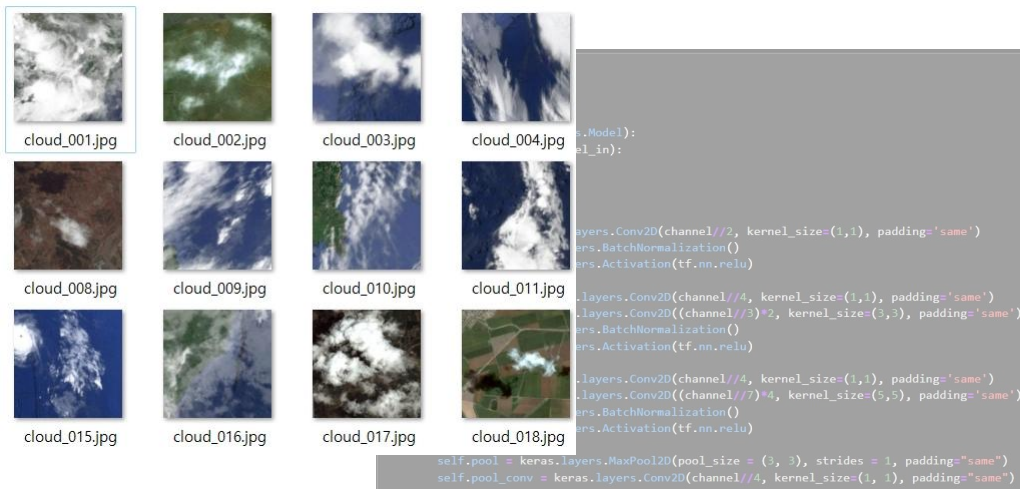


AI를 이용한 항공 사진 구름 판독



2022.05.11
정상택



프로젝트 정보

기간 : 2022.05.06 ~2022.05.11

개발자 : 정상택

프로젝트 목표 : 위성사진에서 구름을 판별

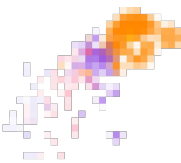
참조 논문 : <https://www.koreascience.or.kr/article/JAKO202104851462472.pdf>

데이터 셋 : NWPU-RESISC45 + 테스트 자료 20장 추가

그림 크기 : 256 x 256

칼라 채널 : 24bit

학습 사진 : 45분류 31,500 장, 이 중에 구름 사진은 700장



프로젝트 목표

프로젝트 목표 : 위성사진에서 구름을 판별한다.

추가 업무 목적 : 캠에서 촬영된 구름을 판별한다.

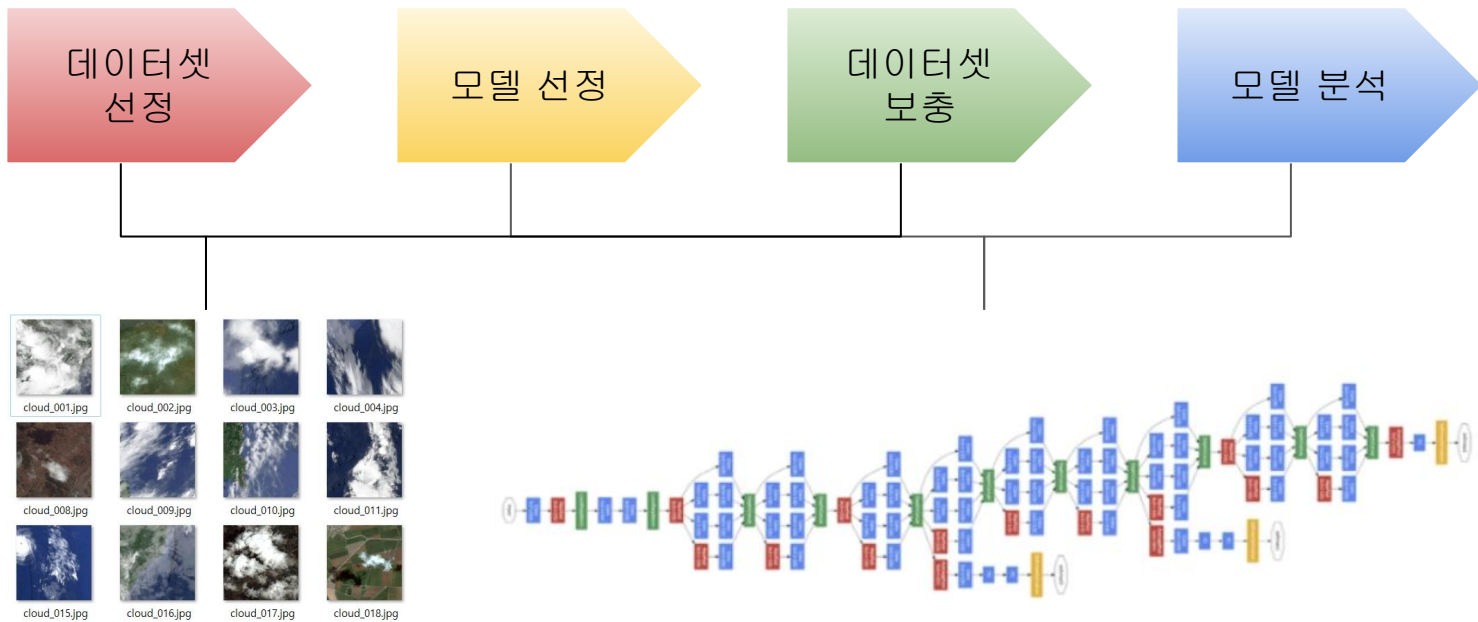
학습 목적1 : 텐서플로우의 **GoogLeNet**을 학습한다.

