

보행자를 위한 장애물 위험도 분류기

위성민^{0*}, 최용호^{**}, 권영서^{*}, 김남훈^{**}, 강산희^{*}
지능기전공학부 무인이동체공학과*, 스마트기기^{**}

요 약

최근 영상정보를 기반으로 한 연구가 활발하게 진행되면서 이에 적용되는 다양한 알고리즘의 개발을 위한 기초 연구가 진행되고 있다. 본 연구에서는 영상정보를 이용하여 도로정보, 물체 정보와 거리정보를 융합하여 보행자를 위한 기초 장애물 위험도 분류 시스템을 구성하였다. 제안된 알고리즘을 확인하기 위하여 시스템을 구성한 뒤 테스트 영상을 이용하여 실험을 진행하였고, 그 결과 세 가지 정보를 기반으로 한 위험도 분류가 가능한 것을 확인할 수 있었다.

1. 서론

최근 시각정보를 기반으로 한 알고리즘이 많아지면서 응용 사례들이 확대되고 있다. 대표적으로 도로검출, 물체검출과 거리추정이 이에 해당된다. 이러한 기법들은 Neural networks의 확장과 여러 가지 학습 방법론으로 실제 생활에 사용될 만큼 발전되었다. 하지만 영상정보를 기반으로 한 대부분의 연구는 차량과 도로주행 중심의 무인 자동차 알고리즘 모델 설계 및 구현이었다. 이 연구에서는 기존의 연구들과는 다르게 보행자의 입장에서 부주의로 인한, 혹은 돌발 상황에 대한 부상을 사전에 차단하기 위하여 사람, 고깔, 갯돌과 같은 장애물들의 위험도를 수치로 나누어 분류해보았다. 따라서 본 연구에서는 세 가지 기법을 융합한 알고리즘을 제안한다. 본 논문의 구성은 도로판별, 물체검출, 깊이 추정의 기법에 대하여 소개하고 다음으로 세 시스템을 동시에 적용한 시스템에 대해 설명한 후 결론을 맺는다.

2. 영상정보 기반 위험도 판별 알고리즘

보행자를 위한 ~ 따라서 영상정보를 기반으로 한 위험도 판별 알고리즘을 개발하였다. 도로정보 판별, 물체 검출, 깊이 추정의 세 가지 정보를 융합하였다.

도로정보 판별 시 000을 사용하여 어떻게 어떻게 해서 인도와 차도를 구분 하였다. 물체 검출 시 영상 데이터와 구글에서 크롤링한 이미지 데이터를 사용하였다. 데이터들은 전처리한 후 VOTT(Visual

Object Tagging Tool)를 이용하여 정답 데이터셋으로 가공되었다. 이를 MobileNet v2를 적용한 SSD(Single Shot Detector)를 이용하여 학습 시킨 후 다른 교내 영상으로 테스트해보았고 결과적으로 검출된 물체들의 좌표(x,y,w,h)를 깊이 추정 단계로 인가하였다. 다음 깊이 추정 단계에서는 영상의 깊이를 추출하여서 이를 이용하여 장애물과 보행자 사이의 상대적인 거리를 나타내었다.

2.1 도로 정보 판별

현재 도로 검출은 주로 분할(Segmentation) 방법론을 이용하여 이루어졌다. 그러나 최근 발표된 Free Space detection 방법론에 의하면, 자동차와 차선을 감지한 후 이 부분을 제외한 영역을 Free space로 인식하여 분할한다. 본 연구에서는 이 분할된 영역 내부에 검출된 물체가 포함될 경우 위험도에 가중치를 부과하여 위험도 고려 대상에 포함시키고, 반대의 경우 그렇지 않게 하도록 구조를 설계하였다.





2.2 물체 검출

2.2.1 데이터

물체 검출을 하기 위해서는 우선 학습시킬 데이터가 필요하다. 구글에서 크롤링한 이미지 데이터에 모델의 성능을 높이기 위해 세종대학교 교내를 촬영한 영상 데이터를 추가하여 데이터 셋을 구성하였다. 수집한 데이터를 전처리 후 주석 및 라벨링 도구인 VoTT를 이용하여 정답 데이터 셋으로 가공하였다. 본 도구는 다양한 형태의 데이터 포맷을 지원하며, 주석 파일의 포맷은 json으로 저장된다. 본 연구에서는 Tensorflow API를 이용하여 학습하므로, 이미지와 주석을 하나의 파일로 통합하여 학습할 수 있는 TFrecord 포맷으로 추출하였다. 장애물의 클래스와 각 클래스의 데이터 개수는 표 1과 같다.

표 1. 데이터 셋 클래스

클래스 이름	크롤링/영상 데이터 개수
tree	2524/195
construct_cone	1928/138
street_lamp(가로등)	1546/100
unknown(사람)	- /1715
bicycle(차량)	- /524

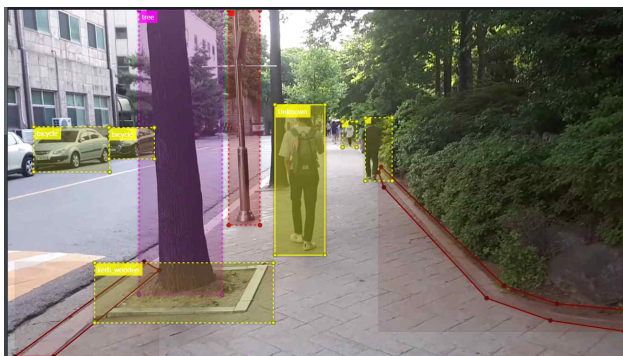
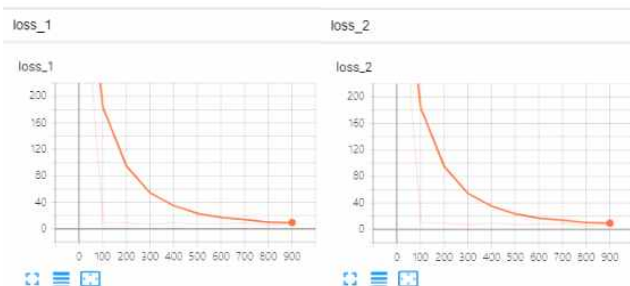
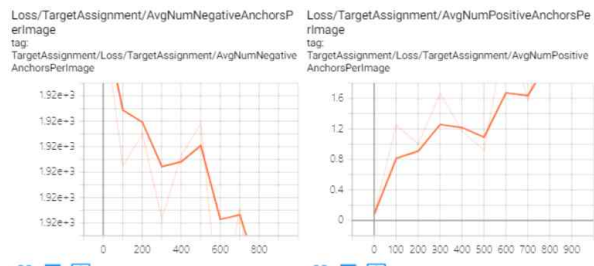
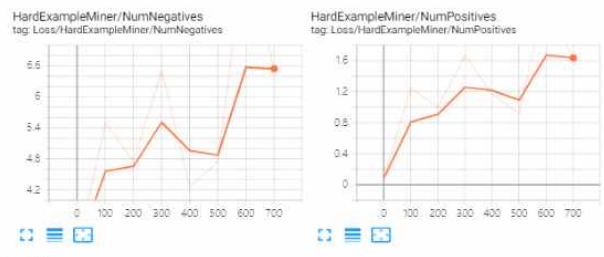


그림1. VOTT를 이용한 데이터 가공

2.2.2 학습 및 검출과정

본 연구는 고가의 GPU 장비 없이도 일반 학생들이 쉽게 학습할 수 있는 Google Colab을 이용하여 학습하였다. 학습에 앞서 데이터 셋을 사용하기 위해 Google Drive에 업로드하거나 로컬 저장소로부터 파일을 직접 업로드한다. 학습에 Tensorflow API를 이용한 물체 검출 데모 코드[4]를 참조하여 학습을 진행한다. Tensorflow model zoo[5]에서 사전 학습된 모델을 골라 대입한다. 본 연구에서는 ssd_mobilenet_v2를 사용하였다. 정규표현식을 이용하여 파이프라인 내의 학습 단계 수, 클래스 수, learning rate 등을 수정한다. Colab은 원격 호스트 특성상 tesnorboard 지원이 공식적으로 이루어지지 않는다. 따라서 ngrok를 이용하여 포트를 우회하여 실행한다. 다만, 코드와 호스트 시스템 불안정성으로 인해 실험적으로 지원되는 기능이기 때문에 안정적이지 않아 권장하지 않는다.



학습이 끝나면 미세 조정된 모델을 저장하고 freeze한다. 이후 테스트 이미지 또는 영상을 업로드하여 진행하면 이미지와 감지된 좌표가 출력되

며, 이를 통합하면 다음과 같은 결과를 얻을 수 있다. 감지된 좌표 변수는 다음 모델로 인가하여 진행한다.



그림2. 물체 검출 결과

2.3 거리 추정

2.3.1 모래시계 모델(Hourglass model)

향상된 Depth의 기술에는 Convolution model이 집약되어있다. 이 기술의 이름은 Hourglass model로 한글로는 모래시계 모델이다. 위 모델은 용어 그대로 다음 그림2와 같은 모래시계의 모형을 한 모델을 띄고 있다. 기존의 Detection 기술에서 잊힌 축소 Convolution 모델인 VGG 기술에 다시 원형으로 복귀시키는 다른 모델과는 다른 semantic segmetation의 네트워크이며, deconvolution, unpooling을 통하여 특징을 더 정확하게 검출해내는데 의의가 있다. 합성곱 모델인 모래시계 모델을 이용하여 depth map의 feature map을 convolution, pooling, deconvolution, unpooling을 거쳐 feature map을 뚜렷하게 하여 기존의 depth보다 밝기 차이로 인한 뚜렷한 Depth를 추출하여 이를 이용하여 장애물과의 대략적인 거리를 나타내었다.

