

중간 보고서

개발환경

Tensorflow lite설치

이미지 전처리

이미지 추론 프로그램

모든 코드는 <https://github.com/hoo0681/raspi4withTF> 에서 볼 수 있습니다.

중요포인트 : 카메라가 엄청난 광각이라 거리에 비해 크기가 심하게 작아짐(추후 반드시 해결방안 마련해야함)

Tensorflow lite문법 사용에 있어 까다로운 부분이 있어 helper함수로 틀을 만들어 놓았음

개발환경

Window10

Anaconda 1.9.7

Python3.6

CUDA 10.0

cuDNN 7.6

Visual studio 2017

Tensorflow 2.0.0

설치

Anaconda 설치

Visual studio 2017 설치

CUDA 설치 주의 2.0.0 은 CUDA 을 10.0 버전으로 반드시 설치

cuDNN 설치

Anaconda 명령줄에서 pip install tensorflow-gpu==2.0.0

자세한사항은 <https://chan-lab.tistory.com/13> 참조 (cuDNN&CUDA)

기타 github 를 사용하여 desktop 에서 프로그램 수정하고 commit 하는 식으로 진행

Tensorflow lite설치

자세한 설명 https://www.tensorflow.org/lite/guide/build_rpi#compile_natively_on_raspberry_pi

sudo apt-get install build-essential

Cd tensorflow/

./tensorflow/lite/tools/make/download_dependencies.sh

./tensorflow/lite/tools/make/build_rpi_lib.sh

<https://www.tensorflow.org/lite/guide/python> 에서-

[tfllite_runtime-1.14.0-cp37-cp37m-linux_armv7l.whl](#) 다운로드 받고

Pip3 install tfllite_runtime-1.14.0-cp37-cp37m-linux_armv7l.whl

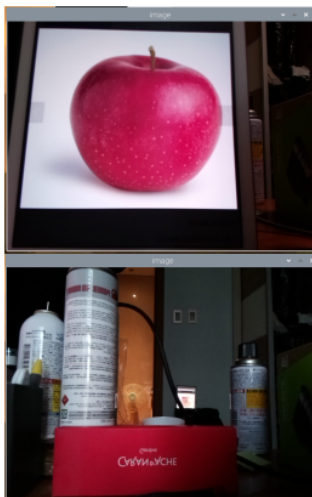
하면 끝

이미지 전처리

꼭지점 찾기 <https://datascienceschool.net/view-notebook/f9f8983941254a34bf0fee42c66c5539/>

참조 <https://webnautes.tistory.com/1384>

문제: 카메라가 너무 광각이라 왜곡이 너무 심함
(거리가 멀어지면 너무 작게 보여 인식 불가 가능성 높음)



위 아래 모두 동일한 사과 이미지를 아이폰에 출력한 것을 촬영한 것이며

아래 이미지에서 카메라와 사과 이미지와의 거리는 1m50cm임.

(추후해결 반드시 필요!!!)

1. 색필터로 출력

Opencv 는 BGR 형식으로 출력/입력 받는다.

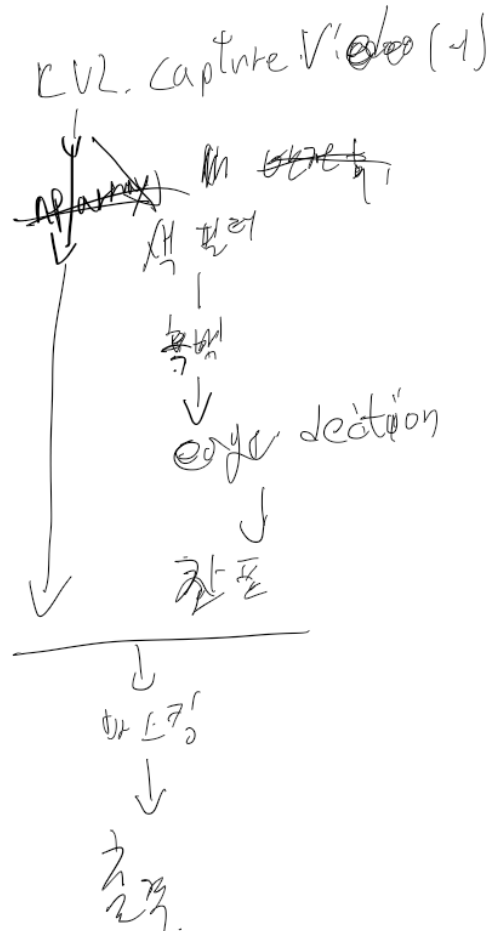
(왜 BGR 을 사용하는가? <http://wanochoi.com/?p=4326>)

