**인공지능 번호판 인식 데이터 보고서**

**<차량 번호판 인식 OCR>**

목차

1. 데이터 명세 정보 ············································································ 1

1.1 데이터 정보 요약 ···································································· 1

1.2 데이터 정의서 ······································································ 1

2. 데이터 구축 가이드 ···································································· 2

2.1 데이터 개요 ··································································· 3

2.1.1 예시데이터 확인 ······························································· 3-4

2.2 테스트데이터 ····································································· 5-6

2.2.2 테스트데이터(2) ······························································· 7-8

2.3 문제점 파악 ··································································· 9

2.4 해결 방안 모색 ····································································· 10

2.5 문제 해결 ······································································· 11-12

3.결론 ···························································································· 13

1. 데이터 명세 정보

1.1 데이터 정보 요약

| 데이터 이름 | 차량 번호판 인식 OCR | | | |
| --- | --- | --- | --- | --- |
| 활용 분야 | OCR 기반 차량 번호판 인식서비스 개발 | | | |
| 데이터 요약 | 일상에서 보이는 차량 번호판을 수집 가공하여 인공지능 학습 데이터 구축 | | | |
| 데이터 출처 | 블로그 -나의 개발일기 [https://velog.io/@mactto3487/%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-OpenCV-%EC%9E%9[…]EB%8F%99%EC%B0%A8-%EB%B2%88%ED%98%B8%ED%8C%90-%EC%9D%B8%EC%8B%9D](https://velog.io/@mactto3487/%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-OpenCV-%EC%9E%90%EB%8F%99%EC%B0%A8-%EB%B2%88%ED%98%B8%ED%8C%90-%EC%9D%B8%EC%8B%9D#%EC%9E%90%EB%8F%99%EC%B0%A8-%EB%B2%88%ED%98%B8%ED%8C%90-%EC%9D%B8%EC%8B%9D-with-opencv)  블로그-정책기자단  <https://blog.naver.com/mltmkr/222446342117>  연합 뉴스 기사- **내달 발급 8자리 번호판 카메라 인식률 46%뿐** 2019-08-26 15:08:02  <https://m.yonhapnewstv.co.kr/news/MYH20190826012200038> | | | |
| 데이터 이력 | 작성자/배포자 | 천재교육 빅데이터4기 김창현 | | |

1.2 데이터 정의서



분석 환경:

사용 라이브러리 

Python Version = 3.11.4

Numpy Version = 1.24.3

Opencv-python = 4.8.1.78

matplotlib = 3.7.0

테스트 이미지



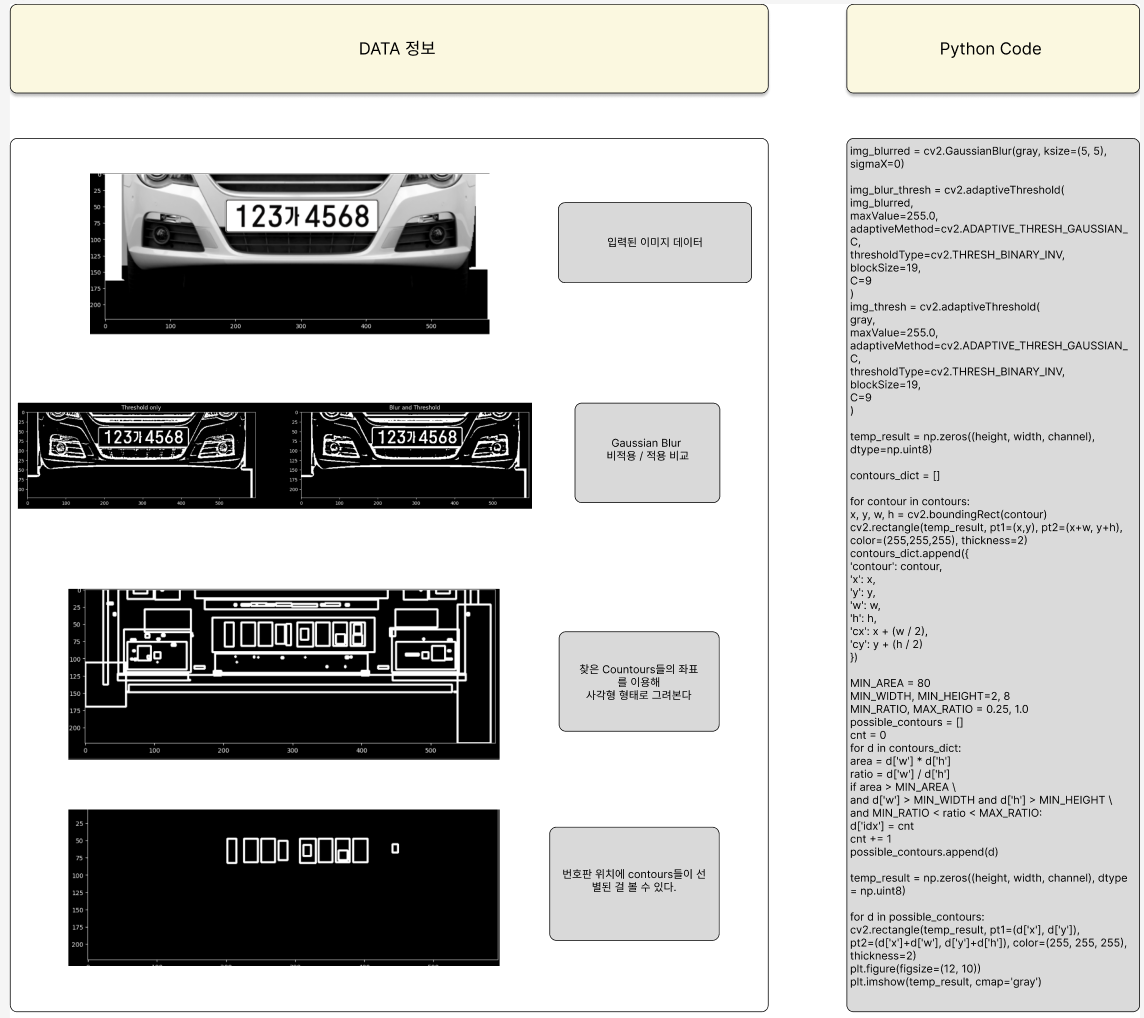
분석 목표:

입력된 이미지 안에서 차량을 인식하고 번호판을 식별하여 번호판 출력하는 모델 적용

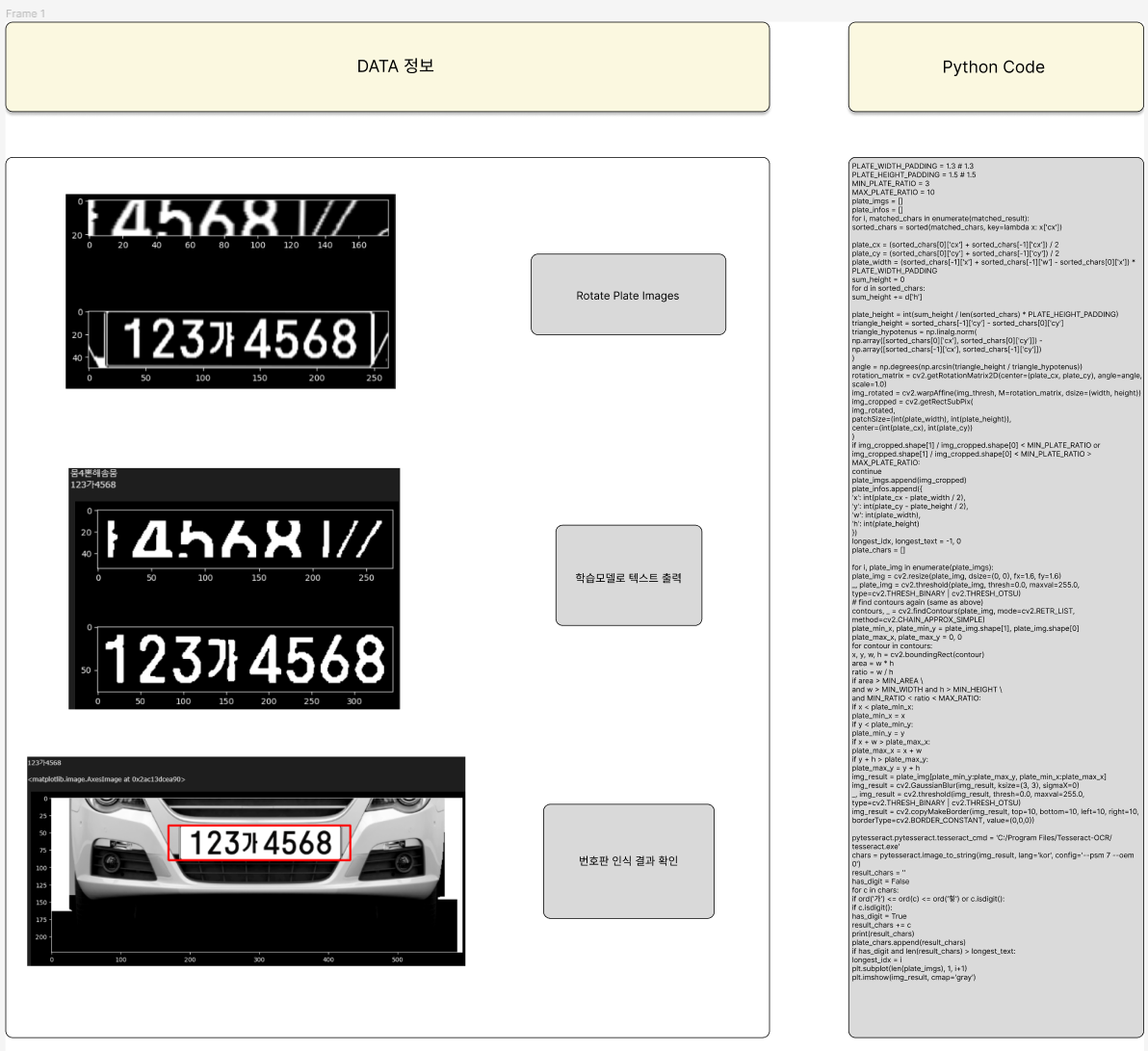


2.1 데이터 개요

2.1.1 모델 예시 데이터 확인



번호판 인식을 위한 Opencv와 tesseract OCR을 이용하여 입력받은 이미지의 번호판구분과 텍스트 출력을 목적으로 하고 모델의 구조를 파악해본다. 입력된 이미지는 HSV의 gray scale을 이용하여 이미지의 강도와 밝기 정보를 가져온 후 이진화만 진행하였을 경우와 GaussianBlur 처리를 같이한 경우를 비교하여 이미지의 주목할 부분을 강조 시킨 후 findContours를 이용하여 물체의 외형을 식별한 다음 이미지 내에서 찾은 Contours의 경계 상자를 그려 시각화 한다.



