Such scenarios can be handled by means of noise removal techniques provided by OpenCV [69]. Commonly occurring noise is usually categorized as Gaussian and salt-pepper noise. The best available option that successfully deals with the various types of noise is *cv2.fastNlMeansDenoising()*. This method uses Non-Local Mean Denoising which employs a sliding window technique around a pixel. A small window is taken around the pixel, a search is conducted for similar windows in the image and an average of all the windows is calculated while simultaneously replacing the pixel with the average value.

3.4 Enhancing Tesseract Accuracy

The processed image obtained so far can be fed into Tesseract and results obtained are relatively accurate. However, it is possible to further enhance the accuracy by carrying out pre-processing, specific to Tesseract. Certain fonts may in fact have alphabets connected to each other. Since Tesseract uses connected component analysis techniques, there is a possibility of misinterpreting two or more alphabets as a single alphabet. Such issues are tackled by employing morphological transformations such as erosion and dilation. Erosion and dilation make use of a kernel which is convoluted into the image. Erosion is used to remove boundaries of the foreground object and white noise. Further, it aids in disconnecting two connected objects which are not meant to be connected. Dilation is the opposite of erosion and is used to increase the object area which is shrunk due to erosion. It also enhances the features of the foreground object. Erosion and dilation are often used together, especially as a pre-processing step in the field of OCR.

3.5 Character Recognition

Character recognition is a term coined for converting printed or handwritten characters from images to ASCII characters that a computer can recognize. An OCR system begins by converting the input image to grayscale, followed by filtering. Filtering is a process of modifying an image, generally for the purposes of enhancement. The task of filtering comprises smoothing, sharpening, and edge enhancement. Feature extraction is the next step. Feature extraction is a method wherein the original data is represented by a reduced number of variables while capturing the most discerning information. This is followed by recognition of the pattern. This pattern is compared with the stored ASCII values and it outputs the match.

The proposed method uses Tesseract OCR for text recognition. Tesseract is an open-source OCR engine

originally developed by HP and was the first OCR engine to handle white on black text efficiently [27]. The architecture of Tesseract engine is shown in Fig. 3. Tesseract performs connected component analysis which aids in recognition of nested outlines, number of child and grandchild outlines, etc. A line finding algorithm is employed to identify the text lines [27]. The outlines are gathered in the form of blobs by nesting. Baseline fitting allows Tesseract to handle images with curved pages, following which, Tesseract differentiates each character of a word [27]. Text is broken into words based on definite and fuzzy spacing. Further, Tesseract attempts recognizing words using a two-stage recognition process. The detected words are used as training data for an adaptive classifier to further improve accuracy on the second pass.

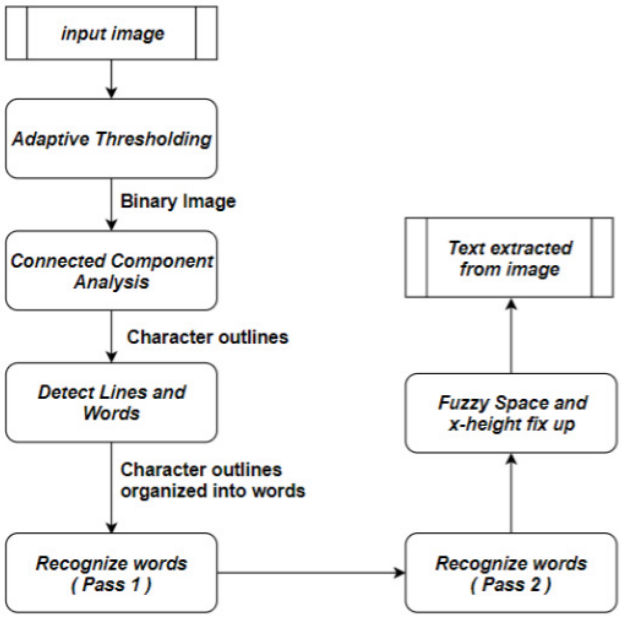

Fig. 3 Tesseract architecture [71]

3.6 Language Translation

An approach known as Statistical Machine Translation (SMT) has been one of the most successful approaches to machine translations. SMT begins with a large text corpus for which translations are already available. When SMT is applied on new texts, it can make a reasonable guess based on the information it has.

