Text extraction and recognition method for license plates

Hanae Moussaoui¹, Nabil El Akkad¹, and Mohamed Benslimane²

¹LISA, Engineering, Systems and Applications Laboratory, ENSA of Fez, Sidi Mohamed Ben Abdellah University, Morocco
²LTI Laboratory, EST, Sidi Mohamed Ben Abdellah University, Morocco

Abstract. Text extraction from images has always been challenging, especially if the image is taken under bad conditions, like lightning and noise that can influence text detection and recognition. This paper introduces a novel text extraction and recognition technique applied to the case study license plates. The main idea of this study is to detect the license plate in an input image and try to figure out the original country of the car based on the license plate. To accomplish this task, we first started collecting images from the internet, which were about 100 images. Afterward, we extracted the license plate using machine learning methods. Subsequently, we applied k-means clustering as well as thresholding in order to segment the extracted license plate and make the character recognition task easier. Thereafter, a sequence of techniques were applied, such as resizing and cropping the image to limit the wanted area of the desired character we want to extract. The last part of the proposed method is reading the text from the image using EasyOcr method, and using the function find in order to search for the character or the word. This proposed method achieved satisfactory results in detection where we achieved an accuracy of 87%, and a recognition of 97%. As for finding the ‘word’ part, the algorithm succeeded in all the examples.

Keywords: Text extraction, text recognition, EasyOcr, image processing.

1 Introduction

In order to detect and recognize a text from images, a variety of devices support optical character recognition ([1], [2], [3]), or OCR which refers to Optical Character Recognition ([4], [5]). With OCR, each character on a page is scanned individually. There are many various OCR solutions that have been used over the years, and the idea of turning a text into an electronic format is broad. The Optophone, which was developed way back in 1914, is one of the earliest electric OCR machines ([6], [7]). It depended on selenium’s unique capacity to conduct electricity differently in the light and the dark. The Optophone identified the black links of text and lighter vacant spaces as it scanned the words on a page. A mechanism that could convert was created later in 1931. One of the earliest methods of converting printed characters into electrical impulses rather than sounds. Furthermore, OCR
didn't start to take on a more recognizable modern shape until the 1960s and 1970s, when postal services started employing it to read addresses in software that could identify a variety of fonts. These days, the first thing that software does when scanning a document is to remove artifacts so that the text may be read clearly without distraction. In order to make the text easier to read, it attempts to eliminate dust and other distracting graphics, align the text correctly, and turn any colors or shades of grey in the image to black and white. The next step is to identify the characters on the page. The more straightforward OCR algorithms ([8], [9]) select the closest match by comparing each scanned letter pixel by pixel to a database of recognized typefaces. However, smarter OCR goes a step further by dissecting each character into its component parts, such as curves and edges, and searching for correspondences between physical traits and actual letters. In order to avoid accidentally producing meaningless words as a result of faulty scanning, OCR software [9] can additionally use a dictionary. OCR is not flawless, but with additional processing power, image segmentation, and machine learning strategies that enable the program to recognize subtler patterns over time, OCR has grown adaptable enough to distinguish difficult-to-read typefaces, inconsistently written content, and even handwriting. OCR can be schematized as shown in the flowchart of the Figure 1 below:

![OCR System Flowchart](image)

**Fig. 1. OCR system**

Image segmentation ([10], [11], [12]) plays a crucial role in OCR systems. As we know, image segmentation is the process of splitting ([13], [15]) an image into a number of informative segments or divisions that will be simpler to study. The term "segmentation" refers to the split or segmentation of an image's pixels into regions of interest, or ROIs. This process may be known as segmentation in some software ([16], [17]), but it may alternatively be known as object identification or detection [18]. The regions of interest may be represented visually using bounding boxes, outlines, or a matrix of labels, which is a collection of randomly colored pixels. because a unique digital label is assigned to each distinct thing [19].

On the other hand, the definition of object detection [20], is a method for finding items in an image or a video. In addition to providing us with the item's classification, object detection
also provides us with the object's bounding box information. To put it another way, object
detection is a sort of image recognition that identifies and pinpoints the presence of an object
in an image. This provides the bounding box information as well as the class [21]. The rest
of this paper will be organized as follows; in the next second section, we'll discuss some
related works in the literature. In the third section, we talk in detail about the proposed
method. Afterward, we bring some obtained results and an evaluation. The paper closes with
a conclusion.

2 Related works

In this section, we'll be citing some previous works in literature, by giving in detail the
proposed method for each work.

