Al-Salam Journal for Engineering and Technology
Journal Homepage: http://journal.alsalam.edu.iq/index.php/ajest
ISSN: 2790-4822

Scene Text detection and Recognition by Using Multi-Level Features Extractions Based on You Only Once Version Five (YOLO$_{v5}$) and Maximally Stable Extremal Regions (MSERs) with Optical Character Recognition (OCR)

Essam Haider Mageed[1]*, **Nidhal Khdhair El Abbadi**[2]

[1]computer science department, college of computer science and mathematics, university of kufa, Najaf, Iraq

[2]computer science department, college of education, university of kufa, Najaf, Iraq

*Corresponding Author: Essam Haider Mageed

DOI: https://doi.org/10.55145/ajest.2023.01.01.002
Received June 2022; Accepted September 2022; Available online September 2022

ABSTRACT: Textual information within scene images is very important in computer vision applications such as image retrieval based on its content, Tourist translator, and Navigation systems assistant. This paper presented a scene text recognition system based on YOLO. The main steps of the suggested methodology are: text detection and localization (using YOLO$_{v5}$), text segmentation (using Morphological processing as a new method), features extraction (using MSER$_s$), character segmentation and word segmentation (using bounding boxes with graph), and finally character recognition (using OCR). In this work we create a new dataset model that includes most of the text challenges such as Font type, Font size, Font color, and Font. The proposed system gives higher performance for detection, localization, and recognition when using dataset containing many challenges, the results were 80%, 96%, and 87.6 for precision, recall, and F-score respectively. Comparing with other similar works it was better. The accuracy of (OCR) is more than 99%.

Keywords: Text detection, Text Localization, YOLOv5, Total-Text, SVT, and LCT.

1. INTRODUCTION

A lot of people think that the world is submerged with hard information that is hard to deal with, both for individual humans and for the technology they use. A large spread of using a camera which is equipped with mobile or cars, etc. has a high impact on people's lives in various ways. Peoples start to capture images of anything around them. Also, the world wide web contains a huge image that contains purely textual information. A lot of communicated images contain text which uses a semiotic code of its kind [1].

Recently, recognition of scene text (extraction of text from natural scenes) has attracted great interest from the community, it's one of the active and challenging research areas in the field of digital image processing. Text appears in the natural sense in different types of objects such as road signs, billboards, product packaging, grocery item labels, and license plates [2]. The importance of extracting text is due to many real-life applications like understanding and recognizing road signs, geolocation, billboard, product packaging, analyzing multimedia content, application in autonomous cars, navigation systems, and assistance for the visually impaired [3].

High-level, significant semantic information is contained in the texts in scene images, aiding in the analysis and comprehension of the surrounding environment. With the fast uptake of smart phones and mobile computing devices, it is now easier and more productive to acquire photos with text data. Figure 1 show Examples of scene text images [4].

*Corresponding author: essamalmosawee@gmail.com

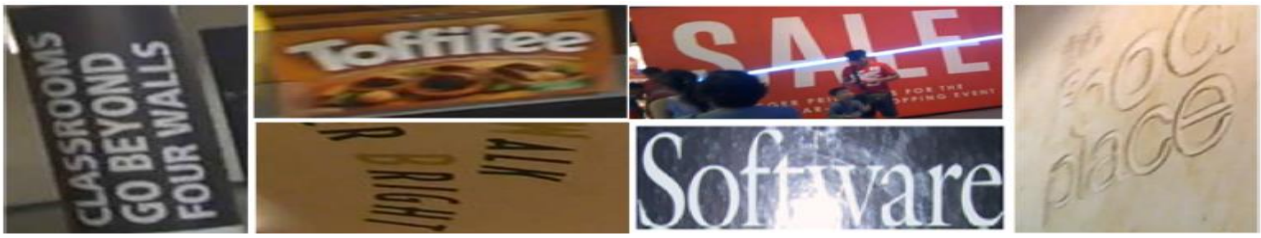


FIGURE 1. Examples of scene text [4].

In the previous methods, the problem of text recognition has been addressed many methods. One of these methods is by detecting the individual characters and then combining them into words. This method suffers from weak detection due to characters' intra-class variations.

In this paper, we choose an alternative path and propose, based on the location of the text in the image based on YOLO, and then recognize the text by segmenting it into many characters and words. The main goal of this paper is to develop a system for improving scene text recognition.

The rest of this paper is organized as follows: section two introduces the Related works are presented in section, followed by an overview of the research background. The methodology presented in section four, the results achieved are introduced in section five and analysis and discussion are introduced in section six. Finally, section seven focuses on the conclusion.

2. RELATED WORK

There are numerous studies in this field such as:

(Fedor Borisyuk, et al ,2018) Introduces Rosetta uses a two-step paradigm for its architecture. The first phase involves doing Faster-RCNN-based word detection. Utilizing a fully convolutional model with sequence-to-sequence (CTC) loss, word recognition is carried out in the second stage. The two models are individually trained. This two-step technique has a number of advantages, such as the capacity to independently support text recognition for several languages, conduct word recognition in parallel, and decouple training process and deployment changes to detection and recognition models [5]. One drawback is that it only applies to the COCO text data set and ignores all other data sets.

(Hui Li,et al, 2019) proposed framework for recognizing irregular text First, a 31-layer ResNet is fed with the input image to produce a 2D feature map. The feature map is then column by column encoded by an LSTM model, and the final hidden state is taken into account as a holistic feature of the input image. The holistic feature is converted into a series of characters using a different LSTM model. Depending on the current hidden state of the LSTM decoder, an attention module computes a weighted sum of 2D features (glimpse) at each time step of decoding. The 2D attention module implicitly and weakly supervises the irregularity of the text [6]. One of the drawbacks is that the recognition rate for the COCO text data set is 66%.

(Ebin Zacharias, et al, 2020) suggests a text translation pipeline for text found in natural situations. The objective is attained in two steps. The text region is first identified, which is a crucial step in the OCR engine's overall effectiveness. Second, Tesseract V5 is used to translate the recognized text area. Google's Tesseract OCR engine is precise and open-sourced [7]. After taking into account the false-positive situations, accuracy drastically decreased to 49%, which is one of the drawbacks.

(Xiqi Wang, et. al., 2021) A suggested technique known as R-YOLO inherits the YOLOv4 Fully Connected Network (FCN) topology. The end-to-end deep learning-based detection algorithm R-YOLO finds the inclined bounding boxes of the text in an image of a natural scene and categorizes them using a single, unified framework. For greatest pooling, the latest update from YOLO employs four scales of 1 1, 5 5, 9 9, and 13 13. The primary benefit of this approach is that it only addresses rotated text difficulties; other challenges remain unresolved. [8]. This method's identification of solely rotated texts while ignoring other difficulties like size is one of its drawbacks.

(Minghang He, et. al., 2021) A Multi-Oriented Scene Text Detector with Localization Refinement, or MOST, was presented. The pipeline of the proposed MOST is made up of a Position-Aware Non-Maximum Suppression (PA-NMS) module, a Position-Sensitive Map Prediction Head, a Text/Non-Text Classification Head, a Localization Branch, and a ResNet-50 backbone with a feature pyramid structure. The localization branch includes a Text Feature Alignment Module (TFAM), a coarse localization head, and a refined localization head [9]. This method gives a small percentage (77.9%) in determining the texts affiliated with the (MSRA-TD500) data set.

(Mingxin Huang et. al., 2022) They proposed SwinTextSpotter for text detection and text recognition. The overall architecture of SwinTextSpotter consists of four components: a backbone based on Swin-Transformer; a query-based text detector; a Recognition Conversion module to bridge the text detector and recognizer; and an attention-based recognizer [10]. This method gives a small percentage (66.9%) in determining the texts affiliated with the (Total Text) data set.

