

Computer Science Capstone:

‘Laser Focus’ Designs Convolutional Neural Network Model for Traffic Lightbox Image Recognition

Western Governors University

William Tucker - 001146698

Table of Contents

| | |
|---|----|
| Letter of Transmittal | 4 |
| A: Project Recommendation | 5 |
| Problem Summary | 5 |
| Application Benefits | 5 |
| Application Description | 6 |
| Data Description | 6 |
| Objectives and Hypothesis..... | 6 |
| Methodology | 7 |
| Funding Requirements | 8 |
| Data Precautions | 9 |
| Developer's Expertise | 9 |
| B: Project Proposal | 9 |
| Problem Statement | 9 |
| Customer Summary | 10 |
| Existing System Analysis..... | 10 |
| Data | 10 |
| Project Methodology..... | 11 |
| Project Outcomes | 11 |
| Implementation Plan | 12 |
| Evaluation Plan | 12 |
| Resources and Costs | 12 |
| Timeline and Milestones..... | 13 |
| C: Application..... | 13 |
| Visualizations..... | 13 |
| Descriptive Method..... | 13 |
| Non-Descriptive Method..... | 14 |
| Machine Learning Application | 15 |
| Interactive Dashboard | 15 |
| Security | 15 |
| D: Post-implementation Report | 16 |
| A Business (or Organization) Vision | 16 |
| Datasets | 16 |
| Data Product Code | 17 |
| Objective (or Hypothesis) Verification..... | 17 |

| | |
|---|----|
| Effective Visualization and Reporting | 18 |
| Accuracy Analysis | 18 |
| Application Testing..... | 19 |
| Application Files..... | 19 |
| User Guide | 20 |
| Summation of Learning Experience | 21 |

Letter of Transmittal



Subject: Letter of Transmittal – Laser Focus to develop Traffic Lightbox color recognition

May 16th, 2022
Mr. Matthew Richardson
Laser Labs
9200 N. Audubon Dr.
San Francisco, CA 94110

Mr. Richardson,

There has been a growing interest in detecting surrounding obstacles around vehicles within the car industry. This started with rearview mirrors and windshields. Over time and with technological advancement, this notion has matured to become mounted cameras, computers, and software using object recognition to detect and identify these obstacles. Some cars can even drive themselves in part using this technology. Fesla Motors has engaged with us and requested that we develop software that will help them display whether traffic lightboxes on the horizon indicate whether the driver should stop or go.

The Machine Learning team and I have evaluated the requirements and details of this engagement and have a recommended solution. Fesla Motors already possess a technology that permits them to take periodic snapshots of the horizon, detect traffic lightboxes and save annotations of all of them in a file. That is where we come in! We will need to parse this information, crop the traffic lightbox(es) from the image(s), and run it through our model. Our model will predict the indication on the traffic lightbox. Once this is complete, we will deliver an output containing the probability of the required action. Fesla Motors will take it from there and display the prediction on the dashboard for the driver.

This solution will benefit our organization as we will build a relationship with Fesla Motors, can capitalize on our investment, implement our technology for other vehicle manufacturers, and be known in the industry for creating machine learning solutions for complex real-world problems.

We estimate that the cost of developing this solution will be no more than \$35,070, including operating costs that were already assumed. We stand a lot to gain from this deal.

I want to emphasize that I am the right person for this job. With my experience and education in computer science, I can assure you that you will not regret entrusting this responsibility to me.

Sincerely,

William Tucker

Machine Learning Engineer

A: Project Recommendation

Problem Summary

There has been a growing interest in detecting surrounding obstacles around vehicles within the car industry. This started with rearview mirrors and windshields. Over time and with technological advancement, this notion has matured to become mounted cameras, computers, and software using object recognition to detect and identify these obstacles. Some cars can even drive themselves in part using this technology. Fesla Motors has engaged with us and requested that we develop software that will help them display whether traffic lightboxes on the horizon indicate whether the driver should stop or go.

This project fits well within our organizational needs. Our company is growing, expanding, and becoming more profitable every day. This is due to our ability to adapt and exceed expectations in developing software for complex, real-world problems. We have an opportunity to create a solution and take part in history.

The Machine Learning team and I have evaluated the requirements and details of this engagement and have a recommended solution. Fesla Motors already possess a technology that permits them to take periodic snapshots of the horizon, detect traffic lightboxes and save annotations of all of them in a file. That is where we come in! We will need to parse this information, crop the traffic lightbox(es) from the image(s), and run it through our model. Our model will predict the indication on the traffic lightbox. Once this is complete, we will deliver an output containing the probability of the required action. Fesla Motors will take it from there and display the prediction on the dashboard for the driver.

Application Benefits

This solution will benefit our organization as we will build a relationship with Fesla Motors, can capitalize on our investment, implement our technology for other vehicle manufacturers, and be known in the industry for creating machine learning solutions for complex real-world problems.

With Fesla Motors in our corner, we may be able to develop more solutions for them. This opportunity could be the gift that keeps giving. There are many different, similar projects they may propose to us, and there are many other opportunities to revolutionize the industry. The main idea of this point is that we could be at the forefront of a vast opportunity and change the game.

Fesla Motors has committed to paying \$500,000 for this solution. If we come through for them and create a great product, we could patent and license this software and sell it to many other vehicle manufacturers. Financially, this is a fantastic opportunity as it will only cost a fraction of that to develop the solution.

My final point is more about our image and perception. Our company could be known in the industry for our achievements with this project. We would be known for our ability to solve complex, real-world problems. We would cement our place in the industry and, even more so, become a brand that everyone trusts.

Application Description

This project will utilize many different technologies to achieve the overall goal of the application. In building this solution, I will utilize Python and Jupyter Notebook and many Python libraries such as Tensorflow, Keras, sklearn, pandas, NumPy, etc. Using Tensorflow, we will implement a convolutional neural network (CNN) model that, once trained, will ultimately predict whether the provided image of a traffic lightbox has an illuminated red, green, or yellow light. The input will be a cropped image of an entire traffic lightbox. The output will be an array with three elements. Each element is a float type. The first is the predicted likelihood that the light is red, the second is green, and the third is yellow.

Data Description

The origin of the data used to develop this solution can be found on Kaggle.com and is called the “LISA Traffic Light Dataset.” This is categorical data and will be used categorically to train our model and determine whether a traffic lightbox has a red, green, or yellow illuminated light.

The structure of the data is fantastic and fits our needs perfectly. It is a series of frames from videos where a driver is navigating through some streets in San Diego. The scene groups each frame it was taken from. Additionally, a file contains annotations for each traffic lightbox and light bulb. We will use the traffic light box annotations for use in this project. There is a single annotation for each object found within each image, meaning if there are five traffic lights in a single shot, there will be five-line items for each annotation per frame.

Objectives and Hypothesis

The primary outcome of this project is to produce a solid solution for the customer. More specifically, that solution is a model that accepts an appropriately cropped and sized image of a traffic lightbox and outputs an array where both elements indicate the model’s prediction of whether the driver needs to stop or go. Ideally, the model’s accuracy would

reach an overall accuracy of 85%, but room for error cannot be allowed at red lights. We must be sure to deliver a product that has at least 85% accuracy for red lights.

Methodology

This development will follow the SEMMA methodology. This methodology is best known as a methodology surrounding functional data mining but is sufficient for this project. There are five steps to the process. This section will go over the steps this project is planned to take.

- **Sample:** We will begin by preparing as much data as possible. This data will be images of the horizon during an average drive through town. Once we have acquired this data, we will categorize it into preparation type data and validation type data. Luckily the dataset we've identified has already done the heavy lifting for us.

- **Explore:** In this step, we will analyze the data. This analysis will go through the images and identify how many there are, how many annotations there are, and how many of each type. Additionally, we can verify that all photos are acceptable quality and size. This exploration phase is critical in ensuring that our dataset is appropriate for our needs.

- **Modify:** Once we have learned all we can from the explore step, we will clean and transform the data to better fit our use case. This step will be exhaustive because it is our opportunity to make the data as conducive as possible for the overall goal.

- **Model:** Modeling the data involves processing the data in a format able to be used by our implementation for our product. At this point in the process, the model should be operational so that we may begin measuring success and testing.

- **Assess:** Finally, we will check the overall usability and reliability of the data. The question that needs to be answered from this step is, "does this data work for our use case?"

Sprint Schedule

| Sprint | Start | End | Tasks |
|--------|--------|--------|---|
| 1 | Week 1 | Week 1 | Planning <ul style="list-style-type: none"> • Create a project plan, charter, scope, etc... |

| | | | |
|---|--------|-------------|--|
| | | | <ul style="list-style-type: none"> Coding environment |
| 2 | Week 2 | Week 4 | Data Acquisition & Programming <ul style="list-style-type: none"> Acquire Dataset Create CNN Model, create surrounding data structures & algorithms |
| 3 | Week 5 | Week 6 | Exploration, Data Cleanup, Modeling <ul style="list-style-type: none"> Categorize data, clean data Create a model and exhaustive training Debugging Optimization Verify, rinse & repeat if necessary |
| 4 | Week 7 | Week 8 ∞ | Deploy <ul style="list-style-type: none"> Provide model to select users <ul style="list-style-type: none"> Quality assurance Develop update/upgrade process for update/retrained models |

Funding Requirements

| Resource | Description | Cost |
|-----------------------------|-----------------------------------|---------------------------|
| One time purchase | ML Computer | \$10,000 |
| 250 hours (2-month project) | ML Engineer Salary 150k/y | \$25,000 |
| Ongoing | Amazon AWS Testing Server for UAT | \$35/month |
| | | Total: \$35,070 |

Data Precautions

Luckily, the data used to implement this project was found on Kaggle.com. This dataset can be used freely so long as you agree to the terms and conditions of Kaggle.com and abide by their acceptable use policy.

Once our project has been implemented, our code will be placed within the hands of Fesla Motors such that they can utilize it in their vehicles. After the code finds its way into the cars, even if there was some data privacy concern, that responsibility falls to Fesla Motors to operate within the different types of federal and local laws relating to it.