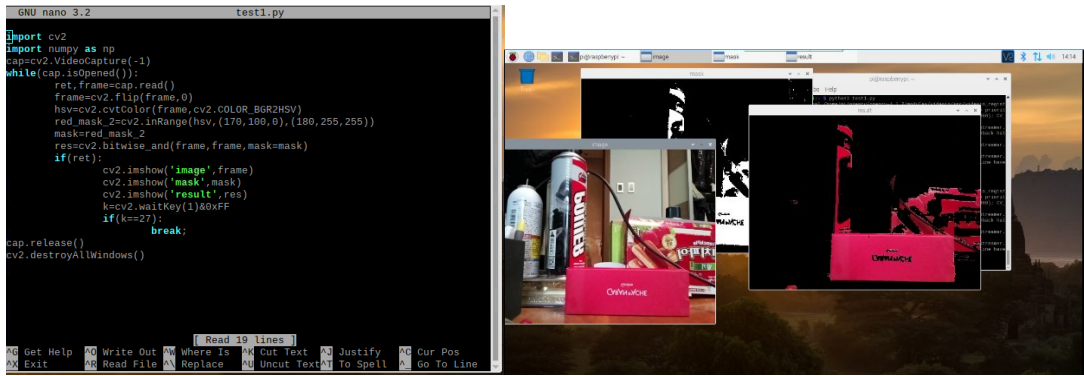
색필터는 hsv 색영역을 사용한다

hsv 개념 <https://bskyvision.com/46>

Hsv 활용예제 <https://076923.github.io/posts/Python-opencv-15/>

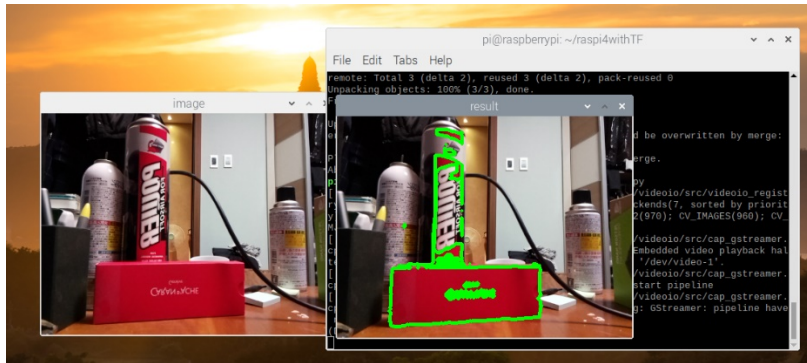
적용



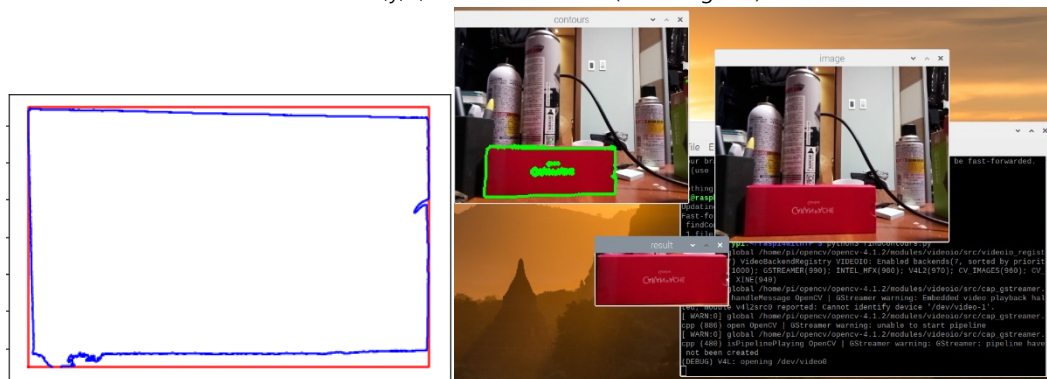


2. 사각형인식->상황에 따라 사각형이 안될수도 있음 +==> 가장 큰 덩어리 찾기로 변경
마스크를 이진화 해서 컨투어를 찾고 컨투어를 둘러싸는 사각형 구하기
컨투어 사용법이 opencv 4.x 버전부터 사용법이 달라짐-> 엄청 고생 중
이미지는 (세로,가로)

