



3/18/2021

# Odin's Eye

Interim Report

Andrew Heaney  
C17374656

## Contents

Introduction .....	2
Hardware .....	2
Software.....	3
Objectives.....	3
E/C/S/E (Ethical/ Commercial/ /Safety/Environmental) .....	4
Literature Review (research) .....	4
Work Plan.....	6
References .....	7

## Introduction

Odin's Eye is an open source camera service that provides added security to a home. This idea was formulated from experience with other similar products like Google's nest camera.

A nest camera is a camera system that streams a video, while recording video 24/7. It also provide notifications for when movement is detected, while they provide a good service, you have to pay to get the full benefit of the camera service.

For this project, Python will be used for the camera recording portion of the project. Ideally, while record and play the videos on the one device is the main goal, AWS ec2/s3 services may need to be used to ease the strain on the Raspberry Pi. When the device is recording the video, OpenCV will be used in order to cut un-necessary clips from the video which don't show much new details. After the recording is complete, the video will be either saved to the devices and streamed to a website hosted on the Raspberry Pi or uploaded to the S3 cloud storage so it can be displayed on the EC2 website.

While some of the details for this project have been finalised, the project is still in development and may change. Some changes may include the use of AWS S3/EC2 instances in this project. Using these Server/Cloud spaces would ease the strain on the raspberry pi and diversify the project. Using the EC2 instances would also more easily allow this project to be used in an online capacity.

## Hardware

This project will mainly be utilising a raspberry pi which runs a 64 bit ubuntu system, raspberry pi high quality camera and a terabyte external storage device.

This may change if an amazon storage solution is utilised. As discussed in the introduction, if the plan to host the website on the Raspberry Pi doesn't work, the videos will be stored on an AWS S3 Bucket and display on the AWS EC2 instance.

An S3 bucket works like a Google drive or Dropbox account, it stores information on it and can store larger amounts at a small cost. S3 is also able to host static websites,

but this means that PHP and Javascript would not be able to be run on the website which may present a problem.

To remedy this, I have thought of using an AWS EC2 instance, which is a server which can run dynamic website. Hosting it on this would work out more expensive though since the more spec'd the server is, the more expensive the cost to rent is.

This is why a mixture of both would work the best, since the S3 can hold larger amounts of data, and using get commands, I could view the videos on a dynamic website, hosted on a EC2 instance.

## Software

This project will be utilising an opensource python module called OpenCV which has multiple applications such as facial recognition but this project will utilise frame analysis so video size may be kept to a minimal. This software will also allow recordings in a mp4 format which is the most commonly used on a html webpage, the reason a website is utilised is so that the video tags can be used to display video, so that a user could remotely check the videos.

For the website standard html/css language, but new languages called Ionic and React may be used, but a basic project must be made first. Providing the success of the basic model, these new languages may be tested to add more functionality and diversity. The website would be used for displaying the videos. This will be don't using the video tags that come in standard html, they are capable of show mp4, ogg and webm file formats.

## Objectives

The first objective is to create a programme that can record video for each day for a week. When the video is recorded, it can then be labelled with the date of the recording. Then create a website which can be used to view the recordings remotely.

Once the website is completed, the basic idea for project is completed, however there are extra features that could be added.

- The first being a way to reduce the size of the files in order to save space. This will be done using an opensource module called OpenCV. This will allow the video to be edited as it is recorded, so if nothing is happening within a specified time, it won't save that video.
- This will lead to another feature that could be added. If the previous objective is achieved, and Open CV can look at frames to determine whether the video should be saved, then add an email notification service when the camera notices a change in the frame.

The way the programme will be refreshed daily is by, instead of looping the programme constantly, it will be restarted using Crontab. Crontab is a scheduling service which allows the user to set when the user wants to run a programme.

After all that is done, a login function for different users will be added, and these users would have different levels of access. So, the lowest level could only view the saved videos. Another could change settings such as email addresses for notifications. This is where a database such as MariaDB could be implemented for storing user information.