(Yuzhong Zhao, et. al., 2022) Explore Faster Localization Learning for Scene Text Detection. The proposed network FANet mainly includes Feature Extraction Network (FEN), Fourier Descriptor Proposal Network (FDPN), and Iterative Text Decoding Network (ITDN). Finally, get the detection results after applying Inverse Fourier transform (IFT), Inverse Normalization (IN), and Non-Maximum Suppression (NMS) to the refined Fourier proposals. The advantage of this method FANet can accurately detect scene text of arbitrary shapes with fewer training epochs [11]. This method gives a small percentage (62.9%) in determining the texts affiliated with the (Total Text) data set.

(Zobeir Raisi and John Zelek, 2022) Text Detection & Recognition in the Wild for Robot Localization. The proposed method of scene text spotting network based on deep learning using a transformer for VPR consists of using pre-trained models of the Vision Transformer architecture (ViT) as the backbone, adapting the single-scale ViT into the multi-scale FPN for capturing different resolution of text regions and extract feature with multi-scaling. The resulting feature maps are fed to the text of the final module to detect and recognize the text instance of a given image. The advantage of this method is being able to detect and recognize text allows the potential to leverage semantics and the features related to the detected text to better localize and map as opposed to just using indirect features [12]. This method is applied only on (SCTP) Dataset and this is not good because when applied on another data-set model maybe give less performance.

3. OVERVIEW

Graphic text and scene text are considered two basic classes of text, Graphic text is usually machine printed, found in captions, subtitles, and annotations in video and digital images on the web and email. Scene text, however, includes text on signs, packages, and clothing in natural scenes, and is more likely to include handwritten material. The scene text recognition system is consist of many steps such as Text detection (detect the existing text), Text localization (draw a box around the text), Text segmentation (isolate text regions from the background), feature extraction such as (Maximally stable extremal Regions (MSERs)) which extracted features that useful in detect only character(s), enhancements (enhancement the result from segmentation because maybe containing noise), and Text recognition (recognize the character as a real attribute using Optical Character Recognition (OCR)). The main steps in text recognition are Text detection and Text localization.

3.1 TEXT DETECTION AND LOCALIZATION

Finding the actual text among scene text images is what it requires. This activity is frequently difficult in the actual world because of problems like complicated backgrounds and varying text type, size, and color. Texture-based approaches, connected component (CC)-based methods, and deep learning-based methods make up the three main categories of current text detection and localization techniques. The text localization process is to find the location of text in the image. This is the most important in the text retrieval from the scene image. The effectiveness of this stage determines how well the text extraction performs overall. The extraction and identification will be more accurate the better the text has been localized. Text localization is a challenging and expensive computer procedure. There are three different types of text localization techniques: region-based, texture-based, and hybrid approaches (which mix region-based and texture-based techniques).[13]–[28] .

3.1.1 CONNECTED COMPONENT (CC) - BASED METHODS

Stroke width transform (SWT) and Maximally stable extremal Regions (MSER$_s$) are two representative methods in the field of scene text detection, which constitute the basis of a lot of subsequent works. The stroke width transform (SWT) is measured by the average distance between corresponding points. Finally, the stroke width is normalized by dividing the height of the bounding box. The Maximally stable extremal region (MSER) is an extremal region whose size remains virtually unchanged over a range of intensity levels.

3.1.2 TEXTURE - BASED METHODS

The concept behind the texture-based approach is that text in a picture has unique textural characteristics that can help us identify it from the background. The textural characteristics of a text section in an image can be detected using algorithms based on Gabor filters, Wavelet, and fast Fourier transformation (FFT). A filter that can instantly recognize text was first developed in the discrete cosine transform (DCT) domain. Although it operates quickly, this algorithm's detection accuracy is just moderate. The cascade scene text detector and localizer are later techniques.

3.1.3 DEEP LEARNING - BASED METHOD

Recently, convolutional neural networks (CNNs) have seen a lot of use. Being immune to geometric deformation, transformation, and lighting is one of CNN's key features. Direct information extraction from the image is possible at a low computational cost. modeled on sliding windows The text area is often slid using a sliding window of a defined size to locate the area most likely to contain the content. A multiscale sub-window is moved across all potential positions in an image as part of sliding window (SW) approaches to first detect text information. A pre-trained classifier is then used to determine if the text is present within the sub-window. For text localization, a support vector machine (SVM) is also utilized.

3.2 THE YOU ONLY LOOK ONCE VERSION FIVE (YOLOv5)

The YOLO algorithm's head separates the images used for detection into S×S grids, each of which has a unique detection task. Convolution layers and full connection layers make up the entire network structure. The tensor S×S (B×5 + C) is produced after the full connection layer, where B stands for the number of anticipated targets in each grid and C stands for the number of categories. By regressing the detection box position and evaluating the category probability of the tensor data, the final detection result may be determined. Each batch of training data is transmitted by the YOLOv5 algorithm through the data loader while being improved. Scaling, color space modification, and mosaic enhancement are three different types of data enhancements that the data loader is capable of carrying out. The YOLOv5's architecture is depicted in Figure 2. Jocher year 2020 can be used to summarize the YOLOv5 model as follows: Backbone: Focus structure, and CSP network; Neck: SPP block, and PANet; Head: YOLO head using GIoU-loss [29]–[31].

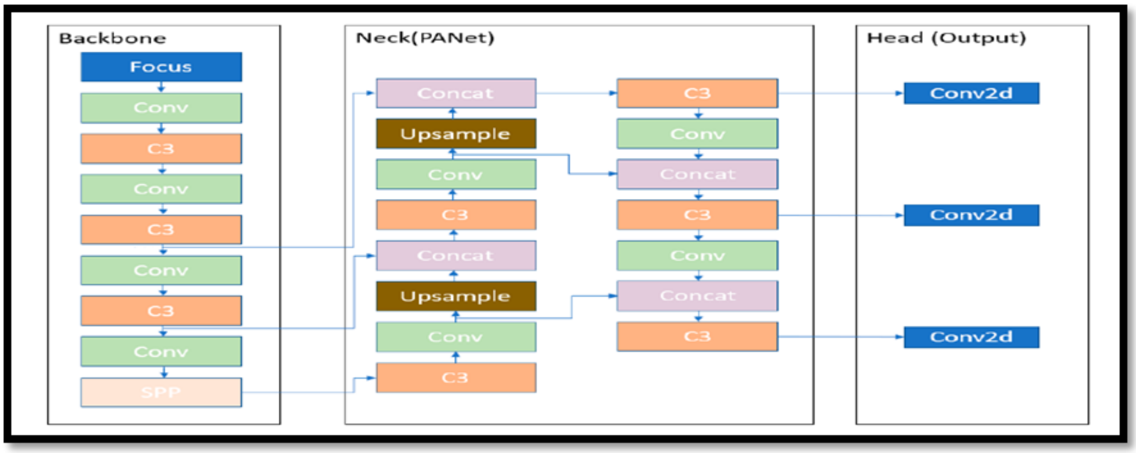


FIGURE 2. **The Whole diagram of YOLOv5** [29].

Those words are used in Figure 2: A convolution layer is indicated by Conv. Three convolutional layers make up C3, along with a module that is cascaded by a number of bottlenecks. The pooling layer known as spatial pyramid pooling (SPP) is utilized to free the network from its fixed size restriction. The preceding layer fusion with the closest node has been upsampled using the upsample technique. Concat, a layer that slices other layers, is used to slice the layer before it.

The final three Conv2d are network's top-level detection modules. In YOLOv5, the Focus layer makes its debut. In the YOLOv3 algorithm, the focusing layer takes the place of the previous three layers. Reduced CUDA memory requirements, a smaller layer, increased forward propagation, and increased backward propagation are all benefits of adopting the Focus layer.

