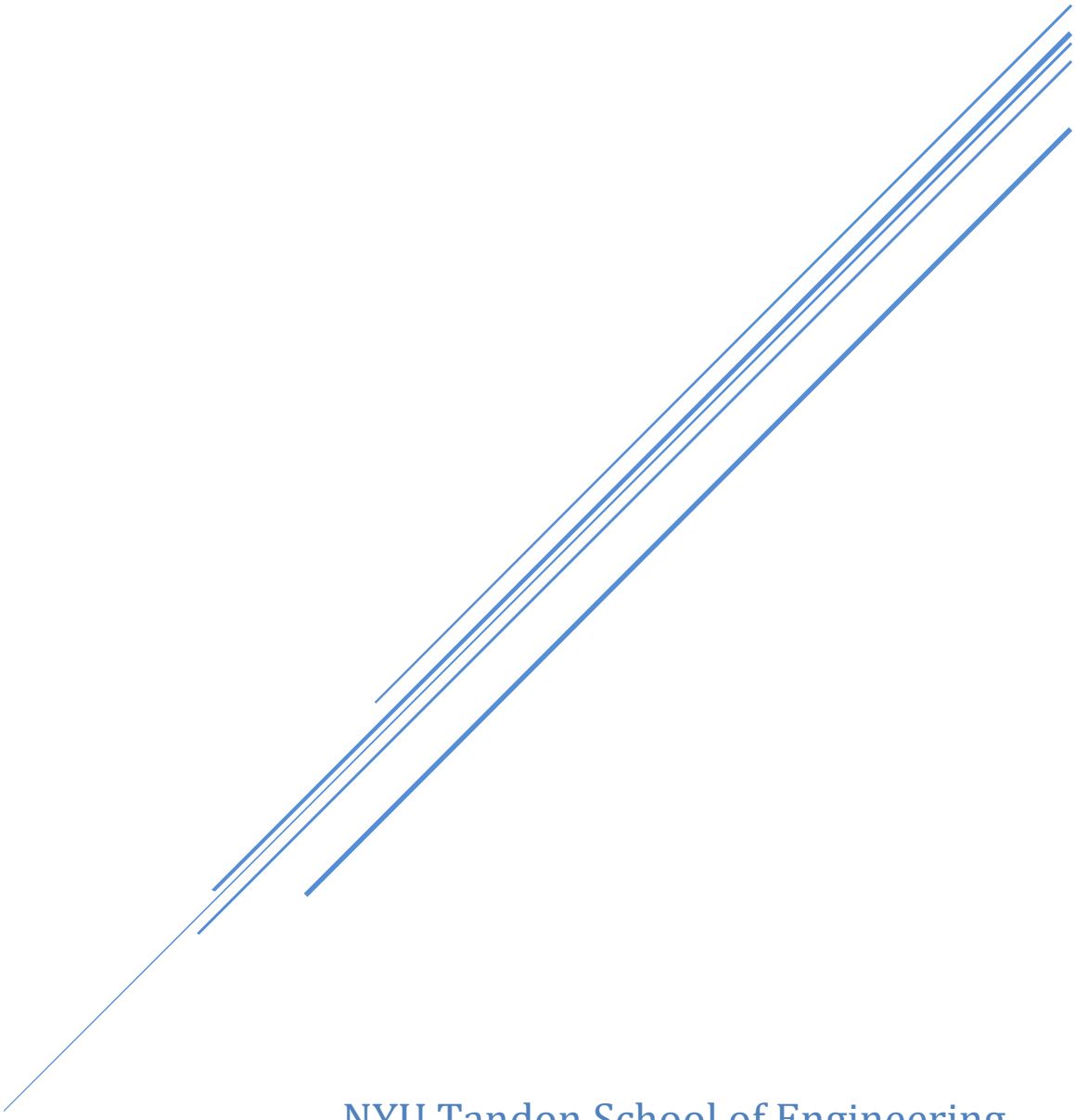
REAL TIME DETECTION OF TEXTS IN VIDEOS USING WEBCAM

By Akshta Suresh(N11857317)
&
Pooja Gupta(N17756215)



NYU Tandon School of Engineering
DSP Lab

Abstract - ***The influence of exponentially increasing camera-embedded smartphones all around the world has magnified the importance of computer vision tasks, and gives rise to a vast number of opportunities in the field. One of the major research areas in this field is the extraction of text embedded in natural scene images. Text detection in natural scene images is an important prerequisite for many content-based image analysis tasks. Natural scene images are the images taken from a camera, where the background is random, and the variety of colors used in the image may be diverse. When text is present in such type of images, it is usually difficult for a machine to detect and extract this text due to a number of parameters. This paper delineates a method using OpenCV and a deep-learning based text detector called EAST for text detection in live video. Here, a system is presented which can detect text from any source as shown by the user using a real-time webcam data.***

I. INTRODUCTION

Detecting text in constrained, controlled environments can typically be accomplished by using heuristic-based approaches, such as exploiting gradient information or the fact that text is typically grouped into paragraphs and characters appear on a straight line. Natural scene text detection on the other hand is much more challenging. Natural scenes suffer from low resolution, distortion, lack of illumination and low quality. All these lead to difficulty in detecting or recognizing texts from such scenes.

