MULTILINGUAL OCR APP FOR MOBILE

Aviral Srivastva [1], Prof. Naveen Kumar N [2]

[1]Student, School of Computer Science, VIT University, Vellore, Tamil Nadu, India
[2]Associate Professor, School of Computer Science, VIT University, Vellore, Tamil Nadu, India

Abstract. The need for a mobile application increase as we move towards different regions. This app aims to provide a platform for people to understand any language apart from their own language. The app work like interested person may post a text or image and specifications of the language involved. The characters present in the text or image will be abstracted by using OCR and they will be converted to the desired language. The language set include all the major languages as well as regional languages of India. The app works on the principle of image processing . The image will read each character using OCR and converted it to desired language by using Ocrspace API and Yandex API. This is very helpful for people going to a place where they have no knowledge of the local language .The language set has around 90 languages spoken across the globe.

1. INTRODUCTION

This app is for people who wish to convert their form of conversation from one language to another.eg English to Spanish. The input of this app is any handwritten or digital image, which is then transformed using image processing algorithm to a gray scale image which is then passed to the other processing part of the software which is responsible for generating the required output in the desired language. The user may be registered or not , but he can sign in as a guest . Here we only he need to do is to scan a language and wait for the output which will be displayed shortly as the processing time of our app is 30 ms.

1.1 Scope/project goals

The project aims to provide a platform for prospective buyers and sellers and for travelers to exchange their thoughts and feelings in which they are interested in and hence it is very important in today's digital society where cross cultural meetings are getting encouraged and increasingly popular. The following will be the specifications of the app:

1. Interested person may post a text or image and specifications of the language involved. The person interested to communicate with the person can look up the application on our store and search for the contact details on the profile of the communicator. Every person can register and log in on the application whenever he wants.
2. The features in the application can be viewed and used by using the Google Play store and downloading the app from their online store .
3. A user needs to register himself first, add his basic information such as username, address, contact details etc.

4. After the initial registration, the person can login to the page by using just the username and the password.

1.2 Project Perspective

First open Google Play Store in your respective mobiles. Then search for Multilingual OCR Reader. The first option which you see is the app you are searching for. A pop –up will arrive to download the app. Click download and wait till download is completed. Once it is completed another pop up will appear and then click the install button if your mobile setting are not compatible then go to settings and uncheck the "install fom unknown sources". Once it is over you will automatically get a icon on your homescreen. Click on the icon and wait for some time. You will be directed to our option page. Scan the image and enjoy our free app.

2. GENERAL ALGORITHM FOR CLEAR TEXT IMAGE

The algorithm used is the simple image processing of RCG to GRAY scale conversion. In this the pixel value will be taken for processing and a appropriate edge detection mask will be applied to them such as Sobel, Prewitt operator so as to find out the edges. Once a particular pattern is observed it is fed to skeletonization which abstracts its middle features and generate a text. The text so generated is fed to the other section of the code which simultaneously converted to desired language by accessing our cloud database located on our laptops by using the Yandex API. Digital image process is employed for laptop algorithms to form, process, communicate, and to show digital image. Digital image process algorithms generally require to measure some accustomed convert signal from image device to the digital pictures, Improve the clarity, and take away the shrie and alternative object, Extract their sizes, scale, or range of the objects in an exceedingly scene , Prepare image for the show and printing, Compress the photographs to speak across our network.[1]