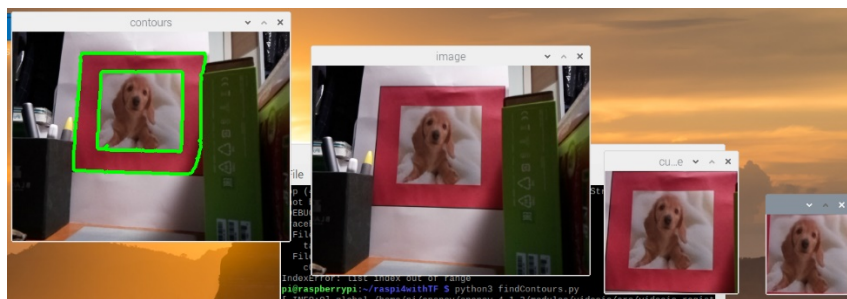
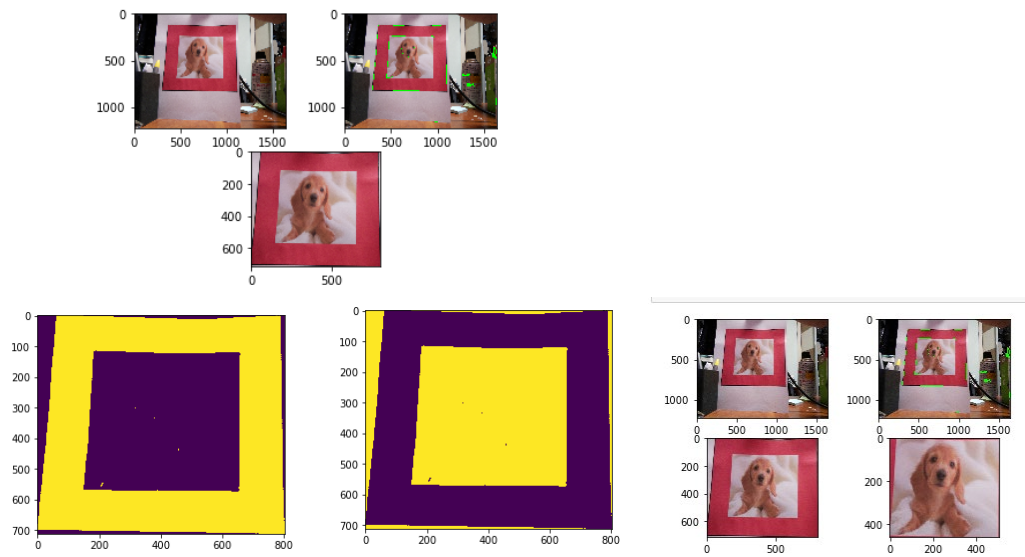
색필터상에서 윤곽선을 잡기위해 findContours 함수사용 -> 엄청많은 contour 를 찾음
가장 큰 것을 찾기위해 contourArea 를 사용하여 가장 넓은 영역탐색



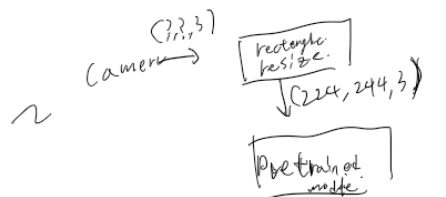
3. 사각형 각 꼭지점 좌표로 이미지 잘라내기& 잘라낸이미지 개별출력
2.에서 찾은 덩어리를 네모나게 잘라 x,y,w,h 를 반환하는 함수(boundingRect)를 사용하여 잘라냄



4. 태두리를 제외 해서 사진을 자르고 싶음
태두리포함해서 자른 사진의 마스크를 반전해서 빨간색이 아닌 컨투어를 찾아 자르기



이미지 추론 프로그램 개요



이미지 크기 조정, 자르기 → NumPy 사용

모델은 데스크톱에서 만들기

뉴럴컴퓨팅스틱은 호환성 보장 X arm7 용은 제공되지만 arm8 인 라즈베리 4 는 아직 보장 X

https://www.tensorflow.org/lite/convert/python_api lite 로 변환

Tflite.interpreter(모델.tflite 주소)하면 모델 로딩 완료

입출력을 위한 tensor 의 index 를 알아내어 tensor 할당을 해야 사용가능

반드시 예러 Ex)

```
RuntimeError: There is at least 1 reference to internal data
in the interpreter in the form of a numpy array or slice. Be sure to
only hold the function returned from tensor() if you are using raw
data access.
```

가 난 경우 interpreter 를 재정의 하여야한다.

Input tensor 는 interpreter.tensor 로 numpy 뷰 포인터를 받아와서 사용해야한다.

