I.J. Information Technology and Computer Science, **2019, 9, 48-54**
Published Online September 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2019.09.06

Text Extraction from Natural Scene Images using OpenCV and CNN

Vaibhav Goel, Vaibhav Kumar, Amandeep Singh Jaggi, Preeti Nagrath
Computer Science Department, Bharati Vidyapeeth's College of Engineering, New Delhi, India
E-mail: vaibhavgoel17@gmail.com, kvaibhavs077@gmail.com, amandeep998877@gmail.com, preetinagrath1@gmail.com

Received: 06 June 2019; Accepted: 24 June 2019; Published: 08 September 2019

***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. 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 presents a technique that uses a combination of the Open Source Computer Vision Library (OpenCV) and the Convolutional Neural Networks (CNN), to extract English text from images efficiently. The CNN model is based on a two-stage pipeline that uses a single neural network to directly detect the characters in the scene images. It eliminates the unnecessary intermediate steps that are present in the previous approaches to this task making them slower and inaccurate, thereby improving the time complexity and the performance of the algorithm.

***Index Terms*—**Text Extraction, Deep Learning, OpenCV, natural scene images, CNN, Optical Character Recognition.

I. Introduction

The extraction of text from natural scene images is a demanding task owing to the wide variations in the text properties such as font, alignment, orientation, size, style, lightning conditions, color, and a bunch of other parameters. Also, the natural scene images usually suffer from low resolution and low quality, perspective distortion, non-uniformity in illumination, and complex background [1]. Moreover, there is a lack of prior knowledge of the text features and the location of the text regions. All these factors contribute to the difficulty in recognizing text from such images.

The primary reason behind performing this task is the fact that the natural scene images may contain important written information that may not be able to reach certain people due to a number of problems like visual impairment, language barrier, etc., or that may be needed to feed to a computer without explicitly typing the information [2]. There may be situations when we want to write down some information present in the form of text in a hard document, an image, or a video. We can just take a snap of the document from the smartphone's camera and in a couple of seconds, the algorithm will extract all the text present in the document. This will save time as well as effort to write down the whole information manually. Thus, autonomous text extraction from an image could prove to be beneficial. Some major applications of this task include document retrieving, vehicle number plate detection, road signs detection (by smart cars), page segmentation, video content summary, and intelligent driving assistance, to name a few [3]. Self-driving cars' efficiency can be much improved if it knows more than just what is around it, like what is being written on destination boards or traffic signs of curves and diversion. It is also useful for providing assistance to visually impaired persons, thus improving magnificence of life for them. These are some of the greatest benefits of this task that can be beneficial for all.

Now, the present OCR (Optical Character Recognition) techniques are able to attain exemplary accuracy on scanned images, i.e., images having text on a monotonous background. But, they still cannot extract text information accurately from images that are directly taken from a camera (i.e. natural scene images). In other words, the current OCR systems are able to handle text only with a monochrome background, where the text can easily be separated from the background, which is definitely not the case with natural scene images. On the other hand, the current techniques which can actually extract text from natural scene images still lack satisfactory accuracy. Thus, this research-based project tries to fill this gap by proposing a technique that uses a combination of the Open Source Computer Vision Library (OpenCV) and the Convolutional Neural Networks (CNN) for efficient text extraction. It also involves the study of different methods of text extraction from a given image, and a thorough comparison of their performances.

The next section lists some of the past work done in this particular area, followed by our research and proposed methodology in section III, results in section IV, and finally conclusion in section V.

Fig.1. Some examples of images containing text information taken from the Total-Text-Dataset [18]

II. Related Works

In layman's terms, at the highest level, this process of extraction of text from scene images is divided into two sub-processes or steps: the first step is Scene text detection, followed by the second step, which is Scene text recognition [4] (Fig.2).

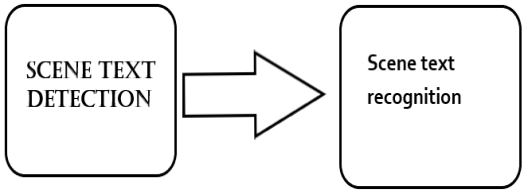

Fig.2. The two most basic steps of text extraction process.

