

YOLOv4를 이용한 차량파손 검출 모델 개선

Improving the Vehicle Damage Detection Model using YOLOv4

전 종 원*, 이 효 섭*, 한 희 일**★

Jong Won Jeon*, Hyo Seop Lee*, Hee Il Hahn**★

Abstract

This paper proposes techniques for detecting the damage status of each part of a vehicle using YOLOv4. The proposed algorithm learns the parts and their damages of the vehicle through YOLOv4, extracts the coordinate information of the detected bounding boxes, and applies the algorithm to determine the relationship between the damage and the vehicle part to derive the damage status for each part. In addition, the technique using VGGNet, the technique using image segmentation and U-Net model, and Weproove.AI deep learning model, etc. are included for objectivity of performance comparison. Through this, the performance of the proposed algorithm is compared and evaluated, and a method to improve the detection model is proposed.

요 약

본 논문에서는 YOLOv4를 이용하여 차량의 부위별 파손현황을 검출하는 기법을 제안한다. 제안 알고리즘은 YOLOv4를 통해 차량의 부위와 파손을 각각 학습시킨 후 검출되는 바운딩 박스의 좌표 정보들을 추출하여 파손과 차량부위의 포함관계를 판단하는 알고리즘을 적용시켜 부위별 파손현황을 도출한다. 또한 성능비교의 객관성을 위하여 동일분야의 VGGNet을 이용한 기법, 이미지 분할과 U-Net 모델을 이용한 기법, Weproove.AI 딥러닝 모델 등을 대조 모델로 포함한다. 이를 통하여 제안 알고리즘의 성능을 비교, 평가하고 검출 모델의 개선 방안을 제안한다.

Key words : Deep Learning, YOLOv4, Object Detection, Computer Vision, Vehicle Damage Detection

1. 서론

최근에는 교통사고, 차량 대여 어플리케이션, 보험, 차량 관리, 렌터카 등, 자동차와 관련된 다양한 분야에서 보다 더 정확하고 객관적인 시각으로 차량 부위별 파손현황을 인식하는 기술이 요구되고 있다. 이러한 사회적 요구에 부응하여 딥러닝 기술을 이용한 차량파손 검출에 관한 연구[1][2][3]가

발표되고 있다. 대표적인 차량파손 검출 알고리즘으로는 VGGNet을 이용한 연구[4]와 WeProov.AI 팀의 연구[5], 쏘카(SOCAR)에서 진행된, 이미지 분할과 U-Net모델을 이용한 기법[6] 등이 있다.

우선, VGGNet을 사용한 차량 파손 검출 모델[4]은 입력 이미지에 대하여 사전에 훈련된 VGG 모델을 통해 자동차의 존재 유무를 판단하고 차량이 존재한다고 판단된 영역에서의 파손 여부를 판단

* Undergraduate Student, Dept. of Information and Communications Engineering, Hankuk University of Foreign Studies

** Professor, Dept. of Information and Communications Engineering, Hankuk University of Foreign Studies

★ Corresponding author

E-mail : hihahn@hufs.ac.kr, Tel : +82-31-330-4356

※ Acknowledgment

This work was supported by Hankuk University of Foreign Studies Research Fund.

Manuscript received Nov. 30, 2021; revised Dec. 28, 2021; accepted Dec. 31, 2021

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

한다. 파손이 있을 시 자동차의 파손된 부분의 위치와 파손의 심각 정도를 판단한다. 이때 파손 부위는 전면, 후면, 옆면 등으로 나누고 파손의 심각한 정도는 세 등급으로 나눈다. VGGNet을 사용한 연구는 VGG16[7]이나 VGG19[8]를 이용하여 서로 다른 전층 학습 모델의 성능을 추정하기 위하여 정밀도, 재현율 및 F1-score의 세 가지 측정 기준을 사용한다. VGG19보다 VGG16에서 더 좋은 성능을 보이고 VGG16의 손상 감지, 손상 위치 및 손상 심각도에서 각각 94%, 71%, 61%의 정밀도를 보인다. 이 연구에서는 차량 파손 부위를 전면, 후면, 옆면 등 세 부분에 국한되어 있고, 각 부위에 어떠한 종류의 파손이 있는지와 개수를 파악하지 못한다는 문제점을 보인다.

