Road and Object-detection based Assistant for Driving (R.O.A.D)

By:
Arzem Cheema
Junaid Javed
Abdul Moqueet
Awais Noor

Supervisor:
Hammad Afzal

Submitted to the faculty of Department of Software Engineering,
Military College of Signals (MCS), National University of Sciences and Technology, Islamabad
as a partial fulfillment criteria for the requirement of the Degree of Bachelors of Engineering in Software Engineering
JUNE 2021

CERTIFICATE OF CORRECTIONS & APPROVAL

It is attested that the information present in this thesis called "Road and Object-detection based Assistant for Driving (R.O.A.D)", carried out by *Arzem Cheema*, *Abdul Moqueet*, *M. Awais Noor* and *Junaid Javed* under the supervision of *Prof. Hammad Afzal* required for the completion of Bachelors of Engineering in Software Engineering (BESE), offered by Military College of Signals, National University of Sciences and Technology, Islamabad during the year 2020-2021 is accurate and accepted.

Approved by

Signature:

Prof. Dr. Hammad Afzal

(Supervisor)

Date: 19 June 2021

DECLARATION

We testify and declare that any part of this thesis has not been reused from any other degree program or any other source or from any other university or educational institutions.

CERTIFICATE OF PLAGIARISM IN THESIS (TURNITIN REPORT)

This thesis has been checked for Plagiarism. Turnitin record advocated by using Supervisor is attached. This document has a collective plagiarism of 20% and has been passed by the supervisor.

Supervisor Name & Signature: Dr Hammad Afzal

Name of Students & Signature:

Arzem Atif Cheema 00000240965

Muhammad Awais Noor 00000240971

Abdul Moqueet 00000240960

Junaid Javed 00000240940

ACKNOWLEDGEMENTS

We are grateful to the Creator Allah Subhana-Watala to have directed us all through this work at each progression and for each groundbreaking idea which You setup in our mind to improve it. Undoubtedly, we might have done nothing without Your extremely valuable assistance and direction. Whosoever helped us throughout my proposition, regardless of whether my folks or some other individual was Your will, so in fact none be worthy of praise but You.

We are profusely grateful to our cherished parents who raised us whilst we had been now not capable of taking walks and endured to assist us for the duration of in every branch of my existence.

We would additionally like to explicit unique thanks to our manager Prof. Hammad Afzal for his assist for the duration of my thesis and for his or her regular route and thought at some point of our task.

Finally, we would really like to specific our gratitude to all of the people who have rendered treasured assistance to our take a look at.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

ABSTRACT

Humans while driving have a disadvantage of not always being attentive (whether it be changing the radio or of being tired etc.) while a computer, if trained, can always be fully attentive at detecting lanes.

Driver support system is one of the most important feature of modern vehicles. This is to ensure driver safety and decrease the vehicle accident on roads. The most important aspect of such a system is its computer vision-based software. This software detects and recognizes the lane in which the car is currently in and also detects objects in the observable-environment of the car/software. But such a system is not suited for Pakistani roads due to the traffic norms and low maintenance of road conditions, so a system is required which can perform well under Pakistani road conditions.

Hence R.O.A.D is developed which is a desktop-based python application, which detects lane and objects from a dash cam video using CNN based trained models. User uploads a video which is broken into frames and pre-processing is applied to each one. Then the model detects the lane from the frame and creates a visualization at the detected pixels. Then the object detection model detects the objects, and the frames are compiled back into a video format and the detected video is displayed to the user.

Key Words: *Computer-Vision, CNN, python*

Contents

CHAPTER 1 INTRODUCTION

Overview

The main objective of our system is to create a desktop application which assists drivers to detect lane and objects while driving. The system will take a video input from the user. After uploading the video lane detection starts, image pre-processing is applied which includes frame extraction, reducing image noise and implement trained model on the extracted frames. After successfully applying trained model the detected lane is super imposed onto the original input video and the detected lane is displayed in the output video along with the surrounding objects.

The system consists of 5 major modules:

- Video Acquisition module
- Pre-Processing module
- Trained models
- Lane Detection module
- Visualization module

Problem Statement

The purpose of the application is to aid in self-driving cars, by not only detecting the current lane of the driver but also the surrounding objects on the road and observable-environment such as pedestrians, vehicles and traffic lights etc.

Scope

R.O.A.D will get input from the user in the form of a video. Applying image processing and object detection to distinguish lanes, vehicles and alert the driver. R.O.A.D will provide a deeper insight to image processing and computer vision and enabling different aspects of lane and object detection and improving the results. The end goal of this object will be to:

- Train a model and process data sets for lane detection.
- Train a model and process data sets for object-detection.
- Improving the image processing and lane detection process.
- Integrating results from both the models into a single processed video file.

- Allow interaction of the software with the user through a GUI.

Objectives

The main objective of the system is to build a desktop application which provides efficient detection of lanes and the objects, which are encountered during driving.

During this project, all the aspects of software engineering are covered i.e survey, and feasibility analysis requirement gathering architectural and detailed design, implementation and testing along with documentation (SRS, SDD, Test Cases, Final Report and User Manual). Students are also expected to develop extensive knowledge and technical skills in the following areas:

- Deep Learning
- CNNs
- Object Detection
- Machine Learning
- Python Programming

Intended Audience and Reading Suggestions

The intended audiences for the R.O.A.D SRS include the Project Supervisor, FYP Group itself (developers), UG Project Evaluation Team, and other persons at MCS CSE Department.

This document is intended for:

- **Developers: (Project Group)**

To be certain that they are building up the correct venture that satisfies the necessities are given in this report.

- **Testers: (Project Group, Supervisor)**

To have a definite rundown of the functionalities and whether the project runs according to the expected output.

- **Users:**

To get familiar with the requirements of the tasked project and how to address different circumstances and propose different highlights that would make it productive.

- **Documentation writers: (Project Group)**

To recognize different features and how they need to be resolved. What improvements are required and how the application will react on different inputs.

- **Project Supervisor:**

This document will be used by the project supervisor to check whether all the requirements have been understood and, in the end, whether the requirements have been implemented properly and completely.

- **Project Evaluators: (CSE Dept. MCS)**

To know the scope of the project and evaluate the project throughout the development of grading.

Deliverables

Deliverable Name	**Deliverable Summary Description**
Literature Review	A detailed study and survey of related research papers
SRS Document	Complete Description of what the system will do along with the detailed description of functional and non-functional requirements and the system features.
SDD	Complete description of how the system is implemented, the detailed design.
Implementation	Pseudo code of the implemented system.
Testing	Result of tests and accuracy of the models being used

Overview of the Document

This document shows the complete working process of our project R.O.A.D. It starts with the literature review which shows past work done in a similar field, requirement analysis of the system, system architecture which highlights the modules of the software and represents the system in the form of a component diagram, Use Case Diagram, Sequence Diagram, and general design of the system. Then it will move on to discuss the detailed Description of all the components involved. Further, the dependencies of the system and its relationship with other products and the capacity of it to be reused will be discussed. In the end test cases and any future work, proposal has been presented.