In the text detection process, the locations where the text is present in the image are identified. This process is also known as text localization [5]. It is the most difficult task because of the aforementioned challenges. The pre-processing of the image is done so as to make it suitable for feeding it to the feature detector model. The pre-processing involves tasks such as converting the image to grayscale, defining a specific width and height for the image that is optimum for the feature detector in which it has to be sent into, detecting the edges in the image using edge-detectors such as the Canny edge detector, etc. After the regions in which the text is present have been identified, the next step is text recognition, i.e., recognizing the text information present in those regions. These regions of image are sent into deep learning models that are vigorously trained to recognize characters, and consequently words. Different models are present to implement this task, which will be discussed later. It requires a lot of data to train these models to recognize different styles, strokes, and spacing between the letters or numbers.

The output of these models will have the words or characters that were present in the image. This can help us to know and record data in low memory (text instead of image) and share the main point in the random image that is selected.

Now, the present OCR (Optical Character Recognition) [6] techniques are able to attain exemplary accuracy on scanned images, i.e. images having text on a monotonous background. But, they still cannot extract text information accurately from images that are directly taken from a camera, i.e. natural scene images. In other words, the current OCR systems are able to handle text only with a monochrome background, where the text can easily be separated from the background, which is definitely not the case with natural scene images. Many researchers have done intensive research work in finding the methods for improving the accuracy of existing text recognition algorithms, but because of the immense variations seen in the natural scene images and the lack of prior knowledge of any kind of text features, it is still a challenging task to achieve almost perfect recognition rate. The next paragraph discusses some of the past work proposed or implemented in this area.

There can be two types of approaches to this task of text detection: conventional approach and deep neural network based approach. The existing methods based on either approach are time consuming and not optimal due to the use of multiple stages and components in the process. The conventional approaches include techniques like Maximally Stable Extremal Regions (MSER) [15, 22] and Stroke Width Transform (SWT). These techniques generally use extremal region detection or edge detection to seek candidate characters in an image. The authors in [7] proposed an approach that employed a clustering method based on density values for the segmentation of candidate characters by combining color features of

character pixels and spatial connectivity. JiSoo Kim [8] proposed three text extraction techniques for natural scene images that were based on the intensity information of the input image. The 1st technique is "Gray Value Stretching" that involves binarization of the image by calculating the average intensity of all the pixels. The 2nd technique introduces the "Split and Merge" approach, which is a popular algorithm for image segmentation. The 3rd technique is derived from the amalgamation of the first two techniques. Busta et al. [9] made FASText (a fast scene text detector) by modifying the famous FAST key point detector which is used for stroke extraction. But, this technique was not as efficient as the ones based on deep neural networks. It lagged behind both in terms of accuracy and flexibility, especially in scenarios with low resolution. Coates et al. [10] introduced an unsupervised learning model that was integrated with a variation of the famous clustering algorithm, the K-means clustering. This model extracts the local features of character patches, followed by pooling them on the basis of cascading sub-patch features. Lu et al. [11] proposed a method that describes a dictionary of basic shape codes, modeling the inner character structure, to perform word and consequently character retrieval on scanned documents, without OCR. Huang et al. [12] proposed a technique with a two pipeline structure. The first step was to deduce the candidate features using the concept of Maximally Stable Extremal Regions (MSER). The second step included the use of a convolutional neural network as a strong classifier. It was employed to suppress false positives, i.e., the regions that does not contain any text but are still positively detected by the algorithm. Mishra et al. [13] acquired an approach that involves the concept of conditional random field, to combine bottom-up character recognition and top-down word-level recognition. Based on the Scale-Invariant Feature Transform (SIFT), Smith et al. [14] built a model of text detection that amplified the posterior probability of similarity constraints by making use of integer programming. SIFT is a corner detection technique invented by D.Lowe in 2004 to overcome the limitations of Harris Corner Detector. Neumann et al. [15] used the idea of extremal regions to propose a real-time text detection and extraction technique. Liu et al. [16] merged three models for the text recognition in scene images. The three models are: 1). the Gabor-based appearance model for feature detection, 2). the similarity model, and, 3). the Lexicon model.

These were some of the major contributions to this field of text extraction that have set the bar high for new researchers and researches in the field. But, the best accuracy achieved till date is still not that great, and has a scope for improvement.

III. Methodology

The authors have used a pre-trained deep learning model, the EAST text detector, for reference. EAST is short for "Efficient and Accuracy Scene Text" detection pipeline [17]. It is a fully-convolutional neural network proposed by Zhou et al. in 2017 in his paper "EAST: An Efficient and Accurate Scene Text Detector". It is a model for determining the locations of text in natural scene images. The output of this model provides per-pixel predictions for characters or words. The model has the ability of running in real-time at 13 frames per second on images with 720p resolution. Also, it has achieved state-of-the-art text localization accuracy. The structure of the EAST text detector is shown in Fig.3 as per Zhou et al. [17].

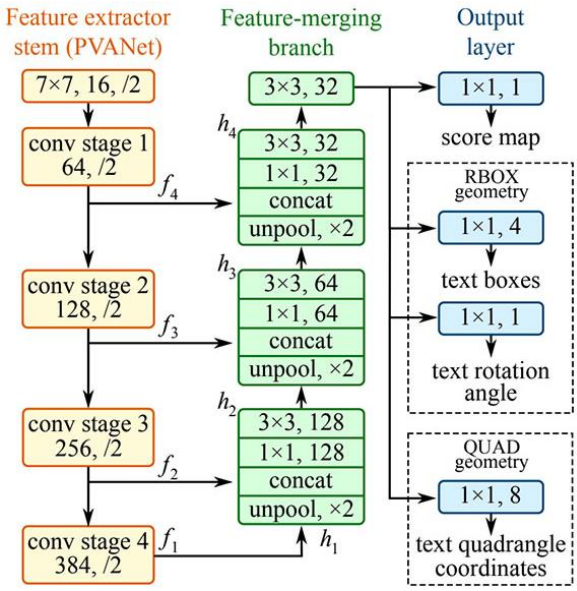

Fig.3. Structure of the EAST text detector Fully Convolutional Network [17].

The CNN model is based on a two-stage pipeline that uses a single fully convolutional neural network to directly detect the characters in the scene images. The first stage includes feeding the image to the neural network. It directly detects the text regions in the image, eliminating the unnecessary intermediate steps that are present in the previous approaches [9, 23, 24, 25, 26] to this task making them slower and inaccurate, thereby improving the time complexity and the performance of the algorithm. The second stage is the Non-maximum suppression (NMS) stage which suppresses the multiple bounding boxes created around the same text region to just one box. This is the high level overview of the pipeline used.

Fig.4. (a). Original Image (b). Converted to grayscale (c). After applying Canny Edge Detection

Now, when it comes to computer vision tasks, there is nothing that can beat OpenCV as of now. It is a specialized open source library specifically designed for developing Computer Vision applications.

The methodology presented in the paper combines both the state-of-the-art technologies for image processing, the Convolutional Neural Networks (CNN) and the OpenCV library. The aforementioned trained deep learning model is exploited using OpenCV 3.4.2.

The already trained tensorflow model named "frozen_east_text_detection" is used as a reference for the new model. There are two output layers in the model. The first layer is a sigmoid convolution layer named "feature_fusion/Conv_7/Sigmoid". This layer outputs the regional probability, specifying whether that region is containing text information. The second output layer is named "feature_fusion/concat_3" which determines and outputs the coordinates of the bounding boxes containing the text information.

The trained model is loaded into memory by feeding it to the OpenCV using the cv2.dnn.readnet() method. Now, before feeding the image to the model, it is essential to preprocess the image in order to obtain accurate results. OpenCV provides different preprocessing features via its dnn module's blobFromImage function, which performs Mean subtraction, scaling by some factor and channel swapping.

