



ARTICLE

Intelligent Image Text Detection via Pixel Standard Deviation Representation

Sana Sahar Guia¹, Abdelkader Laouid¹, Mohammad Hammoudeh^{2,*} and Mostafa Kara^{1,3}

¹LIAP Laboratory, University of El Oued, El Oued, Algeria

²Information and Computer Science Department, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

³National Higher School of Mathematics, Scientific and Technology Hub of Sidi Abdellah, P.O. Box 75, Algiers, 16093, Algeria

*Corresponding Author: Mohammad Hammoudeh. Email: m.hammoudeh@kfupm.edu.sa

Received: 30 September 2023 Accepted: 27 February 2024 Published: 17 July 2024

ABSTRACT

Artificial intelligence has been involved in several domains. Despite the advantages of using artificial intelligence techniques, some crucial limitations prevent them from being implemented in specific domains and locations. The accuracy, poor quality of gathered data, and processing time are considered major concerns in implementing machine learning techniques, certainly in low-end smart devices. This paper aims to introduce a novel pre-treatment technique dedicated to image text detection that uses the images' pixel divergence and similarity to reduce the image size. Mitigating the image size while keeping its features improves the model training time with an acceptable accuracy rate. The mitigation is reached by gathering similar image pixels in one pixel based on calculated values of the standard deviation σ , where we consider that two pixels are similar if they have approximately the same σ values. The work proposes a new pipeline approach that reduces the size of the image in the input and intermediate layers of a deep learning model based on merged pixels using standard deviation values instead of the whole image. The experimental results prove that this technique significantly improves the performance of existing text detection methods, particularly in challenging scenarios such as using low-end IoT devices that offer low contrast or noisy backgrounds. Compared with other techniques, the proposed technique can potentially be exploited for text detection in IoT-gathered multimedia data with reasonable accuracy in a short computation time. Evaluation of the MSRA-TD500 dataset demonstrates the remarkable performance of our approach, Standard Deviation Network (σ Net), with precision and recall values of 93.8% and 85.6%, respectively, that outperform recent research results.

KEYWORDS

Text detection; machine learning; IoT image processing; adaption

1 Introduction

Computer vision in general, in recent years, has become one of the major computer science research topics, where efficient text detection in multimedia content is one of the major challenges. Advancements in artificial intelligence have significantly improved various facets of computer vision, allowing for the automation of numerous tasks. In the realm of image classification, various machine-learning algorithms are frequently used [1]. Deep Learning (DL) stands out as a subset of these



algorithms, with one of its key strengths being its capacity to automatically extract pertinent features from the data, thereby diminishing the necessity for manual feature selection [2].

With the rapid advancement of mobile computing devices and the massive proliferation of smart devices [3], images containing text data are being collected more readily, conveniently, and efficiently. Therefore, recognizing text within an image has become an active research topic in computer vision. The texts in the image often contain rich semantic information of high-level importance that helps in analysis, guidance, and understanding of the corresponding environment. Therefore, detecting and recognizing texts in images have received increasing attention in several applications [4], including automatic navigation, image retrieval, human-computer interaction, security, data integrity [5–9], etc. For example, considering the application of autonomous vehicles, a field rapidly gaining momentum. These self-driving cars navigate complex urban environments, relying on many sensors and cameras to perceive their surroundings. In this context, accurate text detection can be a game-changer.

The extraction and analysis of text from images and videos have emerged as focal points within multimedia understanding systems. Numerous methods have been put forward for extracting text from visual media, and many studies have offered comprehensive reviews [10–13] on this subject. Text recognition and extraction address two primary challenges when locating text within a target image. The first challenge is identifying the geometric representation that encapsulates the text's position, which may take the form of a contour, such as a square, rectangle, oriented rectangle, or quadrilateral. The second task is to transform text-containing regions within images into machine-readable strings. Hence, the first crucial step in text identification is text discovery [14]. Text within an image can be classified into two categories: Scene text, which is inherent in the image at the time of capture, and artificial text, which is introduced into the image during post-production processes [15].

On the other hand, the Internet of Things's significance in the text detection domain cannot be overstated. In today's digital information, a substantial portion of multimedia data emanates from IoT devices and embedded systems, underscoring the sheer volume and diversity of information generated.

Integrating IoT into various applications underscores the pivotal role of text detection in computer vision. This paper addresses IoT-generated multimedia data's critical challenges, focusing on enhancing text detection methodologies to extract valuable information efficiently. IoT-generated data containing embedded text is indispensable for streamlining shipping information processing in logistics and supply chain management domains. Similarly, in the realm of smart cities, the utilization of IoT is instrumental in text detection applications such as smart parking systems, enabling effective traffic management through license plate and signage recognition.

This paper's primary target is to propose novel approaches tailored to the intricacies of IoT-generated multimedia data. Specifically, our research endeavors to enhance text detection algorithms to cope with the unique challenges of variable image quality, diverse orientations, and complex environments inherent in IoT-driven scenarios. We aim to contribute significantly to advancing robust and efficient text detection methods within IoT frameworks by addressing these challenges.

However, a pivotal challenge that emerges is the often poor quality of text data extracted from these sources. Text detection in this context necessitates robust solutions capable of real-time processing and maintaining high levels of accuracy. In response to this challenge, researchers and engineers have proposed a spectrum of innovative solutions designed to address the multifaceted concerns surrounding poor-quality data. These solutions not only enhance the accuracy of text detection but also ensure that it operates seamlessly in real-time scenarios, ultimately facilitating the effective utilization of IoT-generated multimedia data in various applications. The following points, among others, render the detection task more difficult [16]: (1) The absence of prior information

regarding the text's location within the image presents a significant challenge, particularly when the text is distributed throughout the scene. Knowledge of factors such as the number of text lines, line spacing, and word count can significantly streamline the process, especially in the case of scanned documents. Consequently, the lack of such pre-established formatting rules makes direct segmentation implementation more challenging in these image types.

1. Text within an image can appear in many sizes, fonts, and orientations. It may even include embossed or uniquely designed characters, such as calligraphy logos, presentation slides, or messages displayed on a digital bulletin board. Recognizing text with such distinctive and non-traditional appearances poses a significant challenge.
2. The quality may differ from one photo to another, depending on the digital device the image was taken with, which may in several cases be poor.
3. Images often contain numerous patterns resembling letters, set against complex backgrounds, where certain unfamiliar objects like icons, windows, and leaves may closely resemble letters and words. Various other elements may be interwoven with the text, resulting in new styles and representations.

Therefore, one of the critical sources of information in the image and video is the inside text. For instance, the caption text may explain information about where and when the events in the video occur and possibly who participated in the events, and the banner text is used as a visual indicator for navigation and notification in the scene. Additionally, among the different types of objects that appear in the videos, the text that contains abundant semantic information plays an important role in many practical applications, such as video annotation and multimedia retrieval [17]. The accuracy of text extraction is critical for the reliability and effectiveness of these applications, while faster processing times enhance efficiency and user experience. Achieving the right balance between accuracy and speed is often a key consideration in developing text extraction systems. Fig. 1 shows an overview of text detection in images in the literature. In the pre-processing phase of image text detection, particularly in practical real-world applications, substantial efforts have been poured into the pursuit of swift and resilient algorithms [18–21]. The most important, among other challenges, is the imperative for outstanding performance, which includes achieving an acceptable processing time and a high recall rate while maintaining a low false alarm rate. These requested efficiencies must be accomplished while accommodating an intricate web of variables. These variables encompass the intricacies of font size, font color, textual orientation, linguistic diversity, and the often perplexing intricacies of diverse backgrounds. Nonetheless, a further challenge lies in the arduous task of distinguishing text from an eclectic array of text-like elements, ranging from leaves and window curtains to generic textures. Adding to the complexity is the ever-evolving nature of text patterns, as they meander through variations in font sizes, colors, and languages. To complicate matters further, the pervasive influence of noise and the capricious nature of image encoding and decoding procedures conspire to undermine the integrity and quality of the text.