Document Conventions

This section describes the standards followed while writing the document.

Headings

Headings are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, per their level. All the main headings are titled as follows: single-digit number followed by a dot and the name of the section (All bold Times New Roman, size 16, Center-aligned). All second-level subheadings for every subsection have the same number as their respective main heading, followed by one dot and subsequent subheading number followed by name of the subsection (All bold Times New Roman, size 14). Further subheadings, i.e., level three and below, follow the same rules as above for numbering and naming, but different for the font (All bold Times New Roman, size 12).

Figures

All figures in this document have captions and are numbered. Context and flow diagrams are based on UML standards.

References

All references in this document are provided where necessary, however, were not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

Basic Text

All other basic text appears in regular, size 12 Times New Roman. Every paragraph explains one type of idea.

CHAPTER 2 LITERATURE REVIEW

Introduction

In recent years, the problem of lane and object detection using computer vision techniques by leveraging object features has become widespread specially after the introduction of self-driving cars. However, despite its potential usefulness, the task of lane detection and object detection has been less studied in regard to Pakistani road and traffic conditions.

Purpose

Humans while driving have a disadvantage of not always being attentive (whether it be changing the radio or of being tired etc.) while a computer, if trained, can always be fully attentive at detecting lanes.

Driver support system is one of the most important features for modern vehicles to ensure driver safety and decrease the vehicle accident on roads.

Ever since tech giants like Tesla and Google have announced their ambitions to deploy self-driving cars in the next decade, there has been an increase in the number of techniques developed for this using both hardware and software. The most successful have been through the use of a camera and a software that uses computer vision techniques based on a trained deep learning model.

Apart from self-driving cars, it can assist drivers by showing them (potentially on an in-vehicle screen) that the car can always sense the lanes will go a long way in getting them comfortable with a computer doing the driving for them. Even short of this, enhanced lane detection could alert an inattentive driver when they drift from their lane. Also, public datasets are becoming increasingly available that are specifically designed for this task.

The demand for lane and object detection is still increasing as no one has been able to perfect a system for deployment on actual roads. For deployment on Pakistani roads require a system specifically built for these road conditions.

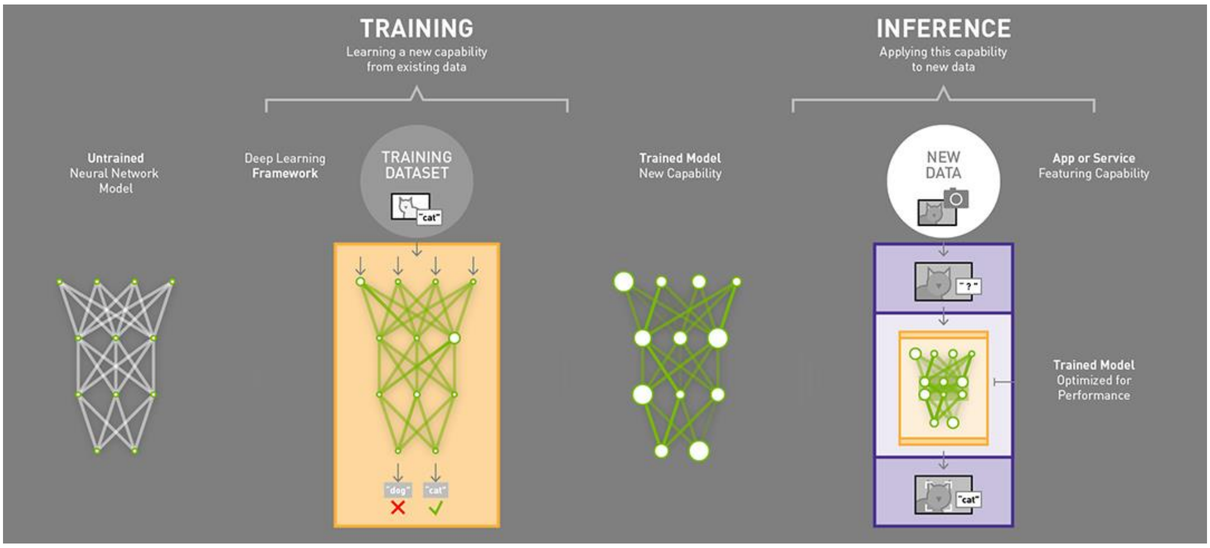
Why Deep Learning

Lane Detection algorithms are based on Computer Vision techniques which are capable of extracting features. These features are human engineered, and accuracy and the reliability of the models directly depend on the extracted features and on the methods used for feature extraction.

The difficulty with this approach of feature extraction in frame classification is that the features that are to be considered must be pre-defined. When the number of classes of the classification goes high or the image clarity goes down it's hard to work with traditional computer vision algorithms.

The main difference in deep learning approach of computer vision is the concept of end-to-end learning. There is no longer need of defining the features and do feature extraction. The neural network extracts those itself and defines the weights accordingly via back propagation.

If a [deep] neural network is to recognize a cat, for instance, it is not told to look for whiskers, ears, fur, and eyes. The neural network is shown thousands and thousands of photos of cats, and eventually it works things out. If it keeps misclassifying foxes as cats, nothing is hard-coded, and the neural network eventually learns itself.



Object Detection Techniques

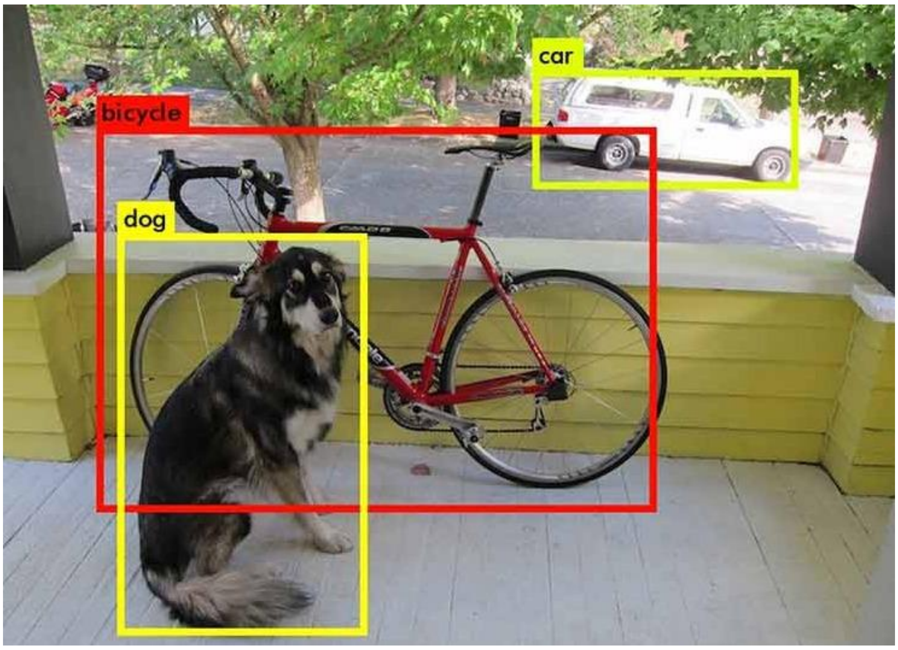
Thus in 2001 for the first time, a real time image detection framework was deployed by Paul Viola and Michael Jones in their paper 'Rapid Object Detection using a Boosted

Cascade of Simple Features’ [7] which detected objects/images based on Haar Features. [7] uses a cascade of weak detectors and the input image is broken down into smaller groups and fed/passed through the cascade of Haar features and if that image passes through all the features in that cascade, then that class is assigned to that image.

