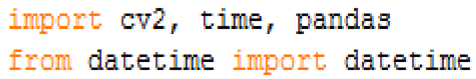
WebCam Motion Detector in Python

This python program will allow you to detect motion and also store the time interval of the motion.

Install Requirments : Install Python3, install Pandas and OpenCV libraries.

In this code we are going to record Videos that can be treated as stack of pictures called frames. Here I am comparing different frames(pictures) to the first frame which should be static(No movements initially). We compare two images by comparing the intensity value of each pixels

import required libraries

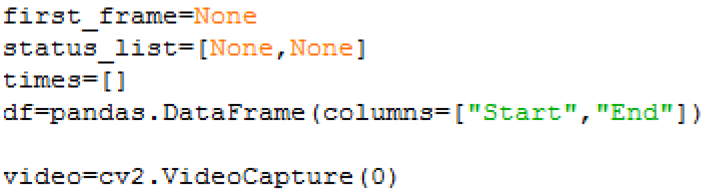


```
import cv2, time, pandas
from datetime import datetime
```

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,built on top of the Python programming language.

time module available in Python which provides functions for working with times, and for converting between representations.

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

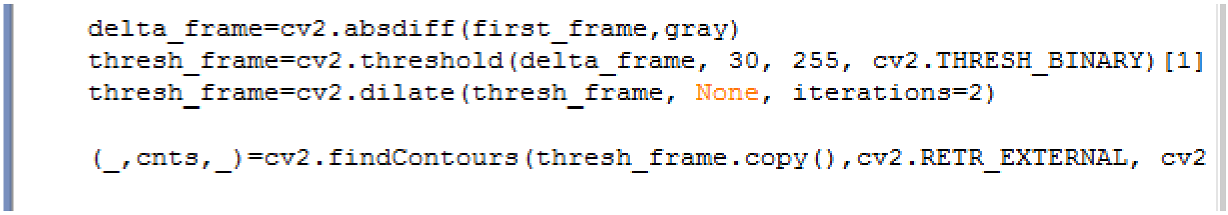


```
first_frame=None
status_list=[None,None]
times=[]
df=pandas.DataFrame(columns=["Start","End"])

video=cv2.VideoCapture(0)
```

Here we initialize the variables and dataframe to capture the vedio with stat and end of the time

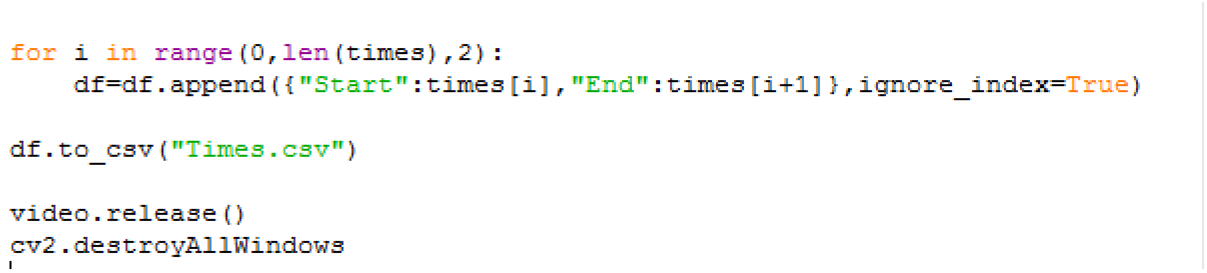
And we further use the while loop to read the image which coverts the colour image to grey scale image



```
    delta_frame=cv2.absdiff(first_frame,gray)
    thresh_frame=cv2.threshold(delta_frame, 30, 255, cv2.THRESH_BINARY)[1]
    thresh_frame=cv2.dilate(thresh_frame, None, iterations=2)

    (_,cnts,_)=cv2.findContours(thresh_frame.copy(),cv2.RETR_EXTERNAL, cv2
```

The above changes the image to white if static background and the current frame is greater than 25,further the code finds the contour and hierarchy of the object which is also appends the current state of motion ,start time of motion and end time of motion

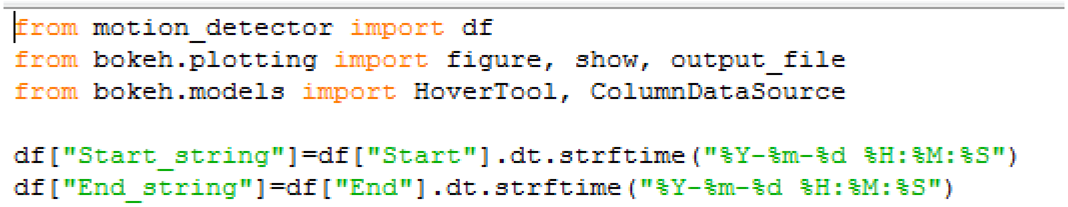


```
for i in range(0,len(times),2):
    df=df.append({"Start":times[i],"End":times[i+1]},ignore_index=True)

df.to_csv("Times.csv")

video.release()
cv2.destroyAllWindows
```

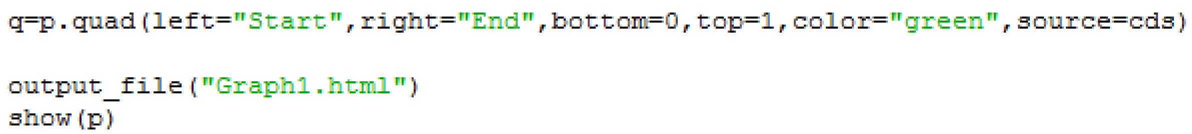
Now this code appends the motion time and user gets the option to stop recording of the object by pressing q thus ,there all the windows will be closed but we can have tracking time in newly created csv file.



```
from motion_detector import df
from bokeh.plotting import figure, show, output_file
from bokeh.models import HoverTool, ColumnDataSource

df["Start_string"]=df["Start"].dt.strftime("%Y-%m-%d %H:%M:%S")
df["End_string"]=df["End"].dt.strftime("%Y-%m-%d %H:%M:%S")
```

We create the csv file and import the required libraries such as bokeh,and dataframe is defined with respective start time and end time



```
q=p.quad(left="Start",right="End",bottom=0,top=1,color="green",source=cds)

output_file("Graph1.html")
show(p)
```

We can also plot a graph for more understanding the timings when the object is moved and so on.

We will get four images

Gray Frame : In Gray frame the image is a bit blur and in grayscale we did so because, In gray pictures there is only one intensity value whereas in RGB(Red, Green and Blue) image thre are three intensity values. So it would be easy to calculate the intensity difference in grayscale.

Difference Frame : Difference frame shows the difference of intensities of first frame to the current frame.

Threshold Frame : If the intensity difference for a particular pixel is more than 30(in my case) then that pixel will be white and if the difference is less than 30 that pixel will be black.

Color Frame : In this frame you can see the color images in color frame along with green contour around the moving objects