학습 목적2 : 기존 데이터셋을 변형하거나 추가하여 독자적인 데이터셋을 생성해본다.

학습 목적3 : **GoogLeNet**을 응용하여 새로운 모델을 추가해서 적용해본다.



프로젝트 작업 절차





프로젝트 진행 성과

Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography
Vol. 39, No. 6, 409-418, 2021
<https://doi.org/10.7848/ksgpc.2021.39.6.409>

ISSN 1598-4850(Print)
ISSN 2288-260X(Online)
Original article

딥러닝에 의한 항공사진 구름 분류 및 탐지 비교 실험 Comparative Experiment of Cloud Classification and Detection of Aerial Image by Deep Learning

송준영¹⁾ · 원태연²⁾ · 조수민³⁾ · 어양담⁴⁾ · 박소영⁵⁾ · 신상호⁶⁾ · 박진수⁷⁾ · 김창재⁸⁾

Song, Junyoung · Won, Taeyeon · Jo, Su Min · Eo, Yang Dam
Park, So young · Shin, Sang ho · Park, Jin Sue · Kim, Changjae

이 논문을 참조로 하여, 구름을 판별하는 AI에 도전하였다.
개발 기간은 5일

<https://www.koreascience.or.kr/article/JAKO202104851462472.pdf>



프로젝트 진행 성과

프로젝트 목표 : 위성사진에서 구름을 판별한다.

추가 업무 목적 : 캠에서 촬영된 구름을 판별한다.

학습 목적1 : 텐서플로우의 GoogLeNet을 학습한다.

학습 목적2 : 기존 데이터셋을 변형하거나 추가하여 독자적인 데이터셋을 생성해본다.

학습 목적3 : GoogLeNet을 응용하여 새로운 모델을 추가해서 적용해본다

성공

resics45 데이터 셋에 이미지 추가 및 Custom
텐서플로우 + GoogLeNet 모델을 재구현
opencv와 연동하여 이미지 카테고리 판별



프로젝트 진행 성과

프로젝트 목표 : 위성사진에서 구름을 판별한다.

추가 업무 목적 : 캠에서 촬영된 구름을 판별한다.

학습 목적1 : 텐서플로우의 GoogLeNet을 학습한다.

학습 목적2 : 기존 데이터셋을 변형하거나 추가하여 독자적인 데이터셋을 생성해본다.

