

퍼스널 모빌리티 카메라 기반 보조 기기 제안



18011573 컴퓨터공학과 이장후
dlwkdgn1@naver.com

진행 상황 / 변경 사항

지난 주 ;
Curve Detection
Base Code 개발

Rpi4 포팅 완료.
Google Coral 도착
Coral 포팅 완료

RPI 포팅 완료,
tf.Lite / Google Coral
이용한 Classification
예제 동작 실험 완료

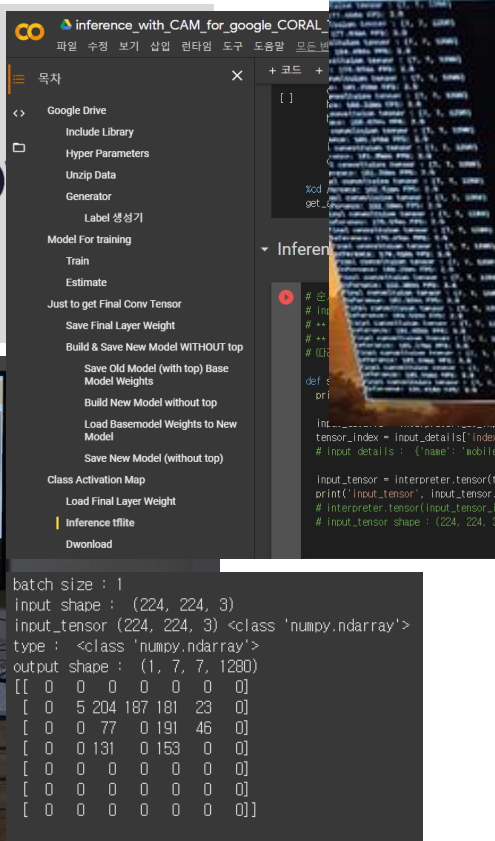
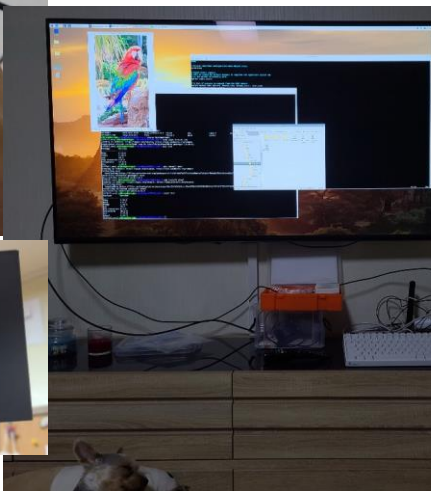
API 문서 읽기 완료.
Custom 모델 압축 후
Transfer Learning
하는 방법 연구 완료

직접 제작한 모델
& 기본 제공
Detection 모델
CORAL 포팅 성공

```
(2, 720)  
(607, 247398587883, 3495.69469800692, 0.1514120723521852)  
(720, 1080, 3)  
(720, 1080, 3)  
Out [292]: <matplotlib.image.AxesImage at 0x2c043fece80>
```



OpenCV + 고전적 CV 이론 기반
Lane Detection + Curve Detection Base Code

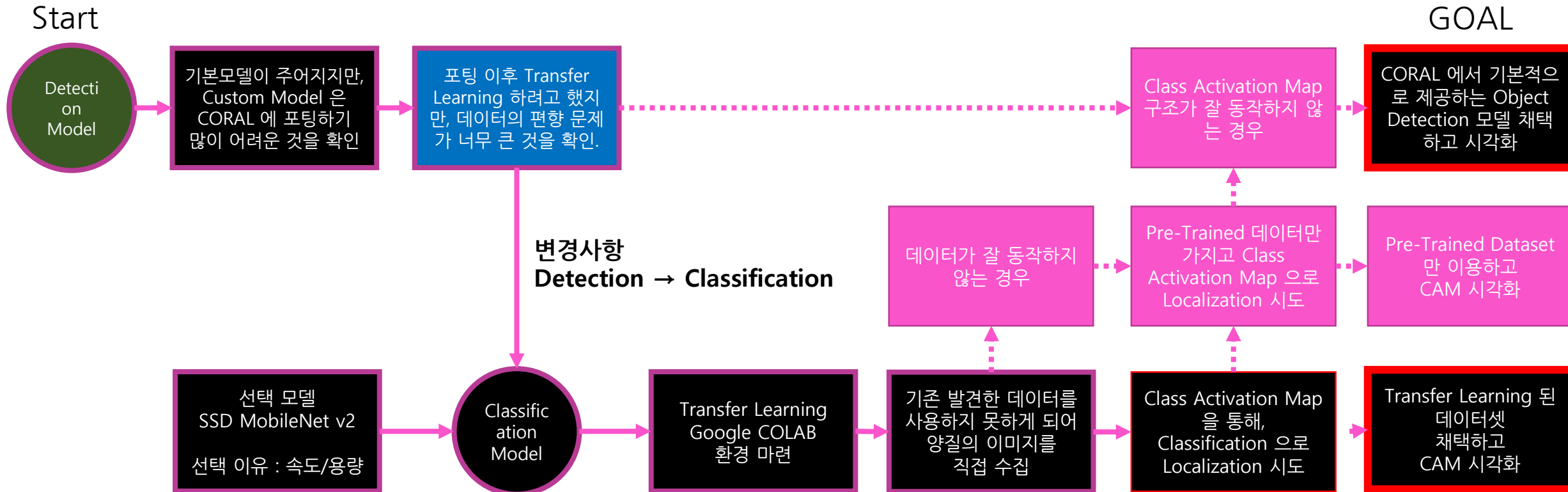


퍼스널 모빌리티 자율주행 보조 기기 제안

진행 상황 / 변경 사항

Custom Network
Class Activation Map
with mobilenet
CORAL 포팅 완료

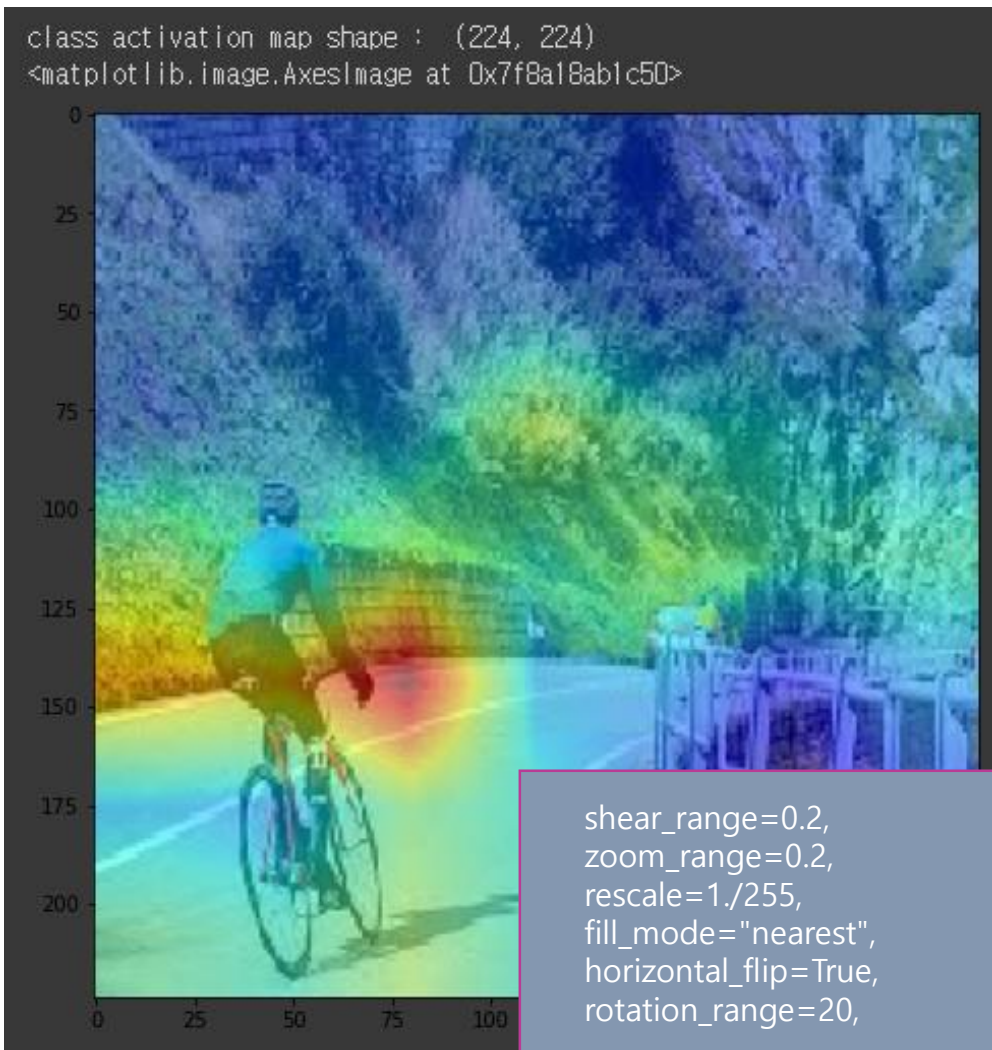
CORAL 에 올리려면, tf 모델과 parameter 을 압축된 tflite 형식으로 바꾸어야 함.
tflite 은 수정 변경이 불가능한 모델이고, Transfer Learning 이 제한적임.
따라서, tensorflow 모델을 COLAB 에서 직접 제작하고 심어 주는 단계를 거쳐야 함.
기본 모델의 구조를 GAP (Global Average Pooling) 구조로 바꿔 포팅하는 데 성공



