Object detection w/ your Raspberry Pi

Computer Vision with your SoC computers

Calvin Tsang, Open Source HK Meet up #76
6-Jun-2024

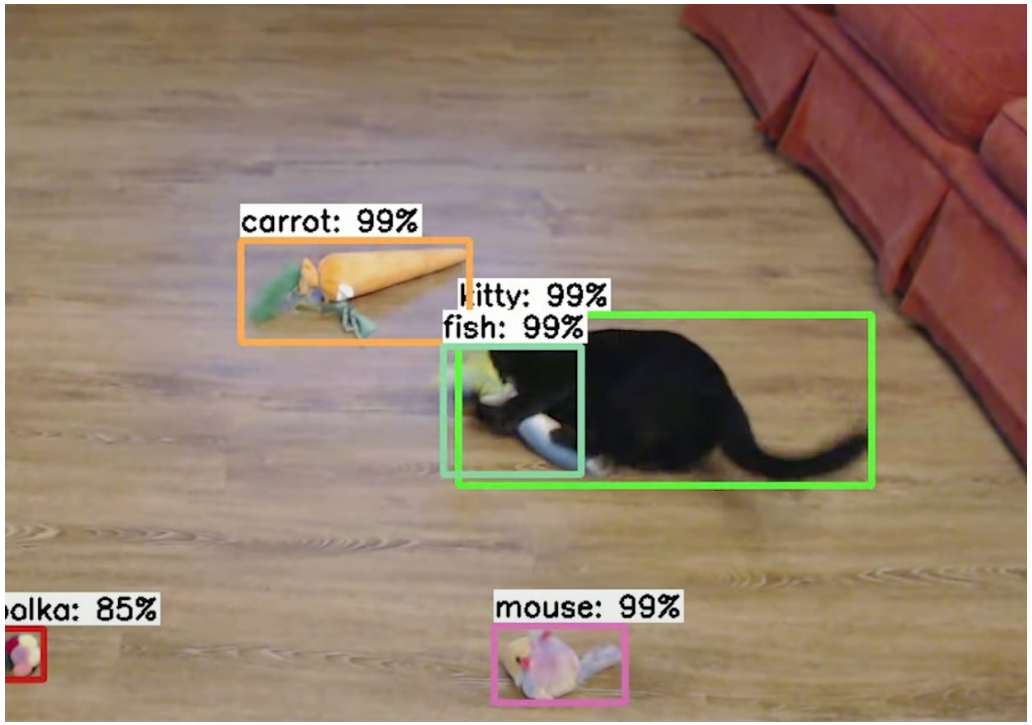
OPEN SOURCE
HONG KONG

Preparation

- Raspberry Pi 4B+ or 5
- USB Camera / Rpi Camera
- Pre-trained machine learning model (Tensorflow file .pb)

