ENSC 405W Grading Rubric for Design Specification

Criteria	Details	Marks
Introduction/Background	Introduces basic purpose of the project.	**/05%**
Content	Document explains the design specifications with appropriate justification for the design approach chosen. Includes descriptions of the physics (or chemistry, biology, geology, meteorology, etc.) underlying the choices.	**/20%**
Technical Correctness	Ideas presented represent design specifications that are expected to be used for construction. Specifications are presented using tables, graphs, and figures where possible (rather than over-reliance upon text). Equations and graphs are used to back up/illustrate the science/engineering underlying the design.	**/25%**
Process Details	Specification distinguishes between design details for present project version (i.e., proof-of-concept and appearance prototypes) and, if details can be known at this point, later stages of project (i.e. engineering prototype and possibly production prototype). While you should end up with a complete plan for your proof-of-concept and appearance prototypes, depending upon the nature of your project there might be uncertainty at this point about the later stages of the project in which case you can discuss those uncertainties and how the earlier prototypes can resolve them. Numbering of design specs matches up with numbering for requirements specs (as necessary and if possible).	**/15%**
Supporting Test Plans Appendix	Provides supporting test plans to address testing of the design details of subsystems and components. (See, for example, supporting test plans in https://en.m.wikipedia.org/wiki/Test_plan) A short Acceptance Test Plan should have been provided in your requirements specification document.	**/10%**
User Interface and Appearance Appendix	Summarizes appearance and requirements for the User Interface (based upon the lectures and the concepts outlined in the Donald Norman textbook – see "Useful links" in Pages on Canvas).	**Graded Separately**
Conclusion/References	Summarizes design. Includes references for information sources.	**/05%**
Presentation/Organization	Document looks like a professional specification. Ideas follow logically.	**/05%**
Format/Correctness/Style	Includes letter of transmittal, title page, abstract, table of contents, list of figures and tables, glossary, and references. Pages are numbered, figures and tables are introduced, headings are numbered, etc. References and citations are properly formatted. Correct spelling, grammar, and punctuation. Style is clear, concise, and coherent.	**/15%**

Version of 25 February 2019 by Craig Scratchley. Updated from Version provided by Steve Whitmore.

March 13th, 2019
Andrew H. Rawicz
School of Engineering Science
Simon Fraser University
V5A 1S6

Re: ENSC 405W Design Specification for NightEagle

Dear Dr. Rawica and Professor Craig Scratchley,

We have thoroughly prepared EagleVision's Design Specifications report of our NightEagle device for review. Our unwavering goal is to create a portable, efficient, and affordable night-vision human and animal detector to alarm drivers, in order to greatly reduce vehicular fatalities during night time. We will combine bleeding-edge Machine Learning technology into an extremely efficient and portable hardware form-factor. The device will take in visual data from an embedded night vision camera and pass the data through our in-house developed AI in order to correctly identify pedestrians under any lighting conditions.

This document will detail the system overview of said device from its functionalities, system integration, software design, and hardware implementation. It also breaks down the device into its core electronics, software, electrical, and system interface elements, as well as provide detailed design specifications for each. Each specification will be indicated to be targeted for proof-of-concept, prototype, or final product, based on the outlined priorities; the appearance prototype will also be rendered in 3D for reference. Please note that the specifications marked as PoC will be strived to be demonstrable at the end of ENSC 405. Furthermore, the included appendix explores the supporting test plan and UI of the device in full, and will be located at the end of the document.

EagleVision currently consists of three engineering students: John Xing, Billy Luo, and John Zhang, all coming from different engineering divergences to ensure the robustness of the final end product. We would like to thank you for taking the time to review our product and giving valuable feedback.
Sincerely,

John Xing

CEO

Enclosed: Design Specification for EagleVision's NightEagle

EagleVision™

PoC Prototype Design Specification

for EagleVision's flagship device, the NightEagle

Project team:

John Xing

Billy Luo

John Zhang

Correspondence:

Billy Luo

Email: luosiyal@sfu.ca

Submitted to:	***Issue date:***
Mr. Craig Scratchley: ENSC 405W	Mar 14th, 2019
Dr. Andrew Rawica: ENSC 440	Version 1.0.0
School of Engineering Science, SFU	Confidential content

Abstract

Throughout the previous Requirement Specifications report, the drive and passion behind the EagleVision team was detailed through night-time vehicular pedestrian collisions. The preliminary hardware, software, and general requirements were outlined, as well as re-compiled in the form of test plans. In the Design Specifications report, all the components which will make up the core CPU processing block and the software user experience will be explored fully. The final product, which will be a high-speed night-vision human recognition device, will be broken down into the various technical in order to achieve the various specifications.

This specification document will serve as a high-level exploration of the proposed solution to pedestrian fatalities due to vehicular collisions, especially at night, for high-speed vehicles of all kinds. EagleVision's mission is to minimize or greatly reduce night-time vehicular collisions with pedestrians by implementing a high-effective yet affordable device, which will recognize pedestrians even in the darkest of circumstances and alert the driver. EagleVision will construct this compact and fully embedded device by combining a high quality night-vision sensors, a micro-PC board, and cutting-edge self-developed machine learning software which reacts in real time. Looking from a top-perspective, the system will pass in the input data through the camera sensor and be decoded by the on-board CPU of our motherboard. The decoded string will then be sent to the concurrently running AI, which will detect and identify pedestrians based on their facial features and send an external audio/visual alert to the user. EagleVision's software team will simplify the user-interface so that it can be operated easily while driving and avoid distractions to the driver.

This document will serve as an overarching guide to both the design language and technical specifications needed to complete the implementation process on each individual piece of the device. Multiple sections of the Appendix will be dedicated to illustrating the PoC Test Plan, the user-interface, and the development plan which will be followed throughout ENSC 440.

Table of Contents

List of Figures

List of Tables

Glossary

Term	Definition
Opencv Library	An open source coding library containing training datasets for software integration
Machine learning AI	Artificial Intelligence based on repeated training cases of software to recognize visual queues
Night-Vision Device	Conversion to visible light of both visible light and near-infrared using an image intensifier tube, to produce a monochrome feed
Feature descriptor	An algorithm which takes an image and outputs feature descriptors/feature vectors
RPi	Raspberry Pi board
Training Model	The base set of data to be fed into the AI in iterations to eventually produce the desired machine learning capabilities

1 Introduction

One of the core driving factors behind the conception of the NightEagle device is the unfortunate rise of vehicular collisions rates with pedestrians over the years.[2] Although there are traffic lights on almost every major street, there are still many trails and crosswalks without any pedestrian or street lights. As a result, research shows that the overall severity of injuries is tripled on roads without light.[1] Even though certain high-end vehicle brands in modern times incorporate some form of frontal collision detection technology, the R&D industry is still in its development stages.

EagleVision aims to provide a safer environment for the current traffic ecosystem. The company is committed to enabling all drivers - not only high-end luxury vehicle owners, to drive safely with its patented frontal human detection device. In order to achieve this, the team has identified several potential tools to integrate. OpenCV, a library built for developing real-time computer vision systems, provides Haar cascade classifiers. Haar cascade classifiers are based on a machine learning approach, which can be trained from both positive (image with faces) and negative (image without faces) sample spaces. The Haar classifier will take in thousands of samples and the code will attempt to search for the face by analyzing each pixel in the image.[2] Through countless iterations of incremental improvements, the AI will become fast and accurate at recognition. The code will also use Histogram of Oriented Gradients (HOG), a feature descriptor for computer vision and image processing, to count occurrences of gradient orientation in localized portions of an image and detect the human body.[3] In terms of hardware input, a night-vision capable camera will be used to record the video stream and output frame-by-frame bitmaps, so that the software component can detect the faces and bodies in each sample. As for portability and ease of usage, the system will incorporate a micro-computer such as Raspberry Pi as the computer at the heart of the device. It is a single-board computer consisting a CPU, GPU, RAM, Ethernet ports, GPIO pins, Xbee sockets, UART, power source input, along with various interfaces for connecting other external peripherals.[4] Raspberry Pi gives the development team at EagleVision a powerful set of tools found only on full-sized PCs, but in a minimal size and cost factor.

1.1 Scope

This document serves as an overview and analysis of the various technical design specifications of the NightEagle device, which EagleVision will strive for during the various development stages. The analysis will be divided into Systems Overview (integration), detailed Hardware, and detailed Software. Finally, the appendix will explore the test plan, user interface, and development plan of the final product.

1.2 Intended Audience

This report functions as the guiding principles at all design stages for EagleVision's employees, and every prototype will be evaluated according to the outlined specifications. In addition, this document is ready to be evaluated by Professor Craig Scratchley, Dr. Andrew Rawicz, the teaching assistants, as well as potential clients and investors.

1.3 Design Classification

NightEagle's design classification is shown below:

Domain-{Specification Number}Priority	Design specification description

Table 1: Design Classification Format

Where Specification Number is the order, with 1 being the highest

Domain is classified as:

- **GE** - General
- **HW** - Hardware
- **SW** - Software
- **ES** - Engineering Standard
- **SS** - Safety/Sustainability Standard

Priority is classfied as:

- **P**: Proof of Concept - Basic specification and functionality which need to prioritized
- **E**: Engineering Prototype - Completed functionality which need to improve
- **F**: Final Product - Professional features and performance improvement.

1.4 System Overview

The device uses a night vision camera as the main input device. The video feed recorded through the camera lense will produce an encoded and continuous digital signal which is then passed along through the micro-motherboard into the concurrently running code. The software element embedded into the board will then decode the image feed frame by frame into a format recognized by the code and begin analyzing it. The complex software backend will detect all the humans in the frame and draw an alert window around the detected objects. It will then re-encode the analyzed feed and transmit it through the hardware, which is connected to a mini-screen. The end result will be an extremely efficient clockwork of hardware and software components, as the screen will display the video feed as the camera sees it, with all the humans detected in real time. Additional hardware alerts can be implemented, which includes sound and visual alerts when the software picks up a specified object. After careful deliberation about the type of alert which held the best balance between clarity and minimal driving distractions, the team has decided to focus on an audio-based system. The device will alert the driver via an external or embedded speaker if it detects a pedestrian, and in later iterations the software may even be able to inform the driver the approximate distance and direction of the object relative to the device.