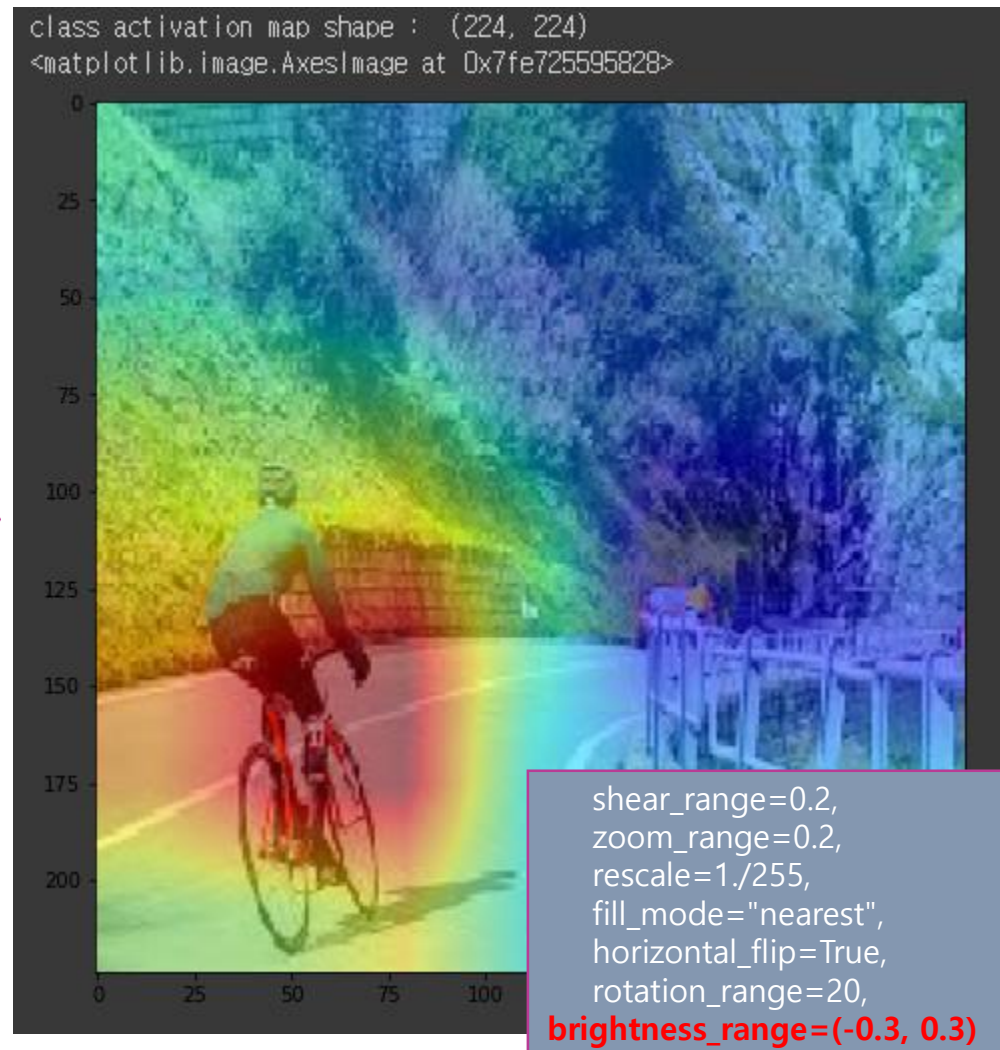
Pretrained : ImageNet 1000 class

데이터 : 2 Class [자전거+킥보드, from google image] 120, [Noobj, from VOC 2005, 2006] 120

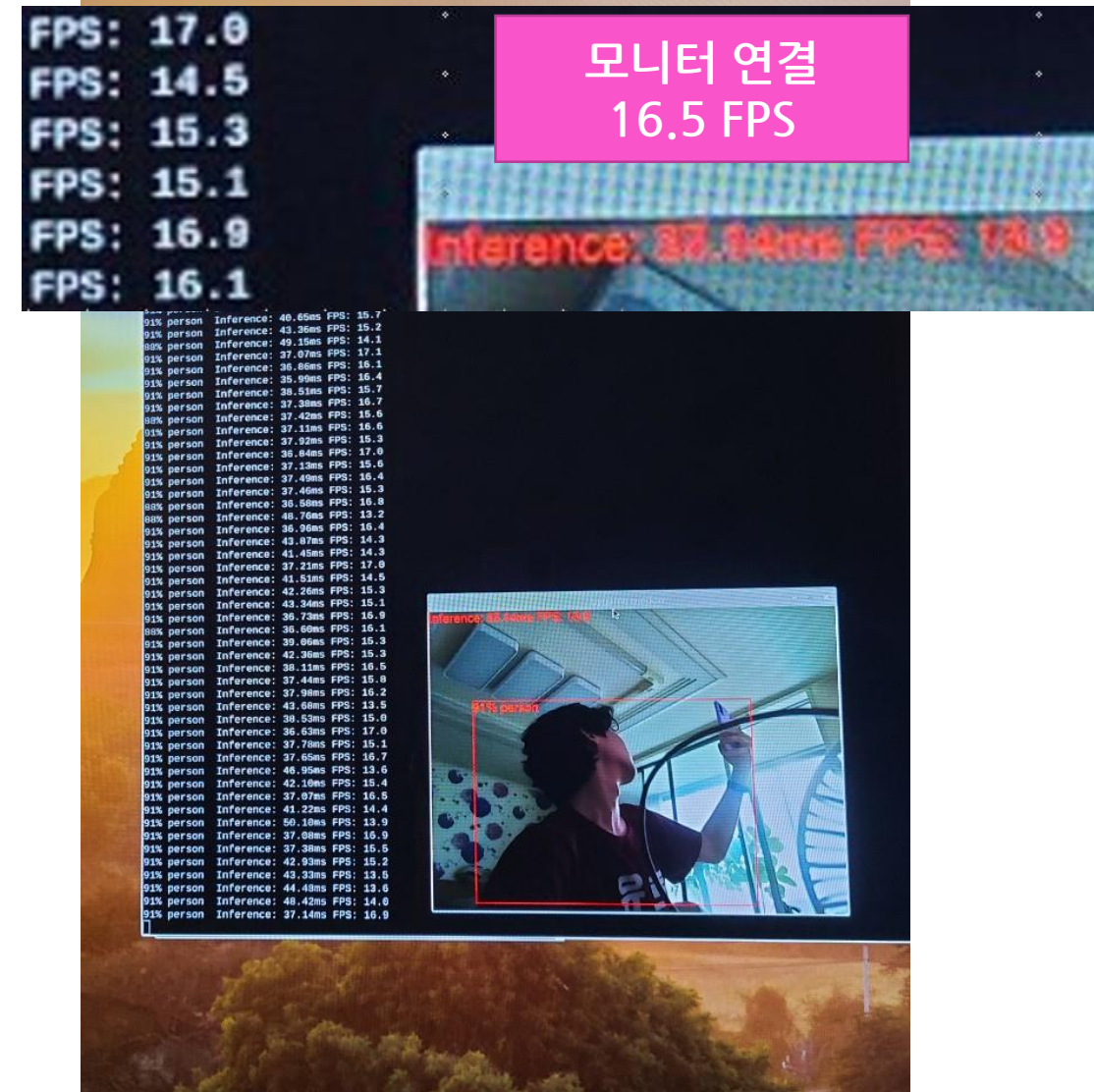
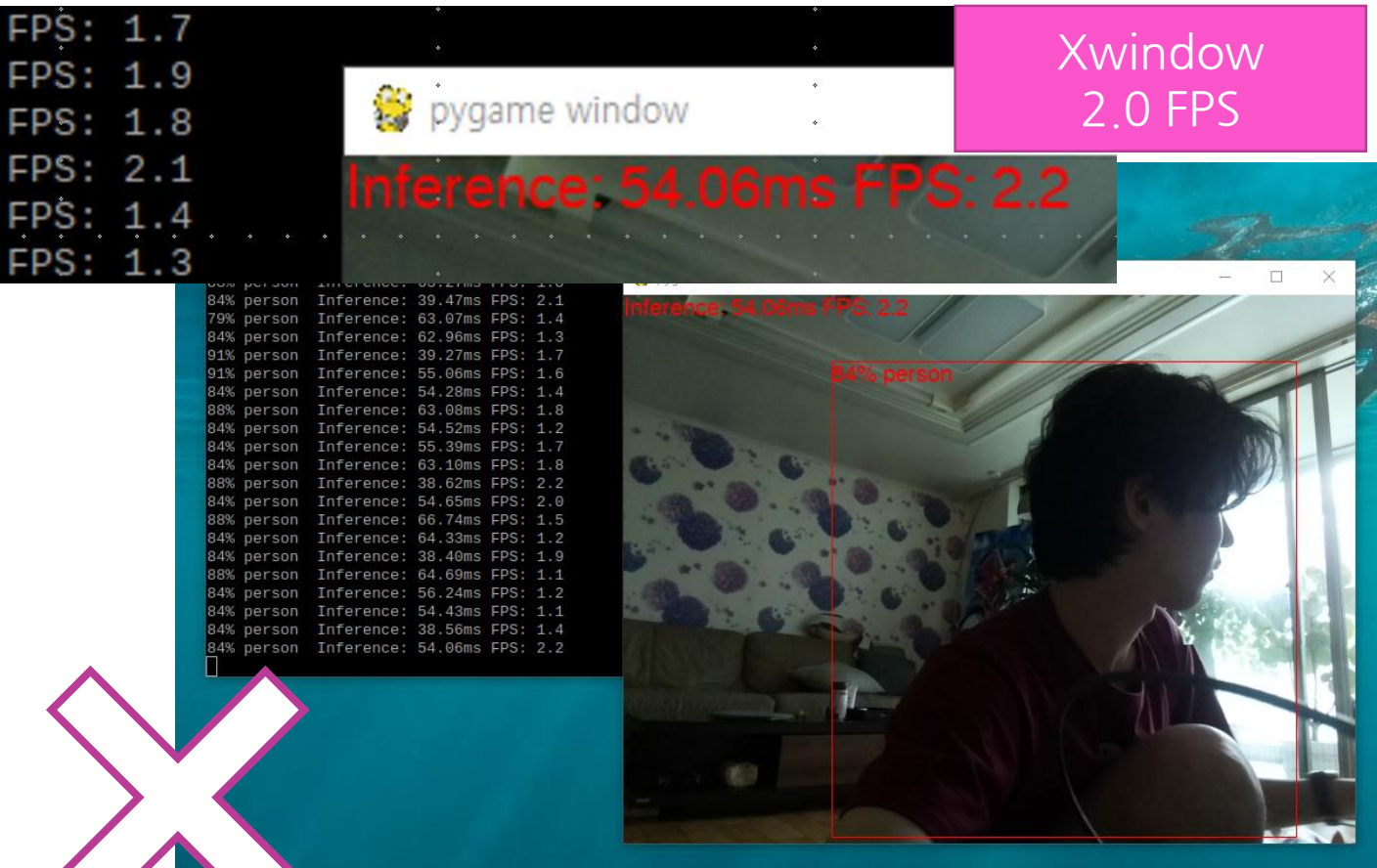
중간 보고 이후 튜닝된 내용



Augmentation
추가



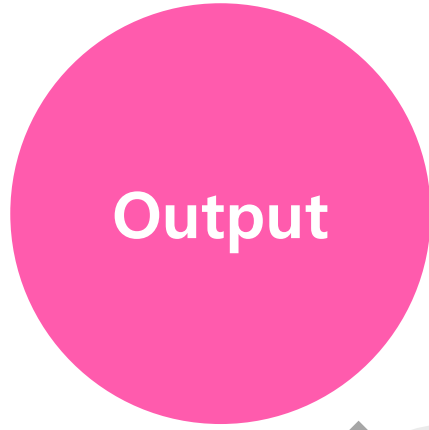
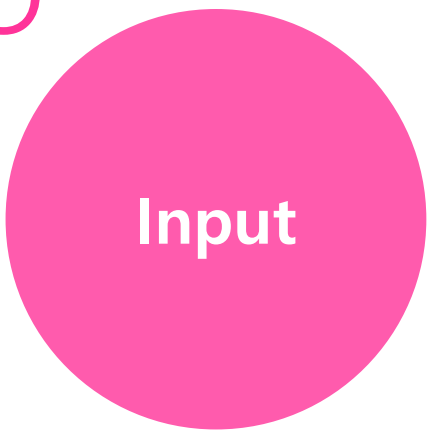
중간 보고 이후 실험한 내용



조교님 조언대로, FPS 를 출력하는 방식으로
함께 시도를 해 보았음. FPS 는 유지되고 모니터 출력만 버벅일 줄 알았는데,
프레임을 처리하는 속도도 함께 나빠지는 모습을 보여줌. (왜지..?)
여튼 자체 모니터가 없는 환경에서 실주행 데모를 보여주는 것은 무리일 것이라 판단

