

Car license plate detection

1. RGB to Gray
2. Median filter
3. Find vertical edge
4. Threshold
5. Use $20 * 30$ matrix morphological closing
6. Find rectangle contours and filter out rectangles with incorrect aspect ratios
7. Remove the rectangle without enough objects by doing character detection



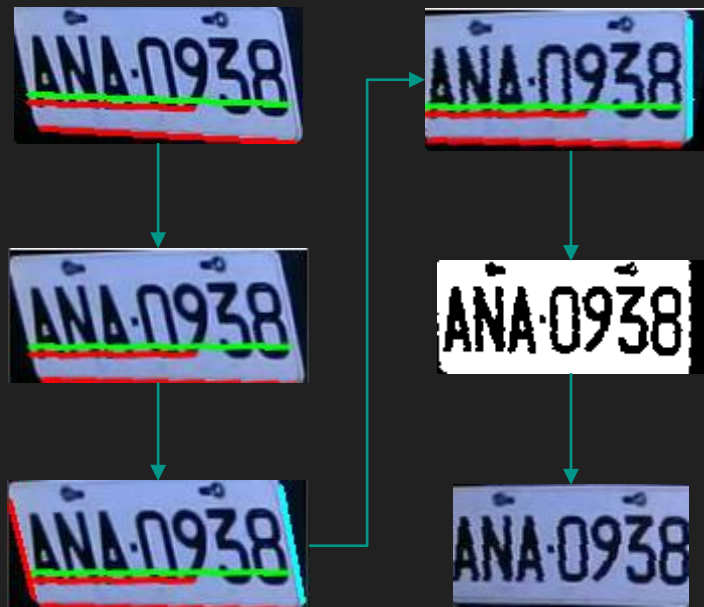
Character detection

1. RGB to Gray
2. Gaussian filter
3. Threshold
4. Apply morphological opening with the vertical kernel and horizontal kernel
5. Find rectangle contours and filter out those where the height is not enough



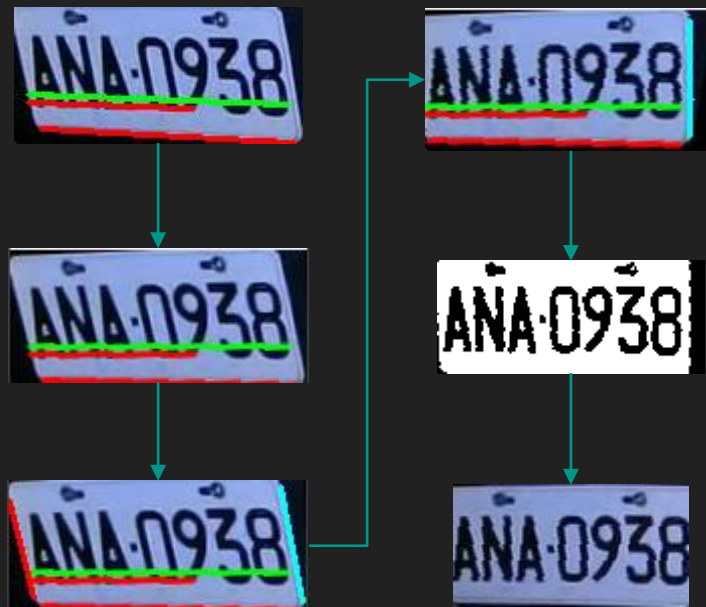
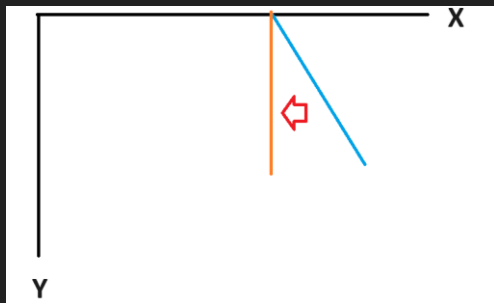
Skew correction on car license plate

1. RGB to Gray
2. Median filter
3. Canny edge detection
4. Find the longest line with a slope less than 1 (green line)
5. Rotate the green line to be horizontal



Skew correction on car license plate

6. Find the longest line with a slope more than 1 (blue line)
7. Horizontal shift
8. Threshold
9. Cut left and right redundant part



OCR to recognize characters

1. Use function in “pytesseract”
2. Filter out not number or Capital letter
3. After recognizing all frames, calculate the average length of the strings
4. Filter out strings under or above the average length
5. Voting
6. Print the result

```
text = pytesseract.image_to_string(gray_plate, config='--psm 8').strip()
filtered_text = re.sub(r'^A-Z0-9', '', text)
```

1	BAF0315
2	RAE0SI9
3	RAED319
4	BAF0S15
5	BRED519
6	AA4517
7	AAPUS17
8	RAFUS17
9	AAUSI7
10	AARUS17
11	AAFAISI7
12	RAL4I59
13	RAAFAIS
14	FASC
15	FAS
16	AA4517
17	RARAI57
18	AAK4ISIF
19	RAL457
20	SALIS7
21	AIS17
22	AATL7
23	AAT7
24	AAT919
25	AARDIZ
26	AAT517
27	AAPUSI7

Result



002 ANA0938