Then different datasets were released for object classification, annotation and accessing them. Also, competitions were held to allow different people to come up with innovative methods and algorithms for object recognition.

From 2012, came the ImageNet Visual Recognition Challenge. A dataset called ImageNet was released which contained a million images and a thousand object classes. Since its inauguration, the ImageNet challenge has become a benchmark in object classification and object detection across a huge number of object categories.

In 2010 and 2011, the ImageNet’s error rate in image classification hovered around 26%. But in 2012, a team from the University of Toronto entered a convolutional neural network model (AlexNet) into the competition and that achieved an error rate of 16.4%. The model, similar in its architecture to LeNet-5. This was a breakthrough moment for CNNs. Since then CNNs are used till now in Computer Vision.



Lane Detection Techniques

Same as with Object detection, lane detection also improved after the advent of ImageNet dataset and re-emergence of AI specifically CNNs, from previously feature-based or model- based approaches. Feature-based techniques work by preprocessing and thresholding of the video frames hence work only in simple scenarios where it is easy to distinguish the lane/road boundary.

Then there are model-based approaches, based on mathematical models, like the Hough Transform, which have a limitation at the Vanishing Point, i.e the virtual horizon of the video frame where both the parallel lines of the lanes meet. Also, they cannot distinguish between non-lane edges or lines in the images (such as road curbs, multiple lane lines, shadows), nor can they handle false negatives caused by erased or inconsistent lane boundary markings.

Finally we have our machine learning (supervised learning) based techniques, where we have a Deep Learning based model trained on an annotated datasets like CU Lane, TuSimple, CalTech and BDD. Deep learning-based models, learn the patterns from the dataset with labelled or annotated ground truth in features that are cascaded in the hidden layers of the architecture. This is the supervised classification of lane markings against the background.

Learning the position of the vanishing point with or without ground truth helps to generate semblance of lane lines, even with the absence of actual lines on the road or with lines obstructed by traffic.

Research Work

• Jae-Hyun Cho et al. (2014)[8] carried out the Hough transform by way of optimizing the cells within the four ROI in parallel and detects lanes with excessive performance. Though Hough Transform can hit upon handiest directly lines therefore recognition charge for curved lanes was low.

• Chan Yee Low et al. (2014) [9] supplied a robust road lane marker detection algorithm to hit upon the left and right lane obstacles. The algorithm includes an optimized version of Canny part detection and Hough Transform. Canny area detection plays functions recognition then implemented Hough Transform for creating visualizations for the lanes. Hough Transform is carried out to find relevant traces that can be used because the left and proper lane barriers. Reducing the image to smaller ROI can lessen excessive computational value.

• Mingfa Li, Yuanyuan Li, and Min Jiang from the Department of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China of their paper 'Lane Detection Based on Connection of Various Feature Extraction Methods' [10] discuss a brand-new technique for preprocessing and selecting ROI, the use of KITTI dataset, the identical dataset which we've used to teach our model in R.O.A.D.

• Kreucher C. Advise in [11] LOIS the algorithm (Likelihood of Image Shape) has been shown to discover strong markings, even within the presence of statement of occlusion and a plurality of mild conditions. He uses the set of rules to follow the laws of the street thru a chain of snap shots, and a warning of a crossing is Imminent.

• Mellon University developed a gadget known as AURORA [12] the lane marks determined on dependent roads as highways and town streets. The lateral role of the car calculated from the detected line marker. If the car starts to stray from the course, alerts the driver with audible and visible alarms AURORA. We also are an lively intervention within the form of the automobile's guidance and pace device that collision avoidance strategies aggressive.

• Satish Kumar Satti, P. Sridivasen from NIT, India proposed a CNN primarily based gadget in their paper ‘A gadget mastering technique for detecting and tracking street boundary lanes’ [13]

Purposed System

The proposed system is a python-based desktop application which uses trained models for detecting lanes and surrounding objects in a video provided by the user.

It uses model based on CNN, trained on annotated dataset, thus making it more accurate and reliable then other proposed systems especially those which are based on features and mathematical polynomial regression. That is because they are not very reliable on curved roads and rely mainly on the accurate preprocessing in the form of filters and morphing techniques to detect edges or groups of pixels (in case of road lanes) and then extrapolates them by regression to visualize the markings.

Our system relies on CNN, which extracts hidden features and is independent from the environment.

Many systems are formulated based on the assumption of good road conditions as they are made for European and other western countries, but our system is also deployed on

Pakistani roads and it is tailored to show effective results on Pakistani road conditions as well.

Only a few systems are detecting both lanes and objects simultaneously, so the proposed system achieves this by deploying two models for each task.

For future developments, the system can be applied for real time detection by using processing devices like NVIDIA and mounting it on the vehicle.