퍼스널 모빌리티 자율주행 보조 기기 제안

Final IO



448x448 (recommend) *No depth camera
*No rider / lidar sensor

RGB Pi
Cam1



Google
Coral
TPU

Power Input

c type power input from notebook
5V, 3A + alpha

Open
CV
python

TF
Light

10~30 FPS

Stabilized
Video
(Hardware)

Save
video

Bicycle
Lane
Detection

(가능하다면)
원본 영상 위에
overlay

display

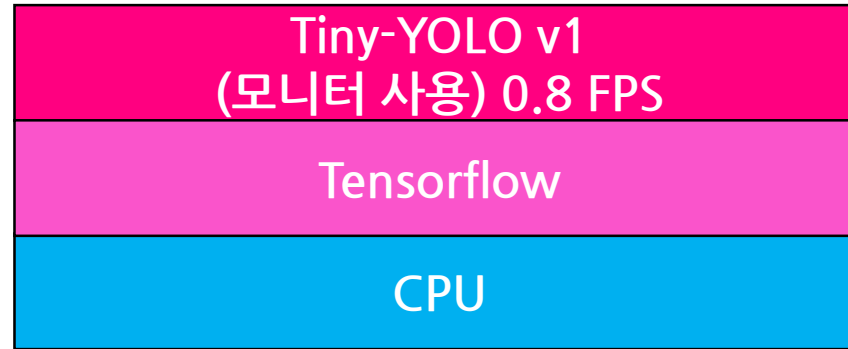
Simple
Detection
With
Pretrained

Class
Activation
Mapping
CustomData

원본 영상 위에
overlay



결과물



Neural Network 를 바탕으로
자전거 탑승자를 **detection**.
(Real Time Detection 은 보여주지 못하지만,
Testset 에서 잘 작동함은 확인)
Google 에서 제공하는 소형 TPU 를 라즈
베리 파이에 포팅하고 (하드웨어적 최적화)
소프트웨어적으로 또한 최적화해 보는 부
분 에 중점을 맞춤.

목표 달성!

*CAM 이 더 느린 이유 : Activation
Map 을 Overlay 하기 위해 전처리하
는 과정을 CPU 가 맡아서 하기 때문에
시간이 오래 걸림.

5배 속도

20 배속도



보여줄 Demo 1 : (직접 만들지 X, 포팅시킨) mobilenet SSD (Detection) - Real time 잘 됩니다.

보여줄 Demo 2 : 직접 튜닝한 mobilenet Class Activation Map - 자전거 탄 사람, 키보드 탄 사람, 태양 조명 느낌의 데이
터로 학습시켰기에, 완전 잘 작동하지는 않으나, 봐야 할 부분을 어느정도 준수하게 파악하긴 합니다.

퍼스널 모빌리티 자율주행 보조 기기 제안

개발 환경 소개

화면



노트북

라즈베리파이

개발 환경 소개

유선 인터넷 / 마우스 / 키보드 연결



Raspberry pi

Pi Camera

구글 edge TPU

스테빌라이저 (짐벌)

연결된 TV

공유기

손선풍기 TPU 쿨러

퍼스널 모빌리티 자율주행 보조 기기 제안

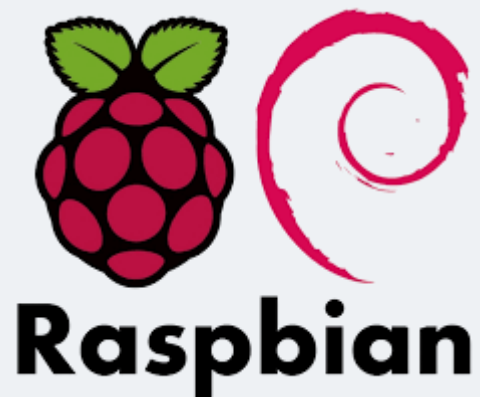
개발 환경 소개



PuTTY

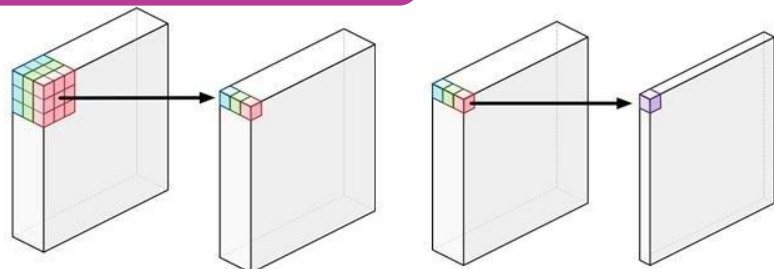


TensorFlow Lite

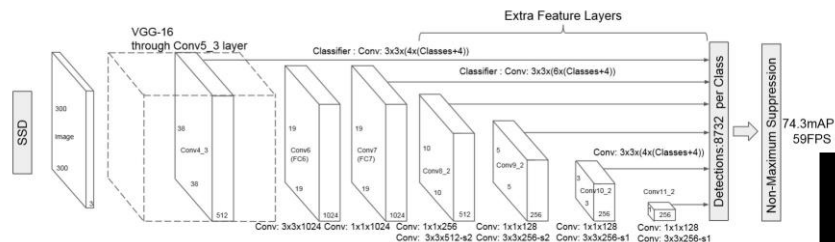


적용된 경량화, 최적화 방법들

선택한 모델의 구조



a) depthwise convolution
b) pointwise convolution
Convolution Layer 에 Mobilenet 구조 (2016 제안)



One Stage proposal 모델을 채택해서 시간 단축 (2016 제안)

선택한 모델 압축

Float32 를 uint8 datatype 으로 압축

어떻게 하면 모델을 최적화할 수 있을지 알아보고, 발견한 모델이 tf-lite + CORAL 연산을 지원하는지 많이 알아보았습니다. 후보 모델 중에, 그나마 언급도 많고 마지막단까지 연산 충돌이 없고 속도가 준수하다고 여겨져 이것을 채택하게 되었습니다. 총 4 단계의 최적화를 거쳤습니다.

컴파일을 통해

자료형을 통해

Mobilenet v2
SSD.h5

Mobilenet v2
SSD.tf-lite

Mobilenet v2
SSD.tf-lite

float32

float32

uint8

Keras

Tensorflow
lite

Tensorflow
lite

Tensorflow

전용 하드웨어를 통해

Google
CORAL

Class Activation Map - CORAL 개발 루트 (1)

```
[ ] try:
    # This %tensorflow_version magic only works in Colab.
    %tensorflow_version 1.x
except Exception:
    pass
# For your non-Colab code, be sure you have tensorflow==1.15
import tensorflow as tf
assert tf.__version__.startswith('1')
```

tf.enable_eager_execution()

```
import os
import numpy as np
import matplotlib.pyplot as plt
```

TensorFlow 1.x selected.

```
[ ] tf.__version__
```

'1.15.2'

```
[ ] %cd /content/gdrive/My Drive/data/example/trialdata
In [ ]: !rm -r custom_cykick
In [ ]: !mkdir custom_cykick
In [ ]: !unzip custom_cykick.zip -d /content/gdrive/My Drive/data/example/trialdata/custom_cykick
In [ ]: !ls

# 직접 제작한 데이터셋으로, google image search 의 미리보기 이미지와, PASCAL VOC 2005 2006 데이터셋에서 총 220 장의 이미지를 준비했습니다.
```

정확히 **1.5 버전만**,
tensorflow light 를 제대로 지원합니다.

COLAB 에서 기본 설정인 2.0 버전도 지원하지 않습니다.

다양한 구조들 중에, Class Activation Map 의 CORAL 포팅을 시도한 이유는 다음과 같습니다.

- 사람이 탑승하는 모빌리티에서 Detection Model 의 설명력, 디버깅 등은 중요하기 때문입니다. CAM 은 XAI 의 한 분야로 여겨집니다. 이 구조를 만들어 두면, 나중에 다양하게 활용 가능할 것 같았기 때문
- Object Detection 모델을 훈련시키기 어려울 때, Classification 데이터로 (weakly-labeled data) 로 Localization 을 수행할 수 있기 때문.

Class Activation Map - CORAL 개발 루트 (2)

1. 모델을 만든다.
 1. 이때, 모델의 내부를 모두 확인해서, tfliite 뿐만 아니라 google coral 호환여부도 확인해야 한다.
 2. 우리는 Detection 문제를 풀 것이 아니므로, F.C Layer (Top layer) 은 떼어내고 가져온다.
2. 가져온 모델의 앞단의 가중치를 얼리고, F.C Layer 을 한 단만 붙여 fully connected layer 만 학습시킨다.
 1. 한 단만 붙이는 이유는, Class Activation Map 을 만드는 과정이 원래 그렇기 때문.
 2. 구체적인 학습 디테일은 코드를 참고. 특히, Augmentation.
3. 학습이 끝나면, 학습된 F.C Layer 의 가중치만 저장한다. 이때, running 할 때도 열어야 하므로 피클파일로 저장.

```
[ ] model.compile(optimizer=TRAINING_OPTIMIZER,
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Model)	(None, 7, 7, 1280)	2257984
gap (GlobalAveragePooling2D)	(None, 1280)	0
final_dense (Dense)	(None, 2)	2562
Total params: 2,260,546		
Trainable params: 2,562		
Non-trainable params: 2,257,984		

[illegible]

- Save Final Layer Weight

```
[ ] import pickle

%cd /content/gdrive/"My Drive"/data/example

with open('dense_layer_param.pkl', 'wb') as f:
    pickle.dump(dense_layer, f)
```

```
↳ /content/gdrive/My Drive/data/example
```



Class Activation Map - CORAL 개발 루트 (3)

4. Session 을 정리하고, F.C Layer 이전의 layer 까지만 가진 모델을 다시 만든다.
5. 다시 만들어진 모델의 압축 방법을 지정하고 (uint8 : CORAL에서는 unit8만 지원) Tflite 모델로 (.tflite) 변환한다.
6. 압축된 모델 평가 (생략)
7. 해당 .tflite 모델과 F.C Layer 의 가중치를 CORAL 이 부착된 라즈베리파이에 내린다.

Save Old Model (with top) Base Model Weights

```
[ ] weights_list = model.get_weights()
    print(len(weights_list))
    model.save_weights('trained_basemodel_weight.h5')
```

262

Build New Model without top

