Software Requirements Documentation

VehID

Version 1.0

Remington Greko, Spencer Hirsch, Thomas Johnson, Alexis Nagle

Project Advisor: Dr. Marius Silaghi
Project Client: Clayton Levins, Executive Director of Smart North Florida

October 2, 2023

Contents

1 Introduction

1.1 Purpose

This document is intended to describe our goals and overall objective of the VehID (1.3.1) project. The documentation that follows serves as an overview of our intend features and product functionalities. Our goal is to clearly define our product and the expectations for the completed system in order for our stakeholders and customers to better understand our product from a more technical perspective.

1.2 Scope

The product that we intend to produce is called VehID (1.3.1). VehID is a machine learning model that will utilize three separate Convolutional Neural Networks (CNNs) (1.3.2) in order to accurately and effectively identitfy vehicles based on set criteria. The criteria that we are intending to train these models with are: vehicle make, vehicle model, and vehicle license plate. If a vehicle has been to be determined a match, our system would return the location of the vehicle to the proper authorities.

1.3 Definitions, Acronyms, and Abbreviations

Reference No.	Abbreviation	Definition	Source Reference
1.3.1	VehID	Vehicle Identification (Our Product Name)	
1.3.2	CNN	Convolution Neural Network: A type of feed-forward neural network commonly used for image recognition and processing.	Wikipedia
1.3.3	AMBER Alert	America's Missing, Broadcast Emergency Response; Public system for distribution of information about missing/kidnapped children	Wikipedia
1.3.4	DOT	United States Department of Transportation	

1.4 Overview

The rest of this documents provides descriptions and detailed specifications of the VehID project. The outline of the rest of the document is as follows:

- Overall Description: This section will provide an overview of the product and it's requirements. This contains descriptions of the operation of the software, the product's functions, the intended user's characteristics, constraints, assumptions, and dependencies.

- Specific Requirements: This section will contain detailed descriptions of all software requirements. This contains external interfaces, functions, requirements, constraints, and system attributes

2 Overall Description

2.1 Product Perspective

Our product is an independent, self-contained product. The product works on it's own from data gathering (either manual entry or scraped from the web) to processing the data and displaying valuable information to the users via the user interface. The product will have some external interaction when scraping data from web sources to help create the database that will be utilized for the search query.

2.2 Product Functions

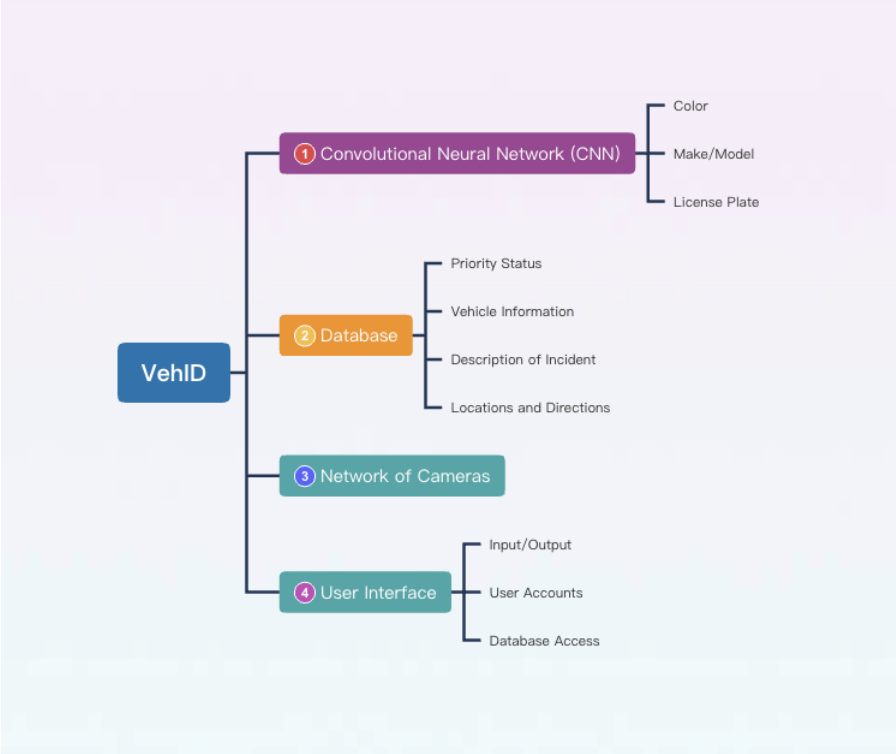

Figure 1: Breakdown of system

In the Figure 1, we see the different functions of our product. The primary portion being the Convolutional Neural Networks. There will be three different networks intended to solve a very specific portion of the task. The remaining components of the project are intended to support the application, focusing on making it user friendly and lightweight. Our product

is intended for law enforcement, therefore they need a lightweight easy-to-use tool to help them better accomplish the task.