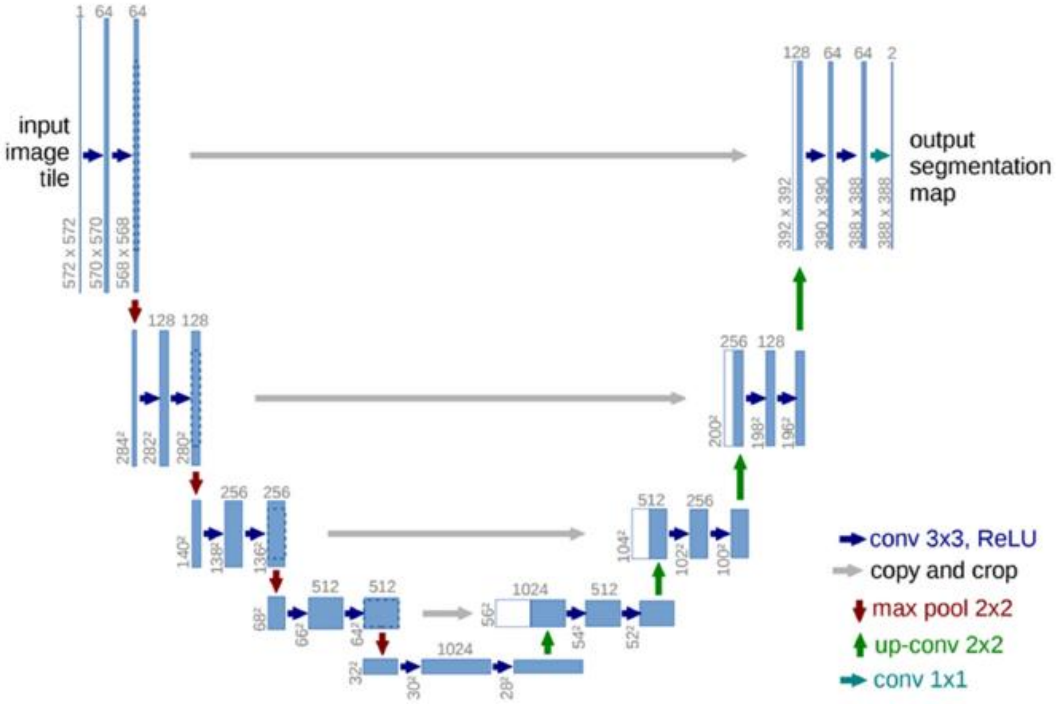
CHAPTER 3 METHODOLOGY

Learning Features

The CNN model will be trained similarly to a regression-type problem, in which it will be given the polynomial coefficients on the training images, and attempt to learn how to extract those values from the given road images. The loss it will minimize is the difference between those actual coefficients and what it predicts (likely using mean-squared error). It will then use computer vision techniques in order to draw the lane back onto the image.

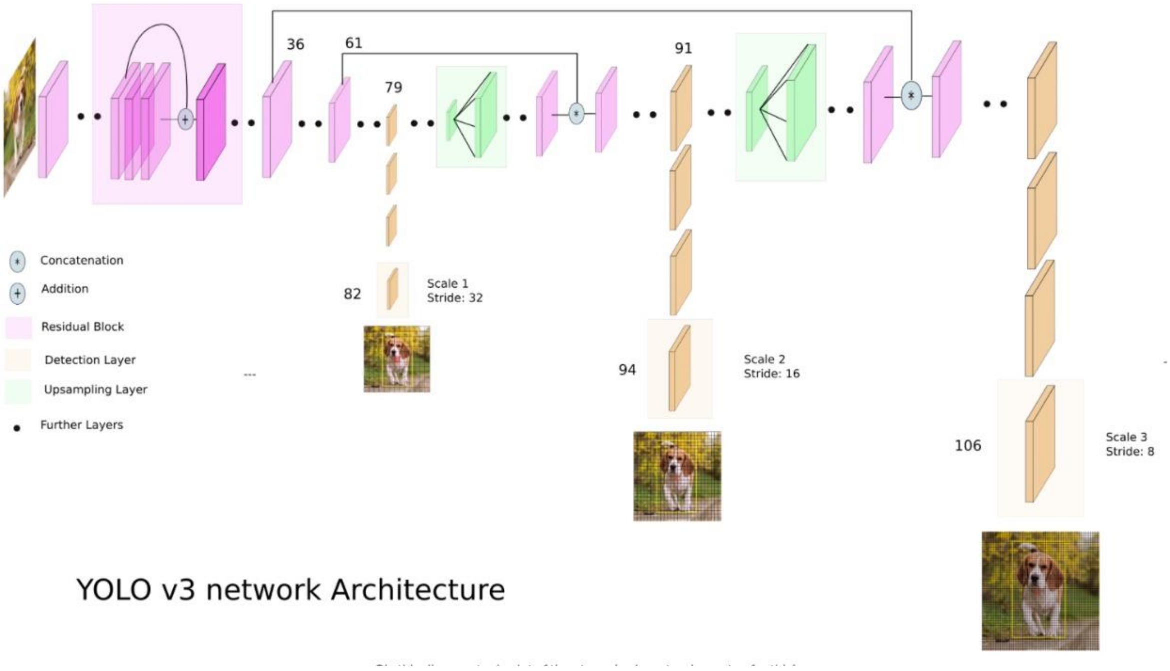
Model Architectures

Applied architecture for Lanes



An architecture based on U-Net (CNN) is deployed. It has 8 convolutional layer and 8 deconvolutional layers. Pooling layers are not considered separate layers. The image segmentation problem can be simply summarized as using the corresponding algorithm to segment the image into an accurate object contour, so as to classify at pixel-level. FCN is the earliest network of deep learning in semantics segmentation. The solution of FCN network is mainly to integrate pool 4, pool 3 and feature map, which can retain image information to a large extent. The advantage of FCN is to segment from end-to-end. U-Net network is also an improvement based on FCNs.

Applied Object Architecture



YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm "only looks once" at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes.

With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes.

CHAPTER 4 SOFTWARE REQUIREMENTS SPECIFICATION-SRS

Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to present detailed description of the project R.O.A.D (Road and Object Detection Based Assistant for Driving) which uses OpenCV and machine learning algorithm to detect and improve lane detection.

Purpose

This document covers the software requirement specifications for project R.O.A.D (**Road and Object Detection Based Assistant for Driving**). The aim of the project is to improve the accuracy and efficiency of the lane detection under different conditions.

R.O.A.D will give a more profound knowledge to picture handling and PC vision and empowering various parts of path discovery and improving the outcomes.

This SRS document contains the features and requirements of R.O.A.D, to serve as a guide to the developers on one hand and a software validation document for the prospective tester on the other.

This document covers all basic features, objectives and attributes of the proposed system. It explains the system's interface, and the constraints under which it must function and how the system will respond to external stimuli. This document will give the user a clear and precise description of the functionality of the R.O.A.D software. For the user, the SRS will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation, testing and maintenance.

OVERALL DESCRIPTION

Product Perspective

Drivers assist system is one of the maximum essential features of current motors to make sure motive force safety and reduce the automobile twist of fate on roads.

A camera is used to capture the view of the road. The front facing camera is mounted on the dashboard or the wind screen to detect the lanes by applying few processes. The system will use computer vision and image processing for lane and object detection. For this purpose machine learning and computer vision techniques will be used which will enable the system to extract lanes and objects.

The proposed machine may be applied on painted and unpainted roads as well as curved and directly roads. This machine doesn't require any more information along with lane width and offset between middle of the lanes.

Product Functions

The major functionalities of R.O.A.D are highlighted below:

1. **Step 1:** getting picture: the created framework can gain pictures from video documents.

2. **Step 2:** parting picture: the framework examinations the parting interaction of video document.

3. **Step 3:** trimming picture: determining the space of interest utilizing crop instrument.

4. **Step 4:** picture improvement: the framework leads the casing to change over RGB shading picture into grayscale picture.

5. **Step 5:** changing over grayscale picture to double picture.

6. **Step 6:** portioning and eliminating objects: utilizing the opening morphological tasks.

7. **Step 7:** characterizing the dissected region inside the picture

8. **Step 8:** the framework works the characterized portion to dissect the Hough line change.

9. **Step 9:** Detect paths for any reason utilizing OpenCV, picture handling and Hough change.

10. **Step 10:** Visualize and display detected lanes on the screen.

User Classes and Characteristics

Our user classes are characterized as follows:

- **Drivers**

It will help drivers to go through lanes much easily under various conditions. It will enable them to assess their driving capabilities much efficiently and accurately.

