Video OCR for Indexing and Retrieval

Mohd Javed Khatri
Department of Computer Engineering,
KJ Somaiya College of Engineering,
Mumbai, India

Abhishek Shetty
Department of Computer Engineering,
KJ Somaiya College of Engineering,
Mumbai, India

Ajay Gupta
Department of Computer Engineering,
KJ Somaiya College of Engineering,
Mumbai, India

Grishma Sharma
Department of Computer Engineering,
KJ Somaiya College of Engineering,
Mumbai, India

ABSTRACT

We present an implementation of a search engine that searches videos based on its textual content. The system consists of four parts video processing, spell correction, indexing and searching. The video processing is done by dividing the video into frames and extracting text out of it. Lecture videos, news having some textual content in it show good results.

General Terms

Optical Character Recognition (OCR), Connected Components (CCs), Video Search Engine.

Keywords

Indexing, Retrieval, Video, OCR, Searching, Search Engine, Apache Solr.

1. INTRODUCTION

The multimedia data is growing at a rapid speed, but still today there’s no convenient method to search videos if they are not named properly. Video search engines like YouTube, Dailymotion etc. index and search videos based on several parameters like title, description, meta tags, ratings etc. Sometimes, in order to get more number of views, a fake name and description is given to the video which improves the unnecessary visibility of the video in the search results. So to tackle this problem we propose a new approach in this paper wherein we actually analyze the textual content of the video and index videos based on that text. We apply OCR to each and every unique frame of the video. Since OCR does not give 100% correct results, we correct the text before indexing it. This approach can be used by websites like Coursera, Udemy etc. to enhance their search. Other than this, we can also index movies with sub titles. Even news shows produce some good results.

The drawback with this approach is the video processing speed. It depends on various factors like video format, resolution, textual content. It takes much time to process the video if it has more textual content. This can be the tradeoff if there’s large amount of video data. But the video processing speed can be improved by processing the frames in a parallel manner. This improves the performance to a large extent.

2. TEXT FEATURES

Text can appear anywhere in the video. They sometime contain important data. There are some distinct features of text which appears in the video so that they are easily readable:

- Characters contrasts with their background.
- Characters are monochrome.
- Characters font do not change from frame to frame.
- Characters have size restrictions.

3. ROBUST TEXT DETECTION

For doing OCR in video, the simplest way is by doing OCR on each frame. To increase the accuracy and performance we apply detection algorithm which is provided by [1], then to do OCR on binary image. Text detection algorithm proposed is very robust and can detect any kind of text, provided parameters are tuned properly.

MSER regions are extracted from the grayscale of the image. It is enhanced using canny edge detection on the input image. Geometric constrains are used on (CC) to filter out false positive. Stroke width transform using distance transform and those regions showing high variation in stroke width distance are filtered out.

3.1 Edge-enhanced MSER (Maximally Stable Extremal Regions)

Normally text do have significant constraint difference from its background and possess uniform intensity or color. That is why MSER is the best option for region detector. Refer fig 2.
MSER is robust against viewpoint, scale and lighting changes but sensitive to image blur. Small Text may go unnoticed. To work against blurred image, constrained properties of MSER and canny edge is used. This is done by taking gradient of an image and then pruning the MSER along the gradient direction i.e. towards the background. Refer fig 4.

3.2 Geometric Filtering

From previous stage, we got binary image. Where foreground CCs are considered as a textual regions. We filter out those regions which are having very large and very small area. We find out aspect ratio of each CCs using eccentricity. Normally CCs have aspect ratio close to 1.
Finally we filter out those CCs having large number of holes, this can be achieve by considering solidity or Euler Number as a parameter. Refer fig 5

3.3 Stroke width Transform by Distance Transform

Unlike the original approach introduced by [2], here distance transform is used. Euclidean distance is applied to each foreground pixel.

3.4 Duplicate Detection

Since most of the frames in video are redundant. Hence we need to remove all the similar detected text. We use SURF feature points to find the similarity between detected text in current frame and previous frame. If some similarity is found then text detected in current frame won't be given to OCR engine. Since SURF feature points are invariant to scale rotation, hence any animation done on texts, can still be detected by above method. Since all the frames are processed independently, it can be ran on parallel threads to increase the performance.

