*Intelligent Automation & Soft Computing*
DOI: 10.32604/iasc.2023.036411
***Article***

Tech Science Press



# Embedded System Based Raspberry Pi 4 for Text Detection and Recognition

**Turki M. Alanazi***

Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka, Saudi Arabia
*Corresponding Author: Turki M. Alanazi. Email: tmanazi@ju.edu.sa
Received: 29 September 2022; Accepted: 14 November 2022

**Abstract:** Detecting and recognizing text from natural scene images presents a challenge because the image quality depends on the conditions in which the image is captured, such as viewing angles, blurring, sensor noise, etc. However, in this paper, a prototype for text detection and recognition from natural scene images is proposed. This prototype is based on the Raspberry Pi 4 and the Universal Serial Bus (USB) camera and embedded our text detection and recognition model, which was developed using the Python language. Our model is based on the deep learning text detector model through the Efficient and Accurate Scene Text Detector (EAST) model for text localization and detection and the Tesseract-OCR, which is used as an Optical Character Recognition (OCR) engine for text recognition. Our prototype is controlled by the Virtual Network Computing (VNC) tool through a computer via a wireless connection. The experiment results show that the recognition rate for the captured image through the camera by our prototype can reach 99.75% with low computational complexity. Furthermore, our prototype is more performant than the Tesseract software in terms of the recognition rate. Besides, it provides the same performance in terms of the recognition rate with a huge decrease in the execution time by an average of 89% compared to the EasyOCR software on the Raspberry Pi 4 board.

**Keywords:** Text detection; text recognition; OCR engine; natural scene images; Raspberry Pi; USB camera

## 1 Introduction

Traditional text detection and recognition methods for document images have shown good results in scanned images. But, it had limited effectiveness in natural scene images due to a lack of black and white backgrounds [1,2]. However, the text may be found in natural scenes such as shop names, road signs, etc. This text can provide important information to understand the content of the image and can be used in various applications such as navigation, translation, search engines, etc.

The variety of scenes and texts contributes to the difficulty of detecting and recognizing text in natural scene images. As a matter of fact, in many contexts, other disturbances like glass, windows, and trees might easily be confused with the text area. In recent years, researchers have focused on text localization, detection,

extraction, and recognition from natural scene images. Consequently, text detection from natural scene images has emerged as a crucial challenge in computer vision.

To solve this problem, a wide range of approaches to detect and recognize text from natural scene images have been proposed, and numerous research reviews have contributed to the field [3–7]. Ye et al. provided an overview of the approaches, techniques, and difficulties in text detection and recognition in images [3]. Zhu et al. gave an in-depth examination of scene text detection and recognition, including current improvements and future developments [4]. Authors in [5] create a novel text location that splits scene images into a number of segments and then extracts the features of the segments, such as stroke direction, distributions of edge pixels, etc. Finally, these fragments are divided into text and non-text categories using the AdaBoost (Adaptive Boosting) classifier, and the text fragments are combined to create a full document. In [6], authors provide a novel image transform approach that assigns a stroke width value to each pixel and then utilizes the stroke width to filter non-text region. In [7], authors employ a text area detector to determine the scale and location of the text, and then split the text into a series of candidate text using the binary technique. After that, a conditional random field model is trained, and then the non-text region is filtered using this model. The authors of [8] provide a strong text detection and recognition system based on support vector machine and maximally stable extremal regions.

In the recent years, deep learning techniques have been utilized to address the problems such as OCR analysis and text detection, extraction and recognition from natural scene images [9–14]. However, better performance can be achieved by utilizing deep learning technique [15]. In fact, the principle concept of deep learning is to separate different levels of abstractions entrenched in experiential data by highly designing the depth of layer and width of layer, and correctly selecting image features which are useful for learning tasks.

In this context, we propose in this work an efficient Text Detection and Recognition Model (TDRM) allowing us to locate, detect, and recognize text from natural scene images. For text localization and detection, the Efficient and Accurate Scene Text Detector (EAST) model is used, which is a deep learning text detector model. The EAST model is able to predict words and lines of text at random orientations. But, for text recognition, the tesseract-OCR is used as the Optical Character Recognizer (OCR) engine. Our TDRM is developed using the Python programming language and is embedded on the Raspberry Pi 4, which is the heart of the designed prototype. This prototype uses the USB camera to capture images which are processed by our proposed TDRM. The whole system is controlled by the Raspberry Pi Operating System (OS).

Thus, the following points summarize the contribution of this work:

- An efficient TDRM based on the EAST model and the Tesseract-OCR engine is proposed to extract, detect, and recognize text from natural scene images.
- A new Raspberry Pi 4-based TDRM is designed to recognize text from the image captured by a USB camera.
- The TDRM is developed using python programming language, compiled for Cortex-A72 ARM processor and executed on the Raspberry Pi 4 board through the Raspberry Pi operating system.
- Our TDRM embedded system is remote controlled through the Virtual Network Computing (VNC) tool.
- The robustness of the TDRM embedded system relative to EasyOCR and Tesseract-OCR is confirmed via the presented analysis.

Our paper is organized as follows: Section 2 describes the proposed text detection and recognition model used for the natural scene images. The embedded implementation of the TDRM based Raspberry Pi 4 and USB camera is presented in Section 3. Section 4 illustrates the experimental results. Section 5 concludes the paper.

## 2 Text Detection and Recognition Model

The whole text detection and recognition model is presented in Fig. 1. However, in the text detection step, the EAST model is used to locate and detect the text from the input image. Then, the Tesseract-OCR engine is applied to each detected text in the image to recognize the text and generate the editable text file. Next, we describe the EAST model and the Tesseract-OCR engine used in our proposed text detection and recognition model.

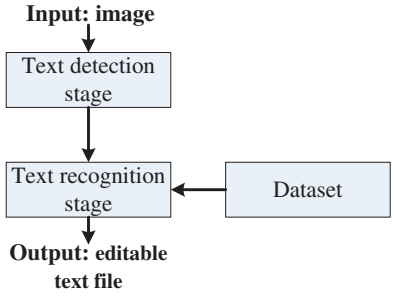


**Figure 1:** Text detection and recognition model

### *2.1 Text Detector Model*

Text detection is an important step in the text extraction and recognition process. In this context, EAST [9] presents the most important and powerful model for text detection [2] and provides an accurate result for text recognition. In fact, according to the authors, the EAST deep learning text detector model is able to identify words and lines of text on 720p at 13 Frames Per Second (FPS) at arbitrary orientations. However, most critically, it is feasible to avoid computationally costly sub-algorithms like word partitioning and candidate aggregation that are frequently used by other text detectors. Thereby, the EAST model employs carefully developed new functions to create and train such a deep learning model. In [9], however, the EAST integrates a Fully Convolutional Network (FCN) model and uses just two pipeline stages to predict text words or lines. Fig. 2 shows a schematic representation of the EAST structure that is divided into three stages: the feature extractor stem, the feature-merging branch, and the output layer.

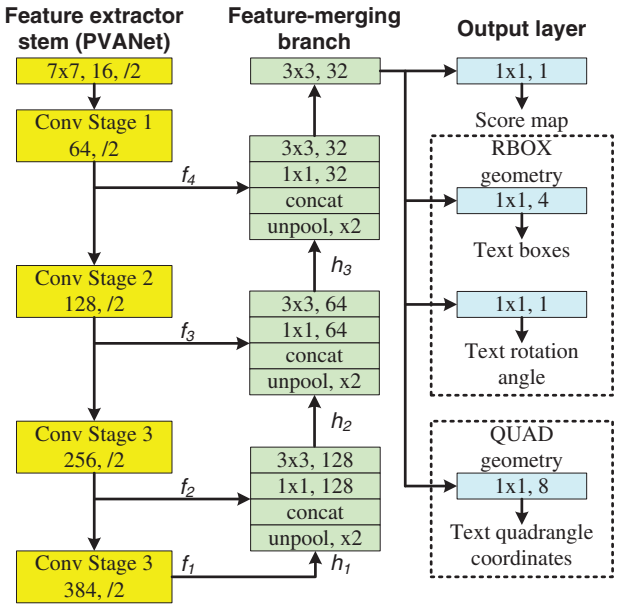


**Figure 2:** EAST structure for text detection [9]

As depicted by Fig. 2, the input image is processed by a convolutional layer to produce the four feature maps f1, f2, f3, and f4. In the second step, the feature maps are unpooled and concatenated along the channel dimension. Afterwards, they go through two convolution layers, 11 and 33, which are given by Eqs. (1) and (2), respectively.

$$g_i = \begin{cases} unpool(h_i) \ if \ i \leq 3 \\ conv_{3\times3}(h_i) \ if \ i = 4 \end{cases} \tag{1}$$

$$h_i = \begin{cases} f_i & if \ i = 1 \\ conv_{3\times3}(conv_{1\times1}([g_{i-1}; f_i])) & otherwise \end{cases} \tag{2}$$

where $g_i$ is the merge base and $h_i$ is the merged feature map and the operator [.;.] represents concatenation along the channel axis.

Finally, in the output layer, score and box prediction occur. RBOX geometry is used to generate rotated boxes. QUAD geometry is used to generate the text quadrangle coordinates.

### *2.2 Text Recognition Model*

As we know, the OCR has been widely used to identify characters from various documents or file formats and further helps to identify the characters, fonts, page layouts, etc. In recent years, different OCRs have become available, such as Google OCR, Tesseract, Python OCR, Amazon OCR, and Microsoft OCR, etc. However, the Tesseract OCR [16,17] is an open-source engine. It has almost all the language-trained data files for character recognition and outperforms most commercial OCR engines [18].

Fig. 3 depicts the process used by the Tesseract-OCR to recognize the text from the input image. In fact, once the input image is analyzed, the detected blobs are organized into text lines. Then, each text line is chopped into words. Next, two pass are used to recognize the words. So, on the first pass, the satisfying word is sent to the adaptive trainer to attempt to recognize the word. Afterward, the second pass is applied to each word that was not successfully recognized by the first pass. At the end, the recognized text is output.

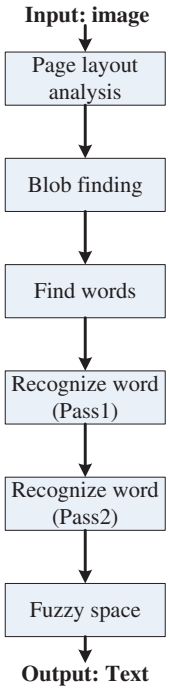


**Figure 3:** Tesseract-OCR block diagram

## 3 Embedded Implementation of the TDRM

The Raspberry Pi 4 [19] is used to implement and evaluate our proposed text detection and recognition model. Fig. 4 presents the Raspberry Pi 4 board. However, the heart of this board is the Broadcom BCM2711 chip, which was designed based on the Quad-core ARM Cortex-A72 Harvard Reduced Instruction Set Computer (RISC) processor. The ARM Cortex-A72 is a 64-bit processor operating at 1.5 GHz. The size of the L1 cache memory is 32 and 48 KB for data and instructions, respectively. But, the size of the L2 cache memory is 1 MB. The Raspberry Pi 4 comes with 4 GB of RAM, which depends on the model, and provides several input/output ports, such as 4 Universal Serial Bus (USB) ports, a High-Definition Multimedia Interface (HDMI) port, a Secure Digital (SD) card, etc., allowing us to develop and implement an embedded system for several applications [20]. The Raspberry Pi 4 is controlled by an OS like Raspberry Pi OS [21], which offers a flexible and efficient environment to develop and evaluate any application through a high-level programming language (e.g., Python, Java, C/C++, etc). Therefore, we can consider the Raspberry Pi 4 as a small computer built on a tiny circuit board.

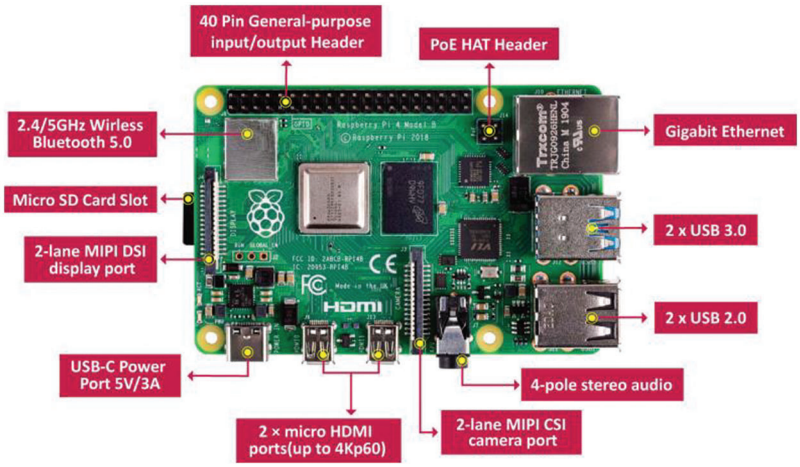


**Figure 4:** Raspberry Pi 4 board

Fig. 5 describes the operating principle of our text recognition system based on the Raspberry Pi 4. In fact, our system starts with the initialization of all General-Purpose Input/Output (GPIO) pins to define the input and output pins. Besides, our system initializes the USB camera to start the video streaming. However, in our work, the Logitech C922 USB camera (Fig. 6) [22] is used for the accuracy of the video acquisition. Indeed, this camera is connected to the Raspberry Pi 4 via a USB port and provides a Full High Definition (HD) video resolution of 1080p at 30 FPS with HD auto light correction, which allows a higher quality frame capture. To give us flexibility for controlling our system, 3 LEDs (red, orange, and green) and a button are used. But, the Raspberry Pi 4 GPIO interface does not contain these peripherals. Thus, to solve this problem, the Traffic HAT board [23] can be mounted on the Raspberry Pi 4 board to extend the GPIO interface as shown in Fig. 7, which includes 3 LEDs, a button, and a buzzer. For this board, no extra libraries are needed apart from those to control the GPIO pins. Nevertheless, when the green LED is switched on, it means that our system is ready to launch the process of detecting and recognizing the text from color images captured from the video streaming. But, this process is launched only if the button on the Traffic HAT board is pressed. Otherwise, our system continues to read the video streaming from the camera and remains on hold until the button is pressed, as illustrated in Fig. 5. So, when the button is pressed, our system captures the image from the camera, switches off the green LED, switches on the red LED, and launches the text detection and recognition for the captured image to generate the editable text file. Then, our system switches off the red LED, switches on the orange LED, and

remains on hold until the button is pressed to restart the video streaming and the process for detecting and recognizing the text from the captured image. Nonetheless, the red switch means that our system is busy. But, the orange LED means that the editable text file is generated.

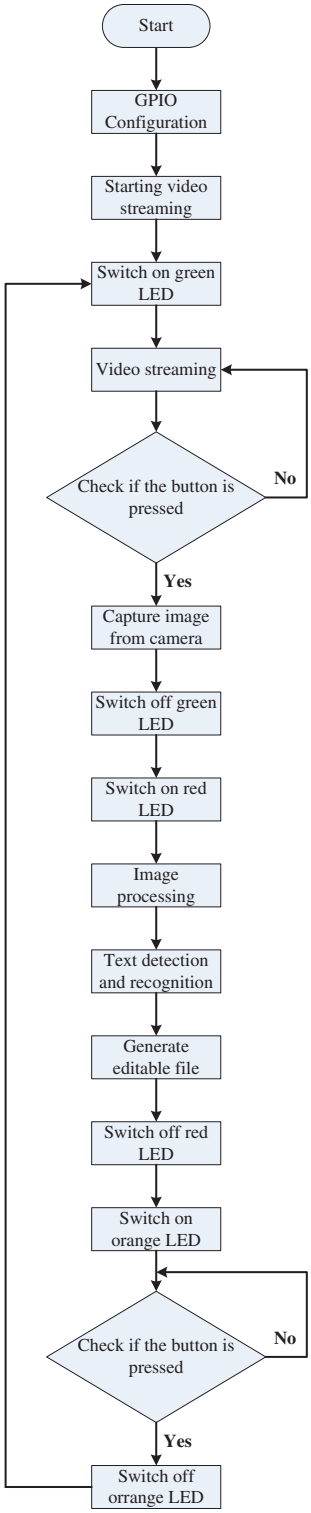


**Figure 5:** Operating of our TDRM based Raspberry Pi 4

**Figure 6:** USB camera (Logitech C922)


**Figure 7:** Traffic HAT board for Raspberry Pi

Fig. 8 shows the prototype of the TDRM embedded system. This prototype contains the Raspberry Pi 4 board, the Traffic HAT board, and the USB camera. It is controlled by the last version of the Raspberry Pi OS. Furthermore, the Virtual Network Computing (VNC) tool is used to communicate with our prototype through the computer via a wireless connection. However, VNC is a remote-control software that allows us to control another computer over a network connection. Besides, Python is used as a software programming language for developing our text recognition system. For that, the OpenCV and the tesseract software are installed in the Raspberry Pi environment.


**Figure 8:** Prototype of the TDRM embedded system

Fig. 9 illustrates the Raspberry Pi development environment. From this figure, we can see the Python integrated development environment and an example of the text recognition for the image captured by a USB camera. These images are processed on the Raspberry Pi 4 board by our TDRM, which is based on the EAST model for text detection and the Tesseract enginge for text recognition. Consequently, the performance evaluation of our proposed is presented in the next section.

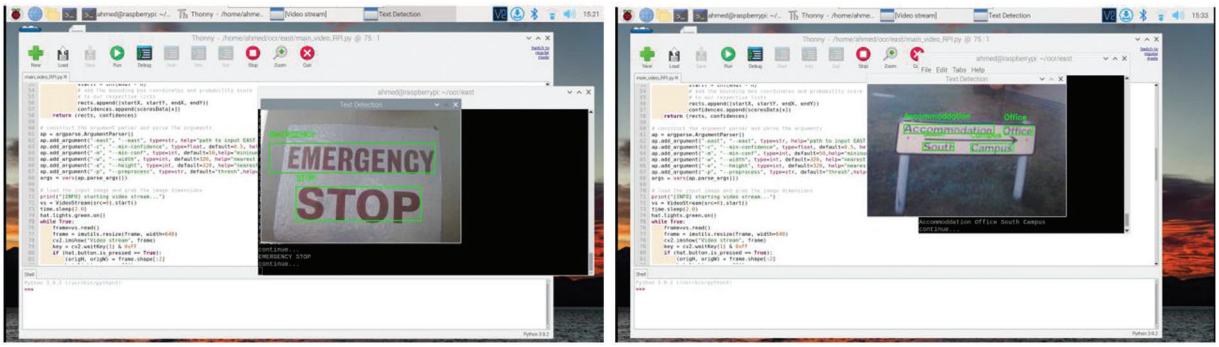


**Figure 9:** Raspberry Pi 4 development environment

## 4 Experimental Results

To evaluate the effectiveness of our proposed for text scene recognition, the International Conference on Document Analysis and Recognition (ICDAR) database [24] is used. The natural scene images used in our experiment from the ICDAR database are depicted in Fig. 10. For a quantitative evaluation, the recognition rate is used to evaluate how many percent of the text regions are correctly detected by our proposed compared to other methods as recorded in Table 1. The recognition rate is calculated by Eq. (3).

$$Recognition\ rate = \frac{S + I + D}{N} \tag{3}$$

where S presents the number of substitutions, N indicates the number of characters in the reference text, D specifies the number of deletions, and I designates the number of insertions.


Image1 Image2 Image3

Image4 Image5 Image6

**Figure 10:** Scene images from ICDAR database

**Table 1:** Comparison of the recognition rate for the text recognition

| | EasyOCR | TesseractOCR | Our proposed |
|---|---|---|---|
| Image 1 | 93.2% | 79.25% | 98.82% |
| Image 2 | 99.88% | 77.5% | 99.38% |
| Image 3 | 98.45% | 63% | 99.26% |
| Image 4 | 99.99% | 0% | 98.89% |
| Image 5 | 97.89% | 60.42% | 98.69% |
| Image 6 | 99.55% | 48% | 99.75% |

In addition, the performance evaluation of our proposed compared to other methods is realized in terms of the execution time. The results of the execution time on the Raspberry Pi 4 board and the Intel i7-1165G7@2.80 GHz processor are recorded in Figs. 11 and 12, respectively.

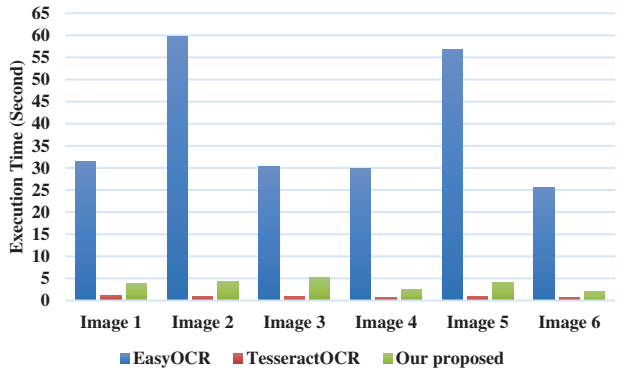


**Figure 11:** Comparison of the execution time (Second) on the raspberry Pi 4 for the text recognition

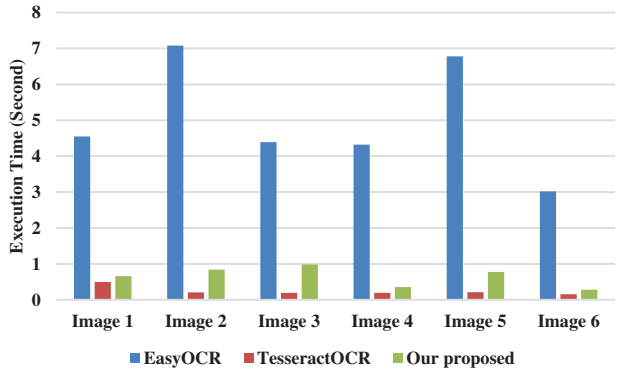


**Figure 12:** Comparison of the execution time (Second) on the intel i7@2.80 GHz processor for the text recognition

However, we can notice from Table 1 that our proposed is more performant than the Tesseract software in terms of the recognition rate, with an increase of an average of 75% and 61% in terms of the execution time on the Raspberry Pi 4 board and the Intel i7-1165G7@2.80 GHz processor, respectively. On the other hand, our proposed has the same performance in terms of the recognition rate relative to the EasyOCR software

[25] as noted in Table 1. But, our proposed allows a huge decrease in the execution time by an average of 89% and 86% compared to the EasyOCR software the Raspberry Pi 4 board and the Intel i7-1165G7@2.80 GHz processor, respectively.

In conclusion, the experimental results show that our proposed can obtain a higher accuracy both in terms of text detection and recognition than the other methods (EasyOCR and TesseractOCR), as depicted in Fig. 13, which presents the text detection and recognition for several scene images by the EasyOCR software, Tesseract software, and our proposed.


(a) (b) (c)

**Figure 13:** Text recognition from scene image by (a) EasyOCR, (b) TesseractOCR and (c) our proposed

## 5 Conclusion

In this work, a prototype based on the Raspberry Pi 4 board and the USB camera for text detection and recognition is proposed. In fact, for text localization and detection, the EAST model is used. The EAST is a deep learning text detector model which is capable of predicting words and lines of text at arbitrary orientations. But the text recognition is processed by the Tesseract-OCR engine. Thus, our text detection and recognition model was developed based on the combination of the EAST model and the Tesseract-OCR engine and is embedded on the Raspberry Pi 4. The performance evaluation of our system on the Raspberry Pi 4 board showed that our proposed provides high accuracy for text detection and recognition from natural scene images with low computational complexity.

**Funding Statement:** This work was funded by the Deanship of Scientific Research at Jouf University (Kingdom of Saudi Arabia) under Grant No. DSR-2021-02-0392.

**Conflicts of Interest:** The author declares that he has no conflicts of interest to report regarding the present study.

## References

[1] P. P. Rege and S. Akhter, "Text separation from document images: A deep learning approach," in *The Machine Learning and Deep Learning in Real-Time Applications*, 1st ed., USA: IGI Global, pp. 283–313, 2020.

[2] Mayank, S. Bhowmick, D. Kotecha and P. P. Rege, "Natural scene text detection using deep neural networks," in *Proc. I2CT*, Maharashtra, India, pp. 1–6, 2021.

[3] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480–1500, 2015.

[4] Y. Zhu, C. Yao and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers in Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[5] C. Yi and Y. Tian, "Assistive text reading from complex background for blind persons," in *Proc. CBDAR*, Beijing, China, pp. 15–28, 2011.

[6] B. Epshtein, E. Ofek and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. CVPR*, San Francisco, CA, USA, pp. 2963–2970, 2010.

[7] Y. F. Pan, X. Hou and C. L. Liu, "A robust system to detect and localize texts in natural scene images," in *Proc. IAPR*, Nara, Japan, pp. 35–42, 2008.

[8] A. Pise and S. D. Ruikar, "Text detection and recognition in natural scene images," in *Proc. ICCSP*, Melmaruvathur, India, pp. 1068–1072, 2014.

[9] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou *et al.,* "EAST: An efficient and accurate scene text detector," in *Proc. CVPR*, Honolulu, HI, USA, pp. 2642–2651, 2017.

[10] Y. Nagaoka, T. Miyazaki, Y. Sugaya and S. Omachi, "Text detection by faster R-CNN with multiple region proposal networks," in *Proc. ICDAR*, Kyoto, Japan, vol. 6, pp. 15–20, 2018.

[11] Z. Zhong, L. Sun and Q. Huo, "An anchor-free region proposal network for faster R-CNN-based text detection approaches," *International Journal on Document Analysis and Recognition*, vol. 22, no. 3, pp. 315–327, 2019.

[12] Z. Tian, W. Huang, T. He, P. He and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. ECCV*, Amsterdam, The Netherlands, pp. 56–72, 2016.

[13] S. Long, X. He and C. Yao, "Scene text detection and recognition: The deep learning era," *International Journal of Computer Vision*, vol. 129, pp. 161–184, 2021.

[14] T. Mithila, R. Arunprakash and A. Ramachandran, "CNN and fuzzy rules based text detection and recognition from natural scenes," *Computer Systems Science and Engineering*, vol. 42, no. 3, pp. 1165–1179, 2022.

[15] N. Dilshad, A. Ullah, J. Kim and J. Seo, "LocateUAV: Unmanned aerial vehicle location estimation via contextual analysis in an IoT environment," *Internet of Things Journal*, pp. 1–1, 2022.

[16] R. Smith, "An overview of the tesseract OCR engine," in *Proc. ICDAR*, Curitiba, Brazil, pp. 629–633, 2007.

[17] R. W. Smith, "Hybrid page layout analysis via tab-stop detection," in *Proc. ICDAR*, Barcelona, Spain, pp. 241–245, 2009.

[18] R. Ani, E. Maria, J. J. Joyce, V. Sakkaravarthy and M. A. Raja, "Smart specs: Voice assisted text reading system for visually impaired persons using TTS method," in *Proc. IGEHT*, Coimbatore, India, pp. 1–6, 2017.

[19] R. Helbet, V. Monda, A. C. Bechet and P. Bechet, "Low cost system for terrestrial trunked radio signals monitoring based on software defined radio technology and raspberry Pi 4," in *Proc. EPE*, Iasi, Romania, pp. 438–448, 2020.

[20] A. Fathy, A. Ben Atitallah, D. Yousri, H. Rezk and M. Al-Dhaifallah, "A new implementation of the MPPT based raspberry Pi embedded board for partially shaded photovoltaic system," *Energy Reports*, vol. 8, pp. 5603–5619, 2022.

[21] Z. Youssfi, "Making operating systems more appetizing with the raspberry Pi," in *Proc. FIE*, Indianapolis, IN, USA, pp. 1–4, 2017.

[22] Logitech, "C922 USB camera," 2022. [Online]. Available: https://www.logitech.com/en-us/products/webcams/c922-pro-stream-webcam.960-001087.html.

[23] Ryanteck, "RTk traffic HAT board," 2018. [Online]. Available: https://learn.pi-supply.com/make/getting-started-with-rtk-traffic-hat/.

[24] C. K. Chng, Y. Liu, Y. Sun, C. C. Ng, C. Luo *et al.,* "ICDAR2019 robust reading challenge on arbitrary-shaped text-RRC-ArT," in *Proc. ICDAR*, Sydney, NSW, Australia, pp. 1571–1576, 2019.

[25] A. I. Jaided, "EasyOCR software," 2022. [Online]. Available: https://github.com/JaidedAI/EasyOCR.