Let *e* represent a text in English language, which is to be translated to Kannada text represented by *k*. As with any language, there are multiple ways to perform translation. Discrepancies obtained are modeled with a probability distribution defined in Eq. (4). It is reasonable to assume that the best translation accuracy is when *e* selected such that it maximizes conditional probability.

Bayes's Theorem is used to maximize conditional probability, shown in Eq. (4).

$$Pr(k \mid e) = \frac{Pr(e \mid k)\, Pr(k)}{Pr(e)} \tag{4}$$

From Eq. (4), *e* is clearly a constant, thus, maximization over *k* can be achieved by maximizing Eq. (5)

$$Pr(e \mid k)\, Pr(k) \tag{5}$$

Google Translate (gtrans). [68] worked by translating the given text to an intermediary language, English, and then to the required language. gtrans API that is used in the proposed work also originally employed statistical models for translation. However, gtrans now uses a deep learning approach for translation.

In the proposed work, data extracted from Tesseract is in English. Extracted data is fed into gtrans API which results in the translations of the given text in Kannada as its output. The translated data is then written into a new document and automatically stored on the user's system. Language Translation is illustrated in Fig. 4.

4. Experiment

In this section, experimental metrics used to obtain results for automatic image cropping. A side by side comparison is also carried out between the state of art techniques and the proposed method.

4.1 Performance Metrics

4.1.1 Precision

In pattern recognition, Precision is defined as the fraction of relevant instances among the retrieved instances. Precision indicates probability that a randomly selected image is relevant. In this work, *Precision* is calculated as the ratio of the number of matching words and the length of output scanned document or the output text. Eq. (6) depicts the formula for *Precision*. *NumberofMatchingWords* represents the