Moreover, the factor of treatment time emerges as a crucial consideration in this landscape. Swift and robust text detection algorithms must also account for timely processing to meet the demands of real-world applications. The pertinence of time in text detection further underscores the complexity of this multifaceted challenge.

Compared with proposed approaches, recent text detection algorithms often face the following challenges:

1. **Text Orientation:** Text can be written in different orientations, such as horizontal, vertical, and diagonal. Detecting text in non-horizontal orientations can be challenging for text detection algorithms.
2. **Font Size and Style:** Text detection algorithms can struggle with recognizing text with a small font size, written in a different style, or complex structure.
3. **Language and Script:** Text detection algorithms may be trained in specific languages or scripts and cannot detect text in other languages or scripts.
4. **Background and Contrast:** Text detection algorithms can have difficulty detecting text if the background is complex or if the contrast between the text and the background is low.
5. **Image Quality:** The image quality can significantly affect text detection algorithms' performance. Low-quality images or images with noise or distortion can make it difficult for the algorithms to detect text accurately.
6. **Computational Resources:** Text detection algorithms can be computationally intensive and may require significant computational resources to operate efficiently. This can be a limitation in some scenarios where computational resources are limited.

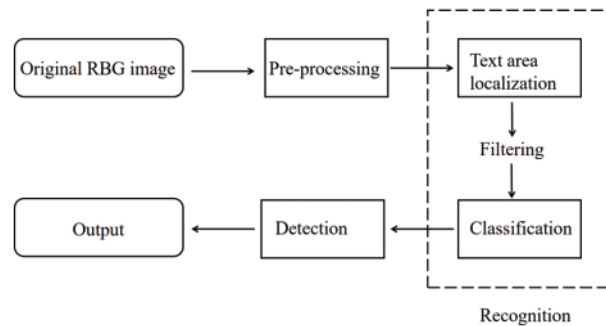


Figure 1: A comprehensive vision of text detection in images

The primary goal of this paper is to achieve the right balance between accuracy and speed in developing text extraction systems, so the significant contribution lies in addressing this balance, ensuring that text extraction systems not only maintain high accuracy but also operate efficiently in terms of speed.

The purpose is to address the three last challenges, where we propose a text detection approach focusing on the pre-processing phase of Fig. 1. To reach this goal, the proposed approach considers the values of the computed σ instead of the whole pixels' image while preserving its features to reduce the data size by reducing both the input image channels and the tensor channels of the intermediate layer of the deep learning model. We run a deep learning model without pre-trained models to reduce the computational time speed and increase the accuracy compared with original RGB images. To improve the accuracy, we introduce σ in different stages of the proposed pipeline approach.

The rest of the paper is organized as follows: Section 2 presents a background and some related works. In Section 3, the proposed approach is explained. Section 4 shows the experimental evaluation and results. The manuscript is concluded in Section 5.

2 Background and Related Work

This section presents an overview with an analysis of text detection techniques of recent and relevant work. The discussed text detection techniques in this section will be reviewed depending on their operating modes and characteristics. We organize our review by classifying text detection techniques into traditional Machine Learning-based (ML-based) approaches and deep learning-based approaches.

Machine learning is a powerful tool that aids in a multitude of tasks, particularly in prediction. Its versatility lies in exploring various methods and algorithms, allowing for effective analysis and interpretation of data [22].

However, traditional ML-based approaches rely on handcrafted features and classifiers. These features could include gradient-based features, edge detection, and texture analysis. The common classifiers used in traditional ML-based approaches are Support Vector Machines (SVM), decision trees, and random forests. These techniques prove their good performance in several applications, which explains their wide uses. However, these approaches have some critical limitations, such as their dependency on the quality of handcrafted features, which could be sensitive to lighting conditions, and their inability to generalize well to unseen data.

Deep learning-based approaches in text detection are often based on neural networks that can learn features automatically from data. Convolutional Neural Networks (CNNs) proved to be effective in text detection tasks. CNNs can learn hierarchical representations of visual features, enabling them to detect text in images with varying orientations, fonts, and sizes. Several deep-learning architectures have been proposed for text detection, including Faster R-CNN, YOLO, and SSD. These architectures exhibited outstanding performance on various text detection benchmarks.

2.1 A Taxonomy of Text Detection Techniques

The main characteristic of text detection is the features' design and testing. Deep learning incorporates many modern methods and enables researchers to avoid the cumbersome work of designing and testing handcrafted features. To provide a comprehensive investigation of the existing techniques for text detection, we classify existing techniques into two categories [23], before deep learning and in deep learning, as shown in Fig. 2.

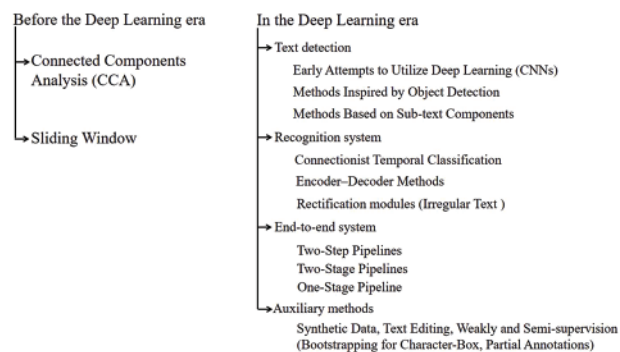


Figure 2: Text detection taxonomy

- Before deep learning: The text detection methods relied on two main techniques: (a) *Connected Components Analysis*, which extracts the candidate components through various methods such as color grouping and then filters the non-text components using manually designed rules, as

well as automatically trained on handcrafted features. (b) *Sliding Window* where each window is specified as text areas. Those deemed positive are then grouped into text regions.

- In deep learning: Deep learning techniques automatically learn features from the input images. In this era, researchers no longer need to design or test handcrafted features, as deep learning models can independently learn the most relevant features for text detection. There are four main known deep learning types:
 - (1) Text detection to localize text in natural images. This method went through several stages, starting with early attempts to utilize deep learning, which uses CNNs passing by methods inspired by object detection and methods based on sub-text components.
 - (2) Recognition strategy that transforms the content of the detected text regions into linguistic characters. This strategy contains several methods, including connectionist temporal classification, encoder-decoder methods, and rectification modules for Irregular text.
 - (3) End-to-end strategy, where text detection and recognition are performed in one unified pipeline. Several techniques can be cited, such as two-step, two-stage, and one-stage pipelines.
 - (4) Auxiliary methods, where these kinds of approaches aim to support the principal task of text detection and recognition. Among these methods, we can mention synthetic data, text editing, and weakly and semi-supervision (bootstrapping for character-box, partial annotations).

