|  |
| --- |
| 『MINI AIFFELTHON 보고서』 |

|  |  |  |
| --- | --- | --- |
| **팀 명** | | 쏘카트 |
| **제안**  **프로젝트** | **명칭** | SPEED DAMAGE SEARCH |
| **소개** | 차량을 대여해주는 앱인 쏘카의 22년 12월 한 달간 이용자는 53만명이 넘고, 누적 가입자는 약 780만명이다. 때문에 차량 대여, 반납시 파손 여부를 일일히 사람이 판단하지 않고, 딥러닝을 이용해 빠르게 판단하는 시스템이 필요하다! |

쏘카트 팀 테스크 : semantic segmentation을 통해 한 이미지에서 하나 혹은 여러개의 레이블을 인식하고, 서로 다른 색으로 외곽선 그리기

단순하게 자동차 파손 부위를 segmentation한 마스크만 제공한다면, 한 이미지에 여러 종류의 파손이 있을 때, 혹은 해당 파손이 육안으로 구별하기에는 너무 작을 때 파손의 종류를 구분하기는 어려울 것 같다. 때문에 차량 파손 부위 사진을 segmentation하되, 레이블에 따라 서로 다른 색의 외곽선으로 표시해 색깔로 파손의 종류도 구분하려고 한다.

이를 응용해 랜트카앱 자체에서 카메라 기능에 실루엣을 제공한다면 대여 전후 사진을 빠르게 비교하여, 반납을 쉽게 만들 수 있을 것 같다!, 혹은 segmentation을 발전시켜 차종별, 부위별 수리비 데이터를 넣어준다면, 사용자의 현재 위치 근처의 카센터와 연결해 쉽게 수리비를 파악하는 응용 앱도 가능하다.

쏘카트팀 아이펠톤 프로세스

1) 문제 정의

쏘카에서 매일 접수되는 차량 이미지만 약 2만장이다. 이러한 이미지를 사람이 직접 보고 파손 여부와 부위를 분류한다면 비용 낭비가 크다. semantic segmentation을 통해 딥러닝으로 차량의 다양한 손상 이미지를 보고 파손의 부위와 그 종류를 인식해 외곽선을 잡아주는 프로그램을 만든다

2) 데이터 수집, EDA

테스크에 맞춰서 원천 데이터 안의 damage 폴더만 사용한다.

segmentation은 데이터의 질이 중요하기 때문에, 가능하다면 damage 폴더의 40만장 중 일부만 골라서 사용하려 한다.

json 파일의 파손 종류와 위치를 이미지와 연결해 레이블로 삼는다.

전처리 전에 미리 학습 데이터의 마스크를 따서 저장한다.

3) 전처리

모델을 고르면 그 input에 맞춰서 resize, 정규화 등의 필요한 전처리를 수행한다.

데이터의 양이 많으므로, 한번 전처리한 이미지를 따로 저장한 후, 학습할 때마다 다시 전처리하지 않고 이를 사용한다.

4) 모델 학습

semantic segmentation 모델 중 우리 테스크에 가장 적합한 모델을 통해 학습한다.

5) 모델 검증 및 시각화

파손 부위와 종류가 드러나게 레이블별 색이 다른 외곽선을 적용한 output을 출력한다.

validation만 이용해서 모델의 성능을 파악할지, train에서 test를 분리해 파악할지는 조금 더 생각해봐야 알 것 같다.

6) 앱 구현

현실적으로 앱 구현까지는 어려우므로, 모델의 성능을 높이는 단계에서 테스크를 끝내려한다.

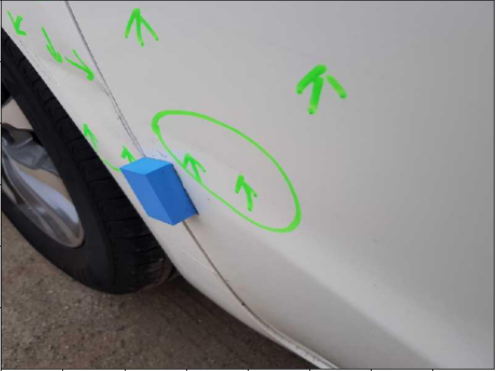
1. **수행 및 분석**

▶수행 결과 정리 및 시각화 (필요 시 사진 등 첨부 가능)

1) segmentation이 핵심이므로, 일단 fundamental 65, going deeper 4 노드를 복습하고, 기존에 차량 파손 데이터를 활용한 프로젝트들을 공부했다. Semantic segmentation에서는 대부분 U-Net, Deeplab V3, YOLO를 사용한다는걸 확인했다. 그 외에도 SOTA에 올라있는 모델이 더 성능이 잘 나올거라고 생각했다.

2) 주어진 이미지 데이터에는 damage와 damage\_part가 있는데, 폴더를 열어 이미지를 보려면 에러가 나는 상황이라 둘 중에 어느걸 사용해야할지 난감했다.

3) 차량 파손 이미지 위에 표시가 되어있는 이미지가 있는데, 모델 학습에 방해가 될 것이라고 판단해서 원천 데이터에서 이런 이미지는 어떻게 골라낼지 고민했다.



4) 데이터는 이미 확보되어있으니 전처리를 해야겠다고 생각했는데, 40만장의 이미지가 너무 많아 다루기에 버거웠다.

5) json 파일의 파일명과 같은 이미지 파일을 활용해야하고, json 파일 안의 damage 종류와 좌표를 이미지와 연결해야하므로, json 파일에 대해 공부했다.

6) 민제님이 공유 github를 만들어 이를 통해 코드를 공유했으며, 귀선님은 팀원별 일일 테스크와 회의록 작성을 맡았다.

