딥러닝 네트워크를 통한 택배 상자 파손 분류

김은강, 김성하, 신혜선, 김소연, 이범식

조선대학교 정보통신공학부

kety26@chosun.kr, tjdgk2323@chosun.kr, sechskies01@chosun.kr, kk7368@chosun.kr, bslee@chosun.ac.kr

Classification for the Breakage of the Package Boxes using a Deep Learning Network

Kim, Eun-Kang, Kim, Seong-Ha, Sin, Hye-Seon, Kim, So-Yeon and Lee, Bumshik

Chosun University, Department of Information and Communication Engineering

요 약

본 설계에서는 택배의 현재 상태를 확인 후 택배 상자의 파손 유무를 분류하고 사진으로 제공하는 기술을 제안하였다. 본 설계에서는 딥러닝 네트워크를 통해 훈련된 인공지능을 통해 일반 상자와 파손 상자를 분류하고, 파손 상태일 시 소비자와 택배사에 알람으로 보고하는 것을 주 기능으로 하고 있다. 딥러닝 네트워크 훈련을 위해 약 1,000장의 데이터셋을 직접 구성하고 학습하였다. 본 설계에서 사용된 택배 상자 파손 여부 분류기의 분류 정확도는 93.33%이고, 이 분류 성능은 택배 상자의 상태를 분류하는 데 있고, 정확도의 분류 성능이라고 할 수 있다.

Ⅰ. 서론

코로나19 등의 여파로 택배 수가 증가함에 따라 택배가 반송되는 비율이 함께 증가하고 있다. [그림 1]과 같이 반송되는 택배의 주원인은 파손이 주원인으로 알려져 있다. 또한 소비자의 입장에서 내용물이 불량이거나 파손된 경우 반송하는 과정이 번거로울 뿐만 아니라 회사의 경우 택배가 배송 중 파손된 경우에는 소비자에게 이에 대한 보상을 해야하는 과정이 필요하다. 이를 사람이 직접 점검하기에는 택배의 물량이 많을 뿐만 아니라, 이로 인한 비용 역시 크게 증가한다. 이러한 문제를 해결하기 위해 딥러닝 네트워크를 통한 실시간으로 택배 파손 여부를 알려주는 시스템을 제안한다. 직접 파손 여부를 판별하지 않아 편리하다는 장점을 가지고 있다. 심하게 훼손된 택배 상자는 안의 내용물까지 파손되었을 가능성이 있다고 판단하여, 이를 방지하기 위한 기술이다.

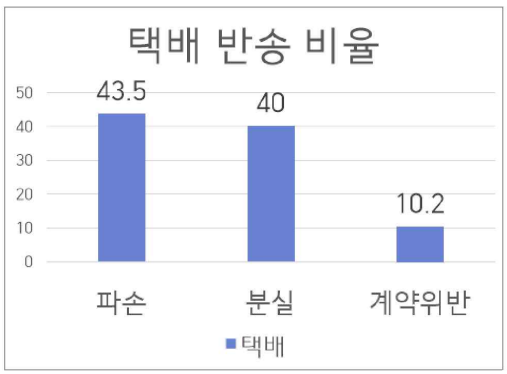

그림 1. 택배 반송 비율

Ⅱ. 기존 방법

기존의 택배를 분류하는 방법은 사람이 직접 파손된 상자를 확인하여 분류하는 방법을 사용하고 있다. [1] 파손된 물건이나 영상을 구분하는 것을 주 기능으로 하는 기술로는 <쏘카>의 차량 파손 탐지가 있다. <쏘카>의 차량 파손 탐지 기술을 사용하면 사람이 구분하지 못하는 구체적인 파손 위치가 표시할 수 있다. 주어진 차량 이미지 내의 파손 영역과 파손 종류를 픽셀단위로 분류하여 파손 여부를 판단한다. [그림 2]와 같이 이미지를 나누어 파손 존재 여부와 픽셀 단위로 파손(Scratch), 이격(Spacing), 찌그러짐(Dent), 해당없음(Background)로 분류한다. 또한 파손된 차량에 대한 데이터를 수집하여 차량의 파손 정도, 위치, 시점 등 히스토리가 있는 데이터로 활용할 수 있다. 일 평균 7~8만 장, 성수기의 경우 최대 11만장의 차량 외관이미지로 검수할 데이터가 존재한다. <쏘카>의 차량 파손 탐지는 꾸준한 모니터링의 필요가 없어 업무 담당자의 차량 외관 이미지를 검수하는 인력과 시간을 절약할 수 있다.