In [22] the authors proposed an approach to predicting locations independently, this method
investigates context information, leading to smoother and more accurate detection. It
achieved a recall of 0.71, a precision of 0.74 and an Hmean of 0.73.

In [23] It is demonstrated that a pipeline built on convolutional neural networks can improve
the effectiveness of text identification and recognition by gathering high-level visual data.
On ImageNet and SynthText, a pre-trained ResNet-50 network was used in this study to
extract low-level visual features. Also included in the proposed structure are new, upgraded
ReLU layer (new.i.ReLU) blocks, which have a wide receptive field and excellent text
component detection capabilities even on curved surfaces. Broadly varying-sized text cannot
be produced by a linear sequence of convolution layers as effectively as a new, better
inception layer.

In [24] ReLaText, an innovative method for arbitrary-shaped text detection, is offered by the
authors by reframing text identification as a visual connection detection problem. In order to
demonstrate the effectiveness of this novel formulation, they start by tackling the challenging
text-line grouping problem using a "link" relationship.

In [25] the research focuses on scene text detection and uses an enhanced YOLOv3 network-
based scene text detection technique. First off, this study suggests a way to replace Darknet53
with Darknet19 since Darknet53's backbone network, which YOLOv3 uses, has too many
layers and cannot train quickly for a single detection target. Second, the original network's
multi-scale detection was kept, and the bounding box predictions were made using three
anchors of various sizes.

In [26] the authors of this study introduce TextField, a unique text detector for identifying
irregular scene texts. Specifically, each text point should have a direction field pointing away
from the closest text boundary. This direction field is learned using a fully convolutional
neural network and represented by an image of 2D vectors. It encodes the direction
information needed to distinguish between neighboring text instances and the binary text
mask, which is difficult for traditional segmentation-based techniques to do.

In [27] the authors suggest a reliable text region representation technique for scene text
detection. A text region proposal network is initially used to extract text proposals from an
input image. These recommendations are subsequently validated and improved using a
refinement network.

In [28] the authors of this paper directly train a cross-modal similarity between the text of the
query and each text instance from the natural images to overcome this issue. We specifically
create a cross-modal similarity learning and scene text identification process-optimized end-
to-end trainable network.

In [29] the Pixel Aggregation Network (PAN), which the authors call an effective and precise
arbitrary-shaped text detector, is presented in this research. It includes a segmentation head
with a cheap computing cost and a learnable post-processing component.
3 The proposed method

In this section, we’ll present our proposed method in detail by giving its architecture and the techniques that we used. Furthermore, we’ll discuss the obtained results.

In the first step of the suggested method and after collecting images from the internet, we used machine learning functions to detect the license plate in the input image. Moreover, we applied k-means clustering as well as thresholding to the detected license plate in order to get a well-segmented license plate and, therefore, a well-segmented character. Consequently, and by using k-means and thresholding, we had well-detected characters. Subsequently, in the next step of the proposed method we applied some required enhancements like resizing and cropping in order to keep only the essential part of the detected license plate. Afterward, we used EasyOcr as a technique to read the text from the image and save it in a text file. Finally, a function named find() was used to read the text file and search for a specific character in order to confirm if it exists or not; subsequently, we display the country of the detected license plate. According to the results, we conclude that the suggested approach provides satisfactory detection and text recognition results. The practice of comparing the OCR output with the original version of the same (ground truth) text is known as OCR Accuracy.

3.1 License plate detection

Finding and recognizing license plates in photos or video streams is the problem of license plate detection, a computer vision task. Numerous uses of this technology exist, including toll collecting, parking management, traffic enforcement, and vehicle tracking. The following are the main phases in license plate recognition:

- **Image acquisition:** Acquiring an image or a frame of video that shows a car with a license plate is the first step. Cameras set up in numerous sites, including parking lots, toll booths, and traffic junctions, can be used for this.
- **Preprocessing:** In order to increase the quality and prepare the image submitted for analysis, preprocessing methods including scaling, image enhancement, and noise reduction may be used before license plate identification ([10], [11], [12]).
- **Object detection:** Using object identification techniques, it is possible to locate potential things in an image, such as automobiles, people, and other objects. A zone of interest (ROI) is often created around the area where the license plate is anticipated to be situated in order to detect license plates precisely.
- **License plate localization:** Localization of the license plate within the ROI is required after the algorithm has identified any potential items. This entails determining the license plate's limits inside the image.