The primary reason for developing such a text detection system from natural scene environments is that it may contain some important information that cannot be reached by certain people because the scene may be blurred, reflective or partially obscured. The major steps for text detection are; processing the real time video and converting it into frames and detect the text regions using EAST.

Some other major challenges faced for the natural scene text detection are as follows:

1. Raw sensor image and sensor noise: In low-priced HIDs, pixels of a raw sensor are interpolated to produce real colours, which can induce degradations. De-mosaicing techniques, viewed more as complex interpolation techniques, are sometimes required. Moreover, sensor noise of a HID is usually higher than that of a scanner.
2. Viewing angle: Scene text and HIDs are not necessarily parallel creating perspective to correct.
3. Blur: During acquisition, some motion blur can appear or be created by a moving object. All other kinds of blur, such as wrong focus, may also degrade even more image quality.
4. Lighting: In real images, real (uneven) lighting, shadowing, reflections onto objects, inter-reflections between objects may make colours vary drastically and decrease analysis performance.
5. Resolution and Aliasing: From webcam to professional cameras, resolution range is large and images with low resolution must also be taken into account. Resolution may be below 50 dpi which causes commercial OCR to fail. It may lead to aliasing creating fringed artefacts in the image.

The main challenge is to design a system as versatile as possible to handle all variability in daily life, meaning variable targets with

unknown layout, scene text, several character fonts and sizes and variability in imaging conditions with uneven lighting, shadowing and aliasing. Our proposed solutions for each text understanding step must be context independent, meaning independent of scenes, colours, lighting and all various conditions.

II. METHODOLOGY

Video Capture:

To capture the real time video and convert the video to frames, OpenCV library in Python is used. OpenCV (Open source computer vision) is a real time computer vision library which contains functions to perform operations on images and videos. It provides a simple interface to capture live stream video using the webcam.

Text Detection:

Text detection is a two-step process which involves basically identifying the region and localizing where the text is present. The localization is done by drawing green bounded text around the text. In the second stage the frame is cropped in order to recognize the text and convert to the machine-readable format.

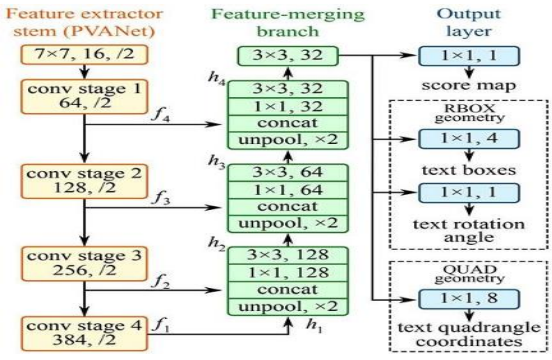


Figure 1:Structure of the EAST text detector fully convolutional network

For text detection, a pre-trained deep learning-based model called **EAST** "Efficient and Accurate Scene Text "detection pipeline is used. It is based on convolutional Neural Network (CNN) and is capable of running at 13 FPS. It is a two-stage pipeline used to detect words at arbitrary orientations. In the first stage, the image is fed to the model and it detects the text region. The other developed text detectors usually have multiple intermediate steps including candidate aggregation and word partitioning but since, this model eliminates those steps, it has improved accuracy and time complexity as compared to them. In the second stage the multiple boxes are suppressed in order to create one box around the detected text. It is known as Non-Maximum suppression stage.

To perform the text detection, we extract the output features maps of two layers. The sigmoid activation provides us the probability if the region contains the text or not whereas the second layer is the output feature map which represents the "geometry" of the image which is used to derive the bounding box coordinates of the text in the video frame.

The captured frame from the video needs to be preprocessed before feeding it to the trained model in order to obtain accurate results. It is done using the blobfromImage function provided by the OpenCV library. After the pre-processing of the frame, it is set as an input to the trained model. Finally, the non-maxima suppression is performed on the bounded boxes and the result is displayed.

III. SOFTWARE TOOLS

Python:

Python is a deciphered, elevated level, broadly useful programming language. Made by Guido van Rossum and first discharged in 1991, Python's structure theory accentuates code readability with its striking utilization of significant whitespace. Its language builds and object-oriented approach to assist software programmers with writing clear, coherent code for small and enormous scale projects.