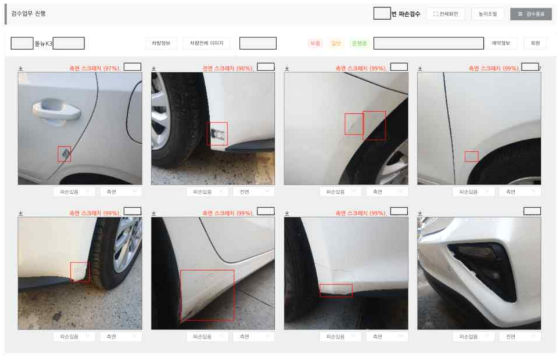

그림 2. <쏘카>의 차량 파손 탐지

그러나 <쏘카>의 차량 파손 탐지의 문제점으로는 크게 4가지가 존재한다. 첫 번째로 초기 데이터 오류로 인해 촬영 가이드라인이 존재하지지 않아 촬영된 파손 사진이 필수적으로 촬영되지 않는 부분이 존재한다. 이 것은 파손에 대한 정확한 판단을 하지 못하게 하는 원인이 될 수 있다. 이러한 파손 탐지 알고리즘의 문제점을 해결하기 위해 본 설계에서는 초기 택배 상자 데이터의 상자 가이드를 정하여 대부분 일정한 구도로 촬영하게 하였다. 두 번째로는 어두운 장소나 너무 밝은 장소에서 촬영 시 분류 모델의 정확도가 감소하는 문제가 있다. 이러한 문제을 해결하기 위해 본 설계에서는 촬영 시 밝기를 조절하여 상자의 형태를 촬영하였다. 세 번째로는, 파손 부분을 촬영 시 리서 찍거나 주차장의 경우 촬영하는데 어려움이 있다. 이러한 문제점은 아두이노를 이용하여 택배의 간격을 조절하는 방법으로 문제점을 해결하였다. 마지막으로 차량의 아래 부분이나 윗 부분등 사람의 손이 닿지 않는 사각지대가 생성된다. 이에 대한 해결방안으로 카메라를 여러 방향에서 촬영하여 사각지대를 성하지 않도록 하였다.

Ⅲ. 시스템 개발

3.1 초기 택배 이미지 수집과 라벨링

본 설계에서는 일반 상자와 파손된 상자, 상자가 아닌 것으로 구분하는 데이터셋을 구성하였다. 이러한 데이터셋을 위해 3가지 방식으로 초기 데이셋을 구성하였다.

1) 검색을 통한 데이터 수집 : 구글 검색 결과로 나오는 이미지를 저장하는 방법을 사용하였다. 검색을 통한 이미지의 경우 그림 3과 같이 겹쳐있는 경우와 촬영 가이드에 맞는 이미지의 차이가 발생하여 상자의 사각지대가 발생하여 인식률이 낮게 나왔다.

그림 3. 검색한 이미지와 촬영 가이드에 맞는 이미지 차이

2) 택배 상자의 이미지를 직접 촬영 : 이 설계에서 다양한 환경에서 상자의 파손 여부를 판별할 수 있도록 데이터셋의 대부분을 직접 촬영하여 새로운 데이터셋을 구축하였다. 상자 이미지를 바닥이나 공중에서도 촬영하여 다양한 배경을 바탕으로 직접 촬영하였고, 상자가 아닌 것의 경우 색상만으로 분류되는 일을 방지하기 다양한 색상의 이미지를 찾아 추가 하였다.