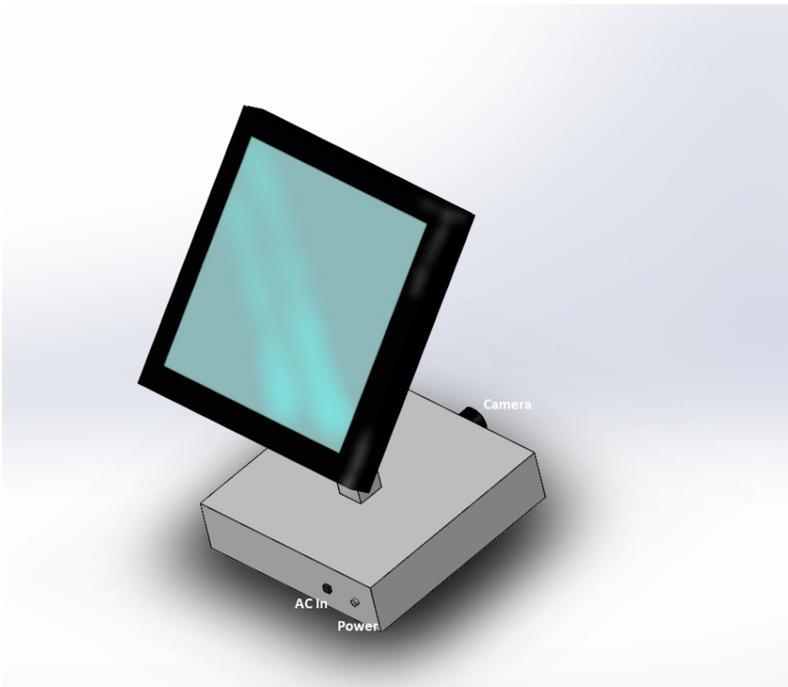

Figure 1: Appearance Prototype of NightEagle

Looking at the Appearance Prototype, the Power Input can be interpreted as the main "on/off" switch of the entire system. It will accept a 5V power input from the user's vehicle, and may offer a chargeable battery option in later iterations. After power is supplied, the main computer board should send the signal to all peripherals, including the camera, the screen, and additional hardware alert options. The micro-computer also concurrently loops the code, which is designed to output to the various hardware components. The code will decode the camera feed and perform human detection on it. The video is then re-encoded and passed onto the external screen in real-time, so the driver can see the recognized objects being highlighted. More detailed inspections of the hardware and software component will be discussed in the following section.

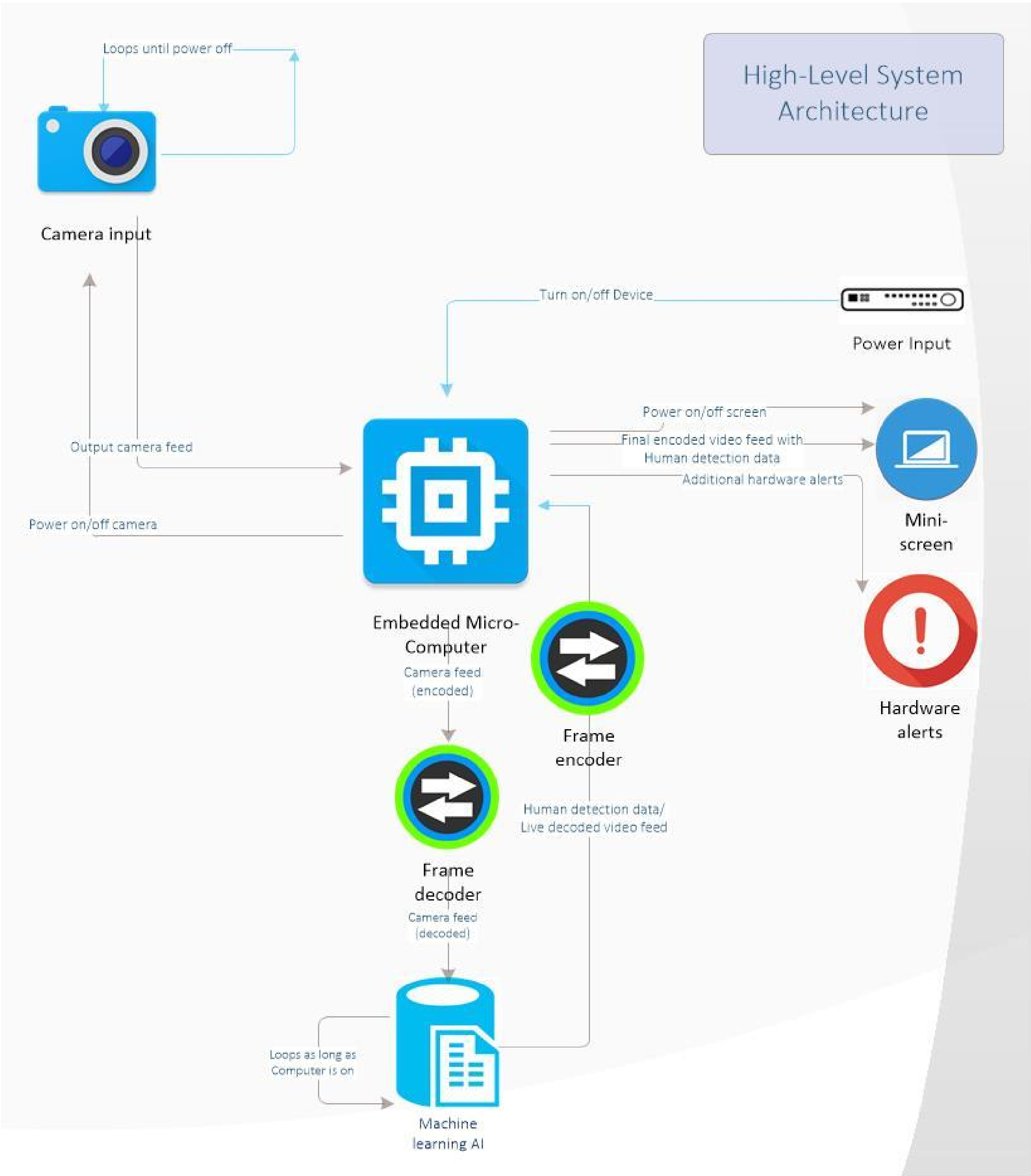

Figure 2: High-level systems architecture of entire device recreated through Visio

As discussed in the requirements specifications document, there are certain essential qualities of the overall system and device which must be met by the EagleVision team. The design specifications in this section highlight the prerequisites regarding the core functionality, the soft/hardware capabilities, and the accepted price range.

Where P = PoC, E = Engineering Prototype, F = Final Product

Design Specification	Detail	P	E	F
GE-1H	Device must be able to recognize and highlight pedestrians detected in any dark environment	x	x	x
GE-2H	Device will be able to recognize pedestrians by both facial features and body structure		x	x
GE-3H	Device will provide audio or visual (LED) feedback to alert user when pedestrian recognized		x	x
GE-4M	Device should be compatible with standard car outlet extensions and be able to receive power through plug-in		x	x
GE-5L	Device should have simple and clear physical buttons for power on/off. Later iterations may have volume button if audio alert implemented			x
GE-6L	The device should be under $150			x

Table 2: General Design Specification Requirements

2 Electronics Design

This section provides a detailed summary of the methods designed to take full advantage of the electronics components. As can be seen from Figure 2, the *final product design* will implement such an embedded system in order to compact all the various pieces into a small and portable factor. There are several types of boards that can be used to integrate software from hardware, and the team has settled on the Raspberry Pi 3, as it is amoungst the most robust and developed of the lot. It will be used as a base to explore the fundamental design specifications .

Looking at the diagram, the Power Input can be interpreted as the main “on/off” switch of the entire system. After power is supplied, the main computer board should send the signal to all peripherals, including the camera, the screen, and additional hardware alert options. The micro-computer also concurrently loops the code, which is designed to output to the various electronics components. The code will decode the camera feed and perform human detection on it. The video is then re-encoded and passed onto the external screen in real-time, so the driver can see the recognized objects being highlighted.

2.1 Main Central Computer

As mentioned above, this document will use the Raspberry Pi 3 as a basis for the hardware core of the device. Since it is a very versatile and developed board, it comes with many advantages.

Before examining the nature of each electronics component, the overall architecture is first explored to form the hardware core of the device. Taking a look at Figure 3, all the different components have been defined and allocated on the Raspberry Pi board. As seen in the previous section, Power Input will regulate the on/off status of the system, and the CPU/memory block houses the software, which will control the data fed to the other hardware components. This should create the end result of an efficient and compact system, where all the fringe components are secured close to the main board, which will prevent parts from coming loose.

