



BLOCK DETECTOR

WITH CNN & YOLOV2

이광록 이유리 김소희 이현경

INDEX

- I. 주제 선정 및 배경
- II. 모델 선정 및 설명
- III. 구현
- IV. 모델 적용
- V. 향후 과제

01
주제선정
배경

02
모델선정
및 설명

03
모델 구현

05
결론 및
활용방안

04
YOLO
모델 적용

1 주제 및 선정배경 주제 설명

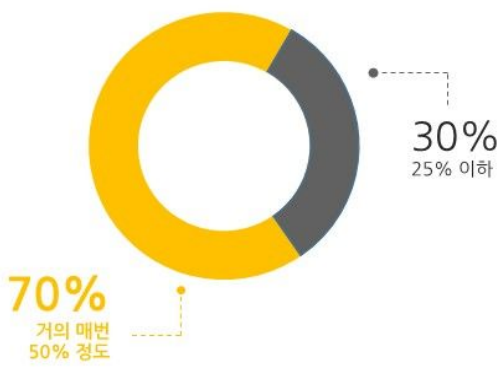


>> 시각장애인의 눈이 되어주는 점자블록,
하지만 제 역할을 하지 못하고 있다

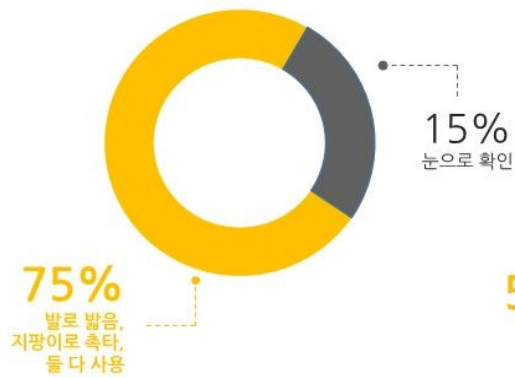
1 주제 및 선정배경 주제 설명

시각장애인 보행 요구사항 조사

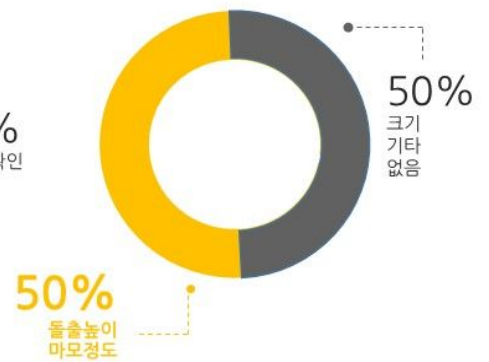
점자블록 활용



점자블록 이용법



현행 점자블록 개선요소



1 주제 및 선정배경 주제 선정 배경

점자블록이 잘못 설치되거나 길을 방해하는 경우가 존재함



점자블록이 잘못 설치된 경우



통행을 방해하는 경우



점자블록이 끊긴 경우

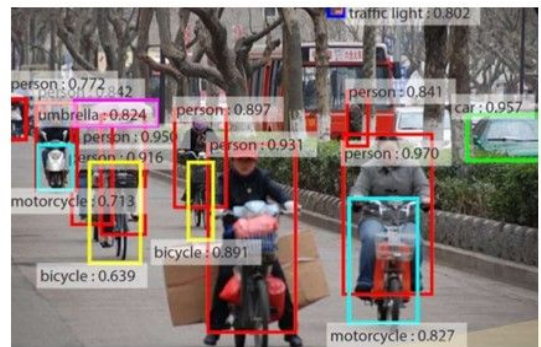
시각장애인이 혼자 외출하는 데에는 안내견 동반등의 많은 불편함이 따른다
>> 그들에게 눈과같은 역할을 해주는 장치의 필요성

2 모델 선정 및 설명 모델 선정



다수가 보유하고 있는 스마트폰

+



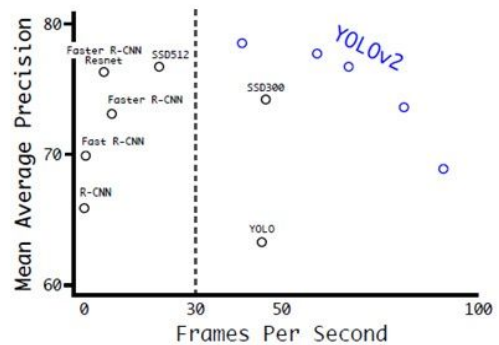
Deep learning

“Deep Learning을 통한 점자블록 탐지 시스템 구현”

2 모델 선정 및 설명 모델 선정

영상을 실시간으로 탐지하여 처리하기 위해 YOLO 사용

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16 [15]	2007+2012	73.2	7
Faster R-CNN ResNet [6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	78.6	40