3) 수집한 이미지 변환을 통한 데이터 증강 : 앞에 설명한 방식으로 수집한 데이터를 바탕으로 회전각도, 확대, 축소, 좌우 이동, 좌우 반전 등 데이터 증강의 기법을 사용하여 데이터를 증폭하였다. 이러한 방법으로 처음 수집한 데이터 1,000장에서 약 3,000장으로 데이터셋의 수를 크게 증가하였다.

위와 같은 방법으로 초기 데이터셋을 수집 후 [그림 4]와 같이 Labelimg를 활용하여 라벨링 작업을 진행하였다. 이 과정에서 작업자가 많을수록 일관되고 정확한 라벨링이 어려워지기 때문에, 정확한 Bounding Box를 구현하기 위해 2가지의 규칙을 정하였다. 첫 번째로 객체의 이미지를 모두 포함하되 빈틈을 20픽셀 이하로 최소화하였고, 두 번째는 규칙은 겹침 현상이 발생한 경우 전체 범위를 추정하여 지정하는 것이다. 라벨링 하기 위한 클래스로는 3가지를 선정하여 상자(box), 파손된 상자(d_box), 상자가 아닌 것(n_box)로 구별하였다. 각 데이터셋의 초기 데이터셋은 그림5, 그림6, 그림 7과 같다. 라벨링한 데이터를 txt파일로 저장하였고, 각 라벨링 데이터는 별도의 공유 저장소에 업로드 하였다.

그림 4. Labelimg를 통한 라벨링 과정

그림 5. box의 초기 데이터셋

그림 6. d_box 초기 데이터셋

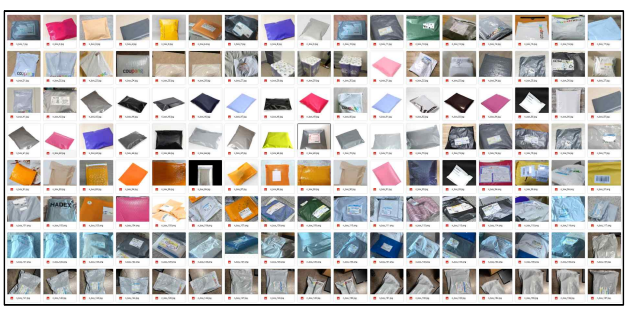

그림 7. n_box의 초기 데이터셋

3.2 YOLO를 통한 모델 개발

[2] 수집한 데이터셋을 기반으로 [그림 8]을 참고하여 YOLO 학습을 진행한다. YOLO는 one-stage object detection 딥러닝 기법으로 물체를 한 번에 인식할 수 있는 매우 빠른 속도의 추론과정으로 처리되는 알고리즘이다. 본 설계에서는 학습을 위한 기반 환경을 설정하고, YOLO로 모델 학습을 진행하였다. 초기에 데이터셋의 훈련데이터(Training data)와 검증 데이터 (Validation data)를 9:1의 비율로 나누어 학습하였다. epoch을 50, 100, 150으로 나누어 진행하였으며 epoch을 50으로 진행한 경우 100으로 진행한 경우에 비해 정확도 낮았고, 150으로 진행한 경우 클래스를 판단하는데 걸린 시간보다 정확도가 크게 높지 않아 epoch을 100으로 설정. 각 epoch 마다 call back을 진행하여 모델 파일을 저장하였다.

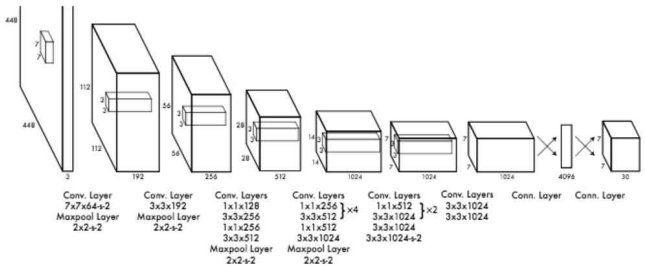

그림 8. YOLO 모델

이후 YOLO 객체 생성을 통한 Bounding box를 구현하여 YOLO 객체를 생성하여 keras-yolo에서 image 처리를 주요 PIL로 수행하였다. 생성된 box_yolo 객체를 통해 image detected를 수행하였다. [그림 9]와 같이 test 이미지를 Bounding box의 위치값을 왼쪽 상단, 오른쪽 상단, 왼쪽 하단, 오른쪽 하단 4가지 영역역으로 나누어 해당하는 위치에 택배 상자의 이미지가 존재한다고 설명하는 방식이다.

