

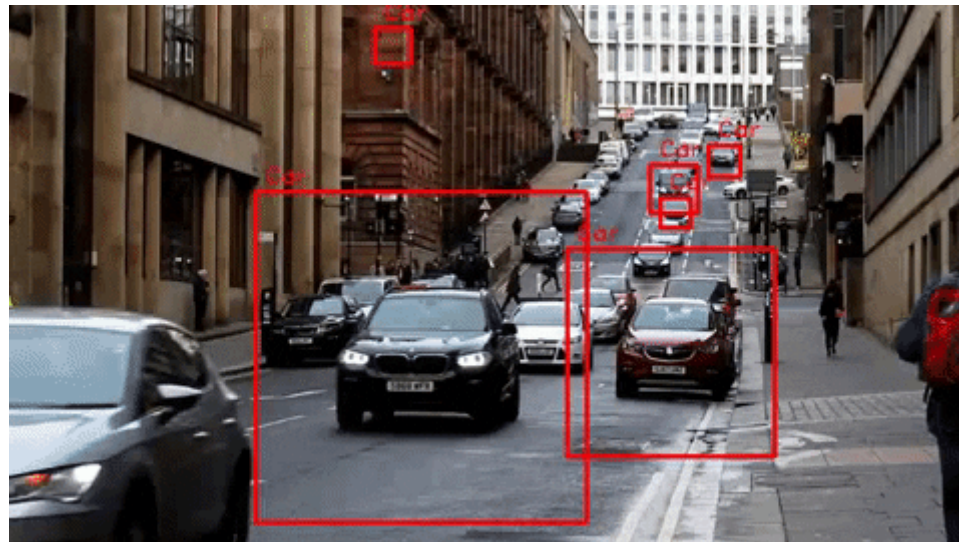
Vehicle (car) Detection in Real-Time and Recorded Videos in Python — Windows and macOS



Venkatesh Chandra

Follow

Jan 2 · 5 min read



[Video source — Linked here](#)

Vehicle detection is one of the widely used features by companies and organizations these days. This technology uses computer vision to detect different types of vehicles in a video or real-time via a camera. It finds its applications in traffic control, car tracking, creating parking sensors and many more.

In this project, we will learn how to build a car tracking system in python for both recorded and live cam streamed videos.

Before we begin, we need to install the OpenCV (Open Source Computer Vision) library which is built to help developers carry out tasks related to computer vision. We need to do a pip install for the OpenCV library.

```
install opencv-python
```

Let us now build the system in Python

We have the required libraries installed. The way the technology works is that we train the model on various image parameters of the object to be detected (car in this case), which is used to identify the object in our target.

Think of it as the train and test datasets of any machine learning model.

In this case:

Train dataset: .xml files which capture the image details of the target object

Test dataset: Live stream video/ Recorded video

The link to the full code can be found at the end of this article. I'll explain the code in steps and blocks to help you understand how it works:

Step 1: Open Spyder

Step 2: Import the library

```
import cv2
```

Step 3: Reference the input to your webcam or to the video file saved on your hard drive (mp4 format)

```
Webcam: cap = cv2.VideoCapture(0)  
Video: cap = cv2.VideoCapture(<enter file path.mp4>)
```

Step 4: We will use a pre-trained .xml file which has data on cars built using individual images. You can download the file [here](#)

```
car_cascade = cv2.CascadeClassifier(<enter file path>/cars.xml')
```

Step 5: The video is divided into frames and the code reads one frame at a time. In each frame, we detect the location of the car in the frame using the APIs which we have imported above. For each car detected, we locate the coordinates and draw a rectangle around it and release the video to the viewer.