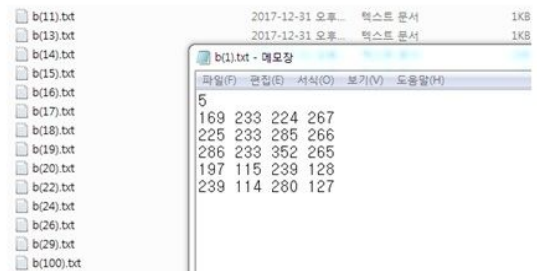
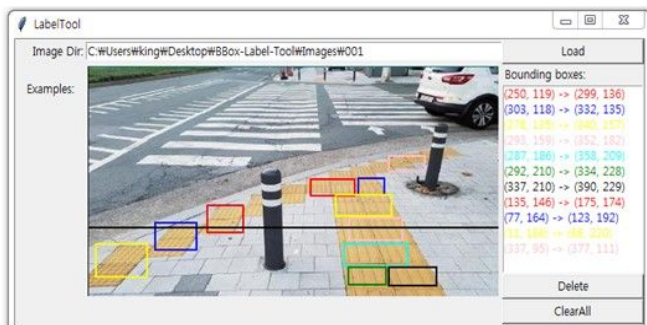


R-CNN과 같은 detection system들은 느린 속도로 인해 영상에 적용하기는 어렵다

>> YOLO는 최고성능의 빠른 detection system

2 모델선정 및 설명 구현 설명

YOLO train을 위해서는 Data annotaion 과정이 필요

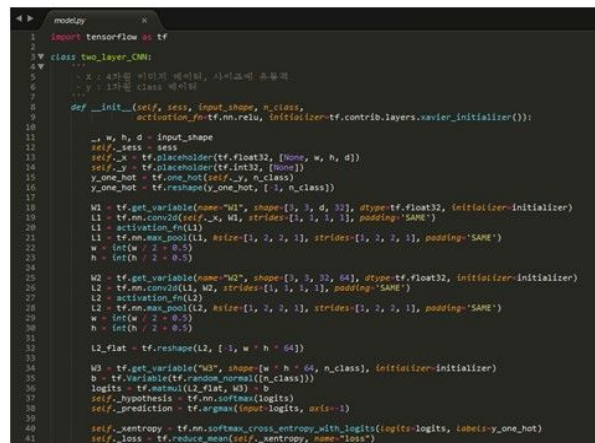
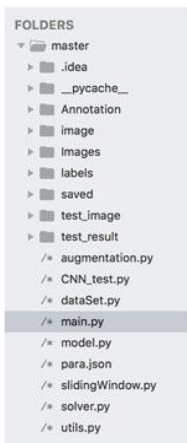


이미지의 object에 해당하는 (x, y, width, height) txt를 만들어주는 과정
Bbox-Label-Tool 등을 이용하여 직접 표시해야 함

수천장의 이미지에대한 좌표표시 과정에서 수작업이 아닌 알고리즘을 통해 자동화 방법 구현
>> 데이터를 쉽게 수정 가능, 많은 데이터로 train 가능

3 모델 구현

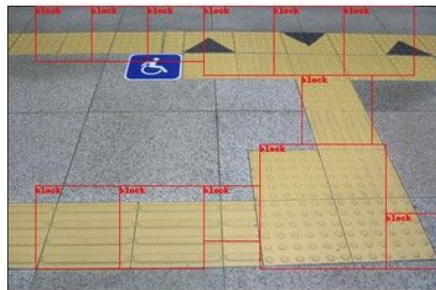
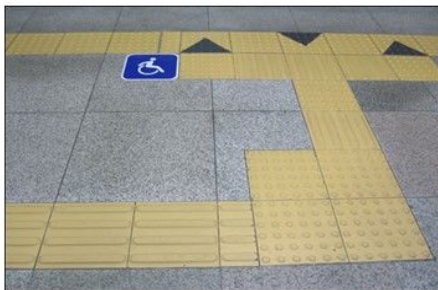
<https://github.com/krlee407/braille-block>



각자의 역할을 하는 모듈을 만들고, 모듈별로 독립적으로 코드 수정이 가능하도록 구현
 >> parameter 수정만으로 여러 실험을 할 수 있는, 대화형 인터프리터에 적합한 형태

3 모델 구현 아이디어

CNN + Image detector



b(1).txt - 메모장				
파일(F)	편집(E)	서식(O)	보기(V)	도움말(H)
5				
169	233	224	267	
225	233	285	266	
286	233	352	265	
197	115	239	128	
239	114	280	127	

R-CNN, YOLO와 같은 대부분의 detection model은 annotate 된 데이터를 필요로 한다

(Training data를 얻기 까다로움)

