International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

Analysis of a method of Video Preprocessing with the help of Optical Character Recognition And Natural Language Processing

B.Tech. Scholar Samarth Bajaj, Assistant Professor Viplove Divyasheesh
Dept. of Computer Science & Engineering
Indore Institute Of Science And Technology, Indore MP, India
samarthbajaj005@gmail.com, viplovedivyasheesh@gmail.com

Abstract **- This research paper mainly focuses on the algorithm of searching for videos on YouTube, Daily Motion, Crackle, etc. At present whenever we want to search a video on any platform we search it by the name of the video and the contents that are given by the video maker, we are not able to search a video on the basis of the theoretical contents in that video. Though we can easily search a video if it is not based on theoretical content but if the video is purely theoretical and we are searching it for the first time then at present we cannot search that video by the content shown in that video. So, in this paper, we will understand how it is possible to search a video on the basis of the theoretical content in that video.**

Keywords- **Video Preprocessing, Optical Character Recognition, Tesseract**

I. INTRODUCTION

YouTube helps its user by giving recommendations on the basis of User's search, but what if user wants to search a video by its content. In this paper we will understand how we can make it possible.The easiest way to make it possible is to give choice to the person who is posting a video that his/her video contains theoretical data or only visual data, if he selects theoretical data then the application will suggest him to upload a text file which will contain all the theoretical data mentioned in that video.

After this our algorithm will work and we would take out some key factors from that text file which could be searched by a user. Now, our second option is that we can covert the content of the video to text with the help of OCR (Optical Character Recognition). In this we will capture every distinct image possible from that video and then each image will be converted into a text file, then after that our algorithm will check by how many distinct ways a user can search for that video.

Last but not the least, we can convert our video to text with the help of video transcription which is now very easy to implement but we cannot use this approach because what if the person who has made that video does not specifically repeat the theoretical content written in that video. So this solution is not that much efficient and accurate.

II. METHODOLOGY

The first important thing about an application is that it should be user friendly, the end user must have less work that means he should be served as simpler as we can.
So, we have analyzed all the three methods to found out which method the end user likes.
if you(user) wants to upload a video on YouTube which contains theoretical or written content in it then we have two approaches-

1. User should upload a document attached with that video which contains all the content written in that video, so that whoever searches for that content should get that video in results.
2. User only have to upload the video, and the application will process your video before uploading it and extract the written content in that video, so that whoever searches for that content should get that video in results.

After analyzing both the approaches we found that for a good user experience we must follow the second approach. In this approach the user will only upload the video, rest all the things will be done by the application.

III. SYSTEM OVERVIEW

The overall working of our video preprocessing is illustrated in Figure-1. It is purely based on the working of OCR (Optical Character Recognition) and NLP (Natural Language Processing). First we generate an API (Application Programming Interface) request to retrieve data (image) from a video, then that image is given as an input to a pre-processor, with the help of pre-processor we do some improvement of our image data that minimizes

some unwanted distortions and enhances some image features important for further processing. After the preprocessing of the image, the next step is to give Trained Data to the Tesseract-OCR Engine; a lot of experiments should be performed during training data preparation. These experiments are necessary to find out the right combination of the training data that provides highest accuracy during recognition. The automated generation of the training data helps us to avoid the difficulties of collecting the large number of data units. We prepare different sets of training data with following parameters.

1. Segmentation
2. Degradation
3. Type of document image
4. Image DPI information
5. Font type and size

Now after this we give this training data to Tesseract - OCR engine. The next job is to prepare Tesseract supported image - In this we generate an image with Tesseract specific encoding. Tesseract is only capable to read an uncompress 1bit/8 bit tiff format image, after this we generate an uncompressed 1 bpp(bit per pixel) tiff image with the help of segmentation information that we have obtained from the previous stage.

