

CSCE 5390 Multimedia Computing

Final Project Report

07/20/2022

Image Detection

Team Members

Jayakishan Presingu -- 11462920

Sita Swapna Lakshmi Challapalli -- 11445422

Venkata Jagadeesh Surapareddy -- 11508352

KrishnaReddy Chaduvu -- 11551974

Naga Sai Sumanth Muppasani -- 11554462

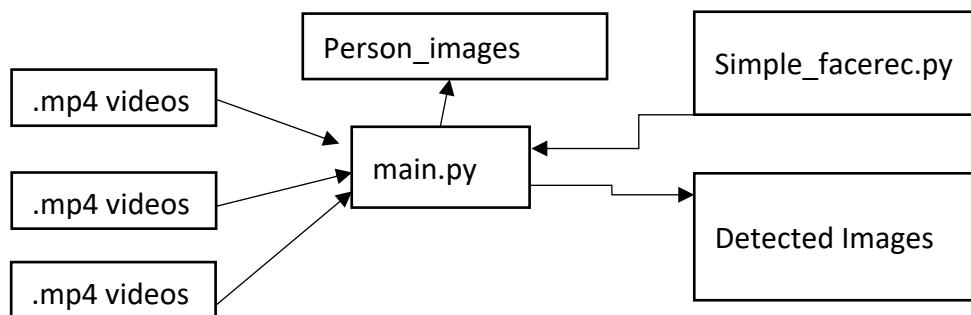
Introduction:

Image detection has been the new technology, various machine learning algorithms are trained to detect objects in the real time. Organizations like Tesla are trying to implement this technology as they are mainly focusing on automatic electric motor cars to reduce human intervention. These cars are installed with a software to detect any objects nearby like bus, car or a human crossing the road. This technology has developed motors that could understand the traffic signs, detects the motion of other vehicles and, they can communicate with each other using sensors.

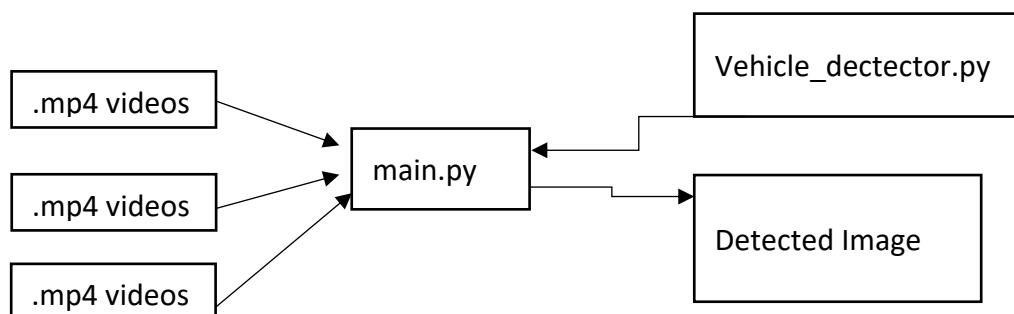
On the other hand, in various tech companies, swiping the Id card to notify the presence has become the old tradition, Face recognition is the new norm. Once the faces of the employees are registered, the software can detect the people without having any contact. Similarly, the fields of security and surveillance also use image recognition. Security firms and enforcement agencies use facial recognition technology to identify criminals. The photographs are analyzed by the deep learning algorithm, who compares them to potential suspects.

Design:

1. Human Detection Flowchart:



2. Vehicle Detection:



Workflow:

A code is developed in python by importing packages like NumPy, Face-recognition, Vehicle detection. Using machine learning algorithm code is developed to detect human and vehicles.

IDE: PyCharm

File structure:

The following files are imported into the IDE

Main.py: consist of code to detect the frames from the videos those containing user requested objects. Files: Facedetect.py and vehicle.py are imported to detect human images and vehicles.

Facedetect.py: consists of prerequisite modules required to detect human images

Vehicle.py: consists of prerequisite modules to detect vehicles like car, bus, and truck.