>> 위와 같은 model을 구현해놓으면 쉽게 training data를 만들 수 있음

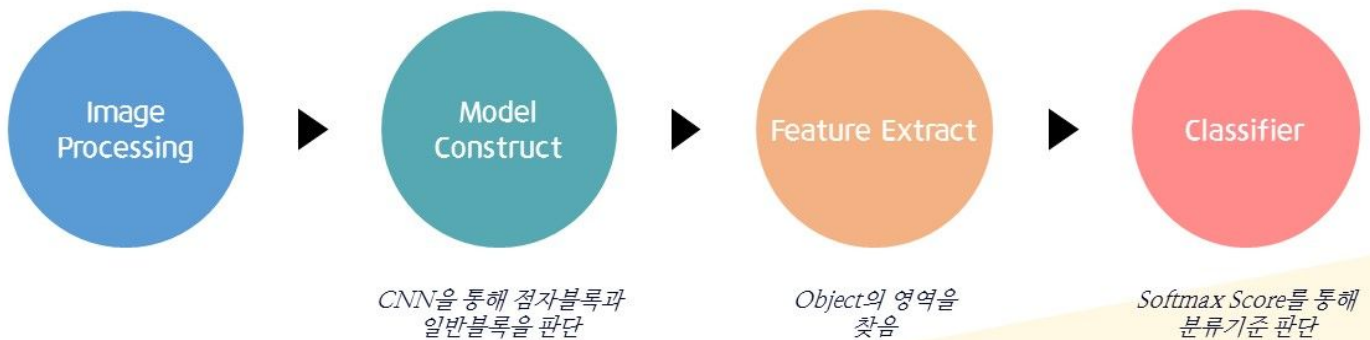
3 모델 구현 구현 절차

자동화 Data annotation 구현

Env - tensorflow1.4, Window7, GTX970

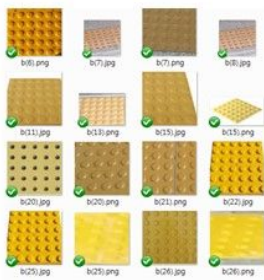
Train data - 점자블럭, 일반블럭, 자동차, 사람, 벽 등

Output data - XML 형태의 annotation data (YOLO training data)

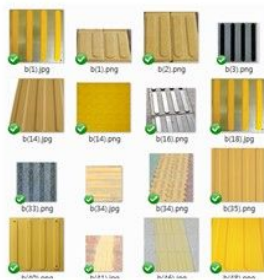


3 모델 구현 모델 구현

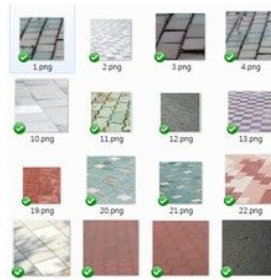
Image Processing



(dot-braille block)

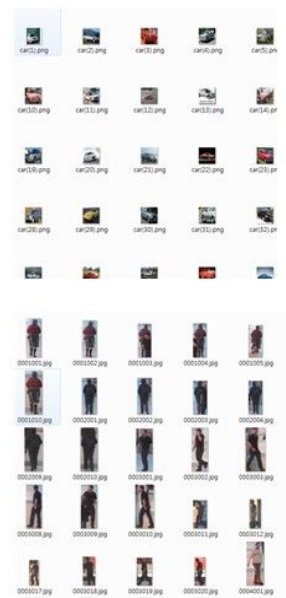


(line-braille block)



(none-braille block)

+



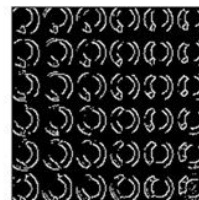
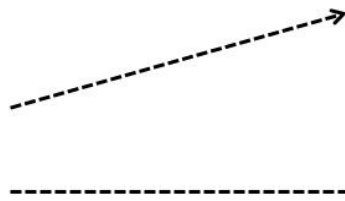
(cifar10, cifar100등에서
car, person data 수집)

3 모델 구현 모델 구현

Image Processing



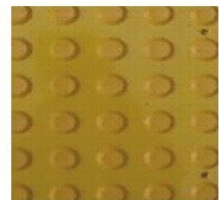
(각각의 이미지를 Histogram of
Gradient, Edge detector, Grayscale,
Augmentation)



(Edge detector)



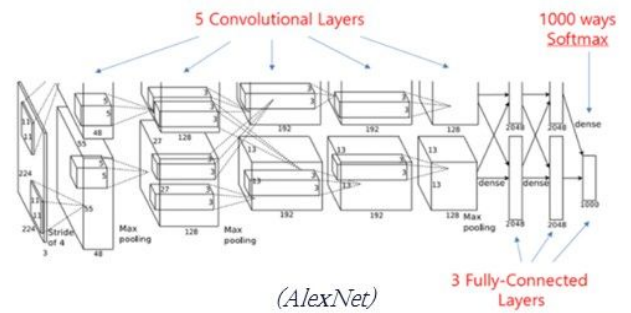
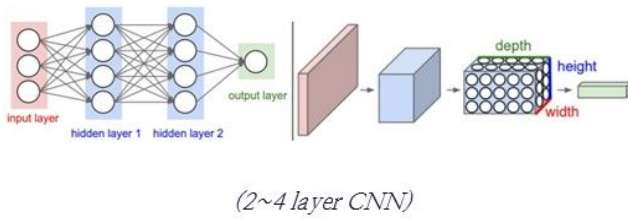
(Grayscale)



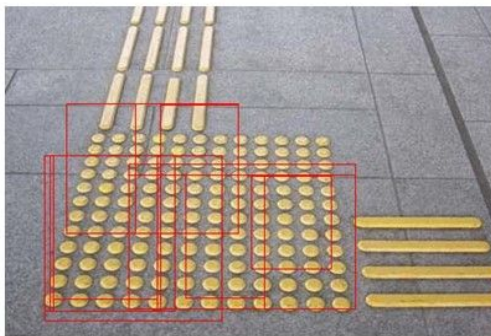
(Augmentation)