```
[ ] new_base_model = tf.keras.applications.MobileNetV2(input_shape=IMG_SHAPE,
                                                         include_top=False,
                                                         )
    new_base_model = tf.keras.Sequential([new_base_model])
    new_base_model.compile(optimizer=tf.keras.optimizers.Adam(),
                           loss='categorical_crossentropy',
                           metrics=['accuracy'])
    new_base_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #

mobilenetv2_1.00_224 (Model) (None, 7, 7, 1280)		2257984

Total params: 2,257,984		
Trainable params: 2,223,872		
Non-trainable params: 34,112		

```
converter = tf.lite.TFLiteConverter.from_keras_model_file(SAVED_KERAS_MODEL_NAME)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
# This ensures that if any ops can't be quantized, the converter throws an error
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
# These set the input and output tensors to uint8
converter.inference_input_type = tf.uint8
converter.inference_output_type = tf.uint8

...

A representative dataset that can be used to evaluate optimizations by the converter.
E.g. converter can use these examples to estimate (min, max) ranges by calibrating.
''' # And this sets the representative dataset so we can quantize the activations
converter.representative_dataset = representative_data_gen

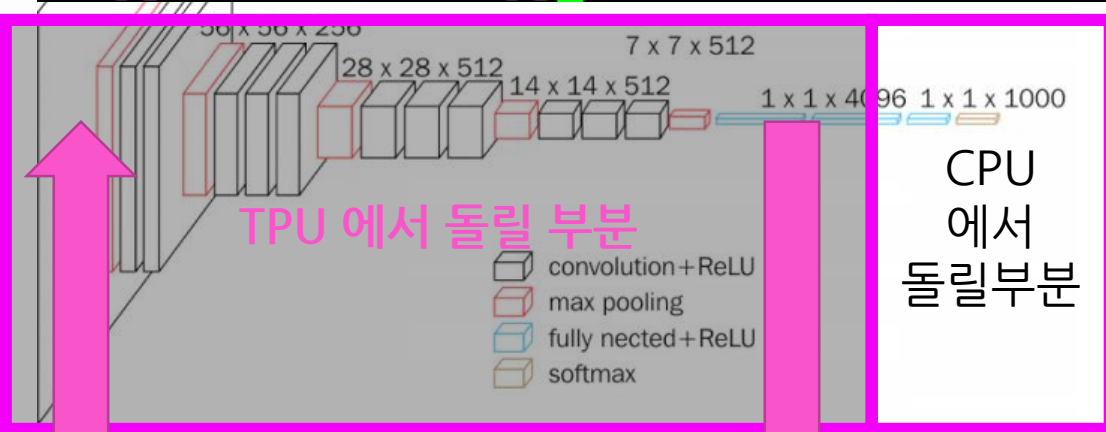
# Convert
tflite_model = converter.convert()
with open('{}'.format(TFLITE_MODEL_NAME), 'wb') as f:
    f.write(tflite_model)
```

convolution+ReLU
max pooling
fully nected+ReLU
softmax

Class Activation Map - CORAL 개발 루트 (4)

8. 라즈베리파이에는, CORAL 드라이버를 설치하고, 가상환경을 만들고 tensorflow lite 전용 interpreter 을 설치한다.
9. Opencv 를 설치하고, 가상환경에 링크해 준다.
10. Interpreter 의 타겟을 CORAL 로 두고, 직접 interpreter 에서 index 로 접근해서 input tensor 에 값을 써 주고, Feed forward 명령하고, TPU 에서 연산 결과를 output tensor 에 넣어주면, 해당 index 접근해서 다시 CPU 에서 처리해 주는 구조.

```
(tfliteinter_env) pi@raspberrypi:~/Desktop/tfliteinter_env/self-driving-PM/LAB/camera/pygame
$ ls
classify_capture.py  common.py  detect.py  install_requirements.sh  __pycache__  README.md
(tfliteinter_env) pi@raspberrypi:~/Desktop/tfliteinter_env/self-driving-PM/LAB/camera/pygame
$ python3 detect.py
```



*참고 : TPU 에 tflite 모델을 올릴 때, CORAL에서 지원하지 않는 연산 또는 형식이 있으면 그 부분부터는 온전히 CPU 의 몫이 됨.

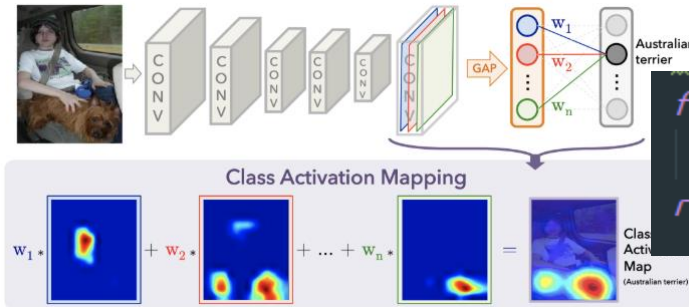
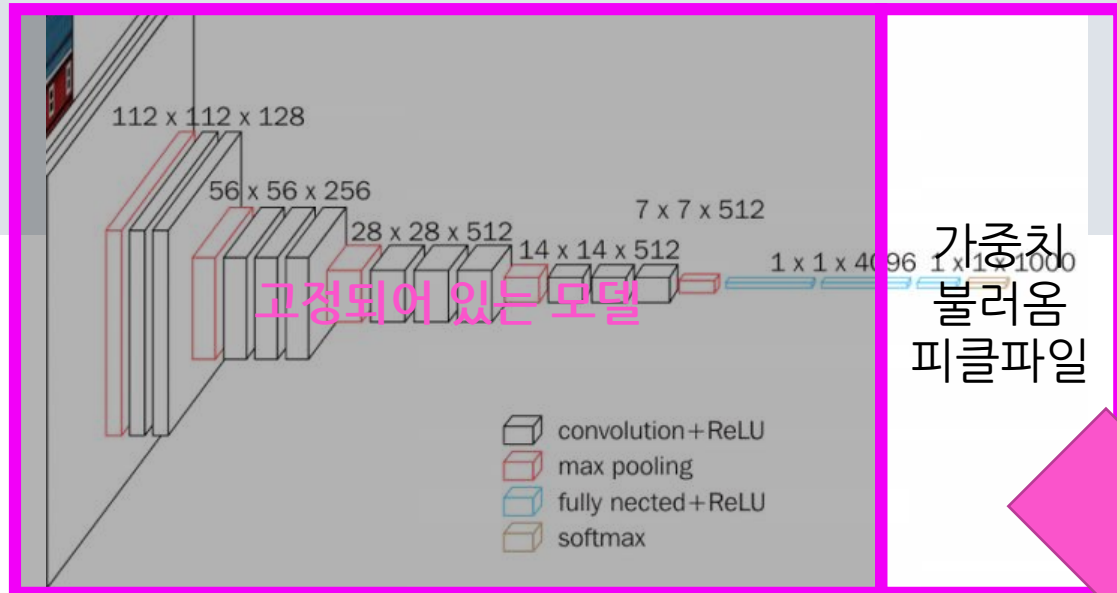
```
def input_image_size(interpreter):
    """Returns input image size as (width, height, channels) tuple."""
    _, height, width, channels = interpreter.get_input_details()[0]['shape']
    return width, height, channels

def input_tensor(interpreter):
    """Returns input tensor view as numpy array of shape (height, width, 3)."""
    tensor_index = interpreter.get_input_details()[0]['index']
    return interpreter.tensor(tensor_index())[0]

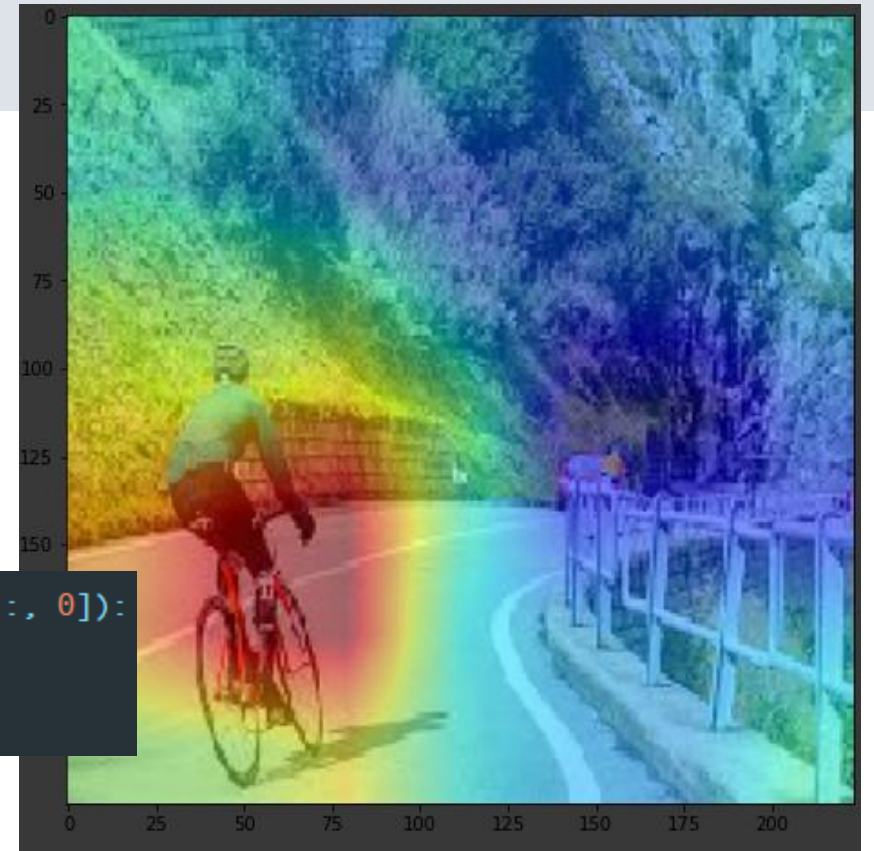
def output_tensor(interpreter, i, dense_layer_weight, scaling = False):
    """Returns dequantized output tensor if quantized before."""
    output_details = interpreter.get_output_details()[i]
    output_activation = np.squeeze(interpreter.tensor(output_details['index']()))
```

Class Activation Map - CORAL 개발 루트 (5)

그렇게 tensor 의 output 결과값을 받아온 이후,
마지막으로 아까 저장해 두었던 pkl 파일을 가져와서, tensor 의 사이즈에 맞게 적절히 연산을 해 준다.



```
for index, weight in enumerate(dense_layer_weight[:, 0]):  
    CAM += weight * output_activation[:, :, index]  
return CAM
```



어려웠던 점들

- tf-lite 한글자료가 적음, CORAL 은 한글자료 없음
- Google CORAL 나온지 2년밖에 안되어 자료가 많이 없어 공식 예제만 참고.
- 정말 설치 할 것들이 너무 많은 부분
- 전역과 가상환경 영역 제어하기.
- 처음 보는 모델에서 모델 수정하기.
- 툭 하면 꺼지고 끊기고 터지는 환경들.
- SSH 특히, hotspot 연결할 때.
- 하드웨어의 제약 극복하기.
- 잠 안 자기.