3.3 MAXIMALLY STABLE EXTREMAL REGIONS (MSERs)

The Maximally Stable Extremal Regions (MSERs) method's main goal is to identify connected areas whose sizes don't significantly vary when the input image is binarized over a range of threshold values. The goal of this assignment is to identify "stable" binary regions by looking at the "growth history" of such regions over a range of threshold values. Finally, if the rate of size change over several thresholds is a local minimum, stable components are categorized as "maximally stable." Building a hierarchical component tree of extremal areas derived from the threshold sets at consecutive levels, then scanning the component tree for maximally stable extremal regions by analyzing their growth histories and computing key feature attributes [32]. Used this method in the proposed system to find a feature that is useful to detect characters and words. Algorithm 4.3 shows the steps to find Maximally stable extremal Regions (MSERs) for the input image.

Algorithm 4.3: Maximally stable extremal Regions (MSERs)

Input: Image I

Output: Maximally stable extremal Regions

Method:

Step 1 : Enter image I.

Step 2 : Image I is a mapping I: D $\subset \mathbb{Z}^2 \rightarrow S$ Extremal regions are well defined on images if:

Step 2.1 S is ordered (total, antisymmetric, and transitive binary relations $\leq$ exist)

Step 2.2 An adjacency relation A $\subset$ $D \times D$ is defined. We will denote that two points are adjacent as pAq.

Step 3 : **Region** Q is a contiguous (a.k.a. connected) subset of D. (for each $p.q$ $\in$ Q there is a sequence $p.a_1.a_2.\ldots.a_n.q$ *such as* : $pAa_1.a_1Aa_2.\ldots.a_{n-1}Aa_n.a_nAq$) Note that under this definition the region can contain "holes" (for example, a ring-shaped region is connected, but its internal circle is not part of Q.

Step 4 :(Outer) region boundary $\partial Q = \{q \in D\ Q: \exists p \in Q: qAp\} \partial Q = \{q \in D\ Q: \exists p \in Q: qAp\}$ which means the boundary ∂Q of Q is the set of pixels adjacent to at least one pixel of QQ but not belonging to QQ. Again, in case of regions with "holes", the region boundary is not obliged to be a connected subset of D (a ring has an inner bound and outer bound which do not intersect).

Step 5 :Extremal region Q $\subset$ D is a region such that either for all p$\in$ Q, q$\in$ ∂Q : $I(p) > I(q)$ (maximum intensity region) or for all p$\in$ Q, q$\in$ ∂Q : $I(p) < I(q)$ (minimum intensity region). As far as S is totally ordered, we can reformulate these conditions as $\min(I(p)) > \max(I(q))$ for maximum intensity region and $\max(I(p)) > \min(I(q))$ for minimum intensity region, respectively. In this form we can use a notion of a threshold intensity value that separates the region and its boundary.

Step 6 : Maximally stable extremal region Let Qi an extremal region such as all points on it have an intensity smaller than $i \in S$. Note $Qi \subset Q_{i+}\Delta$ for all positive $\Delta \in S$. Extremal region Qi_* is maximally stable if and only if $|Q_{i+}\Delta_{\backslash} Q_{i-}\Delta|/|Qi|$ has a local minimum at $i_{*.}$ (Here |.| denotes cardinality). $\Delta \in S$ is here a parameter of the method.

The concept more simply can be explained by the thresholding method. All the pixels below a given threshold are 'black' and all those above or equal are 'white'. Given a source image, if a sequence of threshold result images I_t is generated where each image t corresponds to an increasing threshold t, first a white image would be seen, then 'black' spots corresponding to local intensity minima will appear then grow larger. A maximally stable extremal region is found when the size of one of these black areas is the same (or near the same) as in the previous image.

3.4 OPTICAL CHARCTER RECOGNITION (OCR)

Optical character recognition (OCR) converts text inside the image into real attribute text. An open-source OCR engine called the first Tesseract was created at HP between 1984 and 1994. The following actions are part of the Tesseract. [33]–[35]:

1.Input Binary Image.

2.Page Layout Analysis: splits multi-column text into columns and divides an image into text and non-text parts.

3.Blob Finding: A blob is a hypothetically classifiable entity that could consist of one or more Connected Components that are horizontally overlapping and their inner nested outlines or holes.

4.Find Text Lines and Words: Text-line finding is done by sub-dividing text regions into blocks of uniform text size and line spacing.

5. Word recognition: There are two ways to complete this stage: Successful words—those in a dictionary and not dangerously ambiguous—are sent to an adaptive classifier on the first pass for training. In the event that the adaptive classifier has learned more information during the first pass over the word, words that failed to pass must be analyzed again on the second pass.

6.Fuzzy Space and x-height Fix-up: The x-height estimation procedure first establishes the constraints on the maximum and minimum permissible x-height based on the initial line size computed for the block before moving on to the final fuzzy-space resolution phase, in which doubtful spaces are chosen.

7.Output: Text.

3.5 CHARACTER(S) AND WORD(S) SEGMENTATION

The main process for Character(s) and word(s) segmentation is the isolation of text regions from non-text regions. There are two methods for removing non-text, using Stroke width transform (SWT), and using geometric properties such as Aspect ratio, Eccentricity, Euler number, Extent, and Solidity [36], [37]. The modification of the segmentation of characters and words is that not to use the following two methods: Remove Non-Text Regions Based On Basic Geometric Properties, and Remove Non-Text Regions Based On Stroke Width Variation.

3.5.1 CHARACTER(S) SEGMENTATION

The main process for this method is shown in algorithm 4.5.1. Note that bounding box (x_{min}, y_{min}, x_{max}, y_{max}). In the top-right and bottom-left corners of the region, the initialization function of a pixel p with coordinates (x, y) is a quadruple (x, y, x + 1, y + 1), and the combining operation ⊕ is (min, min, max, max), where each operation is applied to its respective item in the quadruple. The width w and height h of the region are calculated as xmax-xmin and ymax-ymin respectively. ⊕ is an addition (+). There is an additional step to find if the font is regular or not based on the height of each character (if the size regular for each character this lead to be regular font). Figure 3 shows the assumption in detail to find font shape (regular or irregular) and font size.

```
Algorithm 4.5.1: character(s) segmentation
Input: Cropped Text from (YOLO$_{v5}$)
Output: detect character(s), find font size, and test if the font is regular or not
Method:
Step 1 : Input Cropped Text (with RGB color)
Step 2 : Apply the average filter.
Step 3 : Detect MSER$_s$ features and return MSER$_s$ Regions object
Step 4 : Measure properties of image regions with property type (Bounding Box)
Step 5 : Find Concatenate arrays vertically between (step 3 and step4)
Step 6 : Find the bounding box ($x_{min}$, $y_{min}$, $x_{mx}$, $y_{max}$).
Step 7 : Detect character(s) based on (step 6).
Step 8 : Take only one character and find its size, it was based on the assumption
Step 9 : Print that( "regular text with size= height of character").
```