3 모델 구현 모델 구현

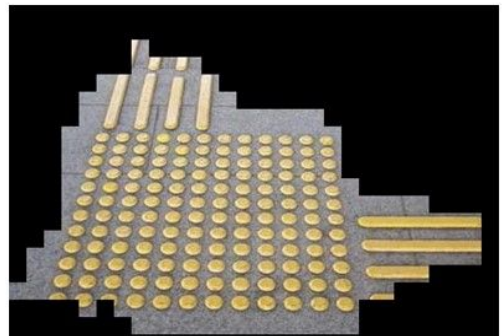
Model Construct



3 모델 구현 모델 구현



(selective search + CNN 결과)

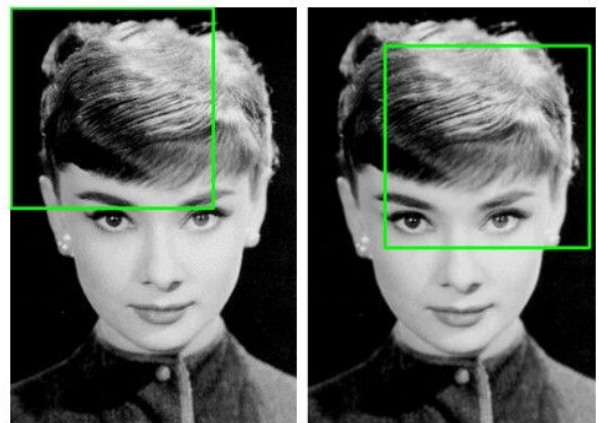
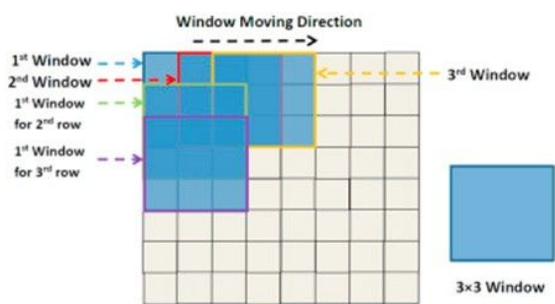


(sliding window+ CNN 결과)

실행시간은 selective search가 빠르나, 정확도는 sliding window가 훨씬 우수
>> Sliding window 사용

3 모델 구현 모델 구현

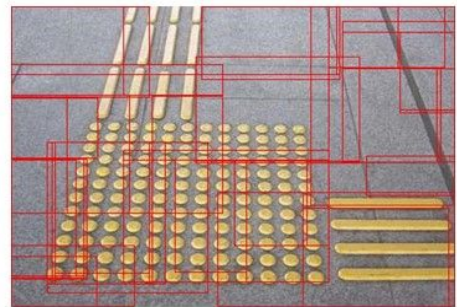
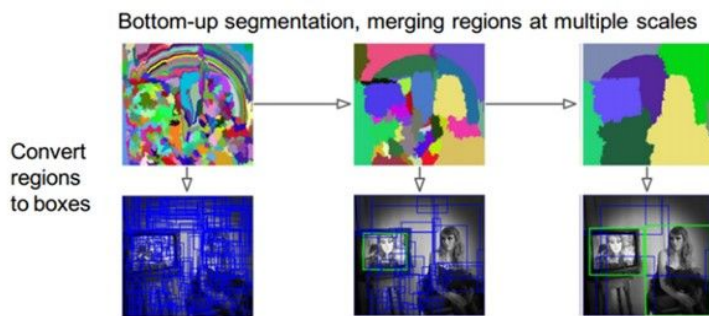
Feature Extract – Sliding window



(sliding window 적용 예시)

3 모델 구현 모델 구현

Feature Extract – Selective search



(selective search 적용했을 때 후보영역)

3 모델 구현 모델 구현



Data : Block, non-Block
Model : two layer CNN
Classifier : predict



Data : Block, non-Block
Model : three layer CNN
Classifier : predict

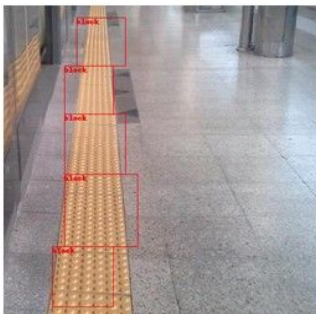


Data : Block, non-Block
Model : four layer CNN
Classifier : Softmax-Score

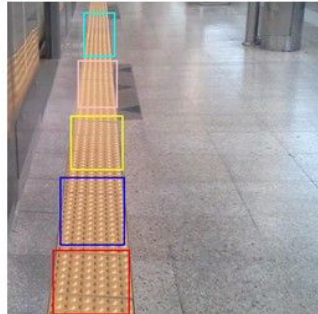
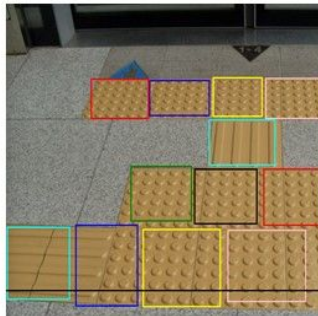
최종적으로,