2.2 Related Work

Advanced technology offers the possibility to place modern camera systems in different places, such as mobile phones, surveillance systems, and autonomous cars, generating high-quality images at a lower cost. The huge amount of generated images and videos increases the demand for the systems to read and interpret these images. In fact, the rapid development of Deep Neural Networks (DNN) has made significant progress in detecting text in scene images, which explains why recently several works have been released using DNN.

Scene text detection can be included taxonomically under general object detection. The generic object detection methods' designs inspire many scene text detection algorithms. However, scene text discovery has different characteristics and challenges that require independent methodologies and solutions. Huang et al. [24] used CNNs to classify local image patches into text and non-text classes. The authors proposed scooping such image patches by using MSER features. Text Flow [25] applied CNNs to the whole images in a fully convolutional approach to detect characters. In [26], a convolutional neural network is utilized to determine if a pixel in the input image belongs to characters if it is inside the text region and the text orientation around the pixel. Hereafter, they considered the connected positive responses as detected characters or text regions. Sun et al. [27] presented a text detection technique belonging to the Connected Component (CC) based methods that divide text detection into two sub-issues: CC generation and text/non-text classification. The proposed technique uses neural networks and Color-Enhanced Contrasting external regions (CER). The authors of [28] presented text recognition in natural scenes. Two steps were performed to achieve the goal. Firstly, detecting the text area which is an important function in the general performance of the OCR engine. The second step translates the detected text area using the accurate and open-sourced OCR engine Tesseract V5. In their work, the authors found that minor lighting and text angle differences resulted in significant text recognition errors. Sayahi et al. in [15] proposed a technique for detecting text in the scene file. Using neural networks and wavelet transformation to classify pixels into text and non-text areas. Yan et al. [29] presented a Uyghur language text detection system in complex background images for intelligent vehicles. They proposed a channel-enhanced maximally stable extremal regions (MSERs) algorithm to detect component candidates.

Using the ICDAR 2003 database, Hanif et al. [30] proposed a text detector that uses a small set of heterogeneous features spatially (choosing the Likelihood Ratio Test (LRT) as a weak classifier) combined to build an extensive set of features; the text segment is equal to 32×16 pixels. They presented a modified AdaBoost technique named CAdaBoost that considers the complexity of the weak classifiers at the feature selection step. The authors used a neural network to learn the localization rules automatically for text localization. As an application of Intelligent Transportation Systems (ITS), the authors in [31] proposed an approach to detect traffic boards in street-level images to recognize their information. After applying blue-and-white segmentation, the technique focuses on extracting local descriptors at some key points of interest. These images are then represented as a bag of visible words and categorized using Naïve Bayes vector machines. Finally, the images in which a pass-through was detected are considered to apply the text-detection method to it. Using the reverse geocoding service, a language model based on a dynamic dictionary for a limited geographical area is adapted to automatically read and save the information shown in the panels. Non-Latin families of cursive scripts like Hindi, Arabic, and Chinese pose a challenge for text detection from natural scene images. Arafat et al. [32] presented a technique to detect, predict orientation, and recognize Urdu ligatures in images. The authors have used the FasterRCNN algorithm and CNNs such as Resnet50 and Googletnet on images of size 320×240 pixels for detection and localization.

In [33], the authors introduce an efficient pipeline that combines simplicity with effectiveness, delivering swift and precise text detection in natural scenes. This approach directly anticipates words or text lines with arbitrary orientations and quadrilateral shapes, utilizing a single neural network.

The authors proposed a Differentiable Binarization (DB) module in [34] to perform the segmentation network's binarization process. In [35], the authors incorporated a DB module and an Adaptive Scale Fusion (ASF) module into the segmentation network, improving the network's performance by fusing multi-scale features. Optimized along with a DB module, a segmentation network can adaptively set the thresholds for binarization, simplifying the post-processing and enhancing the performance of text detection. A new approach is presented in [36] for text detection that combines a text proposal model with a deep relational reasoning network using a local graph and a Graph Convolutional Network (GCN). The resulting network is end-to-end trainable, allowing for efficient and accurate arbitrary shape text detection. The authors of [37] propose a novel approach for representing arbitrary-shaped text contours using the Fourier Contour Embedding (FCE) method, which converts text instances into compact signatures in the Fourier domain. Based on FCE, the FCENet is developed with a backbone, feature pyramid networks, and post-processing with the Inverse Fourier Transformation and Non-Maximum Suppression, enabling end-to-end training for arbitrary-shaped text detection. The proposed FCE method is accurate and robust in fitting highly-curved text contours, and FCENet achieves good generalization for detecting arbitrary-shaped text in scenes.

A Mask R-CNN-based approach for text detection that can detect multi-oriented and curved text in natural scene images is described in [38]. To enhance the feature representation ability of Mask R-CNN, the authors proposed to use the Pyramid Attention Network (PAN) as a new backbone network; PAN can more effectively suppress false alarms caused by text-like backgrounds.

The authors of [39] proposed an efficient and accurate arbitrary-shaped text detector called the Pixel Aggregation Network. It consists of a low computational-cost segmentation head and a learnable post-processing module. The segmentation head comprises a Feature Pyramid Enhancement Module (FPEM) and a Feature Fusion Module (FFM), introducing multi-level information to guide better segmentation. The FFM gathers features from FPEMs of different depths for final segmentation. Pixel Aggregation (PA) implements the learnable post-processing, which precisely aggregates text pixels

using predicted similarity vectors. The proposed method achieves high accuracy on various benchmark datasets while being computationally efficient.

In [40], the authors proposed a Progressive Scale Expansion Network (PSENet) to enable the accurate detection of text instances with arbitrary shapes. PSENet achieves this by generating kernels of different scales for each text instance and gradually expanding the most small-scale kernel to cover the complete shape of the text instance. Because the minimal scale kernels have large geometrical margins between them, this method effectively splits closely spaced text instances, making it easier to use segmentation-based methods for detecting text instances with arbitrary shapes.

This paper proposes a text detection technique based on the channel-reducing step. The idea is based on the standard deviation value extracted from both RGB images to generate new images with clear objects and tensors of intermediate layers of the deep learning model. We implement the algorithm of the released work in [41] to reach this goal. Then, we use these newly generated images instead of the original images in the learning process. The reason for using this procedure is that in this proposed σ Net, we apply the standard deviation over the RGB images in the σ step to avoid using any pre-trained model.

The geometric representation of text positions is foundational in various domains, ensuring accurate and efficient transformation of text-containing regions into machine-readable formats. Several examples and case studies illustrate the significance of these processes.

In scene text recognition [42], geometric representation aids in capturing the spatial layout of text through methods like bounding boxes or quadrilateral shapes. This enhances accuracy in recognizing text within natural scenes. Geometric representation is crucial for aiding visually impaired individuals. By transforming text into machine-readable strings, systems can provide valuable assistance in interpreting and interacting with visual content.

In autonomous driving systems, geometric representation is employed for sign recognition [43]. This enhances the understanding of visual content, contributing to safer and more efficient transportation.