소스 코드 및 개발 이력 (1)



참고한 내용

<https://github.com/ProtossDragoon/self-driving-PM/blob/master/STUDY.md>



개발 과정

<https://github.com/ProtossDragoon/self-driving-PM/blob/master/ROADMAP.md>



Class Activation Map Colab 코드

https://colab.research.google.com/drive/1c5z8XcxIxXQFjRI2X7dHO_DRzJS EytQO?usp=sharing

소스 코드 및 개발 이력 (2)

- 망할 배터리, LCD 아직도 안옴

붉은색 셀

삭제된 부분

노란색 셀

변경된 부분

검정색 셀

예전에 수정

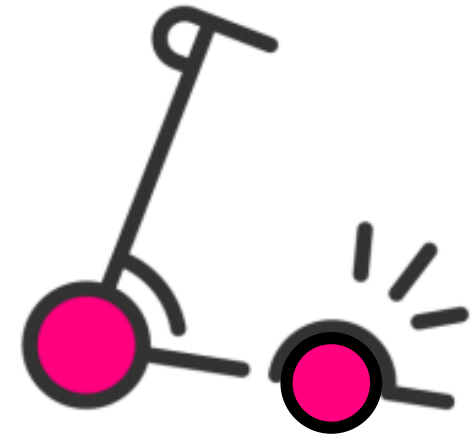
회색 셀

정상 진행

	1주		2주		3주		
	월화	수목금	월화	수목금	월화	수목금	토일
교수님 accept	o						
듀얼카메라 모듈 구매		주문					
Google Coral TPU 주문 배송 / 주행 실험용 보조배터리 주문 배송			주문	CORAL	아직도안옴		
OpenCV / Python 환경 마련		VM					
단일카메라 Lane Detection 패키지/소스코드 준비		o					
단일카메라 Lane Detection 테스트							
단일카메라 곡선 Lane Detection 소스코드 준비			o				
단일카메라 Lane Detection 최적화							
단일카메라 영상 저장 패키지/소스코드 준비/테스트							
듀얼카메라 Real time Depth 패키지/소스코드 준비							
raspberry pi 에 올리기 적절한 lite 모델 후보 2개 선정 (Detection, Localization)	SSD Mobilenet v2 / Mobilenet CAM						
Google Coral TPU 포팅					o	o	
Tensorflow Lite 및 개발환경 포팅				o	o	o	
Pre-Trained Model 테스트					진행중		
듀얼카메라 모듈 배송 완료							
캘리브레이션 아크릴판 제작					필요없음		
듀얼 카메라 캘리브레이션							
캘리브레이션 코드 결합					필요없음	필요없음	
2개 model 의 Cyclist Data Transfer Learning					o		
Test Dataset 을 이용해서 성능 평가					o	o	
Activation Map (시간 남으면 Lane Detection 도) 성능 테스트, 화면에 띄우기					activation 만		
듀얼카메라 Real time Depth 소스코드 병합							
Real time Depth 계산 영역 Lane Detection 결과 이용 최적화							
Real time Depth로부터 Roi Box 그리기							
테스트 영상 제작						o	
발표자료 제작						o	
제출							

퍼스널 모빌리티 자율주행 보조 기기 제안

기존 내용들 (중간보고2, 6/17)



18011573 컴퓨터공학과 이장후
dlwkdgn1@naver.com

진행 상황 / 변경 사항

지난 주 ;
Curve Detection
Base Code 개발

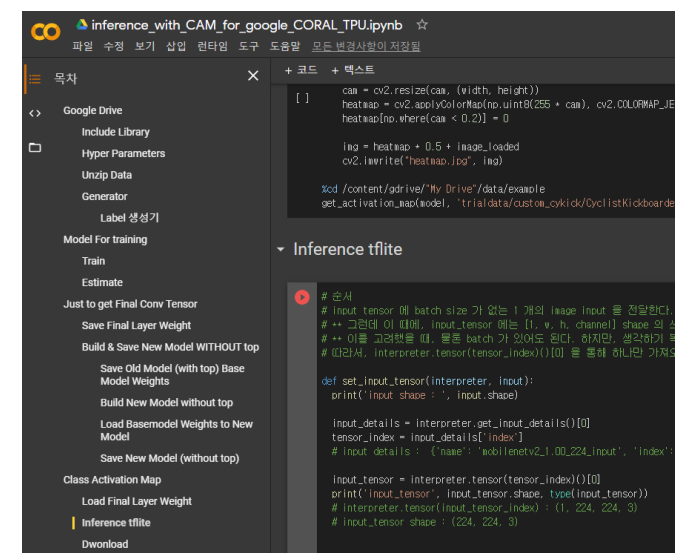
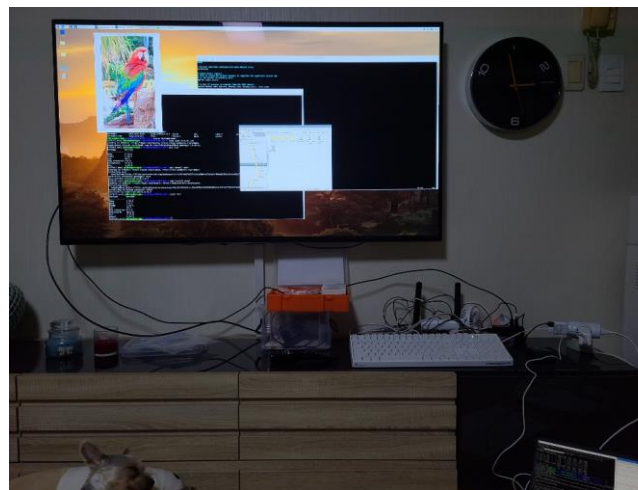
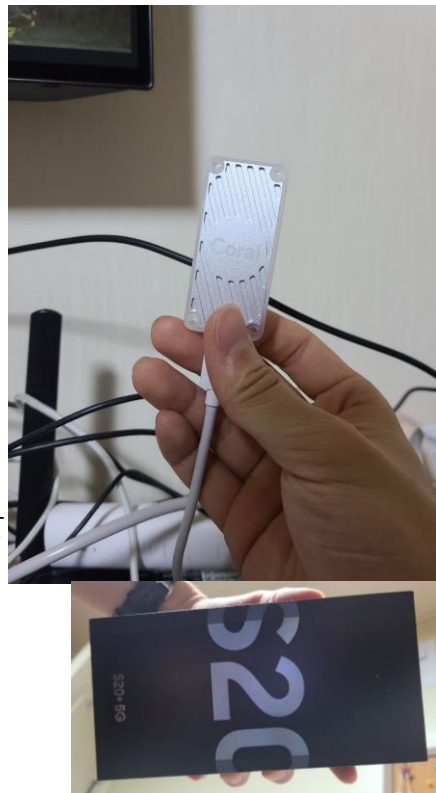
Rpi4 포팅 완료.
Google Coral 도착
Coral 포팅 완료

RPI 포팅 완료,
tf.Lite / Google Coral
이용한 Classification
예제 동작 실험 완료

API 문서 읽기 완료.
Custom 모델 압축 후
Transfer Learning
하는 방법 연구 중



OpenCV + 고전적 CV 이론 기반
Lane Detection + Curve Detection Base Code 개발



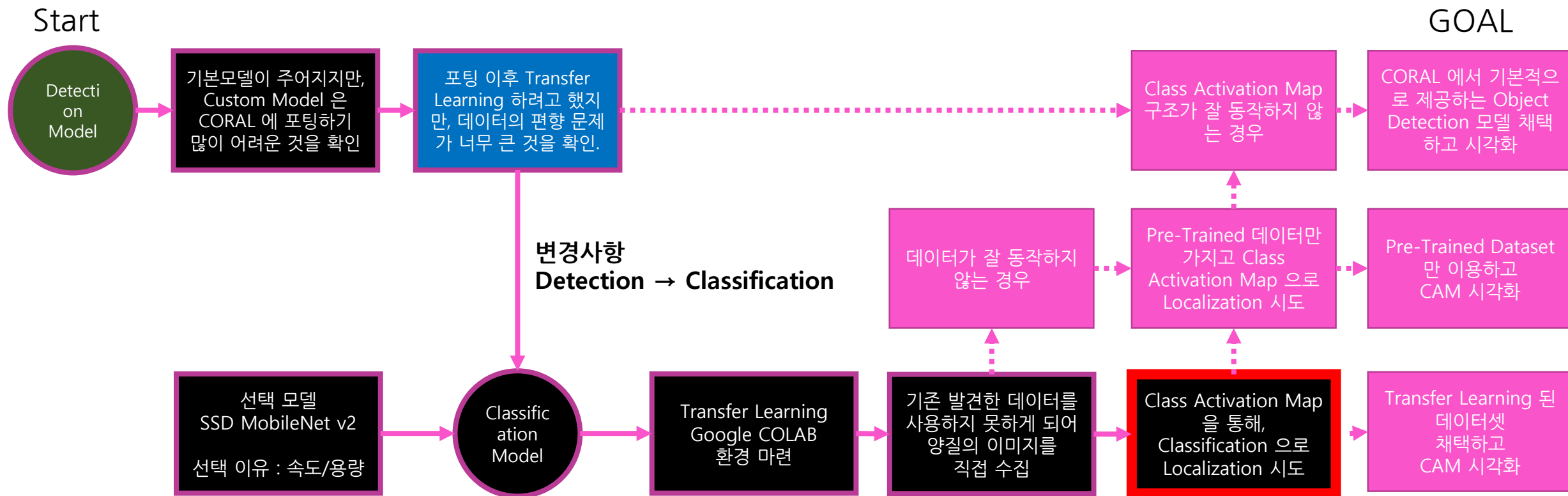
```
batch size : 1
input shape : (224, 224, 3)
input_tensor (224, 224, 3) <class 'numpy.ndarray'>
type : <class 'numpy.ndarray'>
output shape : (1, 7, 1280)
[[ 0 0 0 0 0 0 0]
 [ 0 5 204 187 181 23 0]
 [ 0 0 77 0 191 45 0]
 [ 0 0 131 0 153 0 0]
 [ 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0]
 [ 0 0 0 0 0 0 0]]
```