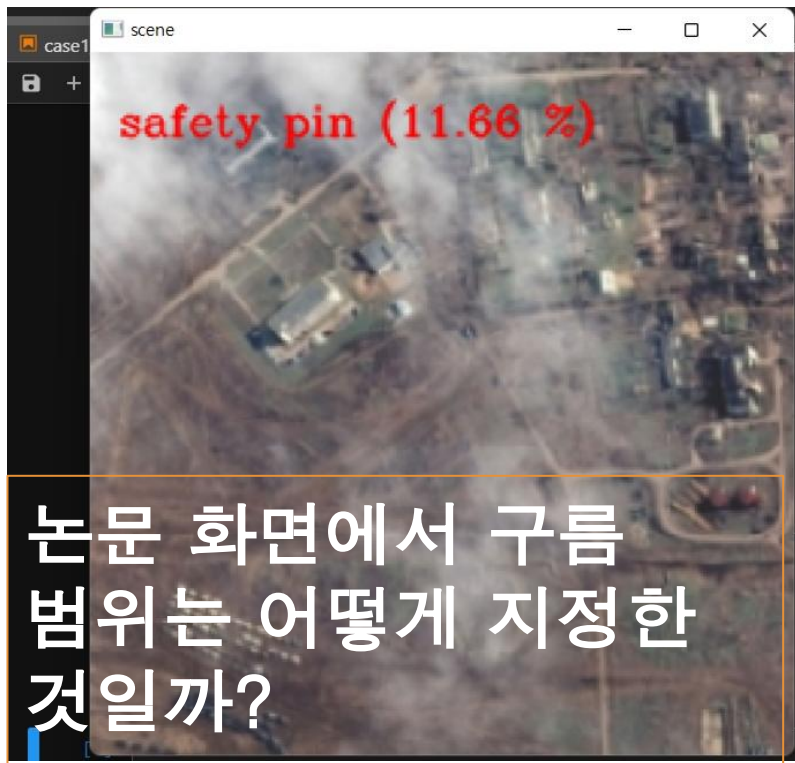
학습 목적3 : GoogLeNet을 응용하여 새로운 모델을 추가해서 적용해본다

실패

캠에 촬영된 영상으로 구름을 판별을 연동하지 못했다.

- OpenCV의 얼굴 판별에는 판별된 x, y, w, h 값이 들어오는데, GoogLeNet 및 Dense에는 판별된 이미지의 `ractange`을 반환하는 요소가 없음.

프로젝트 추가 과제



!=

분명 논문에서는 GoogLeNet도 해봤다고 했는데...