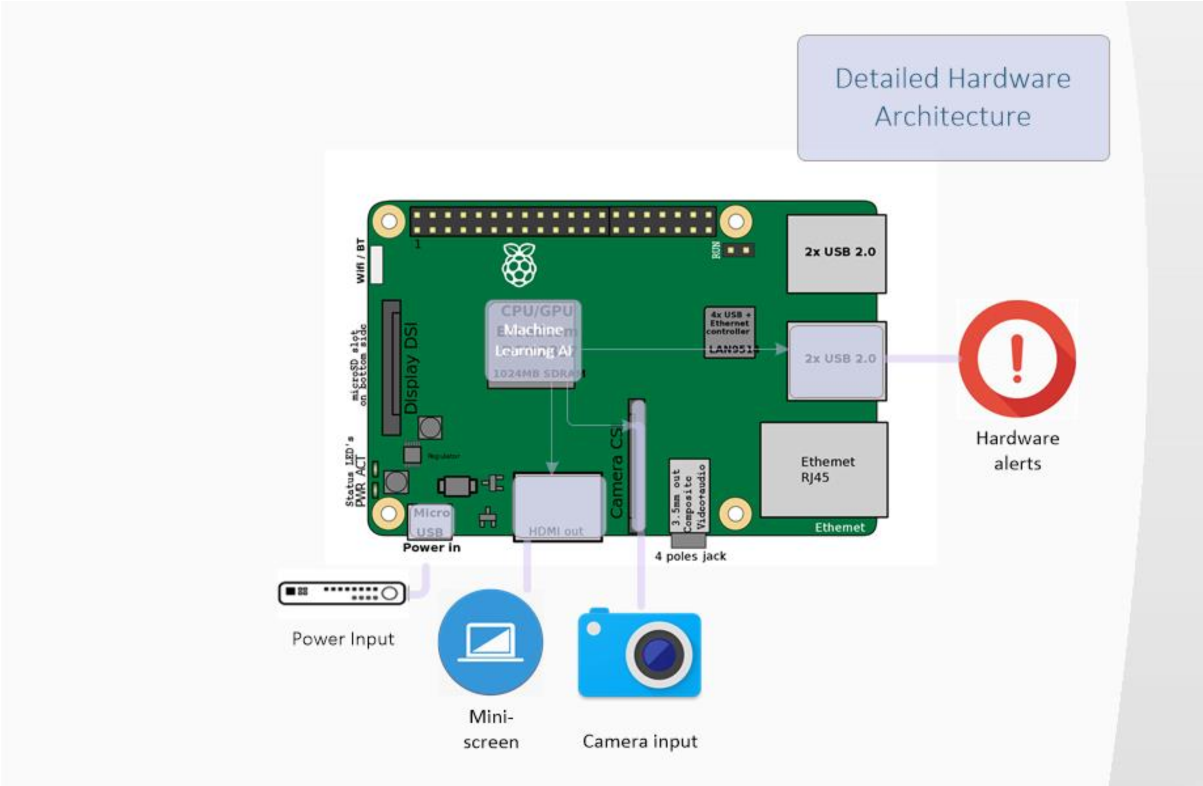

Figure 3: Hardware implementation recreated through Visio; based on Raspberry Pi 3 diagram [5]

There are several critical design specifications which would ensure the board to function. These are some of the most important specifications of the device, as the motherboard serves as the backbone to merge all other components.

Design Specification	Detail	P	E	F
HW-7H	Raspberry board must be able to accomodate machine learning AI and complete recognition within a few seconds	x	x	x
HW-8H	Board must be able to accommodate various electronics components and be compatible with the camera, screen, and audio alerts		x	x
HW-9M	Board will be able to fit within Raspberery Pi Case Model B			x
HW-10L	Board should have reasonable power consumption and no overheating issues			x

Table 3: Hardware Design Specification for Main Central Computer

2.2 Video input interface

If the heart of the device is the motherboard, its eyes is the night-vision camera. As can be seen from Figure 3, there is a specified I/O for camera modules. This means that most of the electronics components can be connected directly to the board, which would in turn be fed through to the software.

Figure 4: Common camera module I/O for Raspberry [6]

In the previous section, it was show that the camera has a predefined slot which can fit onto the board. As it is a night-vision camera, the output footage will be in gray monochrome, which will be fed into the software core through the motherboard. The following specifications highlight the critical specs of the camera for the overall integration process.

Design Specification	Detail	P	E	F
HW-11H	Night-vision camera module must be recognized by the board and communicate without delay	x	x	x
HW-12H	Camera should be in high-definition for minimum error in recognition. Output feed must be night-vision ready		x	x
HW-13M	Camera will be able to fit within Raspberery Pi Case Model B			x

Table 4: Hardware Design Specification for Camera

2.3 Video output interface

In addition, there will be an external screen which will serve to transfer the live feed processed by the software. There are several smaller 5 inch options which would work well with the current model. As the Raspberry comes with HDMI options, it will be compatible with most models. The screen will be secured to the top of the device to ensure stability.

Figure 5: Mini-screen for Device [7]

The software will highlight the pedestrians on the screen and alert the user. The screen will also serve as the User Interface menu for the user when booting up. Later iterations will allow more interactions and control with the screen.

Design Specification	Detail	P	E	F
HW-14H	Screen module must be recognized by the board/software and communicate without delay	x	x	x
HW-15H	Screen should be in high-definition for minimum error in video output		x	x
HW-16H	Screen should have touchscreen options for user input		x	x

HW-17M	Screen will be able to fit and secured on top of Raspberery Pi Case Model B			x
HW-18M	Screen has audio output DAC for alerts option			x

Table 5: Hardware Design Specification for Video Output Interface

2.3 Hardware Alerts Interface

As discussed earlier, one of the integral functionalities of the device is to alert the user once the pedestrian has been correctly identified. As with the other electronics components, there are several options available which are compatible with the existing Raspberry infrastructure. While the exact implementation of alerts will be explored in a later section, the electronics behind such options will be explained here.

If audio alerts are chosen, a mini speaker can be embedded into the casing such that the software will output speech to approximate the distance and direction of the pedestrian. The team will either connect it directly with a DAC adapter to the board, or through the external screen's audio out port.

Figure 6: Mini-speaker system for sound alerts [8]

If hardware lights are chosen, the software will command the LEDs to glow when a pedestrian is recognized. The light should be a soft and non-distracting colour and alert the user to monitor the road condition, or examine the screen if the pedestrian cannot be spotted in the dark.

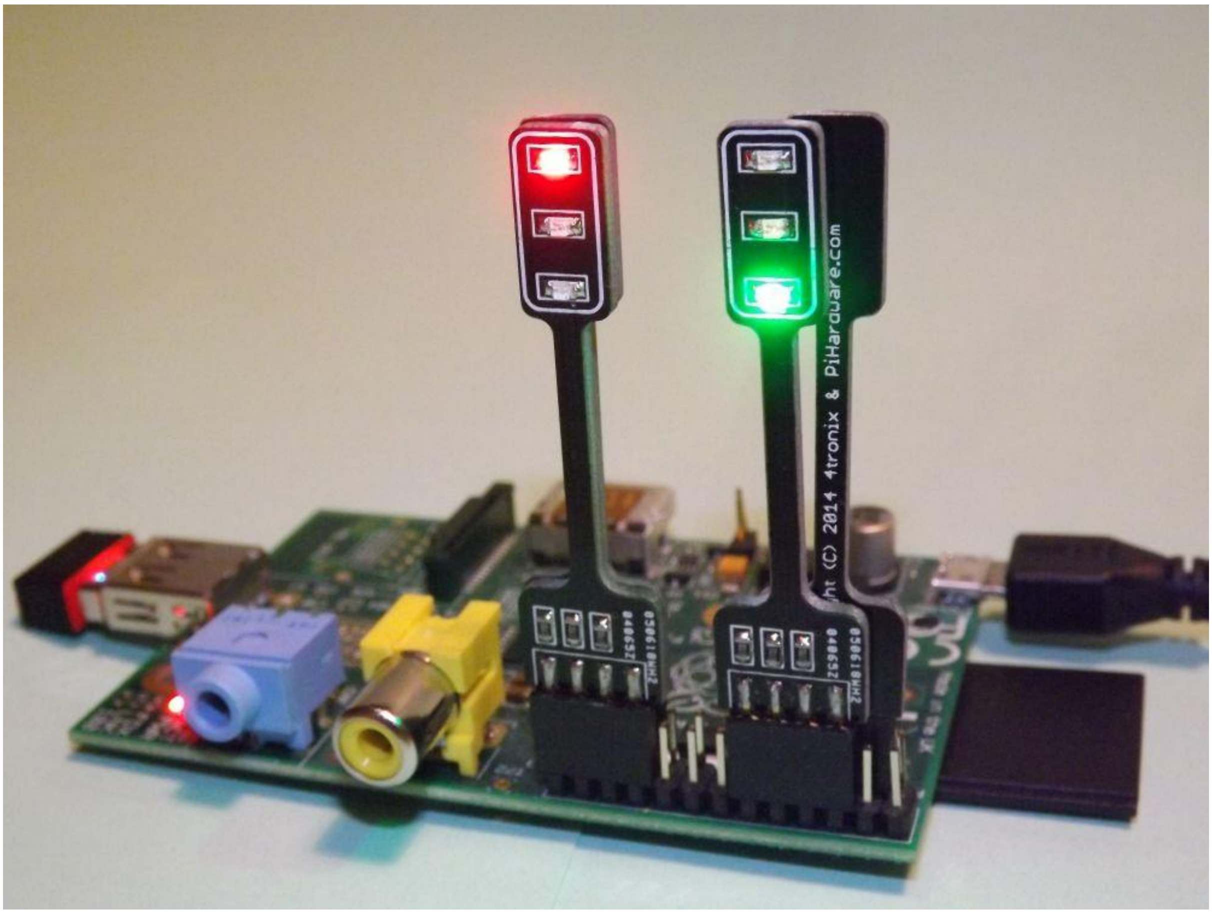

Figure 7: Mini light alerts setup [9]

Design Specification	Detail	P	E	F
HW-19H	Alerts module must be recognized by the board/software and communicate without delay	x	x	x
HW-20H	Audio module must be of good quality and promps are understandable		x	x
HW-21H	Lights module must be noticeable to driver but not distracting		x	x
HW-22M	Alerts will be able to fit and secured on top of or within Raspberery Pi Case Model B			x

Table 6: Hardware Design Specification for Alert Interface

3 Object Detection System

In terms of software, the code will focus on human detection technology from the emerging deep learning industry. With proper integration, it is a versatile computer vision technology that identifies the locations and sizes of human faces or bodies with high accuracy automatically.

The system will make use of Haar feature-based cascade classifiers from the OpenCV library, which is a collection of programming functions mainly aimed at real-time computer vision. Recognized by the industry, Haar cascade classifier is one of the most efficient and robust object detection ecosystems which allows users to train AI from a variety of positive (image with particular objects) and negative (image without the specified objects) samples.[10] After going through a large number of error-elimination iterations, the code will be able to detect any target object accurately.

When the camera turns on, the detection system on the software backend will read in the captured video frame-by-frame, convert the frames into grayscale, and store them into a dynamic list. Afterwards, it will draw a rectangular outline on each target object in the video to accurately indicate the location and distance of the object in the real time on the hardware screen. More importantly, the software will send an alert to the hardware side, so that the driver will take notice when a pedestrian is nearby.