Now, the above blob produced out of the image is set as the input to the model that was previously loaded. On performing a forward pass on the CNN model, it generates as output two feature maps from the two output layers that have been discussed above. The first tells the probability of a particular region containing text, while the second determines the coordinates of bounding boxes containing text in the input image. The model might have detected regions that are having very less probability of containing text. Such regions that does not have sufficiently high probabilities should be ignored. To perform this task, the algorithm loops over all the probability scores that were obtained as output from the model. The regions that have probability less than 0.5 are ignored, while the regions having probability more than 0.5 are considered and the corresponding bounding box coordinates are plotted so as to obtain a box (rectangle) around the text.

Now comes the process of text recognition. The regions having text in the image have been identified successfully at this stage. Those regions are now required to be fed into an OCR system. OCR stands for "Optical Character Recognition". It is a system that converts the images containing text into machine-encoded text, thus recognizing the text present in the images. For the OCR to produce accurate results, instead of sending the natural scene images as it is, only those portions of the image are sent that contains the text information. This is because OCR can't handle images with complex background, and struggles to differentiate text regions from non-text regions. Thus, preprocessing is done on the regions detected by the above deep learning model, in order to alienate the text elements from the background, thereby maximizing the efficiency and accuracy of OCR. The preprocessing includes techniques like BGR to grayscale conversion, and canny edge detection.

In OCR, a CNN model is employed that analyze image for contrast of light and dark areas to recognize characters or numeric digits. The lines are segmented into words and then into characters. On getting the characters, the algorithm compares those characters with the data of predefined pattern images set. As a character is recognized, it is then converted into an ASCII code.

This is the complete algorithm that the authors have adopted for the efficient text extraction. The following steps summarize the complete process in a high-level manner.

Step#1: Input Image.
Step#2: Preprocessing of the input image.
Step#3: Text detection and localization using a fully convolutional neural network.
Step#4: Preprocessing of the detected text regions.
Step#5: Text recognition using another CNN model (OCR).

Now, after going through the complete process, the authors realized that this segmentation of work in two segment pipeline of detection and recognition can be achieved under a single network where: (1). end to end recognition is converted into a spatial transformer network in a semi supervised way which is a part of a major deep neural network (DNN), and, (2). the later part of the network is then trained for recognition of the characters in the text. A CNN model is built that takes an

input feature map A and perform spatial recognition on the image and produce two output map that gives the probability and location of the text in the image, respectively. The output maps are than fed to other algorithms for image sampling. The sampled image is then passed into another CNN model for recognition of the text from the predefined data set. This is known as OCR. The two models can be named as localization networks and recognition networks for the sake of ease of understanding how the networks are going to behave and in what fashion.

IV. Performance Evaluation

There are two quantities, namely Precision Rate and Recall Rate, which are generally used to measure the performance of the algorithm in the task of text extraction. The precision rate is the ratio of the number of correctly detected words to the sum of the number of correctly detected words and false positives. False positives are the regions of the image that do not contain any text but still have been recognized by the algorithm.

$$\%\ \text{Precision Rate} = \frac{\text{correctly detected words}}{\text{correctly detected words} + \text{false positives}} \times 100$$

The recall rate is the ratio of the number of correctly detected words to the sum of the number of correctly detected words and false negatives. False negatives are the regions of the image that do actually contain text but haven't been recognized by the algorithm [3].

$$\%\ \text{Recall Rate} = \frac{\text{correctly detected words}}{\text{correctly detected words} + \text{false negatives}} \times 100$$

This approach can also be used for individual characters rather than words, if the image doesn't contain complete words. These quantities are calculated to determine the efficiency of the algorithm.

Table 1. Performance Comparison of different text extraction techniques

S.No.	Author	% Precision Rate	% Recall Rate
1.	Xiaoqing Liu et al. [16]	96.6	91.8
2.	Xu-cheng yin et al. [19]	68.5	82.6
3.	JiSoo Kim et al. [8]	72.6	69.4
4.	Fang Liu et al. [7]	75	81
5.	Xiaoqian Liu et al. [20]	65	63
6.	Yi-Feng Pan et al. [21]	68	69
7.	**Ours**	**88.3**	**76.8**

The proposed OpenCV implementation achieved a precision rate of 88.3% and a recall rate of 76.8%. This implies that the model is capable of not producing considerable false positives, but may sometimes encounter false negatives, i.e., may not detect regions actually containing text. Table 1 gives the performance comparison of different text extraction techniques. Some of the images that the algorithm produced are shown in Fig.5. The bounding boxes around the text are accurately generated, and the algorithm also performed well on noisy images.

