

KanOCR: Conversion of Printed Kannada Document to Editable form using Convolutional Neural Networks

Pradyumna Mukunda
PESIT, Bangalore

Niraj S. Prasad
PESIT, Bangalore

Mamatha H. R.
Professor, CS Dept, PESIT

ABSTRACT

Optical Character Recognition (OCR) technology in converting an image containing text to an editable text format is of high sense in document image processing. Input to OCR could be a scanned document, or a simple newspaper cut-out. Supervised Learning using Neural Networks yield the output with greater accuracy. Unlike English, Kannada Language has a huge set of characters as it includes kaagunithas, vattaksharas, etc. This makes recognition of the characters much more complex. The paper mainly concentrates on OCR for the Kannada Text which goes through a threshold as a first step converting input image into binary image, making segmentation easier. Characters can be extracted from the documents using various Segmentation methods. The vattaksharas are extracted/differentiated from the words by using base-line technique. When the characters are recognized, they are compared with Unicodes available on the system and then printed. In the above method, CNN plays a pivotal role in reading the character and comparing it with the Unicode look up table values to print the output.

This system has been tested with varying fonts. A total number of 37 sample documents are used for experimentation. The system has been developed for only printed Kannada Text.

General Terms

Optical Character Recognition, CNN.

Keywords

Base-line Identification, CNN, Kannada, Neural Network, Optical Character Recognition, Pre-processing, Python, Segmentation.

1. INTRODUCTION

Optical Character Recognition is basically an electronic or mechanical conversion of image text into editable text format. The image could be hand-written, a scanned document, a photograph with some content, or any printed text. Digitizing printed text is essential in editing and storing valuable information on a software platform.

Demand for a stable regional language OCR system has grown over the past few years. With the advance in technology, a smartphone can now read any text on an image captured.

Using OCR systems, end users now edit documents more efficiently. Digital text data can be stored and saved without much risk of loss, unlike a physical document. Applications of OCR has a wide range one among is Digi-locker. In case of a change in information, scanning and editing a document is currently practiced in a lot of places. Scanning and editing a file is another application.

Digitization and retaining information of the printed or handwritten documents for the future usage on the storage

device is of high demand, as the data could be used for further processing. Usual method of storage is just to scan and store, but cannot be reutilized for correction and processing and also challenging to read or modify text or other information from these image files. Therefore, a method to automatically retrieve and store information, in the form of text, from image files is needed. This is where Optical Character Recognition comes into play.

The paper is organized as follows. Section II is a background study on OCR systems. Section III describes the methodology and implementation of the project. Section IV shows the results and analysis. Finally Section V is the conclusion and references

2. BACKGROUND STUDY

Types of OCR Systems:

There are different types of OCR systems, such as:

- OCR which targets typed text one character at a time.
- Optical Word Recognition (OWR) which focuses on typed text one word at a time. This is possible only for languages that use a word-divider or a space.
- Intelligent Character Recognition (ICR) that involves Machine Learning, ICR targets one sentence at a time.
- Intelligent Word Recognition (IWR) that involves Machine Learning. IWR works for handwritten print scripts really well. Cursive text and those glyphs that are not separated in cursive works really well with IWR.

Offline OCR analyses a static document where online handwriting analysis can be used for hand-written text to generate the output in real-time. This process is called 'Dynamic Character Recognition' or 'ICR', or 'Real-time character Recognition.'

Techniques used in OCR:

Pre-processing:

OCR implements preprocessing techniques on input image to achieve accuracy in end system. To get a successful output, the following techniques are considered –

1. *De-Skew*: The image might be tilted, or the text image might not be aligned perfectly, some de-skewing methods are applied before OCR
2. *De-speckle*: The system smoothen the image and removes the negative and positive spots of the picture. This ensures greater accuracy.
3. *Binarization*: In case the input is a color image, binarization to convert the image to binary image

and then performs OCR.

4. **Line and Word Detection:** Separating line and word helps in easy segmentation. As a preliminary step identifying the baseline of words helps in line segmentation.
5. **Script Recognition:** Document might have different scripts. The system must be able to realize script in document.
6. **Normalizing the scale and the aspect-ratio.**
7. **Character isolation and segmentation:** this is useful in Character OCR. Words are broken down into many characters, hence making reading smooth and accurate.

Character Recognition:

There are currently two types of the core OCR algorithm.

Matrix Matching is a technique wherein a stored glyph is compared with the image. It is also called pattern recognition and image correlation. This technique works best with typewritten text. It does not work well when a different font is encountered.

Feature Matching breaks down the glyphs into lines, lops, line-intersections, and line directions. These features are compared with the abstract vector-like representation of the character. Feature detection in Computer Vision is applicable to this type of OCR System. This method uses Machine Learning and is the most modern OCR Software. In this paper, an entirely different technique in the form of CNN has been used.