Developer's Expertise

I want to emphasize that I am the right developer for this job. Soon I will be graduating with a bachelor's degree in Computer Science from Western Governors University. In my experience during the program, I have been exposed to and had to complete projects that involved machine learning and artificial intelligence. This project directly involves machine learning using convolutional neural networks and Python, both of which I now have experience with. I am not just requesting that I be the developer tasked with this project because I want to; I know that I possess the skills and knowledge to get the job done right the first time! Additionally, I have enjoyed learning from incredibly skilled course instructors and mentors who have helped me during my time there. I have achieved two certifications through the program, the CompTIA Project + and ITIL v4 Foundations certifications. With the industry knowledge I accrued, I was even able to earn the newest CompTIA Security+ certification. In addition to my Microsoft MTA Server Administration certification and other certifications in Ivanti and Epic software, I am a very well-rounded computer scientist with current industry knowledge. To add to that, I intend to become certified in Microsoft Azure and Tensorflow. Let there be no doubt that I am well qualified, equipped, and ready to take on this project.

B: Project Proposal

Problem Statement

Fesla Motors has engaged our company Laser Focus to create a machine learning-based solution to produce a model that will predict whether a traffic lightbox indicates that the driver should go or stop. The product we deliver needs to have at least 85% accuracy, most notably for stops. We will provide a model which will allow a properly formatted image to be passed to it, and once it has established a prediction, it will output that prediction in the form of an array. This array will contain two elements, one to indicate the probability of the driver's need to go or their need to stop.

Customer Summary

Fesla Motors is a global, multibillion-dollar company that makes top-of-the-line and revolutionary battery-powered vehicles. They have a stranglehold on the market and do not appear to have any end in sight. As they continue to grow and remain leaders in the industry, they have opted to engage with us to develop a solution for them.

With the proper conditions, our proposed application will quickly deliver a successful product that will exceed their expectations. It is well understood that machine learning can provide accurate, insightful, and nearly instant results if trained well enough. We will harness the power of machine learning while using libraries written by industry leaders to create this model. There is no doubt that our product will solve their problem and keep them coming back for more.

Existing System Analysis

Fesla Motors uses similar tools that we do for some of their existing machine learning applications. Those tools are Python, sklearn, and Tensorflow. They have been wildly successful in using these tools in the use cases. I do not believe there are any shortcomings in their devices or workforce. However, I think they have opted to work with us to offload some of their workloads. Additionally, it seems clear that they will likely be returning customers if we do well.

Data

The data used in this project has been found on Kaggle.com. The dataset is called “LISA Traffic Light Dataset.” There are two kinds of data found in the data set: various images and some comma-separated value files. The data is organized very well within a series of folders. The folders separate the data by different attributes such as day or night, scenes, and frame folders. Each has a corresponding folder with the comma-separated value file containing annotations. There is a line item for each traffic lightbox or bulb instance within these annotation files. There is often more than one line item per image, as there is often more than one lightbox in each frame. This data can be processed using simple code to navigate this folder structure, iterate through the images, and match the corresponding annotation(s) to the frame. With some loops, you can successfully identify and process each annotation for every frame in the data set. If anomalies, outliers, or incomplete data are found, they can be located and removed manually from the data set. Should I identify something that may cause problems, depending on the situation, I can code around it, remove it, or it will be skipped entirely due to the simplicity of Python and loops.

Project Methodology

Because I will be the only one working on this project, and it is a short two-month project, it doesn't need to be as formal as it may need to be when utilizing a vast network of resources and a high level of costs. Although I have created a waterfall-style project plan outline below in the Timeline and Milestones section, the SEMMA approach as I lined out in section A is sufficient. This methodology is best for and most famous for its use in data science. This approach has five phases: sample, explore, modify, model, and assess. I will prepare as much data as possible in the sample phase because having more than you need is better than not having enough. There almost is no such thing as having too much data. The explore phase is where I will look around the data and become more familiar with it. Once I learn the ins and outs of the data, including its good qualities, I can also identify its shortcomings. In the modify phase, I can use what I learned in the explore phase to prepare the data conducive to working with and suits the model phase well. Next is the model phase, and that is when I will process the data in a suitable structure for training. This is also when I may begin to measure success, general testing, and testing accuracy. It is also good to mention that I may need to step back into one of the other steps or rinse and repeat to benefit my model phase. As usual, this depends on the circumstances. Finally, the assessment phase is used to verify all usability and reliability of the data and final product. Generally, this is where you must identify and confirm that this final product solves the problem.

Project Outcomes

The expected project outcomes are as follows:

- Implement a Convolutional Neural Network Model
- Train the Model to identify prepared images of traffic lightboxes
- The model must have an 85% or higher level of accuracy in predicting whether the image depicts a lightbox with a green light for go or red light for stop
- Create a web application that allows users to test the model using provided and uploaded images.
 - The web application must be easy to use and indicate the prediction.