While numerous existing methods in the field of text detection in images primarily rely on pre-trained deep learning models, the proposed approach takes a distinctive and innovative path by eschewing the use of pre-trained models. This decision carries several distinct advantages that set our work apart from the existing literature:

- **Reduced Processing Time:** By avoiding the need for pre-trained models, our approach significantly reduces the computational overhead typically associated with fine-tuning or adapting these models to specific tasks. This translates into faster text detection, making it well-suited for real-time or resource-constrained applications. Table 1 illustrates the use of a pre-trained model by the most relevant research. Our approach uses no pre-trained model to reduce computational load and increase processing speed.
- **Customization and Adaptability:** Our model's independence from pre-trained models allows for greater flexibility in adapting to various domains and datasets. Researchers and practitioners can tailor the algorithm more effectively to their specific needs, enabling better performance across various scenarios.
- **Competitive Precision Rates:** Our approach maintains competitive precision rates despite not relying on pre-trained models.

Table 1: Text detection results using different pre-trained models

Method	Pretrained model
DB-RenNet [33]	ImageNet
DBNet [34]	Synthtext (model lg)
DRRG [35]	ImageNet
FCENet [36]	ImageNet
σ Net (Proposed model)	Without pre-trained model

3 Text Detection Pipeline Approach Specifications

We present a novel approach to image text detection that achieves state-of-the-art limits while significantly reducing the pipeline's computational complexity. This work focuses on the initial idea proposed in [41] to define the σ step.

The present paper aims to adjust the adequate regions depending on the text detection in automation and robotics domains [44], such as automatic driving, visual search, and robot navigation, to quickly recover precious and meaningful textual information in image scenes.

There are some specifications in each multimedia domain, such as the background and the salient object colors. The best way to detect the salient objects is by eliminating the background and unnecessary objects. Therefore, if the user application area is well-defined, it will be easy to detect and extract the existing texts by considering them salient objects in this work.

So, the pipeline uses the standard deviation principle (σ step) in several pipeline stages. The first step applied to the original image reduces the three RGB channels to one, as shown in Eqs. (1) and (2). Algorithm 1 illustrates in detail these steps, where for each pixel, we compute the distance to (0, 0, 0) and extract its σ value. In the end, Algorithm 1 returns the value of σ of each pixel depending to its distance to the RGB cube corner. The binary semantic segmentation established by the fully convolutional network (FCN) architecture upgraded with the σ step called σ Net in the context of text detection and extraction. The result of the deep learning model is a saliency map that is thresholded to obtain the binary image. Finally, we use Connected Component Analysis (CCA) to generate a bounding polygon that encloses each text region. The first published idea in [41] applies the σ step on only the input RGB three-channel image without using the deep learning model. Fig. 3 presents the proposed approach by showing the role of the standard deviation image in the segmentation process, where for the general tensor T of n channels, we follow a consistent process for each group of three consecutive channels T_i within T with i ranges from 1 to $\frac{n}{3}$. As result, we obtain a single-channel tensor σ_i from the three channels of tensor T_i . This method allows us to preserve the most pertinent information in the image or tensor while significantly reducing the computational complexity of subsequent operations. Text in scene images could be in random places, sizes, and colors. Furthermore, texts in images could be multi-oriented and curved or vertically.

Our approach has two main contributions based on [41]: First, it reduces the number of channels in the image sensor using the standard deviation of the image pixels [41], enabling us to achieve better performance with fewer computational resources. Second, it uses a deep learning model for text detection that leverages both downsampling and upsampling operations and skip connections, with a standard deviation of intermediate tensors, to preserve high-resolution information and improve the

accuracy of the final output. The proposed approach is designed to be a fast and light deep-learning text detection model.

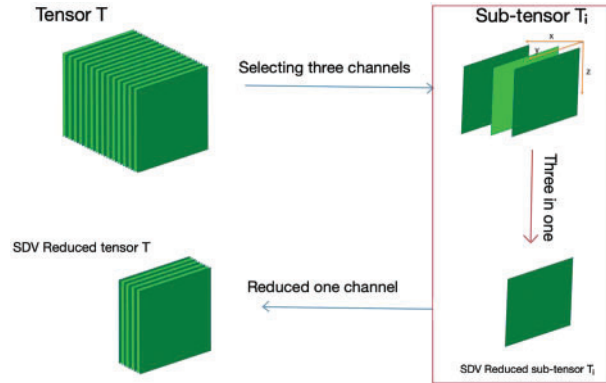


Figure 3: Illustration of channel reduction principle

3.1 Reducing the Channel Step (σ Step)

A wide variety of machine learning and computer vision applications have used the clustering method as an unsupervised algorithm [45]. Clustering as an unsupervised machine learning method has great utility certainly in data science [46]. It groups a subset of the data set characterized by most similar features into the same cluster. A subset of dataset interclusters has the most dissimilarity features [47]. In unsupervised learning, unlabeled datasets are provided for training, and unsupervised algorithms can find clusters of data samples based on the notion of distance or another similarity measure [48]. The color divergence approach is a novel clustering method used for image segmentation and object detection that will be exploited in this work [41].

We name the standard deviation step of an image (RGB image defined as three channels) the tensor T of n channels with $n \geq 3$ as a generalization. In our study case of a tensor T , we take each of three successive channels T_i of T for $i = \{1, \dots, \lfloor \frac{n}{3} \rfloor\}$ and apply Algorithm 1, which resumes the idea of the σ step.

The output of Algorithm 1 will be used instead of using the three values of RGB pixel, where the standard deviation will keep the feature of the computed pixel and reduce its size from three to one value. To understand the secret behind using standard deviation, we discuss our study case by starting with the cube representation of the three channels tensor T_i . In this case, we define the cube C with eight vertices $j = (0 \dots 7)$ in such a way that encloses the tensor. In this representation, we consider each element t_{ik} for $k = \{1, \dots, w \times h\}$ of T_i as have three coordinates (x_k, y_k, z_k) with w and h is the width and the height of the channel's tensor:

- x_k corresponding to the value of the element in the first channel
- y_k corresponding to the value of the element in the second channel
- z_k corresponding to the value of the element in the third channel

Algorithm 1: Standard deviation step

Require: T_i, C ;

Ensure: σ_i ;

(Continued)

Algorithm 1 (continued)

```

1. for ( $j \leftarrow 0..7$ ) do
2.  $C \leftarrow C + vertexe_j$ ;
3. end for
4. for ( $k \leftarrow 0..w \times h$ ) do
5. for ( $j \leftarrow 0..7$ ) do
6.  $d_{jk} \leftarrow \sqrt{x_{jk}^2 + y_{jk}^2 + z_{jk}^2}$ ;
7. end for
8.  $\sigma_{tik} \leftarrow \frac{\sqrt{\sum (d_{jk} - averageD_{tik})^2}}{8}$ 
9.  $\sigma_i[k] \leftarrow \sigma_{tik}$ 
10. end for
11. return  $\sigma_i$ 

```

Then, we compute the Euclidean distance between each element of T_i and the eight vertices of the cube $C_j = (0 \dots 7)$ as shown in Fig. 3. The cube representation of three channels tensor T_i involves calculating the distance between each element t_{ik} of T_i and the cube's eight vertices using Eq. (1). A distance vector $D_{tik} = [d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7]$ is defined for each element t_{ik} , where d_j is the distance between the element t_{ik} and vertex j of the cube. The distance is calculated using the formula

$$d_{jk} = \sqrt{x_{jk}^2 + y_{jk}^2 + z_{jk}^2} \quad (1)$$

where x_{jk}, y_{jk} and z_{jk} are the difference between the element t_{ik} of the tensor and the corresponding vertex j . We use Eq. (2) to calculate the standard deviation vector σ , which can enhance the accuracy of the outputs.

$$\sigma_{tik} = \frac{\sqrt{\sum (d_{jk} - averageD_{tik})^2}}{8} \quad (2)$$

The distance values d_{jk} represent the distance between the element t_{ik} and the vertices $j = (0 \dots 7)$ of the cube. $averageD_{tik}$ denotes the element average distance vector D_{tik} .