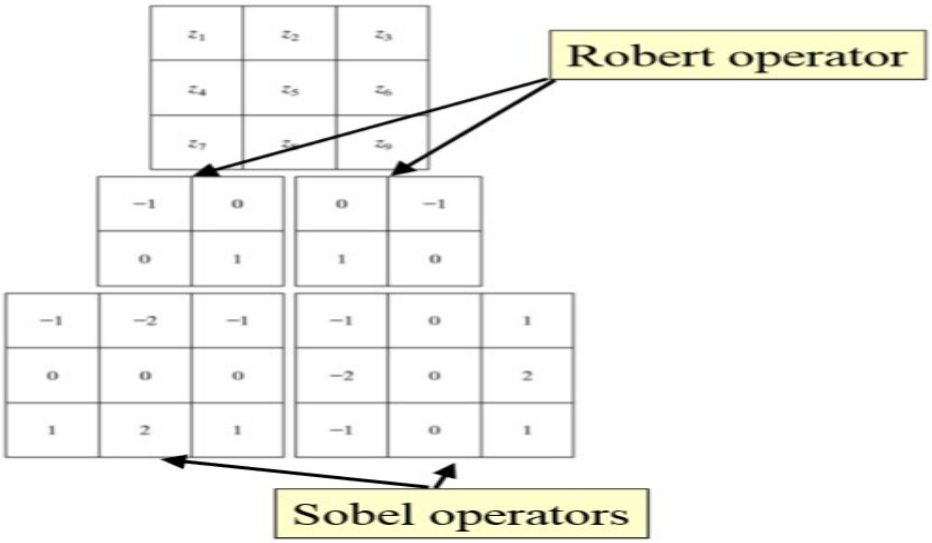

Figure 1--Applying Sobel and Robert Operator

2.1 OCR techniques employed in this algorithmic program[2]

- Matrix Matching converts each character into a pattern among a matrix, then compares the pattern with associate index of famous characters. Its detection and recognition is strongest for the case of monotype and it is uniform single column pages. Mathematical logic may be a multivalent logic that allows intermediate values to be printed between typical evaluations like yes/no, true/false, black / white etc. a trial is made to attribute a extra human-like approach of mention at intervals the programming of computers. Mathematical logic is been used once answers do not have an exact true or false values and there unit of measurement uncertainly involved.

- Feature Extraction defines each character by the presence or absence of key choices, beside height, width, density, loops, lines, stems and totally different character traits. Feature extraction may be a glorious approach for OCR of magazines, device print and top of the range footage.

- Structural Analysis identifies characters by examining their sub feature shapes of the image, sub-vertical and horizontal histograms. Its character repair capability is good for caliber text and newsprints.

- Neural Networks simulates the approach the human neural system works; it samples the constituents in each image and matches them to a famous index of character component patterns. the ability to acknowledge the characters through abstraction is good for mounted documents and broken text. Neural networks unit of measurement ideal for specific sorts of problems, kind of like method stock exchange data or finding trends in graphical patterns.

3. CHALLENGES IN INPUT AND OUTPUT SPACE

3.1 Recognising Text for Unclear and Blurry Images

Here we have introduced an alternative approach to text localization based on the fact that it is often useful to localize text that is identifiable as text but too blurry or small to be read. For time period applications like a mobile app to seek out and browse text, text could at the start be noninheritable from a lower-resolution video image within which it seems too little to be read; once the text's presence and site are established, a higher-resolution image will be taken so as to resolve the text clearly enough to scan it.[3]

We demonstrate proof of conception of this approach by describing a completely unique formula for binarizing the image and extracting candidate text options, referred to as "blobs," and grouping and classifying the blobs into text and non-text classes. Experimental results ar shown on a spread of pictures within which the text is resolved too poorly to be clearly scan, however remains specifiable by our formula as text.[4]

The question image is binarized and therefore the candidate text options, referred to as "blobs", ar extracted. The blobs ar assigned a possible of being text supported histograms of geometric properties of blobs, learned from the coaching knowledge. Neighboring blobs live} sorted into "superblobs" supported a similarity measure. The ensuing superblobs ar classified into text and non-text classes.[5]

A blob could be a assortment of connected horizontal line segments. A horizontal line phase, L_{yi}, is outlined by its y-coordinate, y, and beginning and ending x-coordinates: x_1, and x_2. For a given binary image, one will notice one array of horizontal line segments for every y-coordinate, AL_y. A blob is outlined by a group of connected L_{yi}, that's conjointly organized into a bunch of AL_y in keeping with their y-coordinates.[6]

