summary

The aim of this project is to detect and recognize car license plates based on computer vision and digital image processing techniques. An algorithm based on edge detection and morphological operations was used and implemented in Matlab. We were finally able to recognize license plates with a certain standard shape that will be shown in the examples included in the report by making use of a preprepared list of English character image.

Chapter 1

Project introduction

In this chapter of the report, we present a general introduction, the objective of the project, its work stages, and the flow chart of the proposed algorithm.

General Introduction

Object detection and identification are among the most important challenges facing us today in video surveillance systems. These systems are used to ensure the security of the facilities in which they are placed and to detect violations and intrusions, as they are mainly used for the purpose of monitoring military and security bases homes, hotels, shops and controlling traffic violations

Vehicle identification using images is one of the most important topics that has occupied the attention of image processing enthusiasts all over the world for several years. Many experiments have been conducted in this field over the past years, but despite that, the subject is still a significant challenge.

Car plates are used all over the world to identify and distinguish a car from other cars. These plates can be detected manually or automatically. The automated methods rely on the Automated License Plate Detection System (ANPR)(Automatic Number Plate Recognition). We have always wondered how this system works. The idea is to take a picture of the car to be recognized, provided that the car plate is visible in the picture, whether the front or rear plate. This image is then passed through a set of algorithms that aim to detect the location of the plate in the picture and then read the image of each character of the plate, recognize it and Since the plate can be located .convert it into a text value (number or letter) anywhere in the picture, it is not practical to examine the image pixels by pixel to determine its location, as this will become expensive, complicated and ineffective. This is where the idea came from to adopt faster and more effective methods and algorithms. These algorithms rely on digital image processing techniques such as edge detection, smoothing, erosion and morphological operations.

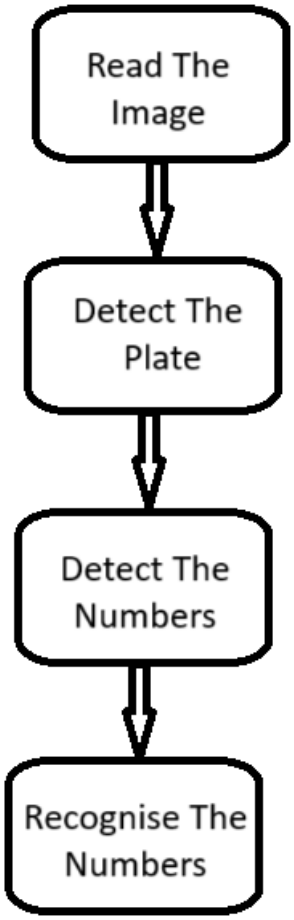
In this work, we present a method for detecting car plates and recognizing their numbers based on computer vision and digital image processing techniques. This algorithm has been implemented in MATLAB.

Project objective

This project aims to detect and recognize car plate using digital image processing techniques using MATLAB.

Flowchart of the algorithm followed

Figure (1) shows the flow chart of the algorithm followed in the projec



.Figure (1): Flowchart of the algorithm followed in the project

The algorithm is sequenced in five stages:

- The first stage: In this stage, a picture of the vehicle whose identity is to be identified is read, showing its front or rear plate.
- The second stage: In this stage, the location of the plate is revealed in the photo- taken of the car .

- The third stage: After revealing the location of the plate in the image, its letters are shown, and then the image of each letter is separated separately .based on the processes of cutting and revealing areas
- The fourth stage: In this stage, the identity of each character is revealed by comparing it with a list of previously prepared images of all English characters.
- The fifth and final stage: In this stage, the characters of the board are displayed after being recognized.

Chapter 2

Implementation of the proposed algorithm

In this chapter of the report, we present a detailed explanation of the algorithm .followed with examples

Stages of the algorithm

Take a picture of the car

At this stage, a front or back photo of the vehicle to be identified is taken provided that the plate number is visible in the photo. Figure (2) shows an example.

Figure (2): Back view of the vehicle to be identified.

Detect the location of the plate

First of all, the color image is converted to grayscale. Figure (3) shows the original image after converting it to grayscale. The aim of this process is to reduce the time required to detect the painting in the image.

Figure (3): The original image after converting it to gray level.

The main task in license plate recognition is to know the area of the plate in the image. Generally, the plate is rectangular in shape, so in order to detect the plate we must first detect the sides of this rectangle, i.e. we must perform an edge detection process. APrewitt filter was used to accomplish this task. Figure (4) shows the image after detecting its edges. In MATLAB, there is a ready-made function for detecting edges callededge .

After detecting the edges, a gap filling operation is performed. MATLAB provides this operation thanks tothe imfill function. The input of this function is the binary image resulting from the edge detection operation. Figure (5) shows the output of this operation. We notice that the location of the car plate has become filled and distinct.

In order to show only the location of the plate and remove the rest of the visible areas in the image, a scraping operation can be performed using the imerode function. We performed an erosion operation on the image shown in Figure (5) using a disk with a radius of 8 pixels. Figure (6) shows the result of the erosion operation. We notice that the location of the plate became clear without noise.