Finally, from the three channels of the tensor T_i we obtain one channel σ_i tensor each element corresponds to σ_{tik} for $k = \{1, \dots, w \times h\}$.

The σ step is applied first to the input RGB three-channel image to reduce it to one σ channel, as illustrated in Fig. 3. After that, it is applied to the tensor in the intermediate layer of the deep learning model.

This method enables us to retain the most relevant information in the image or the tensor while significantly reducing the computational complexity of subsequent operations.

3.2 Deep Learning Text Detection Model

Segmentation-based approaches are commonly employed in scene text detection [34]. The significance of image classification and segmentation in object detection cannot be overstated, particularly in applications like autonomous driving. In addition to their role in object detection, these techniques are pivotal in advancing text detection and recognition systems, which are vital components of autonomous vehicles and blind assistance systems [49]. In digital image processing and analysis, image

segmentation stands out as a fundamental and extensively used method. It facilitates the division of an image into multiple segments or regions, typically based on the distinctive characteristics of the pixels. This process offers several advantages, including the separation of foreground from background elements and the grouping of pixel regions with similarities in color or shape, ultimately enhancing the analysis and understanding of the image.

In the context of our research, image segmentation is a pivotal tool for achieving our text detection objectives. In order to enhance the accuracy of the regression method and introduce greater flexibility in the field of text object detection within scene images, we present a novel approach that involves harnessing the standard deviation matrix derived from the original image as the input to the FCN. In this stage, the proposed FCN incorporates the σ step into the tensors at various intermediate layers of the model.

Certain challenges could appear when using FCN, like the extensive variability in the sizes of word regions in scene images, where effectively detecting these regions requires a strategic adaptation. Recognizing the presence of larger words demands the extraction of features from the later stages of the neural network, while accurately delineating the geometry of smaller word regions relies on the availability of low-level information in the early stages. Consequently, our network must be capable of leveraging features at different levels to address these diverse requirements. To overcome this challenge, we embrace the concept of the U-shape architecture [50]. This approach facilitates the gradual fusion of feature maps while maintaining the compactness of the upsampling branches. This results in utilizing features at different levels while minimizing computational overhead effectively.

The following steps show the pipeline details of the proposed method:

- σ of intermediate channels in hidden layers:** A fully convolutional network (FCN) is a type of neural network architecture commonly used in computer vision tasks such as semantic segmentation. The FCN architecture takes an input image and outputs a pixel-wise prediction of the class label for each pixel. This architecture with skip connection follows the ordinary architecture of a convolution neural network as the downsampling step and up-convolution as an upsampling step. In the first phase, we apply the standard deviation principle to the original three-channel image, which reduces it to a tensor of one channel, to preserve high-resolution information and improve the accuracy of the final output. Then, in the second phase, we develop a deep learning model σNet , as shown in Fig. 4, for text detection that leverages both downsampling and upsampling operations and skip connections. We apply the σ method in the downsampling step of our deep learning model, which consists of a contracting path defined by a sequence of operations that increase the number of feature channels. We reduce the number of feature maps using the standard deviation values, resulting in a tensor of $n/3$ channels instead of n channels. The upsampling step consists of an expanding path defined by the transposed 2D convolution (up-convolution) of the feature map that reduces the number of feature channels to half. Applying the standard deviation step between expanding paths also reduces the n 's tensor channels to $n/3$. We introduce skip-connections between the downsampling and upsampling steps, which preserve and combine high-resolution information from previous layers. This approach enables our deep learning model to capture better the spatial relationships between different elements in the image, resulting in more accurate text detection.
- Downsampling step:** It consists of a contracting path defined by a sequence of operations of two 3×3 convolutions with the same padding, each accompanied by the activation function rectified linear unit (ReLU) and the application of σ method, which reduces the number of the feature map using the standard deviation, after that performing 2×2 max pooling with stride

2. At each step, the number of feature channels is increased by double, then decreased by a third using σ process until the fifth step, we add the attention module instead of 2×2 max pooling before the upsampling step.

- **Attention module:** In this step, we apply both the 2×2 max and average pooling to the output of σ map of the fifth step. We concatenate the result of max and average pooling then we use the dropout regularization to avoid overfitting and improve the generalization ability of the deep neural network.
- **Upsampling step:** It consists of an expanding path defined by the transposed 2D convolution (up-convolution) of the feature map followed by a skip connection and then by a 2×2 convolution that reduces the number of feature channels to half; after that, we add the σ process, which reduces the n 's channel tensor to a third.
- **Skip connection:** After each transposed convolution in the expanding path, the image is concatenated with the corresponding image result of max pooling from the contracting path. Skip connections preserve and combine high-resolution information from previous layers.

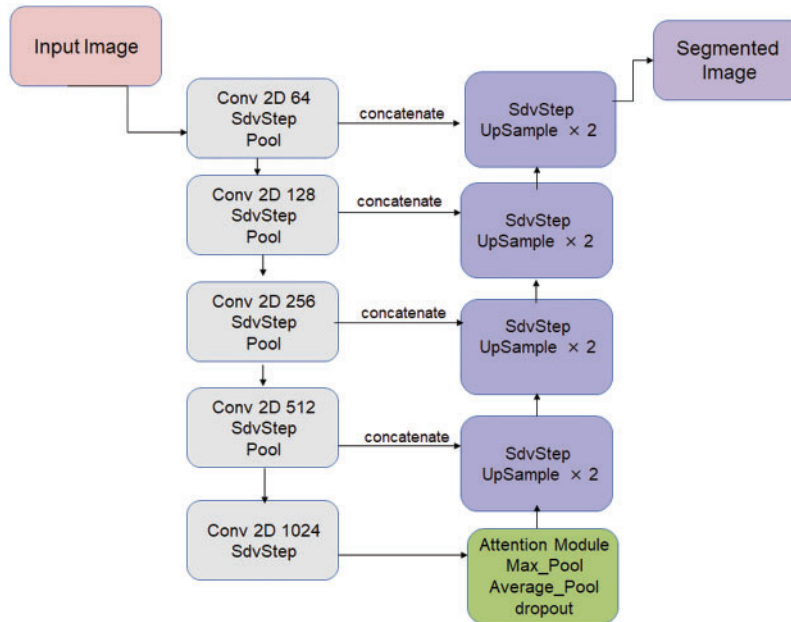









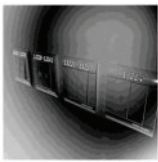












Figure 4: An overall overview of the proposed text detection approach

Unlike conventional deep-learning text detection models, the proposed method does not require feature extraction layer transfer learning. Therefore, the proposed method can be implemented with a relatively small memory and low computational complexity.

4 Experimental Evaluation and Results

This section gives the experimental setup and the three public datasets used to evaluate the proposed algorithm. Table 2 shows the result of the different steps of the pipeline, beginning with the original images through the standard deviation step and the output of the deep learning model. Compared with the ground truth, we remark that the result of the σ Net is near to the ground truth and gives us an enhanced result in extracting text from scene images.