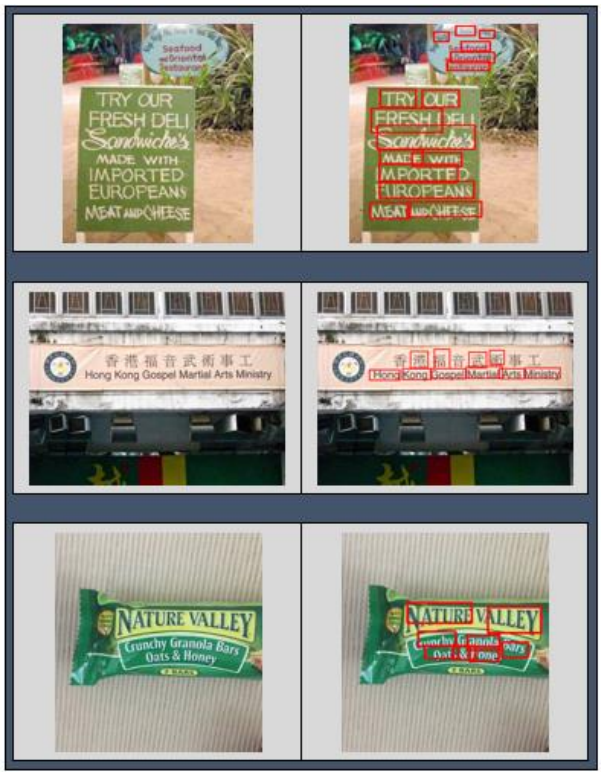

Fig.5. The original images fed to the model on the left side and the output images with bounding text boxes on right side.

The accuracy obtained is quite satisfactory, considering the wide variety of images present in the used dataset. However, it has some basic limitations. The first limitation is that the algorithm cannot detect text that is not horizontally aligned as it cannot produce rotated bounding boxes. Another limitation is that it cannot detect text that is embedded in a circular manner. But these limitations can be easily resolved in future.

V. Conclusion

The paper presented an efficient technique of text extraction from natural scene images, i.e., the images directly taken from a camera, having a random background and a diverse variety of colors. There are a number of significant applications of this task, including document retrieving, vehicle number plate detection, road signs detection (by smart cars), page segmentation, assistance to visually impaired persons, video content summary, and intelligent driving assistance, to name a few. There exists a number of techniques for text extraction, each having its own set of strengths and limitations. There is no single algorithm that works for all the applications due to the wide variations in the type of natural images.

The proposed technique is efficient in producing text predictions on images having horizontally-aligned text. A single convolutional neural network based on the EAST

text detector [17] is employed for the detection task. For the recognition part, another CNN model is trained which produces excellent results when proper preprocessing of the image is done before feeding it to the network.

The future research may possibly include overcoming the limitations of this implementation, i.e. its inability to detect text in images that are not horizontally aligned or images where the text is not in a straight line. Also, this approach can be extended from images to videos, which are nothing but a series of images. Moreover, a smartphone application can be built that can detect and extract text from camera-captured images (or videos) in real-time.

REFERENCES

[1] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform", *Proc. CVPR, 2010.*

[2] H. Raj, R. Ghosh, "Devanagari text extraction from natural scene images", *IEEE 2014.*

[3] T. Kumuda and L. Basavaraj, "Text extraction from natural scene images using region based methods-a survey", *Proc. of Int. Conf.* on *Recent Trends in Signal Processing, Image Processing and VLSI, ACEEE 2014.*

[4] M. Prabaharan and K. Radha, "Text extraction from natural scene images and conversion to audio in smart phone applications", *IJIRCCE, 2015.*

[5] C. Bartz, H. Yang, and C. Meinel, "STN-OCR: A single neural network for text detection and text recognition", *arXiv:1707.08831v1 [cs.CV] 27 Jul 2017.*

[6] S. Mori, H. Nishida, and H. Yamada, "Book optical character recognition", *John Wiley & Sons, Inc. New York, NY, USA, 1999.*

[7] F. Liu, X. Peng, T. Wang, and S. Lu, "A density-based approach for text extraction in images", *IEEE 2008*.

