SCRIPT RECOGNITION WITH MULTI-LINGUAL TRANSLATION

A PROJECT REPORT

Submitted by

1. Jignasa Parikh (16CP024)
2. Seby Amin (16CP031)

in fulfilment for the award of the degree of

B. TECH. (COMPUTER ENGINEERING)

Under the course of

CP442: PROJECT-II

BIRLA VISHVAKARMA MAHAVIDYALAYA
(ENGINEERING COLLEGE)
(An Autonomous Institution)
VALLABH VIDYANAGAR

Affiliated to

GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD
Academic Year: 2019 – 2020

B. V. M. ENGINEERING COLLEGE, VALLABH VIDYANAGAR-388120

APPROVAL SHEET

The project work entitled “Script Recognition with Multi-lingual Translation" carried out by “Jignasa Parikh and Seby Amin with ID No 16CP024 and 16CP031 is approved for the submission in the course CP442,Project-II (UDP) for the fulfilment for the award of the degree of B. Tech. (Computer Engineering).

Date:01/07/2020

Place:V.V.Nagar

CERTIFICATE

This is to certify that Project Work embodied in this project report titled “Script Recognition with Multi-lingual Translation" was carried out by Seby Amin and Jignasa Parikh with ID NO 16CP031 and 16CP024 respectively, under the course CP442, Project-II (UDP) for the partial fulfillment for the award of the degree of B. Tech. (Computer Engineering). Followings are the supervisors at the institute.

Date: 01/07/2020

Place: V.V.Nagar

Dr. H.D.Vasava

Professor of Computer Engineering

Prof. P.B.Swadas

Professor of Computer Engineering

(Dr. Darshak G Thakore)
Prof. & Head
Computer Engineering Department, BVM

DEPARTMENT OF COMPUTER ENGINEERING, B. V. M. ENGINEERING COLLEGE,
VALLABH VIDYANAGAR-388120

DECLARATION OF ORIGINALITY

We hereby certify that we are the sole authors of this report under the course CP442: Project-IIthat neither any part of this report nor the whole of the report has been submitted for a degree to any other University or Institution.

We certify that, to the best of our knowledge, the current report does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations or any other material from the work of other people included in our report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the boundary of fair dealing within the meaning of the Indian Copyright (Amendment) Act 2012, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in the current report and have included copies of such copyright clearances to our appendix.

We declare that this is a true copy of report, including any final revisions, as approved by report review committee.

We have checked write up of the present report using anti-plagiarism database and it is in allowable limit. Even though later on in case of any complaint pertaining of plagiarism, we are sole responsible for the same and we understand that as per UGC norms, University can even revoke the degree conferred to the student submitting this report.

Date: 01/07/2020
Institute code: 007

Jignasa Parikh
16CP024

Seby Amin
16CP031

PLAGIARISM SCAN REPORT

18%
Plagiarised

82%
Unique

Date	2020-07-04
Words	917
Characters	6003

PLAGIARISM SCAN REPORT

24%
Plagiarised

76%
Unique

Date	2020-07-04
Words	943
Characters	6358

PLAGIARISM SCAN REPORT

11%
Plagiarised

89%
Unique

Date	2020-07-04
Words	852
Characters	5423

PLAGIARISM SCAN REPORT

24%
Plagiarised

76%
Unique

Date	2020-07-04
Words	906
Characters	5799

PLAGIARISM SCAN REPORT

17%
Plagiarised

83%
Unique

Date	2020-07-04
Words	816
Characters	4844

PLAGIARISM SCAN REPORT

10%
Plagiarised

90%
Unique

Date	2020-07-04
Words	846
Characters	5203

PLAGIARISM SCAN REPORT

5%
Plagiarised

95%
Unique

Date	2020-07-04
Words	384
Characters	2476

PLAGIARISM SCAN REPORT

12%
Plagiarised

88%
Unique

Date	2020-07-04
Words	960
Characters	5750

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Dr. Prashant B Swadas** and **Dr. Hemant Vasava** for their guidance and constant supervision. We would also like to thank our institution for its support towards achieving our goal.

We are indebted to Birla Vishvakarma Mahavidyalaya Engineering College and our project guides for providing necessary information regarding the project and for their support in completing the project. With their feedback, we were able to improve our project in various aspects.

Finally, we would like to thank our fellow classmates for encouraging us to push ourselves further. After doing a lot of research, we have learned many new things and explored new domains which were unknown to us. During this project, we have come to value the quality of teamwork and hope to improve it more in the upcoming years.

Jignasa Parikh

Seby Amin

ABSTRACT

Nowadays, people move around places across the world and the most common difficulty is across the world is every place have different cultures and hence languages, due to which it becomes somewhere difficult for people to catch up the language. While going through certain documents or finding some interesting pictures captioned,scene text, handwritten text or most common text images with the language you are unaware of, makes us curious to know the quoted words or may be the entire document.

Text-Image mining concerns the extraction of implicit knowledge, image data relationship or other patterns not explicitly stored in the images. Text in images is one of the powerful sources of high-level semantics. Text Extraction plays a major role in finding vital and valuable information. Text extraction involves detection, localization, tracking, binarization, extraction enhancement and recognition of the text from the given image. These text characters are difficult to be detected and recognized due to their deviation of size, font, style, orientation, alignment, contrast, complex colored, textured background. Due to rapid growth of available multimedia documents and growing requirement for information, identification, indexing and retrieval, many researches have been done on text extraction in images. Several techniques have been developed for extracting the text from an image. This project aims to provide a better and faster detection of text from images through various machine learning and image processing algorithms and its easy translation to your preferred language. This project describes the method to extract text through histogram based image processing technique.

The input for our project is an image containing some text. This image is being processing with several image processing techniques so that correct text can be extracted from it and could be translated to the other user understandable language.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

ABBREVATION	FULL FORM
CNN	Convolution Neural Network
RNN	Recurrent Neural Network
ANN	Artificial Neural Network
ReLU	Rectified Linear Unit
ML	Machine Learning
FC	Fully Connected

TABLE OF CONTENTS

Chapter 1: Introduction to Project

1.1 Project Statement

Creating a module to recognize the script (detect image), extract image and translate it touser-preferred or user understandable language for a better flow of communication across the world.

1.2 Introduction to the Project

Research in recent years has given a lot of interest to textual data processing andespecially tomultilingual textual data. This is for several reasons: a growingcollection ofnetworked and universally distributed data, the development of communicationinfrastructure and the Internet, the increase in the number of people connected to theglobal network This has created a need toorganize and process huge volumes of data. The manual processing of these data (expert or knowledge based systems) is very costly in time and personnel, they are inflexible andgeneralization to other areas are virtually impossible, so we try to develop automaticmethods. [1]

In our day-to-day life, people are facing many problems in understanding the language.So, text extraction from documents,images or scenes is becoming prominent work in digital world.Text extraction from an image is one of the complicated tasks in image processing. There are various techniques to process the image and find the necessary output from it. The purpose of this work is to demonstrate that a tight dynamical connectioncan be made between the text and the interactive visualization imagery**.**

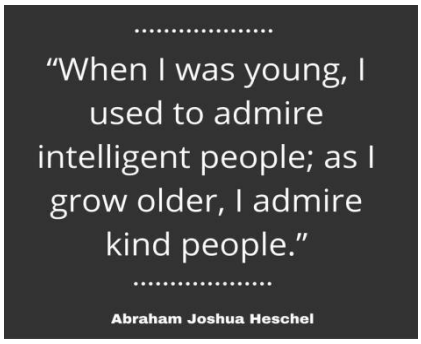


Figure 1.2(a): English input image[27]

Figure 1.2(b): Hindi input image[28]

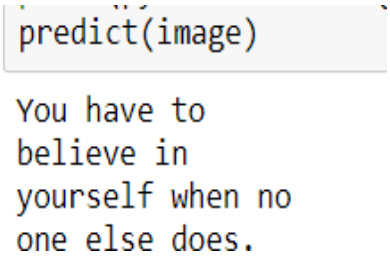
In the proposed system, in the beginning, the text images are received from the high definition camera or any scanner, so that the text in the image can be detected or extracted properly. Further, the image is provided to image pre-processing techniques firstly like gray-scaling, image rescaling, adjusting color contrast and normalization to prepare the image for text extraction from it effectively.

These output images are then segmented into lines and further words from the image to classify each word from image correctly. These images are then provided to CNN model to detect the words from the images. Now, the words detected are ready to translate into other languages.Translation of these texts totally depends on the accuracy of the text extraction and its proper detection.