WeProov.AI의 차량 파손 검출 모델[5]은 관련 연구들 중 차량 부위 및 손상 종류를 가장 세분화한 모델에 속한다. 차량 부위는 44개의 부분으로 나누어 구분하고, 손상 종류는 유리깨짐, 스크래치, 덴트 등, 총 18개의 종류로 분류한다. 학습에 이용한 데이터 또한 100만 장 이상이며, 24 가지의 모듈을 통합한 자체적인 알고리즘을 제작하여 차량의 파손부위와 종류를 검출한다. 인식률이 매우 우수하고 디테일한 판정결과를 출력하는 장점이 있으나 검출하는데 소요되는 시간이 평균 2분 정도로 매우 느리다는 한계가 있다.

이미지 분할과 U-Net[9]을 이용한 딥러닝 모델[6]은 차량 대어 어플인 쏘카에서 2019년 진행되었던 프로젝트이다. 이 모델은 입력 이미지에서 차량이 존재하는 영역만을 선별하고 파손 존재 여부를 판단한다. 파손 존재 시, 이미지 분할 네트워크를 통하여 파손의 종류를 스크래치, 찌그러짐, 이격, 해당없음 등으로 나누어 판정한다. 출력 데이터는 이미지 단위 파손 존재 여부와 픽셀 단위 파손 종류에 대한 인덱스이다. 학습 데이터 수는 육안으로 파손 여부를 확실하게 판단할 수 있는 이미지 2,000장이고 이미지 분석 시 이미지 분할 기법에 사용되는 연산량을 줄이기 위하여 U-Net를 사용한다. 파손 검출 성능은 96%의 정확도와 96%의 F1-score를 보인다. 쏘카 모델은 이미지 화소 단위로 확인하는 이미지 분할 기술을 활용하기 때문에 작은 파손도 감지할 수 있는 장점이 있지만 네트워크를 통과할 때 층이 깊고 연산량이 많다는 문제점을 보인다.

이러한 상황을 고려하여 본 논문에서는 검출시간

을 줄이면서 디테일한 파손 검출정보를 제공하는 기법을 설계하기 위하여 YOLOv4[10]를 이용하여 차량 부위별 파손현황을 검출하는 알고리즘을 제안한다. 파손이 존재하는 차량 사진들을 수집하고 차량의 부위를 나타내는 클래스(뒷문, 범퍼, 휠더, 앞문, 후드, 후면 등)와 차량의 파손을 나타내는 클래스(스크래치, 충돌, 진흙 등)로 라벨링을 한 뒤 각각 학습을 진행한다. 여기서, 진흙을 파손 클래스에 넣은 이유는 진흙이 묻어 있는 부위를 파손으로 인식하는 경향이 많기 때문에 이를 최종 판정에서 배제하기 위함이다.

본 논문에서는 학습된 모델에서 도출되는 바운딩 박스의 중심좌표, 너비, 높이를 이용하여 파손이 있는 부위와 파손종류의 포함관계를 파악하는 알고리즘을 구현한다. 이와 더불어 기 제안된 대표적인 차량 파손검출 알고리즘들과의 판별 성능을 비교 분석한다. 평가지표로는 정밀도(Precision), 재현율(Recall), F1-score(정밀도와 재현율의 조화평균)를 사용하고, 성능을 비교 평가하기 위하여 차량의 파손검출 성능, 부위검출 성능, 최종검출 성능 등에 대한 비교와 함께 파손 부위 종류의 세부내용, 데이터 셋 및 검출시간을 이용한다.

본 논문의 구성은 다음과 같다. II절에서는 제안 기법을 설명하고 III절에서는 선행연구들과 제안 기법의 성능을 비교분석한다. 마지막으로 IV절에서는 결론을 맺고 향후 연구 진행방향에 대하여 논의한다.

II. 제안 기법

본 논문에서 제안한 알고리즘은 학습 과정과 테스트 과정으로 나뉜다. 학습 과정에서는 우선, 학습 이미지를 준비하고 뒷문, 범퍼, 휠더, 앞문, 후드, 후면, 충돌, 스크래치, 진흙 등, 9 개의 클래스를 정의하여 준비된 학습 이미지들에 대하여 라벨링을 진행한다. 정의된 클래스들 중에서 뒷문, 범퍼, 휠더, 앞문, 후드, 후면 클래스들은 차량 부위에 해당되고, 충돌, 스크래치 클래스들은 차량 파손에 해당된다. 라벨링은 YOLO의 라벨링 툴인 YOLO_mark를 이용하여 진행한다. 라벨링을 진행한 뒤 YOLOv4의 사전 학습 모델(pretrained model)을 이용하여 학습을 진행한다. 그림 1은 제안 알고리즘의 흐름도를 나타낸다.

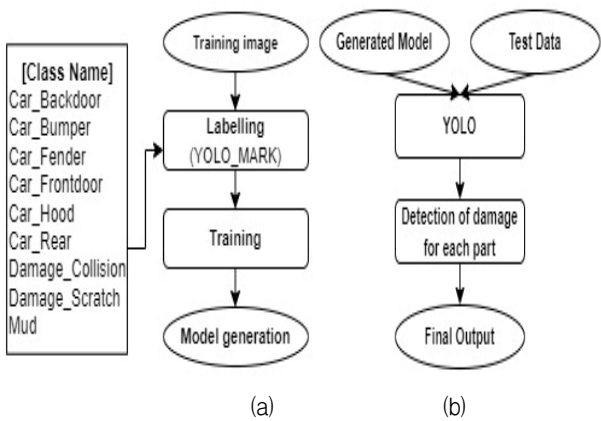


Fig. 1. The flow charts of our proposed algorithm.
 (a) Training process (b) Testing process.
 그림 1. 제안 알고리즘의 흐름도
 (a) 학습 과정 (b) 테스트 과정

테스트 과정에서는 학습 후 생성된 모델에 테스트 이미지를 입력시켜 출력된 좌표 값들을 이용하여 차량의 부위와 파손의 포함관계를 파악한 후 최종적으로 부위 별 파손현황을 도출한다. 그림 2와 표 1은 각각 기존 YOLOv4 네트워크에서 학습이 완료된 후 임의의 테스트 이미지에 대하여 생성된 모델로 구한 출력 이미지와 검출 정보를 나타낸다. 표 1을 보면 검출된 객체들의 클래스 이름과 신뢰도(confidence)가 최종 출력되는 것을 볼 수 있다.

신뢰도는 해당 클래스의 일치도를 나타내는데 객체의 신뢰도만으로는 바운딩 박스의 포함관계를 계산할 수 없다. 따라서 기존 YOLOv4로는 부위별 파손현황을 검출하지 못하는 한계가 있다.

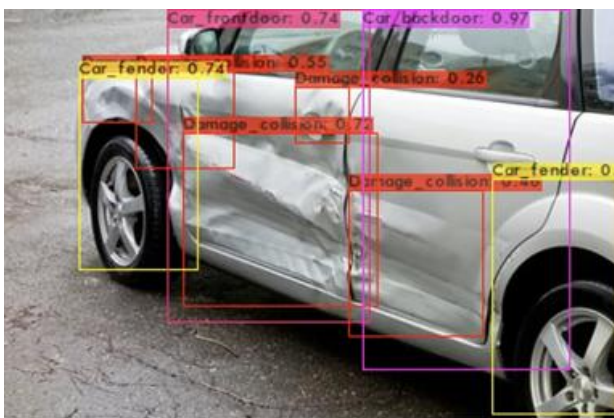


Fig. 2. An example of an output of an existing YOLOv4.
 그림 2. 기존 YOLOv4의 출력 이미지 예시

