**딥 러닝을 이용한 차량 유형 인식**

팀원 : 2012104085 박성철

**개 요**

딥 러닝을 이용하여 해당 객체가 차량임을 인식함과 동시에 해당 차량이 어떠한 차량인지 인지하는 모델을 만들고자 한다. 이는 기존의 수작업을 통해서 이루어지던 차량 대수와 유형 검출을 딥 러닝을 이용하여 자동화한다는 목적에 있다. 또한, 딥 러닝을 통해 특성에 따라 차량의 검출이 가능해지면 자율 주행 차량에서 운전 방침 판단 시에 위험한 화물 차량에서 멀어지거나 응급 차량에게 비켜주는 등의 도움이 될 수 있을 것이라 기대한다. 본 보고서에서는 차량의 유형 중 크기에 초점을 맞춰서 인식시켜보았다. Mask R-CNN[1] 모델을 사용했으며 데이터셋은 Kaggle의 Stanford Car Dataset by classes folder[2]의 이미지를 이용했다. Model train 별 test 결과를 빠르게 보고자 train dataset은 659장, validation 60장을 사용했으며, epoch와 각 epoch 별 step을 적은 수로 하여 train을 한 결과 차량의 위치는 bounding box로 정확하게 체크하는 것을 확인했다. 하지만 해당 차량이 어떠한 크기를 가졌는지에 대해서는 SUV와 같은 큰 차량과 스포츠카와 같은 소형 차량에 대해서는 정확히 구별했으나 2문 차량과 4문 차량의 경우 5:5 혹은 6:4 정도로 정확하게 구별하지 못했다. 이에 대해서는 더 많은 이미지 dataset을 사용하고 hyperparameter를 더 증가시키면 정확도가 높아질 것이라 예상하고 있다.

1. **서론** 
   1. **연구 배경**

오늘 날의 시대는 4차 산업혁명이라 불리우는 유행에 탑승하고 있다. 5G, IoT, Blockchain 등 수 많은 기술들이 떠오르는 가운데 무인 자동차 기술도 포함되어 있다. 무인 자동차가 통용되기 위해서는 기반이 될 많은 기술들이 필요하지만 그 중에서도 충돌 사고를 일으키지 않기 위해 사람과 사물 등을 확인하는 객체 인식이 중요시된다. 이러한 차량 인식은 4차 산업혁명 이전부터 이어져왔다. 영상 처리, 컴퓨터 비전 기술이 발전해가면서 처음에는 번호판 인식으로 시작한 기술이 차량 인식으로 넘어갔다. 하지만 논문을 찾아볼 때, 아직 차량 유형에 대한 분류는 원활하게 연구되지 않는다. 차량 유형에 대한 분류 인식이 효과적으로 이루어진다면 얻을 수 있는 이점은 무인 자동차 기술과도 이어진다고 본다. 무인 운전 알고리즘을 짤 때, 차량의 각 유형을 인식해서 트럭이나 공사 차량임을 확인한다면 사전에 위험도를 높여 대응할 수 있을 것이다. 또한 해당 지역을 진입하는 차종의 가치를 판단하여 상업적인 목적 등으로 사용할 수도 있을 것이다. 위에서 언급했듯이 차량 인식은 이미 영상 처리, 또는 머신 러닝 등의 분야에서 많은 연구가 존재한다. 그렇기에 차량 인식 뿐 아니라 차량 유형 인식이 가능한 모델을 만들고 이 모델의 정확도를 높여보면 좋겠다는 생각을 가졌다.

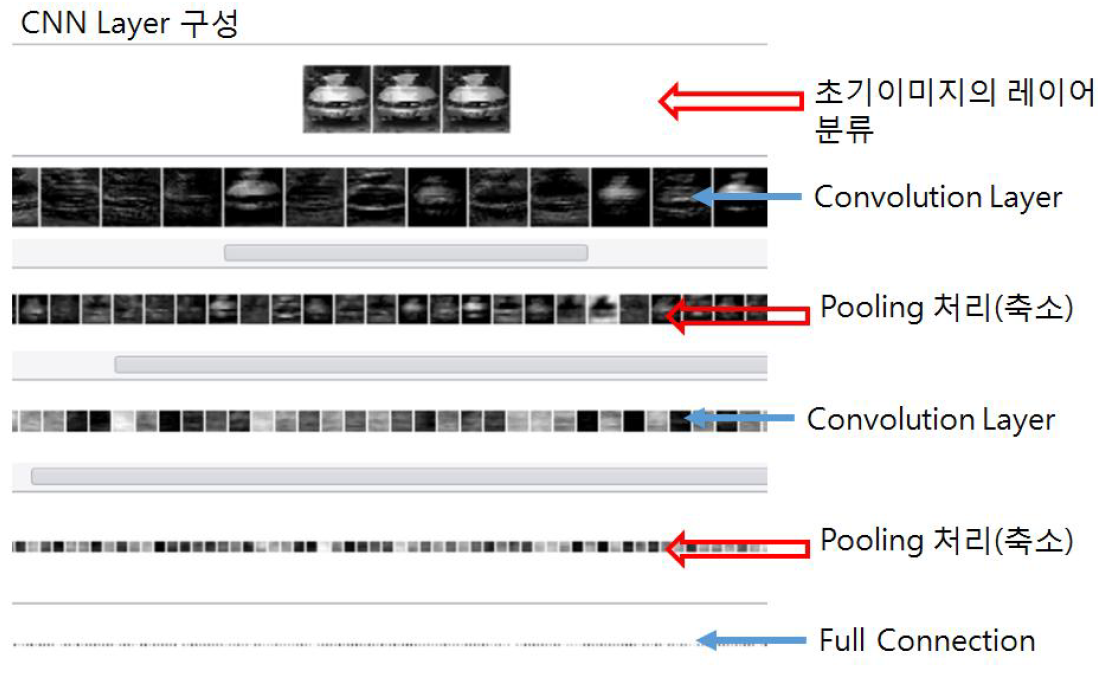
* 1. **연구목표**

이미지에서 차량을 인식한 다음 해당 차량이 어떠한 차량인지 분류하는 것을 목표로 한다. 우선 많은 유형 중 크기로 인한 차량 분류가 이루어지면 차종의 분류로 결과의 정확도를 높이는 것이 연구 목표이다.

1. **기존 연구**
   1. **Convolutional Neural Network을 이용한 차량 인식**[3]

해당 기존 연구는 차량을 인식하게 되면 차량의 종류, 차량 번호판 인식 등의 다음 업무가 진행 될 수 있을 것으로 차량 인식을 중요하게 판단하여 시작되었다. 기존의 차량 번호판의 인식만 해도 차량 검색을 못하고 번호판만을 검색하기 위한 알고리즘을 사용하는 경우에 인식률이 많이 떨어지는 단점을 보완하기 위해서기도 하다.

차량이나 물체를 인식하기 위한 방법은 여러 가지가 존재하며 그 방법에 대해 살펴보면 Scale Invariant Feature Transform을 이용한 방법, 색상을 이용한 방법, Haar-like를 이용한 방법 등이 있다. 이 연구에서는 딥러닝 기법인 CNN을 사용했다.

해당 연구에서는 주변의 차량 데이터를 직접 수집하여 Dataset을 만들어 차량 인식을 시도했으며 그 결과를 보고 차량의 전면부와 후면부의 이미지를 구분할 수 있는지에 대한 테스트를 진행했다. 또한 CNN의 신경망 학습법 중에서도 SGD와 SGD Levenberg-Marquardt 방법을 통해 그 결과를 비교해봤다.

( Fig 1. 기존 연구 2.1에서 사용한 CNN의 Layer 구성 )

실험 결과로 전체 700개의 학습 이미지에서 SGD Levenberg-Marquardt 방법으로 학습했을 때 200개의 차량 샘플 중 9개의 오인식을 나타냈고, SGD 방법으로 학습했을 때 12개의 오인식을 나타냈다. 오류가 나타난 원인으로는 데이터셋의 부족한 수로 학습이 제대로 이루어지지 못했다고 파악했으며 CNN을 이용한 차량 인식은 매우 효과적으로 볼 수 있다고 한다.

( Fig 2. 기존 연구 2.1에서 SGD 방법으로 학습했을 때의 오인식 결과 )

* 1. **기존 연구의 문제점**

기존 연구에서는 이미지를 RGB로 분리한 3개의 이미지로 만들어서 2번의 Convolution과 2번의 Pooling Layer를 거쳐서 Classification했는데, 이 기본적인 구성으로도 좋은 인식률을 보여주었지만 채널의 수를 변화시키거나 레이어를 더 추가해주는 것, 혹은 아예 다른 모델을 사용하는 것이 보다 좋은 결과를 낼 수 있을 것이라 본다.

무엇보다 이 연구에서는 데이터셋의 부족이 문제점으로 생각된다. 기존 연구자 또한 적은 데이터셋이 인식률을 낮추었다고 판단한다. 딥 러닝은 학습량에 따라 정확도가 증가할 수 있으므로 더 많은 양의 데이터셋을 마련할 수 있었다면 좋았을 것이다

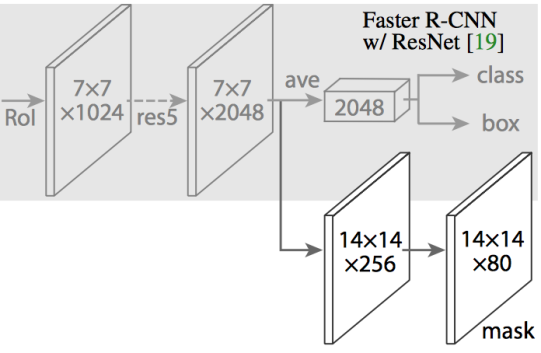
마지막으로 이 연구에서는 단순히 차량의 전면부와 후면부의 인식만을 시도했다. 단순 앞뒤 인식만이 아니라 차량에서도 우리가 더 필요로 할 만한 세부적인 요소나 분류 방향으로 나아가도 될 것으로 본다.

1. **프로젝트** 
   1. **기존 연구와 차이점 및 해결방안**

기존 연구에서는 자연물과 인공물들 중 제시된 물체가 차량인지 인식하는 학습이 이루어졌다. 더불어 차량일 때 이게 전면부인지 후면부인지 확인하는 학습도 이루어졌다. 단순 앞뒤 인식만으로는 실제 사용할 수 있는 범위가 좁으므로 이러한 내용에 해당 객체가 차량일 때 추가적인 요소로 분류 인식하는 것이 기존 연구와의 차이점이다. 이를 위해서는 기존 연구에서 사용한 CNN 모델이 아닌 업그레이드된 모델을 사용하고자 한다. 데이터셋에 대해서도 보다 많은 양의 데이터셋을 구하여 학습할 계획이다. 이를 통해 수 많은 차량의 유형 중에서 우선 크기를 통한 차량 분류 인식을 진행된다. 모델과 데이터셋을 수정해 나가면서 보다 높은 정확도를 맞추려 한다.

* 1. **프로젝트 내용**

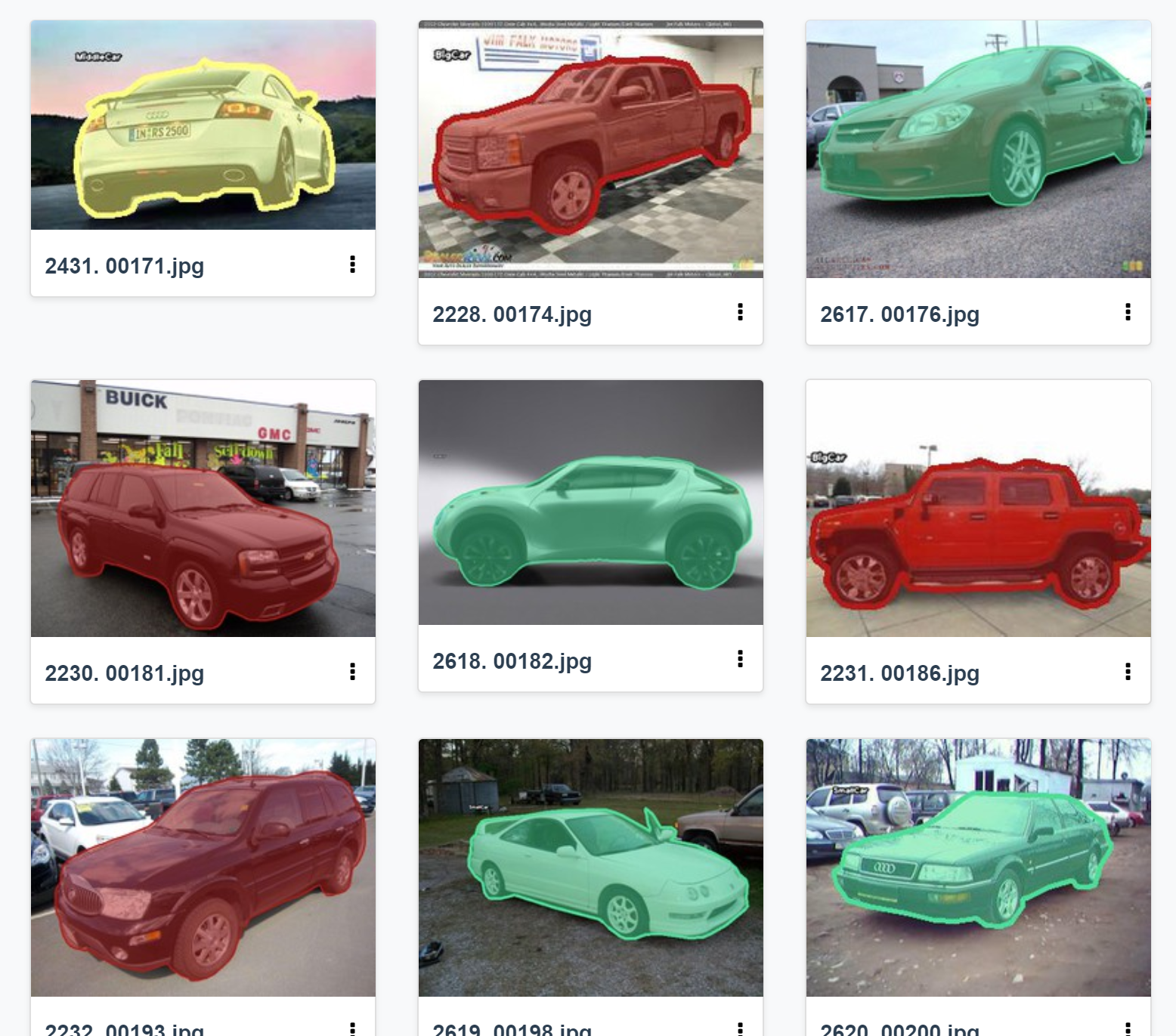
딥러닝 모델 중Mask R-CNN 모델을 통해 차량 유형 인식을 시도한다. Mask R-CNN 모델은 R-CNN 모델에서 Fast, Faster R-CNN모델을 거쳐 만들어진 모델이다. Faster R-CNN 모델까지는 Object detection만을 정확하게 진행한 반면, Mask R-CNN은 이전까지 사용하던 Roi Pooling이 아닌 Roi Align 개념을 사용하여 Object detection만이 아니라 물체의 위치도 그대로 가지고 있음이 특징이다. Roi Pooling을 할 때는 Max pooling으로 이미지 데이터의 특성이 사라져서 위치 정보가 사라졌으나, Roi Align은 bilinear interpolation을 하기에 위치 정보가 남아있는 것이다. 여기에 ResNet을 같이 사용해주는 것이 Mask R-CNN이다.



( Fig 3. Mask R-CNN 모델 )

모델을 학습하기 위한 이미지 데이터셋은 Kaggle의 Stanford Car Dataset by classes folder을 사용했다. 이 데이터셋은 train 데이터 8144장, test 데이터 8041장을 포함하고 있다. 본 연구에서는 빠르게 결과를 보고 반복하기 위해서 train 데이터 659장, validation data 60장으로 간추렸다.

Mask R-CNN 모델을 훈련시키기 위해서 차량 이미지 데이터셋을 준비할 때, 기존 모델에서 성능이 잘 나왔던 COCO dataset 형태로 만들어 준다. COCO dataset 형태로 만들기 위해서 모든 이미지 데이터 각각에게 annotation 데이터를 만들어준다. 분류할 Class에 대해서 우선 3가지 size로 category화 한다. size는 Big, Middle, Small로 준비했으며, Big에 대해선 트럭과 SUV, Middle은 세단과 4문 차량, 그리고 Small은 소형 4문 차량과 2문 차량, 스포츠카로 분류했다.



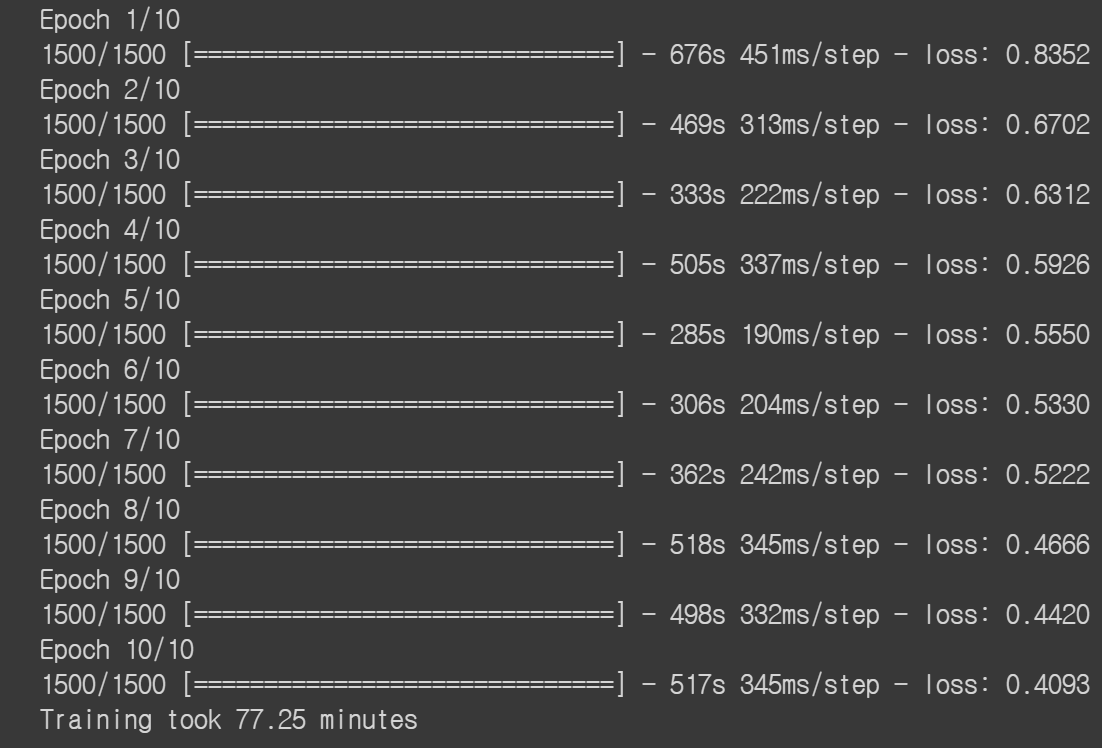
( Fig 4. COCO Dataset을 위한 이미지 별 Annotation 데이터 )

작업 환경은 Google Colab에서 Tensorflow를 이용했다. 기본 Mask R-CNN 모델은 Github에 있는 matterport의 Mask R-CNN을 이용했으며 작업 내용에 따라 학습을 위한 설정 클래스, 차량 데이터셋 관련 클래스를 만들기 위해 이를 상속받아 구현했다.