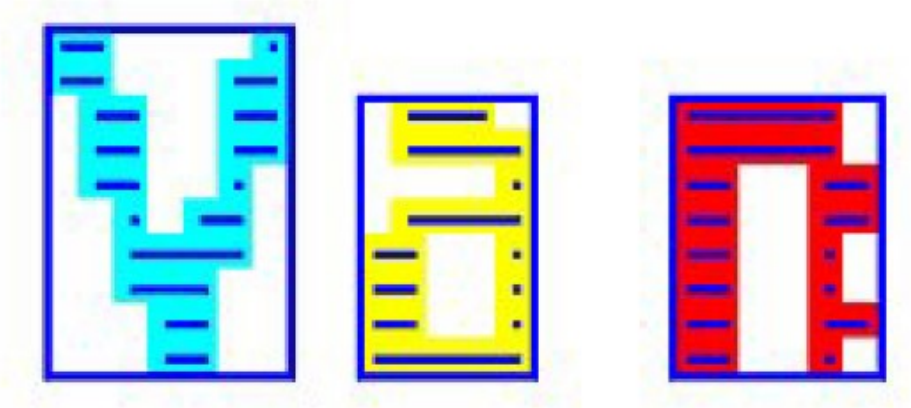

Figure 2-Blobs are a group of horizontal line segments to detect text

Here the horizontal line segments in figure-1 are used to detect letters.
In each blob, there are many horizontal line segments, L_{yi}, represented by blue lines, that are connected. For a given y, AL_y, the array of L_{yi} can have one or more L_{yi}.

Algorithm for Blob Detection [7]

1. For each row, find an array of horizontal line segments, indexed by row number (AL_y).
2. At the end of Step 1, we get an array of line segment arrays (ALA) representing the image.
3. Initiate a blob with one line, L_{yi} from first AL_y of ALA, and remove L_{yi} from ALA.
4. Sweeping downward: for each y where the blob has any line segment, and for each line in the line array, search AL_{y+1} in ALA for connected lines. If a line segment is found, add it to the blob at (y+1), and remove it from ALA.
5. Sweeping upward: for each y where the blob has any line segment, and for each line in the line array, search Al_{y-1} in ALA for connected lines. If a line segment is found, add it to the blob at (y-1), and remove it from ALA.
6. Repeat steps 4 and 5 until no new line segment is added for either step 4 or 5. Store the blob.
7. Repeat steps 3 - 6 until ALA is exhausted.

Once the blobs from the image are extracted, each of those is assigned a unitary potential of being text. We use histograms of geometric properties of the blobs learned from training images to determine the potential. The blob is a bitmap, whose top left corner is (x_{left}; y_{top}), widthW and heightH; and the geometric center of the bounding box is given by (x_c; y_c).

The blobs area unit divided in to 2 teams, ”long” and ”not long” blobs, supported the ratio. Blobs with ratio nine zero.35 area unit treated as ”long”. exploitation the coaching blobs, a bar chart of every of those properties is made, one by one for long and shortly blobs, one every from positive and negative examples. The positive blobs' bar chart offers the chance of a blob being text, and therefore the negative blobs' bar chart offers the chance of a blob being not text. If a question blob is long, then the bar chart set learned from long blobs is employed.[8]

Since we tend to area unit reaching to notice words of text, we tend to cluster the neighboring blobs that area unit “similar” into “superblobs”. The similarity live of 2 neighborhood blobs is predicated upon (a) five hundredth or a lot of similarity tall and (b) ninetieth or a lot of horizontal alignment either at the highest or very cheap, i.e. the distinction within the ytop (or the distinction within the ybot) values of the 2 blobs ought to be but 100 percent of either blobs' height. so the stress is on horizontal alignment. Of course, 2 blobs ought to be neighbors of every different to be thought of similar. A blob is alleged to be almost like a superblob, if the blob is analogous to any of the blobs within the superblob.[9]

The grouping algorithmic rule starts by making the primary superblob with the primary blob. At every more iteration, for every blob within the image, the algorithmic rule goes through all the superblobs up to now. If it's almost like any of these superblobs, add the blob to it superblob. If the blob is analogous to quite one superblobs, then, all those superblobs area unit unified in to at least one superblob. If none of the present superblobs area unit almost like the present blob, then a replacement superblob is formed with the present blob alone. The iteration is recurrent till all the blobs within the image area unit analyzed. every ensuing superblob corresponds roughly to one word.[10]

This step corresponds to classifying every ensuing superblob as text or not text. The potential of a superblob being text is outlined because the multiplication of the potentials of the individual blobs it contains. every superblob is assessed as text or not text supported (a) Its potential and (b) the quantity of the nodes contained in it, and (c) the general space magnitude relation of the supernode. The potential of the superblob must be larger than a threshold. the edge on the potential of a superblob could be a perform of the quantity of blobs contained in it. we tend to expect words to possess quite one letter, thus 2 or a lot of “similar” blobs next to every different within the image to possess a high chance of being text as critical one isolated blob. Naturally, the upper the quantity of blobs in a very superblob, the smaller the edge. Also, the general space magnitude relation of the superblob is taken in to account whereas clasifying. the general space magnitude relation of the superblob must be larger than an exact threshold for the superblob to be classified as text.[14]

