Motion Sensor Using Python

**Introduction**

It is a GUI based Application which is able to detect motion in the video capturing through a webcam. As soon as it detects motion it starts recording a video and save it to the hard drive with the folder name as current date and file name as current time and it also send an email alert to the user on his email id whenever someone is detected in the camera.

**Requirements**

**Software Requirements**

1. Python 3 or above
2. Open CV module
3. OS module
4. Time module
5. Numpy Module
6. Smtplib Module

**Hardware Requirements**

1. A Laptop or a PC to execute the Application
2. A Camera( External Webcam or Integrated Webcam of Laptop)

**Language or Technology Used in Application**

* **Python**

It is created by Guido Van Rossum in 1991. It is a general purpose language. Python’s syntax is easier than any other programming language and the readability of the code is also better than any other language. Since Python is an open source language, it is completely free and distributable. It has several libraries which can be easily used by any developer without any cost.

* **Open CV**

Open CV stands for Open Source Computer Vision. It is a computer vision and machine learning library. It contains more than 2500 algorithms which are used computer vision and machine learning programs. These algorithms can be used to detect and recognise faces, identify objects, classify human actions in videos, track camera movements, track moving objects and many more.

**How it works**

* **Capturing Frames**

The first step in the program execution is to capture frames to determine whether any motion is occurring or not. For this purpose we have to capture consecutive images through the camera. To capture and manipulate images we have a module known as open source computer vision(open-cv) .So first we have to import this module as

**Import cv2 #** Imports open-cv library

In this module, we have a class known as VideoCapture . By creating an object of this class we can easily capture images though the webcam. We can create the object of the class as

**cv2.VideoCapture(0)#**Returns object of the VideoCapture class

Now if we want to check whether the camera is open or not we can use a method

**Cap.isOpened() #** Returns True or False

Now to read a frame there is a method called read() and can be used as

**Cap.read()** #Returns a tuple (Boolean value , a frame)

* **Detecting Motion in the Captured Frames**

To detect motion continuously we have to capture the frame again and again for that purpose we have to put the read() method in a infinite loop and store them in two separate variables to calculate their difference. We calculate their difference as

**cv2.absdiff(frame1, frame2)#** Returns absolute difference between frame as an multidimensional array

but there is a problem whether the motion is occurred or not it will never return a zero or an array of zeros. It will always return an array of non zero elements so to determine the motion we will simply take its average using numpy module . It is a module which has some methods to works with arrays . It has a method called average() and can be used as

**numpy.average(Difference)**# Returns a floating number

Now we have compared the returned floating number with 4.0(depends upon the environment in which the camera is capturing). If the returned value is greater than 4 than the motion is detected.

* **Sending an Alert to the User**

After the motion is detected we have to alert the user that the camera has detected any activity for that we have used email alert facility in our application. To send email we have to import a modules smtplib as

**Import smtplib #** Imports the smtp library

We have used google email service to send email to the user for that we have specified the port of the google gmail service as

**Service=Smtplib.SMTP(‘smtp.gmail.com’,587(portnumber))** #Returns the object of SMTP class

Now login as

**Service.login(‘Sender’s Email\_Id’,’Sender’s Password’)**

We have a method called sendmail to send email to the user

**Service.sendmail(‘Sender’s Email’, ’Receiever’s Email’, ’Message’)**

* **Creating a Directory To Store Video Files**

First we have to create a folder named as the current date for this we have to use the OS and the Time module as

**Folder\_name=path\_of\_the\_directory+str(time.ctime(time.time()))[:10]**

We have the name of the folder but we have to check whether the folder’s name directory is already exist or not and we can check it as

**os.path.exists(folder\_name)#** returns True or False

If a folder is already exists with the same name we just have to return the working directory path concatenated with the folder name or if the folder does not exists then we can create it as

**Os.mkdir(folder\_name)#** creates a folder at the location

* **Recording a Video File**

To record a video we have to create an object of VideoWriter class as

**VW=cv2.VideoWriter(FilenameWithExtension,Codec,Framerate,Resolution)**

We can write a frame to a given location by using a method called write as

**VW.write(frame) #** Used to Write a frame to the hard drive

**Conclusion**

We have seen CCTV Cameras in many places like Streets, Railway Stations, Shops etc. This application can also be used as one but the advantage will be as suppose this application is monitoring any shop. At the day light when the customers are continuously paying visit to the shop it will detect all the motions and record the whole day as a single video but at the night we don’t have to record the video for whole night it will take useless space on the hard drive. We will definitely not interested in recording it. This app will save so much space by ignoring the less interested part and this app will also alert the user whenever someone entered in the shop. In this type of scenarios this application can be a advantage over simple cctv cameras.

**References**

* Python Documentation
* Open-CV Documentation
* OS Module Documentation
* Smtplib Documentation