Commuter Highway Analysis and Detection System

(CHADS)

C964 – Computer Science Capstone

Table of Contents

[A1: Letter of Transmittal 3](#_Toc81418240)

[A2. Project Proposal 4](#_Toc81418241)

[Problem Summary 4](#_Toc81418242)

[Data Product Benefit 4](#_Toc81418243)

[Data Product Outline 4](#_Toc81418244)

[Data Description 5](#_Toc81418245)

[Objective and Hypotheses 5](#_Toc81418246)

[Project Methodology 6](#_Toc81418247)

[Project Outcomes 6](#_Toc81418248)

[Implementation Plan 7](#_Toc81418249)

[Evaluation Plan 7](#_Toc81418250)

[Resource and Costs 7](#_Toc81418251)

[Programming Environment 7](#_Toc81418252)

[Environment Costs 8](#_Toc81418253)

[Human Resource Requirements 8](#_Toc81418254)

[Timeline and Milestones 8](#_Toc81418255)

[C. Project Attributes 9](#_Toc81418256)

[Data Methods 9](#_Toc81418257)

[Datasets 9](#_Toc81418258)

[Data Cleaning 9](#_Toc81418259)

[Data Visualization 10](#_Toc81418260)

[Real-Time Queries 10](#_Toc81418261)

[Adaptive Element 10](#_Toc81418262)

[Outcome Accuracy 10](#_Toc81418263)

[Security Measures 10](#_Toc81418264)

[Product Health Monitoring 10](#_Toc81418265)

[Dashboard 10](#_Toc81418266)

# A1: Letter of Transmittal

Morgan Webber

Software Lead

Ocelot Labs

1234 56th Ave

Portland, OR 97013

Dear Oregon Department of Transportation,

The Oregon Department of Transportation (ODOT) is responsible for managing the function of the public transport infrastructure. This is a multi-faceted responsibility; however, it largely covers managing public highways, freeways, and high-trafficked areas. A large portion of ODOT’s work involves managing zones of transportation that see a massive flux of traffic throughout the day. Thus, knowledge of the levels of traffic throughout the day is imperative to the success of ODOT. A system that automates this analysis would prove to be invaluable in the operations of the department, as it would be able to monitor traffic at multiple locations 24 hours a day, providing incredible insight into the traffic tendencies of the state. This information can be used by ODOT as it would see fit, supporting decisions such as construction planning, detour design, disaster management, or other contingency plans.

The Commuter Highway Analysis and Detection (CHAD) System is designed to analyze and classify highway traffic into severity categories. CHAD analyzes traffic feeds in real-time to determine the current flow of traffic through a monitored CCTV camera feed. CHAD will be trained on WSDOT highway data from 2004, which includes 254 video feeds from different times of the day, of various traffic levels, and with various weather conditions. Using this data, we plan on achieving a >75% accuracy in traffic severity detection. Ideally, our vision is to use CHAD in conjunction with ODOT’s existing CCTV monitoring tools to form a panopticon of highway surveillance for the department.

Funding for the project will be approximately $10,000. These funds will largely support development hours, while a portion of it will flow into research and development for the use of open-sourced AI and machine learning tools. Having spent nearly a decade in the development of AI, my leadership and expertise will ensure the success of this project and a future for ODOT’s surveillance teams.

Warm Regards,

Morgan Webber

Software Lead

# A2. Project Proposal

## Problem Summary

Traffic monitoring is a crucial part of ODOT’s responsibilities. The current state of traffic, traffic forecasting, and historical traffic data is the cornerstone of nearly all the decisions made by the department. Manual inspection of CCTV feeds introduces the human element to the data collection process, and with it comes error, time expenditures, and higher fiscal investments. Staff must be hired to perform the job and they must spend their precious time monitoring camera feeds and documenting traffic. This has worked for many years for the department, however as technology continues to advance, it becomes increasingly important to consider technological replacements for outdated processes.