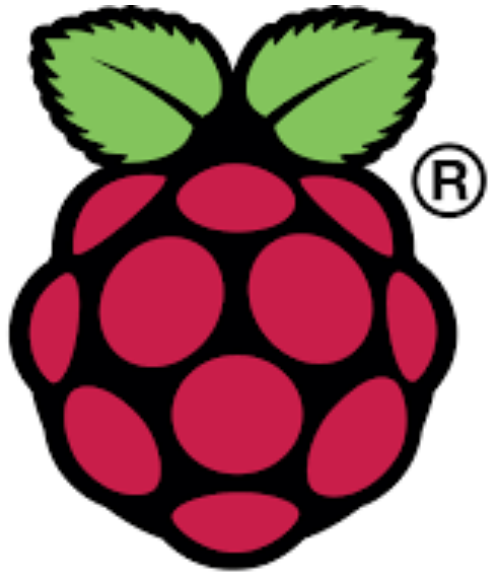
Objective

Spot out the objects with the camera



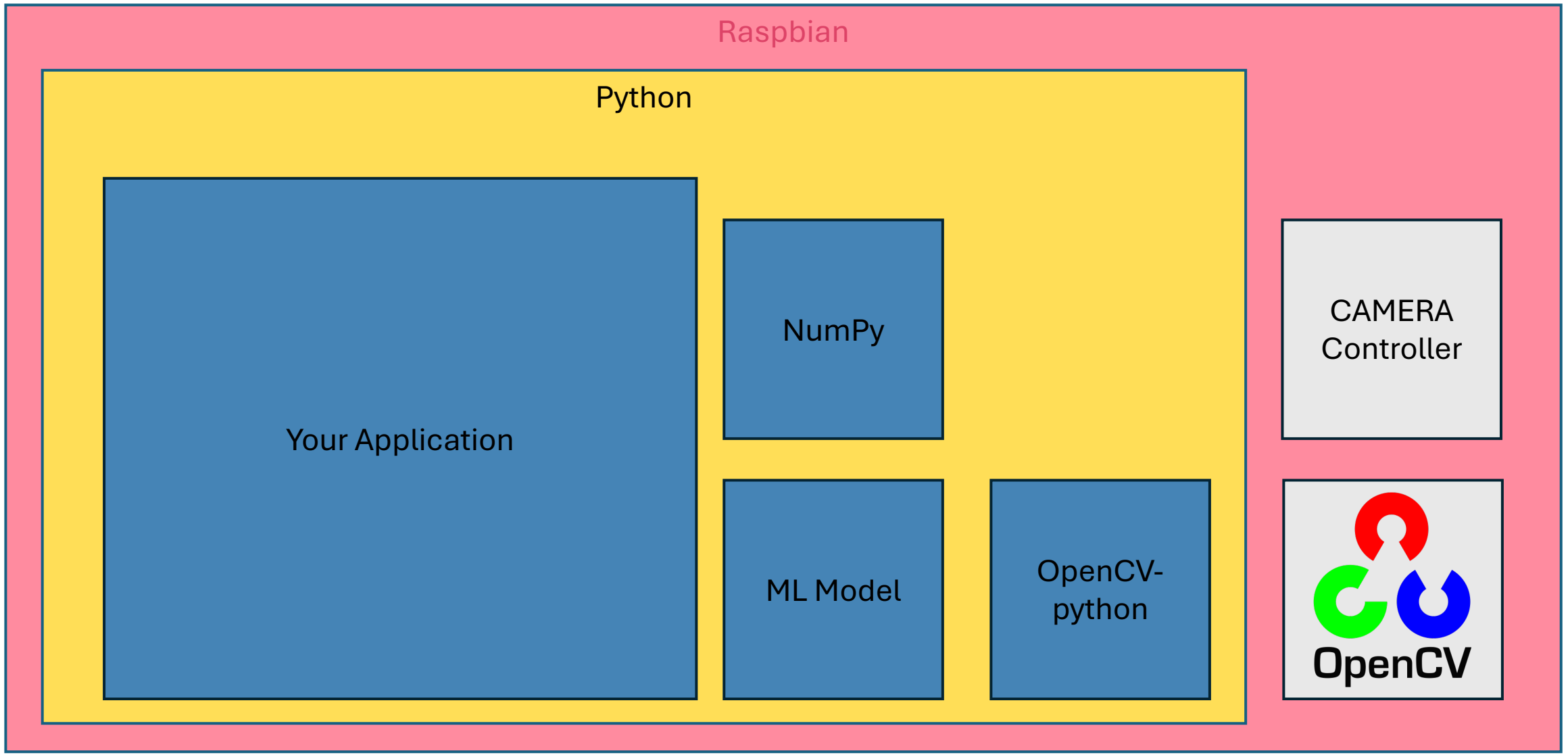
Why Raspberry Pi?