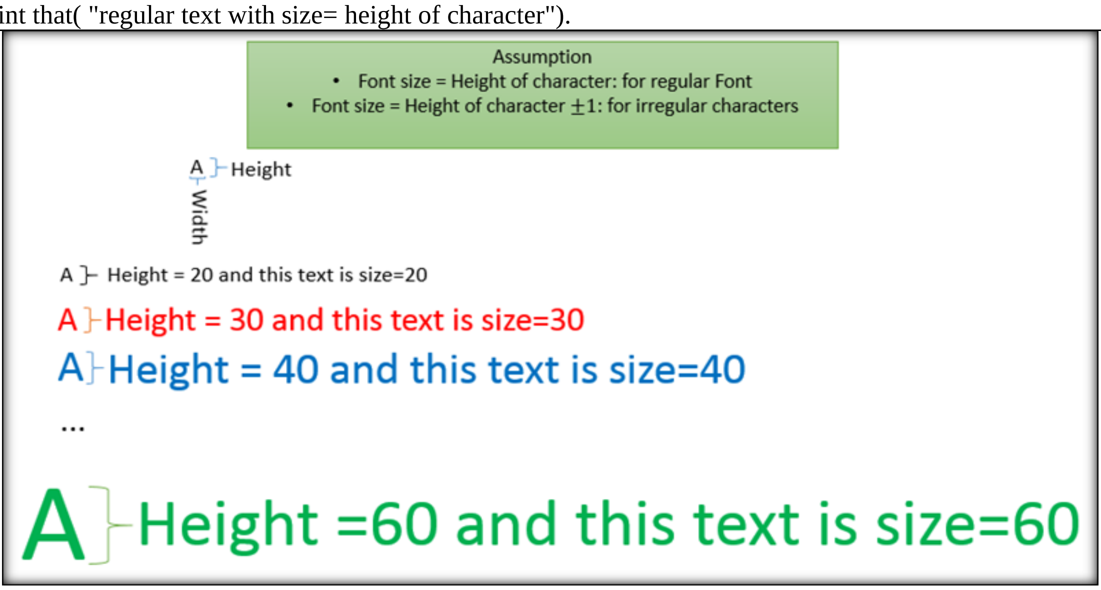

Assumption
• Font size = Height of character: for regular Font
• Font size = Height of character ±1: for irregular characters
A
Height
Width
A Height = 20 and this text is size=20
A Height = 30 and this text is size=30
A Height = 40 and this text is size=40
...
A Height =60 and this text is size=60

FIGURE 3. Our Assumption to find Font shapes and font types that are used in Character(s) segmentation

3.5.2 WORD(S) SEGMENTATION

Word recognition is used to segment the text inside the image into words and isolated it, and based on modified method (Character(s) segmentation). The main process for this method is shown in algorithm 4.5.2. Produced by applying (OCR) on Word(s) segmentation each word is recognized separately while on Remove Background the text is fully recognized.

```
Algorithm 4.5.2: word(s) segmentation
Input: Cropped Text from (YOLO$_{v5}$) with character box from character segmentation
Output: detect word(s)
Method:
Step 1 : Input Cropped Text with character box from character segmentation
```

Step 2 : Find overlap between (the bounding box and itself).
Step 3 : Set overlap values to zero
Step 4 : Create Graph based on the overlap ratio.
Step 5 : Find text regions that are connected with the graph created from step (4).
Step 6 : Merge the boxes based on the minimum and maximum dimensions from step (5).
Step 7 : Compose the merged bounding boxes using the [x y width height] format.
Step 8 : Remove bounding boxes that only contain one text region using histogram bin(s).
Step 9 : Detect word(s).

3.6 EVALUATION PARAMETER

Precision (P), recall (R), and F-measure are all used in the traditional evaluation methodologies for text detection, word spotting, and end-to-end recognition. (F). Precision is the ratio of accurately identified text areas to all text regions that were detected. Recall is defined as the ratio of correctly identified text areas to all text regions in the dataset. Recall and precision were combined to generate the F-measure, a single quality metric [8] These testing procedures are described as:

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \\ F = 2 \times \frac{P \times R}{P+R} \end{cases} \qquad (1)$$

4. PROPOSED SYSTEM

The current proposal system consists of the following steps: Pre-Processing, YOLO$_{v5}$ model, Text segmentation, Character(s) Segmentation, word(s) segmentation, and finally recognizing text. Figure 4 shows the steps for training, while Figure 5 shows the steps for testing the proposed system for detecting and recognizing scene text images.

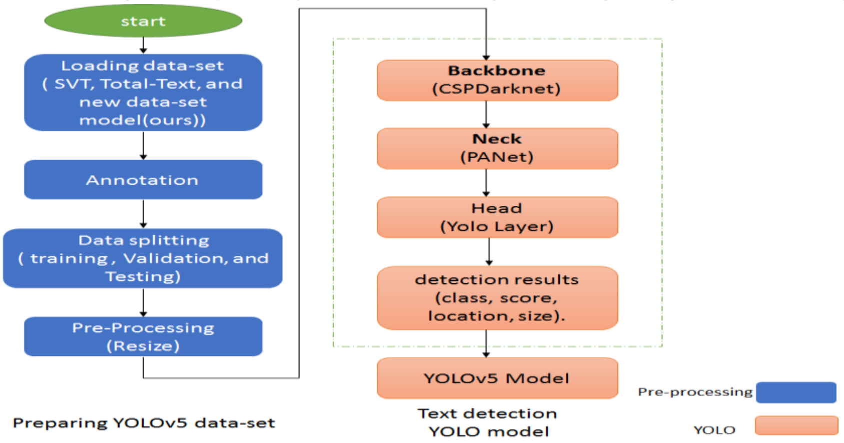

start
Loading data-set (SVT, Total-Text, and new data-set model(ours))
Annotation
Data splitting (training , Validation, and Testing)
Pre-Processing (Resize)
Preparing YOLOv5 data-set
Backbone (CSPDarknet)
Neck (PANet)
Head (Yolo Layer)
detection results (class, score, location, size).
YOLOv5 Model
Text detection YOLO model
Pre-processing
YOLO

Figure 4. **Training Stages.**

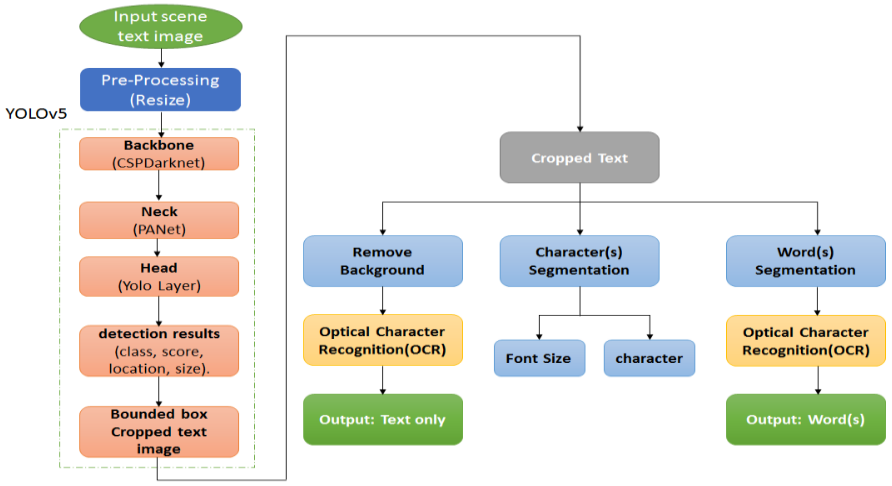


Figure 5. Testing Stages.

4.1 DATA SET USED

There are three data-set used in this proposal, two of them are public datasets which are SVT and Total-Text. While the third one is a private dataset which is regarded as more complex than the public dataset due to a wide variety of challenges.

4.1.1 The Street View Text (SVT)

Google Street View was used to collect this dataset. This data, which includes 350 images, has a significant degree of variability and frequently has low quality. It is split as 212 images for training and 42 for validation and 91 images for testing and ignore the reset 5 images [38]. Figure 6 is a sample of an SVT data-set.

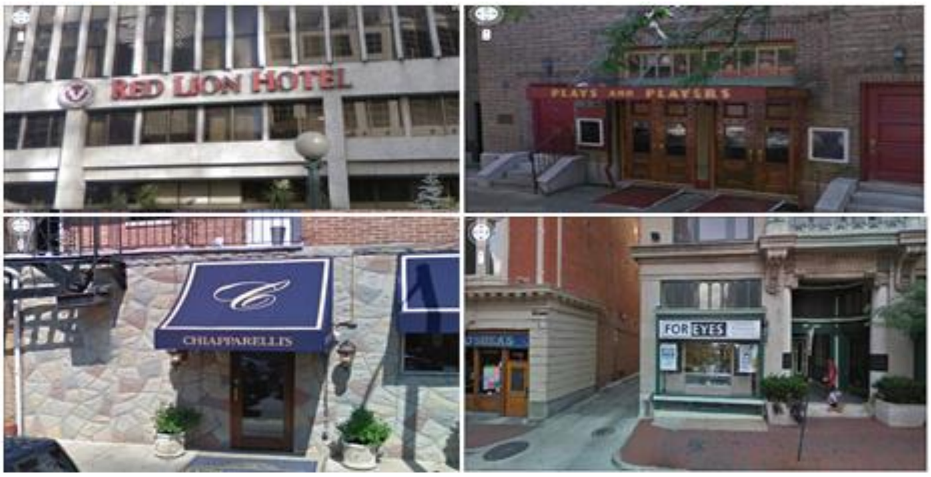


FIGURE 6. Samples of an SVT database with different font types.

