Presentation

Good evening, everyone. It’s our pleasure to present this project to you. The project is about the recognition of vehicle plate. vehicle plate recognition plays an important role in numerous applications such as unattended parking lots, security control of restricted areas, traffic law enforcement, congestion pricing(高峰期行车收费), and automatic toll collection（自动收费）. It can save time and alleviate congestion by allowing motorists to pass toll plazas or weigh stations without stopping. It can save money by collecting and processing vehicle data without human intervention. It can also improve safety and security by helping control access to secured areas or assisting in law enforcement. In result, this project is meaningful and practical.

(change)It mainly includes three parts: First, extract the plate from a picture of vehicle; Second, extract the characters from the plate; Finally, match the characters.

(change)The basic environment we used are Python2.7, tesseract 3.05.00 and Opencv3.1.0.

The main method adopts the way of space domain filtering and morphological processing to locate the vehicle plate, rectifies the beveled plate using a combination of frequency domain and space domain analysis and geometric rectification, and achieves plate number recognition by character segmentation and matching.

(change)As we all know, photos usually have much noise, so we do the grayscale to original images for further processing. In Opencv, we use the below function to change a three-channel RGB color image into a single channel, that is how we obtained the grey image.

(change)The first stage of the process is to locate the plate area in the input photo. In order to do so, we need to emphasize regions of high spatial frequency that correspond to edges. It is noticed that most of vehicles usually have more horizontal lines than vertical lines. If the two vertical edges of a license plate can be detected correctly, four corners of the plate can then be located, so that the license plate can be extracted exactly from the input image even if it is out of shape. Thus, as an alternative, only the vertical edges of input image are used to extract license plate.

We first use the classic Sobel filter. We use the processed grey image as the input to this step, then do x-direction sobel detection to preserve the vertical features of plates and eliminate the vehicle body and contour features(which are horizontal). After that, we convert the image to 8-bit for further processing.

(change) Next in the process is the binarization. We obtain a black and white image from the result of the pre- vious transformation. We use the adaptive binary process in Opencv, which enhances the sharp edges, so that remove useless information in the image. By doing this, we got the binary image. In a binary image, a region is defined as a set of white pixels that are eight connected with each other.

(change)Then is the most important step of this stage. We do dilation and erosion to define the plate area. By dilation we connect the similar area, and remove the small isolated fragments. By doing this, all the characters on the plate can connected together, which is good to contour recognition for selecting plate area. Since the characters are horizontal arranged, we just need to do horizontal dilation to connect them. First, create a square in x-direction; Second, connecting characters by x-direction dilation; Third, eliminating fragment by x-direction erosion; Finally, repeat steps above in y-direction again.

After applying the last step of the previous stage, we have a black image with one or more white areas candidate to be a license plate. However, only one of those areas corresponds to the real license plate, while the others are consequence of noise or other objects in the background. Now, we need an algorithm to deter- mine which one of these white areas is the real plate.

We have used several criteria. First, we only keep the biggest areas, this way we eliminate most of the areas due to noise. We then order those ten plate candidates according to the following criteria:

• ratio of width to heights greater than two, since a license plate is wider than taller.

• distance to the center of the image, the areas far from the center are unlikely to be plates.

Here is the code of this step.

For two areas with similar characteristics we choose the one with lower positions, we do this because we have found that sometimes the area due to the edges in the car radiator will give end as a false positive plate.

(change) After the previous stage, we obtained a license plate image. Similar to extract plate from image, extracting characters also needs grayscale, binarization, contour detection, selecting and locating. We need to do further image processing before starting to locate the characters in the plate. First, the most likely cause of failure is the case of the character extracting light-colored border, it communicates with the license plate letters, resulting contour detection failure

First, the most likely cause of failure is the case of the mis-extracting light-colored border, it connected with the license plate characters, resulting contour detection failure. So after the first initial contour detection, we need further processing of the border, after then do contour detection again, and then filtered. After all these steps, we can basically recognize each character from the plate. For the process of border, Firstly, crop the image progressively by the Y direction until can detect more than five contours. According to the detected rectangular contours to identify the upper and lower bounds of all vehicle license plate characters based on their cut image.

Similar to plate extraction, the detected rectangular contour need further filtration, mainly include sorting of height, width, area.

(change)Using tesseract to match characters

Tesseract has unicode (UTF-8) support, and can recognize more than 100 languages "out of the box". It can be trained to recognize other languages.

(change)To verify the practical of this method, we did experiments on different photos of vehicles. The results show that our project is effective and meaningful.

However, there are some problems we met but don’t have enough time to solve, such as the case of the slope, sometimes we get the photos that the vehicle is slope by a big margin, which raise difficulty to extraction of plate.

Besides, An important characteristic we could extract from the character image is the number of holes in the characters. That increase the classification accuracy for characters like “P” and “F” or “B”, “8”, “6” and “9”. These mind-boggling characters may cause incorrect matching, so that the failure of recognition.

In future version, we will try to solve these problems to make this project close to perfect.

Thanks for listening, this is the end of our presentation.