



License plate detector

Name: Ahmed Hossam

ID: 176147

Major: AI

**Introduction:**

This project is one of the most important application of image processing because it indicates the importance of image processing techniques in the license plate character identification. We use a lot of techniques to extract the characters, number of characters, morphological and logical techniques segmentation and OCR. We use morphological techniques in extract the characters and numbers from the image, segmentation in cropping the plates and get every letter and number individually, then the color of each plate to identify the type of every given plate in the image . Finally, we use OCR by comparing between every segmented object and the characters and number in the dataset to identify every character and number from the license plate.

**Functions’ description**

|  |  |
| --- | --- |
| **Imread()** | To read the image |
| **Rgb2gray** | Converting rgb images to gray scale image |
| **imbinarize()** | To convert gray-scale image to zeros and ones (binary image) |
| **imfill()** | Fill holes in the image to make the images consistent and easier to deal with |
| **bwareaopen()** | Remove objects in the image by a given area threshold to smooth the image and remove any noise |
| **regionprops()** | Return properties of the images like the orientation (which the is rotation value of the shape) and bounding box (which are 4 values that are equal to x-axis and y-axis of the upper left corner of the shape, the height and the width of the shape) of every shape in the image and it returns it in a struct form |
| **imrotate()** | Rotate the plate in the image by a given degree and performs a specific action like cropping the image after rotating. |
| **Cell()** | Creates array by a given dimensions |
| **Imcrop()** | Crop the image by a given size |
| **Imresize()** | Resizing image by a given dimensions |
| **Imerode()** | Erode objects in the image to make it less noisy and the objects become thinner |
| **Corr2()** | Brings the correlation between an image and another image or set of images, the bigger the value the close is the match between the photos. |
| **Numel()** | Count the all the pixels in an image |
| **Dir()** | Creates a directory based on a given path |
| **Strcat()** | Concatenates between two strings |
| **Strel()** | Create a structuring element (SE) with various shapes like “disk, diamond, line” with specific size |
| **Imdelate()** | enlarges objects in the photo by adding pixels with the shape and size of the created (SE) |
| **Bwareafilt()** | extracts all connected objects from the binary image, where the area of the objects is in the specified range, |
| **Rectangle()** | Draws a rectangle shape in the given dimensions and given area and given color |
| **Zeros** | Creates a matrix of zeros with a given dimensions |
| **Imshow()** | Plots the given image |
| **Subplot()** | Creates a multi-plot figure with specific dimensions |

**Code:**

After reading the image and turning it into grayscale we binarize the pixels to be able to do the morphological and logical operations like imfill and bwareaopen to make the image easier to deal with. Using the functions mentioned earlier we fill the holes in the binarized image then we removed every object that has an area smaller than 200.Moreover, we bring the orientation and bounding box from every binarized image using the region props function then we count the number of the bounding box values retrieved from the image and put them in an array. We do two conditions one for one plate and the other for four plates then we crop every plate in the image based on its bounding box values and put in two cells one for the binarized plate to get characters and numbers further on and the second cell have the RGB plates to bring the color of every plate. If is there a rotated plate we correct it to its rectangular position by checking on its orientation value if it is bigger than a certain value we rotate it using imrotate() function to be able to get the characters from the plate as the same way in the other images. We put the template images in the data set in a cell then we resized the template images to 42 by 24 to have the same size of the cropped characters from the plates. We crop the binary image from the top to make the image only contains the characters, then We bring the bounding box of every letter in the binarized image. Looping over bounding boxes and make a rectangle in the edge of every character in the plate then we cropped the letters in the plate and put every letter in a single cell then eroding and resizing every character to 42 by 24 to be of the same size of the dataset images then bringing the correlation between every object in the plate and comparing it to all the characters and numbers in the dataset. We recognize the object on the plate by bringing the maximum correlation from the set of computed correlation from both the letters and the numbers then getting the max between the two max values to identify wither this character is a letter or a number. After, recognizing all the numbers in the image we print the governorate and then plot the license plate characters in a subplot () .in addition to that, we cropped the upper part from the RGB plate then we dilate every cropped image to have the color only by a given range, after we get the new images with these color ranges we compare between them to retrieve the image with the highest white pixels to be the correct color of the plate.

**Challenges:**

The challenge we faced during this project is finding the images of the Arabic letters and numbers due to the wide variety of fonts, to solve this problem we downloaded a lot of Arabic fonts in order to have larger comparing dataset. The second challenge was finding the color ranges of the Egyptian license plates, to solve this problem we used a color picker tool in order to find the right range of the license plates.