### E/C/S/E (Ethical/ Commercial/ /Safety/Environmental)

While there are no safety concerns in the project, as it is mostly a software-based project, this is reflected in the completed project safety statement.

There is an ethical issue with this project in particular. Since this project involves the use of cameras, privacy can become an issue. For the duration of this project though, this project will be worked on in the same place and video captured will be only of the same background. For the development part of the project, this is a very low risk concern. However, if this project becomes a commercial product, the privacy of others may be affected by the consumer that buys this product. This is a problem with most camera services including the likes of Google nest, and while laws are in place to protect people from such problems, the onus lies with the consumer to ensure that camera usage falls within the guidelines set out by the proper authorities.

### Literature Review (research)

#### **General Webpage Design**

For webpage help, resources and example code will be reviewed on sites like W3Schools, they provide free examples for Multiple languages including html, css, javascript and php which will be utilised in the development of the project.

W3Schools, <https://www.w3schools.com/>, [10/03/2021]

## **Crontab**

Crontab is a scheduling programme which is used to run a programme at a scheduled time. This can be used in two different ways, the user can schedule things by running them at a specific time and date, or the user can use crontab to run a programme after certain time intervals, e.g. every hour

D. Both, 'How to use cron in Linux', <https://opensource.com/article/17/11/how-use-cron-linux>, [28/02/2021]

While the previous article is used to set up the software, another website was used which was useful to write the right command.

Cronitor.io, 'crontab guru', <https://crontab.guru/#5>, [28/02/2021]

## **Recording with OpenCV**

This article helped with the process needed to record a video. Some important things to note is the parameters needed for the programmes to run.

A filename is needed, as discussed, a date module will be utilised, so the filename can be changed to the right date. A codec will need to be used, Since the website is being used to display the video, the file type is limited to using either an mp4 format, ogg format or webm format. Since mp4 is the most commonly used, that is the one which will be utilised. Regarding the codec due to mp4 format being used the FFMPEG codec will be used in order to record in that format.

The next argument needed is the fps. Some experimenting has been done on this and concluded that around 24 fps gave me the best framerate. It is also important that not too many frames per second are recorded, since this will take up a lot of memory. Other arguments can be changed like the frame size, but this won't be changed without more testing.

GeeksForGeeks, 'Saving a Video using OpenCV', <https://www.geeksforgeeks.org/saving-a-video-using-opencv/>, [25/02/2021]

## Work Plan

As discussed, in the objective this project has been split into different phases, and this plan presumes that everything goes right although things may change as the project proceeds.

1. The first stage of the project is to set up the raspberry pi so that it can record mp4 files that can be played on a html webpage.
2. Design a suitable website, for the time being there is no need for any login pages as that would come in a later stage.
3. Once video is successfully saved and viewed videos on a separate device, work on the compression of file size using OpenCV.  
Provided the success of this a feature which can email users to notify users of changes in the frame
4. Add increased security for the website by adding a MariaDB database for storing users and passwords, this will then lead to having admins which can change settings and spectators who can just view the videos.
5. The project will be at a satisfactory standard if success is achieved with these steps, but if there is some extra time, a stream service where users can just view what is happening live would be enhance the user experience.

## References

D. Both, 'How to use cron in Linux', <https://opensource.com/article/17/11/how-use-cron-linux>, [28/02/2021]

Cronitor.io, 'crontab guru', <https://crontab.guru/#5>, [28/02/2021]

GeeksForGeeks, 'Saving a Video using OpenCV' ,  
<https://www.geeksforgeeks.org/saving-a-video-using-opencv/>, [25/02/2021]

W3Schools, <https://www.w3schools.com/>, [10/03/2021]

Amazon, <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>,  
[05/03/2021]

Amazon,<https://aws.amazon.com/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc>,  
[05/03/2021]