4.1.2 The Total-Text data-set

The Total-Text text detection dataset consists of 1,555 images. It is divided into 844 training photos, 241 validation images, 426 testing images, and leaves out the reset 44 images. Using several text types, such as curved, multi-oriented, and horizontal text instances [38]. Figure 7 is a sample of a Total Text database.

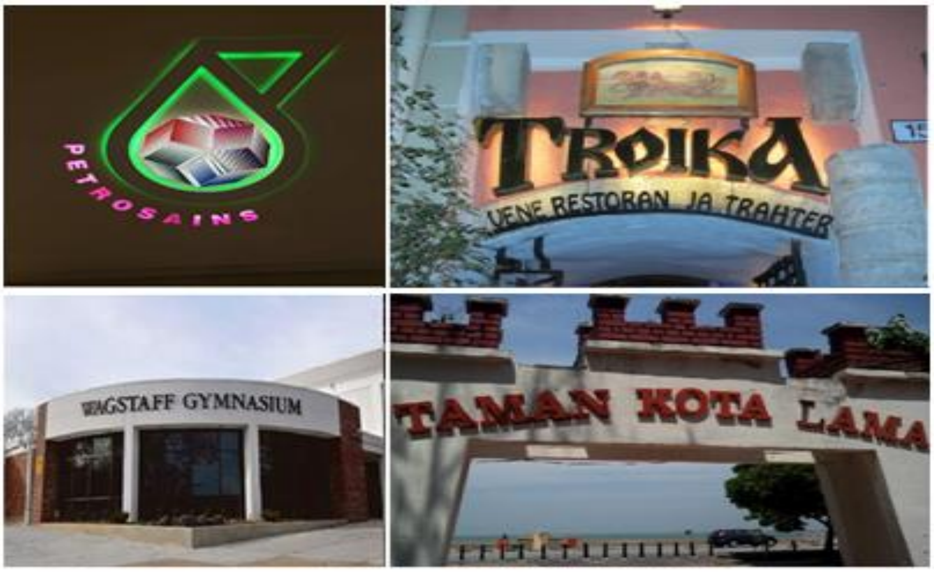


FIGURE 7. Samples of a Total Text database with curve text.

4.1.3 Local Challenges Text data-set (LCT)

This dataset is created by authors and includes (1837) images with text that contains the most challenges in this field. The text language in this data set is English. The type of text fonts of the text included in the images is (Arial, Andulas, Calibri, Batang, and Times New Roman**)**, with four font colors which are (Red, Yellow, Blue, and Black). Also, the images contain various text organizations, in addition to the horizontal text, there are texts inclined with angles (45°, 90°, 135°, 180°, and 280°). Finally, the font size is also different, we select the font size (20, 24, 26, 30, 40, and 70) for the text of created dataset (LCT). The dataset is divided into three groups, one for training consisting of 1265 images, the validation group consisting of 303 images, and the test group included 269 images. Samples of the LCT dataset are shown in Figure 8.

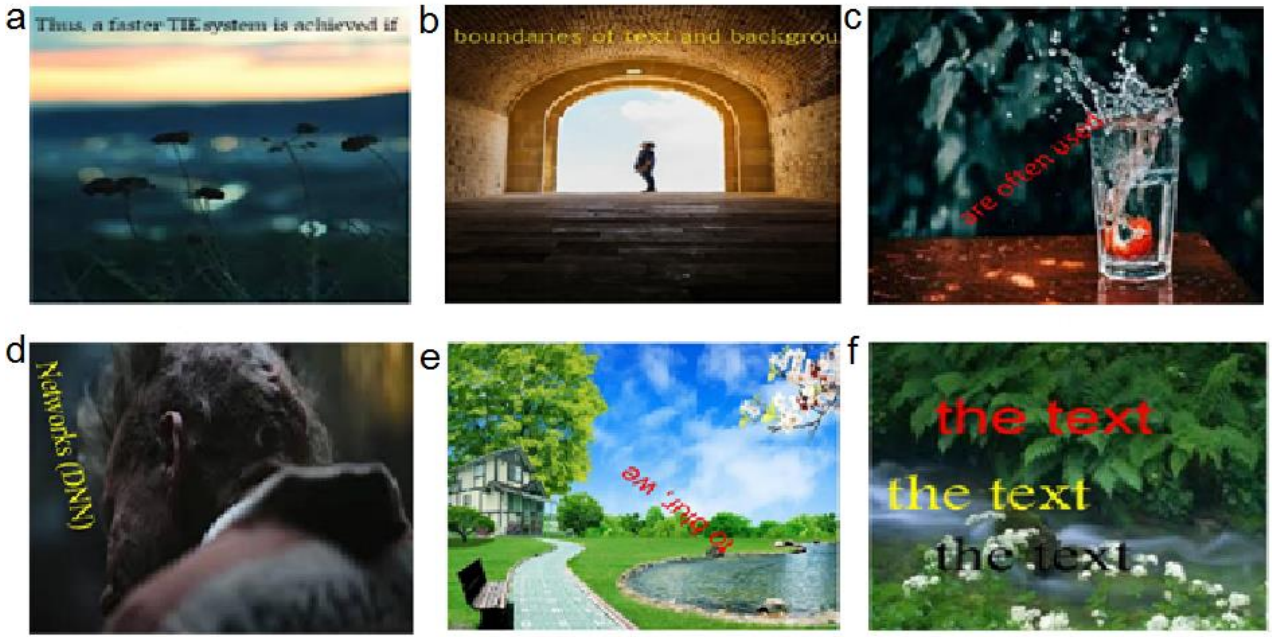

a
Thus, a faster TIE system is achieved if
b
boundaries of text and backgrou
c
are often used
d
Networks (DNN)
e
to blur, we
f
the text
the text
the text

FIGURE 8.- **(a) Andulas type with size 26 and black color; (b) Batang types with size 26 and yellow; (c) Calibri with angle 45°; (d) Times New Roman with angle 280°; (e) Arial Type with 135°; (f) Three text regions with different locations: first one Arial type with font size equal 70, second one Andulas type, and third one Times New Roman.**

4.2 PRE-PROCESSING

The preprocessing step is limited to resizing the input images into (416×416).

4.3 TEXT DETECTION AND LOCALIZATION BY YOLO$_{V5}$

In the current proposal, we try to detect and localize the texts from images based on a deep-learning model called the You Only Lock Once (YOLO$_{v5}$) model. To achieve the goal of this paper, YOLO must be trained on the part of dataset, and then the training network is tested on the different data and measures the algorithm performance.