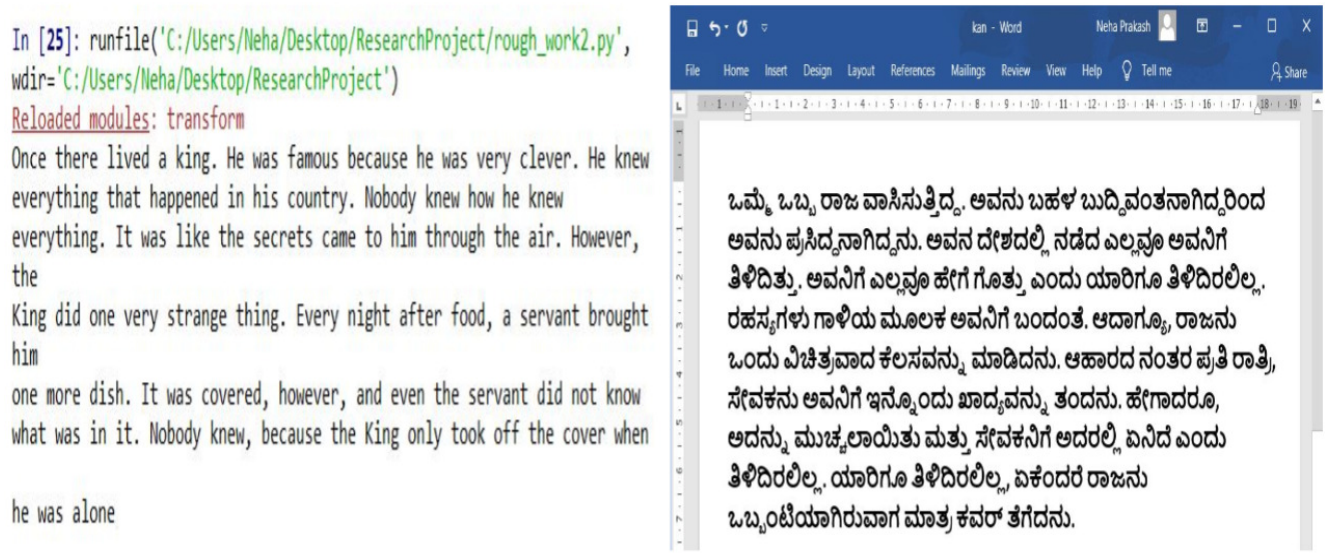

Fig. 4 Translation from English to Kannada

number of words that match between original and output text.

LengthofOutputText represents total number of words in the output text.

$$Precision = \frac{NumberofMatchingWords}{LengthofOutputText} \tag{6}$$

4.1.2 Recall

In pattern recognition, Recall is the fraction of the total amount of relevant instances that were actually retrieved. In the proposed method, *Recall* is calculated as the ratio of the number of matching words to the total number of words in the *ReferenceText*, original scanned document.

The formula for *Recall* is represented in Eq. (7). *NumberofMatchingWords* represents the number of words that match between original and output text and *LengthofReferenceText* represents total number of words in the reference text.

$$Recall = \frac{NumberofMatchingWords}{LengthofReferenceText} \tag{7}$$

4.1.3 F Meaure

F Measure is calculated by taking the harmonic mean of the *Precision* and *Recall*.

The formula for F Measure is represented in Eq. (8),

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{8}$$

4.2 Comparison with state-of-art techniques

In this subsection, performance of the proposed work is evaluated for text extraction and the results are compared with other methods. The proposed model is evaluated against a database of document images, achieving an overall Precision of 0.9 with a Recall of 0.92 and F Measure of 0.91. Table 1 depicts the results obtained are satisfactory and competitive with the latest methods developed. The proposed method performs equally well or shows marginal improvement as compared to [3], [62], [64]. The model shows significant improvement when compared with [2], [61] and [63].

Islam et al. [65] proposed a Hybrid algorithm for extracting text-based regions from scene images. The Hybrid algorithm in [65] is compared with established methods like connected components and edge-based methods based on precision, recall and accuracy achieved. The Hybrid Algorithm achieved a precision of 0.86 and recall of 0.95.

Our proposed work adopts an evaluation technique suggested by Islam et al. Text extraction and machine translations are performed on a specific image using a

Table 1 : Performance comparison of Proposed method with respect to state-of-art methods for Text Recognition and Extraction

Proposed By	Method	Precision	Recall	F Measure
Prasad et al. [2]	Semi-automated technique	0.59	0.69	0.64
Chidiac et al.[3]	Technique based on MSER and stroke width detectors	0.9	0.88	0.89
Epshtein et al. [61]	Stroke-based Method	0.73	0.6	0.66
Yadav et al. [62]	Technique based on FAST key points	0.89	0.87	0.88
Wahyono et al. [63]	Fast Stroke Width Transform	0.61	0.63	0.62
Sahoo et al. [64]	Technique using cellular Automata	0.89	0.93	0.91
Islam et al. [65]	Hybrid Algorithm	0.86	0.95	0.9
Park et al. [66]	Automatic Word Detection System	0.78	0.92	0.84
	Proposed Method	0.9	0.92	0.91

standard technique as used in [65] and results are recorded.

Wahyono et al. proposes Fast Stroke Width Transform (FSWT) which achieves an average precision of 0.61 and recall of 0.63, tested on a dataset consisting of 146 images. Although the values are comparably lower than Islam et al., FSWT is primarily for text extraction in complex environments. In such complex cases, [65] may not be able to achieve the same performance. Furthermore, FSWT is independent of language as it can detect multilingual text. FSWT's results are compared with that of Epshtein et al. The former achieves a significant improvement on the latter in processing time while achieving similar precision and recall rates.