Implementation Plan

My implementation plan is a simple and general strategy. As described in the Project Methodology section above, I will use the SEMMA methodology. This methodology is best known for its uses in data science. I like it because it is not over convoluted and allows for the project's work to take precedence rather than being hung up on the project methodology. The SEMMA methodology has given phases: sample, explore, modify, model, and assess. These are the phases that I will use to approach this project. Each stage in the project will be dependent on the prior. However, the flexibility with SEMMA permits revisiting previous steps like that will likely be necessary.

Evaluation Plan

In every project like this, there are stages of development, and for you to move on to the next or say that you've completed it, you'll need to have a way to gauge and verify it. Depending on the stage or project activity, it may need to be treated differently or tested. Ultimately, the decision to consider the stage of development complete, I must verify that it is satisfied by using some sort of method. For example, when my model is completed, I will run it through a testing phase and measure its model's accuracy. If the model's accuracy is at or above the 85% accuracy requirement I have set out for myself, I will consider it done. I would like to emphasize that there will likely be a unique method to verify its successful completion, depending on the stage.

I have a few criteria for verifying that the project has been completed. The most important thing to me is that the customer is happy, and all their requirements have been met. If the result does not meet all their requirements, they will likely not be satisfied. Additionally, the project needs to be completed on time. Towards the end of the project, around the time I am considering myself close, I will make sure that all the requirements for the project have been met and that they have been met within the time frame discussed. If for some reason, one or more of the requirements cannot be completed in time or at all, I will gather all the details surrounding that and let the customer be aware of this as soon as possible. This will earn a sense of trust with the customer and possibly result in an extension of the project or a follow-up project for the remaining requirements or features.

Resources and Costs

As discussed in the Letter of Transmittal and the Funding Requirements section in section A, this project will have minimal costs that are already assumed. A total of \$37,070 price tag will be observed. Our machine learning computer has

already been purchased. However, that should be noted here. My salary will be an assumed cost and entirely devoted to this project during the project's two-month duration. Lastly, there is a small fee for a user acceptance testing server in the cloud. There is a breakdown of the costs in the section mentioned above.

Timeline and Milestones

| Milestone | Activity | Start | End |
|-----------|---------------------------------------|--------|----------|
| 1 | Project Plan, Charter, Scope | Week 1 | Week 1 |
| 1 | Project Approval | Week 1 | Week 1 |
| 2 | Data Acquisition | Week 1 | Week 2 |
| 2 | Create CNN Model | Week 2 | Week 4 |
| 2 | Create Data Structures, Algorithms | Week 2 | Week 4 |
| 3 | Data Exploration | Week 5 | Week 6 |
| 3 | Data Cleanup | Week 5 | Week 6 |
| 3 | Model Training, Verification | Week 5 | Week 6 |
| 3 | Model Testing | Week 5 | Week 6 |
| 3 | Optimization | Week 5 | Week 6 |
| 4 | Develop a Web App For UAT | Week 6 | Week 6 |
| 4 | Release for UAT | Week 7 | Week 8 |
| 4 | Develop Upgrade/Update for new models | Week 7 | Week 8 |
| | Deliver to customer | Week 8 | Complete |

C: Application

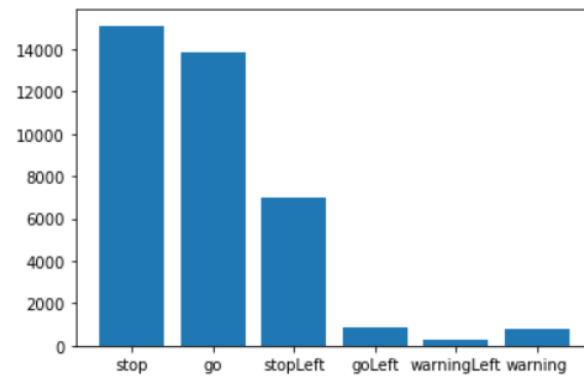
Visualizations

I have prepared three visualizations to be shown within this document. However, there are more than can be viewed within my Jupyter Notebook.