그림 9. 바운딩 박스 결과 이미지

Ⅳ. 모델 평가 및 구현 결과

4.1 모델 평가

본 설계에서 빠르고 정확한 YOLO모델의 평가와 테스트를 위해 box, d_box, n_box로 나누어 진행하였다. 초기에는 학습된 모델을 사용하여 image detected 수행해본 결과 [그림 10]과 같이 accuracy(정확도)는 93.33%에 도달하였다. 학습에 사용되지 않은 이미지를 통해 테스트를 진행하였다. 해당 사진은 웹캠을 연결하여 촬영한 이미지로 테스트 이미지를 detected 하여 클래스의 판별도를 확인한 결과, 세 가지 클래스(box,d_box,n_box)의 판별도가 높은 것을 확인하였다. 실시간으로 촬영으로 판단하는 결과 box는 95%, d_box는 72%, n_box는 71%로 인식되었다.

그림 10. 모델 평가 및 구현에 대한 정확도

4.2 구현 결과

상자(box), 파손된 상자(d_box), 상자가 아닌 것(n_box) 세 가지의 클래스로 나누어 약 3,000장의 데이터를 수집하였고, 수집한 데이터를 전처리하고 라벨링 작업을 직접 수행하여 택배 상자 초기 데이터셋을 구성하였다. 구성한 데이터셋을 epoch 100으로 학습을 시킨 결과, loos 값은 초기 34에서 11까지 감소하였으며 accuary의 경우 93.33%에 도달하였다. Tensorflow, Keras, Yolo를 사용하여 학습시킨 결과를 토대로 학습시키지 않은 임의의 데이터에서 물제(box)가 탐지 여부를 파악하였다. Webcam을 통해 실시간으로 상자의 상태를 확인하여 촬영 후 이를 공유폴더에 업로드하여 이를 서버에서 object detected 하여 파손의 유무를 확인하였다. box와 d_box의 경우 거의 오차나 오류 없이 잘 탐지 되었지만, n_box의 경우 box로 인식되는 경우가 발생하였다. box와 d_box의 두가지 클래스로도 충분히 파손 상자를 골라낼 수 있다고 판단하였다. n_box의 클래스가 제거하는 것이 파손된 상자를 판단하는 부분에서 본 설계의 취지와 더 적합하다고 판단하였다.