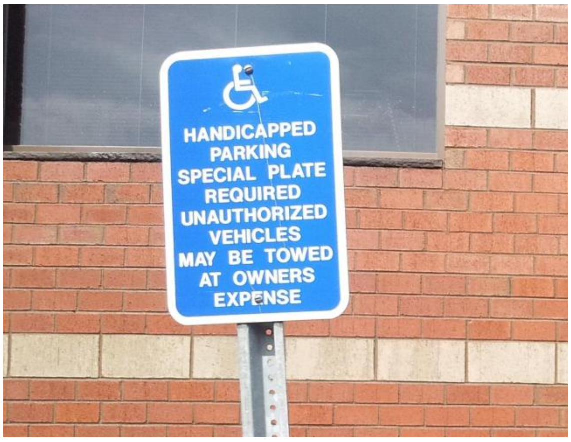


Fig 1: Original Image

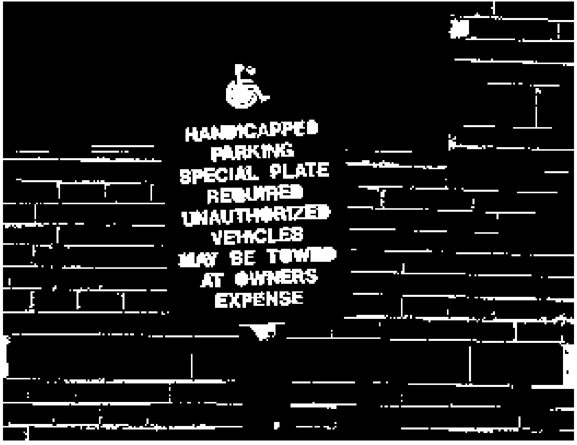


Fig 2: MSER Mask

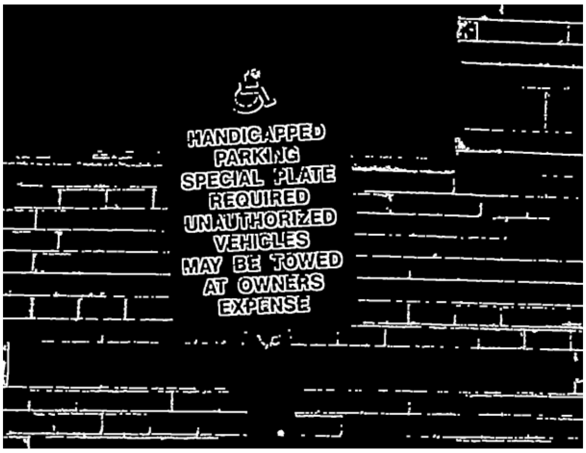


Fig 3: Edge Growing

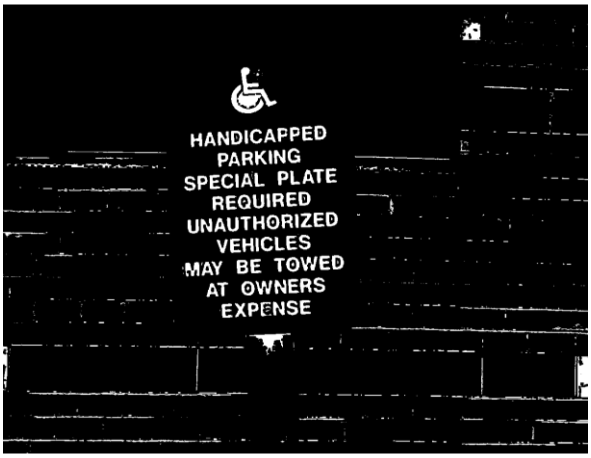


Fig 4: Edge Enhanced MSER

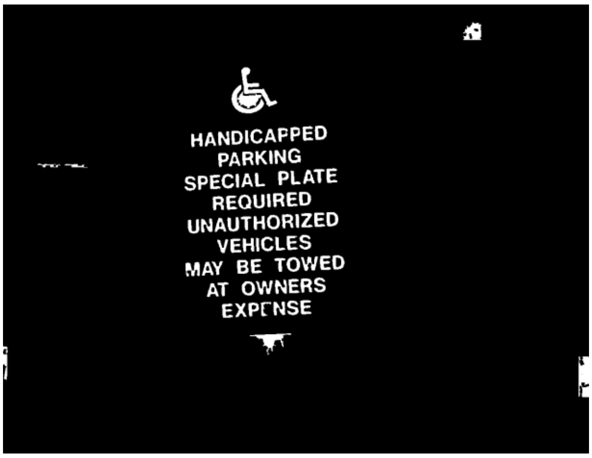


Fig 5: Region Filtering

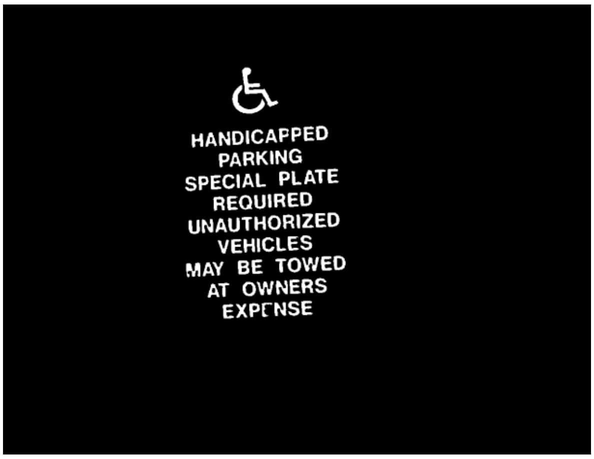


Fig 6: Final Image

4. POST VIDEO PROCESSING

It comprises of spelling correction, indexing and searching/retrieving the video.

4.1 Spelling Correction

The text extracted from the videos after applying OCR is not 100% correct. So, we correct text using Textblob, a python library. The spellCorrect function returns the confidence (0-100) and based on that we decide whether the corrected word is really correct or not.

4.2 Indexing and Searching

Video indexing and searching is done with the help of open source search platfrom Apache Solr.

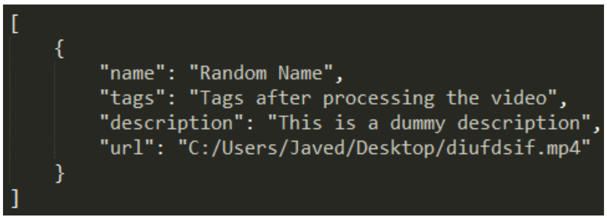
4.2.1 Why Solr?

Apache Solr is an enterprise search server based on Lucene. It was originally developed by CNET Networks as an in-house search platform. It's written in Java and runs on Application server. Some of the features of Apache Solr are:

- Advanced Full-Text Search Capabilities
- Optimized for High Volume Web Traffic
- Standards Based Open Interfaces - XML, JSON and HTTP
- Comprehensive HTML Administration Interfaces
- Server statistics exposed over JMX for monitoring
- Flexible and Adaptable with XML configuration
- Extensible Plug-in Architecture

4.2.2 Indexing