Specific software such as Tesseract and Coneiform uses a two pass approach to character recognition. This is useful when the font is distorted, blurred, or faded.

Post Processing:

A dictionary can be added to check the accuracy of the word, and it can be processed with maximum efficiency possible. The output stream can be a plain text file of characters. Sophisticated OCR systems also preserve the layout and font perfectly. In order to further optimize results from an OCR API in post-processing, the Levenshtein Distance algorithm can be used.

Convolutional Neural Networks:

In a fully connected neural network, every neuron takes inputs from each and every neuron in the previous layer and feeds output to each and every neuron in the next layer. In other words, every possible connection between two neurons in two different layers is made. This architecture of neural network does not scale well for images. Firstly, the images would require a lot of manually engineered pre-processing to obtain an accurate result. Secondly, the size of the input is very large. The use of Convolutional Neural Networks for images reduces the complexity of the problem because they are constructed such that each neuron processes data from only a restricted area of the image known as the neuron's receptive field, and not the entire image. Using CNNs also greatly reduces the amount of manual pre-processing needed for the image. CNNs are classified as deep learning networks having number of hidden layers.

Unicode:

Unicode is defined by Wikipedia as “a computing industry standard for the consistent encoding, representation, and

handling of text expressed in most of the world's writing systems.” Unicode was developed when 8-bit encoding systems such as ASCII were still popular. Since ASCII can hold only 256 characters, only Roman characters are represented.

Unicode uses 16 bits (specifically UTF-16 uses 16 bits), and can hold 65,536 characters, which is way more than enough to represent characters in all of the world's living languages, as well as historic scripts such as Brahmi.

UTF-16 assigns each of its characters with a unique 16-bit identification number known as a code point, and leaves the rendering of the character to the software. The code points for Kannada characters are in the range of 0x0C82 to 0x0CF2. This range of code points is reserved exclusively for Kannada characters.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0C8x	□	◌̣	◌̤	◌̥	◌̦	ಅ	ಆ	ಇ	ಈ	ಉ	ಊ	ಋ	ೠ	ಎ	ಐ	ಓ
U+0C9x	ಐ	ಋ	ಌ	಍	ಐ	ಕ	ಖ	ಗ	ಘ	ಙ	ಚ	ಛ	ಜ	ಝ	ಞ	ಟ
U+0CAx	ಠ	ಡ	ಢ	ಣ	ತ	ಥ	ದ	ಧ	ನ		ಪ	ಫ	ಬ	ಭ	ಮ	ಯ
U+0CBx	ರ	ಱ	ಲ	ಳ		ವ	ಶ	ಷ	ಸ	ಹ			಼	ಽ	ಠ	ತಿ
U+0CCx	ದಿ	ೃ	ೄ	೅	ೆ	ೇ	ೈ	೉	ೊ	ೋ	ೌ	್	೎	೏		
U+0CDx						೐	೑								ಱ	
U+0CEx	ಫಲಿತ	ಫಲಿತ	ಫಲಿತ	ಫಲಿತ		೒	೓	೔	ೕ	ೖ	೗	೘	೙	೚	೛	೜
U+0CFx		ೝ	ೞ													

Figure 1: Unicode for Kannada characters (Source: Wikipedia)

Applications of OCR:

- Data Entry for passports, invoices and other business documents.
- Automatic Number Plate Recognition.
- Key information extraction from insurance documents.
- Making electronic images of printed documents searchable.
- Giving instructions to computers by writing.
- Assistive technology for the blind and visually impaired users.
- Convert printed text-books into editable e-Books easily.

2.1 LITERATURE SURVEY

A significant amount of research has been done on Kannada Character Recognition:

A method of identifying Kannada characters by feature extraction using Curvelet transform and standard deviation followed by a KNN classifier is described in [1]. The system works on offline handwritten Kannada character recognition. Different methods of segmenting lines in a paragraph of Kannada text, such as HPP based segmentation, Morphology based segmentation and Bounding Box based segmentation, are discussed in [2].

Segmentation of overlapping text lines in Indian printed scripts are discussed in [3]. The overlapping of lines of text causes a lot of difficulty in implementing OCR. A method of identifying characters by zoning, followed by feature

extraction by dividing segments into tracks and sectors (like in a compact disc), then an SVM classifier, is described in [4]. A number of methods for feature extraction for the purpose of character recognition are discussed in [5].

The paper [6] discusses and compares different methods of segmentation, feature extraction and classification with respect to the accuracy of the OCR obtained. A method of identifying Kannada characters by zoning, feature extraction using Hu's moments, HPP and VPP, followed by a PNN classifier is described in [7].

Handwriting recognition system for the Kannada character by extracting various features, namely-Hu's Invariant moments, Zernike moments, Zonal features, Fourier-Wavelet coefficients and the recognition process is carried out using Back Propagation Neural Network in [8].