It's vital to remember that the complexity and precision of license plate detecting systems might vary based on the particular application and environmental factors. Since these systems include gathering and processing private information, privacy and ethical concerns should also be taken into account.

3.2 K-means clustering

K-means clustering is considered the most popular unsupervised machine learning algorithm that is used especially for data clustering. Its main objective is to cluster together comparable data points into clusters, where each cluster is represented by the centroid (the mean) of the data points that make up that cluster. K-means clustering is often used in many different industries, such as data analysis, image compression, consumer segmentation, and more. Here is how K-means clustering operates [10]:
**Step 1:** The method starts by picking K initial cluster centroids at random, where K is a user-defined parameter indicating the number of clusters to build.

**Step 2:** Each data point is given to the cluster with a centroid that is closest to it in terms of a distance metric, most frequently the Euclidean distance. With each data point belonging to the cluster indicated by its closest centroid, this phase builds K clusters.

**Step 3:** Centroids Recalculated: After all of the data points have been allocated to clusters, the centroids of each cluster are recalculated as the mean (average) of the data points included inside that cluster.

**Step 4:** Steps 2 and 3 are iteratively repeated until one of the halting conditions is satisfied. A maximum number of iterations or when the centroids reach a certain value are common termination conditions.

**Step 5:** When the algorithm has finally reached a point of convergence, the cluster centroids are what remain. K clusters are created out of the data points.

**Step 6:** The K clusters, each of which is given to a single data point, are the final output. These clusters may be utilized for analysis, visualization, or decision-making depending on the grouped data, among other things.

The kmeans clustering formula can be written as follows ([10], [11]):

\[
J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2
\]  

(1)

J represents the objective function  
K represents the cluster’s number  
N is the case’s number  
x_i^{(j)} refers to the case i  
c_j is the centroid for the cluster j

The following Figure 2 summarizes the whole K-means clustering algorithm:

![K-means process diagram](image)

**Fig. 2. K-means process**

### 3.4 Thresholding

In a binary image, thresholding is a typical image processing technique used to distinguish objects or characteristics of interest from the background. Setting a threshold value, a certain intensity level or range of values in a grayscale image, and then converting the image into a binary format, where pixels above the threshold are assigned to one (white), and those below the threshold are set to zero (black), is the first step in the process.
Here is a brief explanation of how thresholding functions ([13], [14]):

- **Grayscale conversion**: The incoming image is often transformed to grayscale first if it is in color. As it focuses on intensity values rather than color, this makes thresholding simpler.

- **Selecting a Threshold Value**: We must select a suitable threshold value. The specific application and the attributes of the image will determine the threshold to be used. Manual selection, Otsu's approach (which automates threshold selection based on image histograms), and adaptive thresholding (where the threshold moves across the image to accommodate for local differences in lighting and contrast) are some typical techniques for threshold selection.

- **Threshold operation**: The grayscale image's pixels are compared to this threshold once the threshold value has been selected. The pixel is turned to white if its intensity is greater than or equal to the threshold; otherwise, it is set to black. This produces a binary image where the backdrop is black and the things of interest are represented in white. Thresholding is frequently employed in many different image-processing applications, like medical imaging, image segmentation as well as character recognition. Moreover, the quality of the input image, the threshold value used, and the unique qualities of the objects and background in the image all affect how well thresholding works. To choose the best thresholding technique and threshold value for a particular activity, some thought and testing may be necessary.

The basic binary thresholding mathematical formula can be written as follows:

\[
T_{\text{low}}, \text{if } I(x, y) < \text{threshold} \\
T_{\text{high}}, \text{if } I(x, y) \geq \text{threshold}
\]  

Where:

- \(I(x, y)\) represents the pixel’s intensity at a position (x, y) in the greyscale image.
- Threshold is the designated threshold value.
- \(T_{\text{low}}\) is the value given to pixels whose intensities fall below the threshold (often 0, signifying black).
- \(T_{\text{high}}\) is the value given to pixels whose intensities exceed or are equal to the threshold (often 255, which stands for white).

This simple binary thresholding method produces a binary image with a separate foreground and background by setting pixels with intensity values below the threshold to \(T_{\text{low}}\) and pixels with intensity values at or above the threshold to \(T_{\text{high}}\).

### 3.5 OCR

Optical character recognition is known as OCR. It is a technique that makes it possible to transform many kinds of documents, including editable and searchable data from scanned paper documents, PDF files, and digital camera images. OCR software converts the text characters seen in these photos or scanned documents into machine-readable text by identifying them.