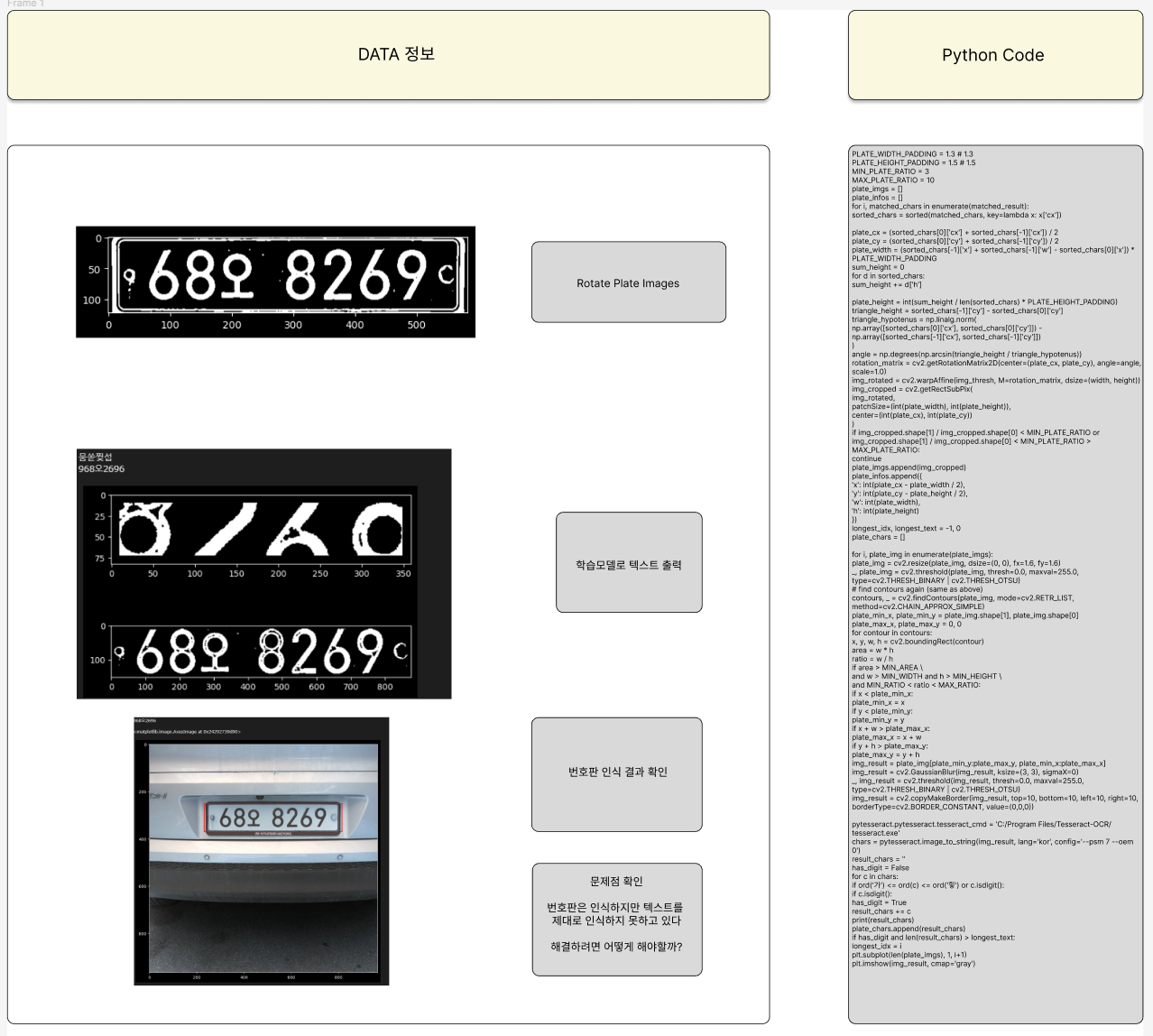
입력되는 이미지 데이터가 기울어지는 경우가 많으므로 pyteseract를 이용하여 번호판을 정방향으로 변환하고

번호판부분만 Crop하여 teseract를 이용하여 text로 전환 후 출력

예제 데이터 결과 :

