

UNDER STANDING CLOUDS FROM IMAGES

주제 설명

기후변화는 다양한 의사 결정 과정에서 중요한 역할을 하고 있다. 구름은 기후변화를 예측하는 데에 있어서 중요한 역할을 하는데, 형태에 대한 이해가 어렵고 구름 종류 간의 경계가 모호해 구름을 활용한 기후 모델을 만드는 것은 어려운 일이 되고 있다. 이에 따라 과학자들은 위성영상을 통해 구름을 4 가지 타입(Flower, Fish, Gravel, Sugar)으로 분류했고, 기계 학습을 통해 정확성이 높은 구름 분류 모델을 만들 수 있다면 앞으로 기후 예측의 정확성을 높일 수 있을 것으로 예상하고 있다. 따라서 이 과제에서는 구름의 위성영상을 입력으로 받았을 때, 최대한 정확하게 해당 영상의 구름을 알맞게 분류하는 모델을 만드는 것을 목표로 한다

조 TEAM 설명

조이름: KMUAI_CAT

곽홍재(20172813)

권민수(20142697)

수행한 업무 분장 (기여도 %) : 50 : 50

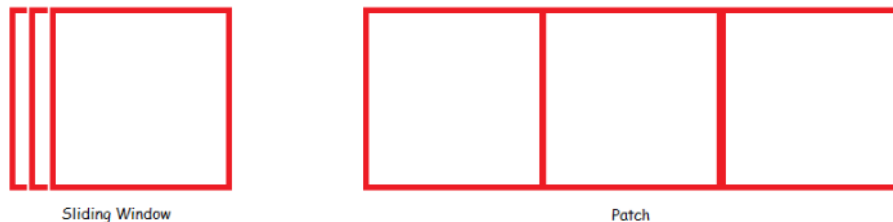
적용한 방법 : U-NET + POST-PROCESSING

i. 세부 설명

U-Net 은 네트워크 형태가 알파벳 U 와 비슷하게 생긴 신경망이다

기존 Segmentation Network 들의 문제점을 크게 두 가지로 나뉘었을 때 느린 속도의 문제, Localization - Context 간의 Trade-off 문제가 있었다.

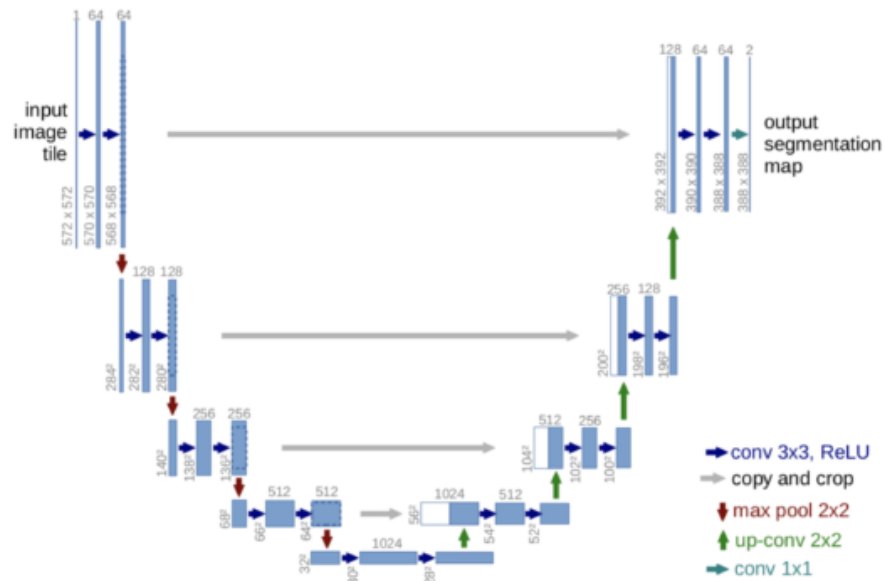
먼저 속도의 경우는 Sliding Window 를 사용할 시 이미 사용한 구역을 다음 Window 에서 다시 검증하는 과정을 거쳤기 때문에 똑 같은 일을 반복하는 비효율을 보였다. 이를 해결하기 위해 U-Net 은 이미 검증이 끝난 부분은 아예 건너뛰고 다음 Patch 부분부터 검증을 진행해 속도가 빠르다.



다음으로 Trade-off 문제는 Patch-Size 의 문제라고 볼 수 있는데, 이 Patch Size 를 너무 크게 하면 Context 인식이 유리하지만 Localization 에 불리해지고, 너무 작게 하면 Localization 은 유리하지만 Context 인식이 불리해지는 Trade-Off 문제가 있었다. 이를 해결하기 위해 U-Net 은 여러 Layer 의 Output 을 동시에 검증해 Localization 과 Context 인식 두 마리 토끼를 모두 다 잡았다.

추가적으로 U-Net 은 Image Mirroring 기법을 사용한다. Contracting Path 에서 이미지의 외곽부분이 잘리는 것이 누적되어 output 의 사이즈가 작아지는 문제가 있었는데, 없어지는 부분을 Zero-padding 하는 대신 안쪽 이미지를 거울에 반사시킨 형태인 Mirror Padding 기법을 사용해 사이즈를 유지한다.

U-Net 의 Architecture 는 다음과 같다



- 컨볼루션 층 : 3x3 Convolution 사용, 활성화함수 = ReLU
- 풀링 층 : 2x2 MAX Pooling, 매 층마다 1/2 다운샘플링, 채널은 2 배
- Expanding Path(이미지 크기를 키우는 부분) : 2x2 UP Convolution
- Mirror Padding 에 대한 보상 :
Contracting Path 의 데이터를 적당한 크기로 자른 후, Concat 하는
방식으로 보강

추가적으로 Output 에 대한 Post-processing 을 진행했다. Post Processing 은 Precision - Recall Curve 를 이용해 라벨 각각의 최적의 Recall Threshold 를 구해서, 해당 Threshold 값을 통과한 Output 만 값이 유효한 것으로 판단하여 EncodedPixels 값을 채워서 제출하도록 했다.