이를 해결하기 위하여 본 논문에서는 부위별 파손현황을 도출하는 알고리즘을 구현하여 YOLOv4

Table 1. The results of detection when the YOLOv4 applied to Fig. 2

표 1. 그림 2에 대응하는 기존 YOLOv4의 출력

| Detected Class Name | Confidence |
|---------------------|-------------------------|
| Car_backdoor | 97% |
| Car_fender | 74%, 99% |
| Car_frontdoor | 74% |
| Damage_collision | 17%, 26%, 46%, 55%, 72% |

의 image.c 화일에 적용한다. 테스트 이미지를 제안 알고리즘에 적용하여 검출된 각각의 바운딩 박스들을 부위와 파손으로 나눈 후 각 박스들의 중심좌표와 너비, 높이 등을 추출한다. 그림 3에서 b1이 부위를 나타내는 바운딩 박스라고 정하고 이의 중심점을 (x, y), 너비와 높이를 각각 w, h로 표기한다. b2는 파손을 나타내는 바운딩 박스이고 이의 중심점을 (x₁, y₁)으로 표시한다. b2 박스의 x 좌표가 b1 박스의 x 좌표 ±w/2 범위 안에 놓이고 b2 박스의 y 좌표가 b1 박스의 y 좌표 ±h/2 사이에 있다면 b2 박스가 b1 박스 안에 존재하는 것으로 판단하도록 한다. 이를 수식으로 나타내면 다음과 같다.

$$x - \frac{w}{2} \leq x_1 \leq x + \frac{w}{2}$$

$$y - \frac{h}{2} \leq y_1 \leq y + \frac{h}{2}$$

(1)

식 (1)에 의해 파손의 중심좌표가 해당 부위 내부에 존재하면 그 부위에 해당 파손이 존재한다고 판단하고 그렇지 않으면 그 부위에 해당 파손이 없다고 판단하고 다음 부위로 넘어가 위 과정을 다시 적용한다.

동일한 부위에서 검출된 동일 종류의 파손들끼리 식 (1)을 이용해 서로 순회시킨 후, 각각의 파손 개체들의 중심좌표가 나머지 개체들의 내부에 있으면 동일 파손으로 인식하도록 한다. 이를 통하여 하나의 파손에 2개 이상의 바운딩 박스가 형성되어 중복검출 발생 경우에도 동일 파손이 하나의 파손으로 인식되도록 파손 중복검출 방지 알고리즘을 구현한다. 부위 별 파손검출 알고리즘과 파손 중복검출 알고리즘을 모두 통과한 후 해당 부위에 파손이 있다고 판정되면 “해당파손 IN 해당부위” 형식으로 결과가 출력되도록 한다.

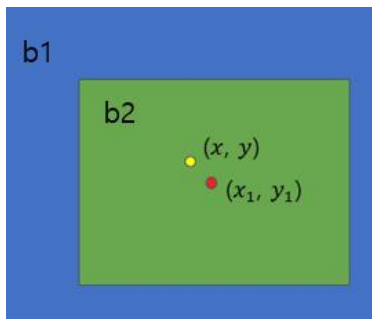


Fig. 3. An image for deriving the inclusion relationship between the bounding boxes of parts and damages.
 그림 3. 부위 바운딩 박스와 파손 바운딩 박스의 포함관계를 도출하기 위한 이미지

III. 실험 결과

본 논문에서는 학습 데이터 구성을 위하여 깃허브(Github)에서 차량 파손 데이터 4,600장[11]과 웹 크롤링, 실제 촬영을 통해 파손이 없는 차량 사진 1,000장, 충돌이 있는 차량사진 1,325장, 스크래치가 있는 차량사진 770장, 진흙이 묻어있는 차량사진 335 장 등, 총 3,430장을 학습 데이터 셋으로 사용한다. 반면에 제안 알고리즘의 성능 테스트를 위하여 웹 크롤링을 통해 얻은 사진들과 직접 촬영한 차량 사진을 포함한 300장의 차량 사진을 테스트 데이터 셋으로 구성한다. 학습 환경은 opencv-4.2.0, CUDA-10.1, CUDNN-10.1, GTX-1080Ti 2개를 사용하고 라벨링 데이터를 담고 있는 train.txt와 클래스 정보를 담고 있는 obj.data, obj.names, 학습 설정 정보를 담고 있는 yolov4.cfg와 함께 YOLO에서 제공하는 사전 학습 모델인 yolov4.conv.137을 사용하여 학습한다. 학습률(learning rate)은 0.0013, 배치 크기는 32, 학습횟수는 20000회로 설정한다.