The full code is shown below — explanations follow below the code

```
while True:
    # reads frames from a video
    ret, frames = cap.read()
    # convert to gray scale of each frames
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)
    # Detects cars of different sizes in the input image
    cars = car_cascade.detectMultiScale( gray, 1.1, 1)
    # To draw a rectangle in each cars
    for (x,y,w,h) in cars:
        cv2.rectangle(frames, (x,y), (x+w,y+h), (0,0,255), 2)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frames, 'Car', (x + 6, y - 6), font, 0.5, (0, 0, 255),
1)
    # Display frames in a window
    cv2.imshow('Car Detection', frames)
    # Wait for Enter key to stop
    if cv2.waitKey(33) == 13:
        break
```

Block 1:

```
# reads frames from a video
ret, frames = cap.read()
# convert to gray scale of each frames
gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)
```

The video is read in individual frames. Next, the frame is converted to grayscale which helps in detecting the car quickly. The reason why the

image is converted to grayscale is that the trained dataset is built in grayscale to reduce the size of the file.

Block 2:

```
# Detects cars of different sizes in the input image
cars = car_cascade.detectMultiScale( gray, 1.1, 1)

# To draw a rectangle in each cars
for (x,y,w,h) in cars:
    cv2.rectangle(frames, (x,y), (x+w,y+h), (0,0,255), 2)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frames, 'Car', (x + 6, y - 6), font, 0.5, (0, 0, 255),
1)
```

The first section of the code detects the cars in the frame and stores their coordinates (x, y axes, and the width and height of the car). The second section draws a rectangle around the area where the car is detected and displays the text 'Car' above the rectangle. You can change the font of the text and the code (0, 0, 255) is the color code of the rectangle and the text in B-G-R sequence.

Block 3:

```
# Display frames in a window
cv2.imshow('Car Detection', frames)
# Wait for Enter key to stop
if cv2.waitKey(33) == 13:
    break
```

The resulting image (frame) is released to the viewer and the loop continues to run until the user hits the Enter key on the keyboard.

Step 6: All captured videos must be released.

```
cap.release()
cv2.destroyAllWindows()
```

Run the program in the command line

The next step is to save the file in .py format and run it in command line/Anaconda prompt.

I ran it in Anaconda prompt by first navigating to the folder using the command cd.

Use Cases

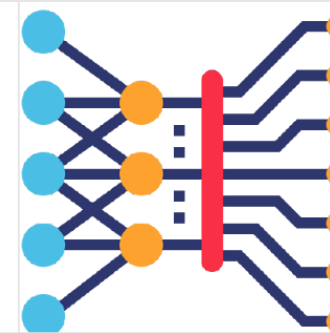
- Sensors at parking stations can identify defaulters
- Detecting the speed of cars
- Identify defaulters who speed up the car while pedestrians cross the road

Codes

chandravenky/Computer-Vision---Object-Detection-in-Python

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

github.com



Related links


[Pedestrian Detection in Python using OpenCV — Windows and macOS](#)

[Real-Time Face Detection System in Python — Windows and macOS](#)

Face Detection on recorded videos in Python — Windows and macOS

Saving output of object recognition in macOS

Where to find me

1. Connect with me on [LinkedIn](#)/ [GitHub](#)/ [My website](#)
2. Feeling generous? Buy me a [coffee here](#) 

[Python](#)[Computer Vision](#)[Object Detection](#)[Artificial Intelligence](#)[Machine Learning](#)

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. [Watch](#)

Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. [Explore](#)

Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. [Upgrade](#)

Medium[About](#)[Help](#)[Legal](#)

I ran it in Anaconda prompt by first navigating to the folder using the command `cd`.

```
cd <folder path>
```

Run the python file

```
python filename.py
```

You will see a pop-up window with the video playing. The video might be slow and it is because the number of frames is usually large in OpenCV. However, if you save the video on your hard drive, the written video is not slow and matches the fps (frames per second) of the input video.

Vehicle (car) Detection in Python



You can download more royalty-free videos [here](#).

Eureka! You have successfully built a setup which detects cars in real-time and in video recordings.

Play around with videos with vintage or concept cars and share your story in the comments section below.

Facing issues? Post your query.