Figure1.2(c): Hindi Word Dataset Image

Figure1.2(d): English Word Dataset Image

We worked with two languages to develop our text extraction model from CNN namely English and Hindi. Data of most common words of these language is provided by teserract (module to detect language and extract text of more than 150 languages) on its official site. We took some of those words converted them into images and finally replicate them to 500 images of those words with variations in image texts. Dataset is being prepared by deforming these text images into different levels. Deformation of images is done in 10 levels. Deformation of images between 3-5 levels provide optimal image text recognition, whereas decreasing it does not affect its deformation(every image looks similar to the other). Hence, dataset gets over fitted. While increasing the deformative level of image above 5, distorts the image completely and thus, the image text becomes unable to read also through the naked eyes.



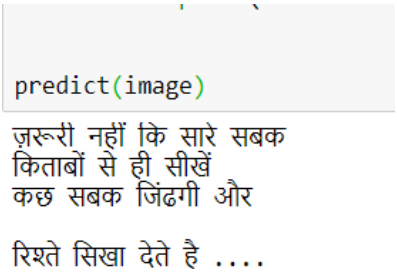


Figure 1.2(e): output for English language

Figure 1.2(f): Output for Hindi Language

1.3 Problem Domain

Problem domain comprises of combined approach to image processing and artificial intelligence. Text image mining is a major concern to image processing domain whereas extracting correct and important informative data from images through model training and learning through experience of training those model.

Image processing techniques provides solution to major issues of clearance of image data, finding minor details from images segmenting image to get only necessary information from it. Hence, image processing can be used anywhere in whatsoever field where image mining is concerned.

Machine learning provides good results working with neural networks. Generally, ANN models are used to detect object in images. But training through a good CNN or RNN models text in the images can alsobe extracted efficiently.

Thus, we have worked with both the trending techniques to approach to our module output.

1.4 Motivation

With the world moving towards digitalization, there could be a time where almost all the paper work wouldbe replaced with documents, and this can be considered an initiative to transform an image to the textual form and furthermore to reshape into the user proposed dialect.

Nowadays, image processing is used in each and every sector whether it is medical sector or defense sector. This is due to its increasing accuracy with the quality of the image whereas the upcoming technologies which give rise to the new techniques emerge in this field.

This can be a stepping stone to move towards Artificial Intelligence, where entire world is advancing. Moving towards digitalizing and developing in such emerging areas of AI leads to faster growth of our nation in technologies.

1.5 Objective of Project

Following objectives have been aimed during the design of the entire project:-

1] To transform image text into appropriate text.

2] To translate the text obtained through the above processing to user-friendly language.

1.6 Outline

The oncoming chapters include the detailed description to the project implementation and the future work that could be associated with it.

The first chapter has a brief introduction of the project statement, its domain, objectives to be fulfilled with the project and the outline of modules.

The second chapter has information regarding the researches and various approaches we have been gone through to complete this project.

Moving forward, now the third one encompasses the actual work done in the module and major theory of the processes used in the project.

Result analysis or observations on the project output is provided in the further chapter.

Moreover, the last chapter has the conclusion and future work statements that can be associated with it.

Chapter 2: Literature Survey

2.1 Overview

Veena Bansal and R.M.K Sinha presented a complete OCR for printed Hindi text written in Devanagari script. It uses various features namely: Coverage of the region of the core strip, Vertical bar feature, Horizontal zero crossings, Number of positions of the vertex points, Moments, Structural descriptors of the characters for classification and tree classifiers with the overall accuracy obtained at the character level is 93%. [24] Sinha and Mahabala designed a syntactic pattern analysis system for Devanagari script recognition. The system stores structural descriptors for each symbol of the script and were able to achieve the accuracy of around 90%.[25] Reena, Lipika and Chaudhury have tried to exploit information about similarity between numerals, Style invariant features and stylistic variations. They presented an approach for recognition of handwritten Devnagari numerals using multiple neural classifiers. Sandhya Arora have used Intersection features with Neural Network for Devanagari script and achieved 89.12% accuracy. [2]

Singh and Budhiraja presented an OCR system for handwritten isolated Gurumukhi script using Zoning, Projection histogram, Distance profile features, and Background directional features and used Support Vector Machines (SVM) for classification and thus obtained 95.04% of overall accuracy. Further Geeta and Rani represented an OCR system for Gurumukhi numerals using Zone Distance features and SVM classifier and achieved 99.73% accuracy. G. S. Lehal and Chandan Singh directed their efforts towards development of OCR system for Gurumukhi. They used Local features (concave/convex parts, number of endpoints, branches, joints) and Global features (number of holes, projection profiles, connectivity etc.). For classification hybrid classification technique, binary decision tree and nearest neighbour was used. They achieved a recognition rate of 91.6%. Dharamveer Sharma and PuneetJhajj used zoning feature with hybrid classification technique using KNN and SVM classifier and achieved 72.7% accuracy. A very influential attempt made by the Jalal, Feroz and Choudhuri for Bangla script. They represent neural network classifier by using Bounded rectangle calculation, Chain code generation, Slope distribution generation features. They achieved 96% system accuracy.[2]

Chaudhuri and Paul represent an OCR system to recognize Bangla and Devanagari using stroke and shaded portion feature with treeclassifier.[26] U. Bhattacharya, M. Shridhar, and S.K. Paruil implemented Neural network classifier for isolated Bangla characters with chain code features and achieved 92.14% accuracy on testing sets and 94.65% on training sets.Negi and Chakravarthy represent an OCR system with 92% performance using template matching, fringe distance for Telegu script. Another attempt was made by Patvardhan and Lakshmi for Telegu script. They used neural classifier by using directional features and they achieved 92% accuracy. Arun K Pujari, and C Dhanunjaya Naidu implemented an adaptive character recognizer for Telugu scripts using Multi resolution Analysis. They represented DNN (Dynamic Neural Network) using Wavelet analysis and achieved 93.46 % success rate. In south India, Kannada and Telugu have similar scripts. R Sanjeev and R D Sudhakar represent an OCR system for printed Kannada Script using two stage Multi-Network (Neural Network) classification technique employing wavelet feature and achieved 91% accuracy at character level. M Sagar, Shobha and Ramakanth designed a syntactical analysis system using Ternary Tree based classification for isolated Kannada characters. They have given more emphasis on Postprocessing step, using dictionary based approach to increase the OCR accuracy. T

V Ashwin and P S Sastry represents a font and sizeindependent OCR system for printed Kannada documents using support vector machines (SVM). B Chaudhuri U Pal gave a prototype.

OCR system for Oriya script. They use Directional features and Global Features and classified them using Decision tree classifier and achieved 96.03% accuracy at character level. Junaid, Umar, and Muhammad Umair attempted to make an OCR system for isolated Urdu characters using NN classifier using structural features like width, height and checksum of the character. Their prototype gained the accuracy of 97.43%. Another good attempt was made by Jhuwair and Abdul for Urdu script. They achieved the 97.12% recognition rate using Sliding window and Humoment feature using KNN classifier.[2]

CNNs have been studied and applied in the field of computer vision for a longtime. More than a decade ago, LeCun trained multilayer neural networks with the back propagation algorithm and the gradient learning technique, and demonstrated its effectiveness on the handwritten digit recognition. Deep learning has provided good generalization power in vision applications. In 2012, Krizhevsky achieved a breakthrough by outperforming the existing handcrafted features on Large Scale Visual Recognition Challenge (ILSVRC). The task includes recognition of 1000 object classes which was a very difficult problem to solve with the approaches of the time. Since 2012, CNNs have drawn a resurgence of attention in various tasks such as image classification, semantic segmentation, object recognition, video analysis, etc. Recently the networks are going deeper from a depth of sixteen to thirty and with the Deep Residual Learning methodology this is extremely facilitated even to 152 layers which won the first place in the ILSVRC 2015 classification competition.

Computer vision, and object recognition in particular, has made tremendous advances in the past few years. The PASCAL VOC Challenge, and more recently the Large Scale Visual Recognition Challenge (ILSVRC) based on the Image Net dataset have been widely used as benchmarks for numerous visualization-related problems in computer vision, including object classification. Since image visualization through image processing techniques have become an easy task to work with images, we got a lots of work in this area to reference to.

The research papers that we have read have been cited in the chapter on references. After reading a few research papers, we were familiar with the approach that we had to use and hence began by studying the various aspects of the system. The main aspect of our system was using CNN in order to detect descriptors. In order to understand the process of classification, we studied from the basic of CNN and neural networks primarily. In order to understand CNN, we had to study the papers and referenced various books too. A great amount of time was spent on modeling as that was the deciding factor for the viability of our system as whether this module is valid for text image extraction or not.

After getting a clear idea about what CNN is, we moved on to study of its different layers of modeling: Convolution, maxpooling and softmax. Beginning with the basic model available in keras namely sequential model, we have defined our layers onto it. The papers we referenced used these methods hence, we decided to implement this to gain much more accuracy than the previous models. We started with our prepared dataset containing 500 images of each word image of a particular language (English or Hindi).

