

GNG 2101 Deliverable G

Prototype 2

Gabriel, 300129767

Gianluca Coletti, 300065278

Hiruni Senarath, 300044216

Thuy-Vi Le, 300119477

Tony Zhao, 300129784

Due: November 5th, 2020

University of Ottawa

Table of Contents

Introduction	1
Client meeting notes	1
Prototype	2
Testing	2
Table 1: Target Specifications	4
Conclusion	4

Introduction

This stage will see the results of our third official meeting with the clients and based on those results in tandem with the previous deliverables, we will create another prototype which will test more important aspects of the project. Similar to the last prototyping deliverable, this will see a classification of the prototype as well as the analysis, testing, and test plan of that prototype. Finally, we will update the project plan according to our progress thus far to ensure we are on schedule.

Client meeting notes

- Our client relies heavily on Apple voiceover so compatibility with it would be a great asset.
- No need for initiation sequence/button to start, client would prefer if it started finding buttons automatically after opening the app.
- Elevator buttons are so close together that the user would touch it anyways to differentiate between the different buttons. Thus an app that tells just the general location of the panel is good enough and then users can find the specific button on their own.
- Leave the app running even after the user finds the button in case they want to relocate the button. App will only turn off once the user hits the 'home' button or uses Siri shortcut to close the application.
- Seeing AI (flip to product mode) is a good example of how instructions should be relayed.

Prototype

This prototype is a Focused, medium Fidelity, Analytical prototype that we used to test the app's ability to recognize a button on the screen. This is the fundamental process for our design as it is the means by which our product will find buttons before it is able to guide users to them. It verifies our most critical assumption, that the app can locate buttons when they appear on screen effectively enough for what we intend. We created this prototype using Apple's Create ML app. Create ML allowed us to train a model by applying a machine learning algorithm to set a training data so that the model could detect elevator buttons.

Testing

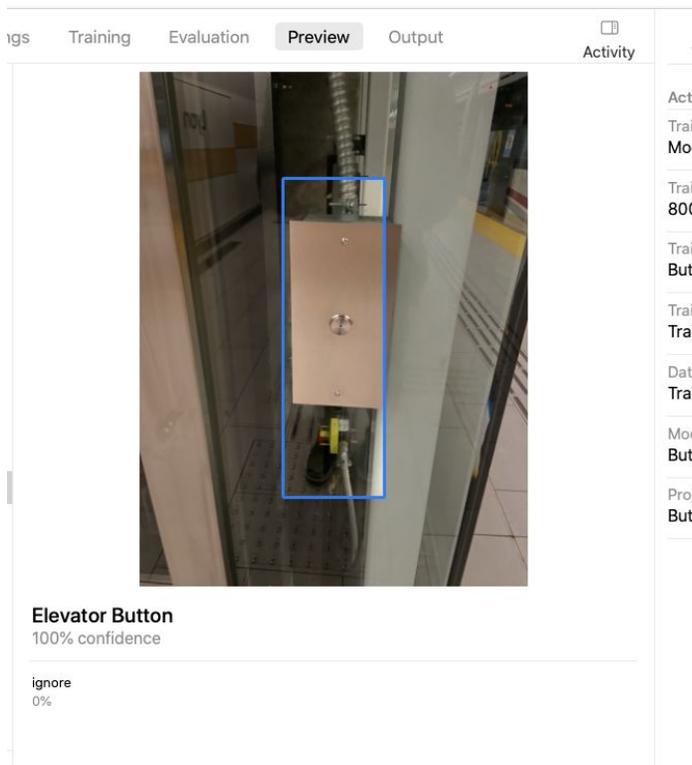


Figure 1. LRT elevator button recognized by Core ML model.