Data : Block, non-block, car, human, wall
Image processing : None
Model : four layer CNN
Feature extract : Sliding window (size=64)
Classifier : Softmax score

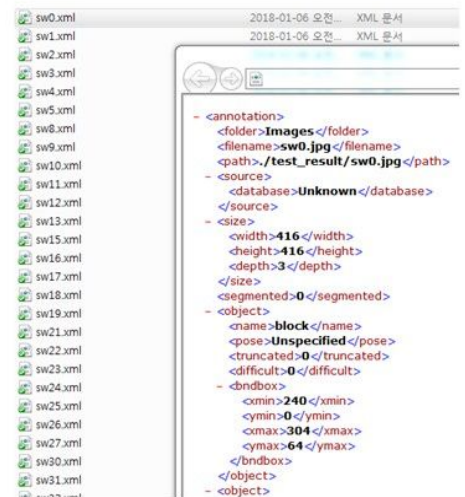
3 모델 구현 구현 결과



(model 결과)



(손으로 찾았을 때)

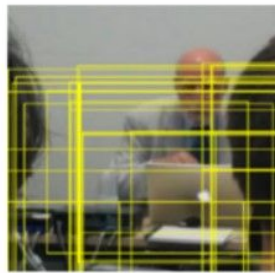


(image당 각 object의 x,y, width,height의 정보를
Xml 파일로 만들어 저장까지)