## Data Product Benefit

Currently, the department of transportation uses deployable devices or manual CCTV reviewing to determine traffic levels. This method is costly, error-prone, and lacks breadth. The department has to manually visit each site or manually monitor each site they’d like to manage, causing a massive amount of overhead and costing valuable man-hours. Further, the data may not be aggregated into a central dashboard for easy use. This makes looking up historical, current, and future data very difficult.

Using the CHAD system, the existing CCTV cameras can be used to provide all the data necessary to let CHAD determine the traffic levels automatically, continuously, and accurately. CHAD will store the data in a central location and provide a dashboard for access to the data. Implementing this solution will remove the need for deployable traffic monitoring devices as well as manual monitoring of CCTV footage. This means greater visibility into traffic patterns, less errors in the documenting process, and more time available for staff to spend on higher priority jobs. Factoring in the initial costs of setting up CHAD, departments can expect to see returns on investment – in both financial and productivity forms –within a couple months of use.

## Data Product Outline

CHAD will rely heavily on TensorFlow for the AI development. Because TensorFlow is a Python API, the project will be written in Python. The base product will include a simple dashboard for loading CCTV footage, viewing loaded footage, and indicators to help classify the footage. Using the dashboard, users will be able to select CCTV “feeds” and will be able to watch the AI grade them in real time, as they play back on the dashboard.

The software could be easily adapted to pull in a live stream of data from a CCTV camera, however no footage is currently available, so details on this will be expanded on later in this document. Additionally, a historical representation of the data will be extrapolated later in the document as well.

## Data Description

The dataset acquired for this project was obtained from Kaggle.com and is from the Washington State Department of Transportation. It is from a single CCTV camera located above I-5 S. and 188th St in Washington, looking South. This footage was obtained from 08/05/2004 to 08/06/2004. The data for the videos contains the following attributes:

|  |  |
| --- | --- |
| Data | Type |
| Filename | String |
| Date | String |
| Timestamp | String |
| Direction | String |
| Day/Night | String |
| Weather | String |
| Start Frame | Integer |
| Number of Frames | Integer |
| Class | String |
| Notes | String |

The above data is represented by a CSV file. In addition to this CSV data, the filename field correlates to an AVI video file that contains the actual CCTV footage. All of the data in the CSV describes the related CCTV footage. A hard limitation of this dataset is that it only contains 254 videos, which is a small number for an AI to train on. However, we’ll make do with what we have, and the AI will be trained on this data set. We will likely use the same videos multiple times while we train. CHAD will be written to accept larger datasets to train on than this one.

## Objective and Hypotheses

The primary objective of the CHAD system is to accurately classify traffic into severity categories in real time. This will, in turn, aid the objective of organizations seeking to utilize CHAD internally. CHAD will hypothesize that a certain video feed containing vehicles on a highway will be a certain severity of traffic. For example, if one car passes every 5 seconds, CHAD might classify this as low traffic.

## Project Methodology

Granted the straightforward nature of the project, CHAD will be developed using the waterfall methodology. The waterfall methodology is a linear progression from one stage to the next, with each successive stage building on the accomplishments of the previous one. The stages of the waterfall method are as follows:

**Requirements**

The product requirements will be defined in this stage. Documents pertaining to the requirements will be developed and used throughout the rest of the project.

**Design**

The design stage focuses on the software architecture. This is where both high- and low-level design decisions will be made about the software. The goal in this stage is to fully flesh out the entirety of the software design so that during the development stage the focus can be on development and not design.

**Implementation**

In this stage the software product is developed. All of the software design documents from the previous stage are realized in the software product.

**Verification**

Software testing occurs during the verification stage. This is where the QA team comes in and executes thorough tests of the software, and the development team makes changes as needed to align with the requirements.

**Maintenance**

Finally, the maintenance stage is an on-going stage wherein the development team maintains the software and keeps it functional.