The two fold cross validation SVM classifier is applied for classification of handwritten Kannada characters. These features are served to SVM classifier for classification of character images. The performance of the proposed algorithm are tested with two fold cross validation. An average recognition accuracy of 90.09% is obtained for Kannada vowels and consonants respectively.[9]

Online Handwriting Recognition (HWR) of Kannada includes methods like data-collection, feature-extraction normalization, classification, training-sample creation and character recognition. Online handwriting recognition of Kannada characters by combining Direction based Stroke Density principle (DSD) with K Neural Network (KNN). Kannada Language with a sample of 20 different handwritings results in an accuracy of 94.4%. [10]

In this paper, Kannada online handwritten character recognizer which is viable for real time applications is implemented. Proposed system is implemented on mobile device using two different approaches namely Principal Component Analysis (PCA) and Dynamic Time Wrapping (DTW). Results of PCA approach is quite capable than DTW. On an average, recognition accuracy up to 88% is achieved for the PCA approach and up to 64% is achieved for DTW approach. [11]

Based on the zonal features, recognition system for handwritten and printed Kannada numerals and vowels are proposed. The Kannada numerals and vowels are circular in nature. Pixel density feature is potential features for handwritten and printed Kannada numerals and vowels. SVM classifier with fivefold cross validation test is applied on a mixture of handwritten and printed Kannada numerals and vowels respectively and an average percentage of recognition accuracy of 97.40% and 95.90% is obtained. [12]

2.2 Existing OCR Systems for Kannada Language:

There are only a few notable OCR systems for the Kannada Language. KanScan is an app that converts a Kannada text image into editable text format. It contains a lot of flaws and gives a lot of errors when it runs. Another system is the i2OCR that offers an accuracy of around 60% with a run-time of almost 1 whole minute for a 200-word article. The links for the same are given below.

KanScan google play store link;

<https://play.google.com/store/apps/details?id=com.kaleidosoftware.kanscan.free>

i2OCR download link;

www.i2ocr.com/free-online-kannada-ocr

Hence, it can be seen that the need for a top-notch Kannada OCR system is essential for the masses. Use of CNN gets us the highest possible accuracy in this system.

3. METHODOLOGY AND IMPLEMENTATION

A. System Architecture:

The sample image is the printed Kannada text given as the input to the system. The Pre-Processing block helps with the segmentation of words and characters. Size-based classification too occurs. 4 CNN's are trained with varied input sizes of the image. The dataset consists of twenty fonts overall. A UID table is then constructed to ensure the Unicode is matched. The output is then compared by the system with the UID table to get the editable text file.

B. Pre Processing

The pre-processing block consists of segmentation of characters. The algorithm goes as follows:

1. First trim out the extra image pixels around the word.
2. Next identify the position of the base line, which will be used later on in the process.
3. Then perform thresholding to differentiate foreground pixels and background pixels.
4. Next dilate the image just enough so that every character becomes a single connected component
5. Then label each of the connected components from left to right.
6. Using the labels as reference, crop out each character individually.
7. Identify each character as a regular character or vattakshara character using the position of the base line.
8. Finally feed the character to the appropriate set of CNNs for identification.

A demonstration of the segmentation is shown below:

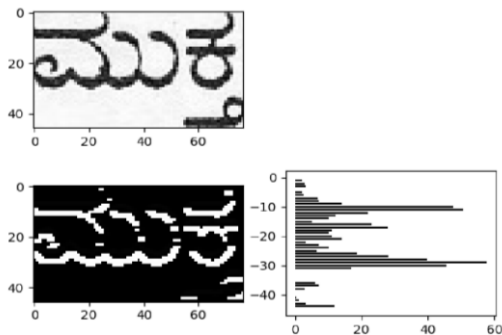


Figure 2. Sample of Segmentation

C. Baseline Identification

Baseline is the imaginary horizontal line on which the non-subscript text "rests" upon. To find the position of this line, the following method is used:

1. First apply thresholding to the input image of the word
2. This is followed by Sobel edge detection to get all the **lower edges** in the image.
3. Then take the HPP (Horizontal Point Projection) of the edge detection result.
4. When the HPP is plotted on a graph two maxima, upper and lower, are obtained. The position of the lower maxima is taken as the position of the base line.

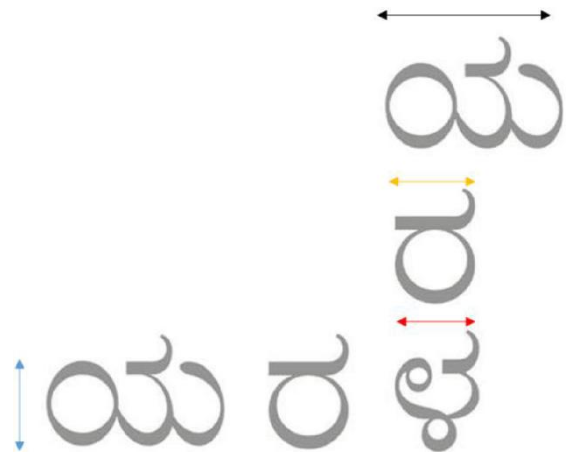


This line is used to differentiate between regular characters and subscript characters.