4.3.1 TRAINING PHASE

In the training stage, we used three types of data-set: SVT, Total-Text, and the LCT data set. YOLOv5 was used to detect the text and localized it. Algorithm 5.3.1 shows the training steps.

```
Algorithm 5.3.1: Training Phase
Input: SVT data-set, Total Text data-set, and LCT.
Output: Text detection and localized
Method:
Step 1 : Loading data-set
Step 2 : Annotate (label) the images.
Step 3 : preprocessing
Step 4 : Create the YOLO$_{v5}$ model as:
                    Step4.1  Backbone: CSP Darknet
                    Step4.2  Neck: PANet
                    Step4.3   Head: YOLO layer
Step 5 : Train Custom YOLOv5 Detector.
Step 6 : detect text and determine its location.
```

4.3.2 TESTING PHASE

The trained model will be tested to determine the proposed method's performance. Testing the suggested model included many stages which are summarized in algorithm 5.3.2.

```
Algorithm 5.3.2: Testing Phase
Testing algorithm input: Scene text image.
Testing algorithm output: Real text as an attribute.
Method:
Step 1 : Enter the scene text image
Step 2 : Preprocessing
Step 3 : Apply the $YOLO_{v5}$ model.
Step 4 : Detect the text and bound the text region by drawing a box around the text region.
Step 5 : Segment the text region.
Step 6 : Cropped text from the text region, this is achieved by the following processing:
        Step6.1 Find the text border and remove the background from the text region (segment the
            characters and words exactly without background).
        Step6.2 Apply Optical Character Recognition (OCR) to find real text based on the text border from
            the previous step.
Step 7 : Segment the Character(s) by the following process:
        Step7.1 Cut only one character based on the dimensions of the bounding box [ x y w h]
        Step7.2 Find the size of character [W H]
Step 8 : Determine the font size.
Step 9 : Recognize Word(s): achieved by the following process:
        Step9.1 Apply Word(s) segmentation
        Step9.2 Apply Optical Character Recognition (OCR) for each word(s).
```

4.4 REMOVE BACKGROUND

The first step in removing the background depends on defining the edges of characters and words based on the proposed morphological process, the accuracy of this process leads to easy recognition of the text. This process isolates the text object from the background. The steps of recognition of the text are shown in algorithm 5.4.

```
Algorithm 5.4: Remove Background
Input: Cropped Text from ($YOLO_{v5}$)
Output: border of text(s) only
Method:
Step 1 Input Cropped Text (with RGB color)
Step 2 Convert the RGB image into a grayscale image.
Step 3 Detect Cell using edge detection with Sobel filter;
Step 4 Dilate the result from (step 3) with two Structure elements first one with a line of 90° $\begin{bmatrix}1\\1\\1\end{bmatrix}$
and the second with 0° $\begin{bmatrix}1 & 1 & 1\end{bmatrix}$ to enlarge the border of cell detection.
Step 5 Fill the gaps that resulted from the dilation process using the Holes type.
Step 6 Remove the connected object in the border from the filling process in (step 5).
Step 7 Apply the erosion process twice for the result from (step 6) using structure element with diamond type. This
    step leads to smoothing the edge of the text.
Step 8 Find the perimeter of objects in the image result from (step 7).
Step 9 Take the grayscale image in (step 2) and convert every location of this image with each perimeter into 255: this
    lead to detecting the border of text only.
Step 10 Draw a text border based on (step 9).
```

5. THE RESULT

In this section, the proposed system is evaluated using two standard data sets and a new data set, and it is compared to a number of existing methods. Analysis and discussions regarding our system are also presented in the detail. Figure 9 and Figure 10 show the result of our proposed system.

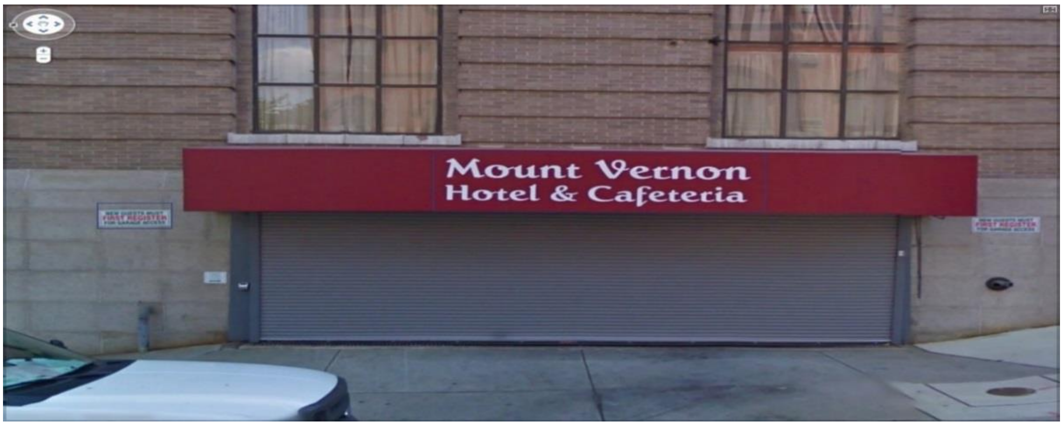
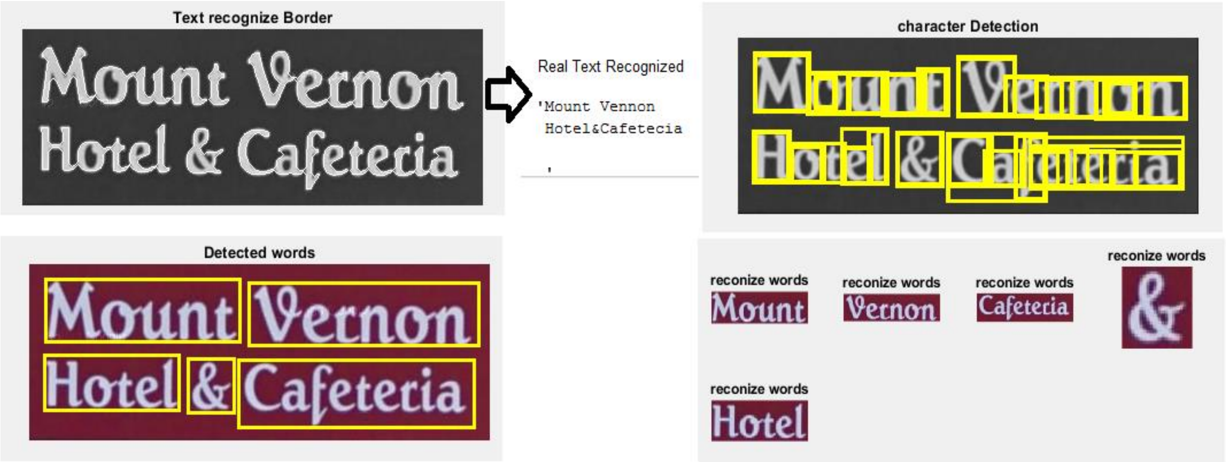


FIGURE 9. **Detection and Localization for SVT database with a blue box around text.**



```
'Mount Vennon
 Hotel&Cafetecia
'
```

FIGURE 10. **The result for whole system (text, character(s), and word(s)) segmentation.**

The terms used in the Figure 10 are: In Text recognizer Border Find the text border and remove the background from the text region (segment the characters and words exactly without background). In character detection this include detect only character and then find font shape and its size. In detected words this mean detect each word isolate from other words and the result for this step is to recognize each word.

5.1 CHALLENGES THAT WERE SOLVED DURIN OUR NEW DATA-SET MODEL(LCT)

The main problem of recognizing text in a scene is how to detect text inside an image. To do this, many challenges must be taken into consideration to deal with. Four challenges were solved during the creation of the new data-set model: First: Color which means characters with different colors are difficult to deal with; Second Character size, which means character size changing may lead to unpredictability to recognize text; Third: Font type: Many font types can be used; Fourth: Direction, means not all characters is in the same direction. Figure 11 shows all challenges that were produced during the new data-set model and detected and localized through the use of YOLOv5.