Python is progressively composed, and trash collected. It bolsters a plethora of programming standards, including useful procedural, object-oriented programming. Python is frequently depicted as a "batteries included" language because of its extensive standard library.

Figure 2: Python Software

Python was considered in the late 1980s as a successor to the ABC language. Python 2.0, unveiled in 2000, presented features like list comprehensions and a trash collection framework equipped for gathering reference cycles. Python 3.0, released in 2008, was a major modification of the language that isn't backward-compatible, and a lot of Python 2 code doesn't run unmodified on Python 3.

Open CV:

OpenCV (Open source computer vision) primarily focused on real-time computer vision is a library of programming functions. Initially created by Intel, it was later supported by Willow Garage. The library is free for use and a cross-platform open-source BSD license. It supports deep learning frameworks such as TensorFlow, Torch/PyTorch and Caffe.

Figure 3: OpenCV Library

NumPy:

NumPy includes support for enormous, multi-dimensional arrays and matrices is a library for the Python programming language with a huge assortment of high-level mathematical functions to work on these arrays. The ancestor of NumPy, Numeric, was initially created by Jim Hugunin with contributions from a few other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the contending Numarray into Numeric, with broad alterations. NumPy is open-source software and has numerous contributors. NumPy is an exceptionally enhanced library for numerical operations. It gives a MATLAB-style linguistic structure.

Figure 4: NumPy Package

Imutils:

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3.

IV. RESULTS

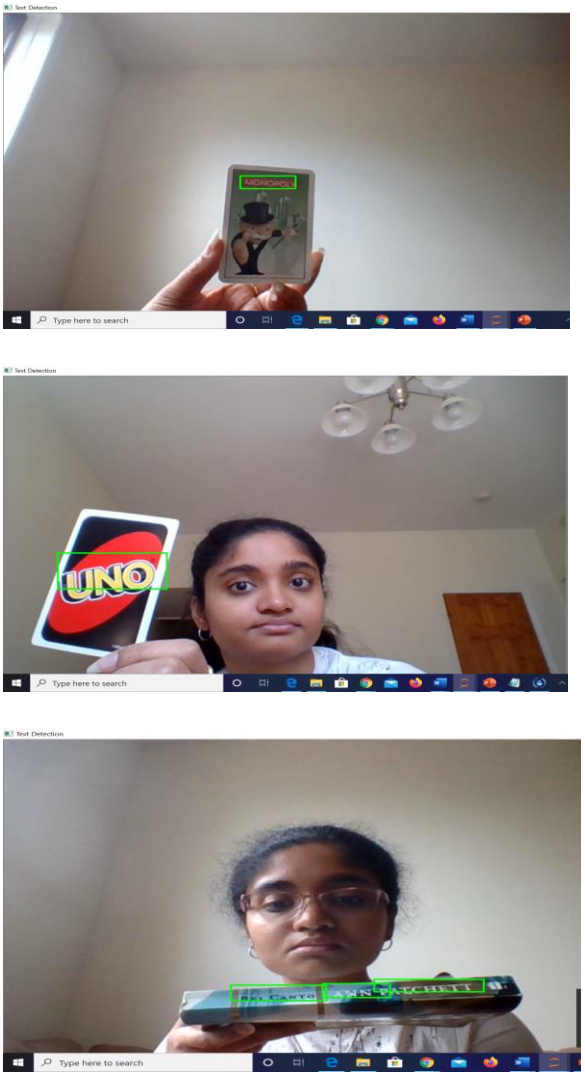


Figure 5: Results obtained

V. CONCLUSION

The project “Real Time Detection of Texts in Videos using Webcam” was executed and the required results were obtained as texts in natural scene environments were detected.

VI. REFERENCES

[1] W. Lu, H. Sun, J. Chu, X. Huang and J. Yu, "A Novel Approach for Video Text Detection and Recognition Based on a Corner Response Feature Map and Transferred Deep Convolutional Neural Network," in IEEE Access, vol. 6, pp. 40198-40211, 2018, doi: 10.1109/ACCESS.2018.2851942.

[2] Goel, Vaibhav & Kumar, Vaibhav & Jaggi, Amandeep & Nagrath, Preeti. (2019). Text Extraction from Natural Scene Images using OpenCV and CNN. International Journal of Information Technology and Computer Science. 11. 48-54. 10.5815/ijitcs.2019.09.06.

[3] Adrian Rosebrock, “OpenCV Text Detection” , 2018

[4] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, Jiajun Liang**, “**EAST: An Efficient and Accurate Scene Text Detector”

[5] ***Siyu Zhu, Richard Zanibbi***; A Text Detection System for Natural Scenes With Convolutional Feature Learning and Cascaded Classification, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 625-632