To create the dataset the first step is to list out all possible Kannada glyphs in a Microsoft Word document.

Figure 4. List of Kannada glyphs

Figure 5. Kannada Characters depicting different fonts



E. Design and training of CNNs

The CNN implemented is a Keras Sequential model beginning with one convolutional layer of kernel size 5x5 with 64 channels, followed by a ReLU activation function layer, followed by a 2x2 max-pooling layer. Next all values are flattened into a single dimension. This becomes an input for a fully connected layer of 1000 hidden neurons with ReLU

activation function and finally a fully connected output layer of 340 neurons with softmax activation function.

```
Using TensorFlow backend.
x_train shape: (2188, 15, 20, 1)
2188 train samples
2188 test samples
Epoch 1/15
2018-04-27 16:15:36.723768: I T:\src\github\tensorflow\tensorflow\core\platform\cpu_fe
2188/2188 [=====] - 6s 3ms/step - loss: 4.6342 - acc: 0.1120
Epoch 2/15
2188/2188 [=====] - 6s 3ms/step - loss: 2.0110 - acc: 0.5603
Epoch 3/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.8301 - acc: 0.7797
Epoch 4/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.4604 - acc: 0.8853
Epoch 5/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.2770 - acc: 0.9250
Epoch 6/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.1610 - acc: 0.9575
Epoch 7/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.1043 - acc: 0.9712
Epoch 8/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0722 - acc: 0.9799
Epoch 9/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0698 - acc: 0.9803
Epoch 10/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0586 - acc: 0.9858
Epoch 11/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0709 - acc: 0.9831
Epoch 12/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0400 - acc: 0.9909
Epoch 13/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0300 - acc: 0.9936
Epoch 14/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0256 - acc: 0.9922
Epoch 15/15
2188/2188 [=====] - 6s 3ms/step - loss: 0.0432 - acc: 0.9913
Test loss: 0.01144077842569888
Test accuracy: 0.9990859232175503
>>>
```

Figure 7. Demonstration of CNN training

The loss function used for training is categorical cross-entropy and the optimization algorithm used is the Adam optimizer, explained in detail in the paper *Adam: A Method for Stochastic Optimization*. [12][13]

Table 1. Lookup Table

Kannada	UID	Hex0	Hex1	Char0	Char1
ಂ		0 C82		0 3202	0
ೇ		1 CD5		0 3285	0
ಕ		2 CB0	CCD	3248	3277
ಃ		3 C85		0 3205	0
಼		4 C86		0 3206	0
ಽ		5 C87		0 3207	0
ಫ		6 C88		0 3208	0
ಬ		7 C89		0 3209	0
ಬ		8 C8A		0 3210	0
ಬ		9 C8B		0 3211	0
ಬ		10 C8E		0 3214	0
ಬ		11 C8F		0 3215	0
ಬ		12 C90		0 3216	0
ಬ		13 C92		0 3218	0
ಬ		14 C93		0 3219	0
ಬ		15 C94		0 3220	0
ಕ		16 C95		0 3221	0
ಕ		17 C95	CBE	3221	3262
ಕ		18 C95	CBF	3221	3263
ಕ		19 C95	CC1	3221	3265
ಕ		20 C95	CC2	3221	3266
ಕ		21 C95	CC6	3221	3270
ಕ		22 C95	CCA	3221	3274
ಕ		23 C95	CCC	3221	3276
ಕ		24 C95	CCD	3221	3277
ಬ		25 C96		0 3222	0
ಬ		26 C96	CBE	3222	3262
ಬ		27 C96	CBF	3222	3263

After using the CNN to identify the character and obtaining its UID the UID is queried into a Lookup Table as above to obtain the Unicode encoding of the respective character for printing.

F. Vattakshara characters

The vattakshara i.e. subscript characters are excluded from the main dataset, because most of them look very similar to their regular character counterparts and therefore did not want to confuse the CNNs and hence reduce the accuracy of the system

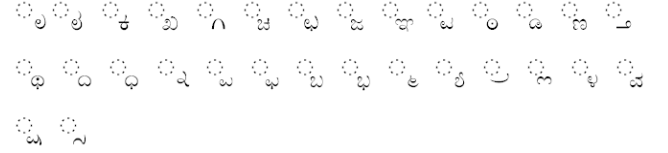


Figure 8. Vattakshara characters in Kannada

Instead a separate set of CNNs and a separate Unicode lookup table is used just for the identification of vattakshara characters. The entire process of creating the dataset, training the CNNs and constructing the lookup table is repeated. In this case, only 3 CNNs is used, with input sizes 15x15, 20x15 and 30x15. A character is identified as a regular character or vattakshara character based on the position of the character with respect to the base line.