Table 2: σ images detection compared with the ground and predicted images

Original image				
σ step				
Predicted result				
Ground truth image				
Final result image				

While the σ Net framework for text detection demonstrates significant advancements, it is important to acknowledge its limitations and areas for improvement. One notable weakness is its vulnerability to scenarios where text regions are closely positioned or intersecting.

4.1 Technical Details

We conduct 100 epochs of fine-tuning the models on real-world datasets. The training batch size is set to 8. The model is compiled with a custom loss function related to the Dice coefficient, which is commonly used in image segmentation tasks, and uses the Adam optimizer with a learning rate of $1e-4$ for training the model.

We harness computational complexity by leveraging Google Colab's GPU resources for our experiments. Specifically, we utilize the T4 GPU, which boasts 2,560 CUDA cores—parallel processing units designed for efficient parallel computation. This GPU comes equipped with 16 GB of GPU RAM, allowing us to work seamlessly with large datasets and complex models.

4.2 Datasets

- ICDAR 2015 dataset [51] includes 1000 training images and 500 testing images; Google Glasses captured this dataset with a resolution of 720×1280 and annotated the text instances at the word level.
- Total-Text dataset Total-Text [52] dataset is a collection of text data in various shapes, such as horizontal, multioriented, and curved. The dataset consists of 1255 training images and 300 testing images. Each text instance in the dataset is labeled at the word level, meaning individual words within the text instances are annotated and labeled for training and evaluation purposes.
- MSRA-TD500 dataset [53] contains text data in English and Chinese. It comprises a total of 300 training images and 200 testing images. The text instances within the dataset are annotated at the text-line level. Additionally, we augment the dataset with an extra 400 training images from the HUST-TR400 dataset [54].

4.3 Dice Coefficient Metric

In the analysis, we evaluate the proposed approach using the dice coefficient metric. The σNet text detection pipeline incorporates a standard deviation step. To evaluate its performance, we compare the performance of σNet with a similar semantic segmentation model. The evaluation results, shown in Table 3, indicate the dice coefficient values for both σNet and the baseline RGB model on three public datasets.

Table 3: Dice coefficient evaluation metric for semantic segmentation of MSRA-TD500 dataset

Model	Training set	Test set	Training size	Test size	Dice_coeff
rgb fcn	TD500TR400	TD500 test	700	200	0.85
σ fcn	σ TD500TR400	σ TD500 test	700	200	0.94
σNet	σ TD500TR400	σ TD500 test	700	200	0.97

The dice coefficient is a commonly used metric for evaluating the similarity between two sets. In the context of text detection, it measures the overlap between the predicted text regions and the ground truth regions. A higher dice coefficient indicates a better performance of the text detection model.

Table 3 presents the evaluation results of the MSRA-TD500 dataset and their corresponding variety in terms of the σ step in the dataset and model.

“rgb fcn” is the proposed fcn model without σ step.

“ σ fcn” is the proposed fcn model with σ step applied only in the input image.

“ σNet ” is the proposed model with σ step.

For each dataset, we provide the size of the training set and test set, denoted as “training size” and “test size”, respectively.

As a result, the standard deviation step in the σNet text detection pipeline significantly improves the dice coefficient values, thereby enhancing the model’s performance on various public datasets as shown in Tables 4 and 5.

This suggests that utilizing a one-channel input image with the standard deviation step effectively enhances the accuracy of the text detection model.

Table 4: Dice coefficient evaluation metric for semantic segmentation of ICDAR2015 dataset

Model	Training set	Test set	Training size	Test size	Dice_coeff
rgb fcn	ICDAR2015 train	ICDAR2015 test	1000	500	0.95
σ Net	σ ICDAR2015 train	σ TD500 test	1000	500	0.97

Table 5: Dice coefficient evaluation metric for semantic segmentation of total text dataset

Model	Training set	Test set	Training size	Test size	Dice_coeff
σ Net	σ TotalTextTrain	σ TotalTextTest	1254	300	0.97

4.4 Precision and Recall

Among the existing methods, σ Net is the only approach with no pre-trained model. This indicates that σ Net does not benefit from pre-existing knowledge and instead learns directly from the provided data. Tables 6 and 7 illustrate the precision and recall values of semantic segmentation with and without σ step.

Table 6: Precision and recall of text detection results of MSRA-TD500 dataset

Model	Training set	Test set	Precision	Recall
rgb fcn	TD500TR400	TD500 test	0.14	0.082
σ fcn	σ TD500TR400	σ TD500 test	0.85	0.77
σ Net	σ TD500TR400	σ TD500 test	0.88	0.79

Table 7: Precision and recall for semantic segmentation of ICDAR2015 dataset

Model	Training set	Test set	Precision	Recall
rgb fcn	ICDAR2015 train	ICDAR2015 test	0.75	0.32
σ Net	σ ICDAR2015 train	σ TD500 test	0.93	0.72

4.5 Comparison with Pre-Trained Models

We conduct comparative analysis between our proposed approach without pre-trained models and with the pretrained model used in [34,36,37] on the MSRA-TD500 dataset.

In this study, we employed a cloud-based NVIDIA T4 GPU for the training and testing of our proposed model, while other studies utilized performance-focused hardware configurations. It is important to note that due to the inherent differences in hardware architecture, design, and optimization, we faced challenges when attempting to directly compare the results of other studies. Therefore, we implemented the proposed approach by incorporating a pre-trained ResNet-50 model which has been trained on the ImageNet dataset, as the feature extraction channel. Table 8 presents the evaluation results on the MSRA-TD500 dataset, where “Resnet fcn” is the fcn model with pre-trained Resnet-50 on ImageNet used in the contracting path of the fcn (feature extraction).

Table 8: Comparison with pre-trained on MSRA-TD500 dataset

Model	Training set	Test set	Dice_coeff	Precision	Recall
Resnet fcn	TD500TR400	TD500 Test	0.92	0.88	0.30
σ Net	σ TD500TR400	σ TD500Test	0.97	0.88	0.79

The comparative analysis conducted on the MSRA-TD500 dataset aimed to showcase the efficacy of the proposed approach.

Table 8 presents the precision and recall results of the semantic segmentation phase. In contrast, Table 9 displays the precision and recall results of the final phase, which involves the prediction of bounding boxes for the text regions after segmentation.

Table 9: Precision and recall of text detection results of MSRA-TD500 dataset

Model	Precision	Recall
East model [33]	87.2	75.3
DB-ResNet [34]	91.5	79.2
DBNet [35]	91.5	83.3
DRRG [36]	88.05	82.30
σ Net	93.8	85.6

Table 9 delineates the performance improvements achieved by the σ Net algorithm in contrast to existing methodologies in text detection.

Analyzing the precision and recall metrics on the MSRA-TD500 dataset provides a comprehensive view of the algorithmic advancements.

The σ Net algorithm demonstrates substantial progress in both precision (93.8) and recall (85.6), outperforming contemporary research endeavors. This outcome underscores its heightened effectiveness in accurately identifying text instances compared to prior state-of-the-art models.

The notable improvement in precision indicates the algorithm's capability to limit false positives, while the enhanced recall signifies its proficiency in capturing a larger proportion of actual text instances. The significantly higher precision and recall values consolidate σ Net as a frontrunner, showcasing its potential for advancing text detection methodologies. These results substantiate the compelling performance of the σ Net algorithm, positioning it as a noteworthy advancement in text detection research, promising higher accuracy and reliability in practical applications.