**2. 결과 및 계획**

▶’미니 아이펠톤’ 결과 및 ‘아이펠톤’ 계획 요약

1) 현재 semantic segmentation 분야에서 SOTA에 올라있는 모델이라도, 우리 테스크에 적합하지 않으면 성능이 좋게 나올 수 없다. 때문에 우리 테스크에 적합한지를 위주로 모델을 찾고있다.

기존 프로젝트 노드에서는 pre-trained 모델을 사용했기 때문에, 이번 아이펠톤을 통해 처음으로 json 파일을 레이블로 삼아 연결하고, 테스크에 적합한 모델을 고르고, 그 모델의 input에 이미지와 json 파일을 모두 넣어 학습시키는 과정을 하게 되는데, 이 파트가 가장 어려운 부분이 될 것 같다.

2) 코드로 각 폴더별로 40장씩 이미지를 뽑아보니, damage\_part는 파손 부위가 어디인지 보여주기 위한 차량 전체가 나온 사진이라, 파손의 종류를 구분하는 테스크에서는 사용할 필요가 없다고 판단했다. 때문에 damage의 이미지 약 40만장만 사용하기로 했다.

3) 사진 위에 덧그려진 이미지와 관련해 퍼실님께 질문 드린 결과 segmentation은 json파일의 좌표로 지정되어있으므로 파손 부위 근처에 덧그린 이미지가 모델학습 하는데 지장 없다는걸 알게되었다. 때문에 해당 이미지들을 골라낼 필요 없이 그대로 사용하면 된다.

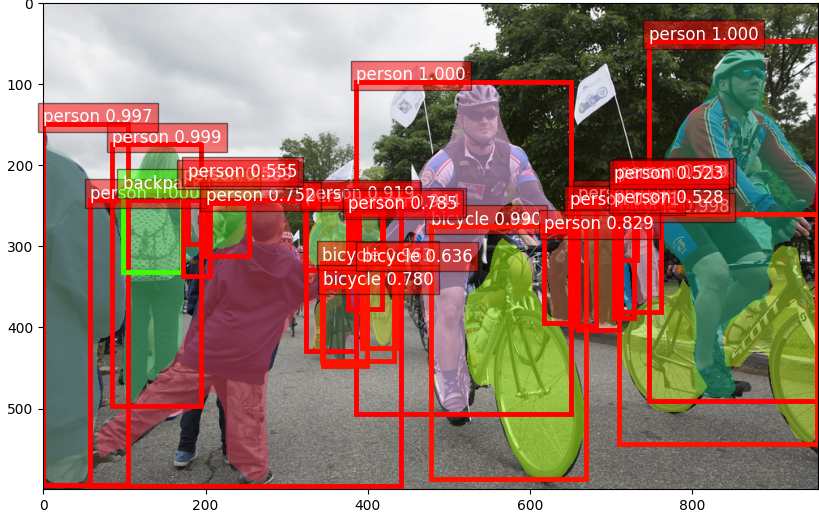
4) segmentation에서는 데이터의 질이 매우 중요하므로, 최대 5만장의 이미지를 어떻게 클래스별로 고르게 솎아낼지를 고심했다. 검색 결과, 코드를 짜서 클래스 별로 이미지를 원하는 개수만큼 골라낼 수는 있지만, 선명한 이미지만을 골라낼 수는 없었다.

하지만 성능이 좋은 대현님의 컴퓨터로 40만장 이미지 전처리하는데 35분 걸리는걸 알게되었고, 또한 h5py라는 것을 이용하면 대용량 이미지 파일을 이용해 딥러닝하는데 속도를 증가시킬 수 있다는 것도 발견했다. 퍼실님과의 미팅을 통해 json 파일의 annotation 안의 id가 특정 개수 이상인 이미지만 골라낸 후, 그 안에서 원하는 양의 이미지만 사용한다면, 한 이미지에 여러개의 파손을 학습 시킬 수 있다는걸 알게되었다.

5) json 파일 안에서 damage와 segmentation 좌표가 여러개라 이해가 안됐는데, 준혁님 코드로 출력해보고 한 이미지에 여러 종류의 파손이 들어갈 수 있다는걸 알게되었다.

한 이미지에 여러 종류의 파손이 들어갈 수 있다는걸 알게되니, json 안에 segmentation과 bbox 좌표를 모두 활용해서, 결과 이미지에 segmentation을 이용한 파손 부위의 외곽선과 detection을 이용해 파손의 종류를 표기하려 했다. 그러나 퍼실님과의 미팅을 통해 segmentation과 detection은 별개의 문제로, 모델 학습을 별도로 진행해야해서, 차라리 U-Net에서 contracting path가 끝난 후 결과물을 classification하고 expanding path가 끝난 후의 마스크와 합쳐서 레이블을 출력하는게 낫다는걸 알게되었다.

추가로 생각했을때, 차라리 레이블 별로 segmentation의 외곽선 색깔을 달리한다면 굳이 파손의 종류를 글씨로 써주지 않고도 구분할 수 있고, task도 단순화할 수 있을 것 같다.



6) 모든 조원의 컴퓨터가 40만장의 이미지를 처리하기는 어려워서, 컴퓨터에 너무 무리가 되는 팀원은 이미지를 솎아내는 과정이 완료될 때까지1200장의 이미지만 있는 샘플 데이터로 코드를 짜보려고 한다.

7) 모델에 대한 이해와 그에 따른 전처리를 어떻게 해야할지를 잘 모르기 때문에, 테스크를 구체화했고, u-net 모델을 segmentation에서 실제로 어떤식으로 사용하는지, 그에 따른 EDA에 관해 추가로 공부하고 있다.