STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, May 2024, pp 685–696.
Published online in International Academic Press (www.IAPress.org)

License plate text recognition using deep learning, NLP, and image processing techniques

Hanae Moussaoui [1,*], Nabil El Akkad [1], Mohamed Benslimane [2]

[1] *Engineering Systems and Applications Laboratory (LISA), National School of Applied Sciences*
[2] *Laboratory of Industrial Techniques (LTI), EST of Fez, Sidi Mohamed Ben Abdellah University, Fez, Morocco*

Abstract Detecting license plates has never been easy, particularly with the proliferation of sophisticated radars on highways and roads. By 2021, the gendarmerie and National Security Road control agents will have access to more than 1 billion smart traffic radars worldwide. This research presents a revolutionary technique for detecting and recognizing Arabic and Latin license plates. After assembling the gathered images to create a novel dataset, we utilized YOLO v7 to locate and identify the number plate in the image as the first step of the suggested procedure. Before the dataset was fed to the detection system, it was manually labeled. Afterward, we improved the recognized license plate using machine learning methods. To do this, we used kernel methods as well as thresholding to get rid of the extra vertical lines on the plate. After that, we employed Arabic OCR along with Easy OCR methods to decipher the Latin and Arabic characters on the number plate. Eventually, the proposed method achieved an F1 score of 98%,with a precision and recall of 97% and 98%, respectively. We also obtained an accuracy of 99% for image segmentation. The segmentation and detection results from the suggested strategy have shown satisfactory results.

Keywords Easy OCR, License plate detection, thresholding, Yolo, Arabic OCR, kernel

DOI: 10.19139/soic-2310-5070-1966

1. Introduction

The problem of object detection [1, 2, 3] in computer vision [4] is locating and identifying things in an image or video. It is a crucial part of numerous applications, such as augmented reality, security monitoring [5, 6], driverless cars, and image retrieval. Identifying what items are present in a scene and pinpointing where exactly they are inside the image [7] are the two objectives of object detection. Object detection [8] is a field that is constantly developing thanks to research and advancements. The precision and speed of detection systems are continually being improved by newer methods and architectures, which increases their value in a variety of real-world scenarios. On the other hand, a computer vision task called image segmentation [9, 10, 11] includes dividing an image into some areas or segments [12, 13], each of which corresponds to a significant object or portion of an item in the image. By dividing an image into semantically significant parts, image segmentation enables the removal of objects or characteristics of interest from the image. Numerous applications, such as medical image analysis, object recognition, and scene comprehension, use image segmentation [14, 15] extensively. Moreover, finding and detecting license plates inside images or video frames is a particular challenge in computer vision known as license plate detection. Applications for this technology include toll collecting, parking management, traffic surveillance, and law enforcement. The more important elements that are involved in license plate detection are the object detection process by finding probable areas of interest (ROI) in an image is usually the first step in the detection of license plates. YOLO, SSD,

*Correspondence to: Hanae Moussaoui (Email: hanae.moussaoui1@usmba.ac.ma). Engineering Systems and Applications Laboratory (LISA), National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Faster R-CNN, and other well-known object identification methods can be utilized to locate these areas. Afterward, preprocessing techniques [16] can be used to improve the image's quality, account for differences in lighting and perspective, and increase the likelihood that license plates will be visible. On the other hand, plates frequently contain distinctive features, such as a fixed aspect ratio, a rectangular shape, and particular character patterns. To exclude objects without a license plate, one can make use of these qualities. Following the identification of the candidate regions [17], the text characters on the license plate are frequently located using this technique. EAST is one of many text detection techniques that can be applied in this situation. Thereafter, specific approaches are used to identify and extract the characters from the license plate when the text regions have been located. Character recognition uses OCR [18, 19] engines like Tesseract or custom software. Finally, post-processing processes [20] could include fixing recognition mistakes, enhancing the accuracy of the outcomes, checking the identified characters against a preset license plate format (e.g., a specific number of characters, allowable characters), and so on. However, different license plate designs, poor illumination, unfavorable angles, and the presence of obstructions can all affect how accurately a system detects license plates. Thanks to the introduction of deep learning methods and the accessibility of big, labeled datasets, license plate identification technology has made considerable strides in recent years. These developments have produced stronger and more precise systems for reading and processing license plates in actual-world situations. The main purpose of this paper is the creation of a new license plate detection and recognition method, for both Arabic and Latin letters. The method was based on Yolo v7 as well as image processing techniques. The rest of the paper will be organized as follows; In the second section, we'll discuss the related works by giving all the used methods and some already proposed techniques in the literature. Afterward, the proposed method will be explained in the third section. Subsequently, experiments and results will be given in the fourth section. Eventually, we'll conclude our work with a conclusion section and future work.