For the final product, additional useful software features such as Google Maps, GPS, and Dash Cam functionalities will be implemented to improve the versatility of the design.

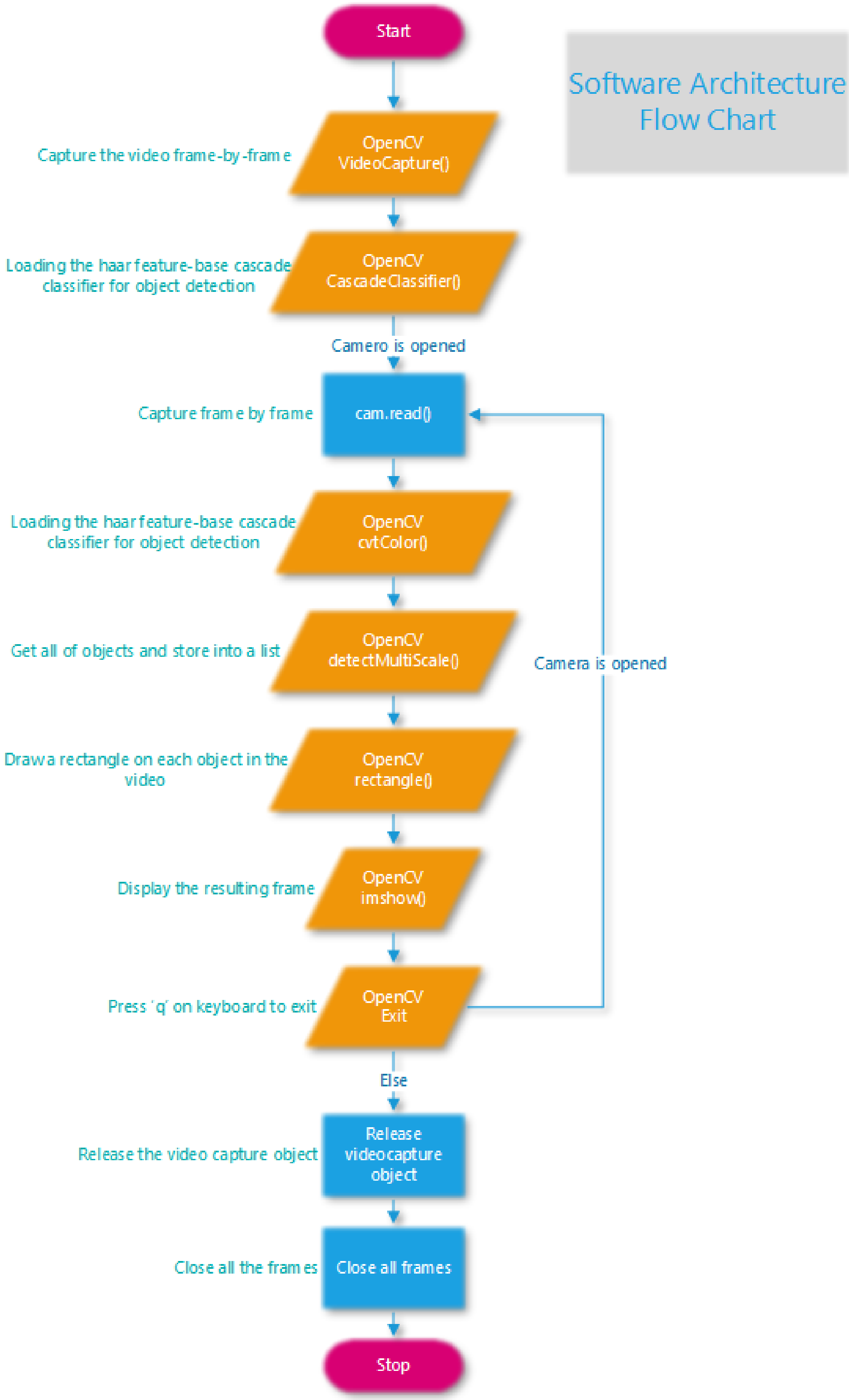

Figure 8: Software Architecture Flow Chart of Machine Learning embedded on NightEagle

With these elements in mind, the specifications for the human recognition software can be compiled and analyzed. The following details the design architecture the team has identified for the software in terms of usability, accuracy, and functionality.

Design Specification	Detail	P	E	F
SW-23H	Software must be able to recognize human face in no-light situations	x	x	x
SW-24H	Software should be able to recognize human body in no-light situations	x	x	x
SW-25H	The trained deep learning AI model should be able to fit onto the board in terms of ram and size		x	x
SW-26M	The AI should be able to function in real-time with little delay even on a lesser processor on Raspberry		x	x
SW-27M	Software should be able to send signals and flags to appropriate hardware pieces		x	x

Table 7: Software Design Specification for Object Detection System

4 Driver Alert System

As mentioned in the electronics section, two different options are being considered for the alert system. The end product should be able to inform the driver as soon as the software detects a pedestrian within the camera's line of sight. However, it is critical that the alerts are non-distracting and clear to the user. The following two solutions have been deemed to fit the requirements, and the final implementation will be decided as testing phases conclude.

Light Alerts Solution

The team at EagleVision has tested many different types of lights, and soft hues such as blue light seems to be a good fit when placed in a dark environment. It is noticeable as it contrasts greatly from external lights, and is not harsh to the human eyes. In this implementation case, the software core would send a flag to the motherboard whenever it recognizes a human from the video data, and in turn the light embedded to the front of the device would light up once it receives the signal. The light-based alert setup would have the following specifications.

Design Specification	Detail	P	E	F
HW-28H	Light must light up as soon as human is detected	x	x	x
HW-29H	Light should be noticeable but not distracting during driving	x	x	x
HW-30L	Light should last for many iterations of usage			x
HW-31L	Multiple lights around the device indicates general direction of the pedestrian relative to the device			x

Table 8: Hardware Design Specification for Light-based Alert Setup

Audio Alerts Solution

The other solution is a more ambitious one, as it would involve programming a voice announcement system. This would take the same signal as with the lights solution, but the board would hook up to a mini-speaker instead. The team can make use of the Pyttsx library to have access to text-to-speech services. This will potentially reduce the conflict of visual alerts and driving, and provide an advanced and safe format of communication.

Design Specification	Detail	P	E	F
HW-32H	Voice alert must activate as soon as human is detected	x	x	x
HW-33H	Voice should be noticeable but not distracting during driving	x	x	x
HW-34L	Speaker should last for many iterations of usage			x
SW-35L	Voice announcement indicates general direction of the pedestrian relative to the device			x

Table 9: Hardware Design Specification for Audio-based Alert Setup

5 Recognition Visualization System

One of the other core functionalities of the device is to highlight the pedestrian that has been recognized and feed it back onto the screen. As described in the Object Detection section, the software core will draw a rectangular box around the faces and bodies detected in each frame of the video, and feed it back in a visual form. This is designed to allow the driver to slow down if they cannot physically see the pedestrian from their windshield, and use the NightEagle device as a reference point to pinpoint the direction of the object.

Taking into consideration the accuracy of the feedback to the driver, the nature of the machine learning backbone can be analyzed. Below is a comparison of the machine learning training library chosen by the team, OpenCV, in comparison to other common libraries on the market. Although it takes the longest out of the three, at 720p (1279x718) and 820p (1600x812), the processing time is comparable to the others. Since the software will be analyzing footage from a small night-vision camera, these resolutions should suffice when considering the efficiency and speed of the recognition.

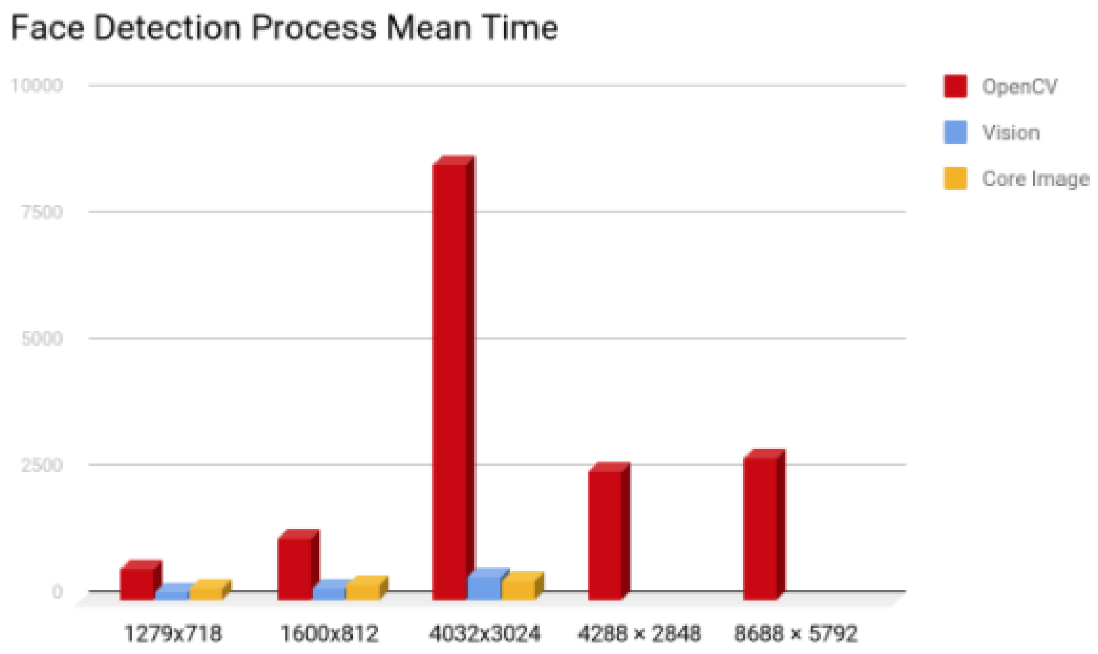

Figure 9 - Facial Detection Process Time comparisons between 3 common libraries [11]

In addition, when examining the accuracies of the 3 libraries, it is clear that opencv has an edge. While it is unrealistic that there will be multiple pedestrians on the road in one frame at night, it does allow more flexibility in rare cases. Meanwhile, in frames with fewer faces, opencv still maintains its accuracy, with reasonable processing time; this will serve the common usage case of the device.