Descriptive Method

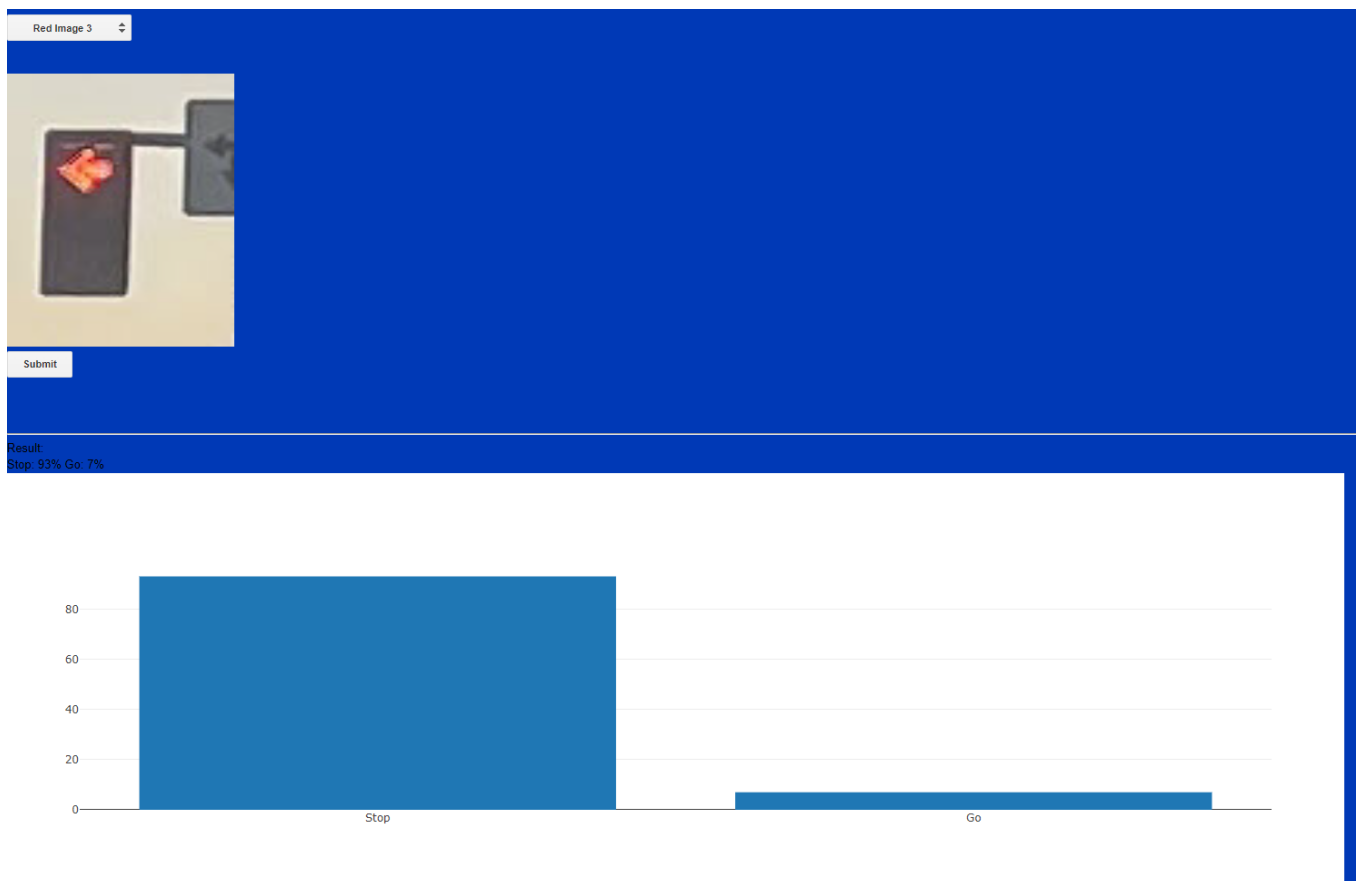
This image was created using a library called 'matplotlib.pyplot.' This bar graph showed me how many annotations were found within my dataset.

Total annotations found by category:

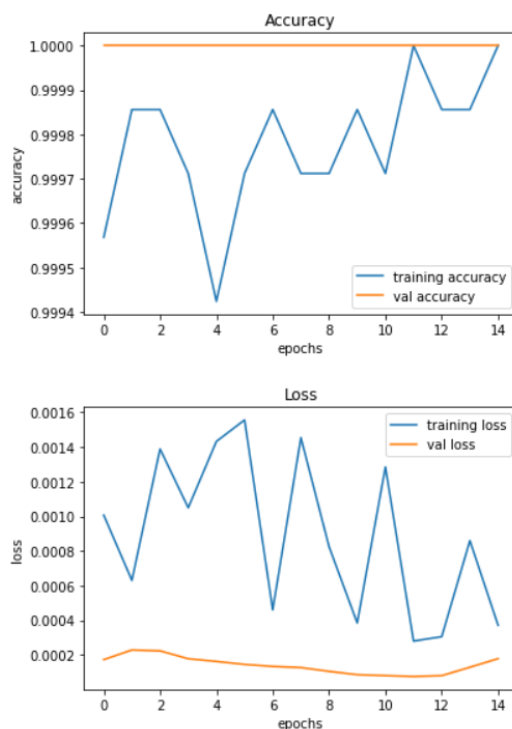


Non-Descriptive Method

The web application I have created utilizes a JavaScript-based library called 'Plotly' to display the prediction obtained from the model. You may use this to take a screenshot, zoom in, pan, and several other features.



This is another image using 'matplotlib.pyplot'. The data found here infer the accuracy and loss of my model's training and validation.



Machine Learning Application

My project employs machine learning in the form of image recognition. This type of machine learning is called deep learning due to a neural network with many different layers. My model was trained so that now it has learned what a properly prepared image of a traffic lightbox may indicate to a driver.

Interactive Dashboard

My project comes with two different ways to interact with it. My Jupyter Notebook and the web application I've created are these two ways.

Please do not try to run any of the code in the Jupyter Notebook. I only include it to showcase some of my hard work and have some valuable images and data I found along the way.

My web application can be reached at: <http://ec2-54-200-119-18.us-west-2.compute.amazonaws.com>.

Security

Luckily there are no security concerns associated with this project.