4 YOLO 모델 설명

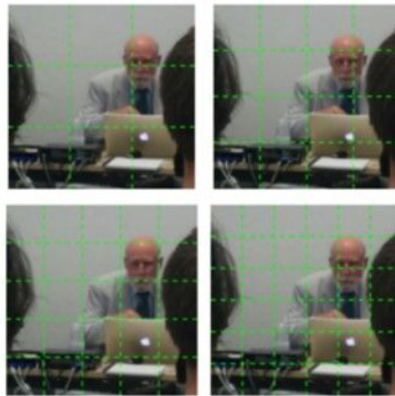


Selective Search
2.24 seconds

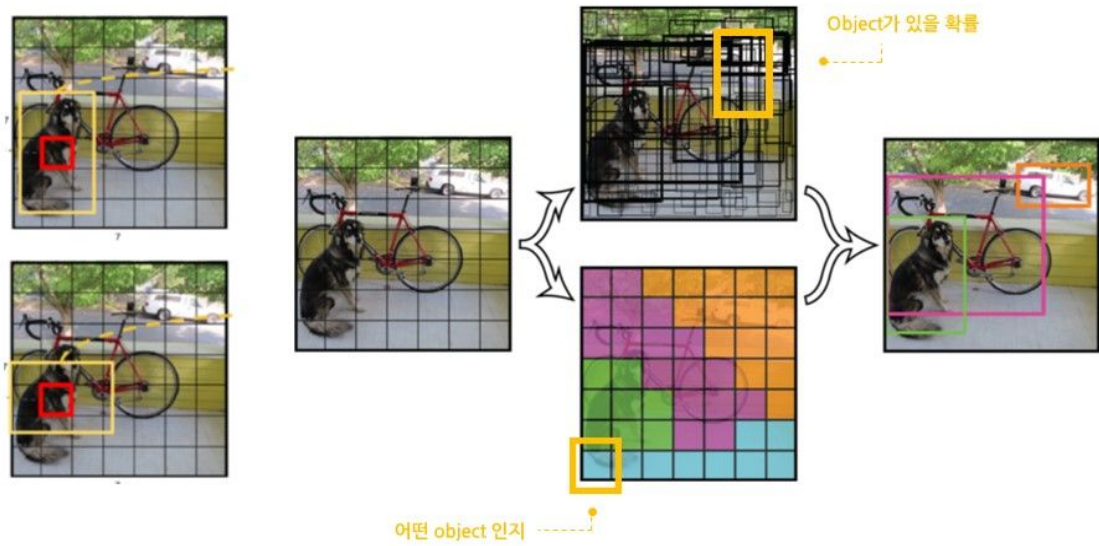


EdgeBoxes
0.38 seconds

Cheaper Alternative: **grids**



4 YOLO 모델 설명



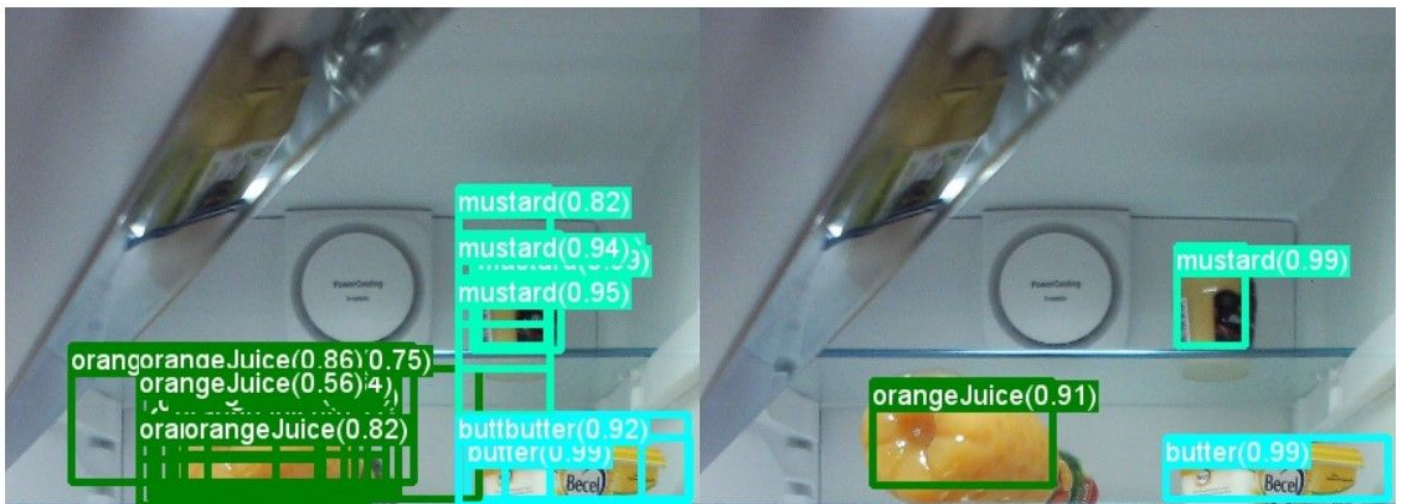
4 YOLO 개념 설명

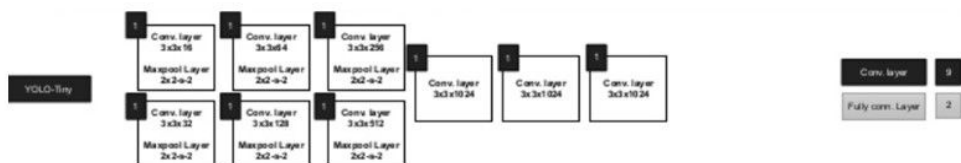
YOLOv2 – 기존의 YOLO모델 보다 성능과 속도가 더 향상된 모델

	YOLO									YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓					
new network?					✓	✓	✓	✓		✓
dimension priors?						✓	✓	✓		✓
location prediction?						✓	✓	✓		✓
passthrough?							✓	✓		✓
multi-scale?								✓		✓
hi-res detector?										✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8		78.6

성능향상을 위해 배치 정규화, High Resolution, Darknet 사용 등의 방법이 추가
>> 여러 가지 방법을 추가하면서 모델의 성능이 향상하는 것을 볼 수 있음

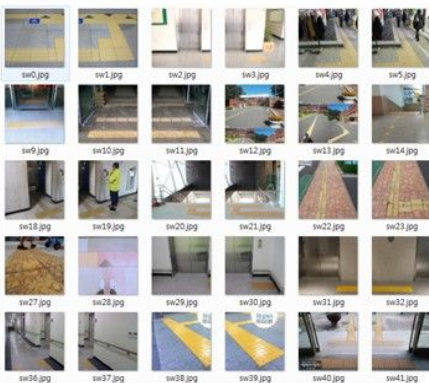
4 YOLO 모델 설명



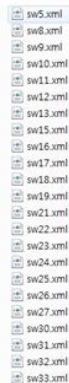


4 YOLO Input data

- Annotations
- Images
- ImageSets
- labels



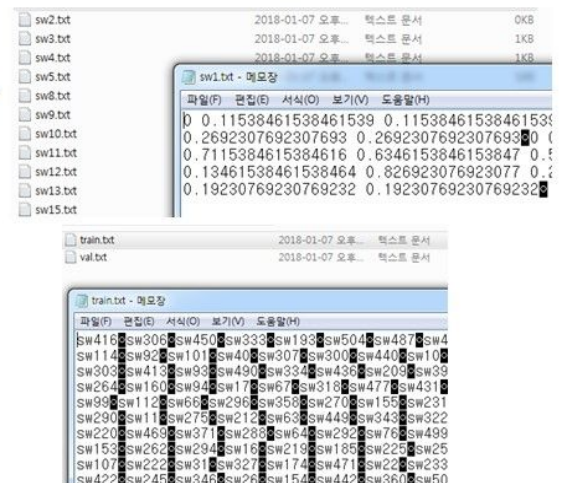
(Images)



(Annotations)

```
<annotation>
  <folder>Images</folder>
  <filename>sw0.jpg</filename>
  <path>./test_result/sw0.jpg</path>
  <source>
    <database>Unknown</database>
    </source>
  <size>
    <width>416</width>
    <height>416</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>block</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
  </object>
  <bndbox>
    <xmin>240</xmin>
    <ymin>0</ymin>
    <xmax>304</xmax>
    <ymax>64</ymax>
  </bndbox>
</annotation>
```

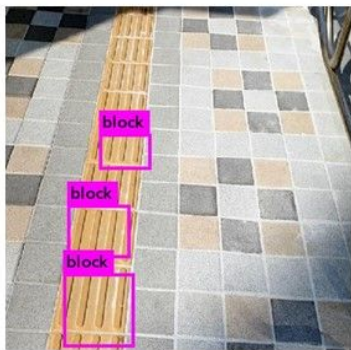
(Labels)