Image with multiple faces	Image Resolution	Image Reference	Process Time (First Process -> Second Process in Millisecond)			Face Detected		
			Core Image	Vision	OpenCV	Core Image	Vision	OpenCV
16	1279 × 718	https://drive.google.com/open?	374 -> 153	174 -> 131	596 -> 577	5	10	14
3	4032 × 3024	https://drive.google.com/open?	454 -> 155	212 -> 277	8688 -> 8533	3	3	3
37	1600 × 812	https://drive.google.com/open?	542 -> 284	529 -> 379	1035 -> 1187	7	34	37
60	4288 × 2848	https://drive.google.com/open?	309 -> 119	157 -> 121	2638 -> 2485	Failed	Failed	60
86	8688 × 5792	https://drive.google.com/open?	393 -> 139	205 -> 107	2835 ->2808	Failed	Failed	86

Figure 10: Table detailing accuracy of each library [12]

Currently, the team has a working AI model trained to recognize humans in regular light, up to multiple faces per frame.. As can be seen from the figure below, the software draws a frame around the recognized object and displays that on the screen. Night-vision compatibility will added and tested with a variety of camera modules.

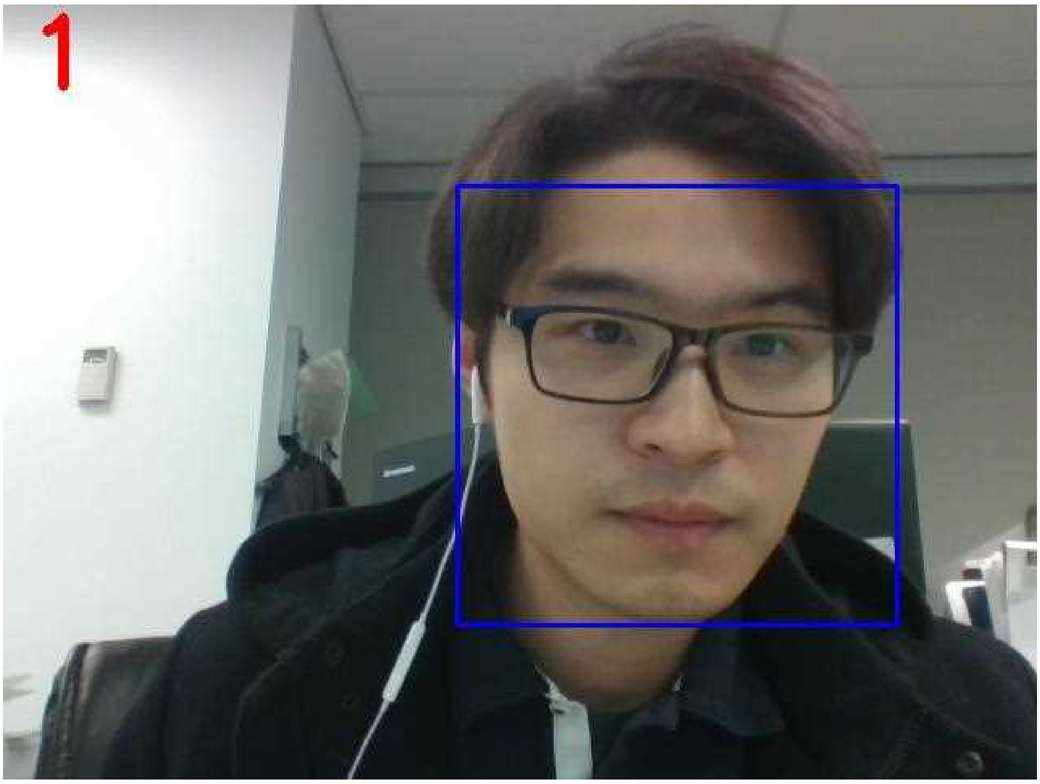

Figure 11: Sample of current prototype EagleVision machine learning AI

With the functionality and requirements of the visualization component in mind, the following specifications can be prescribed.

Design Specification	Detail	P	E	F
SW-36H	AI must be able to draw highlight around detected pedestrians	x	x	x
SW-37H	AI must be able to send output video signal + highlighted shape to external screen		x	x
SW-38M	Video outputted to screen should be of HD quality		x	x
SW-39L	Highlight also displays useful information such as coordinates and distance of pedestrian			x

Table 10: Software Design Specification for Recognition Visualization System

6 Electrical Power System

One of the most basic yet quintessential specifications on the physical device is the electrical component. Examining the circuit diagram in Figure 12 for common Raspberry power unit, it can be noted that it takes in a maximum of 5V, with a current limit set. The board will take in power from the car's outlet via USB in order to be operational. Later iterations may incorporate a rechargeable battery to increase portability.

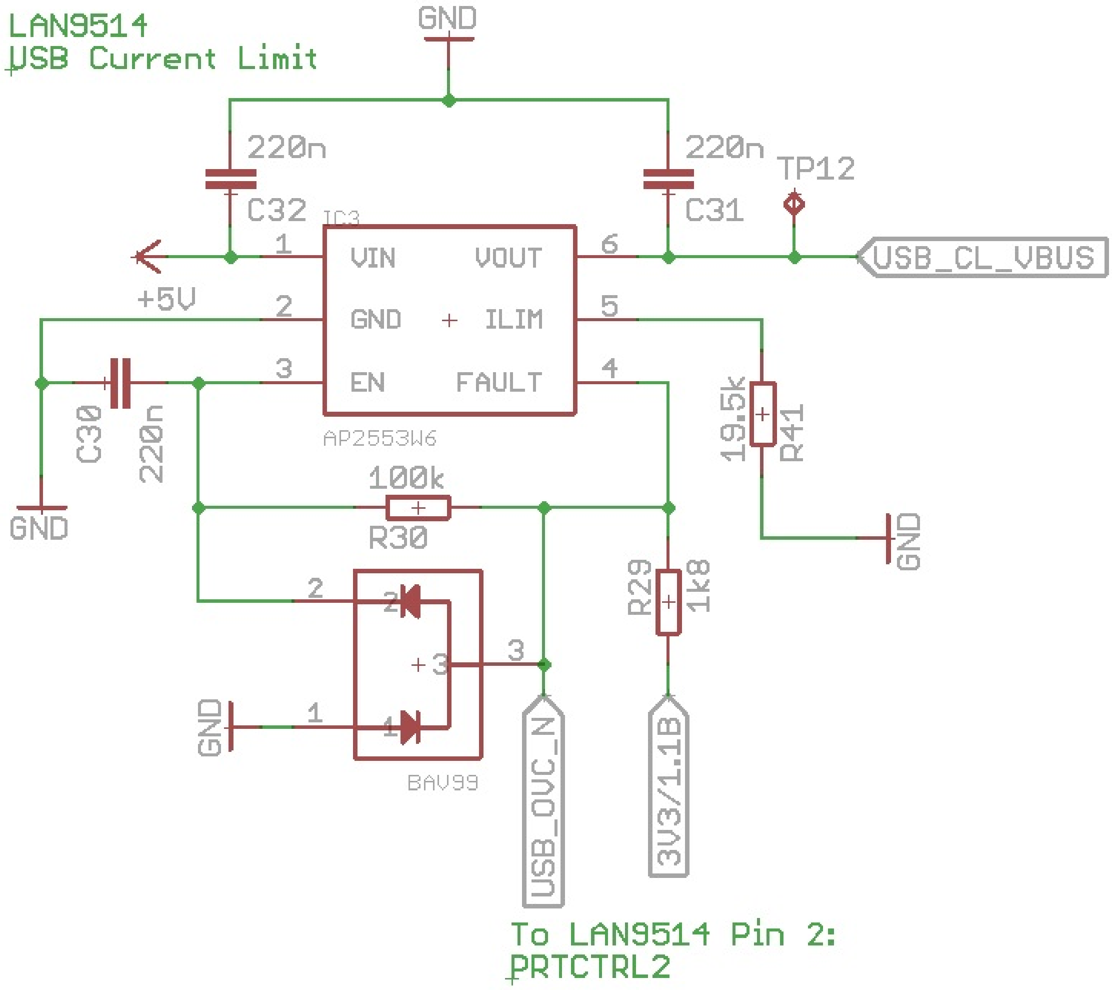

Figure 12: Circuit diagram for Raspberry Pi power block [13]

The one other component which needs an explicit source of power is the external screen. As mentioned in its specific section, there are already multiple third-party options for a Raspberry-compatible screen. Figure 13 details a generic setup for such periphery, including socket planning for HDMI, external control board, and power intake from the GPIO pins.

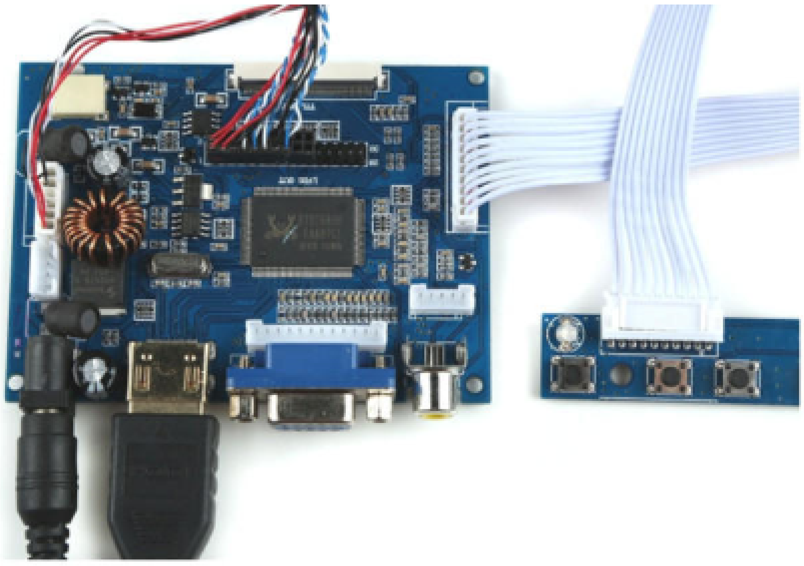

Figure 13: Sample setup of external screen with Raspberry board [14]

With the above two power requirements in mind, the design specifications can be completed below.