## Project Outcomes

The outcome of the project is a functional software product with a dashboard that will display the highway status of the viewed CCTV feed. In this prototype, we will not be displaying live CCTV feeds, but instead recordings of CCTV footage. Our deliverable will be a dashboard that will allow the user to select a CCTV recording to view it and the rating that CHAD assigns to it.

## Implementation Plan

The implementation of the project will be conducted over three phases: the pilot rollout, initial feedback adjustment, and the live push to the working environment. These stages will allow us to roll out the software in a controlled manner, test it in a production environment, then make the necessary changes to be able to roll it out to the rest of the client’s workspace.

The development stage is designed to be spent in close communication with the client. Throughout the development, testing, and review stages of the software lifecycle, we will be working with the client to insure as smooth of an implementation as possible. This means gathering information relevant to the client’s environment such as workstation operating systems, hardware, patch information, and network device information. Ideally, we will have an internal test environment setup that mirrors the client’s environment to further increase the success of the implementation.

## Evaluation Plan

After the product has been developed, we will enter the Verification stage of the lifecycle. This verification stage will include working directly with the client’s staff members to insure usability and alignment with the requirements. We will host an evaluation meeting at our location for the client’s staff to attend. At this meeting, we will have a mock environment that simulates the client’s own environment, but with CHAD installed and operating. We will train the client’s staff on how to use CHAD, then let them use the software themselves. If any faults are discovered or changes requested during this stage, they will be accounted for and implemented prior to the pilot rollout mentioned above.

The success of the software will be graded on accessibility, ease of use, clarify of information, and accuracy. We will ask the users to fill out a survey in which they grade the software on each of these metrics and provide constructive feedback on each if necessary.

## Resource and Costs

### Programming Environment

The software will be developed using Python 3.9.6. This does mean that the software is not compatible with Windows 7 and below, therefore we must run Windows 10 and higher. We will rely on Google TensorFlow as the AI API for the machine learning component. An AWS SQL server will be used to store the database used for the testing and learning of the AI. The total cost of the programming environment will be $0, as all the above-mentioned technologies are free.

### Environment Costs

The environment costs will be $0 due to the fact that there will be no further installation of CCTV cameras necessary and there is no hosting required to run the software. The software simply is installed on an existing workstation.

### Human Resource Requirements

The human resource requirements will amount to $80,000 for the development of the software. This cost is broken down in the following manner:

|  |  |  |
| --- | --- | --- |
| Phase | Hours | Cost |
| Development | 150 | $30,000 |
| Testing | 50 | $10,000 |
| Implementation | 100 | $20,000 |
| Maintenance | N/A | $20,000 |

\*Maintenance includes a contract with the developer to maintain the software

## Timeline and Milestones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Event | Start | End | Duration (Hours) | Dependencies | Resources Assigned |
| 1 | Requirement Meeting | 8/23/2021 | 8/23/2021 | 8 |  | Development Team, Stakeholders, Client representatives |
| 2 | Project Kickoff  (Phase 1 Start) | 8/24/2021 | 8/24/2021 | 2 | 1 | Development Team |
| 3 | Project Planning  (Phase 1 End) | 8/25/2021 | 8/25/2021 | 8 | 2 | Development Team, Stakeholders |
| 4 | Product Design  (Phase 2 Start) | 8/26/2021 | 8/27/2021 | 16 | 3 | Development Team |
| 5 | CCTV Analysis and Review | 8/30/2021 | 8/30/2021 | 2 | 4 | Development Team |
| 6 | CHAD Development | 8/31/2021 | 9/3/2021 | 32 | 4 | Development Team |
| 7 | Dashboard development | 9/6/2021 | 9/6/2021 | 8 | 4 | Development Team |
| 8 | Internal Testing  (Phase 2 End) | 9/7/2021 | 9/7/2021 | 8 | 1-7 | Development Team |
| 9 | Pilot deployment and implementation  (Phase 3 Start) | 9/8/2021 | 9/8/2021 | 8 | 8 | Development Team, Client representatives |
| 10 | Pilot deployment feedback and patching  (Phase 3 End) | 9/9/2021 | 9/9/2021 | 8 | 9 | Development Team, QA Team |
| 11 | Final Prep for Deployment and Implementation  (Phase 4 Start) | 9/10/2021 | 9/10/2021 | 8 | 10 | Development Team, Client reps |
| 12 | Deployment and Implementation | 9/13/2021 | 9/13/2021 | 8 | 11 | Development Team, Client Reps |
| 13 | Acceptance Tests | 9/14/2021 | 9/14/2021 | 8 | 12 | Development Team, Client Reps |
| 14 | Final Implementation Tasks  (Phase 4 End) | 9/15/2021 | 9/15/2021 | 8 | 13 | Development Team, Client Reps |
|  | **Total** |  |  | **132** |  |  |