After spelling correction, the text is written to the JSON document. This JSON data is posted to the Solr server using HTTPPost. This indexes the video on the Solr server.

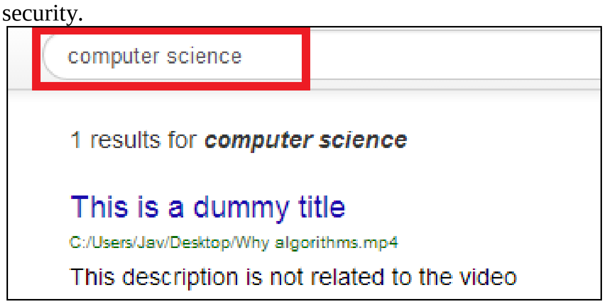


```
[
    {
        "name": "Random Name",
        "tags": "Tags after processing the video",
        "description": "This is a dummy description",
        "url": "C:/Users/Javed/Desktop/diufdsif.mp4"
    }
]
```

Fig 7: JSON format

4.2.3 Searching/Querying

In this, user specified terms are searched and ranked against the videos that are recorded in the index. Solr provides many functionalities, including keyword searching, search terms highlighting, parsing of the results, filtering and facet-based browsing. The videos are searched via REST interface that Solr provides and the responses are retrived in JSON format. This JSON data is again parsed by Solrstrap. It is probably the fastest available rendering engine for Solr. The reason being it does everything in HTML, CSS and JavaScript. JSON which is shot backed from the Solr server is directly interpreted by the web browser. Since there is no middle entity in between, it requires bandwidth lesser than standard search-middleware applications. It offers instant searching, infinite scrolling, faceted browsing. But at the same time it is not SEO friendly and needs clear access to /select/q= which weakens the security.



computer science

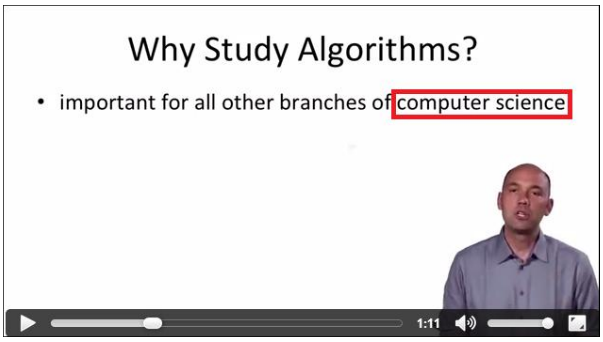
1 results for ***computer science***

This is a dummy title

C:/Users/Jav/Desktop/Why algorithms.mp4

This description is not related to the video

Fig 8: Search interface



Why Study Algorithms?