Design Specification	Detail	P	E	F
HW-40H	Raspberry board must be able to accept 5V from vehical outlet and turn on	x	x	x
HW-41H	External screen must accept power from board or external source and turn on		x	x
HW-42H	Board must be able to function for extended time without electrical short circuits		x	x
HW-43L	Device should avoid issues of overheating and excess power consumption			x

Table 11: Hardware Design Specification for Electrical Power System

7 Device Interface

7.1 Mechanical Casing

In order to keep all the components packed in tightly, a hard outer shell will be applied. There exists various aftermarket options such as in Figure 14, which are designed to encase a Raspberry board with the appropriate ports. The screen and other hardware components will be secured to the outside of the casing, so it would be a rigid and solid device after production.

Figure 14: Casing option for final production [15]

The particular case the team is in favour of is an official option from Raspberry itself. The Official Raspberry Pi 3 Model B offers features such as:

- Easy snap-fit assembly
- Range of case display/access options: fully closed, sides away, birds eye view
- Cut-outs of all connection points
- Plastic LED light protector
- microSD card cut-out
- Stick on rubber feet for case stability

These features align with the specifications for the casing, and is a top-contender in terms of functionality and pricing. The prescribed properties are listed below:

Design Specification	Detail	P	E	F
HW-44H	Raspberry casing must hold all components in place and protect electronics	x	x	x
HW-45H	Casing must provide appropriate cutouts for buttons and out-facing modules		x	x
HW-46M	Casing should not come apart easily		x	x
HW-47L	Casing should not cause damage or overheating issues to components inside			x

Table 12: Hardware Design Specification for Mechanical Casing

In addition, the case will allow for access to the button controls which will be needed to operate the device in its various states.

7.2 Algorithm States

Part of the integration between software and hardware is captured within the algorithm states of the device. As can be seen from the state diagram, there are three possible states for the NightEagle. When the power button is in the off position, the device will also be Off. When the power button is pressed for a reasonably short 1.5 seconds, it will boot up to the On state. The length of this boot up process will depend on the final software optimization, but the team will reduce it as much as possible. During the On state, the device can still be forced back into the Off state by holding the button for 5 seconds. After the On state has concluded, it will automatically transfer to the Active/Looping state. The device will keep on looping and the machine learning AI will be active during this state, and provide all expected functionalities. Finally, the device can be switched back into the Off state by holding the off button for 1.5 seconds.

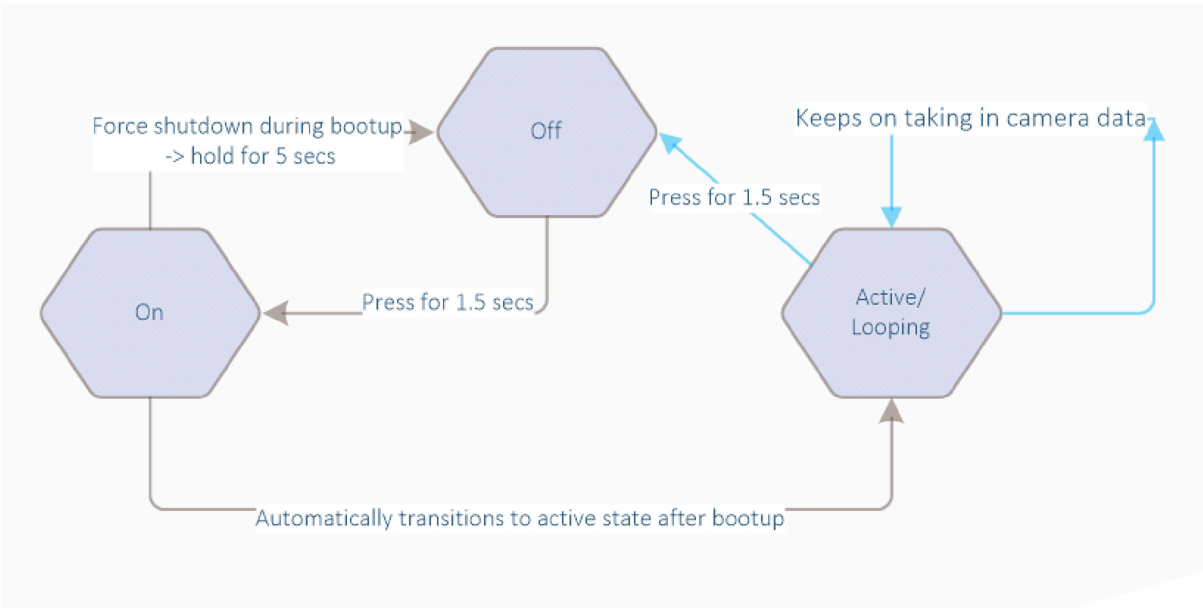

Figure 15: State diagram for NightEagle

The state design of the device leads to the following specifications:

Design Specification	Detail	P	E	F
HW-48H	Device must go into On state when power button is pressed	x	x	x
SW-49H	UI and software must start up when state is On	x	x	x
SW-50H	Machine Learning AI must be able to stay on and	x	x	x

	loop while in Active state until powered off			
HW-51M	Device must go to Off state when power button is pressed during On or Active state		x	x
SW-52L	Program must shut off and close all open sessions when in Off state			x

Table 13: Software Design Specification for Algorithm States

9 Conclusion

This document served as both a high-level summary of the device architecture, as well as a detailed exploration of each hard and software component. The specifications of the overall product along with all the individual components were broken down and analyzed to great detail. In addition, the targeted goal of completion for each was listed for reference (PoC, Engineering Prototype, Final Product). In summary, the following design specifications aspects of the device were explored throughout the document:

a) Electronics Design

i) Main Central Computer
ii) Video input interface
iii) Video output interface
iv) Hardware Alerts Interface

b) Object Detection System

c) Driver Alert System

i) Light Alerts Solution
ii) Voice Alerts Solution

d) Recognition Visualization System

e) Electrical Power System

f) Device Interface

i) Mechanical Casing
ii) Algorithm States

EagleVision Inc. plans to finish the proof of concept and Appearance Prototype in April 2019, and the final product in August 2019. The Test Plan is included in Appendix A and the User Interface document is located in Appendix B. They will be used as a testing standard to guide the usability of the product in the intended direction. Through the different skill sets and attributes of each members in the team, EagleVision strives to create a safer environment for all drivers and pedestrians. The NightEagle device will serve as an affordable and easy-to-access product for any vehicle on the road, ensuring the reduction of night-time vehicular collision fatalities.

10 References

[1] S Plainis, I J Murray, and I G Pallikaris, "Road traffic casualties: understanding the night-time death toll", NCBI, 2006. [Online]. Available:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2564438/

[2][10] "Face Detection using Haar Cascades", OpenCV, 2018. [Online]. Available: https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

[3] "Histogram of oriented gradients", En.wikipedia.org, 2019. [Omline]. Available: https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

[4] "Know all about Raspberry Pi Board Technology", EDGEFX KITS & SOLUTIONS, 2017. [Online]. Available: https://www.edgefxkits.com/blog/raspberry-pi-technology-with-applications/

[5] Raspberry Pi 3 Model B+ [Digital image]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi

[6] Common camera module I/O for Raspberry [Digital image]. Available: https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjgwtTkwongAhUEiFQKHVVuBPIQjRx6BAgBEAU&url=https%3A%2F%2Fcore-electronics.com.au%2Fraspberry-pi-wide-angle-camera-module-seeed-studio.html&psig=AOvVaw3lriLLIHJpcBbvXu0nRb54&ust=1548525906045421

[7] Raspberry Pi Official Screen Case - White [Digital image]. Available: https://www.robotistan.com/raspberry-pi-official-screen-case-white

[8] Mini Speaker with ⅛" Plug [Digital image]. Available: https://tinkersphere.com/audio/964-mini-speaker-with-1-8-inch-plug.html

[9] PiStop - Traffic Light Add-on for Raspberry Pi [Digital image]. Available: https://shop.pimoroni.com/products/pistop-traffic-light-add-on-for-raspberry-pi

[11][12] Face Detection with Vision - Core Image & OpenCV [Digital image]. Available: https://www.softvision.com/blog/face-detection-vision-core-image-opencv/

[13] Schematics of Raspberry Pi 3 Model B [Digital image]. Available: https://www.raspberrypi.org/forums/viewtopic.php?t=138280

[14] 10.1 Inch 1280x800 HD Display TFT LCD Module Kit For Raspberry Pi [Digital image]. Available: https://www.banggood.com/10_1-Inch-1280x800-HD-Display-TFT-LCD-Module-Kit-For-Raspberry-Pi-p-1109750.html?gmcCountry=CA&currency=CAD&createTmp=1&utm_source=googleshopping&utm_medium=cpc_bgs&utm_content=frank&utm_campaign=pla-mix-ca-pc-06

30&gclid=CjwKCAjw1KLkBRBZEiwARzyE7-9h7Xp9KSVD9rSDMMLVOhEGA3eD0ycrQQTr5SpWedWop8i0NfCwShoCRhQQAvD_BwE&cur_warehouse=CN

[15] Raspberry Pi 3 Case Black and Grey [Digital image]. Available: https://www.umart.com.au/Raspberry-Pi-3-Case-Black-and-Grey_39940G.html

[16] “Adafruit PiTFT - 320x240 2.8” TFT+Touchscreen for Raspberry Pi”, Adafruit.com. [Online]. Available: https://www.adafruit.com/product/1601

"PSES IEEE - Product Safety Engineering Society", Ewh.ieee.org, 2019. [Online]. Available: http://ewh.ieee.org/soc/pses

[17] “Discoverability (in UX design)”, WhatIs.com, 2016. [Online]. Available: https://whatis.techtarget.com/definition/discoverability-in-UX-design

[18] “Illuminated Raspberry Pi Safe Shutdown Power Switch”, KICKSTARTER. [Online]. Available:https://www.kickstarter.com/projects/1312527055/illuminated-raspberry-pi-safe-shutdown-switch

[19][21][22][23] “Don Norman’s Principles of Interaction Design”, A Medium Corporation, 2017. [Online]. Available:https://medium.com/@sachinrekhi/don-normans-principles-of-interaction-design-51025a2c0f33

[20] “Audi Optical Parking Sensors Kit”, Advanced In-Car Technologies. [Online]. Available:https://www.advanced-incar.co.uk/products/parking-sensors/audi-optical-parking-sensors-kit/

[24] “IEEE Safe Engineering Methodologies”, ewh.ieee.org, 2019[Online]. Available: http://ewh.ieee.org/soc/pses/

[25] “Google Python Style Guide”, Google.github.io, 2019. [Online]. Available: https://google.github.io/styleguide/pyguide.html

[26] “IEEE 1012-2012 - IEEE Standard for System and Software Verification and Validation”, standards.ieee.org[Online]. Available: https://standards.ieee.org/standard/1012-2012.html#Additional

Appendix A Test Plan

A.1 Introduction

This appendix details the test plan for each subsystem and peripheral of the product.