2. Related works

In many applications, such as automatic number plate recognition (ANPR) [21], vehicle monitoring, and traffic control systems, license plate detection is essential. In this field, numerous initiatives and research publications have been created. Several image processing and computer vision techniques [22, 23], including, contour analysis, edge detection, and template matching, are frequently used in these procedures. The Hough Transform is a widely used tool for finding straight lines, which can be used to determine the limits of license plates. Moreover, the detection of license plates has been revolutionized by Convolutional Neural Networks (CNNs) [24]. Likewise, to recognize and localize license plates in real time, researchers have created various CNN designs, including Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot Multi-Box Detector). Furthermore, Region Proposal Networks [25] are used by techniques like Faster R-CNN and YOLOv2 to find possible regions of interest, especially license plate regions. As a result, the network needs to evaluate fewer regions to detect license plates. Additionally, for license plate detection tasks, pre-trained models like VGG [26], ResNet [27], or MobileNet [28] are frequently used. Working with a small amount of labeled data can enhance accuracy while saving time. Here are some related works for the number plate detection issue: The authors of [29] suggested a method for reading license plates based on edge information. To do this, the original image was first turned into a grayscale image, and then a dilation was used to smudge any sharp edges. Then, morphological processes of erosion and dilation were used to identify both vertical and horizontal edges. The authors achieved an accuracy of 95.5%. In [30], a convolutional neural network was deployed twice through the same first CNN to identify cars in an image, while the second CNN was available to identify characters. For two separate datasets, this technique had identification success rates of 90.94% and 96.09%. In [31], Maskr-cnn was utilized for the recognition, detection, and segmentation of license plates. This method's performance was evaluated using four datasets, and it demonstrated 99.3% accuracy. The authors of [32] proposed a multitask convolutional neural network algorithm for license plate detection. The data from the detected license plate was then recognized using an end-to-end algorithm. The authors attained a recognition precision of 98% after comparing the data with cutting-edge techniques. Although license plate information is sensitive, there is an increasing focus on protecting individual privacy and following data protection laws while creating and implementing license plate detection systems. Furthermore, it's crucial to remember that new methods and datasets are always being developed in the field of license plate detection. To accomplish precise and effective license plate detection for certain use cases, researchers and developers frequently combine several techniques.

3. Dataset

To run our proposed method, we first had to create a new dataset of vehicles. Therefore, we used one of the free Moroccan marketplace websites called “Avito” to download images, as well as other free websites from the internet. The created dataset is composed of almost 300 images. afterward, we employed the label studio tool to label the images. the result can be shown in Figure 2. Thereafter, and to train Yolo v7, we divided the dataset into three main categories, validation, training, and testing as shown in Figure 3.

Figure 1. The created dataset

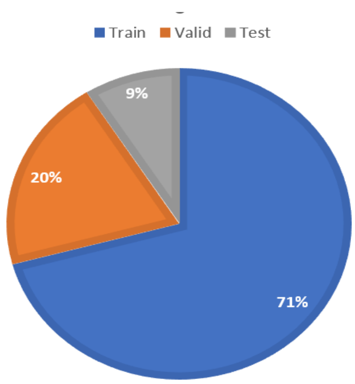

Figure 2. Dataset division

4. The proposed method

After gathering images from the internet, and creating a new dataset of different vehicles, we used first of all Yolo v7 [33] technique to detect and extract the license plate area from the input image. Afterward, we resized the license plate before applying the binary thresholding algorithm to separate the foreground representing the characters from the background in our case. Moreover, to enhance the image we used morphological operations to remove unwanted lines to facilitate the process of extracting the characters later. Subsequently, Optical character recognition was employed to obtain the characters in the number plate. Here, we used both Arabic OCR and Easy OCR depending on the language of the license number. The flowchart below, represented in Figure 1 shows in detail the proposed method:

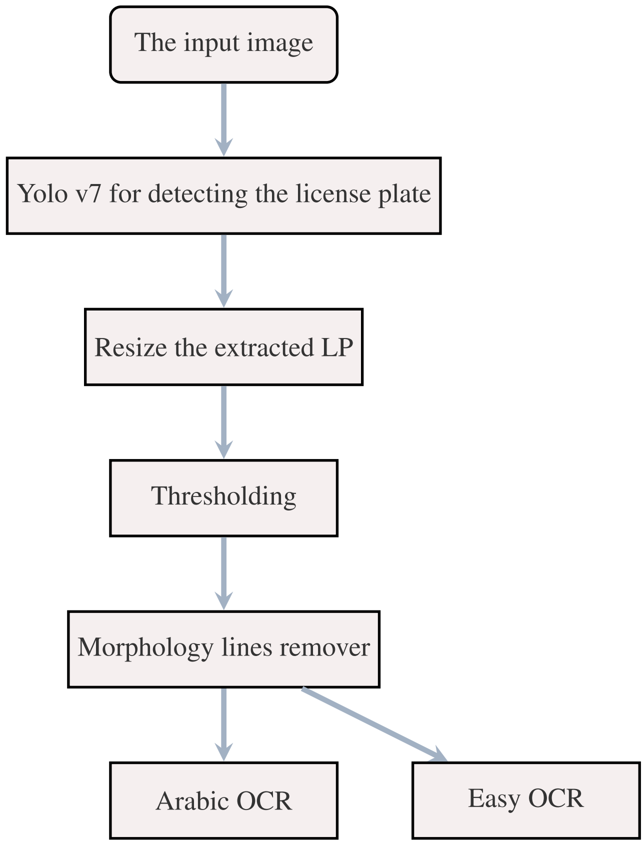

Figure 3. The proposed architecture

4.1. Yolo v7

”You Only Look Once,” or ”YOLO” [33, 34], is a well-known and powerful real-time object recognition technique that has had a big impact on deep learning and computer vision. Real-time object detection and localization in image or video frames is made possible by YOLO. YOLO’s main qualities are as follows:
Real-Time Object Detection: Yolo is renowned for its quickness and effectiveness, which enable it to detect objects in real-time. To do this, the input image is divided into a grid, and each grid cell’s object bounding boxes and class probabilities are predicted.
Single Forward Pass: YOLO does object detection in a single forward pass, which makes it faster than certain other object detection techniques that call for numerous neural network passes.

Bounding box and class prediction: YOLO predicts bounding boxes around objects that are recognized and gives each object in the image a class label. It is therefore appropriate for jobs involving both object detection and object classification.
Anchors: To assist with object shape and placement prediction, YOLO makes use of anchor boxes. These anchor boxes help to increase localization accuracy because they are specified based on the dataset being used.
Numerous Versions [35]: Yolo has seen multiple iterations, each improving on the one before it. Among the well-known versions are YOLOv2 (YOLO9000), YOLOv3, and YOLOv4, until YOLOv8. Each version offers enhancements in terms of accuracy and performance.
Open-Source Implementation: YOLO is available to the research and development community due to its widespread adoption and implementation in many deep learning frameworks and programming languages.
Numerous fields, including robotics, security and surveillance, autonomous cars, and more, have used YOLO. It is frequently applied to applications like generic object recognition, vehicle detection, and pedestrian detection. Nonetheless, YOLO can be applied to any object detection task and is not restricted to these specific uses. The YOLO architecture can be presented in Figure 4 as follows:

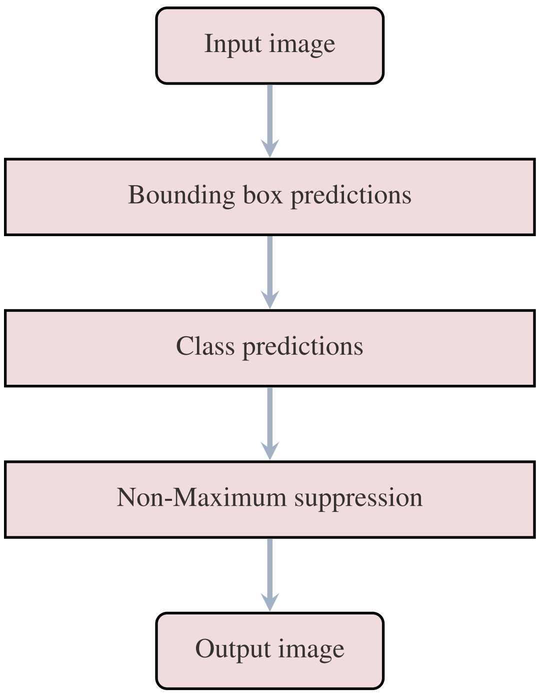

Figure 4. Yolo architecture

It's crucial to remember that computer vision is a dynamic discipline where improvements can happen quickly. Keeping up with the most recent research papers, conference proceedings, and updates from the computer vision community will definitely give a better idea of where YOLO and similar technologies are headed in the future.

Figure 5 below shows the obtained result after applying YOLO v7 on our custom dataset.