( Fig 5. 이미지와 Annotation 데이터를 통해 만든 Mask 확인 )

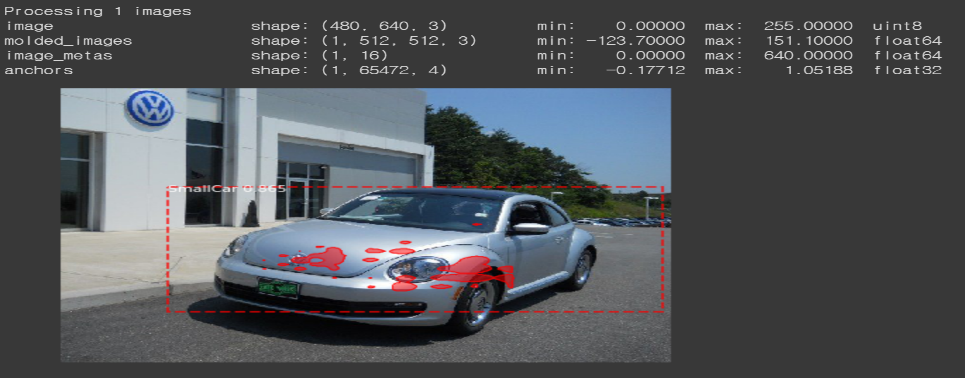
준비된 데이터 셋을 통해 모델을 학습시키고 테스트해서 차량의 크기에 따른 class 분류를 제대로 하는지 test한다. 모델의 Hyperparameter 중 epoch와 step은 각각 10, 1500을 준 것을 최종으로 하였다.



( Fig 6. Train )

* 1. **프로젝트 결과**

최종 Epoch, step에 따른 결과에 앞서 이전에 hyperparameter별 테스트했던 결과를 보면, (epoch, step)이 (4, 50)일 때 bounding box와 mask 둘 다 정확하지 않음을 볼 수 있다. (epoch, step)가 (4, 100)으로 증가하니 mask는 조금 증가했지만 bounding box는 여전히 차량의 위치를 잡지 못한 것을 확인할 수 있다. (epoch, step)를 (10, 1500) 으로 증가시키니 차량의 위치는 bounding box로 정확하게 잡아내는 것을 확인했으나 mask에 대해서는 윤곽을 따라가려고 하지만 부족함을 알 수 있다.