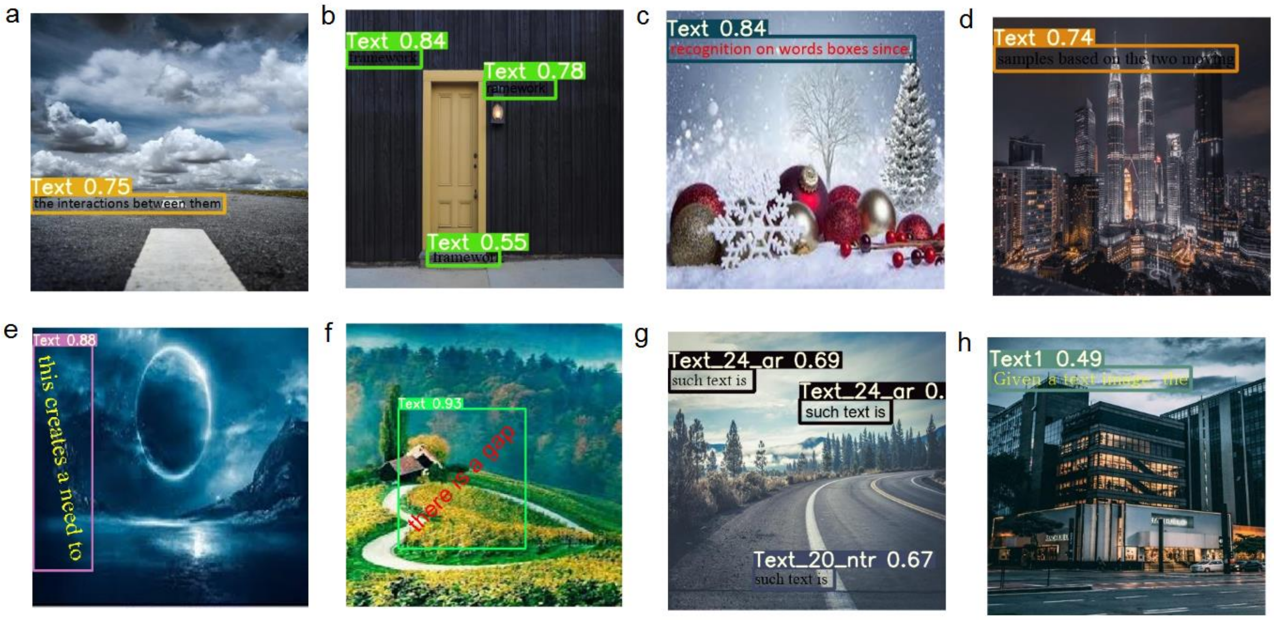


FIGURE 11.- **(a) Calibri type with size 22; (b) three text in different locations with Arial, Andulas, and Times New Roman types and size of 20; (c) Calibri types with size 26; (d) Times New Roman with size 26; (e) Times New Roman with size 26 with angle 280º; (f) Arial type with 26 and angle 45º; (g) three locations with Arial, Andulas, and Times New Roman with size 24; (h) Batang type with size 26.**

5.2 EVALUATION

The input image size must be 416×416 for all data-set types. For the SVT data set the separation of labeling images is: 212 images for training, 42 for validation, and 91 for testing, and there are 5 images ignored by roboflow. Train SVT data-set with batch size=15, and the number of epoch=400. The result for SVT data-set with Precision rate is 74.5%, Recall rate is 58.6%, and F-score is 65.6%. The separate for Total Text data-set is separate into:844 into training, 241 for validation, and 426 for testing and there are 44 images ignored by roboflow. Train Total Text data-set with batch size=16, and a number of epoch=400. The result for the Total Text data-set with Precision rate is 55.7%, Recall rate is 62.2% and F-score is 58.7%. The separate of new data-set model Local Challenges Text(LCT) is separated into: 1265 for training, 303 for validation, and 269 for testing. train new data-set model with batch size=32, and the number of epoch=200 this is an optimal number of epoch for this data-set to give better performance. The result for the new data set with Precision rate is 80%, Recall is 96%, and F-score is 87.6%. Table 1 below shows the contents of architecture components that were used during YOLOv5 and its implementation.

Table 1. - **Evaluation of our system that used YOLOv5 in detail.**

Data-set types	Labeling separation				Batch size	Number of epoch	Precision	Recall	F_score (MAP@0.5)
	Training	Validation	Testing	Discarded images					
SVT	212	42	91	5	15	400	74.5	58.6	65.6
Total-Text	844	241	426	44	16	400	55.7	62.2	58.7
LCT(our)	1265	303	269	zero	32	200	80	96	87.6

6. ANALYSIS AND DISCUSION

Compare the new databases with the SVT data set. From the Evaluation Parameter described in section (4.2- third part) the analysis of the system that applied to SVT data-set, Total Text data-set, and new data-set model we found that the values of criteria it is very suitable for scene text detection and localization and still good for scene text recognition. the proposed system when compared with another method we note that the F-measure increases from 62.9% to 87.6%, and recall rate increases from 67.7% to 96%. Table 2 show the result for the proposed system and a comparison with another method. Figure 12 shows the analysis of the proposed system (our).

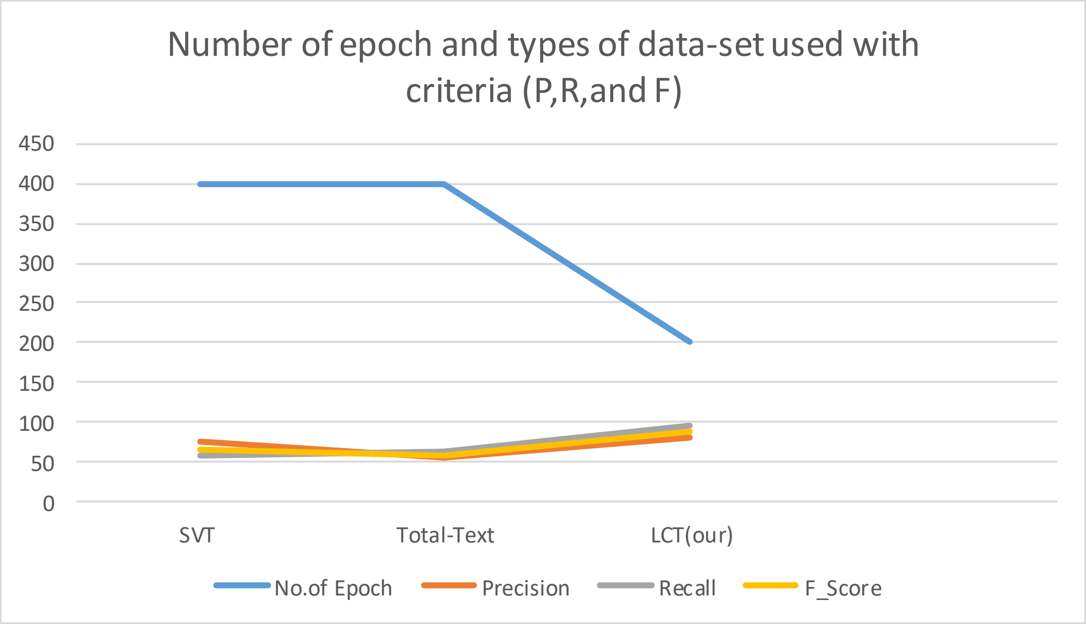


FIGURE 12. The criteria analysis for detection and localization for the data-set used in our system show the relation between the number of epochs (No. of epoch) and criteria (Precision, Recall, and F_score).

Table 2. - Comparison of the proposed system on SVT, Total-Text, and new data-set model (LCT) with another method based on P, R, and F-score represent the precision, recall, and F-measure respectively.