- **Traffic Authorities**

It will assist traffic authorities to monitor and navigate the flow of traffic. It can also help the authorities to investigate accidents and mishaps.

Operating Environment

- **Hardware**

The essential hardware components required for the proper working of the R.O.A.D software are listed below:

  - Laptop or Desktop
  - Video Camera

- **Software**
    - Python IDE (python 3).
    - Windows / Mac / Linux.
    - Python Libraries: opencv, numpy, matplotlib, tensorflow.
    - For Application Dashboard: Tkinter(Python).

Design and Implementation Constraints

- R.O.A.D will not be able to detect lanes on roads with no visible lane dividers.
- Cameras used as the source of videos need to be still/fixed.
- Too much Distortion in Videos
- Insufficient Luminance
- Bad Video Quality

External Interface Requirements

User Interfaces

The R.O.A.D Desktop Application will give its user with a screen which will allow the user to play the video on which the lane detection is to be done. The video being played would be shown on the dashboard along with frames and marking on which lanes and obstructions are detected.

Hardware Interfaces

- Computers/Laptops
- Camera

Software Interfaces

- Python IDE (python 3)
- Windows / Mac / Linux
- For Application Dashboard: Tkinter(Python)
- Deployment of AI Model on System

Communications Interfaces

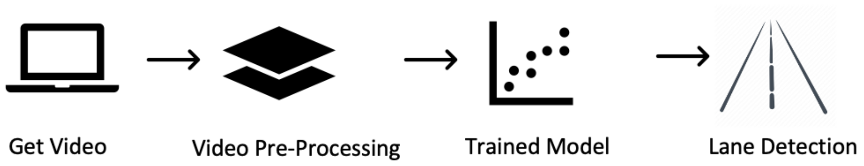
N/A - No communication interfaces are required in our system.

System Features

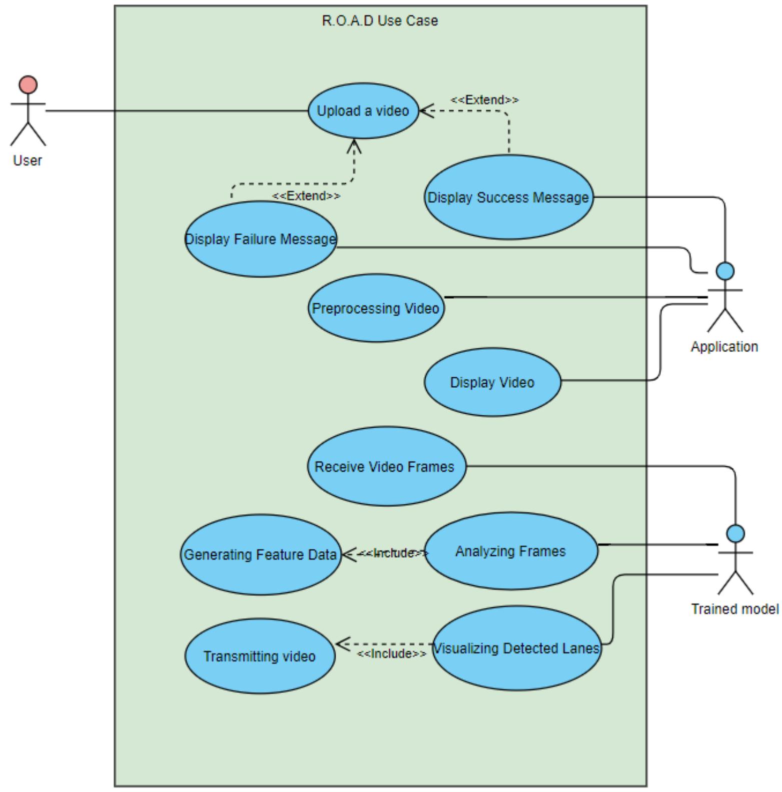
This section illustrates organizing the functional requirements for the project R.O.A.D by system features: -

- Video Acquisition
- Video Processing
- Checking for Lanes
- Providing visualized lane detection

Following is a diagram showing the processes involved:



Overall Use Case Diagram



Video Acquisition

Description

This feature enables the system to acquire video from the user who will upload it into the application. This video will be fed into the system for further processing.

Stimulus/Response Sequences

Normal Path: Video Successfully Uploaded
Preconditions • The video is selected from the local directory and uploaded by the user via the application.
Interactions • The captured video is sent to the system for processing.
Post conditions • Success Message displayed.

Categorization
• **Criticality**: High • **Probability of Defects**: Medium • **Risk**: High

Exceptional Path: Error Message Displayed
Preconditions • The video was not of the correct format or it was too long (max. video length is yet to be decided).
Interactions • An error message is displayed telling the user what are the supported video specs including format and length.
Post conditions • The video upload prompt reappears.
Categorization • **Criticality**: High • **Probability of Defects**: Medium • **Risk**: High

Functional Requirement

1. The system shall be able to acquire videos from the directory's address or link entered by user.
2. If successful, it shall display a Success Message.
3. If not successful, then it shall display a Failure Message and re show the upload prompt.

Video Preprocessing

Description

This feature involves sending the video to the model where it is divided into frames.

Stimulus/Response Sequences

Normal Path: Successfully divided into frames
Preconditions • The video is sent to the system/model for processing.
Post conditions • These frames are, then, fed into the trained model.
Categorization • **Criticality**: Medium • **Probability of Defects**: Medium • **Risk**: Medium

Functional Requirement

1. System shall be able to effectively divide video in to frames as per the specified fps parameter.
2. System shall be able to transmit those video frames to the model.

Check For Lanes

Description

The video frames will be analyzed and features data will be extracted from these frames.

Stimulus/Response Sequences

Normal Path: Model can successfully extract feature data from frames.
Preconditions • The frame images are successfully loaded into the model.
Interactions • Road and lanes detection features are extracted frame by frame.
Post conditions • The model continues to process the next frames.
Categorization • **Criticality**: High • **Probability of Defects**: High • **Risk**: High

Exceptional Path: Model is unable to extract feature data from frames
Preconditions • The frame images are successfully loaded into the model.
Interactions • The model cannot extract features from the frames due to poor image quality or noise in image.
Post conditions • An error message is generated.
Categorization • **Criticality**: High • **Probability of Defects**: Low

• **Risk**: High

Functional Requirement

1. The model shall be able to receive and load video frames into the model.
2. The model shall be able to extract feature data for each frame.
3. In case the frame image quality is poor, then it shall generate an error message.

Visualization

Description

The system shall give a final visualization of lanes detection in video.

Stimulus/Response Sequences

Normal Path: Detected Lanes Visualizations
Preconditions • Features data per frame is generated from the previous stage and analyzed.
Post conditions • The user is notified that the video contains features of lane detections.
Categorization • **Frequency**: High • **Criticality**: High • **Probability of Defects**: Medium • **Risk**: High

Functional Requirement

1. The model shall be able to assess the features data and based on it, give a final visualizations of lanes detections in video.
2. The model shall be able to transmit the results/error back to the application.