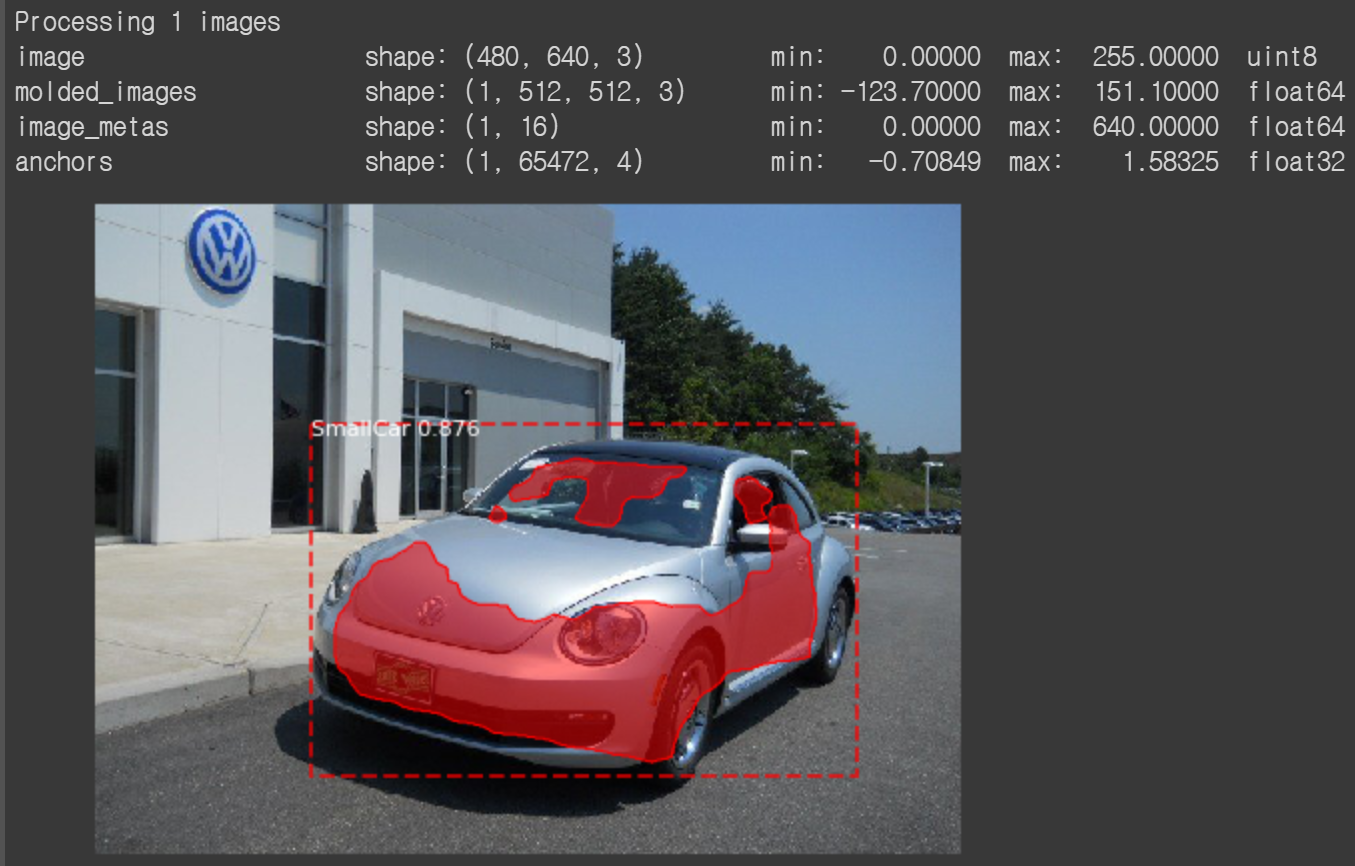


( Fig 7. epoch=4, step=50의 test )

건물, 실외, 하늘, 빨간색이(가) 표시된 사진

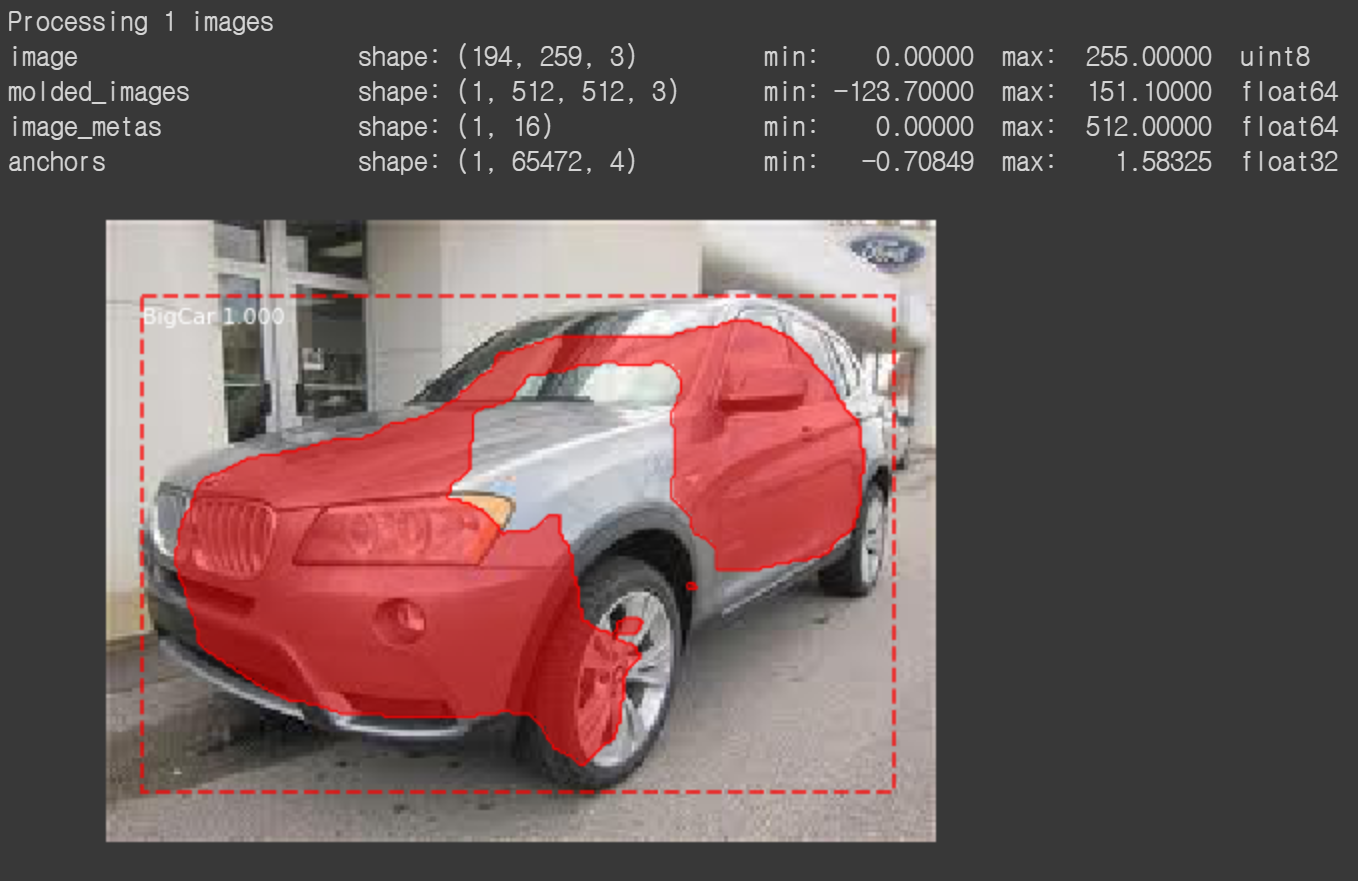
자동 생성된 설명

( Fig 8. epoch=4, step=100의 test )