2.2 Various Approaches

2.2.1 Previous Approaches

Various packages in python are the major sources used in most of the implementation of text extraction. OCR(Optical Character Recognition) is used to recognize the text efficiently from the images. But using only OCR does not provide good accuracy. So, in our implementation we have tried to use some basic image processing tactics combined with neural network model to extract our text from the images.

2.2.2 Issues in Previous Approaches

Issues arise in previous approach when people need to work on different languages rather than preferring to detect just the English language. Accuracy for different languages differs in OCR as the styles and texture of the words in different languages are diverging.

However, using the method of pre-processing of image then its extraction through our prepared dataset may help in increasing certain accuracy at some level. Processing the image before providing to some model increases the chances of getting the correct text from the picture.

2.2.3 Various Texts in Images

Images can be broadly classified into Document images, Caption text images and Scene text images.Text extraction involves detection, localization, tracking, binarization, extraction, enhancement and recognition of the text from the given image. Several techniques have been developed for extracting the text from an image. Following is brief description about types of text images and some approaches used to extract text from image.[7]

2.2.3.1 Document Text Image

A document image usually contains text and few graphics components. Document images are acquired by scanning journal, printed document, degraded document images, handwritten historical document, and book cover etc. The text may appear in a virtually unlimited number of fonts, style, alignment, size, shapes, colors, etc. Extraction of text in documents with text on complex color background is difficult due to complexity of the background and mix up of color(s) of fore-ground text with colors of background. [7]

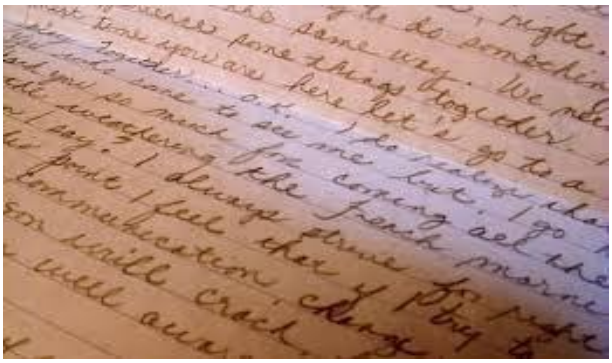


Figure 2.2.3.1: Document text image[30]

Different Approaches used for Document text image:

- A robust approach to segment text from color images was put forth by Y. Zhan. The proposed algorithm uses the multiscale wavelet features and the structural information to locate candidate text lines. Then a SVM classifier was used to identify true text from the candidate text lines. This approach mainly included four stages. In preprocessing step text blocks were enhanced by using cubic interpolation to rescale the input text blocks and a Gaussian filter to smooth the text blocks and remove noises. These image blocks were split into connected components and non-text connected components were eliminated by a component filtering procedure. The left connected components were merged using K-means clustering algorithm into several text layers, and a set of appropriate constraints were applied to find the real text layer. Finally, the text layer was refined through a post-processing step. [8]

- Thai described an approach for effective text extraction from graphical document images. The algorithm used Morphological Component Analysis (MCA) algorithm, an advancement of sparse representation framework with two appropriately chosen discriminative over complete dictionaries. Two discriminative dictionaries were based on undecimated wavelet transform and curvelet transform. This method overcame the problem of touching between text and graphics and also insensitive to different font styles, sizes, and orientations. [9]

- S.Audithan formulated an efficient and computationally fast method to extract text regions from documents. They proposed Haar discrete wavelet transform to detect edges of candidate text regions. Non-text edges were removed using thresholding technique. They used morphological dilation operator to connect the isolated candidate text edge and then a line feature vector graph was generated based on the edge map. This method exploited an improved canny edge detector to detect text pixels. The stroke information was extracted the spatial distribution of edge pixels. Finally, text regions were generated and filtered according to line features. [10]

- Grover described an approach to detect text from documents in which text was embedded in complex colored document images. They proposed a simple edge based feature to perform this task. The image was converted to gray scale by forming a weighted sum of the R, G, and B components. Then edge detection was performed on the gray-scale image by convolving the image with Sobel masks, separately for horizontal and vertical edges. Convolution was followed by elimination of non-maxima and thresholding of weak edges.Next, the edge image was divided into small non overlapping blocks of m x m pixels, where m depends on the image resolution. They performed block classification using pre-defined threshold which would differentiate the text from the image.[11]

2.2.3.2 Caption Image

Caption text is also known as Overlay text or Cut line text. Caption text is artificially superimposed on the video/image at the time of editing and it usually describes or identifies the subject of the image/video content. The superimposed text is a powerful source of high-level semantics. These text occurrences could be detected, segmented, and recognized automatically for indexing, retrieval and summarization. The extraction of the superimposed text in sports video is very useful for the creation of sports summary, highlights etc. These types of caption text include moving text, rotating text, growing text, shrinking text, text of arbitrary orientation, and text of arbitrary size. [7]

Figure 2.2.3.2: Caption text image[29]

Different Approaches used for Caption image:

- A superimposed text extraction method was introduced by V.Vijayakumar for detecting video text regions containing player information and score in sports videos. Key frames from the video were extracted using Color Histogram technique to minimize the number of video frames and converted to gray images. Text image regions were cropped. Canny Edge Detection algorithm was applied to detect edges on the cropped image. From this edge detected images, text region was identified and fed to an Optical Character Recognition system which produces index-able keywords. [12]

- The goal of Min approach was to detect both low-contrast and high-contrast artificial texts invariant with language and font-size in a complex background video image. The sobel color edge detector was applied to detect edges. Non text points were eliminated by applying low threshold determined by the histogram of edge strength and selective local thresholding. Further enhancement was done using Edge-Strength Smoothing (ESS) operator and EdgeClustering-Power (ECP) operator. To locate the text region, coarse-to-fine (horizontal and vertical) projection was used. [13]

- Yih-Ming proposed a scheme to extract the caption text from various sports videos. Iteratively temporal averaging approach was used in caption extraction process. To improve the image quality and to reduce noise spatial-image analysis was performed. Threshold value was determined using binarization process based on the global mean and the standard deviation of the gray level of the averaged video image. Binarization may lead to holes and disconnectivity on video captions with blurred background. This was cured by morphological processing. Each connected component was

used to extract geometrical features to identify the captions. A model-based segmentation approach was applied to accurately extract the caption contents. [14]

- A technique for detecting caption text from videos for global indexing purpose based on hierarchical region-based image model was proposed by Leon. Binary Partition Tree (BPT) was created by combining color and contour homogeneity Criteria. Texture descriptors were estimated on the full image by means of a multi-resolution analysis using a Haar wavelet decomposition to highlight the candidate regions in the BPT. The largest connected component was selected as the area of support for computing geometric descriptors. Region evaluation was carried out by combining region-based texture information and geometric features. Final caption text nodes were selected by analyzing the various subtrees in BPT. [15]

2.2.3.3 Scene Image

Scene text appears within the scene which is then captured by the recording device i.e. text which is present in the scene when the image or video is shot. Scene texts occurs naturally as a part of the scene and contain important semantic information such as advertisements that include artistic fonts, names of streets, institutes, shops, road signs, traffic information, board signs, nameplates, food containers, cloth, street signs, bill boards, banners,and text on vehicle etc. Scene text extraction can be used in detecting text-based landmarks, vehicle license detection/recognition, and object identification rather than general indexing and retrieval. [7]

It is difficult to detect and extract since it may appear in a virtually unlimited number of poses, size, shapes and colors, low resolution, complex background, non-uniform lightning or blurring effects of varying lighting, complex movement and transformation, unknown layout, uneven lighting, shadowing and variation in font style, size, orientation, alignment & complexity of background.

Figure 2.2.3.3:Scene text image[27]

Different Approaches used for Scene image:

- Angadi proposed a methodology to detect and extract text regions from low resolution natural scene images. Their proposed work used Discrete Cosine Transform (DCT) based high pass filter to remove and suppress the constant background. The texture feature matrix was computed on every 50x50 block of the processed image. A newly defined discriminate function was used to classify text blocks. The detected text blocks were merged to obtain new text regions. Finally, the refinement phase was a post processing step used to improve the detection accuracy. This phase used to cover small portions of missed text present in adjacent undetected blocks and unprocessed regions.