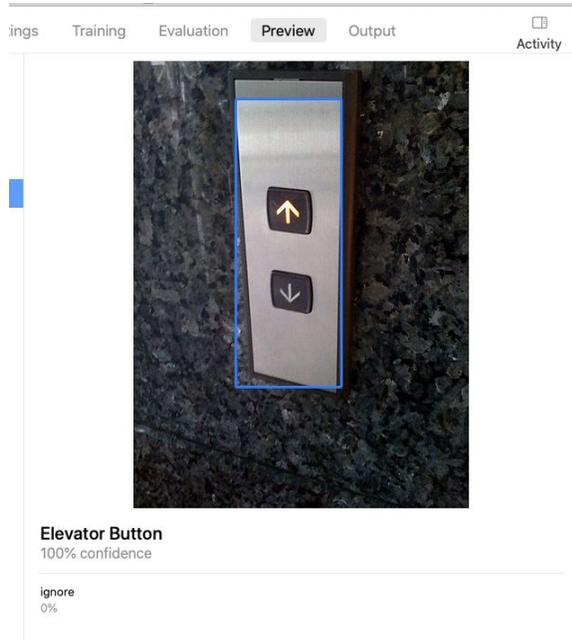


Figure 2. Image of elevator button from Google images recognized by Core ML model.

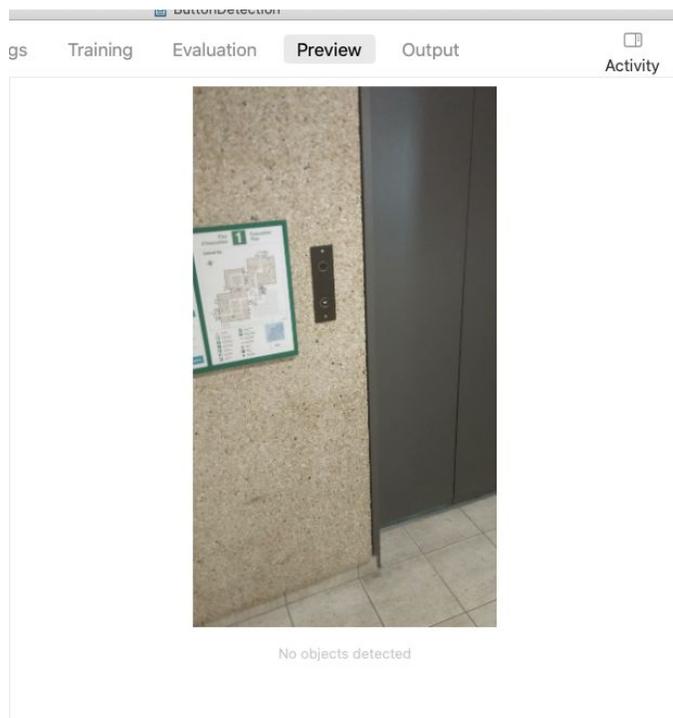


Figure 3. Image of elevator button not recognized by Core ML model.

Findings from Testing

Table 1: Target Specifications

Target specification	Unit	Expected (marginal)	Received Values
Cost	\$	≤ 100	0
Usability	1- Very user friendly 2- Moderately user friendly 3- Not user friendly	1	2
Phone integration	# of platforms	1	1
Time to notify user	Time delay (s)	$\leq 15s$	
# of notification systems	# of ways to notify	≥ 1	1
Route planning feature	Y/N	N	N

When testing out the model, it was able to accurately identify and place boxes around elevator buttons that looked very similar to those of our testing data as seen in Figure 1 and 2. Unfortunately when it comes to any buttons that's visual format looks very different from the training set or the image is taken too far away from the button to detect key features of the button, the model failed as seen in Figure 3.

Conclusion

This deliverable saw the creation of our second prototype based on the information provided both in previous stages of development and in the client statements from the third client meet. Going forward, the information received from this prototype will be instrumental in the creation of our final prototype and project because it lays the foundation for the most important task our app needs to accomplish, namely being able to recognize that there is a button on-screen in order for it to be able to guide a user to it.