기술문서(백엔드)

1. **서버**

GCP를 통해 vm 인스턴스를 생성하여 클라우드 기반 서버를 구축하였음.

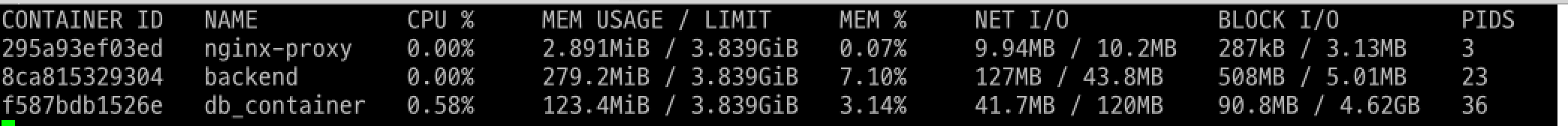
Vm 인스턴스가 오류가 발생할 경우 빠르게 서버를 재 구축하기 쉽게 도커를 사용하여 이식성을 높여주었음.

* Node.js, nginx, mongodb를 docker-compose를 이용하여 연동하여 서버를 배포함.

텍스트이(가) 표시된 사진

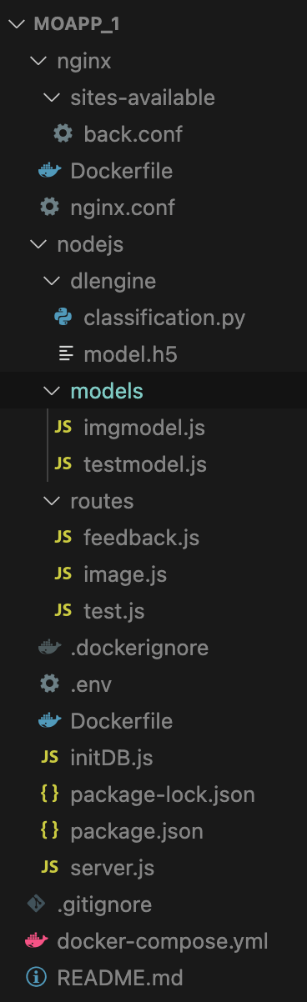
자동 생성된 설명

<docker-compos.yml>



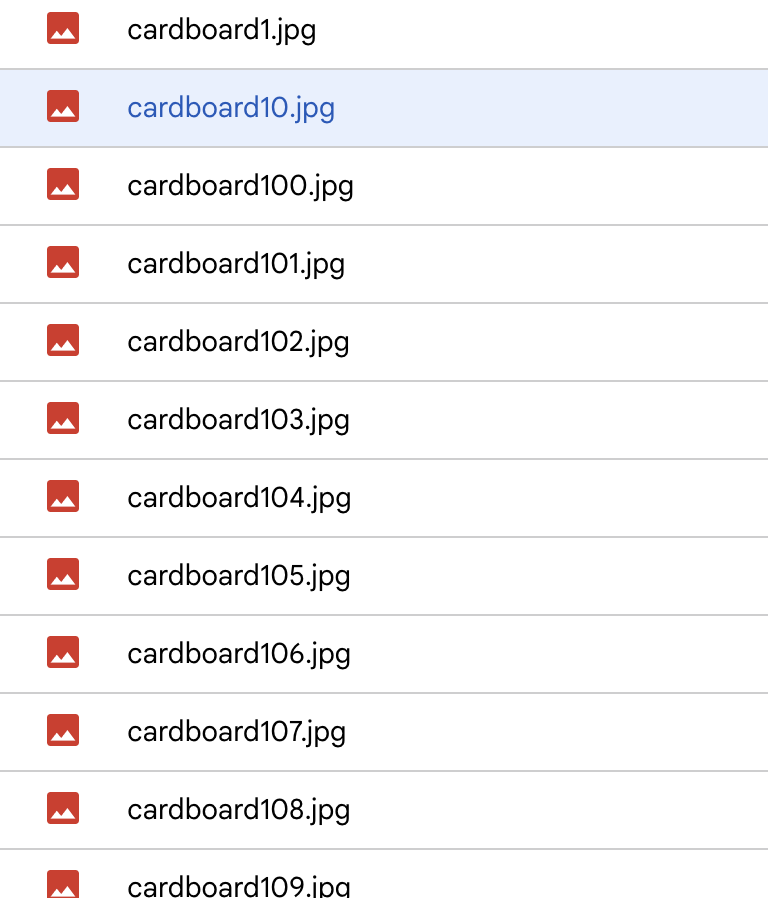
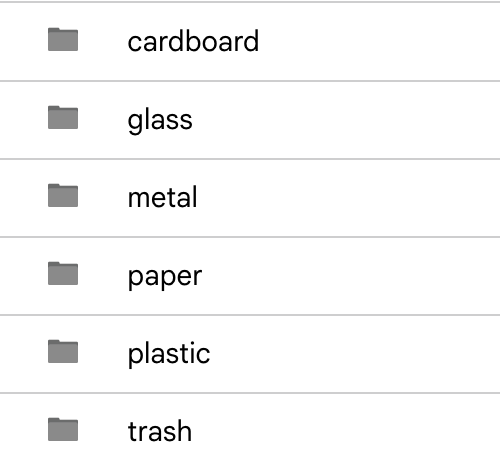
<현재 사용중인 도커 컨테이너>

