License Plate Reader

Dataset

`CharacterData.zip`, single character images.

- 0-9
- A-Z, except I, O
- Province character (mainland)

Totally, 10+24+31=65 characters.

Baseline

Provided by instructor, by CNN

```
[10, 3, 57, 24, 23, 49, 16]
贵AQP679
```

Image Process

Input Image (170 * 625)

```
license_plate = cv2.imread('./车牌.png')
```

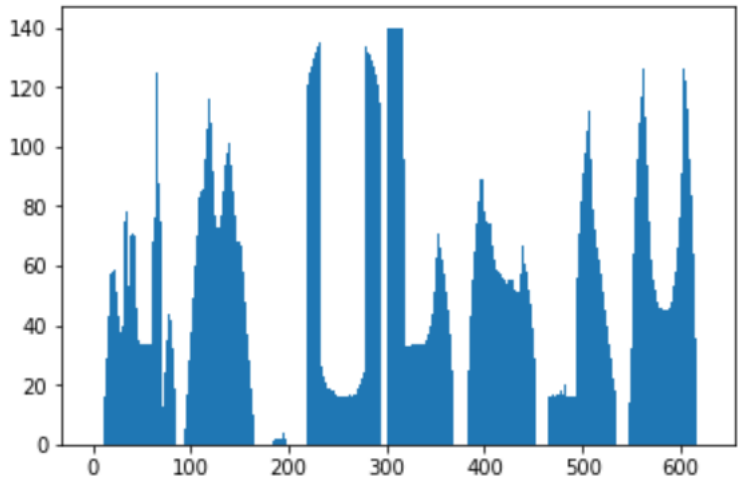
Binary (160 * 625)

```
gray_plate = cv2.cvtColor(license_plate, cv2.COLOR_RGB2GRAY)
ret, binary_plate = cv2.threshold(gray_plate, 175, 255, cv2.THRESH_BINARY)
```

Column Pixels Sum Bar (625 columns)

```
result = []
for col in range(binary_plate.shape[1]):
    result.append(0)
    for row in range(binary_plate.shape[0]):
        result[col] = result[col] + binary_plate[row][col]/255

# Bar
import matplotlib.pyplot as plt
ys = result
xs = np.arange(len(result))
width=1
plt.bar(xs, ys, width, align='center')
```



Get characters edge (column)

```
character_dict = {}
num = 0
i = 0
while i < len(result):
    if result[i] == 0:
        i += 1
    else:
        index = i + 1
        while result[index] != 0:
            index += 1
        character_dict[num] = [i, index-1]
        num += 1
        i = index
```

```
{0: [11, 85],
 1: [94, 165],
 2: [184, 197],
 3: [218, 293],
 4: [301, 367],
 5: [384, 452],
 6: [466, 534],
 7: [548, 617]}
```

Output

```
characters = []
for i in range(8):
    if i==2:
        continue
    padding = (170 - (character_dict[i][1] - character_dict[i][0])) / 2
    ndarray = np.pad(binary_plate[:,character_dict[i][0]:character_dict[i][1]], ((0,0), (int(padding), int(padding))), 'constant', constant_values=(0,0))
    ndarray = cv2.resize(ndarray, (20,20))
    cv2.imwrite('./' + str(i) + '.png', ndarray)
    characters.append(ndarray)

#display
from PIL import Image
imgs = [Image.fromarray(character) for character in characters]
for img in imgs:
    display(img)
```