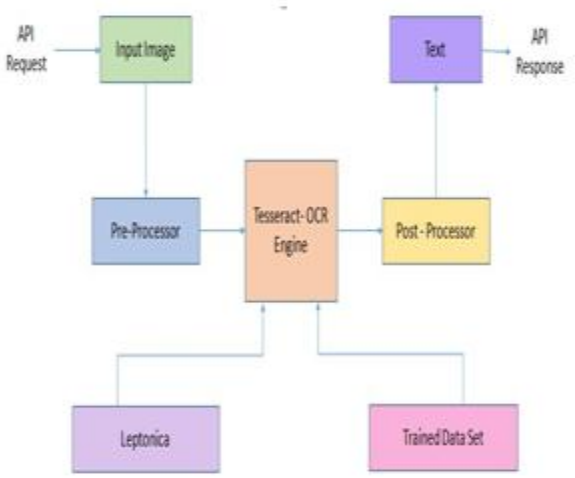

Fig.1. OCR process flow.

Now at last we will apply a two level post processing, Where first post processing will be done on raw text obtained from the previous stage, and second post processing will be applied on the output of first level postprocessor output to check the spelling mistakes and in this we will also have suggestions for misspelled words.

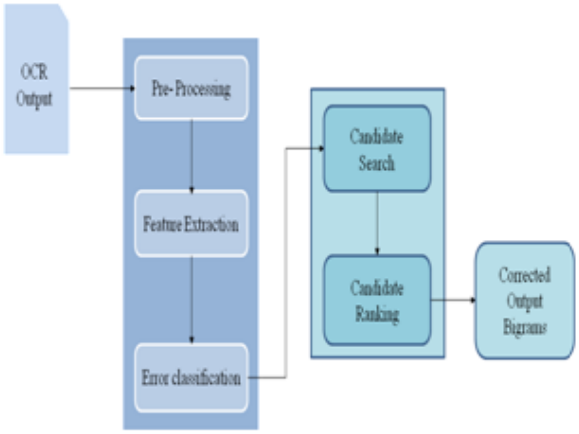

Fig.2. Processing the OCR output.

The first level post processing performs two subtasks Where firstly it arrange the Unicode text and then correct known mistakes that the system has recognized by following specific rules. The system can save the output text in several text file format depending on users choice because the output text is saved as Unicode text and hence enables further editing. The output text obtained above is then further processed with Natural Language Processing to remove unnecessary words in the text. Now after applying NLP our text is ready. This text is then attached to the video and whenever user searches with the content written in that video then he/she gets required results.

IV. CONCLUSION

In this paper, we present the complete process of video preprocessing with the help of Tesseract OCR Engine. We have discussed the Methodology as well as how our system will work. A large amount of data is required for this application and some experiments need to be performed with these training data to achieve good accuracy. Also we are using two level post processor to improve the performance and reliability of application.

REFERENCES

[1]. S.V. Rice, F.R. Jenkins, T.A. Nartker, The Fourth Annual Test of OCR Accuracy, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.
[2]. S.V. Rice, G. Nagy, T.A. Nartker, Optical Character Recognition: An Illustrated Guide to the Frontier, Kluwer Academic Publishers, USA 1999, pp. 57-60
[3]. Ray Smith, "An Overview of the Tesseract OCR Engine".
[4]. http://code.google.com/p/tesseract-ocr/. Last accessed: February 1, 2020.

[5]. Thomas A. Nartker, Stephen V. Rice, and Junichi Kanai. OCR accuracy:
[6]. UNLV's second annual test. Inform, Association for Information and Image Management. 8(1):40+, January 1994.
[7]. Md. Abul Hasnat, S M Murtoza Habib and Mumit Khan. "A high performance domain specific OCR for Bangla script", Int. Joint Conf. on Computer, Information, and Systems Sciences, and Engineering (CISSE), 2007.
[8]. R.W. Smith, The Extraction and Recognition of Text from Multimedia Document Images, PhD Thesis, University of Bristol, November 1987.
[9]. Stephen V. Rice, Junichi Kanai, and Thomas A. Nartker. The third annual test of OCR accuracy. Technical Report 94-03, Information Science Research Institute, University of Nevada, Las Vegas, April 1994.