Text extraction is a technique that has a lot of scope for practical application. Text Extraction was implemented by Prasad et al. for generating slide presentations by extracting text from English documents. The technique used includes a text extractor and slide generator which achieved an overall recall rate of 0.69 and precision of 0.59. They also created ontologies for the input text to further enhance the experimentation stage. The idea of using cutting-edge research in a real-world application is one that is shared with the proposed work. In our proposed work, text extraction is employed to aid in machine translation of scanned documents. The language translation stage can be carried out only if the text is first extracted from the scanned image and this extracted text is then translated.

Table 2 illustrates the performance of the proposed method as compared to directly using Tesseract on images. The proposed work makes use of four-point detection algorithm, skew correction and automatic cropping to obtain a high-quality image. When the

altered image is used, Tesseract can extract the text with relative ease and better accuracy. If Tesseract is directly applied to images without the required preprocessing, the results obtained are not very useful due to presence of spelling errors and missing words.

Performance comparison based on text extraction between the proposed work and Tesseract are carried out on scanned images of documents. Table 2, depicts the results obtained when the proposed method and Tesseract are applied to the same five images. The proposed work shows significant improvement over Tesseract with an increase in F Measure of 14%.

Table 2 : Comparison of Proposed Method with Tesseract

	Algorithm	Precision	Recall	F Measure
Image 1	Tesseract	0.93	0..92	0.92
	Proposed	0.98	0.99	0.98
Image 2	Tesseract	0.7	0.72	0.71
	Proposed	0.81	0.87	0.84
Image 3	Tesseract	0.7	0.78	0.74
	Proposed	0.94	0.94	0.94
Image 4	Tesseract	0.88	0.9	0.89
	Proposed	0.86	0.89	0.88
Image 5	Tesseract	0.68	0.73	0.7
	Proposed	0.89	0.9	0.89

5. Conclusion

In this work, a fast and robust document translation service is presented that makes use of both OCR and language translation functionalities. The entire service is semi-automated, thereby requiring minimum effort from the user.

The input image is first transferred from the user's phone to a high-performance computing system by means of Bluetooth technology. The image is fed into the application which makes use of a extraction algorithm that accurately extracts the document representing the ROI and further improves image quality using de-noising techniques. The processed image is fed into the Tesseract OCR engine for retrieval of English data and language translation

capabilities of Google translate API is leveraged to efficiently translate the English text to Kannada.

The performance is highly dependent on the ontology and quality of the input provided. Input image must be captured in an environment with good lighting. There must be clear distinction between the document paper to be extracted and the background of the image to ensure good results. Images captured in such favorable settings result in high performance when compared to other works as depicted in Table 2. A significant improvement is achieved by using the proposed work instead of using Tesseract directly on the input image as shown in Table 2. The proposed approach has wide spread applications because the entire process is automated, and the application provides a portable document translation service.

6. Future Work

The current proposed system can further be enhanced in the future by carrying out certain optimizations to broaden the scope of usage to multiple domains. One of the main issues that severely affects the accuracy of the OCR, is that images within the input document are treated as data of importance leading to error riddled output data. A noteworthy future enhancement would be for the OCR to extract text irrespective of images, border etc. in the input document.

A second enhancement to the model would be to make the OCR independent of font and image quality requirements. Currently, Tesseract works best for certain types of fonts. In order to facilitate accurate readings, Tesseract requires clean segmentation of the foreground text from the background. The resulting segmentations are further constrained by the fact that they need to be of as high resolution as possible. Additionally, the characters in the input image should not appear pixelated post segmentation.

Lastly, the proposed model can be improved by expanding the number of base languages supported. The proposed model allows only English to Kannada translation and Tesseract provides OCR capabilities for other languages as well. Supporting document translation between multiple languages would be highly useful. Thus, future OCR engines that can overcome these shortcomings will be more effective as compared to the proposed model.