4.6 A Speed Assessment Comparative

A comparative analysis between the proposed model and the East text detection model [33] is conducted, emphasizing speed assessment as the focal point of comparison. Table 10 presents the result of speed comparison analysis.

We deployed the East [33] model to ascertain its performance metrics. The East model achieved an FPS of 1.44 on our system. We observe that the proposed model achieved a significantly higher FPS of 2.37, showcasing markedly superior performance in terms of speed. This notable disparity in

processing speed underscores the efficiency gains inherent in our proposed model compared to the established East text detection model.

Table 10: Comparison of speed assessments with the east model

Model	Training set	Test set	FPS
East model [33]	TD500TR400	TD500 test	1.44
σ Net	σ TD500TR400	σ TD500 test	2.37

In scenarios where text strongly mimics the background, the effectiveness of the proposed text detection methodology may falter, introducing challenges in precisely isolating individual text components. This particular scenario highlights a critical aspect where the accuracy of the detection framework faces potential compromises. We observe that when text strongly resembles the background, the framework's efficiency may diminish, impacting detection accuracy. These weaknesses underline the need for further refinement in handling intricate text layouts, enhancing adaptability to varied backgrounds, and optimizing precision efficiency for broader applicability. These observations underscore the imperativeness of further refinement in addressing intricate text layouts. Enhancements should focus on augmenting the adaptability of the framework, particularly in scenarios with diverse and complex backgrounds.

5 Conclusion and Future Work

In conclusion, image text detection plays a crucial role in computer vision, but it remains a challenging task. This article presented a novel technique that enhances the accuracy of text detection by incorporating standard deviation values into the learning process. The experimental results clearly demonstrated the effectiveness of this technique in improving the performance of existing text detection methods, especially in challenging scenarios with low contrast or noisy backgrounds. The proposed technique holds great potential for improving text detection accuracy in various applications, including document analysis, scene text recognition, and optical character recognition (OCR). By leveraging standard deviation values, the technique can better handle image quality variations and enhance text detection algorithms' robustness.

The proposed technique opens up opportunities for its application in the context of IoT and low-end devices. Using standard deviation values can provide a lightweight and efficient solution for text detection on resource-constrained devices, making it suitable for deployment in scenarios where computational resources are limited. This is particularly relevant in IoT applications, where devices with limited processing power and memory are commonly used. Further future research can investigate the use of this technique in other computer vision tasks and explore its potential for real-world applications. However, it is vital to acknowledge the limitations of this approach. The proposed technique may encounter challenges when dealing with the closest text regions and multiple intersections of texts. Furthermore, in scenarios where text closely resembles the background, its efficiency may be compromised.

5.1 Future Research

The proposed approach not only holds promise but also opens doors to numerous opportunities for expansion. Its applicability in IoT and low-end devices stands as an intriguing prospect. Using standard deviation values can provide a lightweight and efficient solution for text detection, making

it particularly fitting for resource-constrained devices often found in IoT applications. These devices, with limited processing power and memory, could benefit greatly from the enhanced accuracy and efficiency this technique offers. The path ahead beckons further exploration, with future research avenues delving into its potential in other computer vision tasks and its applicability in a diverse array of real-world scenarios.

Acknowledgement: The authors thank all Artificial Intelligence and Its Applications (LIAP) Lab members from El Oued University, Algeria, and KFUPM University, Saudi Arabia. Their vision is to offer a suitable working environment. They have allowed us to use unlimited HPC service to reach our goal in this contribution.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: Literature review and problem identification: Sana Sahar GUIA and Abdelkader Laouid; Data collection: Mostefa Kara; Analysis and interpretation of results: Sana Sahar Guia and Abdelkader Laouid; Draft manuscript preparation: Mohammad Hammoudeh and Mostefa Kara. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used to support the findings of this review paper are derived from existing sources, which are appropriately cited throughout the article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. A. Haq, "Planetscope nanosatellites image classification using machine learning," *Comp. Syst. Sci. Eng.*, vol. 42, no. 3, 1031–1046, 2022. doi: [10.32604/csse.2022.023221](https://doi.org/10.32604/csse.2022.023221).
- [2] M. A. Haq, "CNN based automated weed detection system using UAV imagery," *Comp. Sys. Sci. Eng.*, vol. 42, no. 2, 837–849, 2022.
- [3] S. N. Estiri *et al.*, "A low-cost stochastic computing-based fuzzy filtering for image noise reduction," in *2022 IEEE 13th Int. Green Sustain. Comput. Conf. (IGSC)*, IEEE, 2022, pp. 1–6.
- [4] M. Hammoudeh and R. Newman, "Information extraction from sensor networks using the Watershed transform algorithm," *Inform. Fusion*, vol. 22, pp. 39–49, 2015. doi: [10.1016/j.inffus.2013.07.001](https://doi.org/10.1016/j.inffus.2013.07.001).
- [5] X. C. Yin, X. Yin, K. Huang, and H. W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, 2013.
- [6] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1733–1746, 2009. doi: [10.1109/TPAMI.2009.38](https://doi.org/10.1109/TPAMI.2009.38).
- [7] S. Karaoglu, R. Tao, T. Gevers, and A. W. M. Smeulders, "Words matter: Scene text for image classification and retrieval," *IEEE Trans. Multimed.*, vol. 19, no. 5, pp. 1063–1076, 2016.
- [8] K. Chait, A. Laouid, M. Kara, M. Hammoudeh, O. Aldabbas and A. T. Al-Essa, "An enhanced threshold RSA-Based aggregate signature scheme to reduce blockchain size," *IEEE Access*, vol. 11, pp. 110490–110501, 2023. doi: [10.1109/ACCESS.2023.3322196](https://doi.org/10.1109/ACCESS.2023.3322196).
- [9] M. Kara *et al.*, "One digit checksum for data integrity verification of cloud-executed homomorphic encryption operations," *Cryptol. ePrint Arch.*, vol. 2023, 2023.
- [10] A. Theus, L. Rossetto, and A. Bernstein, "HyText—A scene-text extraction method for video retrieval," in *Int. Conf. Multimed. Model.*, Cham, Springer International Publishing, 2022, pp. 182–193.
- [11] F. Naiemi, V. Ghods, and H. Khalesi, "Scene text detection and recognition: A survey," *Multimed. Tools Appl.*, vol. 81, no. 14, pp. 20255–20290, 2022. doi: [10.1007/s11042-022-12693-7](https://doi.org/10.1007/s11042-022-12693-7).