- important for all other branches of computer science

Fig 9: Video from the search result

In the fig. 7 and fig. 8 it can be seen that although the title and description of the video both are not related to the video but still the video having the text "computer science" is searched.

5. EXPERIMENTAL RESULTS

We have implemented the above algorithm on 100 different images, using MATLAB 2014b with 8 gb ram and 2.4 ghz processor, and following are the observations gathered:

Execution Time:

- Text Detection: 0.48938 sec (Avg.)
- Surf Detection and Matching: 0.21475 sec (Avg.)

Since the robustness of text detection entirely depend upon the threshold which we set, following tried and tested values gives good result for different scenarios:

Lectures & News:

- MSER regions area range: [10-500]
- Maximum Stroke width variation: 0.25
- Eccentricity Threshold: 0.995

Natural Scene Text:

- MSER regions area range: [50-5000]
- Maximum Stroke width variation: 0.45
- Eccentricity Threshold: 0.995

The algorithm proposed above gives an accuracy of 73% on an average.

6. CONCLUSION & FUTURE WORK

Text present in the video sometime provide very useful information for video searching and data mining.

In this paper we have proposed novel video text detection method using MSER and surf properties. MSER based text detection gives best result given that the thresholds are tuned properly. SURF feature points helps in detecting unique texts even if some animation are present in the video.

In future, we plan to assign weights to the text appearing in the video depending upon parameters like font size, text color etc. This will further lead to improved search results. Also a feature can be added wherein a user can directly jump to the desired keywords in the video

8. ACKNOWLEDGMENT

We would like to thank Prof. Grishma Sharma for her guidance and suggestion.

9. REFERENCES

[1] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk and Bernd Girod, “Robust Text Detection In Natural Images With Edge-Enhanced Maximally Stable Extremal Regions ,” IEEE International Conference on Image Processing (ICIP), 2011.

[2] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with strokewidth transform,” in CVPR, 2010, pp. 2963 –2970

[3] Liuis Gomes, Dimosthenis Karatzas, “MSER-based Real-Time Text Detection and Tracking,” IEEE Computer Society Washington, DC, USA, 2014.

[4] Marc Davis, ”Media Streams: Representing Video for Retrieval and Repurposing. Proc.,” ACM Multimedia 94, pp. 478-479, San Francisco, CA, USA, October15-20, 1994

[5] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, Inter-national Journal of Computer Vision, vol. 60, pp. 91–110, 2004.

[6] K. Jung, K. I. Kim, and A. K. Jain, “Text information extraction in images and video: a survey,” Pattern Recognition, vol. 37, no. 5, pp. 977 – 997, 2004.

[7] S. M. Lucas, “ICDAR 2005 text locating competition results,” in ICDAR, 2005, pp. 80 – 84 Vol. 1.

[8] C. Merino and M. Mirmehdi, “A framework towards real-time detection and tracking of text,” in CBDAR, 2007.

[9] M. Donoser and H. Bischof, “Efficient maximally stable extremal region (mser) tracking,” in CVPR, 2006.

[10] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, “Text detection and tracking for outdoor videos,” in ICIP, 2011.

[11] Liu, T. Choudhary, “Content Extraction and Summarization of Instructional Videos,” in IEEE, 2006

[12] Haojin Yang, “Lecture Video Indexing and Analysis Using Video OCR Technology,” in IEEE, 2011

[13] Zi Huang1 Yijun Li2 Jie Shao1 Heng Tao Shen1 Liping Wang1 Danqing Zhang3 Xiangmin Zhou1 Xiaofang Zhou1, “ Content-Based Video Search: is there a need, and is it possible,” in IEEE, 2008

[14] Julien Law-To, Rémi Landais, Gregory Grefenstette, “VOXALEADNEWS: A Scalable Content Based Video Search Engine,” in IEEE, 2012

[15] P.Geetha Vasumathi Narayanan, “An Effective Video Search Re-Ranking for Content Based Video Retrieval,” IEEE, 2011