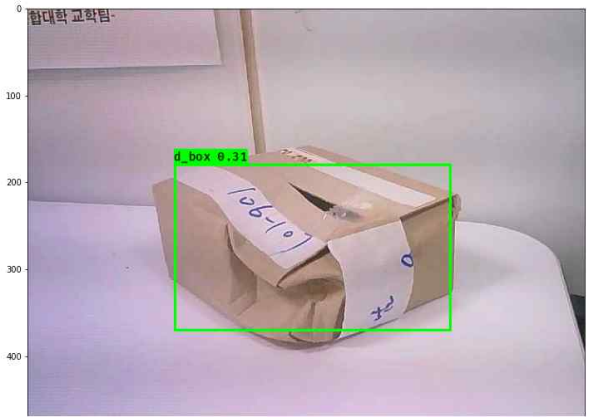

그림 11. 실시간 촬영 테스트 이미지

그림 12. n_box 결과 이미지

V. 기대효과 및 향후 개선 방향

5.1. 기대효과

본 설계는 카메라를 이용하여 택배의 파손 유무를 알려주는 기능이 탑제하여 이를 이용하여 택배가 파손된 것을 사전에 파악하여 회사의 피해를 최소화할 수 있다. 서론에 언급한 바와 같이 택배 사용량이 증가하고 있는 추세이다. 택배의 양이 많아 사람이 직접 택배를 관리하기 용의하지 않아 보통 파손될 때 택배 기사나 택배 회사측에서 소비자에게 배상을 하게되는 일이 종종 발생하게 된다. 택배의 양이 많아 파손 택배를 파악하기 어려워 택배를 받는 소비자는 시스템을 활용하면 안심이 되고 택배 회사 측에서는 불필요한 비용이 소비되지 않는다. 소비자 또한 사전에 파손 유무를 알 수 있어 번거로움을 방지할 수 있고 시간의 절약도 된다. 택배 회사 측에서 이 시스템을 활용한다면 소비자에게 편리한 시스템을 제공 받을 수 있다. 딥러닝을 통한 택배 파손 분류이기 때문에 인력과 시간 절약이 가능하고 데이터에 대한 꾸준한 모니터링이 필요치 않고 수많은 택배에 대한 검수할 필요가 없게 된다.

5.2. 향후 개선 방향

본 설계에서는 딥러닝 분야 중 하나인 객체 인식 기술의 모델 중 하나인 YOLO를 활용하여 택배 상자의 파손 유무를 판단하는 시스템에 대해 기술하였다. 이 시스템을 소비자가 손쉽게 확인 할 수 있도록 별도의 앱을 안드로이드 스튜디오를 통해 제작하여 배포하고, 별도의 공유파일을 제작하여 딥러닝 네트워크를 통해 서버로 활용하고, 소비자에게 제공한 앱을 클라이언트로 사용하는 방법을 고안하고 있다. 웹캠을 통해 이미지 파일을 서버를 통해 클라이언트에게 제공한 앱으로 사진과 파손 유무가 전송되어 실시간으로 확인하고, 파손으로 판별된 경우 앱의 알림을 주는 형식으로 개선할 예정이다. 또한 n_box의 데이터셋 부족으로 인식률이 낮아 추후 초기 데이터셋를 추가적으로 수집하여 YOLO에 대한 인식률을 보안·수정할 예정이다.

참고문헌

[1] SOCAR Tech Blog, 딥러닝, https://tech.socarcorp.kr/data/2020/02/13/car-damage-segmentation-model.html

[2] techblog, YOLO, https://techblog-history-younghunjo1.tistory.com/186

[3] 손진호, 김창욱, 제조 공정에서의 실기간 불량 탐지를 위한 딥러닝 모델 적용 연구 – 라벨 인쇄 공정에서 불량 탐지 사례, 연세대학교 대학원, 융합기술경영공학과, 2021, 8

[4] GitHub, Keras, https://junha1125.github.io/blog/pytorch-docker-git/2020-08-20-GPU-yolo3Raccoon/

[5] Hi PolarBear, Pytorch, https://hipolarbear.tistory.com/19

[6] Yolov3, https://pmin-a.tistory.com/13

[7] honeycomb-makers, 인공지능, https://honeycomb-makers.tistory.com/3

[8] Google Patents, 딥러닝, https://patents.google.com/patent/KR102096386B1/ko

[9] zdnet, 딥러닝, https://zdnet.co.kr/view/?no=20211122175433

[10] 이용한, 김영섭, 객체 검출을 위한 CNN과 YOLO 성능 비교 실험, 원광대학교, 디지털콘텐츠공학과, 단국대학교, 정자전기공학부, 2020. 3

[11] 전성우, YOLO 기반 차선 검출 시스템, 배재대학교 대학원, 컴퓨터공학과 컴퓨터공학전공, 2020.12

[12]S. Y. Lee, J. M. Kim, C. M. Ji and M. P. Hong, "A Study on DoS Attack and Response Technology Trends in V2V Communication Environment of Self-driving Vehicles," J ournal of Information Security Society, vol. 30, no. 2, pp. 41-48, Apr. 2020

[13] G. Han, S. Jinpeng and C. Zhang, “A method based on Multi-Convolution layers Joint and Generative Adversarial Networks for Vehicle Detection,” KSI I Transactions on Internet and Information Systems (TI IS), vol. 13, no. 4, pp. 1795-1811, Apr. 2019

[14] Y. H. Shim and S. H. Kwon, "Trends of Standardization of Autono mous Driving System," Auto J ournal, vol. 41 no. 9, pp. 63-66, Sep. 2019