The proposed methodology had been conducted on 100 indoor and outdoor low resolution natural scene images containing text of different size, font, and alignment with complex backgrounds containing Kannada text and English text. The approach also detected nonlinear text regions and can be extended for text extraction from the images of other languages with little modifications. [16]

- Pan proposed a novel hybrid method where in a text region detector was designed to generate a text confidence map. A Local binarization approach was used to segment the text components using text confidence map. A Conditional Random Field (CRF) model was used to label components as text or non-text which was solved by minimum classification error (MCE) learning and graph cuts inference algorithm. A learning based method by building neighbouring components into minimum spanning tree (MST) and cutting off interline edge with an energy minimization model to group the text components into text lines. [17]

- Fabrizio offered a region based approach that starts by isolating letters, then groups them to restore words. The process was based on a new segmentation method based on morphological operator called Toggle Mapping Morphological Segmentation (TMMS) and a classification step based on a combination of multiple SVM classifiers. The training data base composed of 32400 examples extracted from various urban images and different configurations of classifiers have been tested to get the highest classification accuracy. [18]

- Kohei introduced a new approach to detect and extract text from commercial screenshot images. Their approach implemented edge-based method and connected component labeling method known as blob extraction method. Combination of homogeneity edge detection filter and appropriate threshold number separated the text from the image. [19]

- A method for localizing text regions within scene images was introduced by Luz. A set of potential text regions was extracted from the input image using morphological filters. Connected Components (CC) were identified using ultimate attribute openings and closings, and selected a subset of text region after combining some of the CCs. Decision tree classifier were used to distinguish text or non-text regions. [20]

2.2.4 Proposed Methodology

Fully automatic text extraction from images has always been a challenging problem. The difficulties arise from variations of text in terms of character font, size, orientation, texture, language and color, as well as complex background, uneven illumination, shadows and noise of images. Experiments show that applying conventional OCR technology directly leads to poor recognition rates. Therefore, efficient detection and segmentation of text characters from the background is necessary to fill the gap between image and video documents and the input of a standard OCR system. [6]

Optical Character Recognition (OCR) is a method to locate and recognize text stored in an image and convert the text into a computer recognized form and a uniform representation such as ASCII or Unicode. OCR systems can only deal with printed characters against clean backgrounds and cannot handle characters embedded in shaded, textured or complex backgrounds.

The process of optical character recognition(OCR) is segmentation, correlation and classification. In the first stage, OCR crops each character in the binary image. In the correlation phase OCR matches the cropped characters with the templates. During the classification process, it recognizes the text in binary image, if the cropped character matches with the template. In the template matching process, the characters will be identified as letters and the binary image will be converted to text.[6]

We have tried using a different approach in our module. This module captures image, apply preprocessing to it and then d segment its lines and then words to get a single-word text image and after that, through the histogram based process the text can be detected by differentiating it from the background image. This portion is then applied to CNN model for further text classification.

Histogram based text extraction suggests the differentiation based on the peaks and valleys in the image, which implies the image should be captured through high definition camera to classify the text peak and valleys properly.

Moving towards the CNN model, each layer has its own work, so to define the number of layers to fit the classification becomes a big task which has been successfully classified in this module. The feasibility of CNN to text from text images taken under uncontrolled environment has been studied. The models are designed based on the dataset we have prepared and implemented with caution to maintain the accuracy up to the mark.

With the advent of digitalization, these models will be in demand and can be opt to modify as per the needs of customer. Digital evolution may lead to document most of the work in soft copy, hence opportunities for these types of solution increases with the growing world.

The Flow Diagram of our Methodology is as follows:

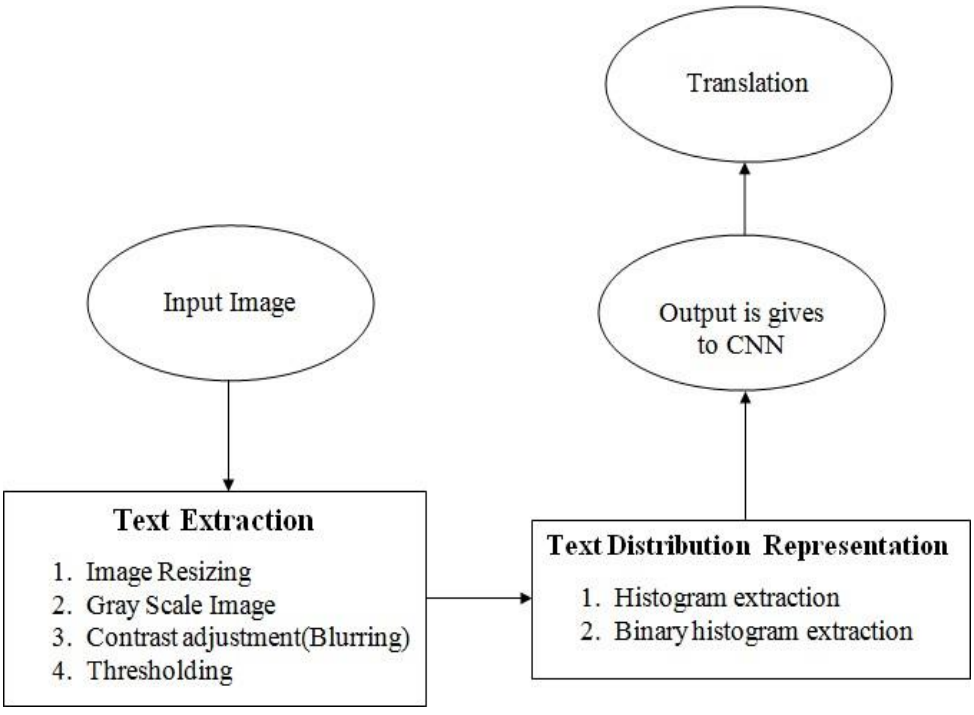


Figure 2.2.4: Flow Diagram of Methodology

2.3 Survey

2.3.1 Pre-Processing of image

Various pre-processing techniques can be included to fit the image completely to extract the image. These may include grey-scaling, binarization, skewing, noise removal, thinning and skeletonization and hence extraction can be done.

- Binarization

As a basic method this can be done by fixing a threshold(normally 127 as it is exactly half of the pixel range 0–255) and considering it as a white pixel if the pixel value is greater than threshold, else black pixel.

But this approachmay not alwaysgive us desired results. In the cases where lighting conditions were not uniform in image, this method fails terribly to binarize the image correctly.

So, the crucial part of binarization is determining the threshold. This can be done by using various techniques.

1. *Local Maxima and Minima Method*

$$C(i,j) = \frac{I_{max} - I_{min}}{I_{\max} - I_{min} + \varepsilon}$$

I_{max}= Maximum pixel value in the image
I_{min}= Minimum pixel value in the image
E = Constant value

2. *Otsu's Binarization* :

This method gives a threshold for the whole image considering the various characteristics of whole image (like lighting conditions, contrast, sharpness etc) and that threshold is used for Binarizing image.

3. *Adaptive Thresholding* :

This method gives you a threshold for a small part of image depending on the characteristics of its locality and neighbours i.e there is no single fixed threshold for whole image but every small part of image has a threshold.[8]

- Gray-scaling:

Gray-scaling means converting the image from RGB to black and white colorsbasically (it may vary from complete black to complete white). Images are converted to gray-scale for various factors. It reduces the dimension, model complexity and also some algorithm are designed to work on only gray-scale images.

2.3.2 Convolutional Neural Network

In machine learning, a convolutional neural network (CNN or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery.

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications take an input image, process it and classify it under certain categories. Computers see an input image as array of pixels and it depends on the image resolution.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1.

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to over fitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. However, CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme. [4]

The CNN architecture is shown in the figure given below:

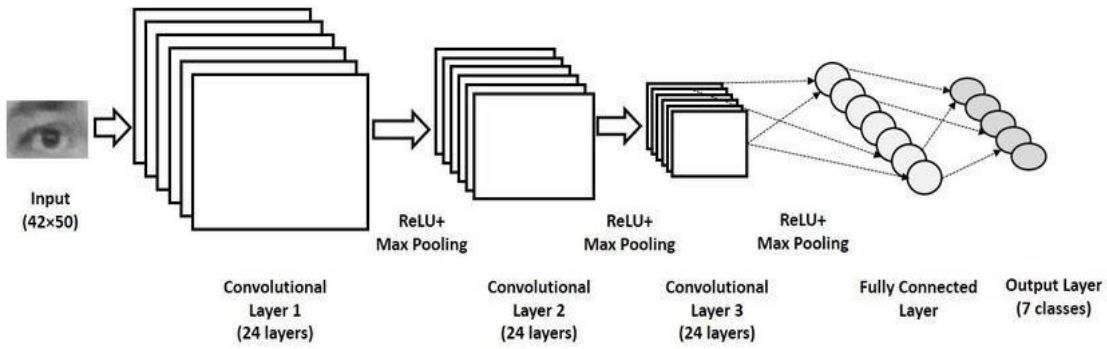


Figure 2.3.2: CNN Diagram [4]

Convolutional Neural Networks are made up of neurons that have learnable weights and biases. In this, each neuron receives some inputs, performs a dot product and optionally follow it with a non-linearity. The whole network still expresses a single differentiable score function which is from the raw image pixels on one end to class scores at the other.