- Python code friendly
- Better support with hardware
- Bigger community
- Better documentation
- Standardized hardware form factor



Raspberry Pi

Architecture



Installation (1) OpenCV

```
sudo apt update
sudo apt upgrade
sudo apt install python3-opencv
```

(Alt.) Build OpenCV by your own … Config & Code

- > sudo nano /etc/dphys-swapfile
- Update CONF_SWAPSIZE=100 to CONF_SWAPSIZE=2048
- Restart Rpi with updated Swapfile
  > sudo systemctl restart dphys-swapfile
- git clone https://github.com/opencv/opencv.git
  git clone https://github.com/opencv/opencv_contrib.git

(Alt.) Build OpenCV by your own … Compiling

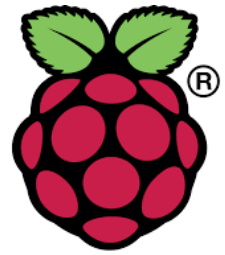
- > mkdir ~/opencv/build
  cd ~/opencv/build
- ```
  > cmake -D CMAKE_BUILD_TYPE=RELEASE \
          -D CMAKE_INSTALL_PREFIX=/usr/local \
          -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
          -D ENABLE_NEON=ON \
          -D WITH_OPENMP=ON \
          -D ENABLE_VFPV3=OFF \
          -D BUILD_TESTS=OFF \
          -D INSTALL_PYTHON_EXAMPLES=OFF \
          -D OPENCV_ENABLE_NONFREE=ON \
          -D CMAKE_SHARED_LINKER_FLAGS=-latomic \
          -D OPENCV_PYTHON_INSTALL_PATH=/usr/lib/python3/dist-packages \
          -D BUILD_EXAMPLES=OFF ..
  ```
- Speed up the compilation process with available processors
  >make -j$(nproc)
  ```

(Alt.) Build OpenCV by your own … Installation & Rollback

- Installation
  > sudo make install
- regenerate the operating systems library link cache
  > sudo ldconfig
- Edit SwapFile Size
  > sudo nano /etc/dphys-swapfile
- CONF_SWAPSIZE=2048 to CONF_SWAPSIZE=100
- Restart system and apply updated config
  > sudo systemctl restart dphys-swapfile

Installation (2) Video Library for Raspbian

- sudo apt-get install build-essential cmake pkg-config
- sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
- sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
- sudo apt-get install libxvidcore-dev libx264-dev
- sudo apt-get install libgtk2.0-dev libgtk-3-dev
- sudo apt-get install libatlas-base-dev gfortran



Raspberry Pi

Installation (3) Python Library

- sudo pip3 install numpy


Show me the code

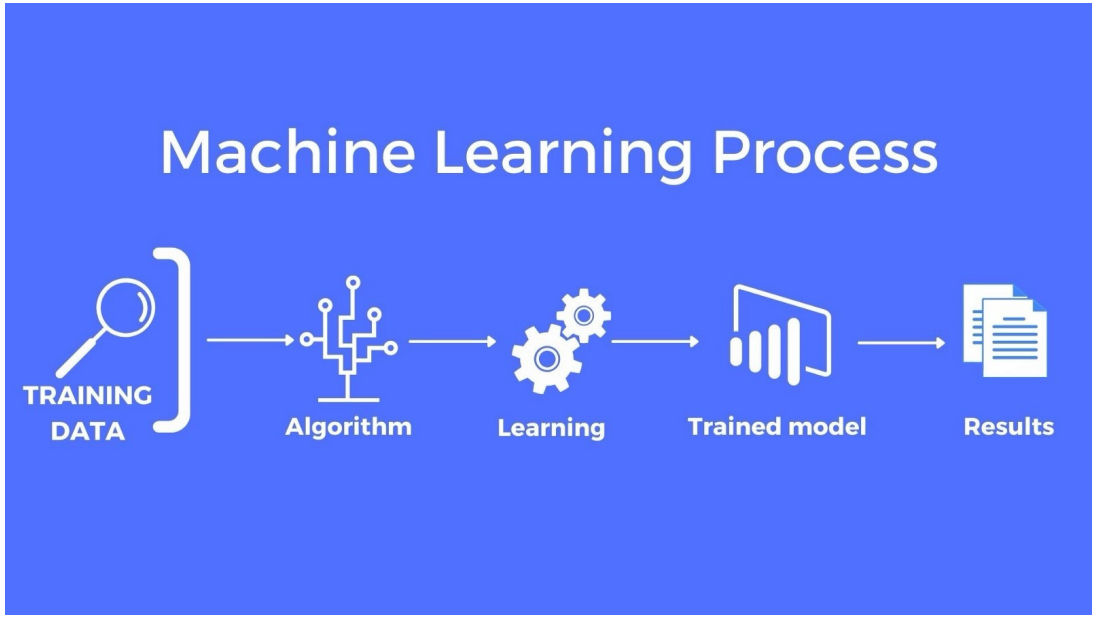
1) Connect Camera

1) Connect your camera w/ Python