D: Post-implementation Report

A Business (or Organization) Vision

Fesla Motors sought to obtain a trained model that they could use in their vehicles to help with a new feature they desired to offer to their customer base. This model was to receive an input of a prepared traffic lightbox image. Finally, it was to predict whether that image indicates stop or go and provide this prediction and output. The main product is a trained model that will predict whether a driver should stop or go based on an image input.

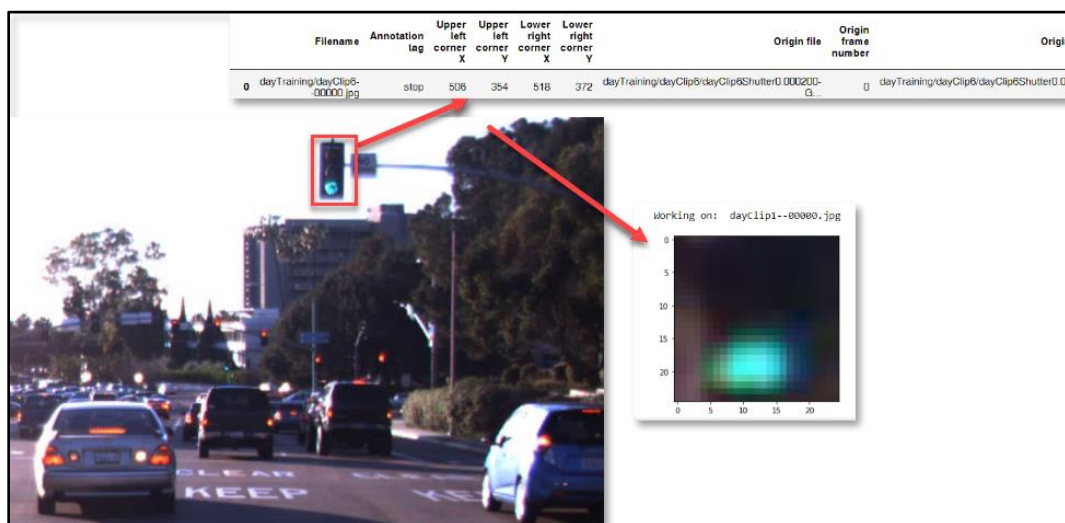
My application solves this problem because I have created a model with greater than 85% accuracy for identifying scenarios where the image provided indicates that a stop or a go is appropriate. It can receive an input image and predict whether the image is indicative of needing to stop or go based on its extensive training.

The model is deployed in a web application so that anyone can access it within their web browser. My CNN Model file is uploaded to the server and utilizes TensorFlow's JavaScript library to pass images to it and obtain an output. This output is used to output the prediction in text and in the form of a bar graph.

Datasets

The dataset that I used for this project was perfect. I couldn't have asked for better data; besides, I wish there were more yellow lights.

I accomplished this by iterating through all the large images, referring to the line items within the annotation files to crop the traffic lightbox out of the image and reshape it. These were the images that I used to train the model.



This dataset can be found here: <https://www.kaggle.com/datasets/mbornoe/lisa-traffic-light-dataset>.

Data Product Code

As a review of the code used to analyze and develop my application, most of this was achieved using Python and libraries for Python. For processing the raw images, I created a function that would iterate through all of the photos, and using the name of the file, I referred to the corresponding annotation file. Once I obtained the annotations, I used them to crop the traffic lightbox out of the image. Often there was more than one traffic lightbox, so there was consequently more than one image created from most of the source images. In the visualization section above, I referred to a descriptive method image that helped me determine how many annotations were found in the dataset. I did not use non-descriptive methods for my data analysis or exploration phases.

More specifically, the places where I did use non-descriptive methods; for my first example, I used Plotly to indicate the success of the prediction. My model provides an array as an output. The collection has two elements. The first element contains the probability that the image indicates stop, and the second is the probability that the image suggests go. This is an appropriate means to display the prediction such that you can gauge the accuracy with ease. Additionally, Plotly has features that can make the data come alive as you can zoom in, drag it and export the data. The second is a plot graph showing the accuracy of the training and validation of the model. This was incredibly useful when I was training my model because, without it, I would have been lost as to how accurate the model was over time.

Objective (or Hypothesis) Verification

My Hypothesis was as follows: The use of enough images will be able to train a model to achieve 85% accuracy in predicting whether a traffic lightbox is indicating stop or go. I was not surprised that my hypothesis was met and ended up exceeding the 85% accuracy requirement, but I was shocked that I could create a model with that level of accuracy. This hypothesis was met and confirmed because after a lot of hard work and retraining my model; I was able to verify my model is accurate. I tested a lot of different images that the model had never seen before, and more often than not, it would predict the image accurately. Additionally, even ideas that were not perfectly matching the photos used to train the model were predicted accurately.

Effective Visualization and Reporting

My descriptive method and first visualization were helpful in the data exploration phase as I could use that data to determine how many cropped images I needed. For example, when I first trained my model, I used all the pictures of the data set, but since there were so few yellow images, it made the entire model inaccurate, even for red and green light images. So, I quickly decided to drop the yellow pictures and use only an even amount of red and green light images.