# C. Project Attributes

## Data Methods

The descriptive method for the project will be in the form of three different data charts. A bar chart, a scatter plot, and a line chart. The bar chart will be used to describe the data as it is light/medium/heavy. The scatter plot will be generated based on the number of cars present in each scene. Finally, the line chart will be representing the success of the AI over time/generations.

The non-descriptive method will be a genetic algorithm. I will use a simple division for the genes, where each gene represents the division between light/medium and medium/heavy for the classifications. The AI will start with a heuristically placed genes and will modify itself as needed to suit the dataset.

## Datasets

The dataset used to train CHAD is provided courtesy of Washington State Department of Transportation (WSDOT). It is available via Kaggle at <https://www.kaggle.com/aryashah2k/highway-traffic-videos-dataset>.

## Data Cleaning

There was no data cleaning required. However, all videos do contain one frame of bad data, so CHAD knows to start at the second frame when processing videos.

## Data Visualization

Data visualization includes a bar chart, scatter plot, and a line graph. The bar chart will be used to describe the data as it is light/medium/heavy. The scatter plot will be generated based on the number of cars present in each scene. Finally, the line chart will be representing the success of the AI over time/generations.

## Real-Time Queries

CHAD allows the users to select videos in real time to run through analysis. In this prototype, this will be simulating selecting CCTV feeds from live cameras, however the videos selected in the prototype will be pre-filmed videos.

## Adaptive Element

CHAD is trained on a fixed data set as mentioned in the “Datasets” section of this document. However, in the future, it would be plausible to extend CHAD’s training to live feedback from operators as well as additional datasets.

## Outcome Accuracy

Metrics involving the success and accuracy of CHAD’s analysis will require feedback from operators once integrated into a live environment. As mentioned in the “Adaptive Element” section of this document, this feedback can be then used to train CHAD further to improve accuracy.

## Security Measures

The data used by CHAD is not sensitive data. With this in mind, CHAD and its relevant data should only be used internally by individual DOTs. There are no networking components to be concerned about with CHAD, so individual workstation security is the only attack vector for the software.

## Product Health Monitoring

Monitoring the health of CHAD will be part of the feedback system used to improve CHAD’s accuracy. If CHADs accuracy drops at any point beyond the initial implementation, this will be indicative of faulty behavior and will require maintenance.

## Dashboard

The dashboard will include observation of the selected CCTV footage, an indicator of the traffic classification, a CCTV footage selector, and some data visualization aspects.

# D. Post-Implementation Report

## Project Purpose

The purpose of this project was to create a data product that can successfully analyze highway traffic CCTV camera footage to classify the current feed as light, medium, or heavy traffic. The data product could then be used by departments of transportation to automate discovery of traffic jams and traffic prediction data.

The project was designed to use computer vision in conjunction with a genetic AI to produce an AI that quickly and accurately labels traffic feeds for users. The project was designed to be used with a dashboard that provides a file browser to select pre-recorded traffic videos. The product was intended to be shown as a prototype, with future versions selecting live camera feeds instead of pre-recorded videos. The dashboard also includes training data visualizations in the forms of a bar graph, a line graph, and a scatter plot.