Other Non-functional Requirements

- **Performance**

The system will perform lane detection as accurately as possible in manner such that the user should not wait that long until the required processes are being done.

- **Safety**

The system shall not accidently lose/delete the files associated with it, such as information related to drivers and the customers. The system shall not affect the performance of the system.

- **Security**

System shall be secure enough not to breach any security. Videos and their classifications should not be compromised.

- **Ease of Use**

The software shall be easy to use and require no technical training.

- **Software Quality Attributes**
  - ***Usability***

    The graphical UI of framework is to be planned with ease of use as the need. The application will be introduced and coordinated in a way that is both outwardly engaging and easy to understand.

  - **Accuracy**

    To ensure reliability and correctness, the system must attain an acceptable level of accuracy with regards to the classification of lanes and objects in the video. Acceptable Range (85% +).

  - **Manageability**

    Ensure efficient control of the system and to keep it fully operational.

  - **Reliability**

Reliability is defined as providing the user, correct information (video and frame classification) when they need it.

- **Business Rules**
  - The uploaded video should exist in a specified directory.
  - The video which is uploaded must not be more than the supported length.
  - The video which is uploaded must be of minimum or higher quality.

CHAPTER 5 SOFTWARE DESIGN DOCUMENT – SDD

Introduction

The introduction of the Software Design Specification (SDS) document provides an overview of the entire SDS with purpose, scope, definitions, acronyms, abbreviations, and references. The aim of this document is to present, in detail, the functional and non-functional aspects of the project R.O.A.D which uses image processing techniques and deep learning to detect road lanes in videos. The detailed descriptions and visualizations of R.O.A.D are provided in this document.

Purpose

This software design document describes the architecture and system design of project R.O.A.D, Version 1.0. The document is meant to provide details about the features design and requirements of R.O.A.D, to fill in as a manual for the engineers on one hand and a product approval report for the likely customers on the other. This record incorporates classes and their between connections, use cases with point by point portrayals, arrangement outlines and different stream graphs.

Scope

R.O.A.D will collect the data from the mounted front view camera and applying image processing and object detection to distinguish lanes, vehicles and alert the driver.

The overall scope of the project includes:

- Image capturing
- Image processing (edge detection, noise reduction)
- Python (openCV, numpy)
- Output with detected road and lanes.

Work Breakdown Structure

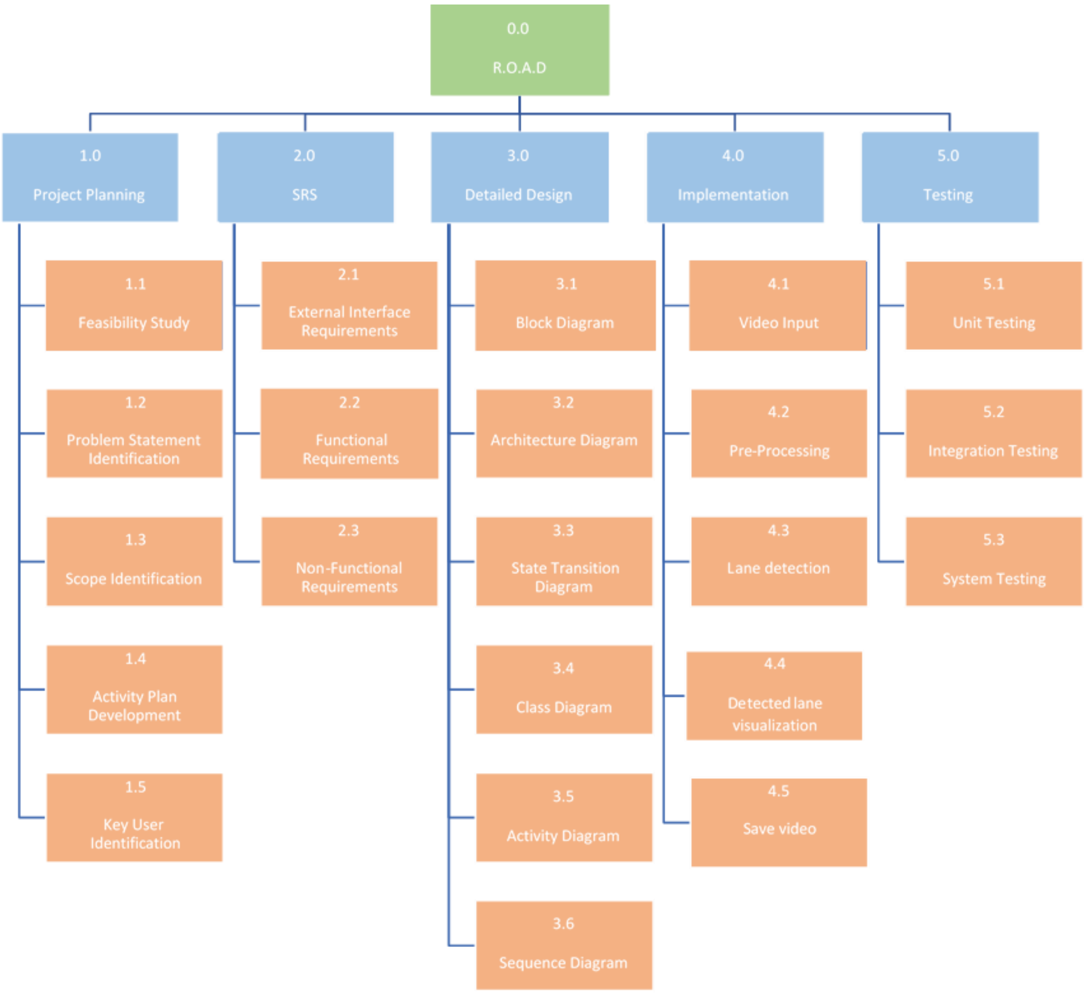


Figure 1Work Breakdown Structure

System Overview

The main objective of our system is to create a desktop application which assists drivers to detect lane and objects while driving. The system will take a video input from the user. After uploading the video lane detection starts, image pre-processing is applied which includes frame extraction, reducing image noise and implement trained model on the extracted frames. After successfully applying trained model the detected lane is super imposed onto the original input video and the detected lane is displayed in the output video along with the surrounding objects.

The system consists of 5 major modules:

- Video Acquisition module
- Pre-Processing module
- Trained models
- Lane Detection module
- Visualization module

System Architecture

This section covers the overall architectural description of R.O.A.D. It encompasses the high-level and low-level descriptions of the project including block diagrams of the application and the deep learning model. Moreover, a complete object-oriented description which includes class diagrams, sequence diagrams and others. Finally, the rationale for the design pattern is provided.

Architectural Design

We are using layered architecture. The modules have a sequential relationship such that the input is dependent on its predecessor module. The video acquisition module will take input and then it will be processed by the Pre-processing module which will extract frames and apply pre-processing techniques. Then the model is applied on the frames for lane detection and visualize it.