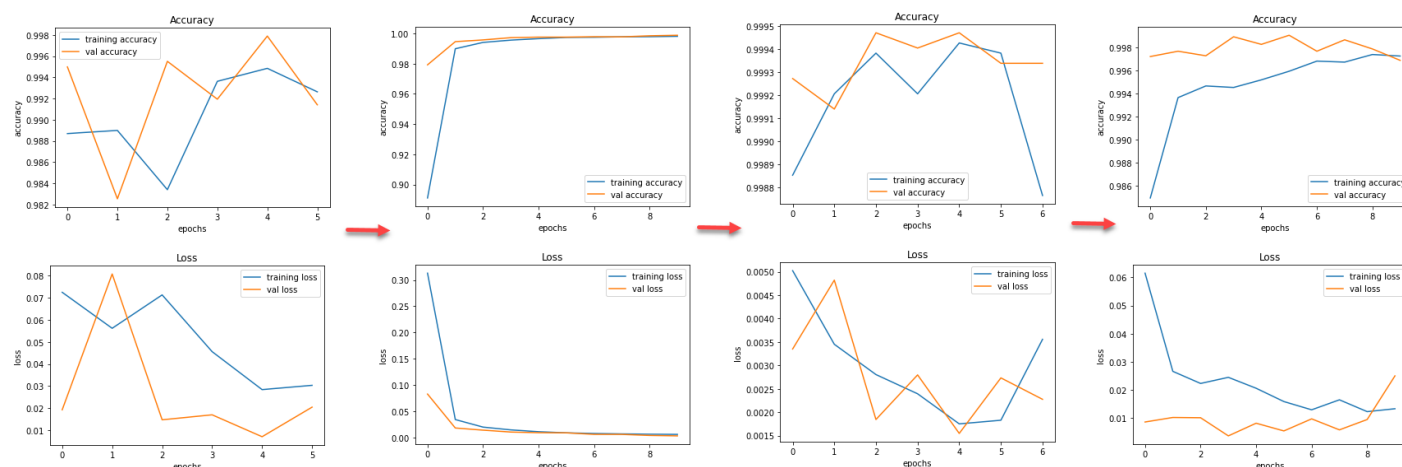
My second visualization and first non-descriptive method were best utilized in the data summary. Plotly eloquently displayed the result of the prediction provided by the model. I feel that this made my web application come together. I also should mention again that Plotly is incredibly useful for displaying data organized and beautifully.

The final visualization and second non-descriptive method were helpful in my data analysis for my model's accuracy. I used the data provided in this plot graph to retrain my model in different ways repeatedly. It helped me get my model to where it is now, and I feel it was the most critical visualization I had at my disposal. It was used when I determined that validating my model with (25x25) sized images was best, training with a 1/3 validation set, and making adjustments to the layers of my model.

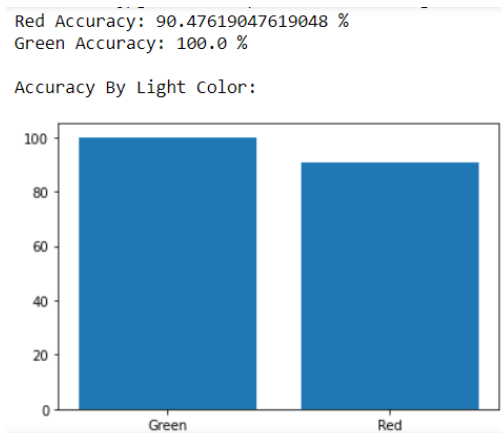
Accuracy Analysis

I used several methods to verify the accuracy of my model. The first method was to use the history provided from training the model. Tensorflow delivers an output of the level of accuracy of validation and training while training the model.

Using this data, you can create a plot of the accuracy. This is one of the non-descriptive methods I provided above. Below is an image representation of several times I retrained my model to improve its accuracy.



Finally, I used a testing method to verify the model's accuracy using authentic images that the model had never seen before. I provided the model with pictures it had never seen before and scripted an output of the accuracy based on the prediction and how many it could predict accurately.



Application Testing

My application was built and tested on Windows 11 (21H2) using Google Chrome - Version 101.0.4951.67 (Official Build) (64-bit) & Opera GX - LVL3 (core: 85.0.4341.79).

I tested my web application by iteratively building, saving, and testing the feature I had just created. Once I was completed, I reset my browser to clear my cache, have a clean environment, and verify that all parts were functional. When it came to my web application, it was straightforward as all I needed to do was select each image and press submit and test uploading a photo and submitting that.

There were several iterations of my web application. My web application started very simple and not styled at all. It only allowed the user to upload an image and outputted only text. It also did not have pictures pre-uploaded to the server for the user to select from. Once the base feature of running an image through the model was complete, I added the combo box and all the photos uploaded to the server. I then found the Plotly library and decided to add that to have a more immersive experience with the web application. I finished by styling the page, so it looked good.

Application Files

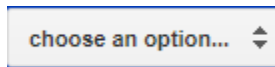
Two files are being submitted for my project.