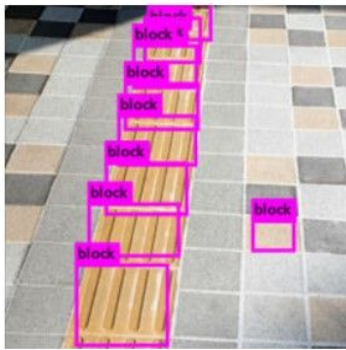
(ImageSets)

4 YOLO 모델 적용

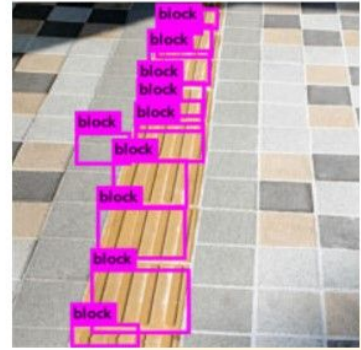
Learning rate 조정



(0.0001)



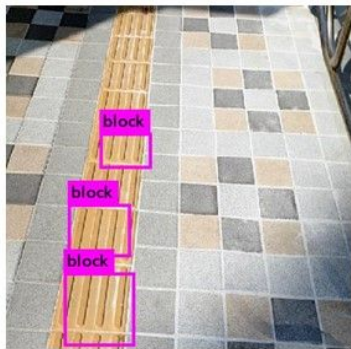
(0.005)



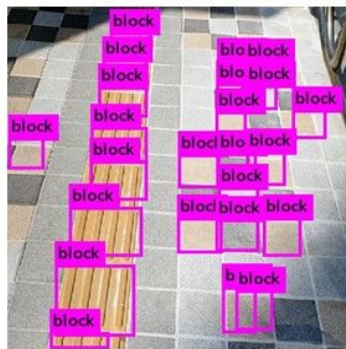
(0.001)

4 YOLO 모델 적용

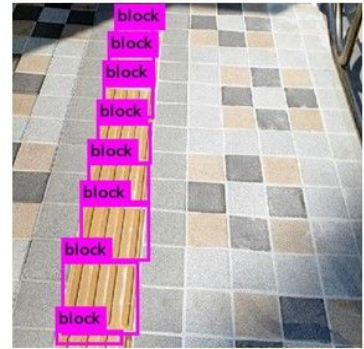
Training iteration 횟수 조정



(1000번)



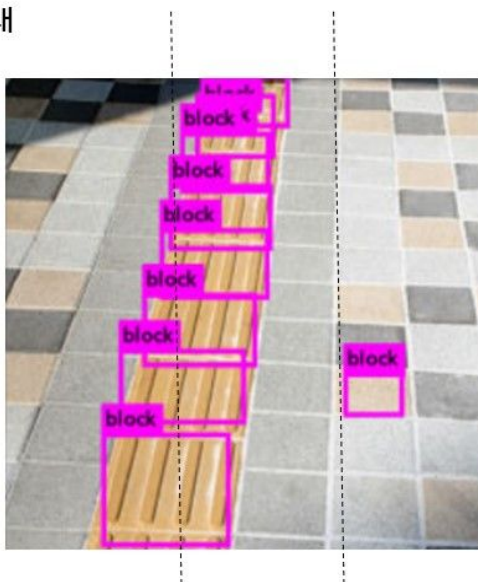
(2000번)



(20000번)

4 YOLO 모델 활용

방향 안내

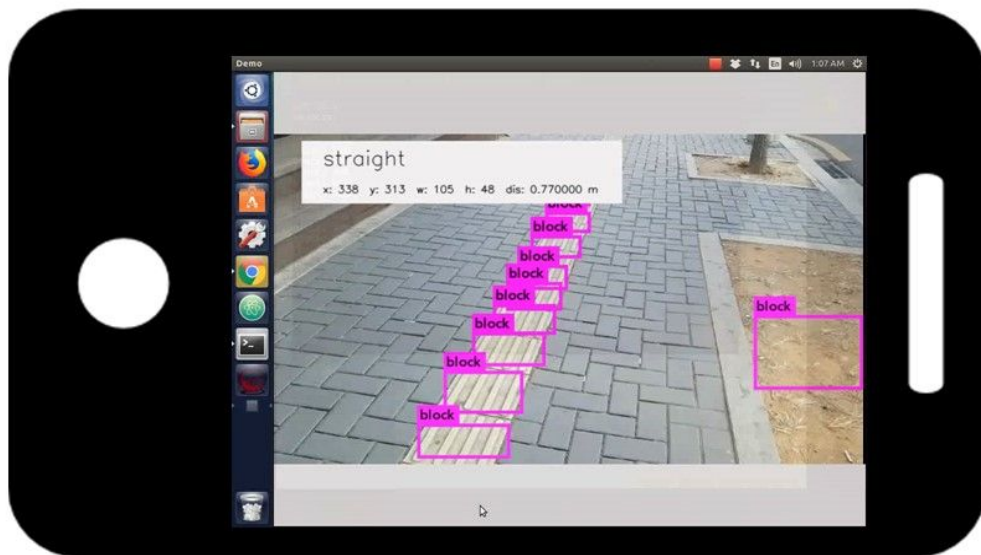


block: 52%
block: 66%
block: 69%
block: 78%
block: 78%
block: 81%
block: 52%
block: 54%
block: 90%