4. RESULTS AND ANALYSIS

A sample image containing Kannada text was taken from an e-paper for checking the accuracy of the system.

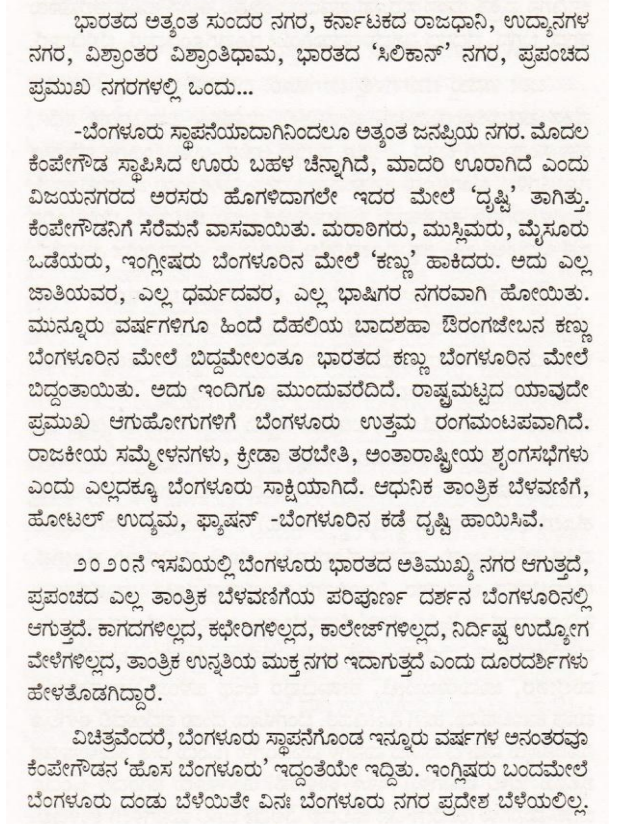


Figure 9. Sample image for test in Kannada

The sample image was tested with KanOCR. The Editable Output Text is as shown below. The errors have been highlighted manually.

KanOCR Result:

ಭಾರತದ ಅತ್ಯಂತ ಸುಂದರ ನಗರ, ಕರ್ನಾಟಕದ ರಾಜಧಾನಿ, ಉದ್ಯಾನಗಳ ನಗರ, ವಿಶ್ರಾಂತರ ವಿಶ್ರಾಂತಿಧಾಮ, ಭಾರತದ ಫಿಲಿಕಾನ್, ನಗರ, ಪ್ರಪಂಚದ ಪ್ರಮುಖ ನಗರಗಳಲ್ಲಿ ಒಂದು.

ಬೆಂಗಳೂರು ಸ್ಥಾಪನೆಯಾದಾಗಿನಿಂದಲೂ ಅತ್ಯಂತ ಜನಪ್ರಿಯ ನಗರ. ಮೊದಲ ಕೆಂಪೇಗೌಡ ಸ್ಥಾಪಿಸಿದ ಊರು ಬಹಳ ಚೆನ್ನಾಗಿದೆ, ಮಾದರಿ ಊರಾಗಿದೆ ಎಂದು ವಿಜಯನಗರದ ಅರಸರು ಹೊಗಳಿದಾಗಲೇ ಇದರ ಮೇಲೆ ದೃಷ್ಟಿ, ತಾಗಿತ್ತು. ಕೆಂಪೇಗೌಡನಿಗೆ ಸೆರೆಮನೆ ವಾಸವಾಯಿತು. ಮರಾಠಿಗರು, ಮುಸ್ಲಿಮರು, ಮೈಸೂರು ಒಡೆಯರು, ಇಂಗ್ಲೀಷರು ಬೆಂಗಳೂರಿನ ಮೇಲೆ ಹೆಚ್ಚು ಹಾಕಿದರು. ಅದು ಎಲ್ಲ ಜಾತಿಯವರ, ಎಲ್ಲ ಧರ್ಮದವರ, ಎಲ್ಲ ಭಾಷಿಗರ ನಗರವಾಗಿ ಹೋಯಿತು. ಮುಸ್ಲಿಮರು ವರ್ಷಗಳಿಗೂ ಹಿಂದೆ ದೆಹಲಿಯ ಬಾದಶಹಾ ಔರಂಗಜೇಬನ ಕಣ್ಣು ಬೆಂಗಳೂರಿನ ಮೇಲೆ ಬಿದ್ದಂತಾಯಿತು ಅದು ಇಂದಿಗೂ ಮುಂದುವರಿದಿದೆ ರಾಷ್ಟ್ರಮಟ್ಟದ ಯಾವುದೇ ಪ್ರಮುಖ ಆಗುಹೋಗುಗಳಿಗೆ ಬೆಂಗಳೂರು ಉತ್ತಮ ರಂಗಮಂಟಪವಾಗಿದೆ ರಾಜಕೀಯ ಸಮ್ಮೇಳನಗಳು, ಕ್ರೀಡಾ ತರಬೇತಿ, ಅಂತಾರಾಷ್ಟ್ರೀಯ ಶೃಂಗಸಭೆಗಳು ಎಂದು ಎಲ್ಲದಕ್ಕೂ ಬೆಂಗಳೂರು ಸಾಕ್ಷಿಯಾಗಿದೆ ಆಧುನಿಕ ತಾಂತ್ರಿಕ ಬೆಳವಣಿಗೆ, ಹೋಟೆಲ್ ಉದ್ಯಮ, ಪ್ರಾಣಿ -ಬೆಂಗಳೂರಿನ ಕಡೆ ದೃಷ್ಟಿ ಹಾಯಿಸಿವೆ