Convolutional Neural Network have the following layers:

- Convolutional Layer
- Pooling Layer
- Normalization Layer
- Fully-Connected Layer
- Converting Fully-Connected Layers to Convolutional Layers.

Convolution Layer

The output feature map of each convolution layer is determined by a convolution operation between the upper feature maps of the current layer and convolution kernels. Generally, the output feature map could be indicated by Equation: [4]

$$x_j^{\lambda} = \sum_{i \in M_j} x_i^{\lambda-1} \times k_{ij}^{\lambda} + b_j^{\lambda}$$

Where, λ means the λ^{th} layer,
k_{ij} represents the convolutional kernel,
b_j is the bias,
M_j is a set of input feature maps.

ReLU Activation Function

The activation function determines the neural network data processing method, and influences the learning ability of the neural network model. The ReLU activation function has a fast convergence speed and alleviates the problem of overfitting. As a result, this method is used for the output of every convolutional layer. The ReLU activation function formula is shown in the following Equation: [4]

$f(x) = \max(0, x)$

Max Pooling Layer

The max-pooling layer, which is a form of nonlinear down-sampling, could reduce the size of the feature maps gained from the convolutional layers to achieve spatial invariance, which leads to faster convergence and improves the generalization performance .[4]

When the feature map a is passed to the max-pooling layer, the max operation is applied to the feature map a, which produces a pooled feature maps as the output. As shown in the following Equation, the max operation selects the largest element:

$$s_j = \max_{i \in R_j} \alpha_i$$

Where,R_j represents pooling region j in feature map a, i is the index of each element within its denotes the pooled feature maps.[4]

Softmax Regression

Softmax regression is used for multiclassification problems. The hypothesis function is shown in Equation:

$$h_\theta(x) = \frac{1}{1+\exp(-\theta^T x)}.$$

The model parameters θ are trained to minimize the cost function J(θ). In the equation below, 1{.} is the indicator function, so that 1{a true statement} = 1, and 1{a false statement} = 0. The cost function J(θ) is shown in the following Equation: [4]

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} \log p(y^{(i)} = j|x^{(i)};\theta)].$$

The training database is denoted $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\ldots,(x^{(m)},y^{(m)})\}$, $y^i \in \{1,2\ldots,k\}$. In Softmax regression, the possibility of classifying x into category j is[4]

$$p(y^{(i)} = j|x^{(i)};\theta) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}.$$

2.3.3 Sequential model

The Keras Python library makes creating deep learning models fast and easy.

The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.

The functional API in Keras is an alternate way of creating models that offers a lot more flexibility, including creating more complex models. [6]

The Sequential model API is a way of creating deep learning models where an instance of the Sequential class is created and modellayers arecreated and added to it.

The model needs to know what input shape it should expect. For this reason, the first layer in a Sequential model (and only the first, because following layers can do automatic shape inference) needs to receive information about its input shape. There are several possible ways to do this.

We have used the following way:

To specify a fixed batch size for your inputs (this is useful for stateful recurrent networks), you can pass a batch_size argument to a layer. Suppose, if you pass both batch_size=32 and input_shape=(150,150) to a layer, it will then expect every batch of inputs to have the batch shape (32, 150,150).[20]

Before training a model, you need to configure the learning process, which is done via the compile method. It receives three arguments:

- An optimizer:

  This could be the string identifier of an existing optimizer (such as rmsprop or adagrad), or an instance of the Optimizer class.

- A loss function:

  This is the objective that the model will try to minimize. It can be the string identifier of an existing loss function (such as categorical_crossentropy or mse), or it can be an objective function.

- A list of metrics:

  For any classification problem you will want to set this tometrics=['accuracy']. A metric could be the string identifier of an existing metric or a custom metric function.

Chapter 3:Implementation of the Project work

Working of the project goes in the following manner:

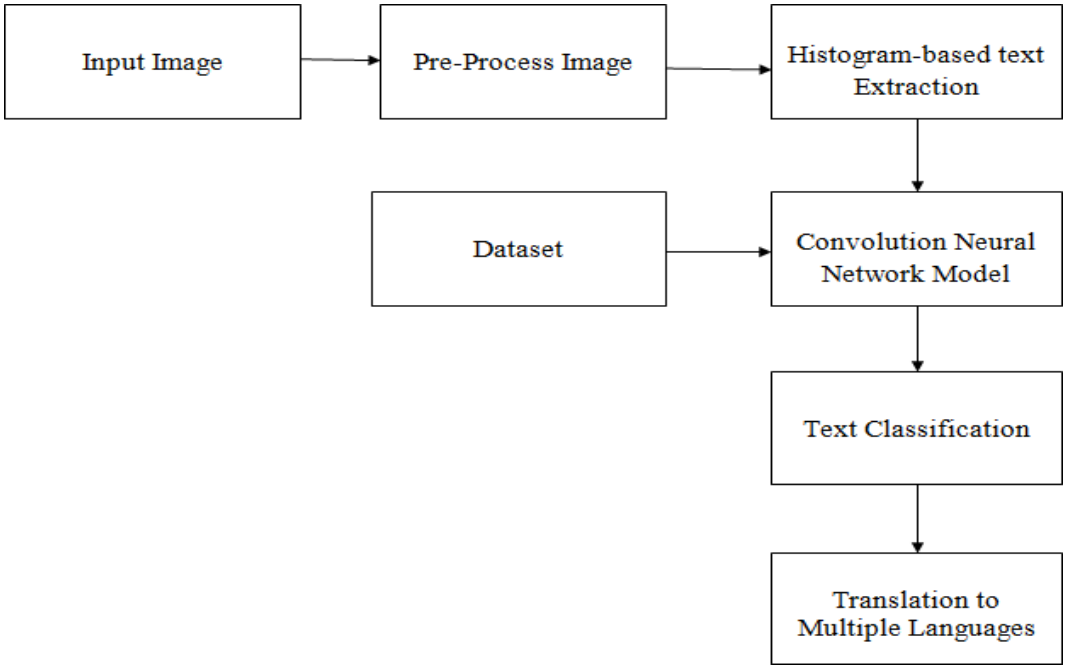


Figure 3: Flow diagram

3.1 Input image

The module takes text image as input which has to be classified into valid text of a particular language(here English and Hindi). Input image should be clear, precise, of high quality and should contain proper text image in it so that proper text can be extracted from it.

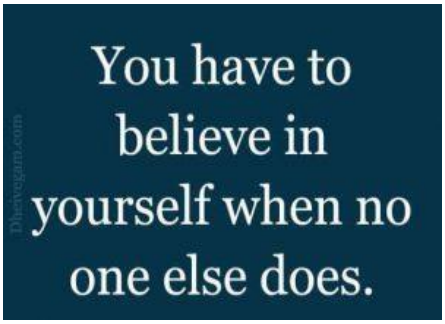


Figure 3.1(a): English input image [27]

Figure 3.1(b): Hindi input image[28]

3.2 Normalization(Gray-Scaling)

These images are then normalized and converted to gray scale to reduce its size and precisely for faster execution of the program.

self.gray_img = cv2.cvtColor(self.img, cv2.COLOR_BGR2GRAY)

```
plt.imshow(img_grayscale, cmap = plt.get_cmap('gray'))
#plt.imshow(img_grayscale)
```

```
<matplotlib.image.AxesImage at 0x1420107b198>
```

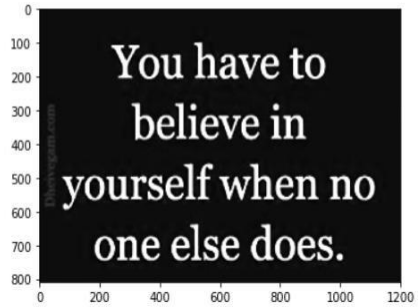


Figure 3.2(a): Gray scale image of English

```
plt.imshow(img_grayscale, cmap = plt.get_cmap('gray'))
#plt.imshow(img_grayscale)
```

```
<matplotlib.image.AxesImage at 0x14207632da0>
```

Figure 3.2(b): Gray scale image of Hindi

3.3 Blurring

For fine extraction of each and every word in the text, blurring and then thresholding image plays a vital role in its processing.

smoothed_img = cv2.blur(self.gray_img, (3, 3), anchor=(-1, -1))

```
<matplotlib.image.AxesImage at 0x142002eed30>
```

Figure 3.3(a): Smooth image for English

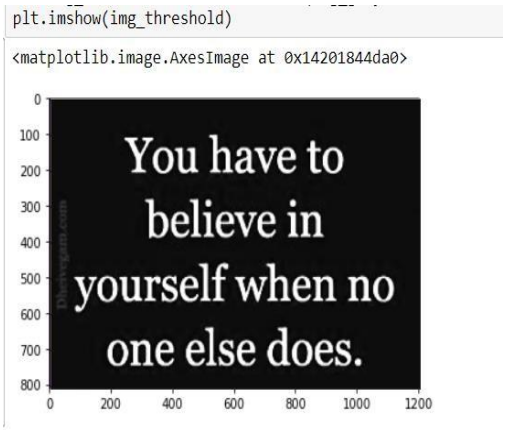
```
<matplotlib.image.AxesImage at 0x14203b22cf8>
```

Figure 3.3(b): Smooth image for Hindi

3.4 Thresholding

Otsu's Binarization : This method gives a threshold for the whole image considering the various characteristics of whole image(like lighting conditions,contrast,sharpness etc.) and that threshold is used for Binarizing image.