학습 진행과정 중 발생한 문제점으로는 크게 두 가지로 나뉠 수 있다. 첫 번째로는 파손이 인식되지 않거나 다른 종류의 파손으로 인식되는 현상이고, 두 번째로 파손이 중복검출 되는 현상이다. 첫 번째 문제를 해결하기 위하여 훈련 차수를 증가시킬 때마다 충돌과 스크래치 학습 데이터 양을 함께 늘린다. 또한 일차 학습에서 스크래치와 진흙을 구분하지 못하는 현상을 해결하기 위하여 이차 학습부터 진흙이 묻은 차량 사진을 학습 데이터에 추가한다. 두 번째 문제인 파손 중복검출을 해결하기 위하여 III절에서 소개한 중복검출 방지 알고리즘을 YOLOv4 네트워크에 적용한다.

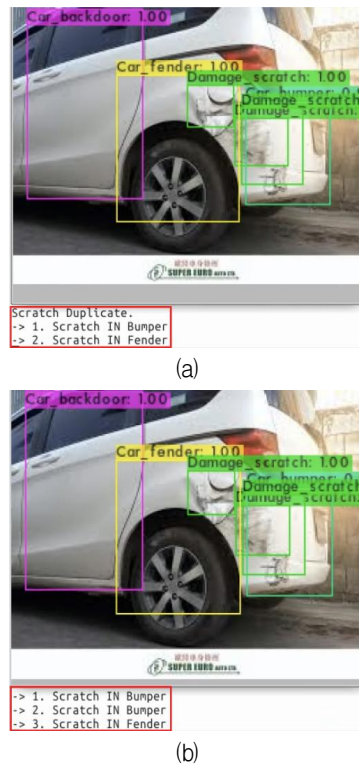


Fig. 4. Comparison of the outputs before and after application of the duplicated damage detection prevention algorithm. (a) Before application (b) After application.
 그림 4. 파손 중복검출방지 알고리즘 적용 전후 출력 비교 (a) 적용 전 출력 (b) 적용 후 출력

그림 4는 중복검출 알고리즘 적용 전후의 출력 차이를 보여준다. 중복검출 방지알고리즘 적용 전에는 차량 후면 범퍼에 스크래치가 중복되어 1, 2번에서 “Scratch IN Bumper”가 출력되지만, 중복검출 알고리즘 적용 후에는 “Scratch Duplicate”가 출력되며 중복검출을 처리한 후 정상적으로 출력되는 것을 확인할 수 있다. 그림 5는 진흙이 묻은 차량 이미지에 대한 결과를 보여 준다. 검출 부위는 Car_hood(0.35), Car_fender(0.79), Mud(0.52 / 0.14 / 0.22)가 출력된다. 진흙이 묻은 부분을 스크래치로 인식하던 오류를 진흙 클래스를 추가함으로써 해결한다.

표 2와 표 3은 그림 2에 대응하는 모든 알고리즘을 통과하고 나온 최종 결과를 나타낸다. Class_ID는 검출된 클래스들의 고유 ID로 0~5는 부위, 6~7은 파손, 8은 Mud 클래스를 나타낸다. X_center와 Y_center는 각 바운딩 박스의 중심좌표, Width와 Height는 각각 바운딩 박스의 너비와 높이를 나타내는데 그 값은 정규화되어 있다. 표 3에서는 Class_ID 6번의 Collision이 총 5개 검출되었고 각각의 중

심좌표와 너비, 높이가 제시되어 있다. 표 1을 참고하면 휠더 2개, 뒷문 1개, 앞문 1개가 각각 검출된 것을 확인할 수 있다.