퍼스널 모빌리티 자율주행 보조 기기 제안

진행 상황 / 변경 사항

API 문서 읽기 완료.
Custom 모델 압축 후
Transfer Learning
하는 방법 연구 중

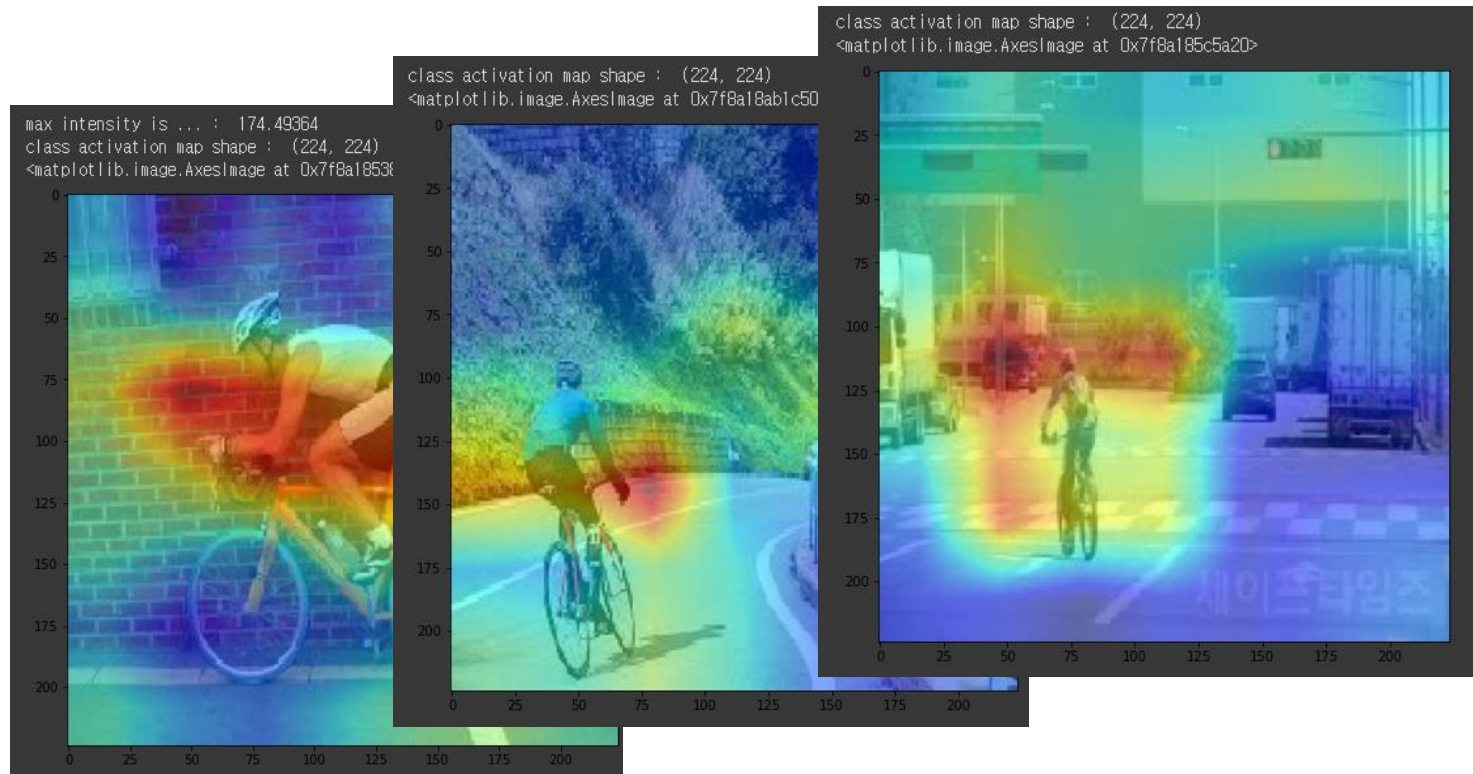
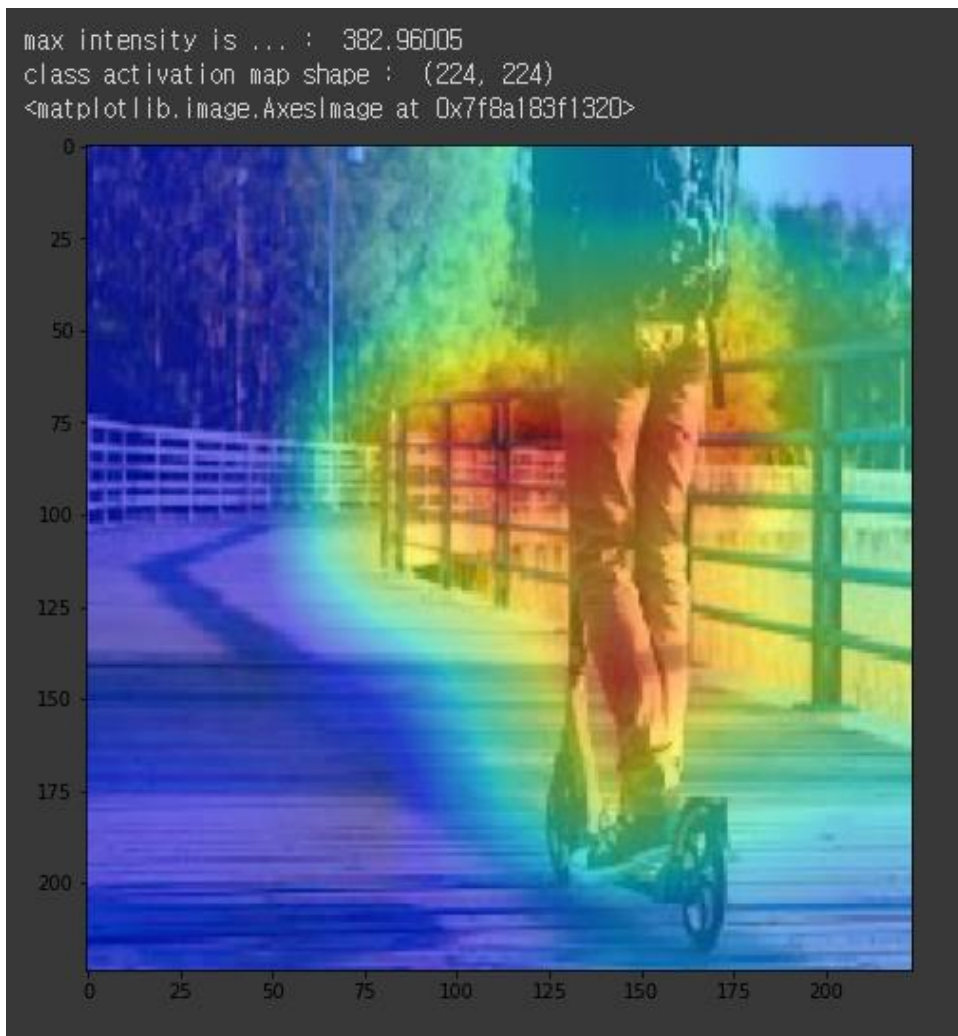
CORAL 에 올리려면, tf 모델과 parameter 을 압축된 tflite 형식으로 바꾸어야 함.
tflite 은 수정 변경이 불가능한 모델이고, Transfer Learning 이 제한적임.
따라서, tensorflow 모델을 COLAB 에서 직접 제작하고 심어 주는 단계를 거쳐야 함.
기본 모델의 구조를 GAP (Global Average Pooling) 구조로 바꿔 잘 동작하는지 확인 중.



Pretrained : ImageNet 1000 class

데이터 : 2 Class [자전거+킥보드, from google image] 120, [Noobj, from VOC 2005, 2006] 120

진행 상황 / 변경 사항



CAM : Class Activation Map, Learning Deep Features for Discriminative Localization (CVPR 2016, Bolei Zhou et al.) : Object Detection (Localization) 문제를 Classification 으로도 풀어갈 수 있음을 제대로 보여 준 연구

Detection 문제로 풀어가지 않고, Classification 의 응용인 CAM 을 선택한 이유는 아래와 같음.

좋은 품질의 detection 라벨링 데이터가 없음 / Detection 문제를 풀려면 엄청난 양의 데이터가 필요함 / Detection 문제보다 더욱 '왜 Cyclist 라고 판별했는가' 를 보여줄 수 있고, 데이터셋 개선 가능성이 있음.

하지만, 자체적 단점은, 저러한 Activation Map 자체 성능을 평가하기 어려워 사람이 직접 확인해야 함. 어쩔 수 없이 직접 만든 데이터셋 120장 중 30장을 쪼개 한 장 한 장 합리성을 따져보는 수밖에 없었음.

데이터

2 class

- Class1 Cykick : 자전거탄사람, 킥보드탄사람 Google 이미지 110 장
 - 양질의 데이터를 만들기 위해 크롤러 도구를 사용하지 않고 직접저장, CCL (저작권) 필터링을 함.
- Class2 Noobject : 무의미한 데이터 110장
 - PASCAL VOC 2005, 2006 데이터에서 dog, cat, cow, person 등 무의미한 데이터를 가져옴.

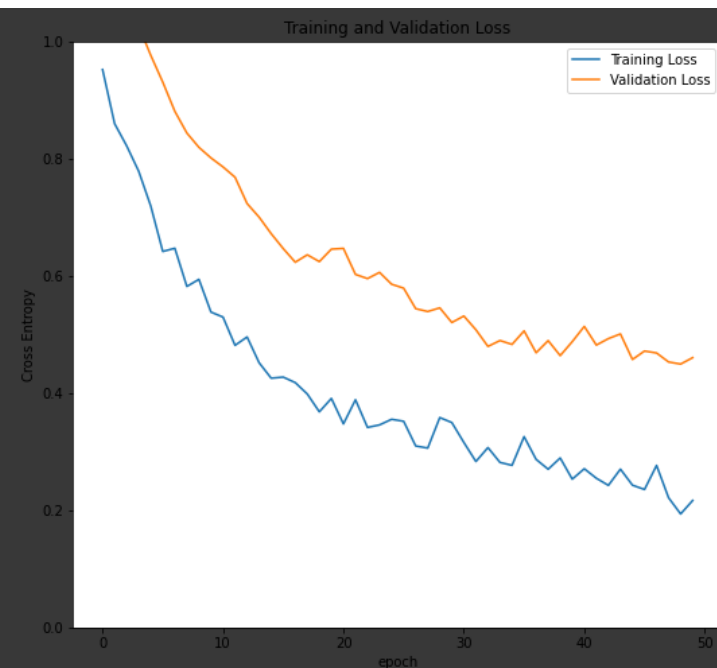
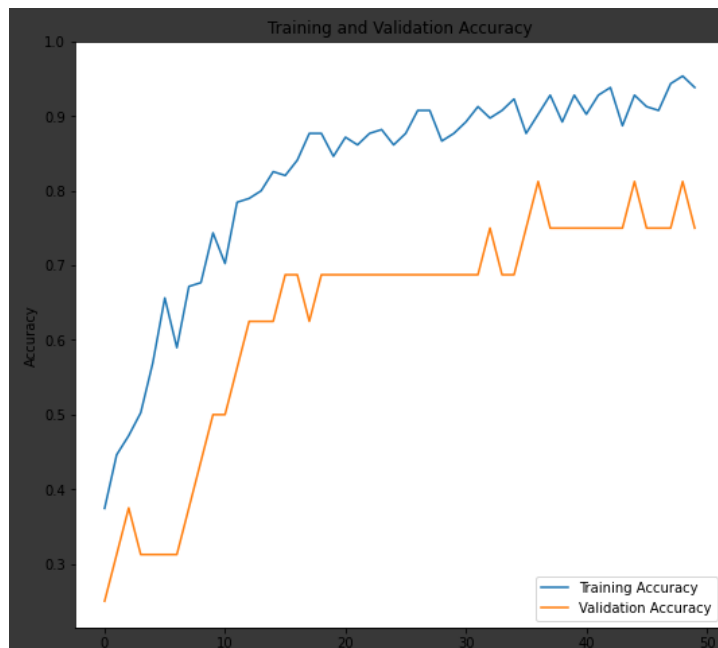
Optimizer : SGD (안정적 수렴 위해)

Learning rate : 0.001

Epoch : 50

Batch size : 16

shear_range=0.2,
zoom_range=0.2,
rescale=1./255,
fill_mode="nearest",
horizontal_flip=True,
rotation_range=20,



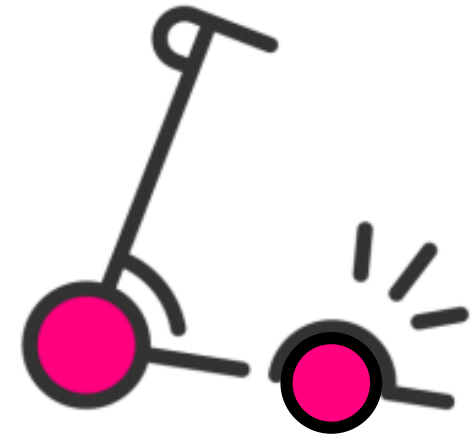
CORAL 개발 구조

- 최종 발표때 간단히 소개하겠습니다.