2.3 User Characteristics

Our product has a very specific target audience, our users would be government bodies and law enforcement. Our product would hide all of the technical components on the backend, therefore, the users will not need to have any technical experience in order to use our product. The user will only be interaction with our web application, this will be a very intuitive input/output system. Our product, does however rely on a database to relay information and communicate, a user will be able to view these components and add to them. This will be done in such a way where the user will easily be able interact with it. No specific training will be required in order to use our product, as for educational level, it is reasonable to assume that any individual will be able to use our product despite educational level.

2.4 Assumptions and Dependencies

2.4.1 External Interfaces

- Assumptions
  - **Availability of Live Video Feed:** The system assumes that a live video feed from existing roadway cameras will be consistently available for input. Any disruptions or failures in this feed may affect the system's ability to analyze images and find matches.
  - **Partial Matches Handling:** The system assumes that it may not always receive complete information about a vehicle from the input image. It should be capable of handling partial matches and make decisions based on the available attributes.
  - **Database Connectivity:** The system assumes that it will have uninterrupted access to the database for storing and retrieving data. Any database connectivity issues could affect the system's ability to store and retrieve matching images and attributes.
- Dependencies
  - **Live Video Feed:** The system's ability to function depends on the availability and quality of the live video feed from existing roadway cameras. Any disruptions or changes in this feed may affect the system's performance.
  - **Notification System:** The system's ability to notify users when a match is found and when the database updates depends on the functionality and reliability of the notification system.

2.4.2 Functions

- Assumptions

- **Web Data Availability:** The system assumes that web scraping will always be able to retrieve information on the target vehicle. Any changes or disruptions to the sources of this web data may affect the system's ability to gather information
- **Input Data Validity:** The system assumes that input data, including images, and other information, will be valid and correctly formatted. Invalid or corrupted data may require additional error handling.

- Dependencies
  - **Web Scraping Dependency:** The system's ability to gather information on target vehicles through web scraping depends on the availability and structure of the external websites providing this data.
  - **Error Handling and Recovery:** The system's responses to abnormal situations, including overflow, communication failures, and error handling, are essential for maintaining system stability and data integrity.
  - **Parameter Effects:** Any changes to system parameters may impact its performance, including the speed and accuracy of image processing and the frequency of database updates.

2.4.3 Performance Requirements

- Assumptions
  - **Multiple User Support:** The assumption is that the user interface will be designed and implemented to support a number of users simultaneously. This assumes that the underlying infrastructure can handle the potentially high concurrent user load.
  - **Database Concurrency:** The system assumes that concurrent database updates by multiple users will be handled without data loss or conflicts. Proper database management and synchronization mechanisms are required to ensure this.
  - **Universal Video Format Compatibility:** The assumption is that the video processing module can effectively handle MP4 video format

- Dependencies
  - **Server Infrastructure:** The ability to support a number of simultaneous users depends on the server infrastructure's capacity and scalability. Adequate resources, and load balancing mechanisms must be in place.
  - **Database management:** Handling concurrent updates without data loss depends on proper database management techniques.

2.4.4 Logical Database Requirements

- Assumptions
  - **Data Integrity:** The system assumes that data integrity will be maintained throughout its life-cycle. This includes accurate and consistent storage of information, data validation, and error handling mechanisms.
- Dependencies
  - **External Data Sources:** Dependence on external data sources, such as AMBER Alerts (1.3.3) and stolen vehicle databases, requires reliable data retreival mechanisms to access and update information.

3 Specific Requirements

3.1 External Interfaces

- Input: Image of a vehicle.
  - The input image will be analyzed by the system to determine if the vehicle is associated with any outstanding warrants or alerts. If the image contains a match, the system will gather location and heading of the vehicle.
  - Input will be recieved from live video feed from existing roadway cameras.
  - The image must exactly match the attributes of the vehicle. Not all information may be available so partial matches may suffice in such situations
  - Image processing must be done in real time to ensure location accuracy when producing output.
  - The output from matches will result in new data to be being stored in the system database.
  - When a match is found the captured image will be sent to the systems database for storage.
  - The user will not be able to view images unless a match is found and the image is stored.
  - The input will be in image format and the output will contain the image as well as textual information pulled from said image.
  - Users will recieve a notification when a match has been found and the database updates.
- Output: Matching images and attributes
  - When a target vehicle has been found and processes by the system it will then be stored in a database which will allow users to interact with the matches.