RESULTANT OUTPUT AFTER APPLYING ALGORITHM

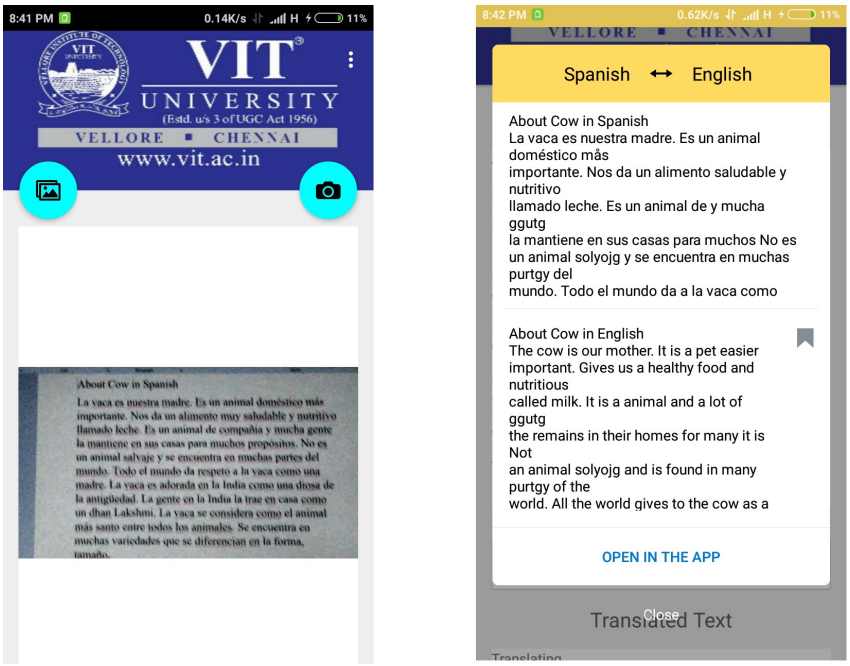

Figure 3-Required Output

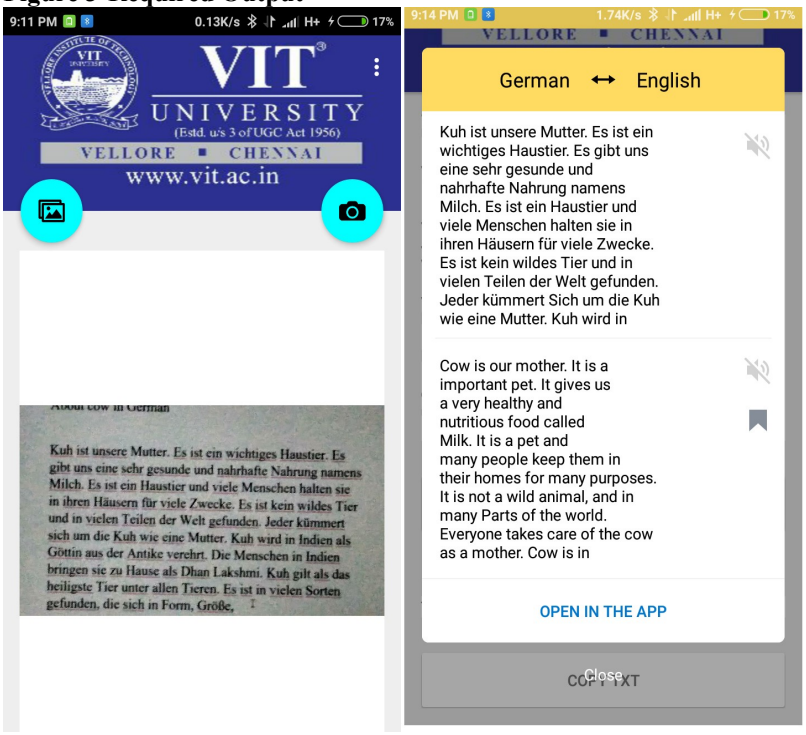

Figure 4 Required Output

4. CONCLUSION

Our approach , addresses the requirement for a multilingual OCR in Indian script setting. one recognition of the system, however comprising of the multiple OCR engines was additionally used. The crux of our complicated resolution lies in police work the derived script before the popularity. This enabled every of the word to be recognized on associate OCR reader, trained completely for every script..