여기서 Precision 은 정확도(실제 물체가 얼마나 포함되어 있는지)를 의미하고, Recall 은 검출율(대상 물체를 빠트리지 않고 얼마나 잘 잡아내는지)를 의미하며, 일반적으로 둘은 반비례 관계이기 때문에 Graph 를 통해 최적의 값을 찾아내는 과정이 필요하다.

ii. 관련 연구 동향 및 배경 지식

Segmentation

이미지를 찾아서 박스를 치는 Detection 과는 달리 이미지를 픽셀단위로 분류하는 기법이다. 즉, 각 픽셀이 어느 클래스에 포함될 확률이 높은 지 판별해내는 기법이라고 할 수 있다.

Segmentation 은 Semantic Segmentation 과 instance Segmentation 두가지로 구분할 수 있다.

Semantic Segmentation 은 클래스만 구분하는 기법이고 instance Segmentation 은 같은 클래스라도 다른 것을 분류한다.

우리 과제에서 사용하는 것은 Semantic Segmentation 이다.

Semantic Segmentation 은 인코더와 디코더 형식과 비슷하다.

인코더에서는 CNN 을 통해 Feature map 을 출력하고 이미지의 사이즈를 줄인다.

사이즈가 줄어들고 차원이 달라진 상황에서는 Segmentation 을 할 수 없기 때문에 Up-sampling 기법을 통해서 Feature map 을 다시 원 공간의 크기로 바꾼다.

대표적인 알고리즘으로는 FCN, deep lab, U-Net 등이 있다.

iii. 상세한 구현 환경, 구현 방법

- 캐글 커널, Google CoLabatory, Local 에서 진행
- Tensorflow(keras 2.2.4-tf), keras 2.3.1 기반으로 모델링, segmentation 모델은 텐서플로우와 케라스 기반으로 만들어진 Segmentation models 패키지 사용
- post-processing 에는 EfficientNet 을 사용(텐서플로우와 케라스 기반)
- 기타 openCV2, numpy, pandas, sklearn, seaborn, matplotlib 패키지 사용

다양한 Discussion 과 notebook 에서 성능이 높은 사람들이 처음에는 Segmentation models 패키지를 통해 Semantic Segmentation 을 하고 이미 검출된 클래스를 보정하는 post-processing 을 적용한다는 사실을 알아냈다.

캐글을 참조해서 Segmentation models 패키지를 통해 기본 모델을 만들고 post-processing 을 적용하는 방향으로 구현함

1. Segmentation models의 Unet 모듈에서 resnet34 encoder 구조를 가져와서 학습시킨다

(class 의 개수와 마지막 층의 activation, data generator 를 통해 들어오는 이미지의 shape 를 추가로 지정해준다.)

2. 예측 값에 적용하고 1차 submission을 만든다.
3. EfficientNet에서 pre-train된 Imagenet가중치를 사용해서 모델을 학습시킨 뒤 클래스별로 threshold값을 구한다.
4. 구한 threshold값을 통해 1차 submission을 수정하여 최종 submission을 만들어 낸다.

iv. 코드 상세 설명

1 차 서브미션 : U-Net+ResNet.pdf

2 차 서브미션 : EfficientNet.pdf

결과

제출한 모든 결과들을 출력해서 화면

12 submissions for KMUAI_CAT		Sort by Most recent	
<div>All Successful Selected</div>			
Submission and Description	Private Score	Public Score	Use for Final Score
new4.csv a day ago by 곽홍재 CAM-post-processing	0.60055	0.60295	<input type="checkbox"/>
new3.csv a day ago by 곽홍재 add submission details	0.63133	0.63019	<input type="checkbox"/>
Understanding Clouds - EDA and Keras U-Net 91090c (version 2/2) a day ago by 곽홍재 From "Understanding Clouds - EDA and Keras U-Net 91090c" Script	0.62639	0.62341	<input type="checkbox"/>
new2.csv a day ago by 곽홍재 post-processing submission	0.60777	0.60499	<input type="checkbox"/>
Understanding Clouds - EDA and Keras U-Net 91090c (version 2/2) a day ago by 곽홍재 From "Understanding Clouds - EDA and Keras U-Net 91090c" Script	0.58968	0.58702	<input type="checkbox"/>
submission.csv a day ago by KMUmskwon cdeotte - CAM + single threshold	0.59540	0.59476	<input type="checkbox"/>
submission.csv a day ago by KMUmskwon cdeotte - CAM + thresholding	0.60086	0.60287	<input type="checkbox"/>
submission.csv a day ago by KMUmskwon cdeotte - CAM	0.27459	0.27048	<input type="checkbox"/>
new.csv 2 days ago by 곽홍재 add submission details	0.60080	0.61012	<input type="checkbox"/>
new.csv 2 days ago by 곽홍재 add submission details	0.58396	0.59208	<input type="checkbox"/>
Satellite Clouds: U-Net with ResNet Encoder (version 1/1) 2 days ago by 곽홍재 From "Satellite Clouds: U-Net with ResNet Encoder" Script	0.58396	0.59208	<input type="checkbox"/>

결론

조사를 하는 과정에서 U-Net 이 Localization 과 Context Tradeoff 를 해결한 방식이 놀라웠다. Zero base 에서 신경망을 짜는 것은 힘들 것 같아서 다른 모델들과 코드를 많이 참고했는데, 시간이 된다면 아예 zero base 에서 시작해보고 싶다.