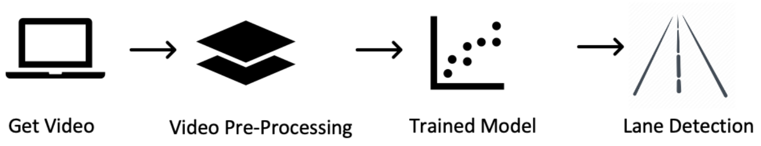


Figure 2Architecture Design

Block Diagram

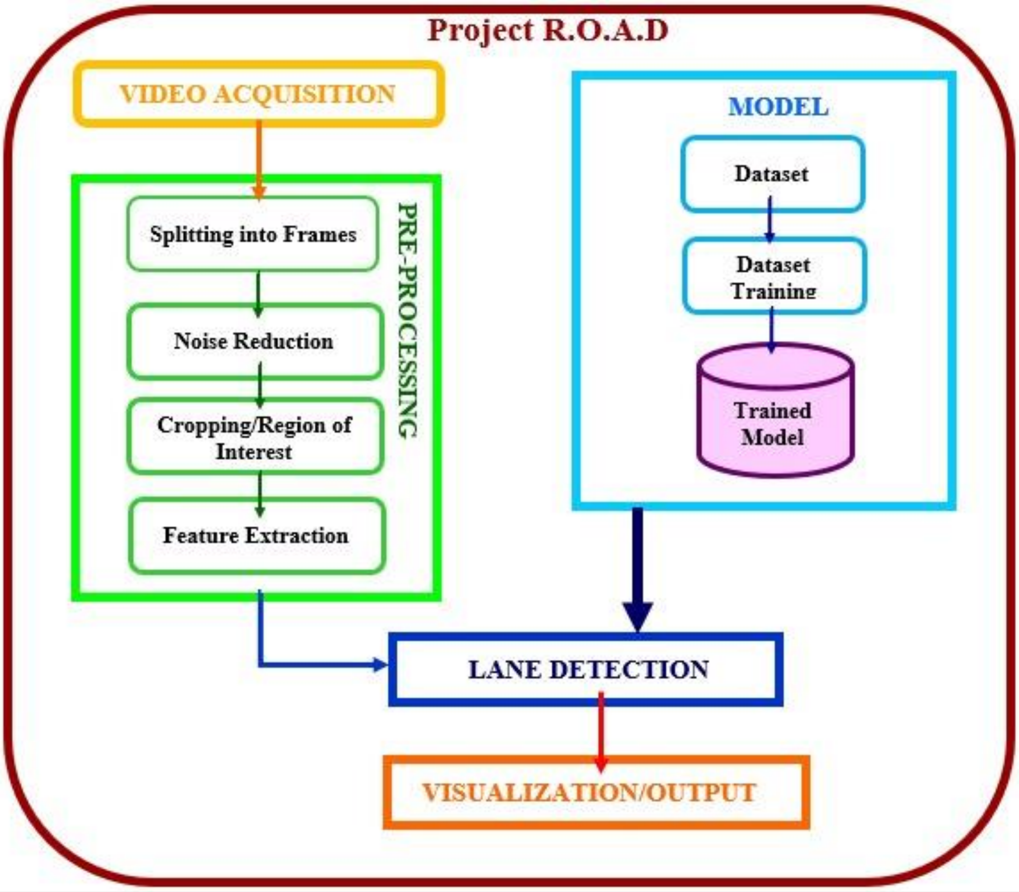


Figure 3Block Diagram

- Video Acquisition enables the system to acquire video from the user who will upload it into the application. This video shall be fed into the system for further processing.
- Pre-Processing module involves sending the video to the model where it is divided into frames.
- Model module frames shall be analyzed, and features data will be extracted from the frames.
- Visualization module shall give a final visualization of lanes detection in video.