For the part of images which are unclear and blurred ,the above algorithm searches for text in one polarity, i.e. light text on a dark background. To search out dark text on a light-weight background, the algorithm is run with the image distinction reversed. Each of the results could be shown to mark solely for light-weight text in the given image. Hence, dark text on light-weight background will not be marked, and not thought of as lost positives. Note that the algorithm detects text even at a very low resolution (height _ 7), which may not be read even by humans.

Though, not every low resolution text region is detected, this shows the effectiveness of the algorithm. The false positives are solely a priority with regards to the runtime speed of the general rule(detection and OCR recognition) and not with regards to the accuracy, because the false positives are eliminated at the end.

With web based system working within the place, we might get an opportunity to judge our processed system on large choice of document. Multilingual code development is important for code developers nowadays to stay competitive within the world and for the technology trade connected through the cloud network. The attention on the importance of Multilingual Code Development, each for industrial gains associated to satisfy the wants of an ever increasing doctrine and multilingual user base, should be governed by the fact of vast variety of people living in society from different parts of the world. A diverse list of tips for multilingual code development that cross cut all the phases of code development life cycle is compiled and unconditionally given during this paper.This includes tips relevant to multilingual code development normally, body and money implications, practicableness study, needs engineering, design and style, code programming, and code testing. Also, tips for multilingual code development for existing code product (post release) and also the most ordinarily used technologies area unit given.

By applying all the algorithm related to unclear text image in the OCR API leads to detection of text from unclear image , so that we can detect text from unclear image using the app.

5. REFERENCES

1. R. M. K. Sinha and H. Mahabala, “Machine recognition of devnagari script," in *IEEE Trans. on Systems, Man and Cybernetics*, (1979)

2. D. Arya, T. Patnaik, S. Chaudhury, C. V. Jawahar,B.B.Chaudhuri, A.G.Ramakrishna, C. Bhagvati, and G. S. Lehal, “ Experiences for Integration and Performance Testing for Multilingual OCR for Printed Indian Scripts,” in *J-MOCR Workshop,ICDAR*, (2011)

3. K. Aparna and A. Ramakrishnan, “A complete tamil optical character recognition system,” in *DAS*, (2002)

4. J. Gao and J. Yang. An adaptive algorithm used for text detection from natural scenes. In Proc. CVPR, pages 84–89, (2001).

5. A. K. Jain and B. Yu. Automatic text location in images and video frames. Pattern Recognition, International Conference on, 2:1497,(1998).
6. H. Li, H. Li, D. Doermann, D. Doermann, O. Kia, and O. Kia. Automatic text detection and tracking in digital video. IEEE Transactions on Image Processing, (1998).

7. A. K. Singh and C. V. Jawahar, “Can RNNs reliably separate script and language at word and line level?” in *ICDAR*, (2015.)

8. Premkumar S. Natarajan and Ehry MacRostie and Michael Decerbo, “The BBN Byblos Hindi OCR system,” in *DRR*, (2005).

9. A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, (2006).

10. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, (1997).

11. C. V. Jawahar and Anand Kumar, “Content-level Annotation of Large Collection of Printed Document Images,” in *ICDAR*,(2007).

12. A. Jain and B. Yu. Automatic text location in images and video frames. In Proc. 14th Intl. Conf. on Pattern Recognition, vol. 2, pp.1497-1499, (1998).

13. B Epshtein, E Ofek, Y Wexler, Detecting text in natural scenes with stroke width transform, in IEEE Conference on Computer Vision and Pattern Recognition. pp. 2963-2970, (2010).

14. YAO, Cong, et al. Detecting texts of arbitrary orientations in natural images. In: Computer Vision and Pattern Recognition (CVPR), pp. 1083-1090, (2012).

15. Matas, J. and Galambos, C. and Kittler, J.V., Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. CVIU 78 1, pp.119-137, (2000).

16. S. Yonemoto, An interactive image rectification method using quadrangle hypothesis, Proc. of International Conference on Image Analysis and Processing (ICIAP2013), pp.51-60, (2013).

17. Clark, P, and M. Mirmehdi. Estimating the orientation and recovery of text planes in a single image. Proceedings of the 12th British Machine Vision Conference, (2001).

18. Yin, Xu-Cheng, et al. Robust Vanishing Point Detection for MobileCam-Based Docu-ments. Document Analysis and Recognition, International Conference on. IEEE,(2011).