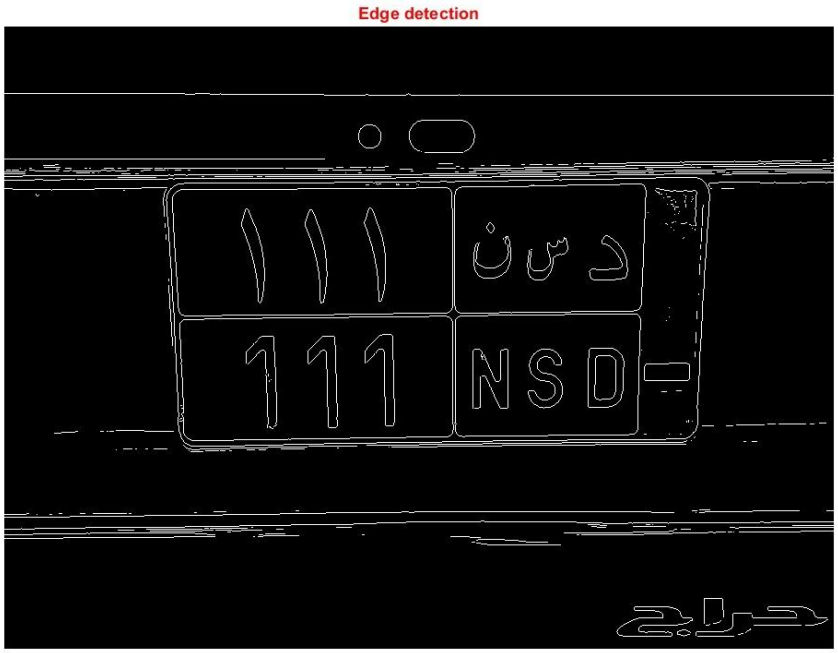


Figure (4): The gray image after performing the edge detection process.

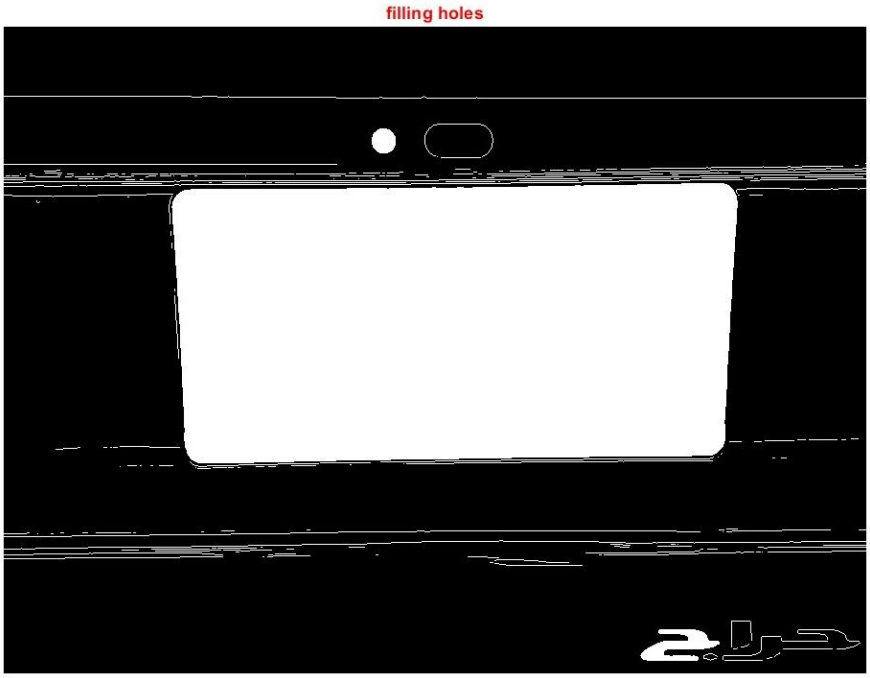


Figure (5): The image after filling in the gaps.