_, self.thresh = cv2.threshold(smoothed_img, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)



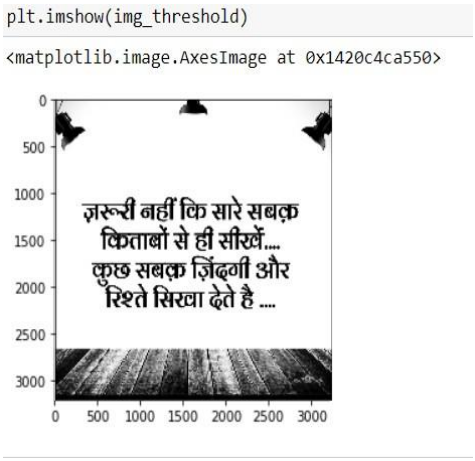


Figure 3.4(a): Thresholding image for English Figure 3.4(b): Thresholding image for Hindi

3.5 Histogram based Text extraction

We find that the most strikingfeature of the text line in a paragraph is theuniform distributionstructure of its layout, i.e. text lines have similar length andheight. Moreover, the gaps between two subsequent lines arealmost constant through an article. Therefore, we proposea histogram-based method to represent the row-wise textdistribution accumulated row by row from the text candidatebinary image. The histogram of the entire image is denotedas; n = (n (1), n (2), . . ., $n(H)$) and it's $y^{th}(y = 1, 2, ..., H)$

value is computed by:

$$n(y) = W - \sum_{x=1}^{W} I(x, y)$$

In order to remove the noise and differentiate the text linesfrom the gap between consecutive text lines, the histogram isfurther converted into a binary one; **b** = (b(1), b(2), . . , b(H))and it's yth(y = 1, 2, ...,H) value is computed by:

$$B = \frac{\sum_{y=1}^{H} n(y)}{H}$$

We plot the values of the binary histogram of the example image and the x-axis represents the indices of rows in the binary image, and the y-axis shows the binary value, where value 1 indicates text lines and value 0 indicates gaps.

There are various functions provided by python library to compute this algorithm with effective results. So, let us define the steps to work forward.

Text recognition from paragraph:

Since we have trained the network for words so we have to do the segmentation of paragraphs in words. For this we applied the two techniques.

3.5.1 Line segmentation

3.5.2 Word segmentation

Then each word is processed individually and stored in a file. Then the file is read and printed in the text area.

1) Line segmentation

Text line segmentation can be roughly categorized into the following:

Smearing methods: The idea is to run with short white regions which are filled with black pixels intending to form large black pixels, which may be considered as text line areas. The limitation of the smearing methods is that it cannot handle the touching and overlapping components well.

Horizontal projections:

This is formed by a vector containing the sum of each image line. The vector's local minimum is assumed to be the projection of white regions in between lines, and the image is segmented consequently. The main drawback of this approach is that it doesn't work well skewed, curved and fluctuating lines. Straight lines in an image can be detected using Hough transform. The approach creates an angle, offset plane in which the local maxima are assumed to correlate with text lines, which in the limitation of Hough transform in detecting curved text lines.

Bottom-up approaches:

Use of connected components or pixels which areclosely connected is formed on geometrical criteria to form text lines. Few other approaches can also be seen, such as: repulsive attractive networks, stochastic methods and text line structure enhancing.

Figure 3.5(a): English text Line Segmentation

Figure 3.5(b): Hindi text Line Segmentation

Approach by Sanchez for text line segmentation in handwritten historical documents is based on computing a white/black transition map to achieve a rough detection of the line regions in the image using transition map. Another work presented involves handwritten text line segmentation by shredding text into lines. Topological assumptionis that for each text line, exists a path from one side of the image to the other that traverses only one text line. Clustering with distance metric learning for handwritten text line segmentation is proposed along with handwritten document image segmentation into text lines and words, where locating the optimal succession of text and gap areas within vertical zones by applying Viterbi algorithm is addressed.

Foreground and background information based handwritten line segmentation is proposed by Roy in which the morphological operation and Run-Length Smearing Algorithm (RLSA) is used to get individual word as a component. The foreground portion of the image is eroded to get some seed components from the individual words of the document. Erosion is also done on background portions to find some boundary information of text lines.

Using the positional information of the seed components and the boundary information, the lines are segmented. The method for Indian scripts is based on interdependency between text-line and inter-line gap. A technique for line segmentation of handwritten Hindi text is based on header line detection, base line.

2) Word segmentation

Word segmentation is the process of determining the word boundaries in a sentence or a document by computer algorithms.

After the Page is segmented into lines, individual lines are segmented into words. Interline distance is calculated to identify the individual lines. Word segmentation isdone using an x axis objection of the text in the line. Segmentation algorithm cannot work when different regions of the handwritten text are in different script.

Figure 3.5(c): English text Word Segmentation

Figure 3.5(d): Hindi text Word Segmentation

The word segmentation is proposed by B. B. Chaudhury deals with text line identification of Bangla and English, Hindi scripts. Stepwise histogram is drawn to detect local text lines and gaps. Vertical strip width is computed and noise would be detected. Initial curves were drawn by calculating average standard deviation. Six different scripts text lines were detected with maximum accuracy of 79%.

Preparation of dataset:

We took different word images in many font styles. Then each word was processed and distorted with value 4 to give the different shape of words. Different sizes of words were generated using the parameter of breadth and height with probability distribution of 0.8.

Thus the data processing generated 1000 words for each word. All the words of the same name were stored in one folder and the name given to the folder acts like a label in training. And this dataset was used for training the model.

Processing of dataset:

Each image of the dataset was preprocessed before sending for training the model. These are the different steps taken for that:

- For transparency we used alpha channel and rgb channel. We wanted the white background of the image and the text written in it as black. So on white background a transparent image was rendered to create the new image.

- Now that image was converted to gray scale image.

Then this gray scale image was resized to (150*150) pixels for uniformity in training.

Pickling and CSV preparation:

- What is a pickle file?

Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it "serializes" the object first before writing it to file. Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

Example:(For using this first import through command)

import pickle

Pickle has two methods. One is dump which dumps an object to a file object and the second one is load, which loads an object from a file object.

- Prepare something to pickle:

```
list1 = ['python','java','ruby']
file_Name = "picklefile"
# open the file for writing
fileObject = open(file_Name,'wb')
# this writes the object list1 to the
# file named 'picklefile'
pickle.dump(list1,fileObject)
# here we close the fileObject
fileObject.close()
# we open the file for reading
fileObject = open(file_Name,'r')
# load the object from the file into var b
list2 = pickle.load(fileObject)
list2
['python','java','ruby']
list1==list2
True
```

- Why cPickle and how image saved?

✓ We have used cPickle in our project because it is 1000 times faster than pickle.
✓ We have changed the image into an array with numpy module.
✓ We found the shape of tensor, mean, standard deviation of the dataset.

- CSV file:

In the csv file we wrote the name of the letter with index starting from 0. It helps in traversing through the dataset.

A snapshot is given here.

Figure 3.5(e): CSV Classes snapshot

Training the model:

We have divided the dataset into two parts.

- 80% of the dataset was used for training.
- 20% of the dataset was used for testing.
- We have used Keras for training the model.Convolutional neural network(CNN) is used for recognition.

The specifications of the network are given below:

- Sequential model from keras is used.
- 3 convolutional layers are present.
- 3 max pooling layers are present.
- 2 dense layers are present with activation function relu and softmax.
- All the convolutional layers use activation function relu.
- The data is passed in the network through batch size of 128.
- We have done 10 epochs.
- So we got an accuracy of 79%.

Predictions of individual words:

We take the image of the word from the user and try to predict it using the trained model.But before predicting it, pre-processing of images need to be done. So we changed to transparent png which means white background with text written in black. Then we change it to gray scale image and resize it to 150*150 pixels.Then we predicted it.In most of the cases it produced the correct output.

3.6 CNN

Instead of handcrafted features, convolutional neural networks are used to automatically learn a hierarchy of features which can then be used for classification purposes. This is accomplished by successively convolving the input image with learned filters to build up a hierarchy of feature maps.