4.2. Thresholding segmentation

The thresholding [36] technique is a fundamental image-processing method that is used to divide or isolate objects of interest in a digital image from its background. Setting a threshold value typically a pixel intensity value and then categorizing each pixel in the image according to whether its intensity is above or below that threshold are the

Figure 5. Yolo v7 on our custom dataset

steps involved in the process. As a result, pixels are classified as either foreground (item of interest) or background, creating a binary image. Thresholding works by following the main steps below:
Step 1: Choosing a threshold value, which divides pixel intensities into two groups, usually involves using domain knowledge or studying the image's histogram. This threshold can be established manually or automatically using a variety of techniques, including adaptive thresholding and Otsu's method.
Step 2: Binary Image Creation, by taking the intensity value of each pixel in the original image and comparing it to the threshold. The pixel is given a value of 1 (often white) and is regarded as being in the foreground if the intensity is higher than the threshold. The intensity of a pixel is given a value of 0 (often black) and is regarded as being in the background if it is less than or equal to the threshold.
Step 3: As a result, the process of thresholding produces a binary image, sometimes referred to as a mask, in which the items of interest are emphasized against a black backdrop, typically in white.
Various thresholding methods and algorithms can be applied to tackle certain problems, like changes in illumination, noise levels, and object complexity. Global thresholding, adaptive thresholding, and approaches like Otsu's thresholding, which can automatically choose an ideal threshold value depending on the image's histogram, are a few common thresholding techniques. A basic and adaptable method in image processing, thresholding serves as the foundation for more complex image analysis tasks like object recognition and machine vision.
The mathematical formula of thresholding [37] is presented below:

$$O(x,y) = \begin{cases} 1, & if\ i(x,y) > t \\ 0, & if\ i(x,y) \leq t \end{cases} \tag{1}$$

Where:
O(x, y) is the output image
I(x, y) represents the intensity value of a pixel at (x, y) in the image.
t is the threshold value.

4.3. Morphological operations

Image processing methods called morphological operations [38] are used to process and modify an image's object's shape and structure. Tasks like feature extraction, noise reduction, and image segmentation benefit greatly from these techniques. Binary images or grayscale are typically used for morphological processes. Additionally, dilation and erosion are the two most used morphological processes, and they can be combined or used separately to accomplish a variety of image-processing objectives.
Dilation: is a morphological process in binary images that makes the white areas or objects larger [39]. A structuring element (a small matrix or kernel) is scanned over the image, and all the pixels it covers are set to white. At each position where the structuring element overlaps with a white pixel in the image, the center of the structuring element is positioned over that pixel. The mathematical formula of dilation for a particular image p using a structuring element s can be written as follows:

$$p \oplus s = \{z|(\widehat{s_z}) \cap p \neq \phi\} \tag{2}$$

Erosion: The reverse of dilation is erosion [40]. In a binary image, it causes the white areas to contract or degrade. Erosion employs a structural element similar to dilation but only converts the target pixel to white if all the pixels beneath it are white, as opposed to setting it to white if there is overlap. The target pixel is set to black if there is a black pixel beneath the structuring element. The mathematical formula of erosion for a particular image p using a structuring element s can be written as follows:

$$p \ominus s = \{z|(s)_z \subseteq i\} \tag{3}$$

Opening: Erosion and dilatation are combined to perform an opening operation. It helps eliminate little things and noise from an image while keeping the general size and shape of larger ones. The mathematical formula of opening [41] for a particular image p using a structuring element s can be written as follows:

$$p \circ s = (p \ominus s) \oplus s \tag{4}$$

Closing: The opposite of an opening operation is a closing operation [42]. It starts with dilatation and ends with erosion. It works well for joining adjacent pieces and sealing tiny gaps in objects. The mathematical formula of closing for a particular image p using a structuring element s can be written as follows:

$$p \bullet s = (p \oplus s) \ominus s \tag{5}$$

Morphological Gradient: This process distinguishes an image between erosion and dilation. It draws attention to the image's object edges.
Top hat and black hat: The processes known as Top Hat and Black [43] Hat entail deducting the outcome of an opening or closing operation from the initial image. Although the black-hat operation accentuates the darker portions, the top-hat operation places more emphasis on the lighter regions.
Hit-or-Miss Transform: In binary images, this morphological procedure is used to identify particular patterns or shapes. Finding matches between the two defined structuring elements one for the pattern and one for its complement in the image is what this technique entails.
Image segmentation, text extraction, object recognition, and other uses of morphological processes are common in computer vision and image processing. They can aid in enhancing features for later analysis, removing noise from photos, and improving image quality. The particular image processing task at hand as well as the image's properties determine the operation to be used as well as the structuring element's size and form.

4.4. OCR

With the use of optical character recognition (OCR) technology [44], many document formats such as scanned paper documents, PDF files, or digital camera images can be transformed into editable and searchable data. It is possible to extract, alter, and search for text within these digital documents thanks to OCR software and algorithms that analyze the text inside them and convert it into machine-readable text. OCR technique [45] works following the main steps below:
Step 1: Image Preprocessing; Since images are typically used as the input document for OCR, this step comes first. This comprises operations like image enhancement, noise reduction, and binarization (turning the image black and white). These procedures aid in raising the input's quality for character recognition.
Step 2: Text Localization; OCR systems frequently have to pinpoint the areas of an image that hold text. Finding the locations where text is present is the process of text localization.
Step 3: Text Segmentation; OCR systems must separate text into individual characters or words when there are several text lines or paragraphs. Segmenting text entails separating it into its component components.