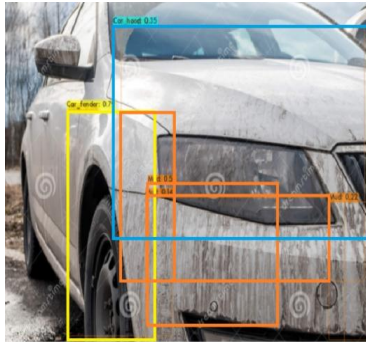


Fig. 5. Output image after adding Mud class.
그림 5. Mud 클래스 추가 후 출력이미지

Table 2. Geometric measurement of the detected parts corresponding to Fig. 2.

표 2. 그림 2에 대응하는 검출부위 결과

| Class ID | X_center | Y_center | Width | Height |
|----------|----------|----------|-------|--------|
| 2 | 0.22 | 0.37 | 0.19 | 0.43 |
| 3 | 0.43 | 0.35 | 0.33 | 0.71 |
| 0 | 0.75 | 0.40 | 0.34 | 0.83 |
| 2 | 0.90 | 0.65 | 0.21 | 0.53 |

Table 3. Output of the detected damages corresponding to Fig. 2.

표 3. 그림 2에 대응하는 파손부위 출력

| Class ID | X_center | Y_center | Width | Height |
|----------|----------|----------|-------|--------|
| 6 | 0.19 | 0.20 | 0.11 | 0.12 |
| 6 | 0.52 | 0.24 | 0.09 | 0.12 |
| 6 | 0.67 | 0.57 | 0.22 | 0.33 |
| 6 | 0.30 | 0.25 | 0.16 | 0.22 |
| 6 | 0.45 | 0.48 | 0.32 | 0.39 |

표 4는 VGGNet 모델과 제안 알고리즘의 성능비교를 보여준다. 제안 알고리즘은 범퍼, 휠더, 앞문, 후드, 후면 등, 6개의 차량 부위를 구분하지만, VGGNet 사용모델은 전면, 후면, 옆면 등, 총 세 부위로 구분한다. 동일 이미지 테스트 사용 시 제안 알고리즘이 보다 더 정확한 부위를 검출해낼 수 있음을 의미한다. 제안 알고리즘은 차량 손상 종류(Collision, Scratch, Mud)를 구분하고, VGGNet 모델은 세 가지 손상 정도(Minor, Moderate, Severe)

를 구분한다. 각각의 검출 성능을 대조해 볼 때, 파손 검출 성능 면에서 정밀도는 동일하지만 재현율과 F1-score는 각각 0.16, 0.09 정도 낮은 것을 확인할 수 있다. 부위검출 성능에서는 더욱 세분화한 부위를 도출하였음에도 정밀도, 재현율 및 F1-score가 각각 0.24, 0.21, 0.23 높은 것을 확인할 수 있다.

Table 4. Comparison of performances of the model using VGGNet and our proposed algorithm.

표 4. VGGNet 이용모델과 제안 알고리즘의 성능비교

| | Damage Detection | | | Part Detection | | | Final Detection | | |
|----------------|------------------|--------|----------|----------------|--------|----------|-----------------|--------|----------|
| | Precision | Recall | F1-score | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Proposed Model | 0.94 | 0.78 | 0.85 | 0.95 | 0.90 | 0.92 | 0.93 | 0.70 | 0.79 |
| VGGNet Model | 0.94 | 0.94 | 0.94 | 0.71 | 0.69 | 0.69 | - | - | - |

제안 알고리즘의 차량 분류부위가 6개이고 손상 종류가 3가지인데 반하여 WeProov.AI는 각각 44개, 18가지로 월등히 많은 종류를 구분한다. 하지만, 학습 데이터 셋 양이 백만 장이상 요구되어 제안 알고리즘에 비해 300배 정도로 많고, 검출 시간 또한 평균 2분으로 제안 알고리즘의 10초에 비해 매우 길어 비용적인 면에서 다소 비효율적일 수 있다.