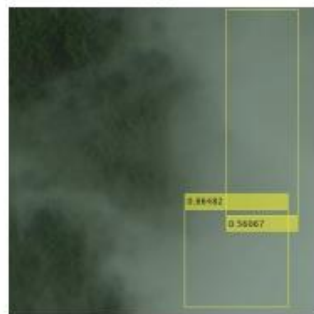


Fig. 1. Object detection in experimental images

(a) Object detection data labeling example,

(b) Examples of detection results

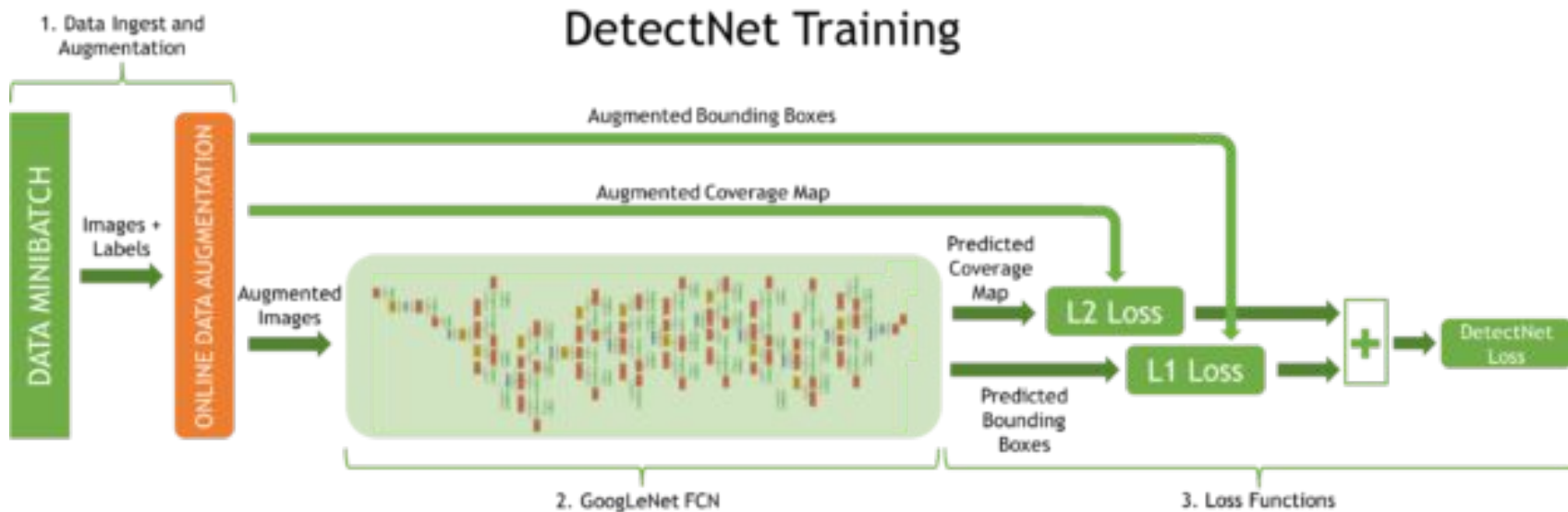


프로젝트 추가 과제

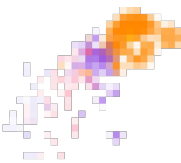


DetectNet이 이런 식으로 최종적으로 구름을 판별할 수 있도록 지원한다고 한다.

프로젝트 추가 과제



가능하다고 하지만 실제로 적용을 해보면 과연 어떤지, 확인이 필요하다.



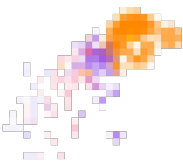
아쉬운 점

데이터 셋을 Custom하는 방법을 몰라서 3일을 소모한 것이 너무 뼈아프다.

구글이나 블로그 등에 해당 내용이 거의 없으며, 처음 접하는 사람이 이해하기 어렵다.

Tensorflow의 가이드는 생략되어 있는 개념과 과정이 너무 많아 실질적으로 도움이 되지 않는다.

데이터 셋마다 들어오는 데이터 구조가 달라서 같은 모델을 적용하기 어려운 것도 많았다.



아쉬운 점

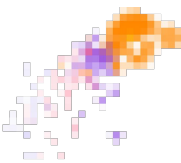
데이터 셋마다 라벨링하는 방법도 다양했다.

- matlab과 xml, json을 사용한 방식
- file folder를 사용하는 방식
- 압축파일 이름을 사용하는 방식
- binary header를 이용하는 방식
- 그 외 솔루션과 config 파일을 이용하는 방식 등등...

들어오는 image shape도 너무 다르다.

Densor에 필요한 디멘션이 없는 경우도 있음

Test, Train이 설정되거나 분리되지 않는 경우도 많음 심지어 태그도 다르다.



아쉬운 점

이미지를 다루는 학습은 시간과 하드웨어 사양을 충분하게 확보하기 어렵다.

시중에 나와있는 모델과 동떨어진 학습 목표가 있다면 개발 시간과 학습시간을 충분히 고려하여 확보할 것.

256 * 256 크기의 이미지를 기준으로 **31500**건을 1번 학습하는데 무려 **10시간** 이상이 걸린다. (**8GB RAM, AMD Ryzen 5 5500U with Radeon Graphics**, 그래픽 카드 없음)

이미지 처리를 할 때에는 **CUDA** 적용, 워크스테이션 임대 또는 **COLAB** 유료로 고려했어야 했다.

아쉬운 점

Progress: | 100.0% Complete

Epoch 1, Loss: 3.809382200241089, Accuracy: 2.019047498703003

Progress: | 5.1% Complete

```
model.summary()
```



CPU
83% 3.77GHz



메모리
5.8/6.8GB (85%)



디스크 0(C:)
SSD
0%



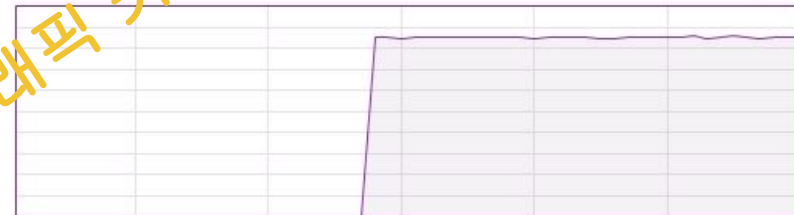
이더넷
VirtualBox Host-Only N
S: 0 R: 0 Kbps

메모리

8.0GB

메모리 사용

6.8GB



60초

0

메모리 구성

