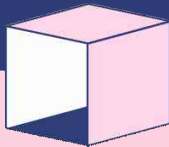
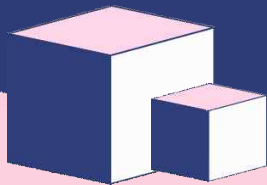




저시력 시각장애인을 위한 보행 안전 프로그램

김병찬 박종훈 이규연 이학현 장석규 - 모도리들

2039.06.30



거리 보행 안전 프로그램

장애물을 감지하고 거리에 따라 다른 색으로 나타내는 프로그램

1. 장애물 감지 (Object detection)
2. 장애물과의 거리 구하기
3. 장애물을 색으로 표현하기

장애물을 감지하는 방법은?

CNN (이미지 인식 신경망)을 이용해 장애물 이미지 학습

<이미지 학습>

이미지를 1차원 배열로 변환 후 신경망에 넣기

이미지에 2차원 필터 적용

합성곱 연산 및 풀링 연산을 통한 신경망 구성

CNN 신경망 구성 (게임 화면을 학습시켜 어떠한 게임인지 맞추기)

Model: "sequential_2"

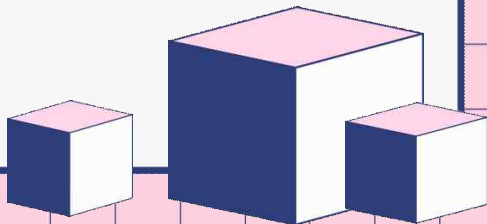
Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 254, 142, 16)	448
max_pooling2d_6 (MaxPooling2D)	(None, 127, 71, 16)	0
conv2d_7 (Conv2D)	(None, 124, 68, 32)	8224
max_pooling2d_7 (MaxPooling2D)	(None, 62, 34, 32)	0
conv2d_8 (Conv2D)	(None, 60, 32, 32)	9248
max_pooling2d_8 (MaxPooling2D)	(None, 30, 16, 32)	0
flatten_2 (Flatten)	(None, 15360)	0
dense_4 (Dense)	(None, 256)	3932416
dense_5 (Dense)	(None, 1)	257

Total params: 3,950,593
Trainable params: 3,950,593
Non-trainable params: 0

합성곱 및 풀링을 반복하고
ReLU를 활성화 함수로 사용한
연결된 신경망을 여러층으로 구성

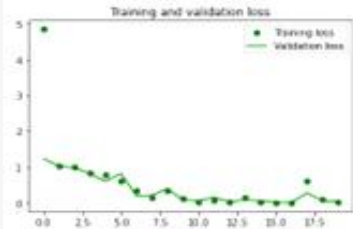
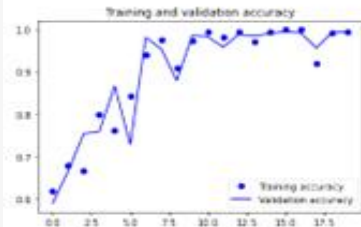
*ReLU(정류된 선형 함수) - 은닉층에서 주로 사용
SoftMax로 이루어진 출력층으로 구성

*SoftMax
softmax함수는 분류될 클래스의 수를 n 이라 할 때
 n 차원 벡터를 입력받아 각 클래스에 속할 확률 추정



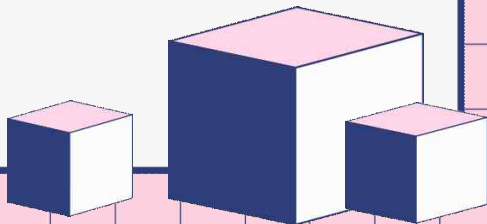
CNN 신경망 정확도 평가 그래프

60/60 [.....] - 110s 2s/step - loss: 0.0285 - accuracy: 0.9949
13/13 [.....] - 25s 2s/step - loss: 0.0443 - accuracy: 0.9944
14/14 [.....] - 24s 2s/step - loss: 0.0422 - accuracy: 0.9945