IV. 결론

본 논문에서는 YOLOv4를 이용하여 차량의 부위와 파손을 검출한 뒤, 각각 검출된 바운딩 박스의 중심좌표와 너비, 높이를 이용한 알고리즘을 적용하여 부위별 파손현황을 도출하는 알고리즘을 제안하였다. 제안 알고리즘은 기존 VGGNet 기법을 이용한 동일 주제의 연구와 비교를 위해, 파손검출 성능, 부위검출 성능, 알고리즘을 통과한 최종 부위별 파손검출 성능의 정밀도와 재현율, F1-score 등을 계산하였다. 파손검출 성능은 떨어지는 모습을 보이지만 부위검출성능에서는 더 뛰어난 성능을 보인다. 또한 VGGNet 사용모델이 차량의 전면, 후면, 옆면 등, 세 개 부위의 파손 정도를 검출하는 반면 제안 안고리즘은 차량의 뒷문, 범퍼, 휠더, 앞문, 후드, 후면 등, 여섯 개 부위와 함께 파손 종류와 개수까지 도출함으로써 결과 면에서 더 구체성을 띄고 있다. 최종 부위별 파손검출성능 또한 모든 평가 지표가 0.70 이상의 성능을 보여주고 있다.

향후에는 입력 이미지의 크기 변화와 전처리를 통하여 차량 사진에서 빛의 반사나 블러링의 영향을 줄이면서 라벨링 재작업, 차량 부위 추가학습 등을 통해 바운딩 박스의 사각지대 등을 감소시킬 계획이다.

References

- [1] K. Patil, M. Kulkarni, A. Sriraman, and S. Karande, "Deep learning based car damage classification," *16th IEEE International Conference on Machine Learning and Applications*, 2017.
DOI: 10.1109/ICMLA.2017.0-179
- [2] H. S. Malik, M. Dwivedi, S. N. Omakar, S. R. Samal, A. Rathi, E. B. Monis, B. Khanna, and A. Tiwari, "Deep learning based car damage classification and detection," *EasyChair preprints*, 2020.
DOI: 10.1109/ICMLA.2017.0-179
- [3] P. Rakshata, H. V. Padma, M. Pooja, H. V. Yashaswini, and V. Karthik, "Car damage detection and analysis using deep learning algorithm for automotive," *International Journal of Scientific Research & Engineering Trends*, Vol.5, Issue.6, 2019.
- [4] P. M. Kyu and K. Woraratpanya, "Car damage assessment based on VGG models," *JSCIS*, 2021.
- [5] WeProof.AI, "AI for vehicle inspection, damages & claims management," 2020.
- [6] SOCAR Tech, "Developing car damage detection model using semantic segmentation," <https://tech.socarcorp.kr/data/2020/02/13/car-damage-segmentation-model.html>, Feb. 2020.
- [7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs.CV]*, Sep. 2014.
- [8] OpenGenus IQ, "Understanding the VGG19 Architecture," <https://iq.opengenus.org/vgg19-architecture/>, 2018.
- [9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs.CV]*, 2015.
- [10] A. Bochkovskiy, C. Y. Wang, and H. M. Liao,

"YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934v1*, 2020.

[11] Car damage dataset, <https://github.com/jennyLuciaV/car-damage-dataset>.

BIOGRAPHY

Jong Won Jeon (Member)



2022 : BS degree in Information and Communications Engineering, Hankuk University of Foreign Studies.

JiHyo Seop Lee (Member)



2022 : BS degree in Information and Communications Engineering, Hankuk University of Foreign Studies.

Hee Il Hahn (Member)



1984 : BS degree in Control and Instrumentation Engineering, Seoul National University.

1986 : MS degree in Control and Instrumentation Engineering, Seoul National University.

1995 : PhD degree in Electrical and Computer Engineering, University of Arizona.

1998~ : Professor, Dept. Information and Communications Engineering, Hankuk University of Foreign Studies.

1987~1998 : Senior Researcher, Electronics Telecommunication Research Institute.