smitshetye /
Realtime-Object-Detection-in-Python-Jupyter-Note-Book-Open-CV-Kandi-Kits-By-Smit-Shetye Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Realtime-Object-Detection-in-Python-Jupyter-Note-Book-Open-CV-Kandi-Kits-By-Smit-Shetye /
Realtime_Object_Detection_in_Python_Jupyter_Note_Book_Open_CV_Kandi_Kits_By_Smit_Shetye.ipynb

smitshetye Add files via upload History

1 contributor

245 lines (245 sloc) | 11.7 KB

Realtime Object Detection

Detects objects in realtime video streaming via webcam

Import libraries for preprocessing and annotation

In [1]:
```
import numpy as np
import cv2
from pathlib import Path
from IPython.display import Image
import torch
```

Load pretrained models

In [2]:
```
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
```

```
Using cache found in C:\Users\espn/.cache\torch\hub\ultralytics_yolov5_master
YOLOv5  2022-7-15 Python-3.8.10 torch-1.8.1+cpu CPU

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
Adding AutoShape...
```

Run object detection on example image

In [11]:
```
imgs = ['https://smitshetye.github.io/images/smitshetye.jpg']
Image(url=imgs[0])
```

Out[11]:

In [4]:
```
results = model(imgs)
results.print()
results.save(".")
```

```
WARNING: NMS time limit 0.330s exceeded
Saved 1 image to runs\detect\exp8
image 1/1: 708x716 1 person, 1 tie
Speed: 3878.2ms pre-process, 7077.4ms inference, 490.0ms NMS per image at shape (1, 3, 640, 640)
```

In [12]:
```
Image(filename='smitshetye.jpg')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [12], in <cell line: 1>()
----> 1 Image(filename='smitshetye.jpg')

File D:\Program Files\Python38\lib\site-packages\IPython\core\display.py:957, in Image.__init__(self, data, url, filename, format, embed, width, height, retina, unconfined, metadata, alt)
    955 self.unconfined = unconfined
    956 self.alt = alt
--> 957 super(Image, self).__init__(data=data, url=url, filename=filename,
    958         metadata=metadata)
    960 if self.width is None and self.metadata.get('width', {}):
    961     self.width = metadata['width']

File D:\Program Files\Python38\lib\site-packages\IPython\core\display.py:327, in DisplayObject.__init__(self, data, url, filename, metadata)
    324 elif self.metadata is None:
    325     self.metadata = {}
--> 327 self.reload()
    328 self._check_data()

File D:\Program Files\Python38\lib\site-packages\IPython\core\display.py:992, in Image.reload(self)
    990 """Reload the raw data from file or URL."""
    991 if self.embed:
--> 992     super(Image,self).reload()
    993     if self.retina:
    994         self._retina_shape()

File D:\Program Files\Python38\lib\site-packages\IPython\core\display.py:353, in DisplayObject.reload(self)
    351 if self.filename is not None:
    352     encoding = None if "b" in self._read_flags else "utf-8"
--> 353     with open(self.filename, self._read_flags, encoding=encoding) as f:
    354         self.data = f.read()
    355 elif self.url is not None:
    356     # Deferred import

FileNotFoundError: [Errno 2] No such file or directory: 'smitshetye.jpg'
```

Run object detection on realtime video via webcam

In [ ]:
```
print("Press q to exit the object detection window!")
cap = cv2.VideoCapture(0)
while False:
    ret, image_np = cap.read()
    results = model(image_np)
    df_result = results.pandas().xyxy[0]
    dict_result = df_result.to_dict()
    scores = list(dict_result["confidence"].values())
    labels = list(dict_result["name"].values())

    list_boxes = list()
    for dict_item in df_result.to_dict('records'):
        list_boxes.append(list(dict_item.values())[:4])
    count = 0

    for xmin, ymin, xmax, ymax in list_boxes:
        image_np = cv2.rectangle(image_np, pt1=(int(xmin),int(ymin)), pt2=(int(xmax),int(ymax)), \
                                 color=(255,0, 0), thickness=2)
        cv2.putText(image_np, f"{labels[count]}: {round(scores[count], 2)}", (int(xmin), int(ymin)-
                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36,255,12), 2)
        count = count + 1

    cv2.imshow('Object Detector', image_np);

    if cv2.waitKey(1) & 0xFF == ord('q'):
        cap.release()
```

```
        cv2.destroyAllWindows()
        print("The window has been exited!")
        break
```

```
Press q to exit the object detection window!
The window has been exited!
```

In [ ]: