High Level Design (HLD)

FLIGHT FARE PREDICTION

Revision Number: 2.0

Last date of revision: 20/09/2022

Document Version Control

Date Issued	Version	Description	Author
19/09/2022	1	Initial HLD - VI .0	Durgesh Bhardwaj

Contents

Abstract

The **Flight ticket prices** increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights. In the proposed system a **predictive** model will be created by applying machine learning algorithms to the collected historical data of **flights**.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Security
  - Reliability o Maintainability
  - Portability
  - Reusability
  - Application compatibility o Resource utilization
  - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	Description
UGV	Unmanned Ground Vehicle
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
A WS	Amazon Web Services

2 General Description

2.1 Product Perspective

The UGV based Surveillance solution system is a deep learning-based object detection model which will help us to detect the anomalies in the society and take the necessary action.

2.2 Problem statement

To create an Al solution for surveillance using rover or UGV and to implement the following use cases.

- To detect mob (illegal) activities and inform police.
- To detect disasters (like fire, smoke, etc..) and send details to concerned authorities.
- To detect a medical emergency (accident, etc..) and take emergency action (call ambulance and raise an alarm to the nearest hospital) for help.

2.3 PROPOSED SOLUTION

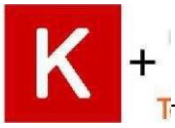
The flight ticket buying system is to purchase a ticket many days prior to flight take-off so as to stay away from the effect of the most extreme charge. Mostly, aviation routes don't agree this procedure. Plane organizations may diminish the cost at the time, they need to build the market and at the time when the tickets are less accessible. They may maximize the costs. So, the cost may rely upon different factors. To foresee the costs this venture uses AI to exhibit the ways of flight tickets after some time. All organizations have the privilege and opportunity to change its ticket costs at any time .

2 machine learning :

Machine Learning algorithms are applied on the dataset to predict the dynamic fare of flights. This gives the predicted values of flight fare to get a flight ticket at minimum cost. Data is collected from the websites which sell the flight tickets so only limited information can be accessed. The values of R-squared obtained from the algorithm give the accuracy of the model. In the future, if more data could be accessed such as the current availability of seats, the predicted results will be more accurate. Finally, we have created the entire process of predicting an airline ticket and given a proof of our predictions based on the previous trends with our prediction.

2.7 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, TensorFlow, Keras and Roboflow are used to build the whole model.




g z: ROS e

GAZEBO

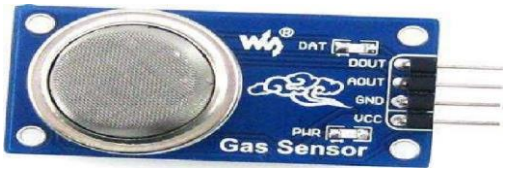
- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- AWS is used for deployment of the model.
- Tableau/Power Bl is used for dashboard creation.
- MySQL/MongoDB is used to retrieve, insert, delete, and update the database. • Front end development is done using HTML/CSS • Python Django is used for backend development. • GitHub is used as version control system.

2.7.1 Hardware Requirements

- USB Camera for object Detection
- LM35 temperature sensor

- MQ Smoke Detector Sensor
- PC (check you are system supports: https://7dfps.com/ros-systemrequirements/)
- HC-SR04 Ultrasonic sensor
- Ground vehicle
- Raspberry Pi

Vcc (SV) %round



2.7.2 ROS (Robotic Operating System)

Robot Operating System is an open-source robotics middleware suite. Although ROS is not an operating system but a collection of software frameworks for robot software development, it provides services designed for a heterogeneous computer cluster such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management.

2.8 Constraints

The UGV based Surveillance solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

High Level Design

2.9 Assumptions

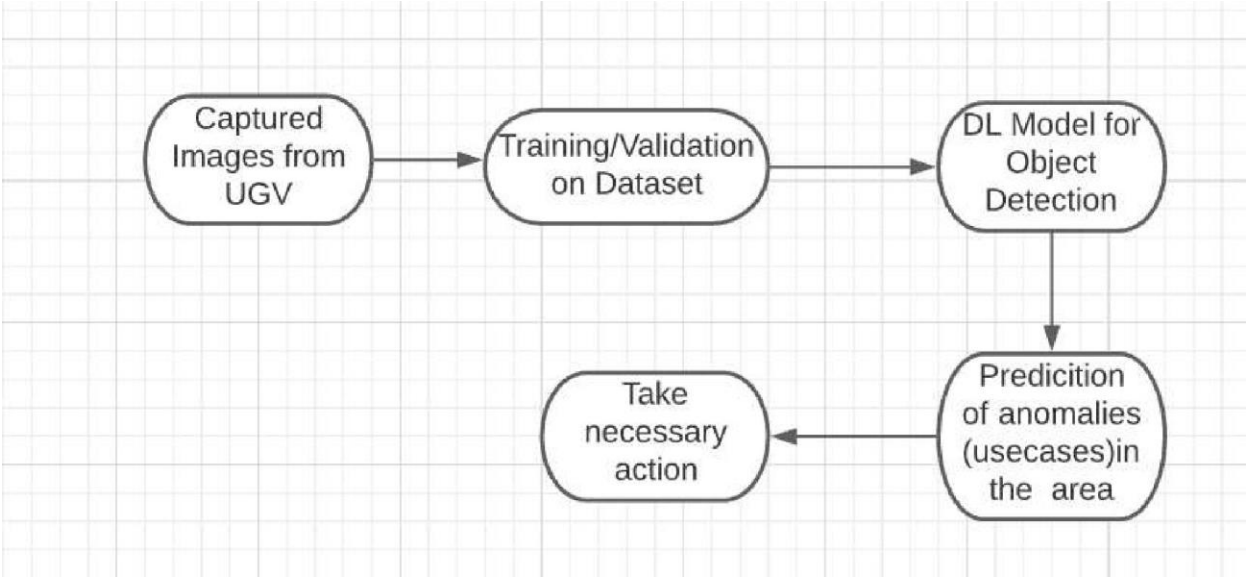
The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes through UGV vehicle which has camera installed for capturing the live videos. Deep Learning based object detection model is used for detecting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in thevwthe designer is expecting.

3 Design Details

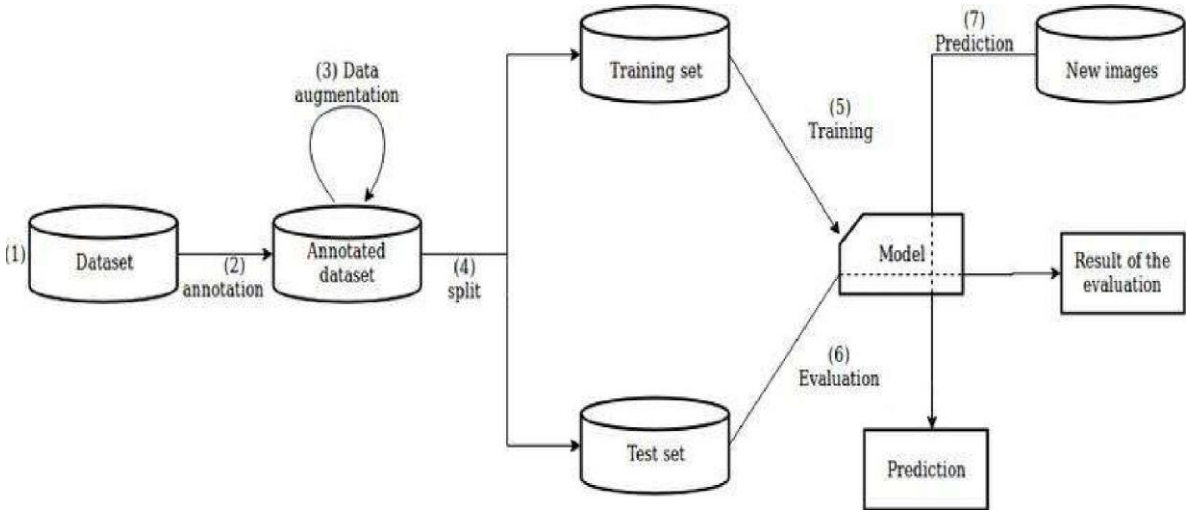
3.1 Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram is as shown below.

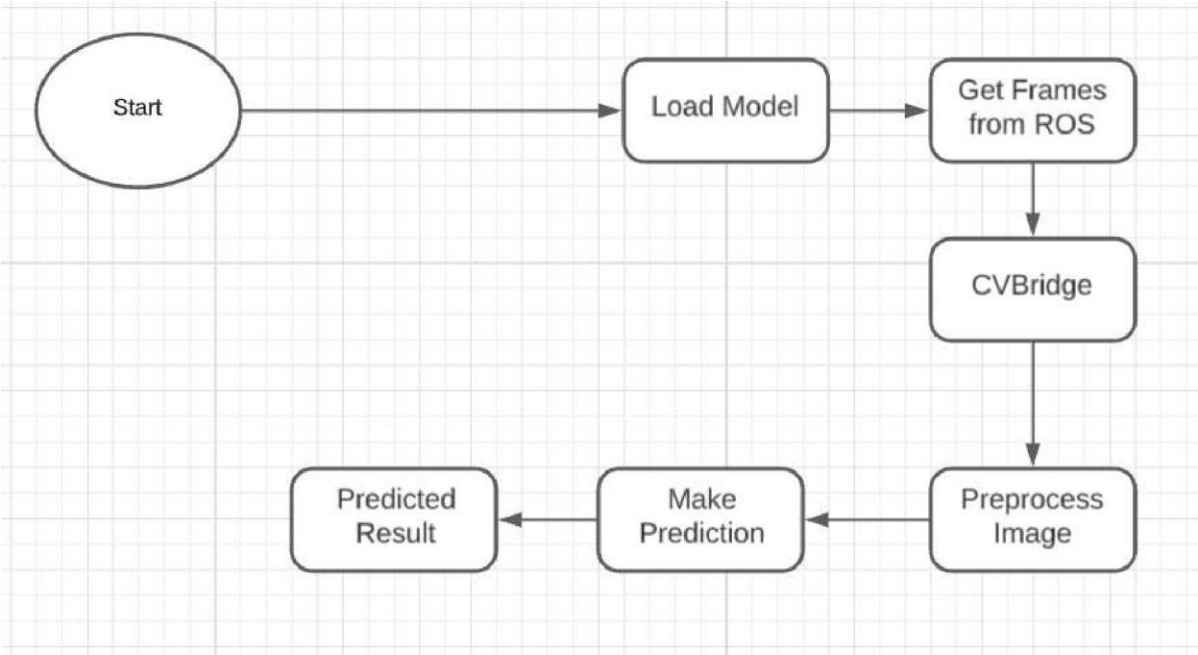
Proposed methodology



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4

Performance

The UGV based surveillance solution is used for detection of anomalies in the society whenever UGV detects any anomalies (mob, medical emergency, fire, smoke, etc...) it will inform concern authorities and takes necessary action, so it should be as accurate as possible. So that it will not mislead the concern authorities (like hospitals, cops, etc..). Also, model retraining is very important to improve the performance.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that

function is finished.

4.4 Deployment

Microsoft

Azure webservicesruamazon

Google Cloud

6

Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.

As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.

5.1 KPIs (Key Performance Indicators)

1 . Key indicators displaying a summary of the anomaly detection in the society/area.

2. Time and workload reduction using the UGV based surveillance.
3. To detect mob (illegal) activities and inform police.
4. On time alert to nearest hospital on medical emergency (accident).
5. Taking adequate evidence of mob.
6. Send disaster details to concerned authorities.
7. Display of battery life and percentage of UGV.
8. Distance travelled by UGV.
9. Get the exact location of UGV.

Conclusion

The Designed UGV (Unmanned Ground Vehicle) will detect an anomaly in the locality based on various anomalies data used to train our algorithm, so we can identify the

imbalance in the society in early stages and can take necessary action to stop them immediately, so we can have a pleasant environment in that area or location.

8

15

References

1 . https://en.wikipedia.org/wiki/Unmanned ground vehicle

2. Google.com for images of UGV hardware.
3. https://www.ros.org/