Human Image and Vehicle detection:

A 5 second HD videos containing different human and objects are imported and the reference to the Image is provided. The python code uses the YOLO algorithm. When user enters the name of the person/object he wants to find:

-> the code traverse through all the videos and matches the person image with all the videos.

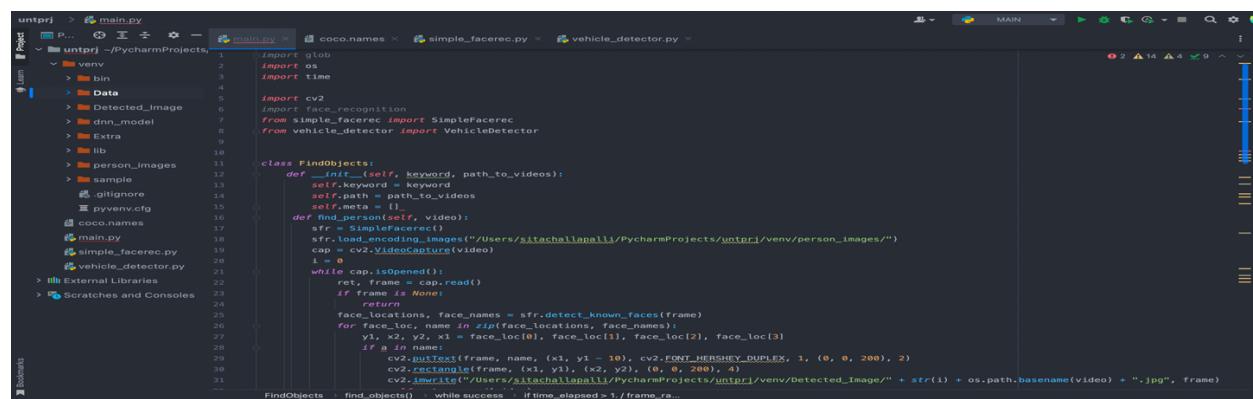
After all the frames of the videos are traversed an array of all videos that containing the person images.

->the code is trained against the four-wheeler vehicles. Thus, the code would be able to detect the vehicles like car, bus, and truck.

The detected frames containing the required images is stored in Detected_Image folder.

Implementation:

main.py:

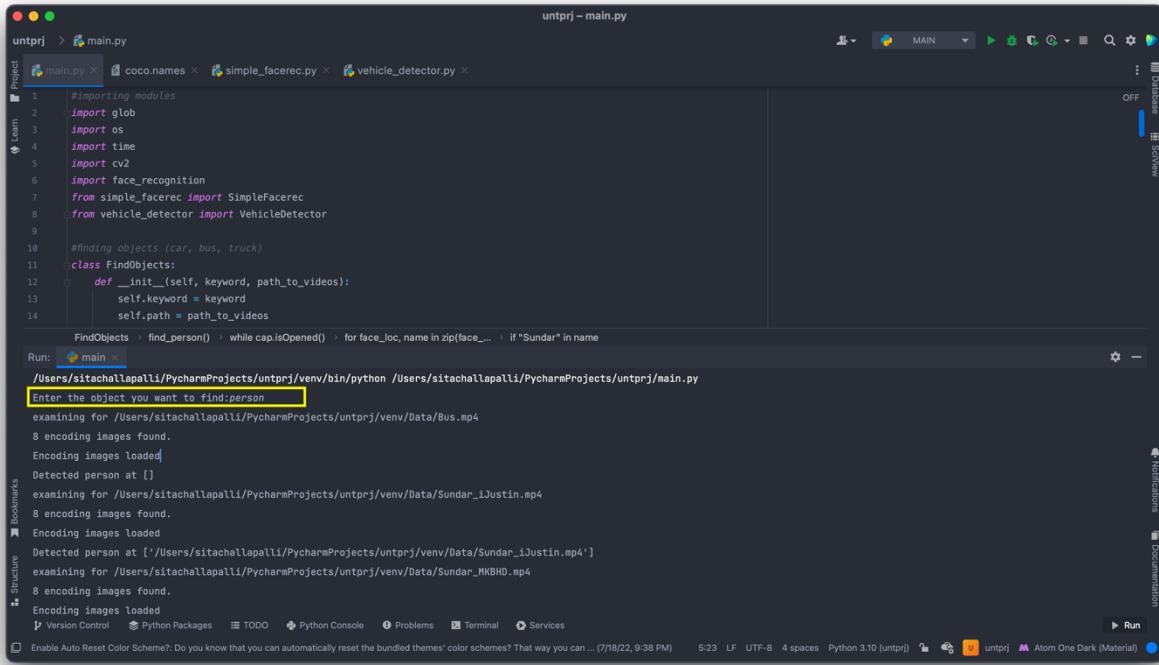


```
untpj ~ PycharmProjects/untprj/main.py
1 import glob
2 import os
3 import time
4
5 import cv2
6 import face_recognition
7 from simple_facerec import SimpleFacerec
8 from vehicle_detector import VehicleDetector
9
10 class FindObjects:
11     def __init__(self, keyword, path_to_videos):
12         self.keyword = keyword
13         self.path = path_to_videos
14         self.meta = []
15
16     def find(self, video):
17         sfr = SimpleFacerec()
18         sfr.load_encoding_images("/Users/sitachallapalli/PycharmProjects/untprj/venv/person_images/")
19         cap = cv2.VideoCapture(video)
20         i = 0
21         while cap.isOpened():
22             ret, frame = cap.read()
23             if frame is None:
24                 return
25             face_locations, face_names = sfr.detect_known_faces(frame)
26             for face_loc, name in zip(face_locations, face_names):
27                 y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2], face_loc[3]
28                 if name == self.keyword:
29                     cv2.putText(frame, name, (x1 - 10, y1 - 10), cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 200), 2)
30                     cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 200), 4)
31                     cv2.imwrite("/Users/sitachallapalli/PycharmProjects/untprj/venv/Detected_Image/" + str(i) + os.path.basename(video) + ".jpg", frame)
32
33         FindObjects > find_objects() : while success : if time_elapsed > 1 / frame_ra...
```

Human Image Detection Flow:

Import the .mp4 file containing the human images and a person image for image mapping. Initially a prompt is created to let the user find out which person the code should detect.

Example: We are trying to detect “Sundar”



The screenshot shows the PyCharm IDE interface. The project navigation bar at the top lists 'main.py', 'coco.names', 'simple_facerec.py', and 'vehicle_detector.py'. The main code editor window displays Python code for object detection, specifically for finding 'person' objects in a video. The code includes imports for modules like cv2, face_recognition, and SimpleFacerec. A class 'FindObject' is defined with a constructor '_init_'. The 'Run' tab in the bottom left shows the command '/Users/sitachallapalli/PycharmProjects/untpj/venv/bin/python /Users/sitachallapalli/PycharmProjects/untpj/main.py'. The terminal window below shows the execution of the script. It prompts 'Enter the object you want to find:person'. The output indicates it is examining files 'Bus.mp4' and 'Sundar_iJustin.mp4', finding 8 encoding images each, and detecting persons at specific frame ranges. The status bar at the bottom right shows the current time as 5:23 PM, the file type as LF, encoding as UTF-8, 4 spaces, Python 3.10 (untpj), and the theme as Atom One Dark (Material).

Input:

→ Enter the object you want to find: person

Reference Image given:



(Sundar.jpg)

Output:

Sundar was detected in the Sundar_iJustin.mp4 and Sundar_MKBHD.mp4 videos.