학습 진행 횟수에 따라
Training Accuracy가 상승하고
Training Loss가 감소

> 최종적으로 99.4%의 정답률을 보임

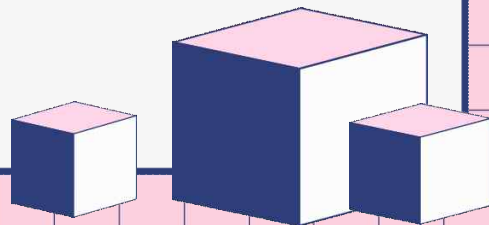


실행 결과



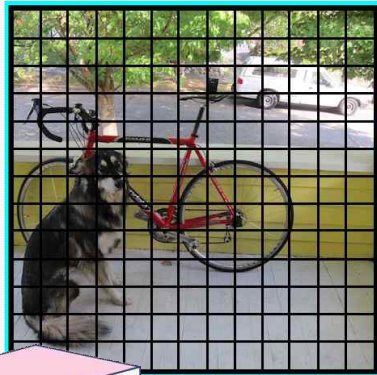
Counter Strike 100.0 %

총 3가지 게임을 학습 시켰고
게임 스크린샷을 통해 게임의 종류를
판단할 수 있었음

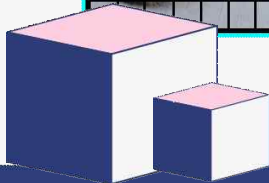


Object Detection

<TensorFlow Object Detection>

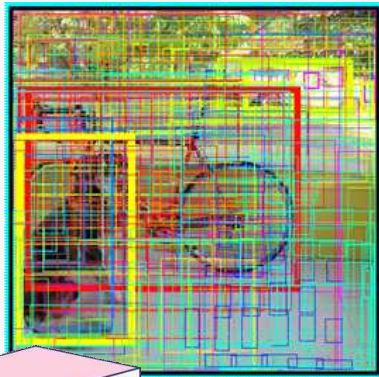


객체 인식 모델이 적용될 수 있도록
사진에서 경계상자 집합 생성



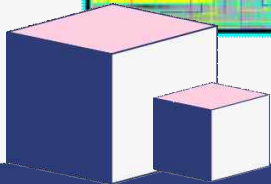
Object Detection

<TensorFlow Object Detection>



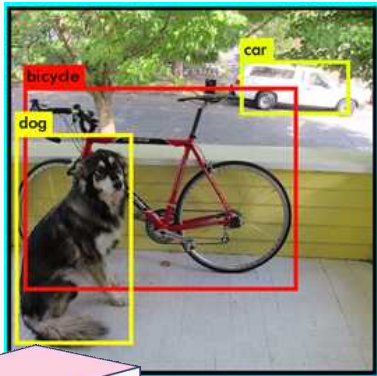
각 경계상자에서 시각적 특징을 추출해
객체인식 모델이 경계상자들을 조합

조합한 구역에 어떤 물체가 있는지 평가
및 판단

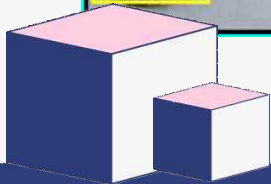


Object Detection

<TensorFlow Object Detection>



조합된 구역이 단일 상자로 결합되고
시각화를 통해 객체를 보여줌



거리를 측정하는 방법은?

OpenCV를 이용해 disparity map(시차 맵) 생성

<양안 시차를 이용>

스테레오 이미지 구하기(같은 대상에 대한 좌, 우 사진)

스테레오 이미지 2개를 이용해 disparity map 생성

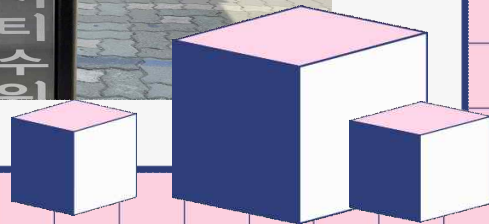
삼각측량법 이용

아주대학교 내 스테레오 이미지 구하기

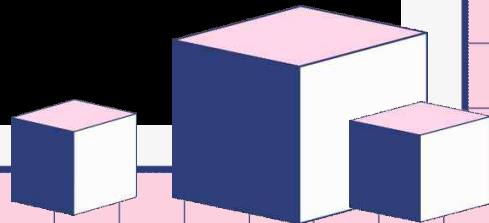
<왼쪽 사진>



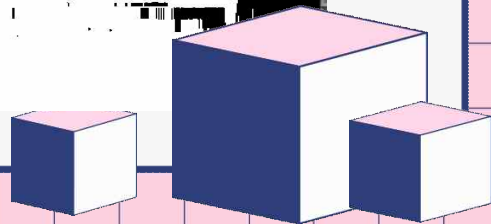
<오른쪽 사진>



두 이미지간 공통 점 찾기

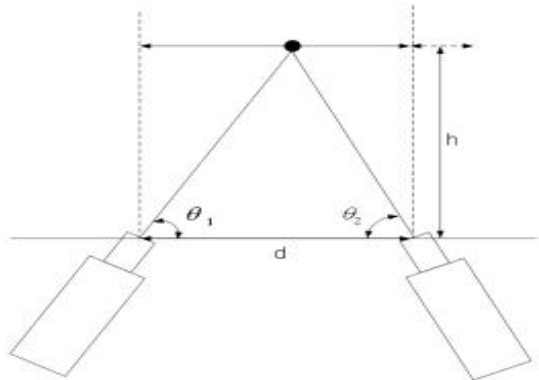


시차 맵 생성 (픽셀 당 시차 구하기)



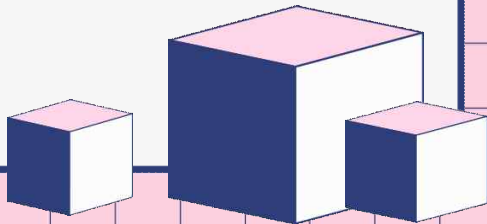
삼각측량법 이용

$$h = \frac{d}{\tan(90^\circ - \theta_1) + \tan(90^\circ - \theta_2)}$$



삼각측량의 원리를 이용해 거리를 측정

두 이미지간 시차를 얻었기 때문에
(실제 거리) = (카메라 간 거리) * (초점거리) / (시차)를
통해 얻을 수 있음

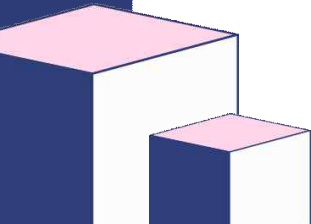


장애물 색 표시

거리에 따라 다른 색으로 표현

실제 거리 공식을 이용해 픽셀 별 거리 추출

장애물의 경계 상자를 이용해 거리에 따라 다른 색으로 표현



THANK YOU.