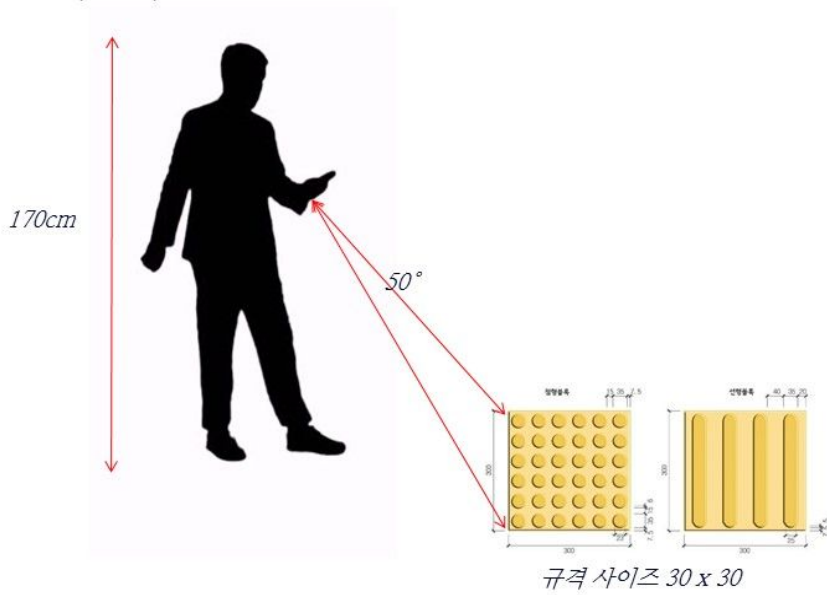
적용결과

4 YOLO 최종 결과



4 YOLO 모델 활용

방향, 좌표, 거리 안내

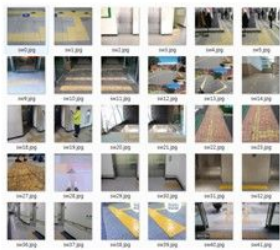


적용결과

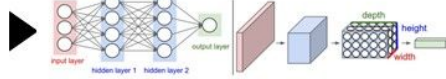
5 결론 및 활용방안

결론

최종 정리



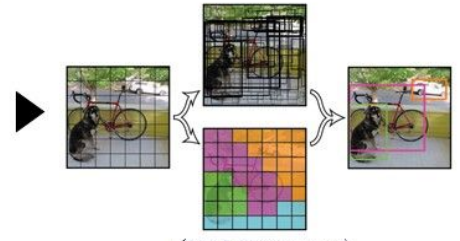
(Images)



(CNN+Image detector를 이용한
자동화 data annotation)

```
<annotation>
  <folder>Images</folder>
  <filename>sw0.jpg</filename>
  <path>./test_result/sw0.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>416</width>
    <height>416</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>block</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>240</xmin>
      <ymin>0</ymin>
      <xmax>304</xmax>
      <ymax>64</ymax>
    </bndbox>
  </object>
  <object>
    <name>block</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>240</xmin>
      <ymin>0</ymin>
      <xmax>304</xmax>
      <ymax>64</ymax>
    </bndbox>
  </object>
  ...
</annotation>
```

(annotate된 train data)

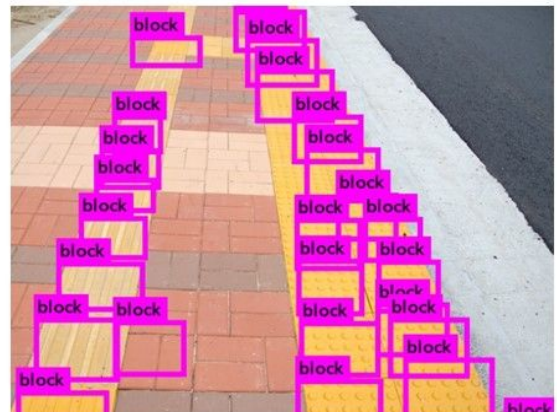


(YOLO training)

5 결론 및 활용방안 보완점



여백이 너무 많은 경우



차도가 옆에있는 경우

자동화 과정 성능 개선, YOLO 모델 성능 개선 필요

5 결론 및 활용방안 발전 방향



(구글 글래스)



(기본 카메라 앱과 Goggles를 연동한 모습)

건물, 신호등, 사람등의 많은 데이터로 train할 경우 시각장애인 전용의 내비게이션 등 활용가능
>> 조금더 편리하고 윤택한 삶 제공 가능

A photograph of a red measuring tool with a white sensor tip, positioned on a road surface marked with yellow dashed lines. The tool is oriented vertically, and its tip is in contact with the road. In the background, a person's legs and feet in brown boots are visible, slightly out of focus. The image is framed by four white corner brackets, suggesting a focus on the central subject.

Thank You

WITH CNN & YOLOV2