A.2 Electronics Testing

Date:	**Test Name:** Power Consumption
Test Description: Measure the power consumption of the motherboard, the peripherals (touchscreen, camera, speaker, charger IC) and the whole system.	
Expected Outcome: The system's total power must not exceed 24 watts.	
Actual Outcome:	

Date:	**Test Name:** Motherboard Signals
Test Description: Simulate different product features, check if the motherboard input, output, and intermediate signals react correctly.	
Expected Outcome: The motherboard input, output, and intermediate signals react correctly.	
Actual Outcome:	

Date:	**Test Name:** Camera Settings Compatibility
Test Description: The camera settings (e.g. frames per second, pixels, etc.) must be compatible with the hardware and software on the motherboard.	
Expected Outcome: The motherboard is capable to process the camera signals in an effective manner.	
Actual Outcome:	

Date:	**Test Name:** Touchscreen Settings Compatibility

Test Description: The touchscreen settings (e.g. frames per second, pixels, etc.) must be compatible with the hardware and software on the motherboard.
Expected Outcome: The touchscreen is capable to interact with the motherboard correctly when performing all predefined functions.
Actual Outcome:

Date:	**Test Name:** Battery Charging Power
Test Description: Charge the completely discharged battery. Monitor the charging power until fully charged.	
Expected Outcome: The charging current must not exceed 1 amp.	
Actual Outcome:	

Date:	**Test Name:** Battery Charger Safety
Test Description: Leave the charger connected to the battery after the battery is fully charged. Monitor the charger's current draw.	
Expected Outcome: The current draw must be near zero.	
Actual Outcome:	

A.3 Software Testing

Date:	**Test Name:** Target Identification
Test Description: Test the program by feeding it with video footages containing target objects.	
Expected Outcome: The program should identify the target objects and send appropriate outputs.	
Actual Outcome:	

Date:	**Test Name:** No Target Identified
Test Description: Test the program by feeding it with video footages containing no target	

objects.
Expected Outcome: The program should maintain its regular processes.
Actual Outcome:

Date:	**Test Name:** Alerts Triggering
Test Description: Feed the program with video footages containing target objects. Monitor alert signals.	
Expected Outcome: Appropriate alert signals are triggered.	
Actual Outcome:	

Date:	**Test Name:** Real-time Effectiveness of Recognition and Alerting
Test Description: Measure the reaction time elapsed from target object appearance to alert triggering.	
Expected Outcome: The total reaction time must not exceed 1 second.	
Actual Outcome:	

Date:	**Test Name:** RAM Usage
Test Description: Monitor the RAM usage when testing all program functions.	
Expected Outcome: The RAM usage must not exceed 512 MB.	
Actual Outcome:	

A.4 Enclosure and Mechanical Testing

Date:	**Test Name:** Drop Test
Test Description: Drop the product onto carpeted floor from 0.5 metre and check for integrity.	
Expected Outcome: The product should not have catastrophic damage.	
Actual Outcome:	

Date:	**Test Name:** Dust Test
Test Description: Expose the product to dusty (falling dirt) environment for 8 continuous hours and check for ingress of foreign objects.	
Expected Outcome: Little to no foreign object should be found inside the enclosure. The proper operation of the device should not have been affected.	
Actual Outcome:	

Date:	**Test Name:** Moisture Test
Test Description: Expose the product to humid environment with over 90% relative humidity and check for condensation and device functionality.	
Expected Outcome: Little to no water condensation should be found inside the enclosure. The proper operation of the device should not have been affected.	
Actual Outcome:	

Date:	**Test Name:** Screen Adjustment Test
Test Description: Adjust the screen angle 10,000 times and check for hinge / slide tightness.	
Expected Outcome: The mechanical connection between the screen and the body of the device remains tight.	
Actual Outcome:	

ENSC 405W Grading Rubric for User Interface and Appearance Design
(5-10 Page Appendix in Design Specifications)

Criteria	Details	Marks
Introduction/Background	Appendix introduces the purpose and scope of the User Interface and Appearance Design.	**/05%**
User Analysis	Outlines the required user knowledge and restrictions with respect to the users' prior experience with similar systems or devices and with their physical abilities to use the proposed system or device.	**/10%**
Technical Analysis	Analysis in the appendix takes into account the "Seven Elements of UI Interaction" (discoverability, feedback, conceptual models, affordances, signifiers, mappings, constraints) outlined in the ENSC 405W lectures and Don Norman's text (*The Design of Everyday Things*). Analysis encompasses both hardware interfaces and software interfaces.	**/20%**
Engineering Standards	Appendix outlines specific engineering standards that apply to the proposed user interfaces for the device or system.	**/10%**
Analytical Usability Testing	Appendix details the analytical usability testing undertaken by the designers.	**/10%**
Empirical Usability Testing	Appendix details completed empirical usability testing with users and/or outlines the methods of testing required for future implementations. Addresses safe and reliable use of the device or system by eliminating or minimizing potential error (slips and mistakes) and enabling error recovery.	**/20%**
Graphical Presentation	Appendix illustrates concepts and proposed designs using graphics.	**/10%**
Correctness/Style	Correct spelling, grammar, and punctuation. Style is clear concise, and coherent.	**/05%**
Conclusion/References	Appendix conclusion succinctly summarizes the current state of the user interfaces and appearance and notes what work remains to be undertaken for the proof-of-concept and appearance prototypes. References are provided with respect to standards and other sources of information.	**/10%**
CEAB Outcomes: Below Standards, Marginal, Meets, Exceeds	1.3 Engineering Science Knowledge: 4.1 Requirement and Constraint Identification: 5.4 Documents and Graphic Generation: 8.2 Responsibilities of an Engineer:	

Version of 25 February 2019 by Craig Scratchley. Updated from Version provided by Steve Whitmore.

Appendix B User Interface and Appearance Prototype Design

B.1 Introduction

B.1.1 Purpose

This document focuses on helping users understand the functionalities of NightEagle and how to navigate the user interface properly. Diagrams and detailed annotations will be provided to explain the concepts and design language of the UI.

B.1.2 Scope

As the team is still in the initial design stage of NightEagle, this report will focus on the Proof of Concept and Engineering Prototype UI by the following scope of analysis:

- User Analysis
    - Describes user knowledge required for safe operation of the device and the restrictions the user has to abide by
- Technical Analysis
    - Discusses seven technical elements of user interface which includes discoverability, feedback, conceptual models, affordances, signifiers, mappings, and constraints for hardware and software interfaces.
- Engineering standards
    - Introduces specific engineering standards for the NightEagle user interface to be proper produced and consumed by the market.
- Usability Testing
    - Includes analytical usability testing and empirical usability testing for the product usability.

B.2 User Analysis

With the features of this product in mind, NightEagle’s main market is drivers of all classifications of vehicles, including cars, motorcycles, and especially bicycle. For the UI design, the primary goal of EagleVIsion is to provide an easy-to-use user interface for any user to get started. Users can become fluent on its usage based on their common sense, without any special knowledge.

Figure 16: Raspberry Pi Touchscreen [16]

This product is ready for use without any other additional installations or configurations after purchase. Users just need to fix it onto the vehicle and adjust the point of view properly. The core usage of NightEagle is similar to most electronic products on the current market, such as smartphones and tablets (Figure 16), which guides users with a touch screen and a modular operational interface.

For long-term use of this product, it is recommended that users permanently fix it internally within the vehicle and, recharge the battery regularly. Also, a simple user manual will be provided with this product, detailing the control scheme.

B.3 Technical Analysis

The technical analysis section will describe both hardware and software interfaces from the perspective of the seven elements of UI interaction: discoverability, feedback, conceptual models, affordances, signifiers, mapping, and constraints.

B.3.1 Discoverability

Discoverability, in the context of product and interface design, is the degree of ease with which the user can find all the elements and features of a new system when they first encounter it.[17] EagleVision is committed to minimalism, and will simplify the appearance, user interface, basic operation, and user experience.

NightEagle only has one physical button in the form of the power ON/OFF switch (Figure 17), and a simple touch screen to display the modular user interface on the device. It is extremely simple to use without any other additional configuration, and will appeal to the average user.