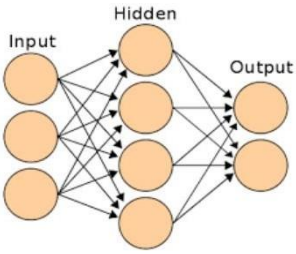


Figure 3.6(a): CNN [3]

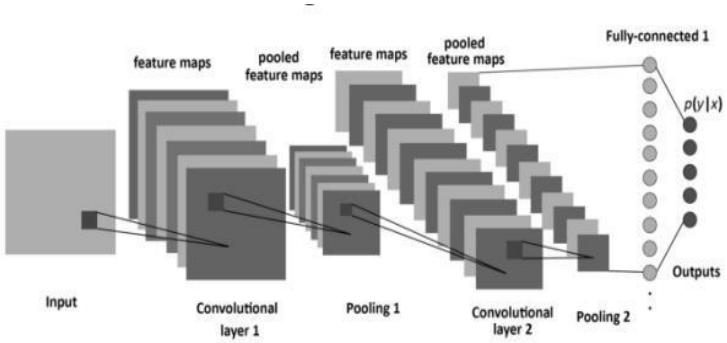


Figure 3.6(b): CNN layers[3]

A Convolutional Neural Network (CNN) has four Type of layers as follows –

1. Convolutional Layer (CONV)
2. Rectified Linear Unit Layer (Re Lu)
3. Pooling Layer (POOL)

3.6.1 Convolutional Layer

Convolution layer is used to extract features from an input image. It preserves the relationship between pixels by learning image features using small squares of input data. The Layer contains N filters which are small in size (for example [3x3] as in fig). These 3x3 filters are convoluted with the input image matrix by sliding the filter slide through the width and height of the image. [1]

Firstly, the feature matrix is multiplied pixel by pixel with the selected square from the image. Then the values are added and finally divided by the total number of pixels (in our example it is 9 due to 3x3 filter size). The obtained value is inserted in a new matrix. This process helps to reduce the image without loss of any feature. [5]

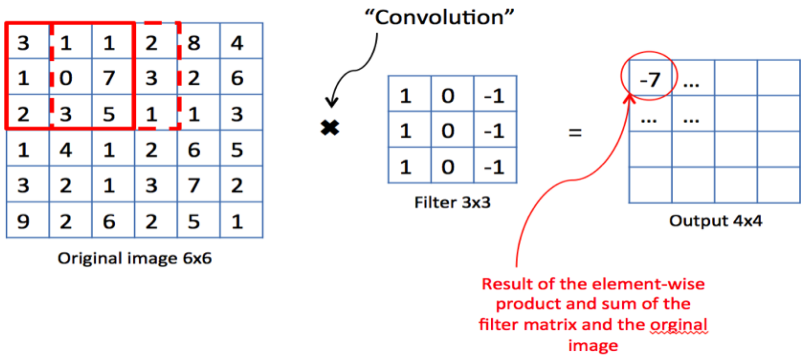


Figure 3.6.1: Convolution layer[22]

3.6.2 Rectified Linear Unit Layer (Re Lu)

The positive pixels are important for the further finding of features and the negative values are of less importance. The Re Lu layer either converts the pixel to 0 or 1. If the value of pixel is negative then it is converted to 0 and for any value greater than 0 it retains the same value.

It is a piecewise linear **function** that will output the input directly if is positive, otherwise, it will output zero.

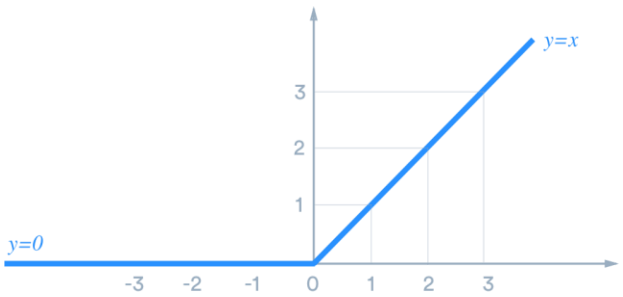


Figure 3.6.2: ReLU layer[22]

3.6.3 Maxpooling

The Pooling layer does a simple job of down sampling or compressing the dimensions of the input image. A stride is selected which can be 2x2 or 5x5 etc. After the selection of stride, it is applied to the dimension matrix obtained from the Convolution Layer. Maximum value is taken from each stride and stored in a new matrix. Depending on the stride Pooling is of two types Max Pooling and Minimum Pooling. [1]

When the stride is large such Pooling is known as Max Pooling whereassmall stride is known as Minimum Pooling. For example, if the input is [64*64*12] and if a stride of 2x2 is applied then after down sampling the output will be [32*32*12]. Figure 10 shows max Pooling Layer.

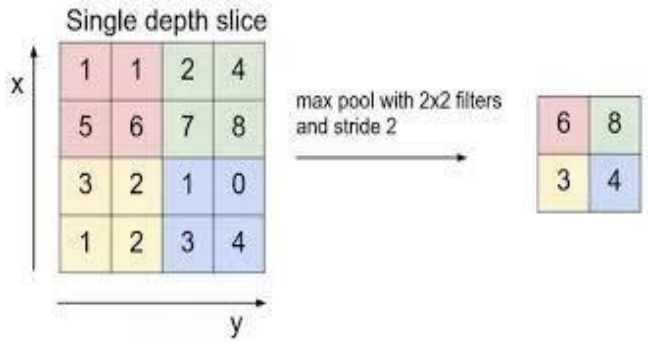


Figure 3.6.3: Maxpoolinglayer [23]

3.6.4 Flatten layer

Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

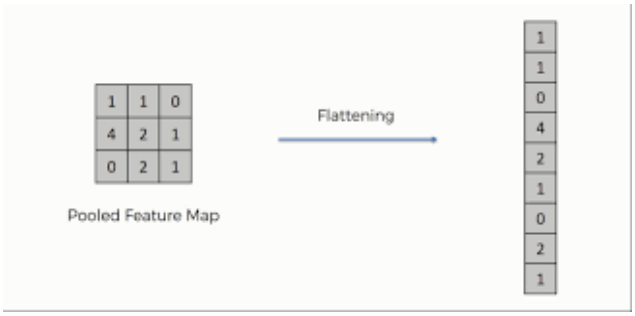


Figure 3.6.4: Flatten layer[21]

3.7 Classification

To perform classification of words of entire language is a difficult task. Hence, we took a set of words to train, which we have further replicated and through augmentation of whose words formed a dataset to perform a simple text extraction from an image text.

This dataset is applied to a CNN model that classifies the text inside the image and completes our module of text extraction from the image.

3.8 Translation to multiple languages

The output text is hence provided to the translate module for its translation to user-preferred language. Accuracy of this translation totally depends on the accuracy of text extraction from the entire module of this project. Thus, each step is essential for the module to reach to the perfect output of the text.

```
from translate import Translator
translator= Translator(from_lang="english",to_lang="hindi")
translation = translator.translate(trans)
print (translation)
```

```
आपको करना होगा
लॉयस) bCeaomel
अपने आप को जब नहीं
एक और करता है ।
```

Figure 3.8(a): Image of Translation of English language to Hindi Code and Output

```
from translate import Translator
translator= Translator(from_lang="hindi",to_lang="english")
translation = translator.translate(trans)
print (translation)
```

```
Not necessarily all lessons
Learn from books
A lesson will be a life and

Teach relationships ....
```

Figure 3.8(b): Image of Translation of Hindi language to English Code and Output

3.9 Implementation Requirements

All the modules of this project require certain software and hardware support and preparation of dataset required some basic hardware too.

Following are the required software tools which were used for project's implementation: -

1. Python 3.6.4- This is an open source programming language with a certain powerful library for designing neural network and performing image processing.

2. Libraries-
    a. Tensorflow (to identify patterns)
    b. Keras(as wrapper to low-level libraries like tensorflow)
    c. Matplotlib(for mathematical calculations)
    d. Numpy(for mathematical calculations)
    e. Pickle (serializing python data structure)
    f. os (to interact with Operating System)
    g. cv2 (for image related operations)
    h. math(for mathematical calculation)
    i. sys(to work with other files)

3. Dataset- prepared from a set of generated image text
4. IDE- Anaconda package

Following are the required hardware specification which were used for project's implementation:-

Operating System	Windows 10 or above
Packages	Anaconda
Processor	Minimum 64 bit
RAM	Minimum 4GB RAM
Hard Disk	Minimum 10GB

Table 3.9: Hardware specification

3.10 Design and Implementation Constraints

Sometimes dataset provided may not have proper or sufficient data to work on, which can create problem in training those data. Other cases can be modeling with inappropriate models or using more hidden layers than required. These issues can affect the accuracy of the system.

While image quality affects the result to the greater extent as it is the only way to lead through the entire process performance.