입력된 이미지의 번호판부분 구별과 구별된 번호판의 Text 출력

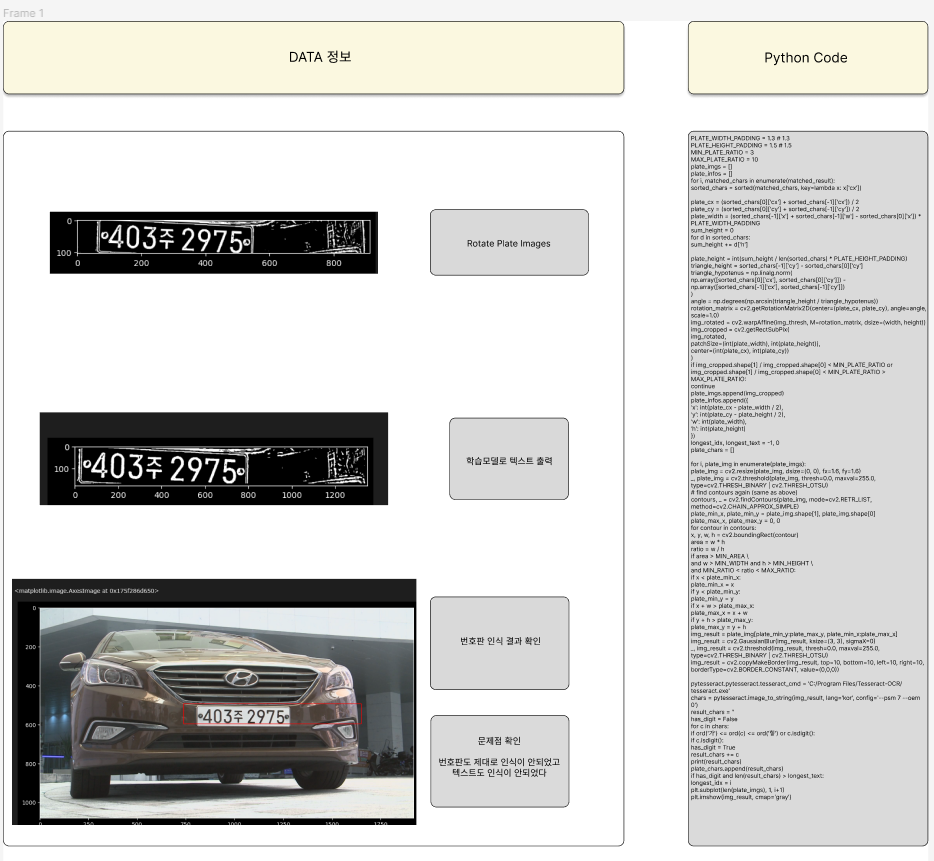
2.2.1 테스트 데이터



2.2.1.1 테스트데이터 문제점 확인

번호판은 제대로 인식이 되지만 텍스트로 제대로 출력되지 않는 문제 확인

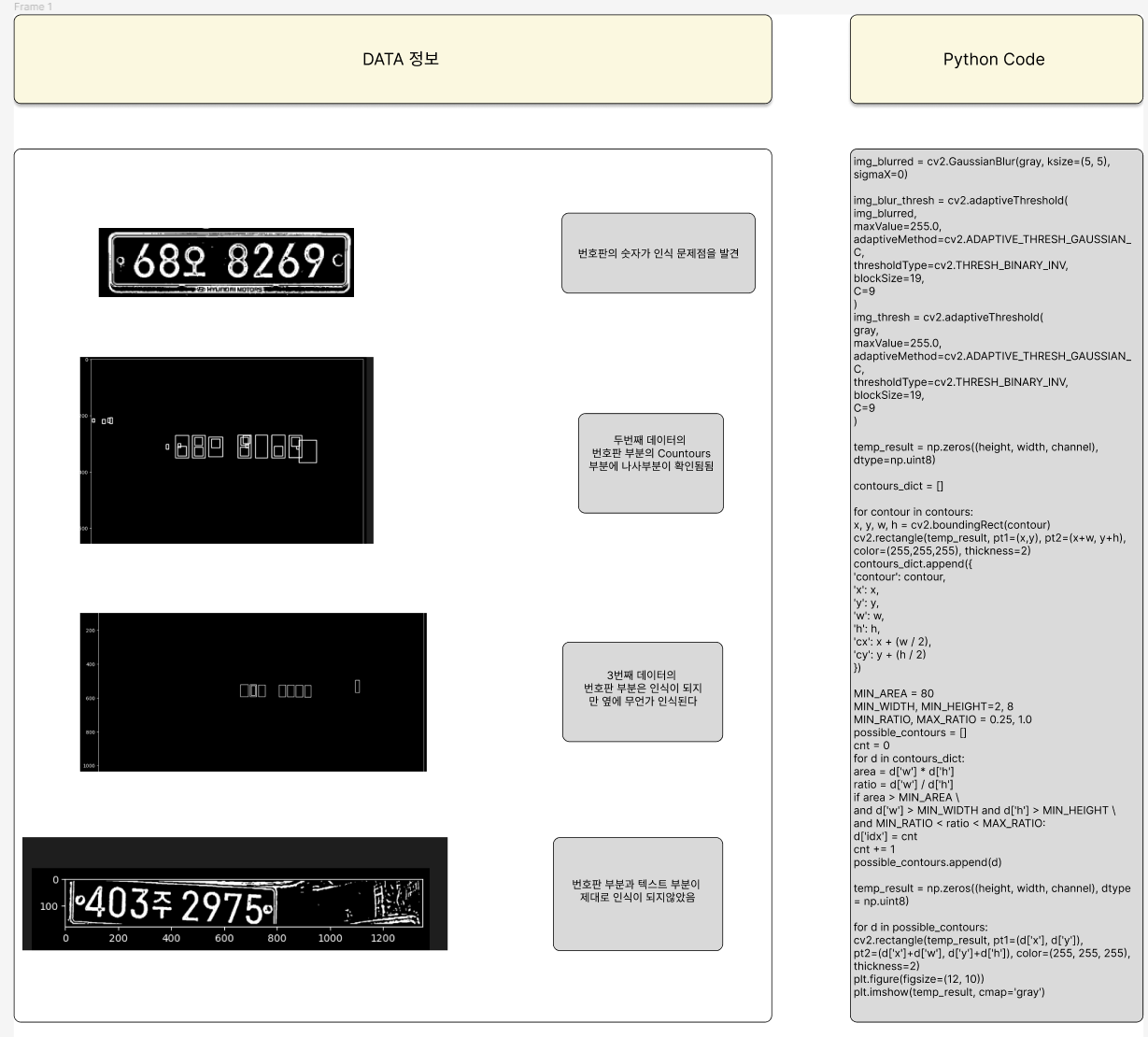
2.2.2 두번째 테스트 데이터



2.2.2.1 두번째 테스트데이터 문제점 확인