진행 일정

	1주		2주		3주		
	월화	수목금	월화	수목금	월화	수목금	토일
교수님 accept	o						
듀얼카메라 모듈 구매		주문					
Google Coral TPU 주문 배송 / 주행 실험용 보조배터리 주문 배송			주문	CORAL			
OpenCV / Python 환경 마련		VM					
단일카메라 Lane Detection 패키지/소스코드 준비		o					
단일카메라 Lane Detection 테스트							
단일카메라 곡선 Lane Detection 소스코드 준비			o				
단일카메라 Lane Detection 최적화							
단일카메라 영상 저장 패키지/소스코드 준비/테스트							
듀얼카메라 Real time Depth 패키지/소스코드 준비							
raspberry pi 에 올리기 적절한 lite 모델 후보 2개 선정 (성능위주, 속도위주)			yolo v4 / SSD Mobilenet v2 / CAM				
Google Coral TPU 포팅					o		
Tensorflow Lite 및 개발환경 포팅				o	o		
Pre-Trained Model 테스트					진행중		
듀얼카메라 모듈 배송 완료							
캘리브레이션 아크릴판 제작							
듀얼 카메라 캘리브레이션							
캘리브레이션 코드 결합							
2개 model 의 Cyclist Data Transfer Learning							
Test Dataset 을 이용해서 성능 평가							
캘된 단일 카메라 Object Detection (시간 남으면 Lane Detection 도) 성능 테스트							
듀얼카메라 Real time Depth 소스코드 병합							
Real time Depth 계산 영역 Lane Detection 결과 이용 최적화							
Real time Depth 로부터 Roi Box 그리기							
테스트 영상 제작							
발표자료 제작							
제출							

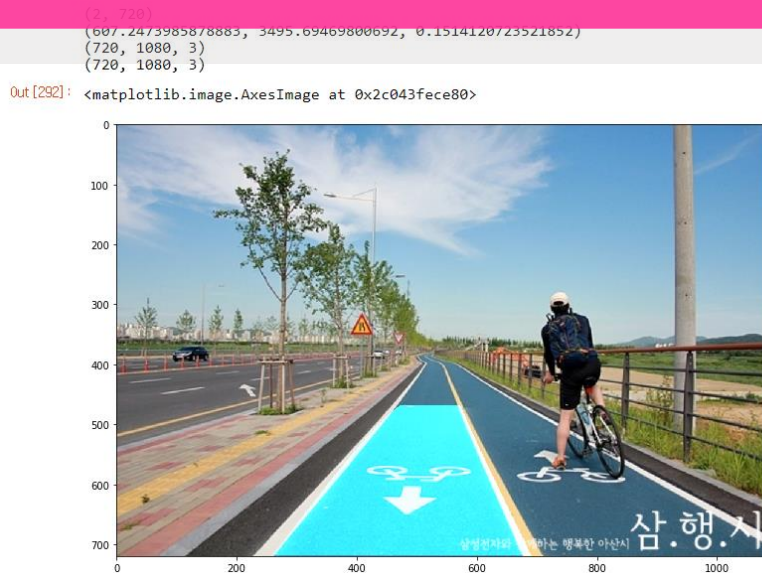
기존 내용들 (중간보고1, 6/10)



18011573 컴퓨터공학과 이장후
dlwkdgn1@naver.com

진행 상황 / 변경 사항

현재까지의 진행 상황



OpenCV + 고전적 CV 이론 기반
Lane Detection + Curve Detection Base Code 개발 완료

알고리즘 최적화 고민 중에, 듀얼 카메라 HW 배송 3주 지연 소식을 들었습니다.



어서오세요. '가치창조기술' 입니다.
궁금한 내용이 있으시면 메시지를 남겨주세요

가치창조기술 소식받기

배송지연 고객님, 가치창조기술 쇼핑몰에서 주문해주셔서 감사합니다. 죄송하게도 주문하신 상품의 배송이 지연되어 알려드립니다. 배송이 순조로워지면 조금만 기다려주세요.
듀얼 카메라의 연동을 도와주는 Raspberry Pi hat 부품 배송이 3주 지연됨. 따라서 프로젝트를 변경할 예정

아두캠 스테레오 싱크 카메라 HAT -라즈베리용 (Arducam Syn



주문상세보기 >

결제일	2020. 6. 3.
판매자지연사유	상품준비중
발송예정일	2020. 7. 3.까지
판매자 상담	070-7770-6256

주문하신 제품이 제조사 재고가 부족하여 입고가 지연되고 있습니다. 대략 3-4주가량 소요될 예정입니다. 확인부탁드립니다. 감사합니다.

CORAL TPU 또한
배송이 지연되는 경우

CORAL TPU 없이
Detection Model 의
최적화 시도를 목표로 하겠습니다.

Neural Network 를 바탕으로
자전거 탑승자를 detection.
Google 에서 제공하는 소형 TPU 를 라즈베리 파이에 포
팅하고 (하드웨어적 최적화)
소프트웨어적으로 최적화해 보는 것에 중점을 맞춤.

퍼스널 모빌리티 자율주행 보조 기기 제안

I / O



1080x720x3 *No depth camera
*No rider / lidar sensor

RGB Pi
Cam1

Google
Coral
TPU

Power Input

c type power input from notebook
5V, 3A + alpha

Open
CV
python

TF
Light

10~30 FPS

Stabilized
Video

Save
video

Bicycle
Lane
Detection

(가능하다면)
원본 영상 위에
overlay

display

Cyclist
Detection

원본 영상 위에
overlay

400x240x3

개발할 내용 / 가져올 내용

가져옴/포팅

가져와서 튜닝

완전히 제작

Bicycle
Lane
Detection

Camera Calibration
Algorithm

Camera Calibration
Board

openCV

OpenCV 포팅

Lane Detection
Algorithm

연결

Cyclist
Detection

Training Data
Pre Trained Model

Tensorflow lite

Google coral TPU

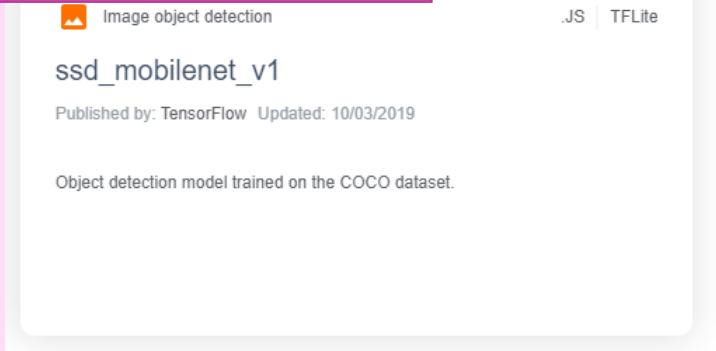
Machine Learning
Model

연결

데이터 / 데모

학습 시

Pre-trained-model



- Coco Dataset
- Tensorflow 제공
- 또는, mobilenet 용이 아닌 weight 을 이용해서 lite version 용으로 변환

Transfer-learning

Download all of the following files and install them into the same directory:

Dataset part	Size	MD5 hash sum
README_TDCR.pdf		
labelData/train	25 MB	7e17bca380056fd75e6586ef0ed598a
labelData/test	10 MB	c63c6d0485fb7a0c8e7bd470f88cd66
labelData/valid	3 MB	2adb1fc2f2b289108f1dfbae27fa2c4
labelData/NonVRU	2 MB	d5895c27299721b69d63cd3face88d0
camera/train	10 MB	1925d6b9897b2865656cd00ba830e4c4
camera/test	3 MB	1434097ea7daeb22e64c15a54cd102c6
camera/valid	1 MB	d0ae2e6d31b67880c0529ce97720e396
camera/NonVRU	1 MB	c667797cc6bb4878e621143b474769b9
leftImg8bit/train	22 GB	03a0c18fd281ea770244d094936eeffe
leftImg8bit/test	7 GB	82fc17983ab452b0b1a5c874d04e89
leftImg8bit/valid	2 GB	08c9fc63173b47a0716e519e3b5a55
leftImg8bit/NonVRU	2 GB	64d76ee7c65f5ae78bb0b790e1984ec
disparity/train	8 GB	2f4b2cd11d506ab6751a6b112d2a7405
disparity/test	2 GB	c72375453d3bce522e0f78efab9a6
disparity/valid	836 MB	6366b94297a0ef62e1e1ab177e181f
disparity/NonVRU	793 MB	e063cfd2670ca522d172b24e993c1e61

- 자전거 탑승자
- 휠체어
- 이론차

등 모빌리티 탑승자
레이블링 데이터셋

테스트 시

학습 데이터 split



- 단일 클래스이므로, AP로 성능 테스트
- FPS 측정을 통한 속도 테스트

실제 주행 테스트
(실제 자전거도로)

데모 시연 영상



변경 일정

붉은색 셀

삭제된 부분

노란색 셀

변경된 부분

기존 Lane Detection 은
시간이 남는 경우 추가적으로 진행하고,

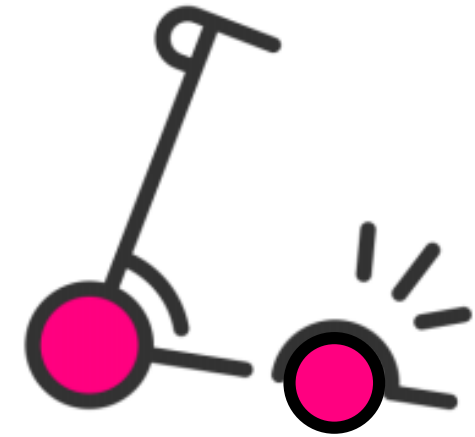
프로젝트 기간 동안은,
Google Coral 이 도착하는 대로
Cyclist Detection 에 초점을 맞추어

라즈베리파이 보드에
모델을 올리는 것을 목표로 준비하겠습니다.

	1주		2주		3주		
	월화	수목금	월화	수목금	월화	수목금	토일
교수님 accept	o						
듀얼카메라 모듈 구매		주문					
Google Coral TPU 주문 배송 / 주행 실험용 보조배터리 주문 배송			주문				
OpenCV / Python 환경 마련		VM					
단일카메라 Lane Detection 패키지/소스코드 준비		o					
단일카메라 Lane Detection 테스트							
단일카메라 곡선 Lane Detection 소스코드 준비			o				
단일카메라 Lane Detection 최적화							
단일카메라 영상 저장 패키지/소스코드 준비/테스트							
듀얼카메라 Real time Depth 패키지/소스코드 준비							
raspberry pi 에 올리기 적절한 lite 모델 후보 2개 선정 (성능위주, 속도위주)							
Google Coral TPU 포팅							
Tensorflow Lite 및 개발환경 포팅							
Pre-Trained Model 테스트							
듀얼카메라 모듈 배송 완료							
캘리브레이션 아크릴판 제작							
듀얼 카메라 캘리브레이션							
캘리브레이션 코드 결합							
2개 model 의 Cyclist Data Transfer Learning							
Test Dataset 을 이용해서 성능 평가							
캘된 단일 카메라 Object Detection (시간 남으면 Lane Detection 도) 성능 테스트							
듀얼카메라 Real time Depth 소스코드 병합							
Real time Depth 계산 영역 Lane Detection 결과 이용 최적화							
Real time Depth 로부터 Roi Box 그리기							
테스트 영상 제작							
발표자료 제작							
제출							

퍼스널 모빌리티 자율주행 보조 기기 제안

기존 내용들 (제안서, 6/3)



18011573 컴퓨터공학과 이장후
dlwkdgn1@naver.com

아이디어들

개 똥 인지 임베디드 기기

원래 Neural Net 바탕으로 강아지 똥오줌을 인식, 주인에게 스마트폰 알림이나 소리 형태로 alert, 반려견 주인이 타이밍 안 놓치고 배변훈련 잘 학습할 수 있도록 돕는 하드웨어-소프트웨어를 만들려고 했으나, 그렇게 되면 너무 무거워지고, 최근 공지해 주신 대로 “임베디드시스템에 맞지 않는” 프로그램이라고 생각이 들어 **피벗팅**.
(심지어 강아지 배변 데이터셋을 발견하기도 했음.)