References

[1] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss and J. Han, "Automated Phrase Mining from Massive Text Corpora", IEEE Transactions on Knowledge and Data Engineering, vol. 30, pp. 1825 - 1837,Oct. 2018

[2] K.G. Prasad, H. Mathivanan, M. Jayaprakasam and T.V.Geetha, "Document Summarization and Information Extraction for Generation of Presentation Slides" in 2009 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala, India, 27-28 Oct. 2009

[3] N. M. Chidiac, P. Damien and C. Yaacoub, "A Robust Algorithm for Text Extraction from Images" in 2016 39th International Conference on Telecommunications and Signal Processing (TSP), Vienna, Austria, 27-29 Jun. 2016

[4] J. Yuan, Y. Zhang, K. K. Tan and T. H. Lee, "Text Extraction from Images Captured via Mobile and Digital Devices" in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, Singapore, 14-17 Jul. 2009

[5] C. Yi and Y. Tian, "Scene Text Recognition in Mobile Applications by Character Descriptor and Structure Configuration", IEEE Transactions on Image Processing, vol. 23, pp. 2972 - 2982, Jul. 2014

[6] Z. Zhu, X. Baojie and W. Guoxin, "Canny edge detection based on Open CV", 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 20-22 Oct. 2017

[7] B. Wang and S. S. Fan, "An Improved CANNY Edge Detection Algorithm" in Second International Workshop on Computer Science and Engineering, Qingdao, China, 28-30 Oct. 2009

[8] J. Chen, G. Bai, S. Liang and Z. Li, "Automatic Image Cropping: A Computational Complexity Study" in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 Jun. 2016

[9] C. Cao, J. Liu and Y. Zuo, "Automatic image cropping via the novel saliency detection algorithm" in IEEE 4th International Conference on Software Engineering and Service Science, Beijing, China, 23-25 May. 2013

[10] Y. L. Chen, T. W. Huang, K. H. Chang, Y. C. Tsai, H. T. Chen and B. Y. Chen, "Quantitative Analysis of Automatic Image Cropping Algorithms: A Dataset and Comparative Study" in IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24-31 Mar. 2017

[11] Y. Kao, R. He and K. Huang, "Automatic image cropping with aesthetic map and gradient energy map" in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5-9 Mar. 2017

[12] G. Guo, H. Wang, C. Shen, Y. Yan and H. Y. M. Liao, "Automatic Image Cropping for Visual Aesthetic Enhancement Using Deep Neural Networks and Cascaded Regression", IEEE Transactions on Multimedia, vol. 20, pp. 2073 - 2085, Aug. 2018

[13] W. Li, M. Breier and D. Merhof, "Skew correction and line extraction in binarized printed text images" in IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27-30 Sep. 2015

[14] M. Wagdy, I. Faye and D. Rohaya, "Document image skew detection and correction method based on extreme points" in International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 3-5 Jun. 2014

[15] M. Golpardaz and H. Nezamabadi-Pour, "Perspective Rectification and Skew Correction in Camera-Based Farsi Document Images", 7th Iranian Conference on Machine Vision and Image Processing, Tehran, Iran, 16-17 Nov. 2011

[16] C. Sun and. Si, "Skew and slant correction for document images using gradient direction" in Proceedings of the Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, Germany, 18-20 Aug. 1997

[17] Z. Junyou, "A Quickly Skew Correction Algorithm of Bill Image" in 2010 Third International Conference on Information and Computing, Wuxi, China, 4-6 Jun. 2010

[18] S. Malakar, B. Seraogi, R. Sarkar, N. Das, S. Basu and M. Nasipuri, "Two-stage skew correction of handwritten Bangla document images" in Third International Conference on Emerging Applications of Information Technology, Kolkata, India, 30 Nov.-1 Dec. 2012

