차량번호판 인식 코드분석

- OpenCV 모듈은 사진을 행렬을 이용하여 처리

- 가상환경 구축  
 모듈의 의존성을 분리하기 위함

- KNN 알고리즘 (Deep learning)을 이용하여 영문 및 숫자 인식

**주요 외부 모듈 (Python 3.6)**

1. [O] **opencv-python** ; not support Python 3.7

// https://github.com/skvark/opencv-python

[필요한 모듈]

[O] **numpy**

// https://docs.scipy.org/doc/numpy/index.html

- https://www.numpy.org/devdocs/reference/

2. [X] PIL (Python Imaging Library) ; ~ 2011

// http://effbot.org/imagingbook/

[O] **Pillow** ; 2011 ~

// <https://pillow.readthedocs.io/en/5.2.x/>

**// 자세한 사항은 코드 및 주석을 참고**

전체적인 처리과정

**1. 각 사진에 전처리 과정을 적용** 정확도를 높이기 위해 원본 사진에서 흑백사진, 흑백사진에서 이진화 사진을 추출



**2. 이진화 사진으로부터 윤곽단위로 나누고, 각각의 윤곽이 번호판이 될 수 있는 경우에만 이 객체를 저장** - 새 이미지를 생성하여 이 윤곽들만 그림

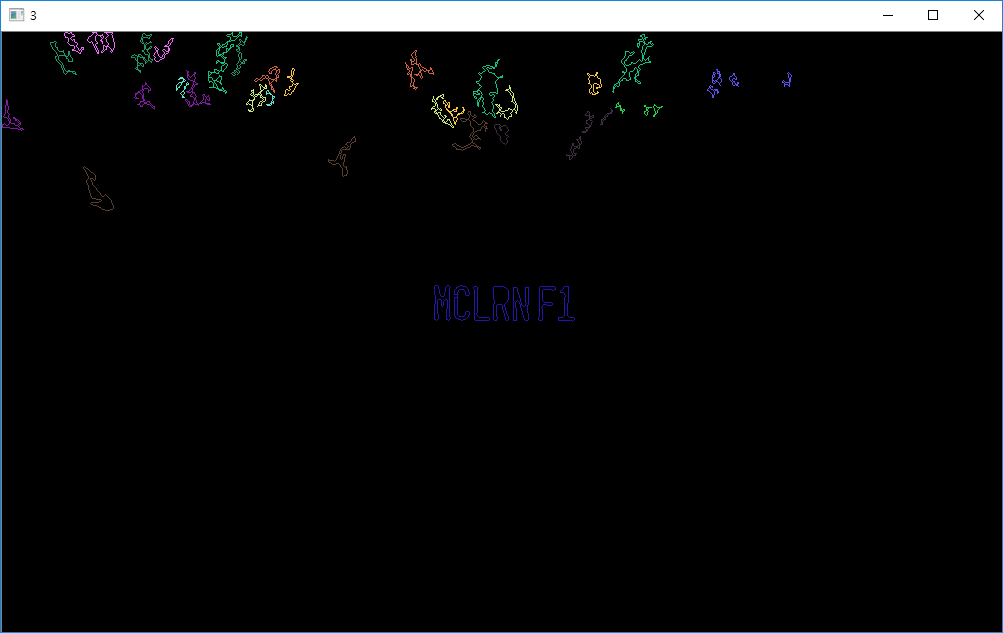