## Datasets

The dataset was acquired from Kaggle.com (<https://www.kaggle.com/aryashah2k/highway-traffic-videos-dataset>). The videos are stored in a directory, with an info file containing relevant video data in a CSV format. An example of the dataset is provided in the table below.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| filename | date | timestamp | direction | day/night | weather | start frame | number of frames | class |
| cctv052x2004080516x01638 | 20040805 | 16.01638 | south | day | overcast | 2 | 53 | medium |
| cctv052x2004080516x01639 | 20040805 | 16.01639 | south | day | overcast | 2 | 53 | medium |
| cctv052x2004080516x01640 | 20040805 | 16.0164 | south | day | overcast | 2 | 48 | light |
| cctv052x2004080516x01641 | 20040805 | 16.01641 | south | day | overcast | 2 | 52 | medium |
| cctv052x2004080516x01642 | 20040805 | 16.01642 | south | day | overcast | 2 | 51 | medium |
| cctv052x2004080516x01643 | 20040805 | 16.01643 | south | day | overcast | 2 | 53 | medium |
| cctv052x2004080516x01644 | 20040805 | 16.01644 | south | day | clear | 2 | 53 | medium |
| cctv052x2004080516x01645 | 20040805 | 16.01645 | south | day | overcast | 2 | 52 | medium |
| cctv052x2004080516x01646 | 20040805 | 16.01646 | south | day | overcast | 2 | 49 | heavy |
| cctv052x2004080516x01647 | 20040805 | 16.01647 | south | day | overcast | 2 | 52 | medium |
| cctv052x2004080516x01648 | 20040805 | 16.01648 | south | day | overcast | 2 | 53 | medium |
| cctv052x2004080516x01649 | 20040805 | 16.01649 | south | day | overcast | 2 | 53 | heavy |
| cctv052x2004080516x01650 | 20040805 | 16.0165 | south | day | overcast | 2 | 53 | medium |
| cctv052x2004080517x01652 | 20040805 | 17.01652 | south | day | overcast | 2 | 53 | heavy |
| cctv052x2004080517x01653 | 20040805 | 17.01653 | south | day | overcast | 2 | 53 | heavy |

## Data Product Code

A descriptive function was used as the basis of the analysis system. After training the AI, computer vision libraries found in OpenCV 2 provided the requisite analysis and fed the data into the AI, which then classified the data. To create the AI, OpenCV would analyze a video feed to detect vehicles. At the end of each video, an average of the non-zero frames would be fed into the AI for storage and future processing. This number would represent the number of vehicles ‘detected’ in the video. Once the AI has seen all of the training videos, the AI would then look back at its own history of videos and determine where to draw the line between light/medium and medium/heavy. To encourage large movements where different generations are far apart from each other, these differences would be “pulled” with a force equal to the following equations: , , where and represent the absolute value of the difference in generations between light/medium and medium/heavy, respectively.