[19] H. Y. Wang, R. Z. Zhao and J. A. Cui, "Fast and robust skew correction in scanned document images based on low-rank matrix decomposition" in International Conference on Machine Learning and Cybernetics, Lanzhou, China, 13-16 Jul. 2014

[20] L. Junjuan and T. Guoxin, "An Efficient Algorithm for Skew-Correction of Document Image Based on Cyclostyle Matching" in International Conference on Computer Science and Software Engineering, Hubei, China, 12-14 Dec. 2008

[21] Q. Yuan and C.L. Tan, "Text extraction from gray scale document images using edge information" in Proceedings of Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, USA, 13 Sep. 2001

[22] Antony P.J, Ajith V.P. and Soman K.P., "Kernel Method for English to Kannada Transliteration" in International Conference on Recent Trends in Information, Telecommunication and Computing, Kochi, Kerala, India, 12-13 Mar. 2010

[23] M. M. Kodabagi and S. A. Angadi, "A methodology for machine translation of simple sentences from Kannada to English language" in 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, India, 14-17 Dec. 2016

[24] S. Parameswarappa and V.N. Narayana, "Sense disambiguation of simple prepositions in English to Kannada Machine Translation" in International Conference on Data Science & Engineering (ICDSE), Cochin, Kerala, India, 18-20 Jul. 2012

[25] T. C. Wei, U. U. Sheikh and A. A. A. Rahman, "Improved optical character recognition with deep neural network", 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), Batu Feringghi, Malaysia, 9-10 Mar. 2018

[26] S. Thakare, A. Kamble, V. Thengne and U.R. Kamble, "Document Segmentation and Language Translation Using Tesseract-OCR" in 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS), Rupnagar, India, 1-2 Dec. 2018

[27] R. Smith, "An Overview of the Tesseract OCR Engine" in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007),Parana, Brazil, 23-26 Sept. 2007

[28] D. Berchmans and S. S. Kumar, "Optical character recognition: An overview and an insight" in International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, 10-11 Jul. 2014

[29] B. Dhruv, N. Mittal and M. Modi "Analysis of different filters for noise reduction in images" in Recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 26-27 Oct. 2017

[30] R. K. Pandey, S.R.Maiya and A. G Ramakrishnan, "A new approach for upscaling document images for improving their quality" in 14th IEEE India Council International Conference (INDICON), Roorkee, India, 15-17 Dec. 2017

[31] K. He, X. C. Luan, C. H. Li and R. Lui, "Gaussian Noise Removal of Image on the Local Feature" in Second International Symposium on Intelligent Information Technology Application, Shanghai, China, 20-22 Dec. 2008

[32] N. U. Khan, K.V. Arya and M. Pattanaik, "An efficient image noise removal and enhancement method" in IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, Istanbul, Turkey, Istanbul, Turkey

[33] C. Yi, "An English POS Tagging Approach Based on Maximum Entropy" in International Conference on Intelligent Transportation, Big Data and Smart City, Halong Bay, Vietnam,19-20 Dec. 2015

[34] X. Han, Y. Dang, L. Mei, Y. Wang, S. Li and X. Zhou, "A Novel Part of Speech Tagging Framework for NLP Based Business Process Management" in 2019 IEEE International Conference on Web Services (ICWS), Milan, Italy, 8-13 Jul. 2019

[35] M. Zhang, N. Yu and G. Fu, "A Simple and Effective Neural Model for Joint Word Segmentation and POS Tagging", IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, pp. 1528 - 1538, Sep. 2018

[36] D. N. Mati, M. Hamiti, J. Ajdari, B. Selimi, and B. Raufi, "A Systematic Mapping Study of Language Features Identification from Large Text Collection" in 8th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 10-14 Jun. 2019

[37] Ankita and K. A. Abdul Nazeer, "Part-of-speech Tagging and Named Entity Recognition Using Improved Hidden Markov Model and Bloom Filter" in International Conference on Computing, Power and Communication Technologies, Greater Noida, Uttar Pradesh, India, 28-29 Sep. 2018

