**Cypher Security Alarm**

**Table of contents**

1. **Cover …....... P. 1**
2. **Specifications …....... P. 2**
3. **Features …....... P. 5**
4. **Learning Cycle …....... P. 7**
5. **Developmental cycle …....... P. 8**
6. **Physical models …....... P. 10**
7. **Documenting methods …....... P. 11**
8. **Expectations …....... P. 12**

**2- Specifications**

This section will be divided into two parts: Physical and Software specifications

Part 1:

Cypher security camera will start off by using the Nvidia Jetson Nano as the main board,

The Jetson packs a 128-core Maxwell GPU which will theoretically provide enough graphical processing power to achieve the desired detections on it’s own without needing to be connected to a central service following an edge processing technique. It also uses a quad-core arm A57 processor which slightly falls behind when comparing it to the raspberry pi 4 but since most of this camera’s tasks will be GPU intensive, the trade off is well worth it.

The first tests on this system will use the raspberry pi camera attached to the Jetson since it is compatible with the camera input slot. After enough research is done and results from these tests come are present, we might opt to using a higher resolution USB camera instead to achieve higher detection rates (if they weren’t sufficient enough)

A power supply will also be included as a part of the bundle using the power barrel input on the Jetson since the micro USB option will not run the Jetson to the max of it’s potential power.

Lastly, a custom 3d printed case will package everything together in a compact design with the aim of making the board run as quietly and as cool as possible for it to run smoothly for the longest period of time it could

Part 2:

On the software side of things, the Jetson will be running Ubuntu linux as its main operating system with a program set to run on bootup that could be accessed remotely and doesn’t require any user input to be started. The program will be written mainly in python for the high availability of machine learning related libraries and will use custom implementations of YOLO and FaceNet optimized to achieve the highest detection rate possible built to automatically train itself on a inputted database for facial recognition purposes

**3- Features**

* A live feed that could be accessed throughout the network through an authenticated link
* Live notifications sent through email/SMS and other configurable methods to anyone who’s given access to Cypher
* A comprehensive administrator page to configure all the camera’s settings
* Fast and intuitive detection
* Two way VOIP with a physical button to initiate the call
* No need for a centralized server for the Cypher to run!
* The option of live updates on the algorithm
* The ability to copy backups of the recorded videos over the network
* Protection against malicious attacks
* Live statistics of the area the camera is observing (The amount of traffic, times where area is most busy, and more)
* The ability to turn on detection for objects from a predefined list
* Time settings when the observed area should be clear to increase motion detection sensitivity

**4- Learning Cycle**

The main topics that I will need to become proficient at to complete this project are:

1. Python design patterns to organize and build scalable projects
2. Software documentation methods for documenting the whole process of development and the code
3. Become proficient at interacting with OpenCV for the manipulations needed
4. Learning the basics of Machine learning to be able to interact and customize previously built models
5. Learn how to use Flask to build the interactive sites needed for settings pages and the viewing the live feeds
6. Read research papers about “YOLO 9000” and “FaceNet” to get a deeper understanding of how the algorithms were built and work
7. Basic functions of NumPy for gathering and displaying stats
8. Database implementation concepts on python for storing gathered data

A full understating of these concepts is not required to start the project but will most likely be necessary to finish it.

**5- Developmental cycle**

The project has a development cycle of approximately 4 and a half months from the start date, some of the steps needed will be started before the developmental cycle starts mainly the research and testing phase of the project. The project will use a mix between the features of agile and waterfall models for development, it will take an iterative form where a minimalistic working version will be developed and features are added on to it while I’ll inherit the research and documentation methods used in the waterfall model. The project will also be developed in a modular way where each module is fully developed and tested before moving on to the next one with the idea in mind that all these modules will be tied together by a central module that wouldn not make it difficult to add, remove or change modules in the future. The development of this project will be split into different phases as follows:

1. The research phase – 2 weeks:

The first phase of this project will involve researching the various elements mentioned in the learning cycle section of this document with a focus on gaining enough knowledge and gathering resources to be able to refer back to them if when I need to get more in depth about any specific subject to achieve the task in hand.

1. The experimenting phase – 2 weeks:

In this phase, the main focus would be to comprehensively test and increase my knowledge on the subjects I’ve picked up in the learning phase. Things that I will most likely do in this phase is build small scale projects (2 to 4 projects) that would be focused on strengthening my knowledge on these concepts.

1. The model development phase – 4 weeks:

In order for this project to be successful, I will need to develop working machine learning models for the detection rates that I desire to achieve. In this phase I will be focusing on just that keeping track of important variables such as the speed of the algorithm, the percentage of detections it’s able to make and the percentage of false positives that the model outputs.

1. The User interface phase – 3 weeks:

In this phase, I will convert the developed model to be user friendly visually in addition to developing the authenticated site that the user can access to view the live camera feed or tweak specific settings. The implementation of gathering, storing and displaying statistics will also be done in this stage.

1. The testing phase – 1 week:

In this phase, the testing of the project as a whole will be done by comprehensively testing it myself in addition to having other developers and users test it taking notes and committing changes to any module that needs to be improved.

1. The networking phase – 2 weeks:

I dedicated a sperate phase for networking mainly because I think this will be one of the subjects that will be outside my direct expertise.

1. The physical/polishing phase – 2 weeks:

The final phase of the project of the project will be spent on gathering the project in a comprehensive physical package while polishing the software making sure it runs on boot without direct user interaction and that the system will be ready for future remote updates.

**6- Physical model**

The cypher’s first physical model will be held in a custom 3d printed case that holds the Jetson nano and the camera with only the necessary I/O exposed (Ethernet port and barrel power jack)

Possibly, the I/O could be routed to be on the side the device is going to be mounted on to hide all the wires for much neater, cleaner look on the wall.

Although this is an important part of delivering a full product, it won’t be a priority due to the short period of the time the product has to be finished in. A more focused approach towards this step could be revised later on in a second iteration over the cypher camera.

1. **Documenting methods**

Since the software written for this product going to be done in a modular way, I will focus on documenting every module when it’s finished to get a more comprehensive documentation of how everything functions and what role every piece plays. Another version of the documentation is going to be written aimed more towards users explaining functions of the cypher camera, this version will use the more developer aimed documentation as the guide of it’s content with a more simplistic user-friendly way of writing explaining only what the user needs to know to use the product. For source control, this project will use GitHub and upon completion all the source code will be available publicly to make the project open source for others who might be interested in similar concepts to look at and learn from.

1. **Expectations**

From my point of view, I think it would be unrealistic for me to expect all the features I’m implementing to fully work with no issues in the short time period I have to get this project completed. A more realistic expectation is to have at least %70 of the features working well with documentation of what could be done to get the product to a market ready stage. With all that said, I expect myself to spend about 8-10 hours a week working on this project during the spring of 2020 to ensure I’ve optimized my success rate getting this project to a state I’m happy with.