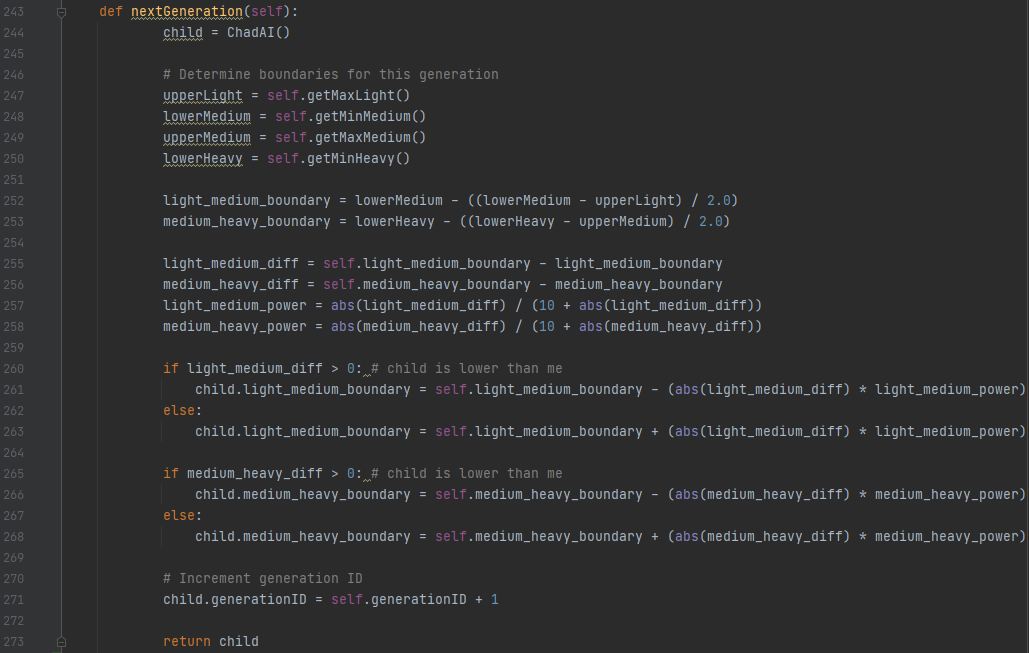


Figure – Creating generations



Figure – Detecting vehicles (contours)

## Hypothesis Verification

The original hypothesis was that CHAD will be able classify highway video feeds as light, medium, or heavy traffic. In initial testing and the prototype version, this hypothesis is correct. Each video that CHAD is given is properly classified as either light, medium, or heavy traffic. Additionally, this classification aligns with what a human tester would classify the video as. Of course, this hypothesis will need to be tested again once implemented fully into a live environment. It will be verified against live testers, as mentioned in the *Outcome Accuracy* section of this document. While CHAD is very successful at classifying the videos as light, medium, or heavy, it is not very good at accurately counting the number of vehicles in each scene. However, this number is not needed, as the metrics CHAD uses are still in line with the severity of the traffic.

## Effective Visualizations and Reporting

The dashboard includes four different visualizations: the video being analyzed, a bar graph, a line graph, and a scatter plot.

The video being analyzed is displayed in the center of the dashboard much like a typical media player. This is available so the user can watch the video in real time as the AI grades it. The bar graph is a representation of the training data provided and shows how many videos of each category were available to CHAD to train with. The line graph generates two images, one for each of the boundaries between light/medium and medium/heavy. These graphs display the change in the boundaries for each generation, representing the accuracy changes for CHAD as it evolves. Finally, the scatter plot represents the number of objects detected in each scene, color coded by light, medium, and heavy. The scatter plot is particularly useful in determining distributions of each video for training purposes and verifies that the training set provided thorough and robust videos.

## Accuracy Analysis

The accuracy of the system will be determined by users post-implementation, as discussed in the *Hypothesis Verification* and *Outcome Accuracy* sections. Simply put, we include a button on the dashboard that will allow the user to override CHAD’s classification. In this override, the user will need to tell CHAD how many cars are actually present and what the classification should be. This data will be used by CHAD to create a new generation with more accurate AI.

## Application Testing

Testing took place at the end of the developmental cycle. Testing was done by selecting video files from the dataset and feeding them into CHAD, then verifying that CHAD properly classifies them. An example of this being done is provided in the figure below. Testing was performed for known video files of each classification to confirm rigidity. Additionally, I had a few friends and family test the software to validate usability of the dashboard and navigation buttons.

## Application Files

Application files are hosted on GitHub at the following URL: <https://github.com/morgan-webber/WGU_Capstone>. The repository contains a directory name CHAD, which is the application directory and contains the following files:

|  |  |
| --- | --- |
| File | Purpose |
| .idea | JetBrains IDEA Project File |
| TrafficDataset | The dataset containing the video files |
| ChadAI.py | The CHAD class |
| Dashboard.ui | A Qt designer file |
| Dashboard.py | The dashboard python file |
| Main.py | The main python file |
| Playbutton.png | Icon for the play button |

## User’s Guide