[38] Kemal Kurniawan and Alham Fikri Aji, "Toward a Standardized and More Accurate Indonesian Part-of-Speech Tagging" in International Conference on Asian Language Processing (IALP), Bandung, Indonesia,15-17 Nov. 2018

[39] L. Q. Zuo, H. M. Sun, Q. C. Mao, R. Qi and R. S. Jia, "Natural Scene Text Recognition Based on Encoder-

Decoder Framework", IEEE Access, vol. 7, pp. 62616 - 62623, May. 2019

[40] A. Pawar, S. Du, V. Mago and H. Kaur Janda, "Syntactic, Semantic and Sentiment Analysis: The Joint Effect on Automated Essay Evaluation", IEEE Access, vol. 7, pp. 108486 - 108503, Aug. 2019

[41] J. L. Wu and W. Y. Ma, "A Deep Learning Framework for Coreference Resolution Based on Convolutional Neural Network" in IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 Jan.-1 Feb. 2017

[42] Y. Huang, A. Li, J. Huang, L. Lan, X. Yin and Y. Jia, "Person Entity Attribute Extraction Based on Siamese Network", IEEE Access, vol. 7, pp. 64506 - 64516, May. 2019

[43] S. Joty, G. Carenini, R. Ng and G. Murray, "Discourse Processing and Its Applications in Text Mining" in 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17-20 Nov. 2018

[44] P. Gamallo, M. Garcia, C. Pineiro, R. Martinex-Costani and J. C. Pichel, "LinguaKit: a Big Data-based multilingual tool for linguistic analysis and information extraction" in 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), Valencia, Spain, 15-18 Oct. 2018

[45] P. Das, A. K. Das, J. Nayak, D. Pe5.lusi and W. Ding, "A Graph based Clustering Approach for Relation Extraction from Crime Data", IEEE Access, vol. 7, pp. 101269 - 101282, Jul. 2019

[46] J. Xu, H. He, X. Sun, X. Ren and S. Li, "Cross-Domain and Semi-Supervised Named Entity Recognition in Chinese Social Media: A Unified Model", IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, pp. 2142 - 2152, Jul. 2018

[47] T. Tran and D. T. Nguyen, "Text Generation from Abstract Semantic Representation for Summarizing Vietnamese Paragraphs Having Co-references" in 5th NAFOSTED Conference on Information and Computer Science (NICS), Ho Chi Minh City, Vietnam, 23-24 Nov. 2018

[48] H. Li, J. Zhu, C. Ma, J. Zhang and C. Zong, "Read, Watch, Listen and Summarize: Multi-modal Summarization for Asynchronous Text, Image, Audio and Video", IEEE Transactions on Knowledge and Data Engineering, vol. 31, pp. 996 - 1009, May. 2019

[49] S. García-Méndez, M. Fernández-Gavilanes, E. Costa-Montenegro, J. Juncal-Martínez, F. J. González-Castaño and E. Reiter, "A System for Automatic English Text Expansion", IEEE Access, vol. 7, pp. 123320 - 123333, Aug. 2019

[50] Shivakumar K.M, Shivaraju N, V. Sreekanta and D. Gupta, "Comparitive Study of Factored SMT with Baseline SMT for English to Kannada", 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26-27 Aug. 2016

[51] M. Bansal and G. Jain, "Improvement of English-Hindi Machine Translation using ConceptNet", 2017 recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 26-27 Oct. 2017

[52] K. K. Yadav and U. C. Jaiswal, "A Survey Paper on Performance Improvement of Word Alignment in English to Hindi Translation System", 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, 23-24 June 2017

[53] J. Nair, A. Krishnan, R. Deetha, "An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism", 2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, Sept. 21-24, 2016.

[54] P. Kumar, S. Srivastava and M. Joshi, "Syntax directed translator for English to Hindi language", 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India, 20-22 Nov. 2015

[55] M. Kasthuri and S. B. R. Kumar, "Rule Based Machine Translation System from English to Tamil", 2014 World Congress on Computing and Communication Technologies, Trichirappalli, India, 27 Feb.-1 March 2014

[56] S. Rajpirathap, S. Sheeyam, K. Umasuthan and A. Chelvarajah, "Real-time direct translation system for Sinhala and Tamil languages", 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, 13-16 Sept. 2015

[57] V. Goyal and G. S. Lehal, "Hindi to Punjabi Machine Translation System", International Conference on Information Systems for Indian Languages, vol. 139, pp 236-241. Springer, Berlin, Heidelberg

[58] K. A. Patel and J. S. Pareek, "GH-MAP: translation system for sibling language pair Gujarati-Hindi", CSI Transactions on ICT, vol. 1, pp. 183-192, June 2013

[59] Z. Islam, J. Tieddemann and A. Eisele, "English to Bangla Phrase-Based Machine Translation", EAMT, St. Raphael, France, May 2010

[60] R. Narayan, V. P. Singh and S. Chakraverty, "Quantum Neural Network Based Machine Translator for Hindi to English", The Scientific World Journal, vol. 2014, 27 Feb. 2014

[61] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting Text in Natural Scenes with stroke width transform," IEEE Conf. Comput. Vis. Pattern Recognit, no. d, pp. 2963–2970, 2010.

[62] V. Yadav and N. Ragot, "Text Extraction in Document Images: Highlight on Using Corner Points", 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11-14 April 2016

[63] Wahyono, M. Jeong, and J. Kang-Hyun, "Multi Language Text Detection Using Fast Stroke Width Transform," 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV), Jan. 2015.

[64] G. Sahoo, T. Kumar, B. L. Raina and C. M. Bhatial, "Text extraction and enhancement of binary images using cellular automata", International Journal of Automation and Computing, Aug. 2009

[65] R. Islam, Md. R. Islam and K. H. Talukder, "An approach to extract text regions from scene image", 2016 International Conference on Computing, Analytics and Security Trends (CAST), Pune, India, 19-21 Dec. 2016

[66] A. Park, K. Jung. (2007) Automatic Word Detection System for Document Image Using Mobile Devices. In: Jacko J.A. (eds) Human-Computer Interaction. Interaction Platforms and Techniques. HCI 2007. Lecture Notes in Computer Science, vol 4551. Springer, Berlin, Heidelberg

[67] R. Smith "Hybrid Page Layout Analysis via Tab-Stop Detection" in 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26-29 Jul, 2009.

[68]S. Chernenko. "Free Google Translate API" Available: https://github.com/sergei4e/gtrans [Accessed Apr. 22, 2019]

[69] "OpenCV Documentation", Jun, 2000. [Online]. Available: https://docs.opencv.org/master/. [Accessed: Mar. 12, 2019].

Authors –

Chirag Kamath is pursuing his Bachelor of Engineering degree and is a final year student at BMSCE, Department of Information Science and Engineering. He is currently an Associate Software Engineering Intern at Sabre Corp and has previous industry experience as a summer intern at IBM. His research interests include AI & image processing.

Neha Prakash is currently pursuing her Bachelor of Engineering degree at BMSCE, Department of Information Science and Engineering. She is currently working at Clumio Inc as a Software Engineering Intern. She has previously interned at American Express.

Dr Radhika K R is a professor at B.M.S College of Engineering. She has an experience of 23 years in teaching a wide area of subjects in Information Science Department at BMSCE. She received her Phd in 2010 and Mtech in 2000 from Vishesvaraya and Bangalore University respectively. She has 40+ publications in various reputed journal. She is a senior member of IEEE. Her area of interests is network security, data mining, cloud security and Biometrics.