೨೦೦ ೨೦೦ನೆ ಇಸವಿಯಲ್ಲಿ ಬೆಂಗಳೂರು ಭಾರತದ ಅತಿಮುಖ್ಯ ನಗರ ಆಗುತ್ತದೆ, ಪಪಂದ ಎಲ್ಲ ತಾಂತ್ರಿಕ ಬೆಳವಣಿಗೆಯ ಪರಿಪೂರ್ಣ ದರ್ಶನ ಬೆಂಗಳೂರಿನಲ್ಲಿ ಆಗುತ್ತದೆ ಕಾಗದಗಳಿಲ್ಲದ, ಕಛೇರಿಗಳಿಲ್ಲದ, ಕಾಲೇಜುಗಳಿಲ್ಲದ, ನಿರ್ದಿಷ್ಟ ಉದ್ಯೋಗ ವೇಳೆಗಳಿಲ್ಲದ, ತಾಂತ್ರಿಕ ಉನ್ನತಿಯ ಮುಕ್ತ ನಗರ ಇದಾಗುತ್ತದೆ ಎಂದು ದೂರದರ್ಶಿಗಳು ಹೇಳತೊಡಗಿದಾರೆ

ವಿಚಿತವೆಂದರೆ, ಬೆಂಗಳೂರು ಸ್ಥಾಪನೆಗೊಂಡ ಇನ್ನೂರು ವರ್ಷಗಳ ಅನಂತರವೂ ಕೆಂಪೇಗೌಡನ ಫಿಲಿಕಾನ್ ಬೆಂಗಳೂರು, ಇದ್ದಂತೆಯೇ ಇದ್ದಿತು ಇಂಗ್ಲೀಷರು ಬಂದಮೇಲೆ ಂಗಳೂರು ದಂಡು ಬೆಳೆಯಿತೇ ವಿನಃ ಬೆಂಗಳೂರು ನಗರ ಪ್ರದೇಶ ಬೆಳೆಯಲಿಲ್ಲ

Same sample image was tested with i2OCR. Result is listed below:

i2OCR Result:

ಭಾರತದ ಅತ್ಯಂತ ಸುಂದರ ನಗರ, ಕರ್ನಾಟಕದ ರಾಜಧಾನಿ, ಉದ್ಯಾನಗಳ ನಗರ, ವಿಶ್ರಾಂತರ ವಿಶ್ರಾಂತಿಧಾಮ, ಭಾರತದ "ಸಿಲಿಕಾನ್ ನಗರ, ಪ್ರಪಂಚದ ಪ್ರಮುಖ ನಗರಗಳಲ್ಲಿ ಒಂದು...

=ಬೆಂಗಳೂರು ಸ್ಥಾಪನೆಯಾದಾಗಿನಿಂದಲೂ ಅತ್ಯಂತ ಜನಪ್ರಿಯ ನಗರ. ಮೊದಲ ಕೆಂಪೇಗೌಡ ಸ್ಥಾಪಿಸಿದ ಊರು ಬಹಳ ಚೆನ್ನಾಗಿದೆ, ಮಾದರಿ ಊರಾಗಿದೆ ಎಂದು ವಿಜಯನಗರದ ಅರಸರು ಹೊಗಳಿದಾಗಲೇ ಇದರ ಮೇಲೆ "ದೃಷ್ಟಿ ತಾಗಿತ್ತು. ಕೆರಿಪೆಗೌಡನಿಗೆ ಸೆರೆಮನೆ ವಾಸವಾಯಿತು. ಮರಾಠಿಗರು, ಮುಸ್ಲಿಮರು, ಮೈಸೂರು ಒಡೆಯರು, ಇಂಗ್ಲೀಷರು ಬೆಂಗಳೂರಿನ ಮೇಲೆ "ಕಣ್ಣು ಹಾಕಿದರು. ಅದು ಎಲ್ಲ ಜಾತಿಯವರ, ಎಲ್ಲ ಧರ್ಮದವರ, ಎಲ್ಲ ಭಾಷಿಗರ ನಗರವಾಗಿ ಹೋಯಿತು. ಮುಸ್ಲಿಮರು ವರ್ಷಗಳಿಗೂ ಹಿಂದೆ ದೆಹಲಿಯ ಬಾದಶಹಾ ಔರಂಗಜೇಬನ ಕಣ್ಣು ಬೆಂಗಳೂರಿನ ಮೇಲೆ

ಬಿದ್ದಂತಾಯಿತು. ಅದು ಇಂದಿಗೂ ವಬಂದುವರೆದಿದೆ. ರಾಷ್ಟ್ರಮಟ್ಟದ ಯಾವುದೇ ಪ್ರಮುಖ ಆಗುಹೋಗುಗಳಿಗೆ ಬೆಂಗಳೂರು ಉತ್ತಮ ರಂಗಮಂಟಪವಾಗಿದೆ.