To apply OCR, we need to follow the main steps below:

**Step 1**: Preprocess the input image. Therefore, cleaning and enhancing the image will improve it in terms of quality, noise removal, contrast enhancement, and suitability for character recognition.

**Step 2**: Find areas or bounding boxes in the image that contain text using the bounding box detection technique. This technique is called text localization.

**Step 3**: in the third step of the OCR process, we segment the detected text by:

- Dividing the text into lines using line segmentation technique.
- Word Segmentation by separating the lines into their constituent words.
- Segment words into characters to make them easier to recognize.
**Step 4:** Extract different characteristics, such as shape, size, texture, etc., from the segmented characters or words. This technique is called Feature extraction.

**Step 5:**
- Identifying characters by comparing each character's extracted features to pre-built templates or models. This technique is called pattern matching.
- Employ machine learning models, such as neural networks, to identify characters by training them on labeled character data.

**Step 6:**
- By taking into account the context of nearby letters or phrases, language models can be used to increase the precision of character recognition.
- Error Correction: Use correction algorithms to fix recognition mistakes and boost the output of OCR's overall accuracy.

The following figure 3 presents the essential key steps that are involved in OCR process:

![OCR process diagram](image)

**Fig. 3. OCR process**

The figure 4 represents a detailed explanation of the proposed method:

![Proposed method diagram](image)

**Fig. 4. The proposed method**
Table 1 below shows the obtained results:

<table>
<thead>
<tr>
<th>Original image</th>
<th>Detected license plate</th>
<th>K-means + Thresholding</th>
<th>OCR</th>
<th>Character-level accuracy (CLA)</th>
<th>The scanned results</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Original image" /></td>
<td><img src="image2" alt="Detected license plate" /></td>
<td><img src="image3" alt="K-means + Thresholding" /></td>
<td><img src="image4" alt="OCR" /></td>
<td>95%</td>
<td>This car if from France</td>
</tr>
<tr>
<td><img src="image5" alt="Original image" /></td>
<td><img src="image6" alt="Detected license plate" /></td>
<td><img src="image7" alt="K-means + Thresholding" /></td>
<td><img src="image8" alt="OCR" /></td>
<td>96%</td>
<td>This car if from Spain</td>
</tr>
<tr>
<td><img src="image9" alt="Original image" /></td>
<td><img src="image10" alt="Detected license plate" /></td>
<td><img src="image11" alt="K-means + Thresholding" /></td>
<td><img src="image12" alt="OCR" /></td>
<td>98%</td>
<td>This car if from Germany</td>
</tr>
<tr>
<td><img src="image13" alt="Original image" /></td>
<td><img src="image14" alt="Detected license plate" /></td>
<td><img src="image15" alt="K-means + Thresholding" /></td>
<td><img src="image16" alt="OCR" /></td>
<td>97%</td>
<td>This car if from Ukraine</td>
</tr>
</tbody>
</table>

To evaluate the performance of our proposed method, we used the character level accuracy (CLA) metric. Character-level accuracy quantifies the proportion of correctly identified characters out of all characters. It's outlined as [30]:

\[
CLA = \frac{\text{Number of correctly recognized characters}}{\text{Total number of characters}} \times 100\% \tag{3}
\]

## 4 Conclusion and future work

In this paper, we presented a new technique for text extraction and recognition in the case of license plates. For this, we first applied machine learning algorithms to extract the license plate from the input image. Secondly, we applied K-means clustering and thresholding techniques in order to segment the extracted license plate. Afterward, we added some enhancements like resizing and cropping in order to limit the area. EasyOcr was afterward applied for text extraction, consequently, a text file will be generated containing the text or the character. Finally, we used a function called ‘find’ to search for the word in the text file and get the name of the country of the car as a result. The proposed method obtained satisfactory results for detection, text extraction, and recognition. This technique can be used in any use case that requires text detection and recognition as well.

In future work, we’re planning to integrate deep learning into this process and generalize it to different types of license plates. Therefore, in the first stage, we’ll gather images from the
internet of the different types of license plates, and by using data augmentation we can develop the number of the created dataset. Subsequently, a set of image processing algorithms will be applied in order to enhance the quality of the images. Afterward, we’ll use convolutional neural networks (CNNs) in order to detect the license plate area. Eventually, we will use the technique that we suggested in this paper to determine the country to which the car belongs.

5 References