[8] J. Kim, S. Park, and S. Kim, "Text locating from natural scene images using image intensities", *IEEE 2005.*

[9] M. Busta, L. Neumann, and J. Matas, "Fastext: Efficient unconstrained scene text detector", in *Proc. of ICCV*, 2015.

[10] Coates et al., "Text detection and character recognition in scene images with unsupervised feature learning", *Proc. ICDAR 2011*, pp. 440–445.

[11] S. Lu, T. Chen, S. Tian, J. H. Lim, and C. L. Tan, "Scene text extraction based on edges and support vector regression", IJDAR, 2015.

[12] W. Huang, Y. Qiao, and X. Tang, "Robust scene text detection with convolution neural network induced mser trees", in Proc. of ECCV, 2014.

[13] N. Mishra, C. Patvardhan, Lakshimi, C. Vasantha, and S. Singh, "Shirorekha chopping integrated tesseract ocr engine for enhanced hindi language recognition", International Journal of Computer Applications, Vol. 39, No. 6, February 2012.

[14] R. Smith, "An overview of the tesseract OCR engine", in Proc. Int. Conf. Document Anal. Recognition, pp. 629–633 2007.

[15] L. Neumann and J. Matas, "Real-time scene text localization and recognition", 25th IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[16] X. Liu and J. Samarabandu, "Multiscale edge-based text extraction from complex images", 2006 IEEE International Conference on Multimedia and Expo.

[17] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: an efficient and accurate scene text detector", *arXiv:1704.03155v2 [cs.CV] 10 Jul 2017.*

[18] Chee Kheng Ch'ng & Chee Seng Chan, "Total-Text: a comprehensive dataset for scene text detection and recognition", *14th IAPR International Conference on Document Analysis and Recognition {ICDAR}, 2017,* pp. 935-942.

[19] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images", IEEE transactions on pattern analysis and machine intelligence, 0162-8828/13, 2013, IEEE.

[20] X. Liu, K. Lu, and W. Wang, "Effectively localize text in natural scene images", 21st international conference on pattern recognition(ICPR), November 11-15, 2012, Tsukuba, Japan.

[21] Y. F. Pan, X. Hou, C. L. Liu, "A robust system to detect and localize texts in natural scene images", unpublished.

[22] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images", in Proc. of ACCV, 2010.

[23] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan, "Text flow: A unified text detection system in natural scene images", in Proc. of ICCV, 2015.

[24] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks", International Journal of Computer Vision, 116(1):1– 20, jan 2016.

[25] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images", arXiv preprint arXiv:1604.06646, 2016.

[26] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks", in Proc. of CVPR, 2015.

Authors' Profiles

Vaibhav Goel: Under-graduate student pursuing Bachelor of Technology (B.Tech) at Bharati Vidyapeeth's College of Engineering, New Delhi, India, major in Computer Science.

Vaibhav Kumar: Under-graduate student pursuing Bachelor of Technology (B.Tech) at Bharati Vidyapeeth's College of Engineering, New Delhi, India, major in Computer Science.

Amandeep Singh Jaggi: Under-graduate student pursuing Bachelor of Technology (B.Tech) at Bharati Vidyapeeth's College of Engineering, New Delhi, India, major in Computer Science.

Preeti Nagrath: Professor at Bharati Vidyapeeth's College of Engineering, New Delhi, India, major in Computer Science.

How to cite this paper: Vaibhav Goel, Vaibhav Kumar, Amandeep Singh Jaggi, Preeti Nagrath, "Text Extraction from Natural Scene Images using OpenCV and CNN", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.9, pp.48-54, 2019. DOI: 10.5815/ijitcs.2019.09.06