ರಾಜಕೀಯ ಸಮ್ಮೇಳನಗಳೆ, ಕ್ರೀಡಾ ತರಬೇತಿ, ಅಂತಾರಾಷ್ಟ್ರೀಯ ಶೃಂಗಸಭೆಗಳೆ ಎಲ್ಲದಕ್ಕೂ ಬೆಂಗಳೂರು ಸಾಕ್ಷಿಯಾಗಿದೆ. ಆಧುನಿಕ ತಾಂತ್ರಿಕ ಬೆಳವಣಿಗೆ, ಹೋಟೆಲ್ ಉದ್ಯಮ, ಪ್ರಾಣಿ =ಬೆಂಗಳೂರಿನ ಕಡೆ ದೃಷ್ಟಿ ಹಾಯಿಸಿವೆ.

೨೦೨೦ನೆ ಇಸವಿಯಲ್ಲಿ 'ಬೆಂಗಳೂರು ಭಾರತದ ಅತಿಮುಖ್ಯ ನಗರ ಆಗುತ್ತದೆ, ಪ್ರಪಂಚದ ಎಲ್ಲ ತಾಂತ್ರಿಕ ಬೆಳವಣಿಗೆಯ ಪರಿಪೂರ್ಣ ದರ್ಶನ ದೆರಿಗಳಪುರಿನಲ್ಲಿ ಆಗುತ್ತದೆ. ಕಾಗದಗಳಿಲ್ಲದ, ಕಛೇರಿಗಳಿಲ್ಲದ, ಕಾಲೇಜುಗಳಿಲ್ಲದ, ನಿರ್ದಿಷ್ಟ ಉದ್ಯೋಗ ವೇಳೆಗಳಿಲ್ಲದ, ತಾಂತ್ರಿಕ ಉನ್ನತಿಯ ಮುಕ್ತ ನಗರ ಇದಾಗುತ್ತದೆ ಎಲ್ಲದೂ ದೂರದರ್ಶಿಗಳು

ಹೇಳತೊಡಗಿದಾರೆ.

ಏಚಿತವೆಂದರೆ, ಬೆಂಗಳೂರು ಸ್ಥಾಪನೆಗೊಂಡ ಇನ್ನೂರು ವರ್ಷಗಳ ಅನಂತರವೂ ಕೆಂಪೇಗೌಡನ "ಹೊಸ ಬೆಂಗಳೂರು" ಇದ್ದಂತೆಯೇ ಇದ್ದಿತು. ಇಂಗ್ಲೀಷರು ಬಂದಮೇಲೆ ಬೆಂಗಳೂರು ದಂಡು ಬೆಳೆಯಿತೇ ಎನಃ ಬೆಂಗಳೂರು ನಗರ ಪ್ರದೇಶ ಬೆಳೆಯಲಿಲ್ಲ.

The words were classified as Easy, Medium, and Hard. Easy words are simple words consisting of only aksharas, without kaagunithas or vattaksharas (like ನಗರ). Medium words include kaagunithas but not vattaksharas. (like ರಾಜಧಾನಿ). The hard words consisted of everything - aksharas, kaagunithas and vattaksharas (like ವಿಶ್ರಾಂತಿಧಾಮ)

The accuracy was then calculated for each category, and then as a whole:

Table 2. Accuracy for each category of words

Category	No. of Words	Incorrect	Accuracy
Easy	17	1	94.11%
Medium	64	2	96.87%
Hard	44	6	86.36%
-	-	Overall	92.45%

The same sample was used to check the accuracy of some existing systems. The results and comparison are as shown below:

Table 3. Comparison of accuracy of different systems

System	Accuracy
Mobile App – KanScan	- no clear output -
Website – i2OCR	60% approx.
KanOCR	92.45%

Three other random samples from the internet were used to further compare the accuracy of the system. The results are provided below:

Table 4. Comparison of accuracy of different systems for different samples

Sample No.	No. of Words	i2OCR Accuracy	KanOCR Accuracy
1	23	60.9%	100%
2	38	47.36%	97.36%
3	55	63.63%	96.36%
-	Overall Accuracy	57.3%	97.9%

There is an increase in accuracy of around **40** percent from the already existing popular OCR system to our Kannada OCR System.