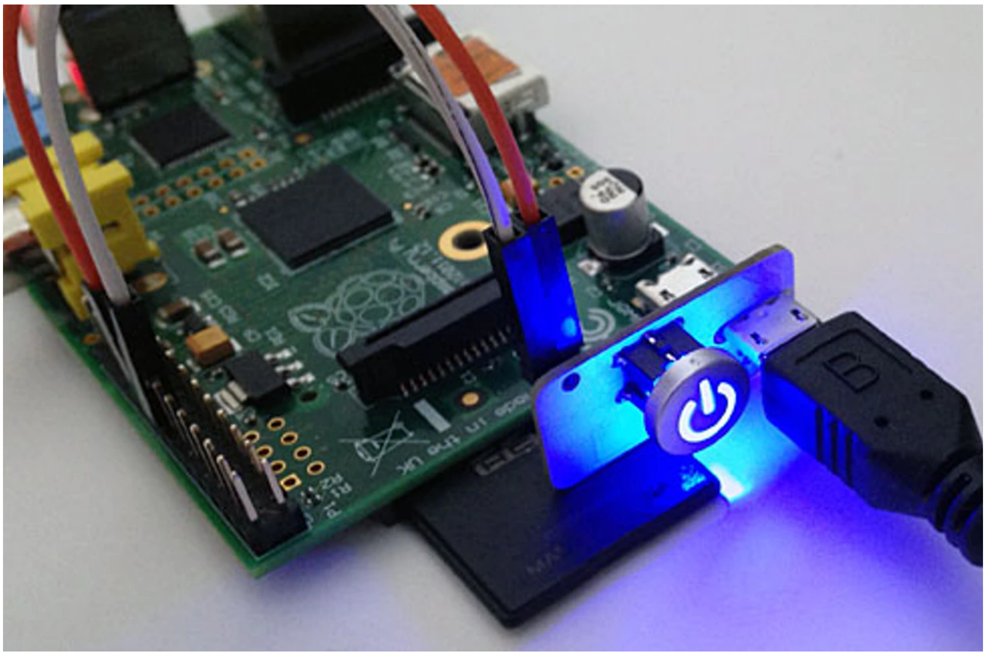

Figure 17: Illuminated Raspberry Pi Safe Shudown Power Switch [18]

B.3.2 Feedback

Feedback is the principle of making it clear to the user what action has been taken and what has been accomplished.[19]

Currently, similar products on the market are generally express their feedback by visual, tactile, audio, and other cues. For the NightVision, a speaker or Light Emitting Diode (LED) will be chosen to execute the feedback for the engineering prototype. In the final product, EagleVision plans to develop an feedback system to alert the driver by illuminating the frontal area of the device, lined with LEDs, depending on the different directions the pedestrian has been detected in.

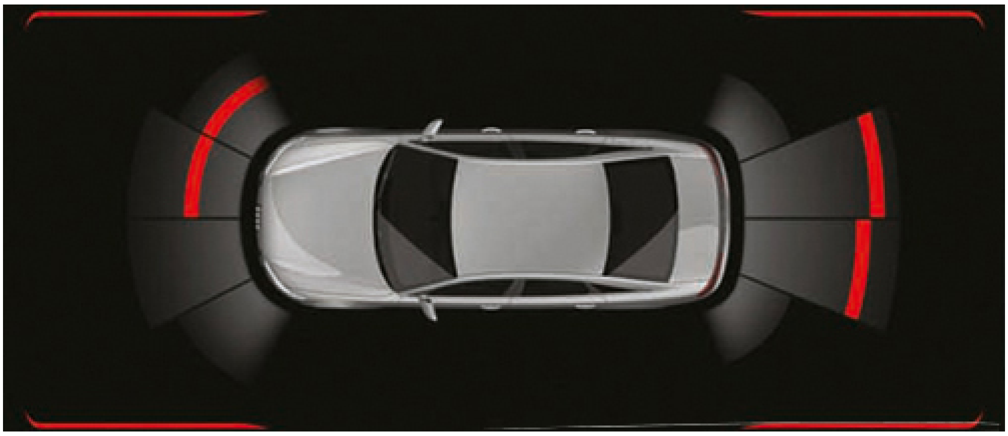

Figure 18: Feedback System [20]

B.3.3 Conceptual Models

The Conceptual Model of a device will help users understand the concept and usage of the product. It also explains the product benefits and main feature operations. Figure 19 displays an alpha render of what the team is striving to develop for the UI. Each feature is laid out in large tiles, with the name right below. This simple yet classy design will allow common users of modern smartphones to feel right at home and be able to navigate fluently.

Figure 19: Conceptual UI

B.3.4 Affordances

Affordance refers to an attribute of an object that allows people to know how to use it. Essentially to afford means to give a clue.[21] NightVision will provide a high affordance power on/off button with a symbolic label, and an user interface that is human factor compliant. Users will know how to use it as soon as they see the product. As was displayed in the Conceptual Model, the overall UI is very similar to current iterations of Android and iOS, and combines the large icon tiles from Windows 8. This setup will remove all unnecessary clutter from the main menu, as well as provide a modern and clean look.

B.3.5 Signifiers

Signifiers are supplement of Affordances. When the perception of affordances is not enough to remind user on how to use the product, signifier will be necessary. As labels will be provided on the power on/off button and the AC in slot, it will help users immediately recognize the usage of each. The team will make sure the signifiers are far apart and clearly labeled, to avoid users mixing them up.

B.3.6 Mapping

Mapping is about having a clear relationship between controls and the effect they have on the world. Users always want this mapping to feel as natural as possible.[22]

In NightEagle's case, users will not have any mapping confusions due to the extremely simplified operating platform, and will even be usable by children and elders. As was displayed in previous sections, there is a clear 1-to-1 mapping between the three features and the buttons they correspond to. Once the user enters the main feature of the device, which is the human-detection mode, there will be no more input needed from their part, as the program will loop by itself.

B.3.7 Constraints

Constraints is about limiting the range of interaction possibilities for the user to simplify the interface and guide the user to the appropriate next action.[23] Referenced in the Technical Analysis section, NightEagle only has a power on/off button with a simplified software user interface. Users will not need to worry about confusion from multiple buttons without any signifiers and a complex user interface operating platform.

Since the final design will most likely be based on a Linux system, the UI team will be limiting the device so that it boots into the main program as soon as it starts up. This effectively constrains the user to the three options presented in the conceptual model, and will avoid consumers accidentally entering the Linus environment. In terms of hardware, the switch will be carefully designed so that the users only have two clear states, either on or off.

B.4 Engineering Standards

To achieve the standard of a marketable product, NightEagle's UI will be designed and tested by the following engineering standards:

1. IEEE Safe Engineering Methodologies.[24]
2. Google's Coding Standards for Programming and Scripting Languages.[25]
3. IEEE 1012-2012 - IEEE Standard for System and Software Verification and Validation.[26]

B.4.1 Safety Considerations

To ensure the safety of operating NightEagle's UI, the team will consider the following safety standards to ensure the stability of this product:

1. Component manufacturer sustainability and fir labour policies.
2. Reusing components in the PoC prototype to build the prototype.
3. Providing environmental impact and recycling information to distributors and end-users.

B.5 Usability Testing

Every single device produced has to be tested to simulate difficulties users may encounter when using it, before releasing it to the market. It will help the development team to improve the product by solving bugs, following human behaviours, abilities, limitations, and obtain ideas for any overlooked features. The EagleVision team will execute the usability testing from both an analytical and empirical direction.

B.5.1 Analytical Usability Testing

1. Start & Stop
    a. The power on/off button is accessasible from the device surface.
    b. The power on/off button is at visible position of product surface.
    c. The power on/off button has physical feedback to user's finger when pressed down.
    d. The power on/off button has a simbolic label to indicate its feature.
    e. While the device is deactive, it can be turned on by pressing the power on/off button.
    f. The camera is turned on when the device is turned on.
    g. The red LED is turned on when the camera detects the object in front of the vehicle.
    h. The speaker is turned on when the camera detects the object in front of the vehicle.
    i. The touch screen of final product is on when the device is turned on.
    j. While the device is active, it can be turned off by pressing the power on/off button.
    k. The touch screen of final product is off when the device is turned off.
2. Touch screen status (in the final product)
    a. The UI on the screen operates properly.
    b. The detection system on the UI alerts the driver when the device detects the object in front of the vehicle.
    c. The screen displays what the camera sees and highlights the object.
3. Camera status
    a. The camera runs properly during the entire operation cycle.

    b. The night vision feature of the camera operates successfully in the dark.
4. Battery Recharge
    a. Battery can be charged without any risks.
    b. The recharging port is easy to find and plug in.
    c. The recharging port is waterproof.

B.5.2 Empirical Usability Testing

For the empirical testing phase, EagleVision will employ numerous vehicle owners to test and answer a series of questions regarding the user experience. The team will analyze any feedback from users to fix possible problems and optimize the final product. The NightEagle team will ask the following questions for the product functionality and user experience survey during the empirical testing stage.

1. Power on & power off
    a. Could the device power on and power off successfully?
    b. Do all other components (screen, speaker, charging) work properly?
2. Alert system
    a. Is the speaker and LED enough to alert the driver?
    b. Are the alerts distracting to the driver?
    c. Did the device alert you with enough time to slow down or stop the vehicle?
    d. Did the device alert you too early while the pedestrian was far away?
    e. Did the device fail to recognize pedestrians, or recognized other objects as humans by mistake?
3. User interface in final product
    a. Is easy to navigate the user interface without prior knowledge?
    b. Was it obvious for you to recognize the active detection system on the UI in relation to the road?
    c. Were you able to successfully locate the pedestrian as was highlighted on the screen?
    d. Was the screen display of the road large/wide enough to reflect what the driver was seeing in front of them?
    e. Was the device easy to set up in the vehicle upon first use?
    f. Were there any lag/mismapped parts on the UI?
4. Battery recharging
    a. Do you think the battery is enough for daily using?
    b. Do you think the recharging is insulated and safe to use?

Base on these evaluations, NightEagle team will make adjustments and improve the performance and functionalities to match the user requirements.

B.6 Summary

The user interface of any device largely determines the customer experience., and thus is quintessential for EagleVision. A well-designed UI helps users get started quickly, without reading the user manual or receive training. It can also simplify the overall process with one-click operations that eliminates confusion.

In the proof of concept model, the NightEagle will be the basic edition of the product with an embedded Raspberry Pi board and the power on/off button. Its main purpose will be to verify the team's software human detection functionality and the practicality of the board model chosen.

For the engineering prototype, the NightEagle team will provide a more complete product which includes an embedded Raspberry Pi, power on/off button, night vision camera, and speaker or LED alerts. It will run on AC power through an extension cable. It is the alpha edition with the majority of the final components needed to detect humans by the night-vision camera in real-time, before the final product.

The final product will add the touch screen to render the finished user interface. It should be able to upkeep the FPS of the output footage at least 20 FPS to detect human efficiently. A portable battery will be installed so that users have more flexibility and freedom to install the device in any way that fits their vehicle best, along with improving portability.

Moreover, The NightEagle team will pursue the consistent improvement of the UI design based on feedback from the market and its end users, in order to provide the most efficient, simple, and affordable night-time human detection device on the market.