Method	Database type	Precision (P) %	Recall(R) %	F_score (F1) %
MOST[7]	IC15	76.7	75.2	75.9
	MSRA-TD500	81.2	74.8	77.9
SwinTextSpotter[8]	RoIC13 with angle 45°	72.5	83.4	77.6
	RoIC13 with angle 60°	72.1	84.6	77.9
FANet[9]	CTW1500	79.2	87.4	83.1
	MSRA-TD500	58.8	67.7	62.9
Snooper Text [18]	SVT	36	54	43
Word spotting in the wild ECCV [19]	SVT	67	39	49
DeconvNet [21]	Total-Text	33	40	36
Synthetic data for text localization in natural images. [26].	SVT	26.2	27.4	26.7
R-YOLO[27]	MSRA-TD500	81.9	90.2	85.8
	ICDAR2013	82.9	90.1	86.4
Convolutional Character Networks - Textboxes [28]	Total-Text	45.5	62.1	52.5
maximally stable extremal region (MSER) [37]	SVT	0.38	0.69	0.49
The proposed system (our)	**SVT**	**74.5**	**58.6**	**65.6**
	Total-Text	**55.7**	**62.2**	**58.7**
	Local Challenges Text	**80**	**96**	**87.6**

(LCT)

7. CONCLUSION

As far as I know, this is the first use of YOLOv5 for detection and localization at the same time, unlike the previous methods that detect and localized separately. We detect and recognize text from images with many challenges such as (different font types, sizes, colors, Orientations), and different image illumination. Build Local Challenges Text (LCT) data -set is a new data-set model that is very useful because it solved many challenges that still problem in another method to solve scene text recognition problem. A new way is proposed to remove most of the background by drawing a box around the text and then segmenting only the text region. MULTI-Level features extractions Using Maximally stable extremal Regions (MSER) on the result from You Only Look Once version five (YOLO_{v5}). Character segmentation is a very important step to test if the text is regular or not. Determine the font size based on character height, we discover the relation between character height and font size. Words isolation based on the word(s) segmentation this lead to be each word is recognizing separately from other. Finally, recognize text during Optical Character Recognition (OCR). Future work The system will be applied to the video, after converting the video into a number of frames.

REFERENCES

[1] J. Libovický, “Text Extraction from Image Data.” Charles University in Prague, 2015.

[2] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, “Robust scene text recognition with automatic rectification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4168–4176.

[3] C. Bartz, J. Bethge, H. Yang, and C. Meinel, “Kiss: Keeping it simple for scene text recognition,” *arXiv Prepr. arXiv1911.08400*, 2019.

[4] H. Lin, P. Yang, and F. Zhang, “Review of scene text detection and recognition,” *Arch. Comput. methods Eng.*, vol. 27, no. 2, pp. 433–454, 2020.

[5] F. Borisyuk, A. Gordo, and V. Sivakumar, “Rosetta: Large scale system for text detection and recognition in images,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 71–79.

[6] H. Li, P. Wang, C. Shen, and G. Zhang, “Show, attend and read: A simple and strong baseline for irregular text recognition,” in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, no. 01, pp. 8610–8617.

[7] E. Zacharias, M. Teuchler, and B. Bernier, “Image processing based scene-text detection and recognition with tesseract,” *arXiv Prepr. arXiv2004.08079*, 2020.

[8] X. Wang, S. Zheng, C. Zhang, R. Li, and L. Gui, “R-YOLO: A real-time text detector for natural scenes with arbitrary rotation,” *Sensors*, vol. 21, no. 3, p. 888, 2021.

[9] M. He *et al.*, “MOST: A multi-oriented scene text detector with localization refinement,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8813–8822.

[10] M. Huang *et al.*, “SwinTextSpotter: Scene Text Spotting via Better Synergy between Text Detection and Text Recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4593–4603.

[11] Y. Zhao, Y. Cai, W. Wu, and W. Wang, “Explore Faster Localization Learning For Scene Text Detection,” *arXiv Prepr. arXiv2207.01342*, 2022.

[12] Z. Raisi and J. Zelek, “Text Detection & Recognition in the Wild for Robot Localization,” *arXiv Prepr. arXiv2205.08565*, 2022.

[13] B. Shi, X. Bai, and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2016.

[14] X. Zhou *et al.*, “East: an efficient and accurate scene text detector,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[15] C. K. Ch’ng and C. S. Chan, “Total-text: A comprehensive dataset for scene text detection and recognition,” in *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, 2017, vol. 1, pp. 935–942.

[16] S. P. Faustina Joan and S. Valli, “A survey on text information extraction from born-digital and scene text images,” *Proc. Natl. Acad. Sci. India Sect. A Phys. Sci.*, vol. 89, no. 1, pp. 77–101, 2019.

[17] W. He, X.-Y. Zhang, F. Yin, Z. Luo, J.-M. Ogier, and C.-L. Liu, “Realtime multi-scale scene text detection with scale-based region proposal network,” *Pattern Recognit.*, vol. 98, p. 107026, 2020.

[18] S. Bin Ahmed, S. Naz, M. I. Razzak, and R. Yusof, “Arabic cursive text recognition from natural scene images,” *Appl. Sci.*, vol. 9, no. 2, p. 236, 2019.

[19] J. Ma *et al.*, “Arbitrary-oriented scene text detection via rotation proposals,” *IEEE Trans. Multimed.*, vol. 20, no. 11, pp. 3111–3122, 2018.

[20] X. Liu, G. Meng, and C. Pan, “Scene text detection and recognition with advances in deep learning: a survey,” *Int. J. Doc. Anal. Recognit.*, vol. 22, no. 2, pp. 143–162, 2019.

[21] Y. Zhu, C. Yao, and X. Bai, “Scene text detection and recognition: Recent advances and future trends,” *Front. Comput. Sci.*, vol. 10, no. 1, pp. 19–36, 2016.

[22] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, “Text detection, tracking and recognition in video: a comprehensive survey,” *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, 2016.

[23] L. Neumann and J. Matas, “Efficient scene text localization and recognition with local character refinement,” in *2015 13th international conference on document analysis and recognition (ICDAR)*, 2015, pp. 746–750.

[24] P. B. Chavre and A. Ghotkar, “A survey on text localization method in natural scene image,” *Int. J. Comput. Appl.*, vol. 112, no. 13, 2015.

[25] C. Yu, Y. Song, and Y. Zhang, “Scene text localization using edge analysis and feature pool,” *Neurocomputing*, vol. 175, pp. 652–661, 2016.

[26] A. Zhu, R. Gao, and S. Uchida, “Could scene context be beneficial for scene text detection?,” *Pattern Recognit.*, vol. 58, pp. 204–215, 2016.

[27] A. Kaur, R. Dhir, and G. S. Lehal, “A survey on camera-captured scene text detection and extraction: towards Gurmukhi script,” *Int. J. Multimed. Inf. Retr.*, vol. 6, no. 2, pp. 115–142, 2017.

[28] S. Wang, C. Fu, and Q. Li, “Text detection in natural scene image: a survey,” in *International Conference on Machine Learning and Intelligent Communications*, 2016, pp. 257–264.

[29] W. Wu *et al.*, “Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image,” *PLoS One*, vol. 16, no. 10, p. e0259283, 2021.

[30] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, “A Review of Yolo algorithm developments,” *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022.

[31] D. Thuan, “Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm,” 2021.

[32] W. Burger and M. J. Burge, “Maximally Stable Extremal Regions (MSER),” in *Digital Image Processing*, Springer, 2022, pp. 765–795.

[33] R. Smith, “An overview of the Tesseract OCR engine,” in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, 2007, vol. 2, pp. 629–633.

[34] R. Smith, D. Antonova, and D.-S. Lee, “Adapting the Tesseract open source OCR engine for multilingual OCR,” in *Proceedings of the International Workshop on Multilingual OCR*, 2009, pp. 1–8.

[35] R. W. Smith, “Hybrid page layout analysis via tab-stop detection,” in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 241–245.

[36] A. Gonzalez, L. M. Bergasa, J. J. Yebes, and S. Bronte, “Text location in complex images,” in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 617–620.

[37] L. Neumann and J. Matas, “Real-time scene text localization and recognition,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3538–3545.

[38] S. Panda, S. Ash, N. Chakraborty, A. F. Mollah, S. Basu, and R. Sarkar, “Parameter tuning in MSER for text localization in multi-lingual camera-captured scene text images,” in *Computational Intelligence in Pattern Recognition*, Springer, 2020, pp. 999–1009.