**3. 이 각각의 윤곽들 안의 문자열들의 문자가 영문 및 숫자로 인식될 수 있는 길이, 크기 등의 전제조건을 확인하여 범위를 좁힘**

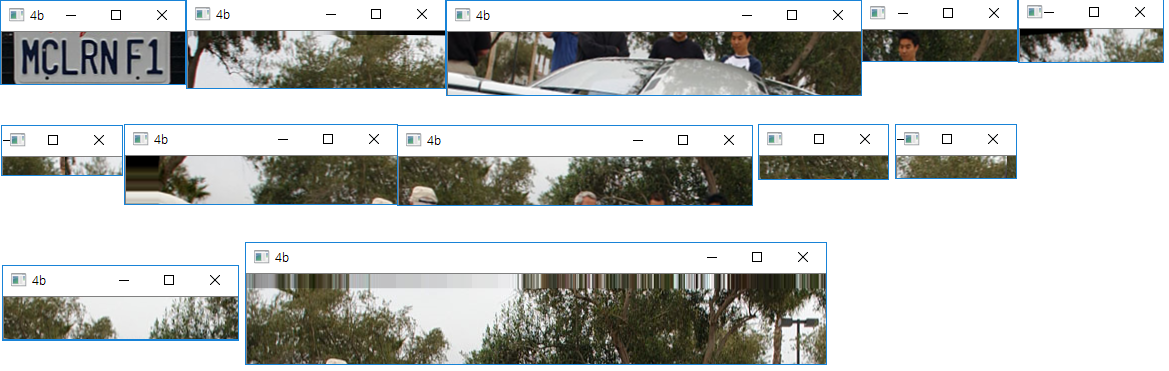


- 새 이미지를 생성하고 좁혀진 윤곽들을 그림



- 좁혀진 각 윤곽마다 사각형을 그리고 원본 이미지에서 이 위치를 잘라 추출



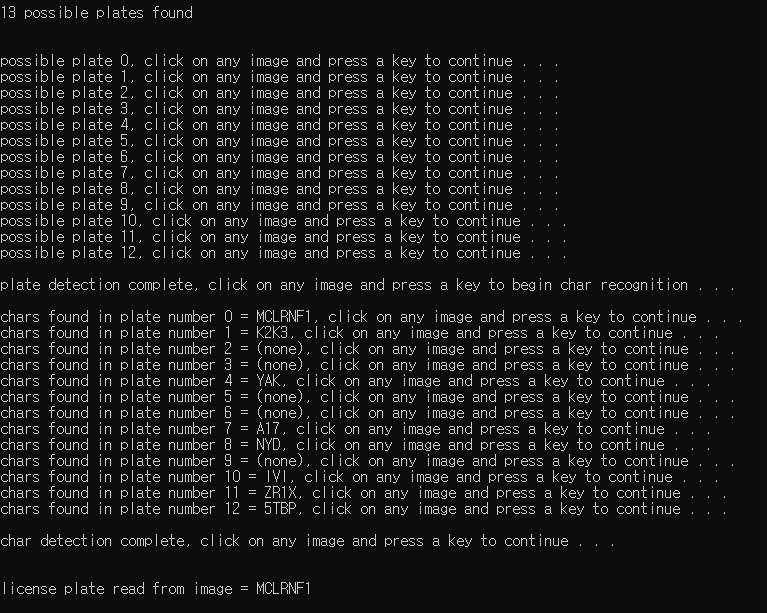


**4. 위에서 각 사진들을 받아 “1” 의 전처리 과정을 똑같이 적용함**

**5. 각 사진들의 이진화 사진에서 각도를 변경 및 가능한 문자들인지 확인**

- 번호판이 될 수 있는 문자들의 개수나 크기 등의 조건을 확인, 최대 문자의 개수를 정하고 그 이하의 문자들만 허용  
 - 문자들의 개수가 제일 긴 것 중 정확도가 제일 높은 것들을 추출





Reference

<https://pillow.readthedocs.io/en/5.2.x/>

<https://pypi.org/project/opencv-python/>

<https://github.com/opencv/opencv/releases>

<https://pypi.org/project/numpy/>

<https://docs.opencv.org/2.4/modules/refman.html>

<https://en.wikipedia.org/wiki/Gaussian_function>

<https://docs.opencv.org/3.4/d3/dbe/tutorial_opening_closing_hats.html>

<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>