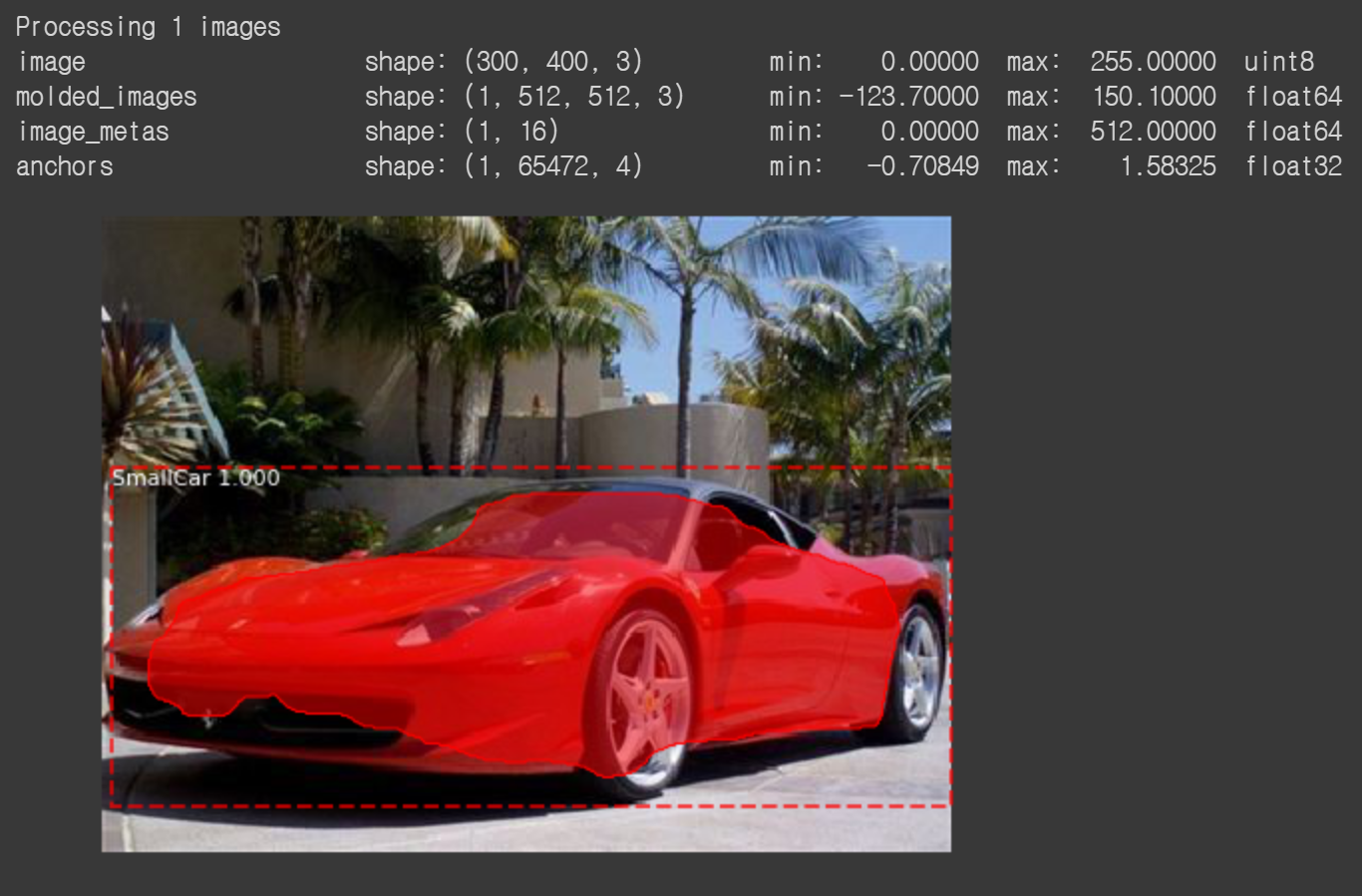


( Fig 9. epoch=10, step=1500의 test )

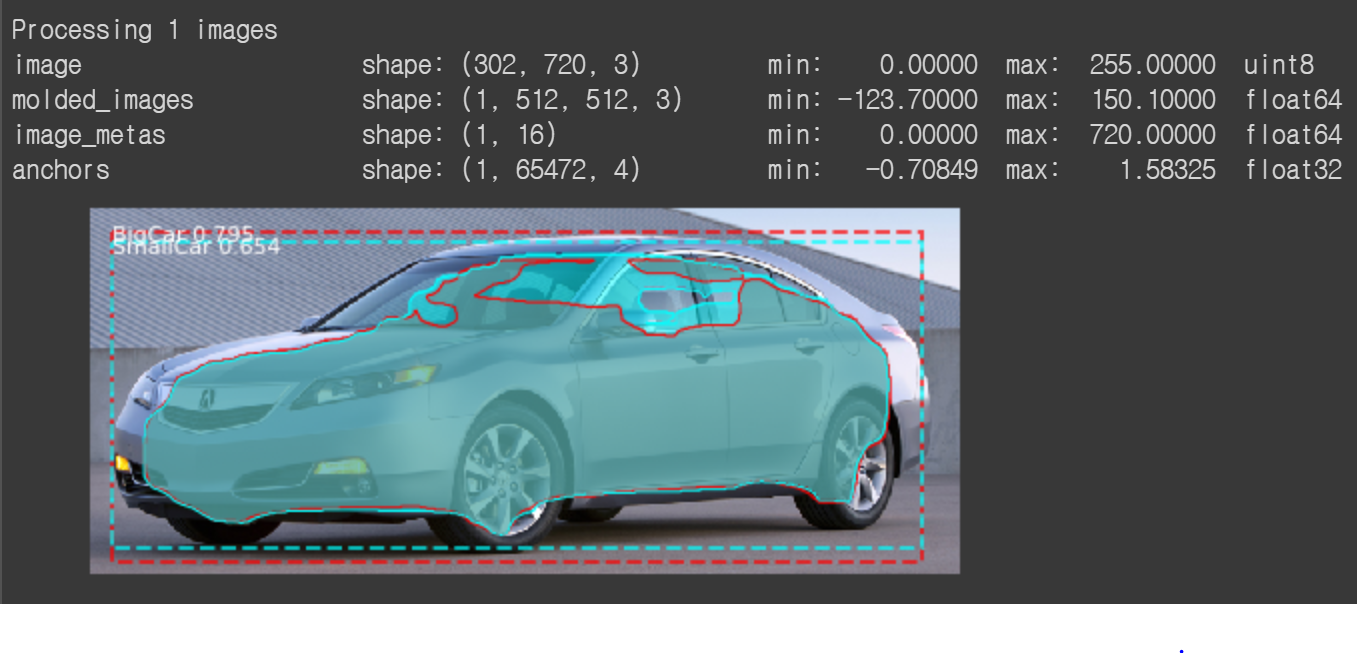
또한 SUV와 같이 명확히 큰 차량과 스포츠카와 같이 특이하게 생긴 소형 차량에 대해서는 Score가 0.9 이상으로 확실히 구분하는 것을 볼 수 있으나, 4문 차량과 같이 애매한 중소형 차량에 대해서는 확실한 구별을 하지 못한 것을 확인할 수 있었다.



( Fig 10. BigCar Test )



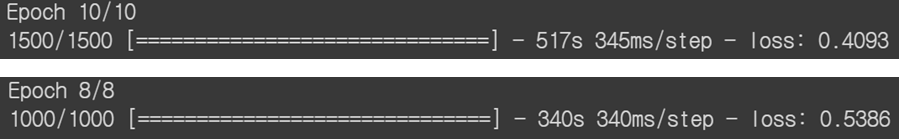
( Fig 11. SmallCar Test )



( Fig 12. MiddleCar Test )

1. **결론**

각 train 분석 결과를 볼 때, Epoch와 step 변화간 loss의 감소량을 보면 epoch와 step을 늘리면 object detection 과 masking을 더 잘 할 것이라 추측한다. 또한 test 결과를 보면 object detection은 지금도 잘하는 상태에서 Loss가 감소하면 masking 또한 object의 윤곽에 맞게 칠해질 것이라 예상한다.



( Fig 13. epoch, step별 loss 감소 분석 )

이미지에서의 객체 인식은 오랜 기간동안 연구되어왔다. 번호판 인식에서부터 기술의 발달로 차량 인식을 할 수 있게 되었으면 이제 단순히 차량 인식이 아니라 무언가 특성을 가진 유형 인식 차례가 되었다고 생각한다. 높은 정확도로 원하는 유형 분류가 가능하게 된다면, 그 이후에는 분류된 차량 데이터는 상업적인 빅데이터로의 사용이나 특수 목적을 가지고 추적하는 용도, 혹은 무인 자동차의 발전을 위해 사용될 것이라 기대할 수 있을 것이다.

**참고 문헌**

**[1]** Matterport, “Mask\_RCNN”, <https://github.com/matterport/Mask_RCNN>, 2019.03.10

**[2]** Jesús Utrera, “Stanford Car Dataset by classes folder”, <https://www.kaggle.com/jutrera/stanford-car-dataset-by-classes-folder>, 2018.07.02

**[3]** 윤성민, “Convolutional Neural Network을 이용한 차량 인식”, 한양대학교 공학대학원, 2016.