  - The database will be accessed through a web application with an interactive GUI allowing the user to filter and search matches.
  - The database must be able to be updated in real time to ensure users are recieving up to date information.
  - The output is the information from matching vehicles displayed on the users screen.
  - The application will run in the browser and will have a standard database view layout with a navbar.

3.2 Functions

- Upon new data on a target vehicle:
  - The system shall web scrape information on the target vehicle and store it so that the system may search for it.
- Upon recieving a frame containing vehicles from the live video feed:
  - The system shall process the image by using multiple CNNs to gather the attributes of the vehicle.
  - The system shall discard images without matches and keep those with matches.
- Upon recieving an image with a matching vehicle:
  - The system shall store the image and its attributes in the database for the user to view.
- Upon the database recieving an update
  - The system shall update the database to allow the user to view the most recent information on a vehicle.

3.3 Performance Requirements

Static Requirements:

- The user interface should be able to support a number of users simultaneously. This would include allowing multiple users to update the database simultaneously without losing either user's entry.
- The video processing portion of the product should be able to handle MP4 file formats.
- The image classifier should be able to recognize 11 colors: white, black, grey, silver, red, blue, brown, green, beige, orange, gold, yellow, purple, pink, and tan.
- The image classifier should be able to recognize over 100 make/model combinations.

Dynamic Requirements:

- User's shall not have to wait for updates to the database to be processed, these updates should be immediate.
- The video processing should be able to extract 70% of the vehicles within each frame.
- Images extracted from video footage should be classified within 5 seconds.

3.4 Logical Database Requirements

Our database will hold a variety of information including:

1. Associated Name
2. Associated Address
3. Vehicle Identifiers
   (a) Color
   (b) Year
   (c) Make
   (d) Model
   (e) License Plate
4. Case Information
5. Description
6. Locations

This information will be interacted with using a web interface. A user will be able to add to this data, an administrator will be able to remove and edit information as new information comes in. Some information will be changed automatically, such as, location. Location is a multi-variable attribute that will change as new information comes to light. This will be be the primary information that will alert the proper authorities. In addition to manual entry, web-scraping will be employed to extract data from credible resources, in the cases of AMBER Alerts (1.3.3) and stolen vehicles that information is made widely available to aid in the search.

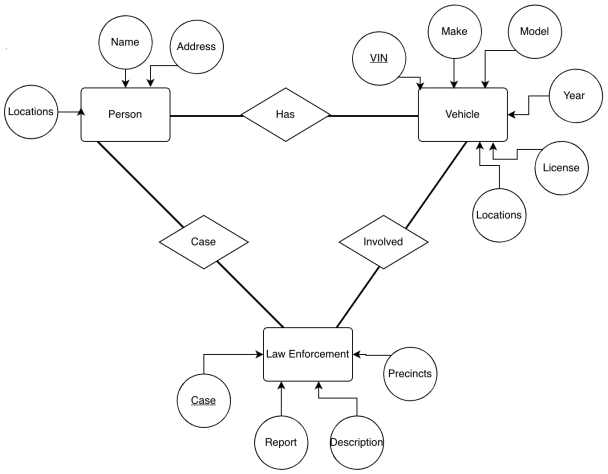

Figure 2: Entity-Relationship Diagram

3.5 Design Constraints

- The cameras used in the system must be high enough resolution and refresh rate that the license plate can be seen by the system.
- The camera must be able to transmit images to the system in real time.
- The system must be able to detect multiple vehicles in a single frame and process the image in real time.
- The system must securely store information to ensure personal information in the database is protected.
- The system must meet a certain level of accuracy and minimize false positives.
- The system must be compatible with multiple existing operating systems used in roadway cameras.
- The system must comply with local laws and regulations.

3.6 Software System Attributes

3.6.1 Reliability

- The system should exhibit a high level of uptime, minimizing service disruptions.
- Fault tolerance mechanisms should be in palce to handle unexpected errors.
- Image processing algorithms are crucial for accurate vehicle identification.

3.6.2 Availability

- The system should support a number of simultaneous users without performance degradation.
- Checkpoint mechanisms should enable rapid recovery in case of system failures.
- Reail-time database updates are essential to ensure users receive up-to date information.

3.6.3 Security

- User authentication and access control mechanisms should ensure authorized access.
- Regular security assessments and updates are essential to address vulnerabilities.

3.6.4 Maintainability

- Well structured, and documented code.

3.6.5 Portability

- Compatibility with multiple operating systems.
- Minimize platform specific dependencies.

Works Cited

"Convolutional Neural Network." Wikipedia, Wikimedia Foundation, 3 Sept. 2023, en.wikipedia.org/wiki/Convolutional_neural_network.

"Amber Alert." Wikipedia, Wikimedia Foundation, 12 Sept. 2023, en.wikipedia.org/wiki/Amber_alert.