2.2.1 거리 추정

Object Detection에서 받은 GT(Ground Truth)를 이용하여 중점을 찍었다. 이 중심점에는 오차 범

위나 더 튀어나온 부분의 밝은 밝기가 빠질 우려가 있기 때문에 그림3과 같이 중점 부분에서 오차 범위 ± 25 정도의 50x50크기인 픽셀을 주어 가장 밝은 값(밝은 점)을 검출하여 더 정확한 장애물과 보행자의 거리를 나타내었다.

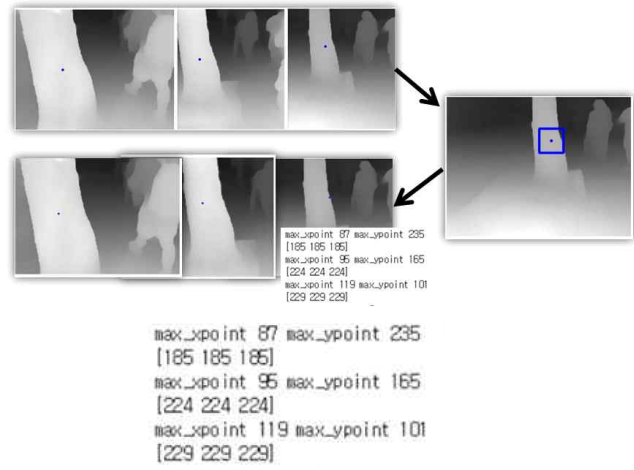


그림 3. distance estimation course

3. 실험 결과 및 분석

본 연구에서는 도로정보, 물체인식, 거리정보를 이용한 보행자 전용 장애물 검출 및 위험도 분류를 하나의 시스템으로 통합하였고, GUI(Graphic User Interface)를 구성하여 각각의 시스템의 출력을 가시적으로 확인하였다. 위험도 분류를 위한 기초 시스템의 블록도는 그림 4와 같다.

그림5는 본 연구에서 구현한 GUI이다. 통합한 시스템들은 모두 tensorflow기반으로 이루어졌기 때문에 python GUI framework인 Pyqt를 사용하여 시스템을 구성하였다. 영상의 출력은 pyqt의 위젯을 이용하여 도시하였다. 그림5의 위의 3개의 영상은 각각 3가지의 방법론을 테스트 영상에 적용하여 출력된 화면이고 밑의 큰 화면은 3가지의 방법을 통합한 알고리즘을 적용하여 출력한 화면이다. 위험도가 높은 장애물일 경우 붉은 색 박스와 거리정보, 물체 정보, 도로상의 위치 여부를 나타내어 주고 위험도가 작을 경우 푸른 색 박스로 위험도를 표현하였다.

저희를 낚아주시고 길러주신 부모님 감사합니다.

참고문헌

- [1] Towards End-to-End Lane Detection: an Instance Segmentation Approach, IV, 2018
- [2] Learning the Depths of Moving People by Watching Frozen People
- [3] Spatial As Deep: Spatial CNN for Traffic Scene Understanding, AAAI2018
- [4] object_detection_demo (Tony607, Github)
- [5] Tensorflow model zoo (Tensorflow, Github)
- [1] Free-space-detection (balaji-kss, Github)

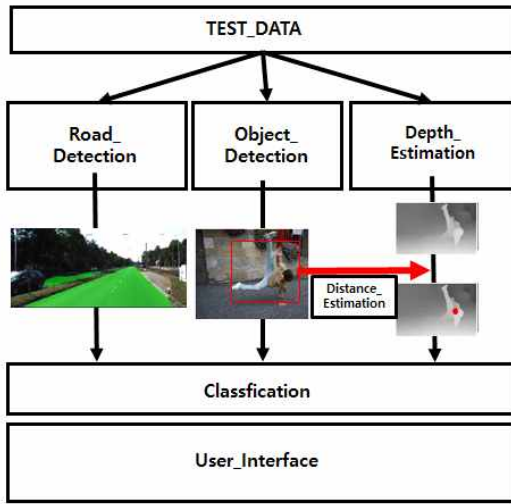


그림4. 전체적인 시스템 블록도



그림5. 통합시스템 GUI

4. 결론

본 논문에서는 영상정보를 기반으로 하여 보행자의 시야범위 내에 들어오는 장애물은 검출한 후 검출된 물체 중 전방 관심영역 내에 존재하는 장애물에 대해 위험도를 분류하는 알고리즘을 설계하였고 해당 장애물의 위험도에 따라서 경고하는 시스템을 구현하였다. 제안한 시스템을 테스트 영상을 통하여 모의실험을 해본 결과 도로정보와 물체검출, 거리정보를 통하여 장애물의 위험도 분류의 출력력을 확인하였다. 본 연구를 바탕으로 장애물 위험도 분류기는 작게는 교내의 장애물 위험도 분류기지만 추가적인 연구를 통하여 현재 급속하게 진행되고 있는 고령화 사회에서 노안으로 인한 불편함을 겪는 사람들에게 시각장애인의 지팡이와 안내견과 같은 새로운 시각적 보조도구 수단까지 광범위하게 사용될 수 있을 것으로 기대된다.

감사의 글