* 서버 프로젝트 구조도

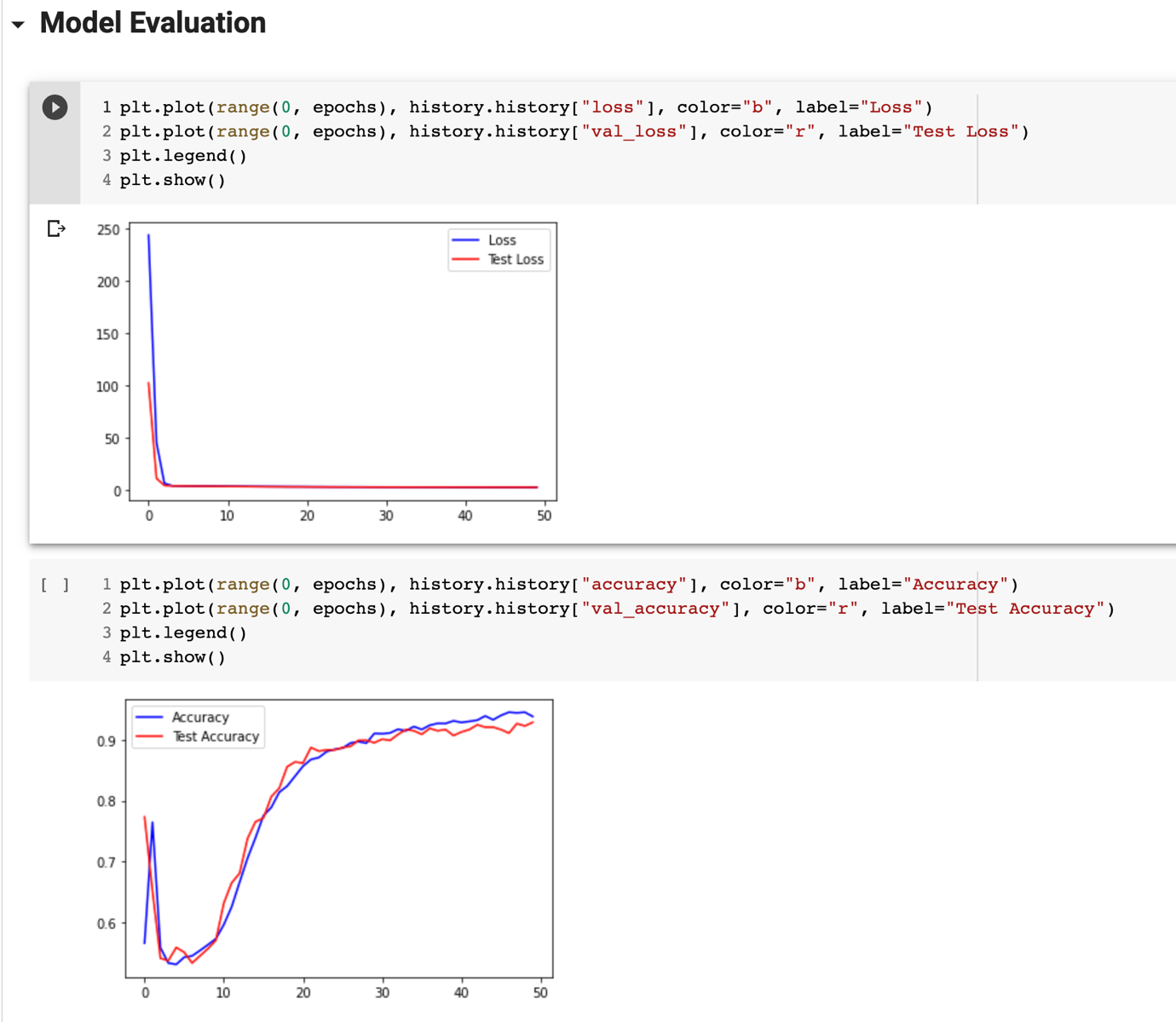


1. **딥러닝 모델**

Python Keras 라이브러리를 활용하여 재활용 쓰레기의 이미지들을 딥러닝 엔진에 학습시켜 재활용 쓰레기들의 이미지를 입력하면 그것이 어떤 분류의 쓰레기에 해당하는 지 판별할 수 있도록 하였음.