- [12] L. Yang *et al.*, “A review of natural scene text detection methods,” *Procedia Comput. Sci.*, vol. 199, pp. 1458–1465, 2022. doi: [10.1016/j.procs.2022.01.185](https://doi.org/10.1016/j.procs.2022.01.185).
- [13] S. Long *et al.*, “Towards end-to-end unified scene text detection and layout analysis,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1049–1059.
- [14] Q. Ye and D. Doermann, “Text detection and recognition in imagery: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, 2014.
- [15] S. Sayahi and M. B. Halima, “An intelligent and robust multi-oriented image scene text detection,” in *2014 6th Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, IEEE, 2014, pp. 418–422.
- [16] S. Uchida, “25-text localization and recognition in images and video,” in *Handbook of Document Image Processing and Recognition*, London, Springer, 2014, pp. 843–883.
- [17] H. Yu, Y. Huang, L. Pi, C. Zhang, X. Li and L. Wang, “End-to-end video text detection with online tracking,” *Pattern Recognit.*, vol. 113, pp. 107791, 2021. doi: [10.1016/j.patcog.2020.107791](https://doi.org/10.1016/j.patcog.2020.107791).
- [18] S. Surana *et al.*, “Text extraction and detection from images using machine learning techniques: A research review,” in *2022 Int. Conf. Electron. Renew. Syst. (ICEARS)*, IEEE, 2022, pp. 1201–1207.
- [19] S. D. Pande *et al.*, “Digitization of handwritten Devanagari text using CNN transfer learning—A better customer service support,” *Neurosci. Inform.*, vol. 2, no. 3, pp. 100016, 2022. doi: [10.1016/j.neuri.2021.100016](https://doi.org/10.1016/j.neuri.2021.100016).
- [20] K. Boukthir *et al.*, “Reduced annotation based on deep active learning for arabic text detection in natural scene images,” *Pattern Recogn. Lett.*, vol. 157, pp. 42–48, 2022. doi: [10.1016/j.patrec.2022.03.016](https://doi.org/10.1016/j.patrec.2022.03.016).
- [21] P. M. Manwatkar and S. H. Yadav, “Text recognition from images,” in *2015 Int. Conf. Innov. Inf., Embed. Commun. Syst. (ICIIECS)*, 2015, pp. 1–6.
- [22] P. Alipour and S. E. Charandabi, “Analyzing the interaction between tweet sentiments and price volatility of cryptocurrencies,” *Eur. J. Bus. Manage. Res.*, vol. 8, no. 2, pp. 211–215, 2023. doi: [10.24018/ejbm.2023.8.2.1865](https://doi.org/10.24018/ejbm.2023.8.2.1865).
- [23] S. Long, X. He, and C. Yao, “Scene text detection and recognition: The deep learning era,” *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, 2021. doi: [10.1007/s11263-020-01369-0](https://doi.org/10.1007/s11263-020-01369-0).
- [24] W. Huang, Y. Qiao, and X. Tang, “Robust scene text detection with convolution neural network induced mser trees,” in *Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 497–511.
- [25] 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. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4651–4659.
- [26] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou and Z. Cao, “Scene text detection via holistic, multi-channel prediction,” arXiv preprint arXiv:1606.09002, 2016.
- [27] L. Sun, Q. Huo, W. Jia, and K. Chen, “A robust approach for text detection from natural scene images,” *Elsevier Pattern Recognit.*, vol. 48, no. 9, pp. 2906–2920, 2015. doi: [10.1016/j.patcog.2015.04.002](https://doi.org/10.1016/j.patcog.2015.04.002).
- [28] Z. Ebin, T. Martin, and B. Bénédicte, “Image processing based scene-text detection and recognition with tesseract,” arXiv preprint arXiv:2004.08079, 2020.
- [29] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang and Q. Dai, “Effective Uyghur language text detection in complex background images for traffic prompt identification,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 220–229, 2017.
- [30] S. M. Hanif and L. Prevost, “Text detection and localization in complex scene images using constrained adaboost algorithm,” in *2009 10th Int. Conf. Doc. Anal. Recognit.*, IEEE, 2009, pp. 1–5.
- [31] A. Gonzalez, L. M. Bergasa, and J. J. Yebes, “Text detection and recognition on traffic panels from street-level imagery using visual appearance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 228–238, 2013.
- [32] S. Y. Arafat and M. J. Iqbal, “Urdu-text detection and recognition in natural scene images using deep learning,” *IEEE Access*, vol. 8, pp. 96787–96803, 2020. doi: [10.1109/Access.6287639](https://doi.org/10.1109/Access.6287639).
- [33] X. Zhou *et al.*, “East: An efficient and accurate scene text detector,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5551–5560.
- [34] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, “Real-time scene text detection with differentiable binarization,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 11474–11481, 2020.

- [35] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, 2022.
- [36] S. X. Zhang, X. Zhu, J. B. Hou, C. Liu, C. Yang and H. Wang, "Deep relational reasoning graph network for arbitrary shape text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9699–9708.
- [37] Y. Zhu, J. Chen, L. Liang, Z. Kuang, L. Jin and W. Zhang, "Fourier contour embedding for arbitrary-shaped text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3123–3131.
- [38] Z. Huang, Z. Zhong, L. Sun, and Q. Huo, "Mask R-CNN with pyramid attention network for scene text detection," in *2019 IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2019, pp. 764–772. doi: [10.1109/WACV.2019.00086](https://doi.org/10.1109/WACV.2019.00086).
- [39] W. Wang *et al.*, "Efficient and accurate arbitraryshaped text detection with pixel aggregation network," in *Proc. ICCV*, 2019, pp. 8439–8448.
- [40] W. Wang, E. Xie, X. Li, W. Hou, T. Lu and G. Yu, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9336–9345.
- [41] S. S. Guia, A. Laouid, R. Euler, M. A. Yagoub, A. Bounceur and M. Hammoudeh, "A salient object detection technique based on color divergence," in *4th Int. Conf. Future Netw. Distrib. Syst. (ICFNDS)*, 2020, pp. 1–6.
- [42] X. Chen *et al.*, "Adaptive embedding gate for attention-based scene text recognition," *Neurocomput.*, vol. 381, pp. 261–271, 2020. doi: [10.1016/j.neucom.2019.11.049](https://doi.org/10.1016/j.neucom.2019.11.049).
- [43] J. Zhang *et al.*, "CCTSDB 2021: A more comprehensive traffic sign detection benchmark," *Hum.-Centric Comput. Inf. Sci.*, vol. 12, 2022.
- [44] W. Wang and K. Siau, "Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda," *J. Database Manage. (JDM)*, vol. 30, no. 1, pp. 61–79, 2019. doi: [10.4018/JDM](https://doi.org/10.4018/JDM).
- [45] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, 2005. doi: [10.1109/TNN.2005.845141](https://doi.org/10.1109/TNN.2005.845141).
- [46] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
- [47] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020. doi: [10.1109/Access.6287639](https://doi.org/10.1109/Access.6287639).
- [48] I. K. Artúr, F. Róbert, and G. Péter, "Unsupervised clustering for deep learning: A tutorial survey," *Acta Polytech. Hung.*, vol. 15, no. 8, pp. 29–53, 2018. doi: [10.12700/APH.15.8.2018.8.2](https://doi.org/10.12700/APH.15.8.2018.8.2).
- [49] F. Naiemi, V. Ghods, and H. Khalesi, "A novel pipeline framework for multi oriented scene text image detection and recognition," *Expert. Syst. Appl.*, vol. 170, pp. 114549, 2021. doi: [10.1016/j.eswa.2020.114549](https://doi.org/10.1016/j.eswa.2020.114549).
- [50] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, Springer, 2015, pp. 234–241.
- [51] D. Karatzas *et al.*, "ICDAR, 2015 competition on robust reading," in *2015 13th Int. Conf. Doc. Anal. Recognit. (ICDAR)*, IEEE, 2015, pp. 1156–1160.
- [52] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *2017 14th IAPR Int. Conf. Doc. Anal. Recognit. (ICDAR)*, IEEE, vol. 1, 2017, pp. 935–942.
- [53] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *2012 IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, 2012, pp. 1083–1090.
- [54] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, 2014. doi: [10.1109/TIP.83](https://doi.org/10.1109/TIP.83).