5. Experiments and evaluation

The model was trained and tested using Python code on Collaboratory. For this, we installed the Ultralytics library. Furthermore, to evaluate the obtained results of the proposed method, we used several evaluation metrics that are cited below. Furthermore, the main training parameters are shown in Table 1 below:

Model	Parameters (M)	Epochs	Batch	Device	mAP@0.5(%)	mAP@0.5:0.95(%)	Fitness
Yolo v7-tiny	6.2M	300	64	GPU	95.5%	87.99%	89.1%

Table.1 Experimental results and training parameters for Yolo v7 on our dataset

5.1. Precision

In machine learning, statistics, and other domains, the term ”precision” [46] is frequently used to quantify the degree of accuracy in a classification system or predictive model. It is a gauge of how closely actual positive occurrences match the system's positive predictions. Precision is determined in the setting of binary classification, or a straightforward ”yes/no” prediction, using the formula below:

$$Precision = TruePositives(TP)/FalsePositives(FP) + TruePositives(TP) \tag{6}$$

Where the true positives are the occurrences that were accurately anticipated to be positive. Moreover, false positives are the instances that have been incorrectly identified as positive. The obtained results are shown in the Table 2 below:

Epochs	30	55	80	85	90	110	115	135	195	299
Precision	0.0027	0.1725	0.4873	0.8110	0.8245	0.9034	0.9130	0.9787	0.9851	0.9899

Table.2 The obtained precision/epoch

5.2. Recall

In statistics and machine learning, recall, also referred to as sensitivity or true positive rate, is a parameter used to assess how well a classification model performs, especially in binary classification scenarios. Recall quantifies how well the model can identify every pertinent instance in the dataset. The recall [47] can be defined as follows:

$$Precision = True\ Positives(TP)/False\ Negatives(FN) + True\ Positives(TP) \tag{7}$$

The obtained results are shown in the Table 3 below:

Epochs	30	55	80	85	90	110	115	135	195	299
Recall	0.0235	0.4597	0.7911	0.7954	0.8245	0.8261	0.8435	0.8775	0.9517	0.9998

Table.3 The obtained recall/epoch

5.3. F1-score

In statistics and computer learning, the F1-score [48] is a frequently used metric, particularly for binary classification problems. It is a metric that provides a fair evaluation of a model's performance by combining recall and precision into a single number. The following formula is used to determine the F1-score:

$$F1 - score = (2 * Recall * Precision)/(Recall + Precision) \tag{8}$$

The obtained results are shown in the Table 4 below:

Epochs	30	55	80	85	90	110	115	135	195	299
F1-score	0.0048	0.2509	0.6031	0.8031	0.8245	0.8630	0.8769	0.9253	0.9681	0.9948

Table.4 The obtained F1-score/epoch

5.4. Accuracy

Accuracy is a frequently used indicator that assesses how accurate a predictive model is overall [49]. It measures the percentage of all incidents in the dataset that were accurately predicted. The following is the accuracy formula:

$$Accuracy = (Number\ of\ Predictions)/(Total\ number\ of\ predictions) \tag{9}$$

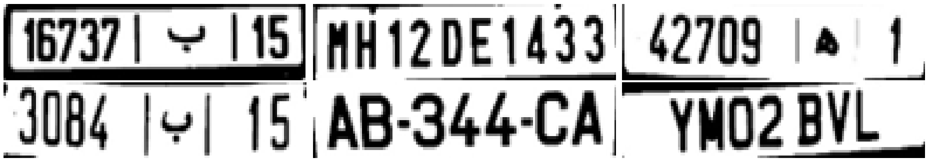

Figure 6. The obtained thresholding results

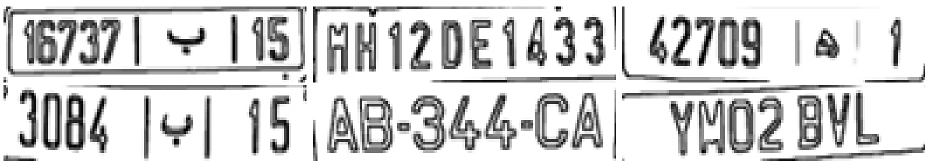

Figure 7. The obtained morphological operations results

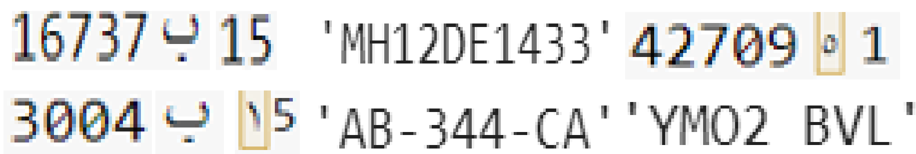

Figure 8. The obtained OCR results

In Figure 6, Figure 7, and Figure 8, we displayed the obtained results after applying thresholding and morphological operations. As well as the OCR technology.

Images	The Obtained Accuracy
Image 1	99%
Image 2	97%
Image 3	99%
Image 4	98%
Image 5	87%
Image 6	80%

Table.5 The obtained accuracy

In Table 5 above, we display the obtained accuracy for each character recognition using the proposed hybridization, as we can see, we achieved 99% in recognition. By applying the thresholding as well as the morphological operations we could enhance the quality of the first detected license plate. Therefore, we conclude that the proposed method gives satisfactory results in terms of precision, accuracy, F1 score, and recall. Accordingly, we achieved 99% in accuracy, 98% in F1-score, we obtained 98% in the recall, and a precision of 97%. On the other hand, a comparison with some already existing methods is presented in Table 6.

Methods	The proposed method	[50]	[51]	[52]
Detection (F1-score)	99%	98.33%	65.5%	97.9%
Character recognition (accuracy)	97%	90.37%	87%	97.5%

Table.6 Comparison of the proposed technique and other methods

In the paper [50], the authors proposed a method for license plate detection and recognition. For this, they used Fast Yolo and Yolov2 to detect cars and license plates. Moreover, for character segmentation, they employed convolutional neural networks. In the paper [51], the authors suggested a technique as well for license plate detection and recognition. Consequently, they first went through a preprocessing step using the Median filter and the Histogram equalizer to enhance the quality of the input image. Afterward, to detect the license plate, they employed the Sobel Edge detection technique, followed by thresholding for character segmentation. Eventually, they used the BPNN architecture for character recognition. In the paper [52], the authors used Mask-RCNN to detect the license plate, as well as for segmentation and recognition.

6. Conclusion

For a very long time, extracting and recognizing license plates has been an extremely difficult task. In the field, numerous strategies have been put out, and each one has advantages and disadvantages. In this paper, we discussed a new technique for extracting and identifying license plates that combine Yolo v7 which is a deep neural network-based image processing algorithm, thresholding, as well as morphological operations, and OCR. The experiments demonstrated very excellent results in both recognition and detection. Future development will hopefully include the addition of a text categorization component that will assign the proper country to each retrieved license plate.

REFERENCES

1. MOUSSAOUI, Hanae, EL AKKAD, Nabil, et BENSLIMANE, Mohamed, *A Brain Tumor Segmentation and Detection Technique Based on Birch and Marker Watershed*, SN Computer Science, 2023, vol. 4, no 4, p. 339.
2. Solanki, P, Israni, D, & Shah, A, *An Efficient Satellite Image Super Resolution Technique for Shift-Variant Images using Improved New Edge Directed Interpolation*, Statistics, Optimization & Information Computing, 6(4), 619-632. https://doi.org/10.19139/soic.v6i4.433, 2018.
3. KATTENBORN, Teja, LEITLOFF, Jens, SCHIEFER, Felix, et al, *Review on Convolutional Neural Networks (CNN) in vegetation remote sensing*, ISPRS journal of photogrammetry and remote sensing, 2021, vol. 173, p. 24-49.
4. XU, Yongchao, FU, Mingtao, WANG, Qimeng, et al, *Gliding vertex on the horizontal bounding box for multi-oriented object detection*, IEEE transactions on pattern analysis and machine intelligence, 2020, vol. 43, no 4, p. 1452-1459.
5. Filali, Y., Abdelouahed, S., & Aarab, A, *An Improved Segmentation Approach for Skin Lesion Classification*, IStatistics, Optimization & Information Computing, 7(2), 456-467. https://doi.org/10.19139/soic.v7i2.533, 2019.
6. LONG, Xiang, DENG, Kaipeng, WANG, Guanzhong, et al. PP-YOLO, *An effective and efficient implementation of object detector*, arXiv preprint arXiv:2007.12099, 2020.
7. MOUSSAOUI, Hanae, EL AKKAD, Nabil, et BENSLIMANE, Mohamed, *A hybrid skin lesions segmentation approach based on image processing methods*, Statistics, Optimization & Information Computing, 2023, vol. 11, no 1, p. 95-105.
8. MOUSSAOUI, Hanae, BENSLIMANE, Mohamed, et al, *Reinforcement Learning: A review*, International Journal of Computing and Digital Systems, 2023, vol. 13, no 1, p. 1-1.
9. MOUSSAOUI, Hanae, BENSLIMANE, Mohamed, et EL AKKAD, Nabil, *A novel brain tumor detection approach based on fuzzy C-means and Marker Watershed algorithm*, Digital Technologies and Applications: Proceedings of ICDTA 21, Fez, Morocco. Cham : Springer International Publishing, 2021. p. 871-879.
10. Khrissi, L., El Akkad, N., Satori, H., & Satori, K, *A Performant Clustering Approach Based on An Improved Sine Cosine Algorithm*, International Journal of Computing, 159–168. https://doi.org/10.47839/ijc.21.2.2584,2022.
11. MOUSSAOUI, Hanae, BENSLIMANE, Mohamed, et EL AKKAD, Nabil, *Image segmentation approach based on hybridization between K-means and mask R-CNN*, WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems. Springer Singapore, 2022. p. 821-830.
12. MOUSSAOUI, Hanae, EL AKKAD, Nabil, et BENSLIMANE, Mohamed, *Moroccan Carpets Classification Based on SVM Classifier and ORB Features*, Digital Technologies and Applications: Proceedings of ICDTA'22, Fez, Morocco, Volume 2. Cham : Springer International Publishing, 2022. p. 446-455.
13. HUANG, Yuhao, YANG, Xin, LIU, Lian, et al, *Segment anything model for medical images?*, Medical Image Analysis, 2024, vol. 92, p. 103061.
14. KHRISSI, Lahbib, EL AKKAD, Nabil, SATORI, Hassan, et al, *Simple and efficient clustering approach based on cuckoo search algorithm*, Fourth International Conference On Intelligent Computing in Data Sciences (ICDS). IEEE, 2020. p. 1-6.