텍스트이(가) 표시된 사진

자동 생성된 설명

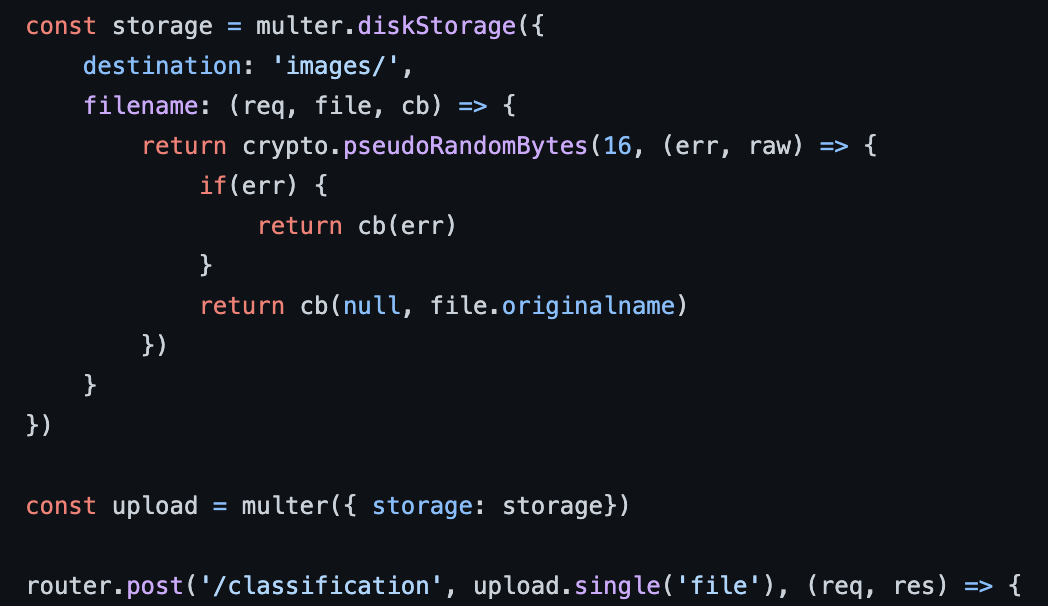


텍스트이(가) 표시된 사진

자동 생성된 설명

1. **Api**

Multer 모듈을 이용하여 서버로 전송된 이미지를 서버의 지정된 경로에 저장하여 python 딥러닝 엔진에 입력할 수 있도록 하였음.



<nodejs/routes/image.js>

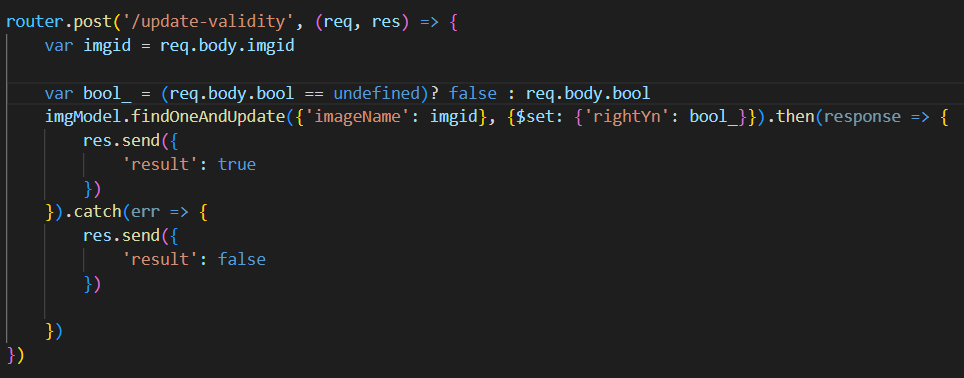
/classification 경로에 이미지 파일을 post형식으로 전송하면 spawn으로 자식프로세스를 생성하여 그 이미지 파일을 딥러닝 엔진에 입력하여 그 결과를 데이터베이스에 저장하고 동시에 그 결과를 response로 보내도록 하였음.

텍스트이(가) 표시된 사진

자동 생성된 설명

<nodejs/routes/image.js>

촬영한 이미지의 분류 결과가 올바르지 않게 산출되었을 때에는 분류가 올바르지 않음을 표지하여 추후 쓰레기 분류 데이터 모델을 재학습시킬 때 활용 할 수 있게 함.



<nodejs/routes/feedback.js>

**데이터베이스**

Mongoose 모듈을 활용하여 MongoDB를 이용해 이미지를 딥러닝 엔진으로 분류한 결과들을 데이터베이스에 저장할 수 있도록 하여 추후 딥러닝 엔진 재학습을 용이하게 하였음.

텍스트이(가) 표시된 사진

자동 생성된 설명

<nodejs/models/imgmodel.js>