- [C964.pdf](#)

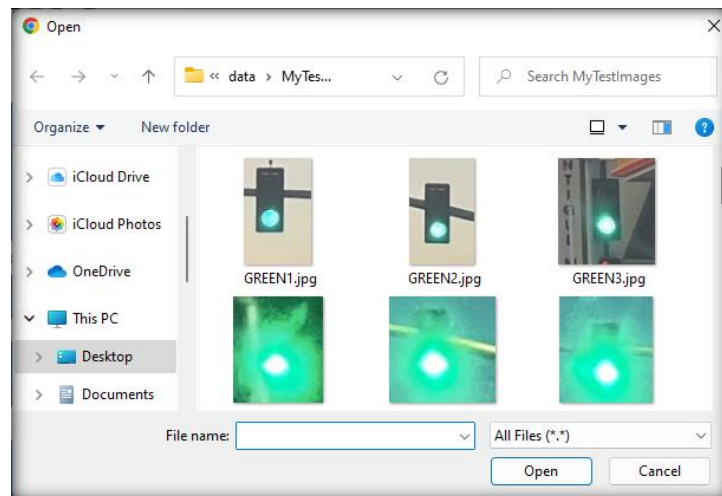
- This file is the document you are reading.
- *C964JN.ipynb*
 - This file is my Jupyter Notebook. Please do not try to run it or interact with it at all. It is only attached to the project to show my hard work and offer more images and data.

User Guide

1. Go to <http://ec2-54-200-119-18.us-west-2.compute.amazonaws.com> by clicking this link or pasting it into your browser's address bar.
 - a. A If you encounter any security-related messages from your browser, you can use the advanced option to permit your browser to open the page. There are no security issues with this web application.
2. Using the 'choose an option' combo box, select an option.



- a. A A If you choose 'Select my image,' a button that says 'choose file' will be exposed. Click it.
 - i. Using the Windows File Explorer menu, Choose an image from your computer.

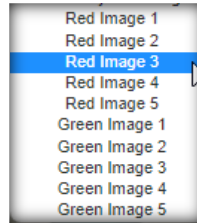


1. **Note:** the model will work best if you have a cropped picture of a traffic lightbox.

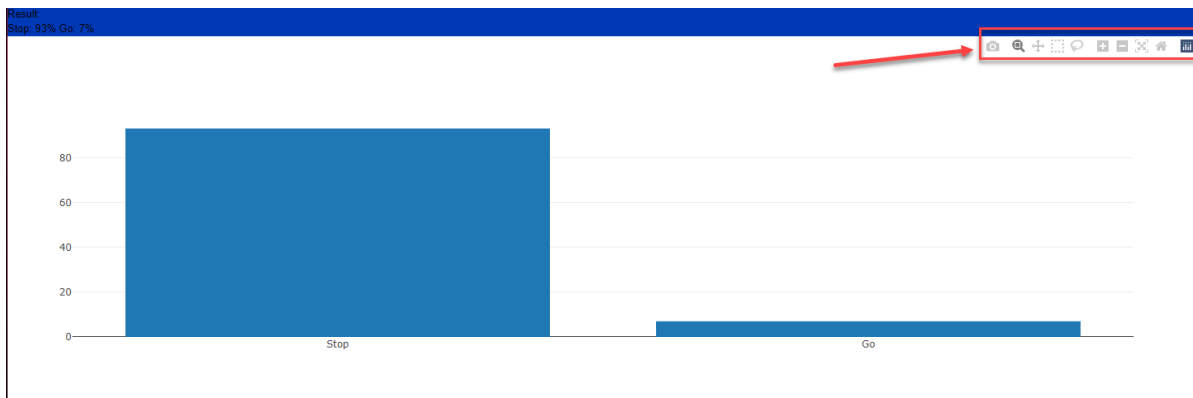
Finding one online or screenshot one from Google Maps may be best.

- ii. Click 'Open.'

- iii. Your image will be uploaded to the server. You can change the image at any time by repeating the last two steps.
- b. You may choose any of the ten provided images uploaded to the server.



- i. The image you selected will be displayed on the web page.
3. Click the 'Submit' button. This will run the image through the model.
-
4. The results will be displayed at the bottom of the page and a representative bar graph.
- a. A You may use the feature of Plotly to analyze the data.



Summation of Learning Experience

I feel that I was well prepared for this project because of my experience at Western Governors University. I became very comfortable with Python from one of the courses. Without that experience, this project would have been much more difficult. The help I received from my course instructors and course mentor was paramount to my success in this program. My familiarity with programming and computer science and applying those skills to real-world issues has prepared me for this project.

I learned a lot of what I needed to know but didn't already know from practicing machine learning exercises found on Kaggle.com. I took code from Kaggle and played with it until I understood how to make models, train them, and handle their input and output. Additionally, I familiarized myself with the libraries I was using by reading the documentation for those libraries online.

The experience I am taking away from this course because nothing is too difficult to jump in and get started. When I first began considering what project I would do for this course, I didn't know if I could handle creating a CNN. Now I know that I can, and I can do amazing things with it. It has even got me thinking that I may want to pursue a career in machine learning. I enjoyed learning about CNN and do not want to stop here. Overall, this has been a motivating process for me, and I am looking into getting a Tensorflow certification after I graduate.