15. KHRISSI, Lahbib, EL AKKAD, Nabil, SATORI, Hassan, et al, *An Efficient Image Clustering Technique based on Fuzzy C-means and Cuckoo Search Algorithm*, International Journal of Advanced Computer Science and Applications, 2021, vol. 12, no 6.
16. HURTIK, Petr, MOLEK, Vojtech, et HULA,Jan, *Data preprocessing technique for neural networks based on image represented by a fuzzy function*, IEEE Transactions on Fuzzy Systems, 2019, vol. 28, no 7, p. 1195-1204.
17. BAEK, Youngmin, LEE, Bado, HAN, Dongyoon, et al, *Character region awareness for text detection*, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. p. 9365-9374.
18. MEMON, Jamshed, SAMI, Maira, KHAN, Rizwan Ahmed, et al, *Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)*, IEEE Access, 2020, vol. 8, p. 142642-142668.
19. KIM, Geewook, HONG, Teakgyu, YIM, Moonbin, et al, *Ocr-free document understanding transformer*, European Conference on Computer Vision. Cham : Springer Nature Switzerland, 2022. p. 498-517.
20. WANG, Junke, WU, Zuxuan, CHEN, Jingjing, et al, *Object former for image manipulation detection and localization*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022. p. 2364-2373.
21. TANG, Junqing, WAN, Li, SCHOOLING, Jennifer, et al, *Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases*, Cities, 2022, vol. 129, p. 103833.
22. LI, Yinglong, *Research and application of deep learning in image recognition*, IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA). IEEE, 2022. p. 994-999.
23. KAUR, Jaskirat et SINGH, Williamjeet, *Tools, techniques, datasets and application areas for object detection in an image: a review*, Multimedia Tools and Applications, 2022, vol. 81, no 27, p. 38297-38351.
24. KAUR, Parneet, KUMAR, Yogesh, AHMED, Shakeel, et al, *Automatic License Plate Recognition System for Vehicles Using a CNN*, Computers, Materials & Continua, 2022, vol. 71, no 1.
25. ZOU, Yongjie, ZHANG, Yongjun, YAN, Jun, et al, *License plate detection and recognition based on YOLOv3 and ILPRNET*, Signal, Image and Video Processing, 2022, vol. 16, no 2, p. 473-480.
26. PHAM, The-Anh, *Effective deep neural networks for license plate detection and recognition*, The Visual Computer, 2023, vol. 39, no 3, p. 927-941.
27. SHAHIDI ZANDI, Mojtaba et RAJABI, Roozbeh, *Deep learning based framework for Iranian license plate detection and recognition*, Multimedia Tools and Applications, 2022, vol. 81, no 11, p. 15841-15858.
28. AWALGAONKAR, Ninad, BARTAKKE, Prashant, et CHAUGULE, Ravindra, *Automatic license plate recognition system using ssd*, International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA). IEEE, 2021. p. 394-399.
29. ARAFAT, Md Yeasir, KHAIRUDDIN, Anis Salwa Mohd, et PARAMESRAN, Raveendran, *Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework*, IET Intelligent Transport Systems, 2020, vol. 14, no 7, p. 712-723.
30. Silva, S. M., & Jung, C. R., *Real-Time License Plate Detection and Recognition Using Deep Convolutional Neural Networks*, Journal of Visual Communication and Image Representation, 102773. doi:10.1016/j.jvcir.2020.102773.
31. Selmi, Z., Halima, M. B., Pal, U., & Alimi, M. A., *DELP-DAR System for License Plate Detection and Recognition*, Pattern Recognition Letters. doi:10.1016/j.patrec.2019.11.007.
32. Wang, W., Yang, J., Chen, M., & Wang, P., *A Light CNN for End-to-End Car License Plates Detection and Recognition*, IEEE Access, 1–1. doi:10.1109/access.2019.2956357.
33. WANG, Chien-Yao, BOCHKOVSKIY, Alexey, et LIAO, Hong-Yuan Mark, *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023. p. 7464-7475.
34. WANG, Yang, WANG, Hongyuan, et XIN, Zihao, *Efficient detection model of steel strip surface defects based on YOLO-V7*, IEEE Access, 2022, vol. 10, p. 133936-133944.
35. JIANG, Peiyuan, ERGU, Daji, LIU, Fangyao, et al, *A Review of Yolo algorithm developments*, Procedia Computer Science, 2022, vol. 199, p. 1066-1073.
36. PARE, S., KUMAR, A., SINGH, Girish Kumar, et al, *Image segmentation using multilevel thresholding: a research review*, Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 2020, vol. 44, p. 1-29.
37. HOUSSEIN, Essam H., HELMY, Bahaa El-din, OLIVA, Diego, et al, *A novel black widow optimization algorithm for multilevel thresholding image segmentation*, Expert Systems with Applications, 2021, vol. 167, p. 114159.
38. SERRA, Jean, *Mathematical morphology*, Encyclopedia of Mathematical Geosciences. Cham : Springer International Publishing, 2022. p. 1-16.
39. SAMADI, Farnaam, AKBARIZADEH, Gholamreza, et KAABI, Hooman, *Change detection in SAR images using deep belief network: a new training approach based on morphological images*, IET Image Processing, 2019, vol. 13, no 12, p. 2255-2264.
40. SALVI, Massimo, ACHARYA, U. Rajendra, MOLINARI, Filippo, et al, *The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis*, Computers in Biology and Medicine, 2021, vol. 128, p. 104129.
41. TANG, Siyuan et YU, Feifei, *Construction and verification of retinal vessel segmentation algorithm for color fundus image under BP neural network model*, The Journal of Supercomputing, 2021, vol. 77, p. 3870-3884.
42. AGRAWAL, Prateek, CHAUDHARY, Deepak, MADAAN, Vishu, et al, *Automated bank cheque verification using image processing and deep learning methods*, Multimedia Tools and Applications, 2021, vol. 80, p. 5319-5350.
43. DENG, Lizhen, ZHANG, Jieke, XU, Guoxia, et al, *Infrared small target detection via adaptive M-estimator ring top-hat transformation*, Pattern Recognition, 2021, vol. 112, p. 107729.
44. MEMON, Jamshed, SAMI, Maira, KHAN, Rizwan Ahmed, et al, *Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)*, IEEE Access, 2020, vol. 8, p. 142642-142668.
45. NGUYEN, Thi Tuyet Hai, JATOWT, Adam, COUSTATY, Mickael, et al, *Survey of post-OCR processing approaches*, ACM Computing Surveys (CSUR), 2021, vol. 54, no 6, p. 1-37.

46. PADILLA, Rafael, NETTO, Sergio L., et DA SILVA, Eduardo AB, *A survey on performance metrics for object-detection algorithms*, international conference on systems, signals and image processing (IWSSIP). IEEE, 2020. p. 237-242.
47. LI, Yi, ZHAO, Junli, LV, Zhihan, et al, *Medical image fusion method by deep learning*, International Journal of Cognitive Computing in Engineering, 2021, vol. 2, p. 21-29.
48. CAICEDO, Juan C., ROTH, Jonathan, GOODMAN, Allen, et al, *Evaluation of deep learning strategies for nucleus segmentation in fluorescence images*, Cytometry Part A, 2019, vol. 95, no 9, p. 952-965.
49. PADILLA, Rafael, PASSOS, Wesley L., DIAS, Thadeu LB, et al, *A comparative analysis of object detection metrics with a companion open-source toolkit*, Electronics, 2021, vol. 10, no 3, p. 279.
50. Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Goncalves, G. R., Schwartz, W. R., & Menotti, D, *A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector*, International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2018.8489629.
51. BARNOUTI, Nawaf Hazim, NASER, Mustafa Abdul Sahib, et AL-DABBAGH, Sinan Sameer Mahmood, *Automatic Iraqi license plate recognition system using back propagation neural network (BPNN)*, annual conference on new trends in information & communications technology applications (NTICT). IEEE, 2017. p. 105-110.
52. SELMI, Zied, HALIMA, Mohamed Ben, PAL, Umapada, et al, *DELP-DAR system for license plate detection and recognition*, Pattern Recognition Letters, 2020, vol. 129, p. 213-223.