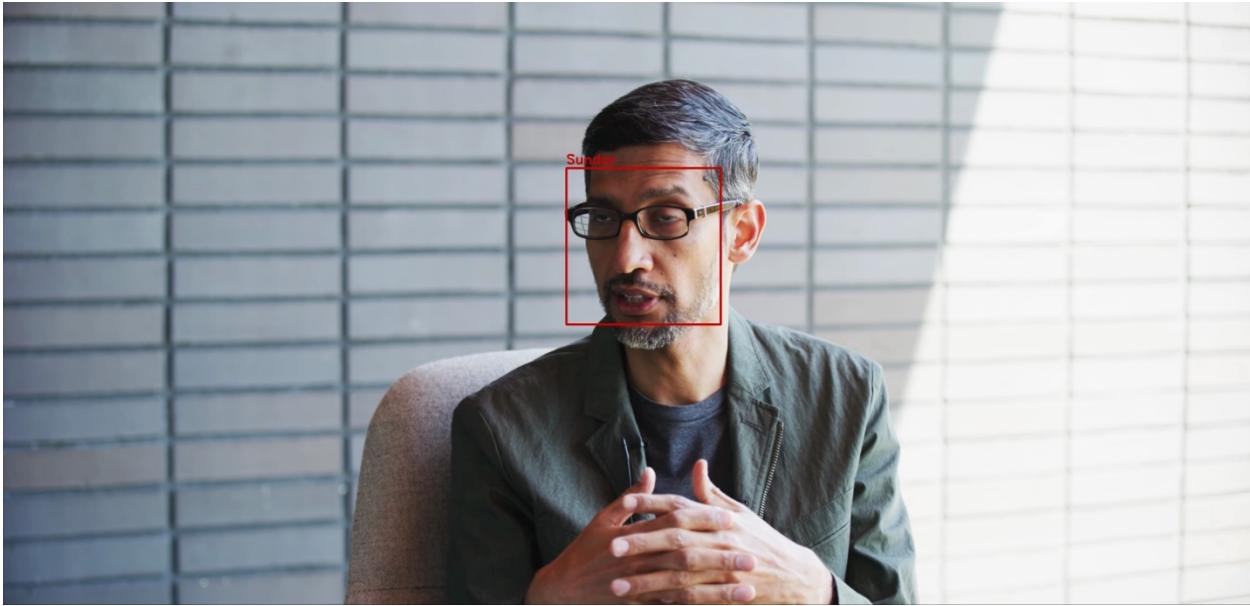
→ Detected person at

['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_iJustin.mp4',
 '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_MKBHD.mp4']

```
untprj - main.py
main.py > coco.names > simple_facerec.py > vehicle_detector.py
Project Team
1 #importing modules
2 import glob
3 import os
4 import time
5 import cv2
6 import face_recognition
7 from simple_facerec import SimpleFacerec
8 from vehicle_detector import VehicleDetector
9
10 #finding objects (car, bus, truck)
11 class FindObjects:
12     def __init__(self, keyword, path_to_videos):
13         self.keyword = keyword
14         self.path = path_to_videos
FindObjects > find_person() > while cap.isOpened() > for face_loc, name in zip(face... > if "Sundar" in name
Run: main
Encoding images loaded
Detected person at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_iJustin.mp4', '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_MKBHD.mp4']
examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Swapna.mp4
8 encoding images found.
Encoding images loaded
Detected person at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_iJustin.mp4', '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_MKBHD.mp4']
examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus1.mp4
8 encoding images found.
Encoding images loaded
Detected person at [/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_iJustin.mp4, '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Sundar_MKBHD.mp4']
Process finished with exit code 0
Version Control Python Packages TODO Python Console Problems Terminal Services
Enable Auto Reset Color Scheme? Do you know that you can automatically reset the bundled themes' color schemes? That way you ca... (7/18/22, 9:38 PM) 37:1 LF UTF-8 4 spaces Python 3.10 (untprj) Run
untprj Atom One Dark (Material)
```

Detected Images:





Object Detection Flow:

1. Input: car

→ The user is trying to detect cars

The screenshot shows the PyCharm IDE interface with the main.py file open. The code is as follows:

```
unprj > main.py
Project ~/PycharmProjects/unprj
main.py x coco.names x simple_facerec.py x vehicle_detector.py x
def run(self):
    for files in os.listdir(self.path):
        path_to_video = os.path.join(self.path, files)
        if "DS_Store" in path_to_video:
            continue
        print("examining for " + path_to_video)
        if self.keyword == "Elon":
            self.find_person(path_to_video)
        else:
            self.find_objects(self.keyword, path_to_video)
            print("Detected " + self.keyword + " at " + str(self.meta))
dir_path = "/Users/sitachallapalli/PycharmProjects/unprj/venv/Data/"
keyword = input("Enter the object you want to find:")
#s=keyword
FindObjects(str(keyword), dir_path).run()
```

The terminal window below shows the command run and the keyword entered:

```
Run: main
/Users/sitachallapalli/PycharmProjects/unprj/venv/bin/python /Users/sitachallapalli/PycharmProjects/unprj/main.py
Enter the object you want to find:car
examining for /Users/sitachallapalli/PycharmProjects/unprj/venv/Data/Bus.mp4
53.50877192982456
1658264037.520173
bus
['bus', 'bus']
['bus', 'bus']
3658264039.530232
```

Output:

The image car is detected in 3 videos: Car.mp4, Traffic.mp4 and Aeroplanes.mp4

The screenshot shows the PyCharm IDE interface. The project structure on the left includes files like main.py, coco.names, simple_facerec.py, and vehicle_detector.py. The main.py file contains Python code for object detection. The terminal window at the bottom shows the execution of the script and its output, which lists three video files where a car was detected: Car.mp4, Traffic.mp4, and Aeroplanes.mp4.

```
untprj  main.py
Project  P...  main.py  coco.names  simple_facerec.py  vehicle_detector.py
untprj ~PycharmProjects/
  venv    78
  bin    79
  Data
    Aeroplanes.mp4  81
    Bus.mp4  83
    Car.mp4  84
    Elon.mp4  86
    Swapna1.mp4  87
    Traffic.mp4  88
    train.mp4  90
  Detected_Image
    OSumanth.mp4.jpg  92
    OSwapna.mp4.jpg  94
    OSwapna1.mp4.jpg  95
    102Elon.mp4.jpg
    Aeroplanes.mp4.jc
    Car.mp4.jpg
Run:  main
      /'Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Aeroplanes.mp4'
      examining for /'Users/sitachallapalli/PycharmProjects/untprj/venv/Data/train.mp4
      60.0
      1658264062.16395
      1658264064.168003
      1658264066.1716568
      Detected car at ['/'Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Car.mp4', '/'Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Traffic.mp4',
      '/'Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Aeroplanes.mp4']

Process finished with exit code 0
```

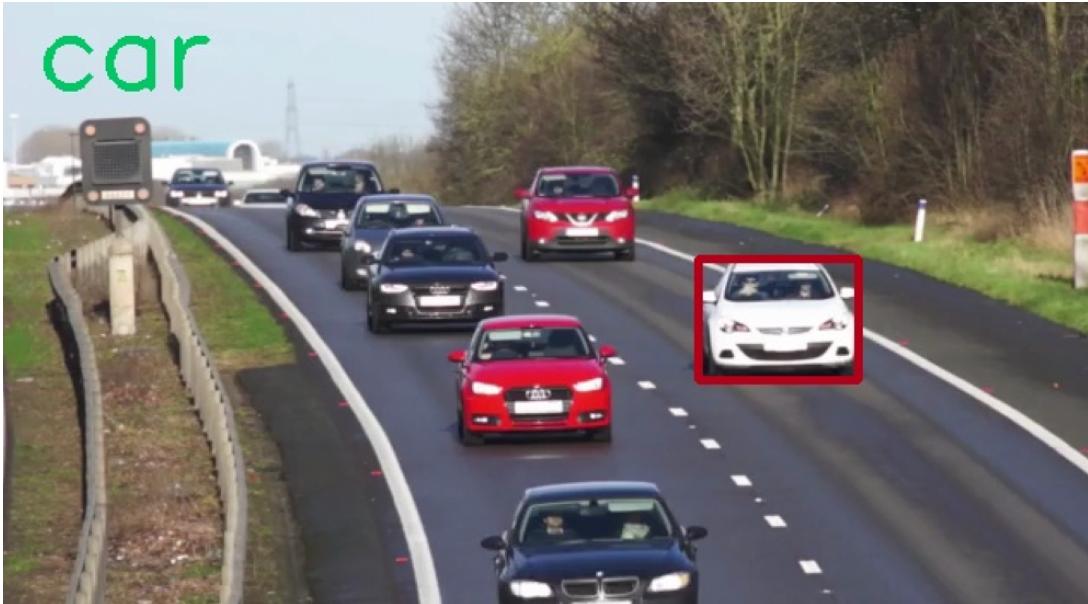
Aeroplane.mp4:



Traffic.mp4



Car.mp4:



2. Input: bus

→ The user is trying to detect bus

The screenshot shows a PyCharm project named 'untprj' with a file 'main.py' open. The code is a script for detecting objects in videos. A yellow box highlights the command 'Run: main' and the terminal output. The terminal shows the user entering 'Enter the object you want to find:bus'. The script then prints 'examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4', followed by numerical values (53.50877192982456, 1658289753.691467), and the word 'bus'. It then lists detected objects: 'bus' and '['bus', 'bus']'. Finally, it prints 'Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4']' and 'examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Car.mp4'.

```
    return
#uncomment_to_see_the_detected_images
# cv2.imshow(object_name, image)
# if cv2.waitKey(1) == 13: # is the Enter Key
#     break

78 def run(self):
79     for files in os.listdir(self.path):
80         path_to_video = os.path.join(self.path, files)
81         if "DS_Store" in path_to_video:
82             continue
83         print("examining for " + path_to_video)
84         if self.keyword == "Swapna":
85             self.find_person(path_to_video)
86         else:
87             self.find_objects(self.keyword, path_to_video)
88         print("Detected " + self.keyword + " at " + str(self.meta))
89
90     dir_path = "/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/"
91     keyword = input("Enter the object you want to find:")
92     #keyword
93
94     FindObjects = run() for files in os.listdir(self.pa... : if self.keyword == "Swapna"
Run: main
Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4']
examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus1.mp4
68.0
1658289788.089797
car
['car', 'bus', 'truck']
Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4', '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus1.mp4']
```

Output:

The screenshot shows a PyCharm project named 'untprj' with a file 'main.py' open. The code is identical to the one in the previous screenshot. A yellow box highlights the command 'Run: main' and the terminal output. The terminal shows the user entering 'Enter the object you want to find:bus'. The script then prints 'examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4', followed by numerical values (68.0, 1658289788.089797), and the word 'bus'. It then lists detected objects: 'bus' and '['bus', 'bus']'. Finally, it prints 'Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4']' and 'examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Car.mp4'.

```
    return
#uncomment_to_see_the_detected_images
# cv2.imshow(object_name, image)
# if cv2.waitKey(1) == 13: # is the Enter Key
#     break

78 def run(self):
79     for files in os.listdir(self.path):
80         path_to_video = os.path.join(self.path, files)
81         if "DS_Store" in path_to_video:
82             continue
83         print("examining for " + path_to_video)
84         if self.keyword == "Swapna":
85             self.find_person(path_to_video)
86         else:
87             self.find_objects(self.keyword, path_to_video)
88         print("Detected " + self.keyword + " at " + str(self.meta))
89
90     dir_path = "/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/"
91     keyword = input("Enter the object you want to find:")
92     #keyword
93
94     FindObjects = run() for files in os.listdir(self.pa... : if self.keyword == "Swapna"
Run: main
Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4']
examining for /Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus1.mp4
68.0
1658289788.089797
car
['car', 'bus', 'truck']
Detected bus at ['/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus.mp4', '/Users/sitachallapalli/PycharmProjects/untprj/venv/Data/Bus1.mp4']
```

Detected Bus Images:

1. bus.mp4



2. bus1.mp4