3.11 Assumptions and Dependencies

This project is developed on the windows platform and Jupyter notebook of Anaconda IDE hence it may not be compatible with any other operating system and thus can cause some problem while using it with other operating environment.

.

Chapter 4: Result Analysis

Our module provides the accuracy up to 79% from the dataset we have generated. Hence, we can say that you can get good result from a well generated dataset.

Image becomes readable from the pre-processing steps we took to project our text image to histogram based text extraction mechanism.

Histogram based text extraction may sometimes fail to get the text if the image quality degraded after/before the pre-processing of the image. But there are many other methods which could be tried to compare the results.

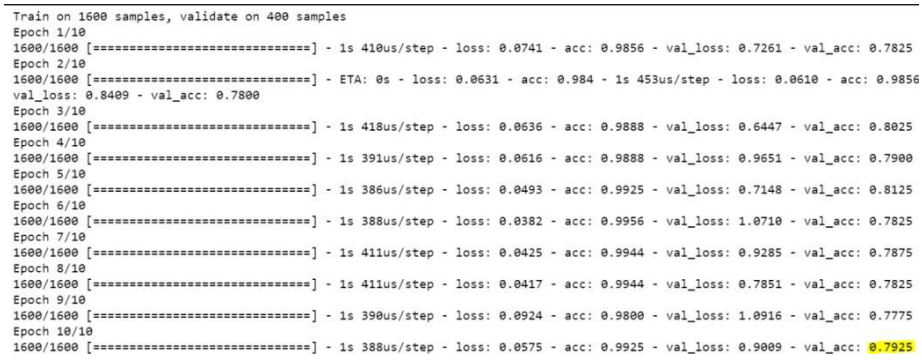


Figure 4:Final accuracy gained

Chapter 5: Conclusion and Future work

We are exposed to various techniques of image processing including OCR and other methods to extract text. We were exposed to ML in this project, also to deep learning. We are able to identify the difference and come to know how we are going to recognize images and how the world uses machine learning to understand the world more by identifying so many images.

In this study convolutional neural network is used to detectand classify text from images. The Neural Network istrained using the images taken in the natural environment and achieved 79% classification ability. This shows the ability of CNN to extract important features in the naturalenvironment which is required for plant disease classification.

We propose a text detection method to differentiateimages embedded with sufficient text (term as textparagraph images) and other images. Since the quantity oftext paragraph images is much smaller compared with non-textparagraph images through the internet, this technique canhighly reduce government's or internet service provider's laborin blocking defamation or illegal commentaries which areconveyed by images. After text paragraph image detection, thegovernment agencies or internet service providers only need tofocus on a small portion of suspected images.

Every approaches its own benefits and restrictions. Even though there are many numbers of algorithms, there is no single unified approach that fits for all the applications. [21] The future work mainly concentrates on developing an algorithm for exact and fast text extraction from an image.

Future work can be done in defining this module for language identification and then its extraction. Identifying language from a multi-lingual document and then its conversion to different languages will become a proficient work in these fields. So, such methods should be explored more to get a better hold to these techniques.

LIST OF REFERENCES

[1] Abdelmalek Amine, ZakariaElberrichi, Michel Simonet, ' AUTOMATIC LANGUAGE IDENTIFICATION: AN ALTERNATIVE UNSUPERVISED APPROACH USING A NEW HYBRID ALGORITHM' , International Journal of Computer Science and Applications, ©Technomathematics Research Foundation Vol. 7, No. 1, pp. 94 – 107, 2010.

[2] S.BanuChitra, DrR.S.Vetrivel, 'TEXT RECOGNITION USING DIGITAL IMAGE PROCESSING TECHNIQUES', International Journal of Computer Engineering and Applications, Volume XII, Issue I, Jan. 18.

[3] Tanmay A. Wagh, R. M. Samant, Sharvil V. Gujarathi, Snehal B. Gaikwad," Grapes Leaf Disease Detection using Convolutional Neural Network" published at International Journal of Computer Applications (0975 – 8887) Volume 178 – No. 20, June 2019.

[4] Bin Liu 1, Yun Zhang," Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks" published at 29 December 2017, article.

[5] SharadaPrasannaMohanty, David Hughes, and Marcel Salathé," Using Deep Learning for Image-Based PlantDisease Detection" published on April 15, 2016

[6] Ms. N. Geetha, Dr. E. S. Samundeeswari, ' Image Text Extraction and Recognition using Hybrid Approach of Region Based and Connected Component Methods' , International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181, Vol. 3 Issue 6, June – 2014

[7] C.P. Sumathi, T. Santhanam and G.Gayathri Devi, 'A SURVEY ON VARIOUS APPROACHES OF TEXT EXTRACTION IN IMAGES', International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.4, August 2012

[8] Y. Zhan, W. Wang, W. Gao (2006), "A Robust Split-And-Merge Text Segmentation Approach For Images", International Conference On Pattern Recognition,06(2):pp 1002-1005.

[9] Thai V. Hoang , S. Tabbone(2010),"Text Extraction From Graphical Document Images Using Sparse Representation"in Proc. Das, pp 143–150. International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.4, August 2012 41

[10] Audithan,,R.M.Chandrasekaran (2009), "Document Text Extraction From Document Images Using Haar Discrete Wavelet Transform",European Journal Of Scientific Research, Vol.36 No.4 , pp.502-512.

[11] Sachin, Grover,Kushal Arora,,Suman K. Mitra(2009),"Text Extraction From Document Images Using Edge Information",IEEE India Council Conference.

[12] V.Vijayakumar,R.Nedunchezhianm(2011),"A Novel Method For Super Imposed Text Extraction In A Sports Video",International Journal Of Computer Applications,Volume 15– No.1.

[13] Min Cai, Jiqiang Song, Michael R. Lyu(2002),"A New Approach For Video Text Detection",Proceedings International Conference On Image Processing , Volume 1, pp: I-117-I120.

[14] Yih-Ming Su, Chaur-Heh Hsieh(2006), "A Novel Model-Based Segmentation Approach To Extract Caption Contents On Sports Videos", IEEE International Conference On Multimedia And Expo,pp:1829 - 1832 .

[15] Miriam Leon, Veronica Vilaplana, Antoni Gasull, FerranMarques(2009) , "Caption Text Extraction For Indexing Purposes Using A Hierarchical Region-Based Image Model", ,Proceedings Of The 16th IEEE International Conference On Image Processing, pp:1869-1872.

[16] S. A. Angadi , M. M. Kodabagi(2009) , "A Texture Based Methodology For Text Region Extraction From Low Resolution Natural Scene Images ", International Journal Of Image Processing (Ijip) Volume(3), Issue(5).

[17] Yi-Feng Pan, XinwenHou, Cheng-Lin Liu(2009), "Text Localization In Natural Scene Images Based On Conditional Random Field," ICDAR,pp 6-10.

[18] .J.Fabrizio, M. Cord, And B. Marcotegui(2009), "Text Extraction From Street Level Images,", CMRT, Vol. Xxxviii, Part 3/W4 , pp. 199–204.

[19] Kohei Arai1 , Herman Tolle(2011)," Text Extraction From Tv Commercial Using Blob Extraction Method", International Journal Of Research And Reviews In Computer Science Vol. 2, No. 3

[20] Wonder Alexandre Luz Alves And Ronaldo Fumio Hashimoto(2010),"Text Regions Extracted From Scene Images By Ultimate Attribute Opening And Decision Tree Classification", Proceedings of the 23rd Sibgrapi Conference On Graphics, Patterns And Images.

[21] Super data science site, Convolution neural network,'Flattening Image', https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening

[22] medium website platform,Machine learning-Deep learning,'Convolution Layer Image'https://medium.com/machine-learning-bites/deeplearning-series convolutional-neural-networks-a9c2f2ee1524

[23] Quora, 'What is Maxpooling in CNN Artical, "Maxpooling Image", https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

[24] Veena Bansal, R.M.K.Sinha, Segmentation of touching characters in devanagri, 2014.

[25] Sinha, R.M.K, Mahabala, H.N.: Machine Recognition of devanagri Script IEEE Trans.sys. Man Cybern (1979)

[26] Pal, U., Chaudhuri, B.B.; Indian script recognition: a survey, Pattern Recognition, 1887-1899(2004)

[27] 'English Text Image', https://www.lovesove.com/cards/wp-content/uploads/2016/07/ahankar-aatmvishwas-hindi-suvichar-image-lovesove.jpg

[28] 'Hindi Text Image' https://statusprince.com/FinalPosterSave/UserFiles/StatusPrince_5e3a59dd925d2.jpeg

[29] 'Caption Text Image', https://i.pinimg.com/originals/9f/00/d5/9f00d5862734fd010cec5555c3b5939b.jpg

[30] 'Document Handwritten Text Image', https://blog.4tests.com/wp-content/uploads/2015/03/handwriting.jpg