Accuracy for 33 other samples was carried out. The samples were taken from the internet and they had different sizes and fonts. The accuracy for all the samples was found out to be **82.01** percent.

Table 5: Accuracy calculation for each sample

Sample #	Word Count	Accuracy
1	14/18	77.78
2	17/19	89.47
3	18/19	94.74
4	20/22	90.91
5	17/31	54.84
6	210/250	84.00
7	257/304	84.54
8	248/296	83.78
9	274/326	84.05
10	175/236	74.15
11	78/92	84.78
12	218/236	92.37
13	288/316	91.14
14	270/312	86.54
15	30/38	78.95
16	197/248	79.44
17	226/279	81.00
18	212/251	84.46
19	267/306	87.25
20	133/173	76.88
21	214/271	78.97
22	112/144	77.78
23	198/240	82.50
24	246/270	91.11
25	177/215	82.33
26	213/266	80.08

27	199/275	72.36
28	134/172	77.91
29	64/77	83.12
30	58/70	82.86
31	119/154	77.27
32	188/229	82.10
33	102/130	78.46

5. CONCLUSION AND FUTURE WORK

Using Convolutional Neural Networks in OCR systems is found to be a very reliable method of converting an image to a text document. OCR system for the Kannada language with an accuracy of over 80 % has been constructed. The runtime for an article of a minimum of 100 words was found to be less than two seconds, which is majorly lesser than the existing systems available in the market.

However the system has the following drawbacks:

1. It is designed to work with printed documents, it cannot identify handwritten characters.
2. The characters must be crisp and at least 20 pixels in height otherwise the system does not work properly.
3. The system can be calibrated for only one Kannada font and size at a time, it does not work properly when there are multiple fonts and sizes in the same sample.
4. It has a lot of trouble recognizing punctuation.
5. Certain characters such as double vattaksharas have been excluded from the dataset entirely because of difficulties.
6. The accuracy can be improved by reworking algorithms or using a dictionary to constrain the output of the system

All of these challenges can be addressed in future work.

6. REFERENCES

- [1] HR Mamatha, S Sucharitha, Srikanta Murthy, "Multi-font and Multi-size Kannada Character Recognition based on the Curvelets and Standard Deviation", International Journal of Computer Applications, Foundation of Computer Science, New York, USA, 2011.
- [2] R Prajna, VR Ramya, HR Mamatha "A study of different text line extraction techniques for multi-font and multi-size printed kannada documents", International Journal of Computer Applications, Foundation of Computer Science, 2015.
- [3] M.K Jindal, R. K. Sharma & G.S. Lehal, "Segmentation of Horizontally Overlapping Lines in Printed Indian Scripts", International Journal of Computational Intelligence Research. ISSN 0973-1873 Vol.3, No.4 (2007), pp. 277–286
- [4] Ashwin T.V and P.S Sastry, "A font and size independent OCR system for printed Kannada using SVM", Sadhana, vol. 27, Part 1, February 2002, pp. 35–

58.

- [5] Anil. K. Jain, "Feature Extraction methods for Character Recognition – A survey", Pattern Recognition Volume 29, Issue 4, April 1996, Pages 641-662
- [6] K. Indira, S. Sethu Selvi, "Kannada Character Recognition System: A Review", InterJRI Science and Technology, Vol. 1, Issue 2, July 2009
- [7] Netravati Belagali, Shanmukhappa A. Angadi, "OCR for Handwritten Kannada Language Script", International Journal of Recent Trends in Engineering & Research (IJRTER) Volume 02, Issue 08; August – 2016.
- [8] C V, Aravinda, "Kannada handwritten character recognition using multi feature extraction techniques". International Journal of Science and Research (IJSR). Vol 10, 2014
- [9] Shashikala Parameshwarappa1 , B.V.Dhandra, "Basic Kannada Handwritten Character Recognition System using Shape Based and Transform Domain Features", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, July

2015

- [10] M. Vishwaas, M. M. Arjun and R. Dinesh, "Handwritten Kannada character recognition based on Kohonen Neural Network," 2012 International Conference on Recent Advances in Computing and Software Systems, Chennai, 2012, pp. 91-97.
- [11] G. Keerthi Prasad, I. Khan, N. R. Chanukotimath and F. Khan, "On-line handwritten character recognition system for Kannada using Principal Component Analysis Approach: For handheld devices," 2012 World Congress on Information and Communication Technologies, Trivandrum, 2012, pp. 675-678.
- [12] Gururaj mukarambi , dhandra b.v , mallikarjun hangarge, "recognition system for handwritten and printed kannada numerals and vowels", International Journal of Machine Intelligence ISSN: 0975–2927 & E-ISSN: 0975–9166, Volume 3, Issue 4, 2011, pp-259-262.
- [13] CS231n Convolution Neural Networks for Visual Recognition; <http://cs231n.github.io/convolutional-networks/>