Mobilenet 등 연산 압축 방법들과 fewshot 등 방법들을 도입하기에는 내가 다 완성하지 못할 것 같다는 생각이 들음.

PM 카메라 보조기기 제안

OpenCV 와 기본적인 영상 처리 기법을 바탕으로,
퍼스널 모빌리티의 주행 환경 특성과, 주행 특징을 고려한
Vision 기반의 제한된 성능의 녹화 / Lane Detection / Depth Prediction 기기 제안

Pivoting (포기이유)

예전에, Neural Network 를 Raspberry pi 에 올려 보는 시도를 해 보았습니다.

혹시 이번에 Neural Network 를 올리려는 사람이 있을 것 같아서,

CNN 기반 개 똥 인지 임베디드 기기를 주제로 하고, 최적화 방법을 제안해 보고자 만들었던 슬라이드를 잠깐 공유합니다.

이때, 너무 안 올라가고 어려웠기에 이번에는 임베디드 기기다운 순수 OpenCV 응용 프로젝트를 해 보기로 했습니다.

개 똥 인지
임베디드
기기

성능
0.8FPS

발열..
어마어마한

Model : Tiny-YOLO v1 (2016)
(13 layer)

Darkflow <https://github.com/thtrieu/darkflow>
(Darkent Tensorflow 1.3 Implementation)

Tensorflow Python API
하도 무거워서 설치 중 잘 터짐. 7번만에 성공

Raspbian OS

Raspberry Pi 4B, (RAM 4G) + Raspberry Pi Camera

Pivoting (포기이유)

예전에, Neural Network 를 Raspberry pi 에 올려 보는 시도를 해 보았습니다.

혹시 이번에 Neural Network 를 올리려는 사람이 있을 것 같아서,

CNN 기반 개 똥 인지 임베디드 기기를 주제로 하고, 최적화 방법을 제안해 보고자 만들었던 슬라이드를 잠깐 공유합니다.

이때, 너무 안 올라가고 어려웠기에 이번에는 임베디드 기기다운 순수 OpenCV 응용 프로젝트를 해 보기로 했습니다.

개 똥 인지
임베디드
기기

성능
???

발열..
어마어마한

F.C Layer 제거, Mobilenet 구조+
다양한 모델 압축방법 등 도입
그리고 직접 코드 작성

이 추상계층 삭제

Tensorflow Lite 또는
그에 준하게 가벼운 Lib

Raspbian OS

Raspberry Pi 4B, (RAM 4G) + Raspberry Pi Camera
우리가 받은 보드는 **RAM 2G** 임.

주제 및 선정 이유

PM 카메라 보조기기 제안

OpenCV 와 기본적인 영상 처리 기법을 바탕으로,
퍼스널 모빌리티의 주행 환경 특성과, 주행 특징을 고려하여,
제한된 성능의 녹화 / Lane Detection / Depth Prediction 을 수행할 수 있는 기기 제안

퍼스널 모빌리티 시장의 성장

법적 개선에 따른 환경 개선 : 퍼스널 모빌리티의 자전거도로 진입 허용 개정법 국회 통과
소형 기기 기록 녹화 필요성 : 퍼스널 모빌리티도 사고가 날 가능성이 있으므로, 촬영장치가 요구됨.

새로운 시도

개인용 프리미엄 퍼스널모빌리티 : 자동 line tracing 이나 오토크루즈 기능 탑재 시도
공유 퍼스널모빌리티의 수거 문제 해결 : 발전되면 도시에 난립한 퍼스널 모빌리티를 자동으로 수거하게 만들 수 있을까 에 대한 실험

주제 및 선정 이유

'면허 없이 운전' 전동 킥보드 제조사 주가 급등

조선비즈 | 박정엽 기자

입력 2020.05.22 10:05

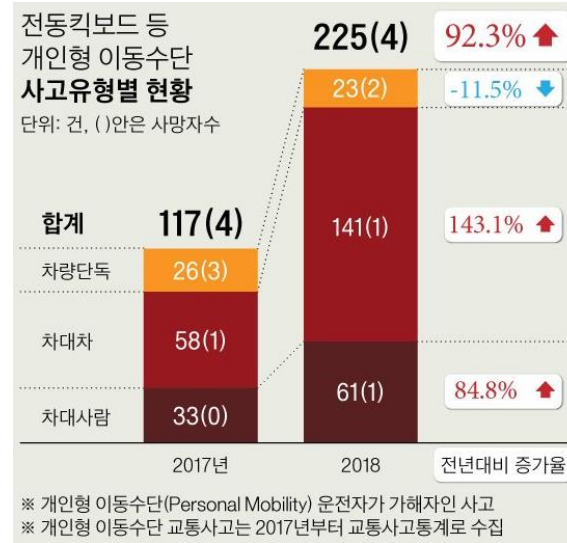
전동 킥보드 관련 규제를 완화해 사용 범위를 넓히도록 한 법안이 국회를 통과하면서 이들 기기를 제조하는 자전거 업체들의 주식이 급등하고 있다.

두 배씩 증가하는 사고에도, 블랙박스 등 사고검토장치 상품이 존재하지 않음

2020년, 전동킥보드 자전거도로 주행 허용 검토

- 시장 커지며 안전에 대한 인식 높아질 것
- 단일 차선의 한강변 자전거 도로에서 속력 오토크루즈 기능을 원하는 사람들 많아질 것

퍼스널 모빌리티 자율주행 보조 기기 제안



Google

퍼스널모빌리티 블랙박스

검색결과 약 160,000개 (0.33초)

www.mimint.co.kr > article > board_view

맨 인 블랙박스 퍼스널 모빌리티 뺑소니범, 블랙박스에 포착된 ...
2019. 8. 31. - 31일 '맨 인 블랙박스'에서는 퍼스널모빌리티 뺑소니 사고에 대해 알아보고, 개인형 이동수단과의 뺑소니 사고 예방을 위한 제도적인 대책을 모색해 ...

www.youtube.com > watch

[맨 인 블랙박스] Ep.237 예고 추적! 뺑소니 전담반 '퍼스널 ...

19년 08월 31일 Sat(토) 밤 08:45PM 본방송 #맨인블랙박스 #뺑소니 #전동킥보드 > Subscribe NOW! ... Hit-and-run task ...
2019. 8. 29. - 업로더: SBS NOW

www.ijan.kr > news > articleView

[맨 인 블랙박스] 전동 킥보드(퍼스널모빌리티) 뺑소니범, 블랙 ...

2019. 8. 31. - 31일, '맨 인 블랙박스'에서는 퍼스널모빌리티 뺑소니 사고에 대해 알아보고, 개인형 이동수단과의 뺑소니 사고 예방을 위한 제도적인 대책을 모색해 ...

www.evshop.co.kr > goods > goods_view

아이나비 INAVI - 이브이샵 퍼스널모빌리티 No.1

아이나비 전동킥보드 블랙박스 TC-1 자전거 겸용 Full HD, G-Sensor. 이전 상품 이미지. 아이나비 전동킥보드 블랙박스 TC-1 자전거 겸용 Full HD, G-Sensor.

it.chosun.com > site > data > html_dir > 2020/04/03

"대세는 퍼스널 모빌리티" 톱크웨어, 전동 킥보드 출시 - IT조선 ...

2020. 4. 3. - 내비게이션-블랙박스 전문기업 톱크웨어가 퍼스널 모빌리티 영역으로 사업을 확대한다. 접이식 전동 킥보드로 제품군 강화에 나선다. 톱크웨어는 ...

청소년 이용 많고 인도운행 잦아 '규제 사각지대'
보험상품 없어 접촉사고 제대로 보상받기 어려워
국정현안점검회의 "개인형 이동수단 내년 법제화"



I / O



1080x720x3

*No depth camera
*No rider / lidar sensor

RGB Pi
Cam1

RGB Pi
Cam2

Power Input

c type power input from notebook
5V, 3A

Open
CV
python

10~30 FPS

Stabilized
Video

Save
video

Video
file

400x240x3

Bicycle
Lane

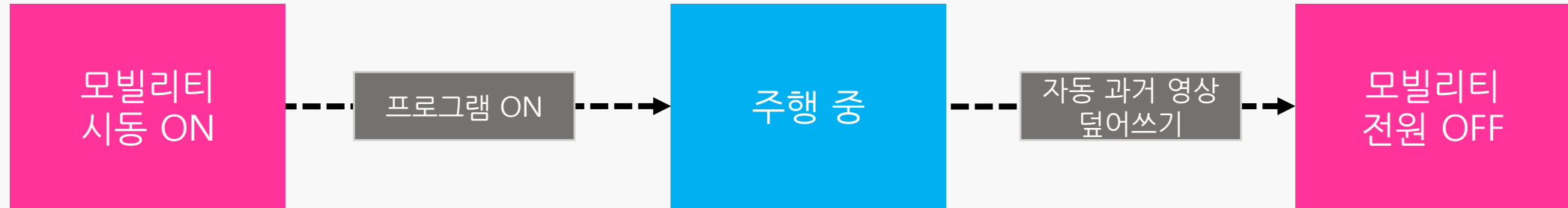
800x480x3

display

전방
Depth

800x480x3

Interface



Display : 물체 없음



Display :
Lane 내부에 물체 있음



*사용자가 주행 영상을 보게 될텐데, 모빌리티가 덜컥거리면 보기 불편할 것으로 예상하여, 아까 Stabilization 을 언급함.

퍼스널 모빌리티 자율주행 보조 기기 제안

제한 조건



왼쪽 그림과 같은 형태로 핸들과 나란히 부착될 것이기 때문에 두 카메라가 Y 좌표값이 동일하다는 것을 가정함.

자전거도로는 왕복 2차선, 편도 1차선임을 가정함.

디스크에 계속 저장할 수 없으므로,
전원이 들어오고 프로그램을 실행할 때
또는 프로그램을 종료할 때마다
기존 블랙박스 영상이 삭제된다는 것을 전제함.
이를 선택할 수 있는 UI 가 잘 구현되었다고 가정함.

배터리가 전부 방전되기 전에 영상을 저장한다고 가정함.

자전거도로에서만 정상적으로 작동하므로,
자전거도로에서만 모빌리티를 탑승한다고 가정함.

낮에만 주행한다고 가정함.

개발 일정

	1주		2주		3주		4주	
	월화	수목금	월화	수목금	월화	수목금	월화	수목금
교수님 accept								
듀얼카메라 모듈 구매								
OpenCV / Python 환경 마련								
단일카메라 Lane Detection 패키지/소스코드 준비								
단일카메라 Lane Detection 테스트								
단일카메라 곡선 Lane Detection 소스코드 준비								
단일카메라 영상 저장 패키지/소스코드 준비/테스트								
듀얼카메라 Real time Depth 패키지/소스코드 준비								
듀얼카메라 모듈 배송 완료								
캘리브레이션 아크릴판 제작								
듀얼 카메라 캘리브레이션								
캘리브레이션 코드 결합								
캘된 단일 카메라 Lane Detection 성능 테스트								
듀얼카메라 Real time Depth 소스코드 병합								
Real time Depth 계산 영역 Lane Detection 결과 이용 최적화								
Real time Depth로부터 Roi Box 그리기								
테스트 영상 제작								
발표자료 제작								