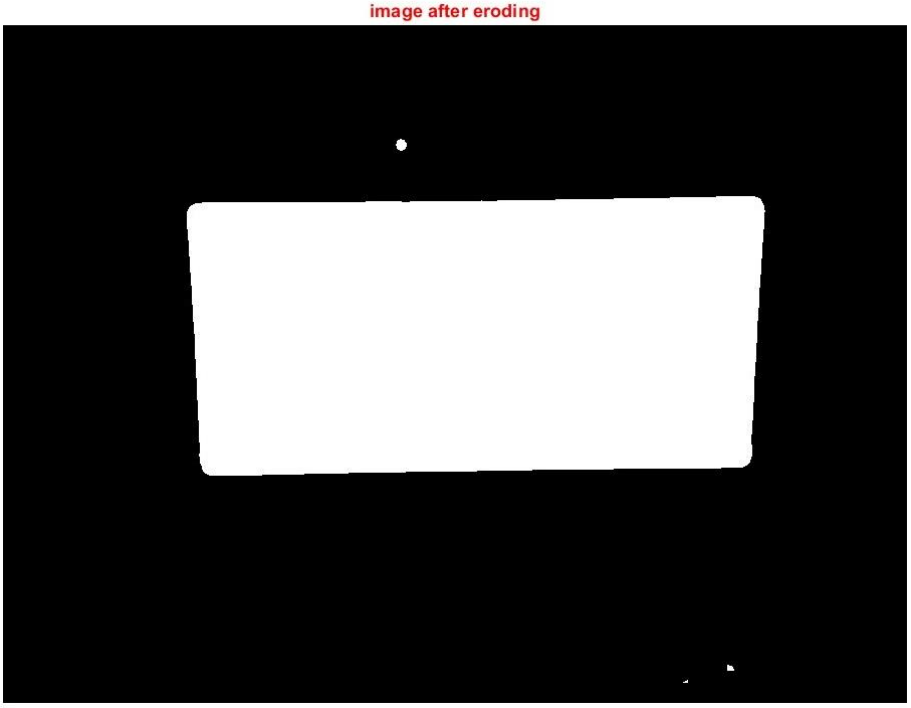


Figure (6): The image resulting from the erosion process.

Recognizing keyboard characters

The characters of the painting can now be easily displayed by multiplying the image in Figure (6) resulting from the erosion process with the binary image resulting from the detection of the edges of the gray image. Figure (7) shows the painting after displaying its characters. We still have to subtract the painting from the image. This can be done using the regionprops function which gives us the properties of the image. This function returns us the smallest bounding boxthat contains all the regions of the image. The regions in the image are only the characters of the painting, so the subtraction process gives us only the painting. The subtraction is done usingthe imcorp function Figure (8) shows the image resulting from subtracting the painting. Figure (9) shows the plate after removing small and non necessary areas.

To identify characters, we need to extract each character from the image and then compare it with our list. This can be done using the regionprops function which extracts each region from the image and stores it in a variable, where the regions are the characters of the board. The comparison is done by calculating the correlation coefficient between the extracted image and all the characters of the list using the corr2 function The correlation coefficient is maximum when they match. Thus, we can deduce the value of the character. Figure (10) shows the extracted character D. Figure (11) shows the extracted character S. The same applies to the remaining characters.

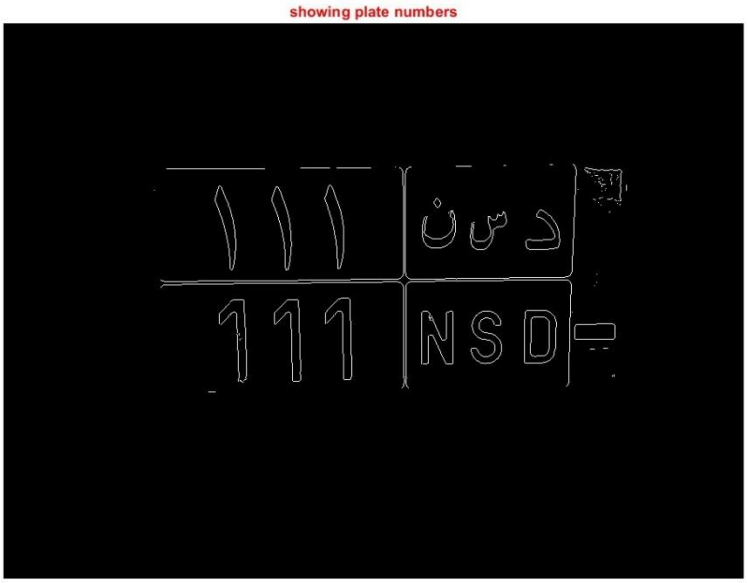


Figure (7): Showing the panel characters.

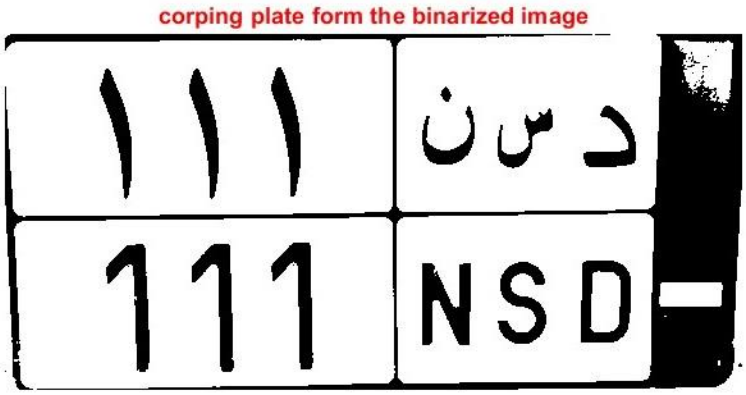


Figure (8): Car plate truncation.

.Figure (9): Car plate after removing small areas

Figure (10): The truncated letter D.

Figure (11): The truncated letter S.

Show characters

The characters are stored in a variable namedPlatenumber Figure (12) shows the . result in a MATLAB program.

```
Platenumber =

    '111NSD'
```

Figure (13): Showing the panel characters.