Interpreter.allocate_tensors() : 모델의 tensor 를 사용가능하게 해준다. 자세한 것은

<https://github.com/tensorflow/tensorflow/blob/1cf0898dd4331baf93fe77205550f2c2e6c90ee5/tensorflow/lite/python/interpreter.py#L258> 참조

Interpreter.get_input_details() : 모델의 입력 tensor 의 정보(name,index,shape,dtype,quantization)의 리스트를 반환함

~~Interpreter.tensor() : Interpreter 의 메소드인 tensor(인덱스번호) 는 인수로 받은 인덱스의 tensor 의 numpy view 를 반환 즉, numpy 변수 처럼 사용가능해진다.~~

Interpreter.tensor(index) : 함수다. 반환값은 numpy array 를 반환하는 함수 포인터 이고 반환값을 사용하거나 .copy 를 사용하여야 한다. 함수를 단독으로 호출 할 경우 이후 인터프리터를 사용하는 작업들이 에러를 반환한다. EX)

```
asdf=output()
Interpreter.invoke()

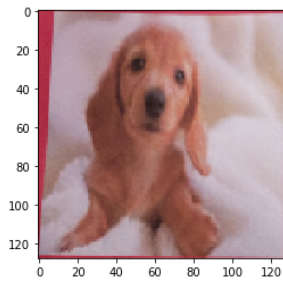
~#Anaconda3#envs#tensorflow2.0#lib#site-packages#tensorflow_core#lite#python#interpreter.py in invoke(self)
  450     ValueError: When the underlying interpreter fails raise ValueError.
  451     """
--> 452     self._ensure_safe()
  453     self._interpreter.invoke()
  454

~#Anaconda3#envs#tensorflow2.0#lib#site-packages#tensorflow_core#lite#python#interpreter.py in _ensure_safe(self)
  270     in the interpreter in the form of a numpy array or slice. Be sure to
  271     only hold the function returned from tensor() if you are using raw
--> 272     data access."""
  273
  274     def _get_tensor_details(self, tensor_index):

RuntimeError: There is at least 1 reference to internal data
in the interpreter in the form of a numpy array or slice. Be sure to
only hold the function returned from tensor() if you are using raw
data access.
```

(반드시 <https://github.com/tensorflow/tensorflow/blob/r2.0/tensorflow/lite/python/interpreter.py#L390> 확인)
홈페이지에서 다운로드 받은 모델로 테스트성공

```
In [146]: plt.imshow(what)
plt.show()
```



```
In [217]: import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
```

```
In [139]: def load_interpreter(model_path):
interpreter_=tf.lite.Interpreter(model_path=model_path)
interpreter_.allocate_tensors()
input_ =interpreter_.tensor(interpreter_.get_input_details()[0]['index'])
output = interpreter_.tensor(interpreter_.get_output_details()[0]['index'])
return {'model': interpreter_, 'input': input_, 'output': output}
```

```
In [140]: dic=load_interpreter("C:/Users/hoo/Documents/_hoo/tensorflow_2_ass/mobilenet_v1_1.0_224_quant_and_labels/mobilenet_v1_0.25_128_quant/mobile
```

```
In [148]: test_data=cv2.resize(test_image,(128,128))
dic['input']()[0][:,:]=test_data
```

```
In [142]: dic['model'].invoke()
```

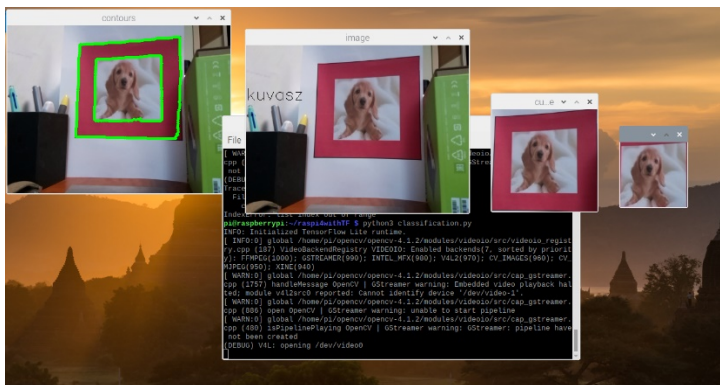
```
In [143]: np.argmax(dic['output']()[0])
```

```
Out [143]: 208
```

```
In [147]: labels[208]
```

```
Out [147]: 'golden retriever'
```

라즈베리파이에서도 성공



Helper function

```
def load_interpreter(model_path):
    interpreter_=tf.lite.Interpreter(model_path=model_path)
    interpreter_.allocate_tensors()
    input_ =interpreter_.tensor(interpreter_.get_input_details()[0]['index'])
    output = interpreter_.tensor(interpreter_.get_output_details()[0]['index'])
    return {'model': interpreter_, 'input': input_, 'output': output}
```

```
        return {'model': interpreter_, 'input': input_, 'output': output}

def load_labels(path):
    with open(path, 'r') as f:
        return {i: line.strip() for i, line in enumerate(f.readlines())}

dic=load_interpreter("/home/pi/Downloads/mobilenet_v1_1.0_224_quant.tflite")
labels=load_labels("/home/pi/Downloads/labels_mobilenet_quant_v1_224.txt")

test_data=cv2.resize(target_image,(224,224))
dic['input']()[0][:,:]=test_data
dic['model'].invoke()
ans=labels[np.argmax(dic['output']()[0])]
```