```python
import cv2

video_capture = cv2.VideoCapture(0) # connect to the first USB camera
cv2.__version__
video_capture.set(3,640) # set Video Capture width
video_capture.set(4,480) # set Video Capture height
while True: # Keep looping until the 'q' key is pressed
    result, video_frame = video_capture.read()  # read frames from the video
    if result is False:
        break  # terminate the loop if the frame is not read successfully

    cv2.imshow(
        "USB Camera Test", video_frame
    )

    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

video_capture.release()
cv2.destroyAllWindows()
```

Demo (1)

How to enable the computer vision in Rpi?



- Pre-trained Machine Learning Model – Coco Library (Common Object in Context)
- Coco Lib is large-scale object detection, segmentation, and captioning dataset.
- Pre-trained Model is a blackbox program you pass input and return output, we can reuse it
- .pb file

2) Catch Object from your camera

2a) Hock the Pre-trained ML model

```python
import cv2

thres = 0.6 # Threshold to detect object

classNames = []
classFile = "[redacted]Object_Detection_Files/coco.names"
with open(classFile,"rt") as f:
    classNames = f.read().rstrip("\n").split("\n")

configPath = "/ho[redacted]ect_Detection_Files/ÍÍ_coco_2020_01_14.pbtxt"
weightsPath = "[redacted]Object_Detection_Files/frozen_inference_graph.pb"

net = cv2.dnn_DetectionModel(weightsPath,configPath) # Hook the model to the configuration and weights
net.setInputSize(320,320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)
```

2b) Capture and spot out the object

```
20  def getObjects(img, thres, nms, draw=True, objects=[]):
21      classIds, confs, bbox = net.detect(img,confThreshold=thres,nmsThreshold=nms) # detect the image with threshold confi
22      #print(classIds,bbox)
23      if len(objects) == 0: objects = classNames
24      objectInfo =[]
25      if len(classIds) != 0:
26          for classId, confidence,box in zip(classIds.flatten(),confs.flatten(),bbox):
27              className = classNames[classId - 1]
28              if className in objects:
29                  objectInfo.append([box,className])
30                  if (draw):
31                      cv2.rectangle(img,box,color=(0,255,0),thickness=2)
32                      cv2.putText(img,classNames[classId-1].upper(),(box[0]+10,box[1]+30),
33                      cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
34                      cv2.putText(img,str(round(confidence*100,2)),(box[0]+200,box[1]+30),
35                      cv2.FONT_HERSHEY_COMPLEX,1,(0,255,0),2)
36
37      return img,objectInfo
```

2c) Follow up action

```
40  if __name__ == "__main__":
41
42      cap = cv2.VideoCapture(0)
43      cap.set(3,640)
44      cap.set(4,480)
45      #cap.set(10,70)
46
47
48      while True:
49          success, img = cap.read()
50          result, objectInfo = getObjects(img,thres,0.2)
51          for i in objectInfo:
52              #print(objectInfo)
53              print(i[1]) # ANYTHING you want to do with the object detected
54
55          cv2.imshow("Output",img)
56          cv2.waitKey(1)
57
```

Demo (2)

Caution

- Use the branded USB camera or Pi Camera so that it can be recognize

Reference

- Installing OpenCV on the Raspberry Pi
  https://pimylifeup.com/raspberry-pi-opencv/
- Install OpenCV on a Raspberry Pi (compatible with all RPi boards)
  https://randomnerdtutorials.com/install-opencv-raspberry-pi/
- Source Code
  https://github.com/DebugTsang/rpi-obj-recognition
- Common Objects in Context (Coco)
  https://cocodataset.org/#home