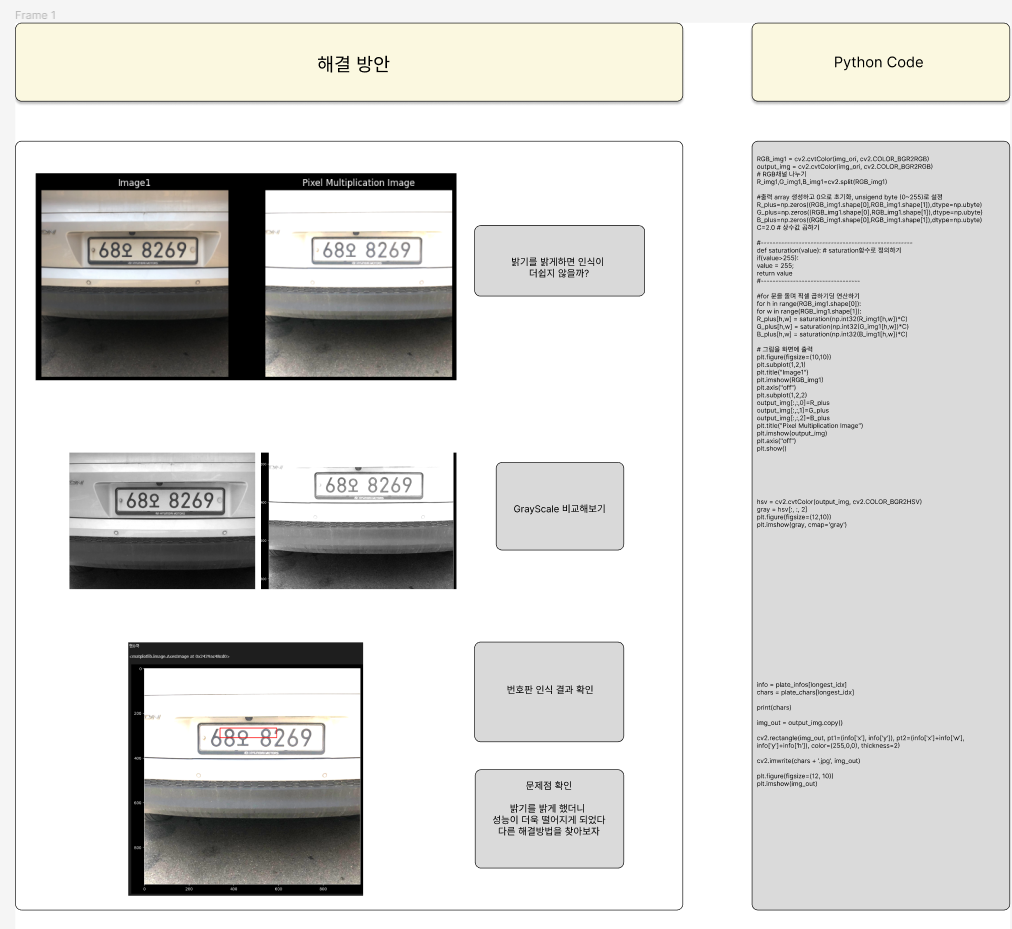
* 번호판이 제대로 인식되지 않고 Text도 인식이 되지않는 문제 확인

2.3 문제점 파악



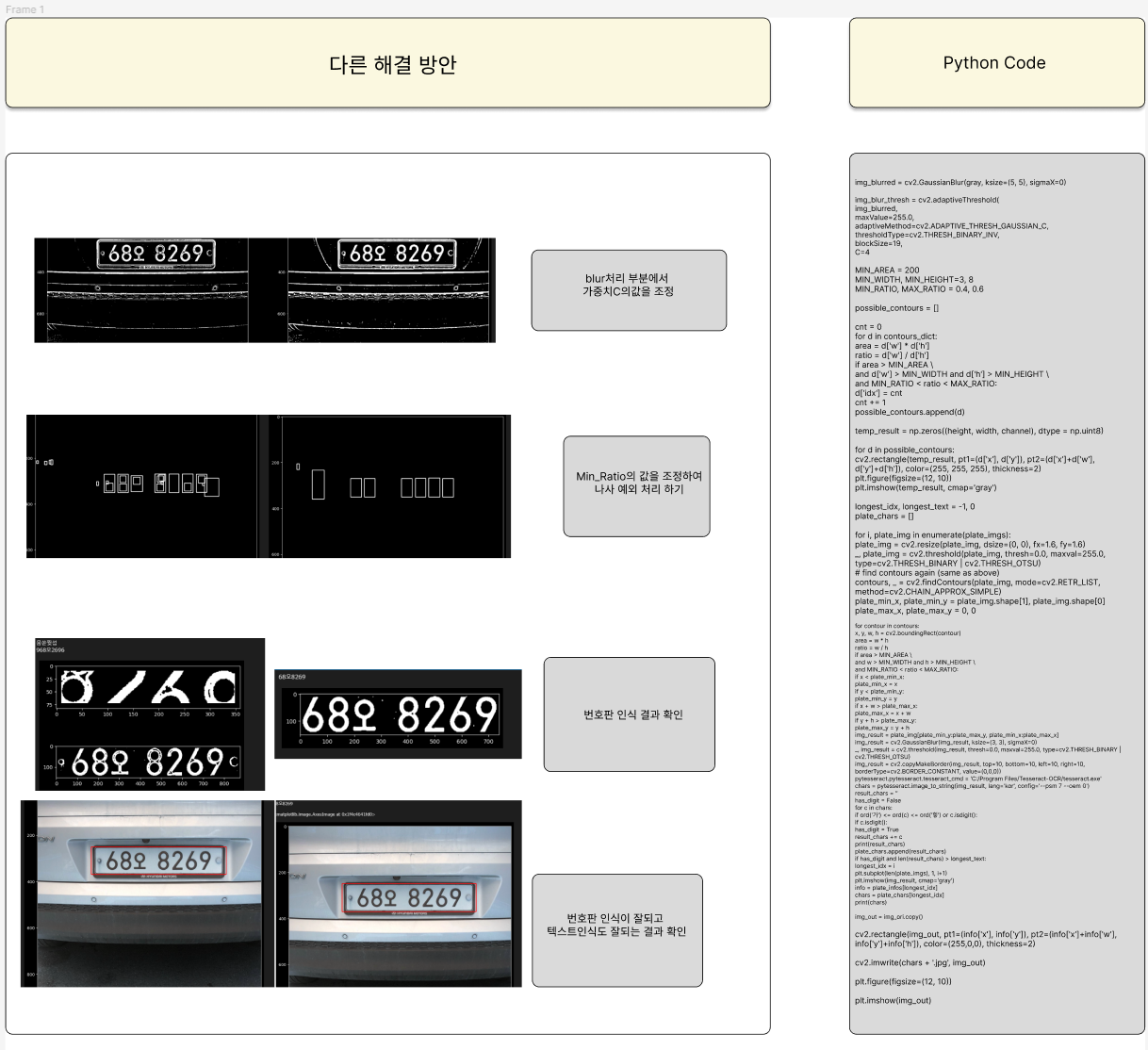
번호판의 나사부분과 빛이 반사되는 부분이 번호로 인식되는 것을 확인

2.4 문제 해결방안 모색



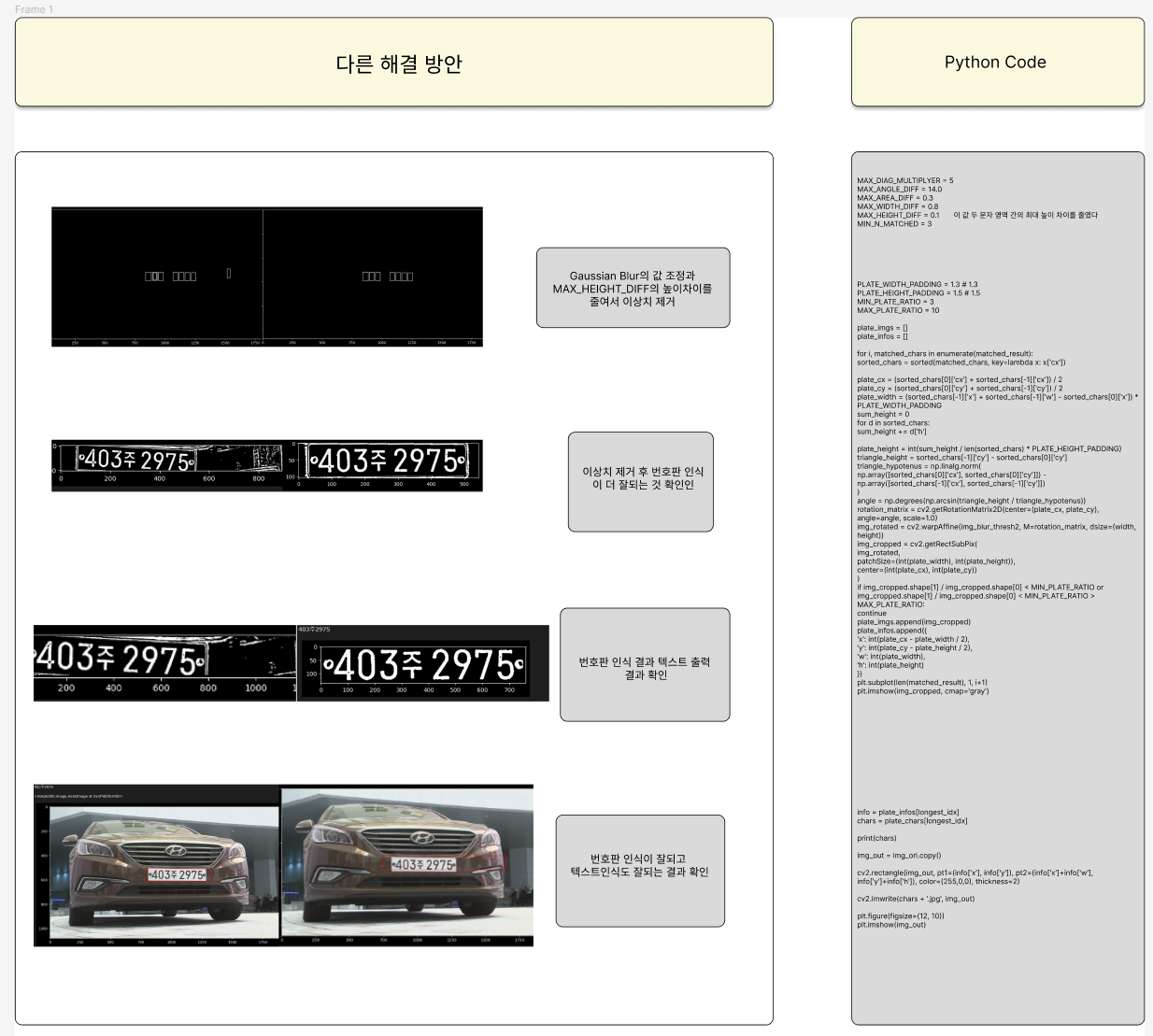
처음 모델에 적용되어있는 것보다 안좋은 결과가 나오는 것을 확인하여 다른방법 모색

2.5 문제 해결



blur처리부분에서 가중치 C의값을 조정하여 더선명하게 이진화가 되는 것을 확인 영역의 최소 면적을 나타내는 MIN\_AREA를 조정하여 면적보다 작은 값을 제거하여 나사부분이 인식이 안되게 조정

2.5.2 문제해결 (2번째 데이터)



영역간의 최대 높이 차이 비율을 조정하여 영역들이 같다고 판단되는 부분을 확인하여

Contours의 영역비율 0.2>0.1로 조정하여 이상치 제거

3. 결론

기존 모델에 테스트 데이터를 확인 해본 결과 기존모델에 조금 변수 값을 조정하여 이상치를 제거하였지만 더 많은 테스트 데이터를 수집하여 모델에 대입해보고 적용되는지 확인 후 다른 모델도 적용해보고 비교해 보아야 할 것 같고 데이터를 어디서 수집할지 JSON파일은 어떤구조로 저장할지에 대한 고민이 더 필요한 것 같다.