Decomposition Description

Component Diagram

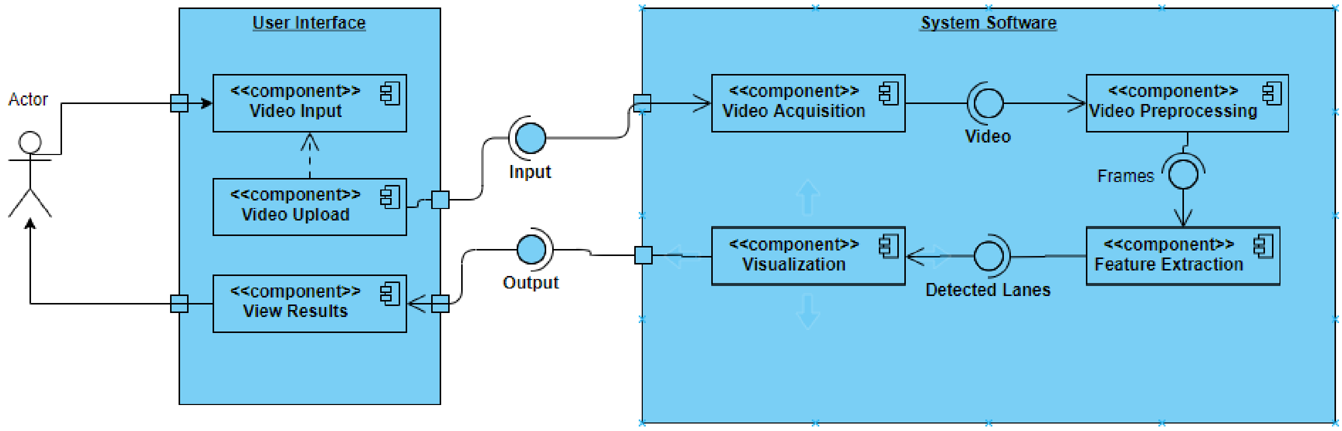


Figure 4Component Diagram

Class Diagram

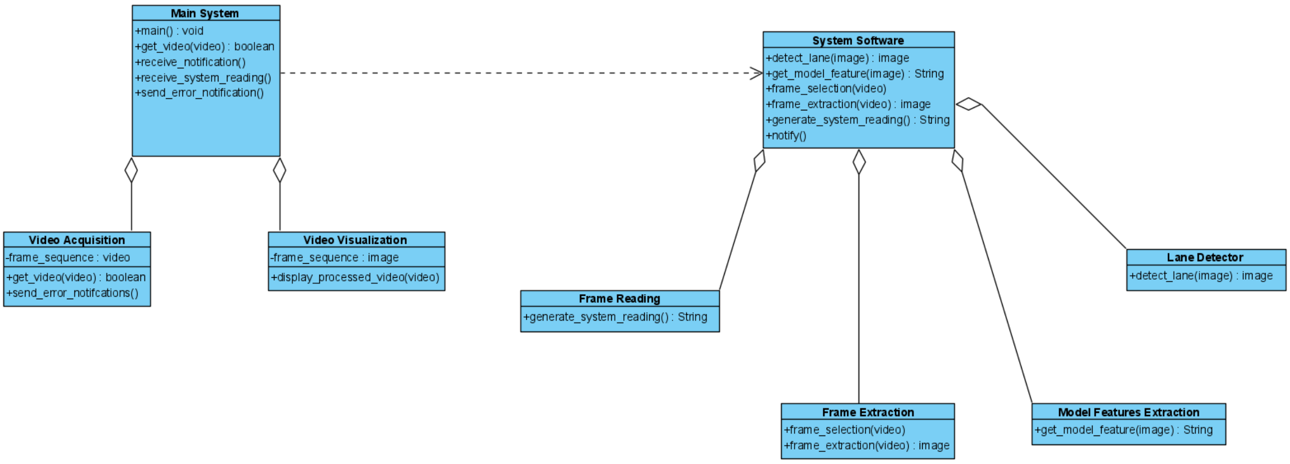


Figure 5Class Diagram

Class Description

Class name	Description
Main System	It is main class that starts when the user interacts with the system. It receives notification and readings from System Software for lane detection.
Video Acquisition	This class gets video from the user.
Video Visualization	This class displays video to the user.
Model Features Extraction	This class uses extracted frames and performs feature selection technique.
Frame Extraction	This class extracts frames from the video which will be then further used.
Lane Detector	This class will classify and implement the video lane detection methods.
Frame Readings	This class gets readings about the lane detection frames in the video and saves the data.

System Software	It receives video from the Main System and then uses its methods to detect lanes.

Use Case Diagram

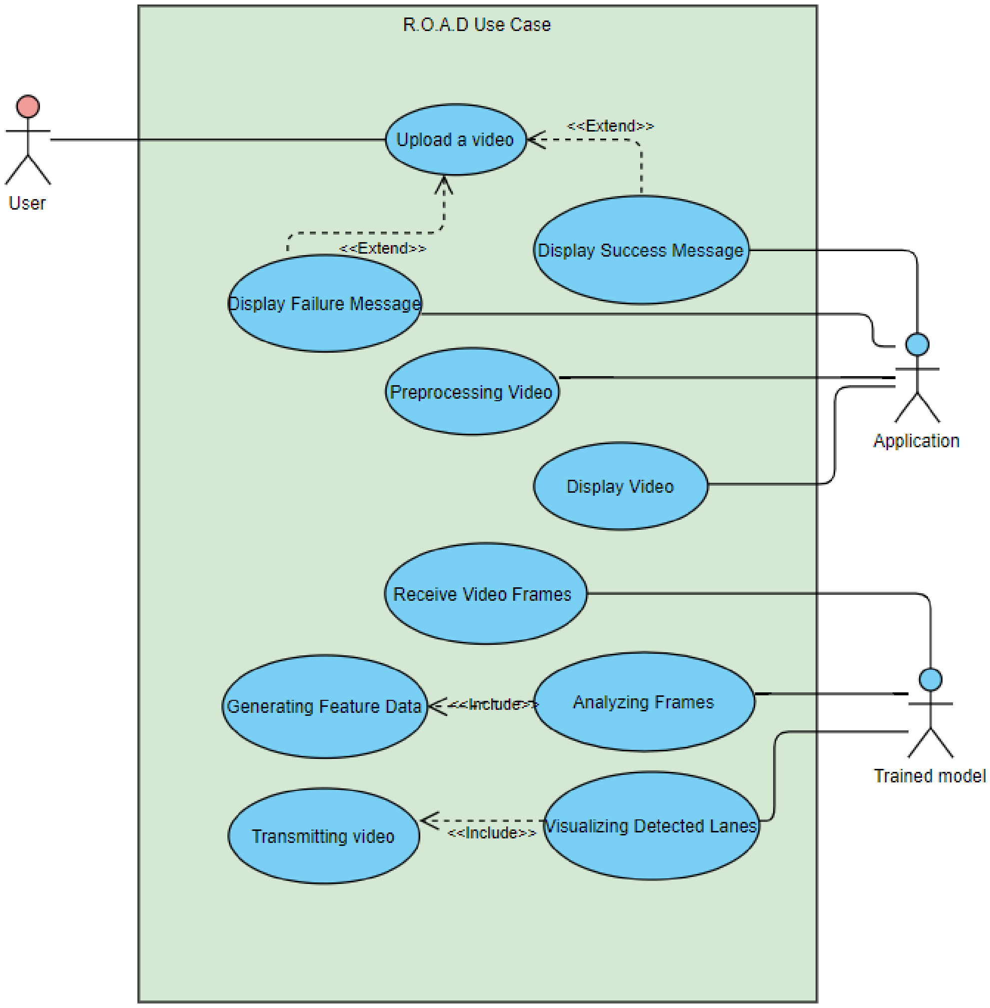


Figure 6Use Case Diagram

Activity Diagram

Video Acquisition

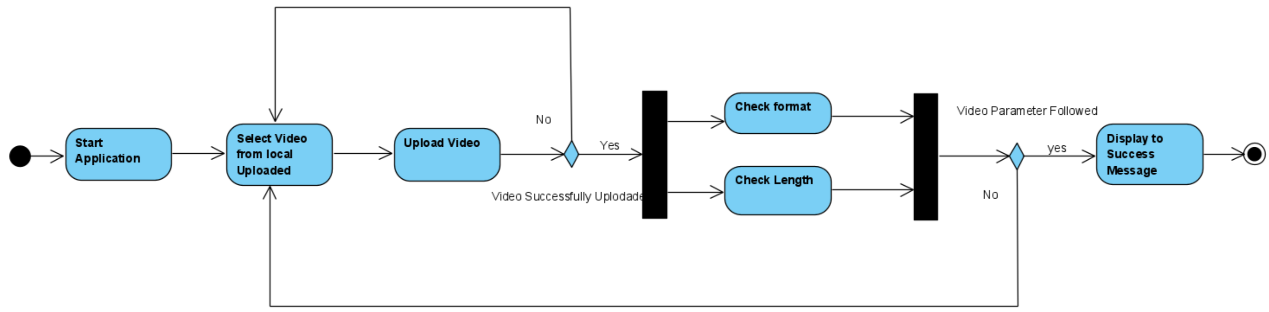


Figure 7Activity Diagram acquisition

Video Preprocessing

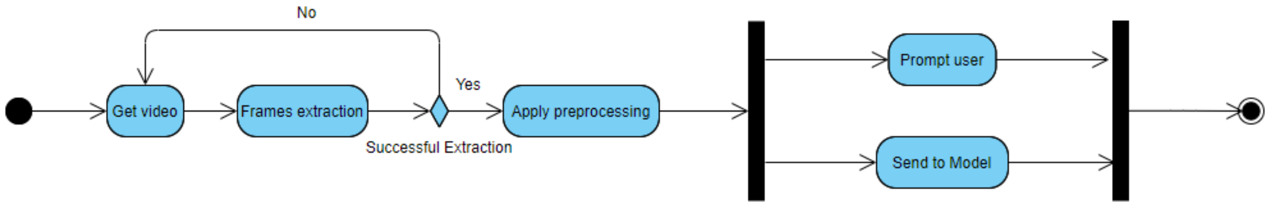


Figure 8Activity Diagram video processing

Check for Lanes

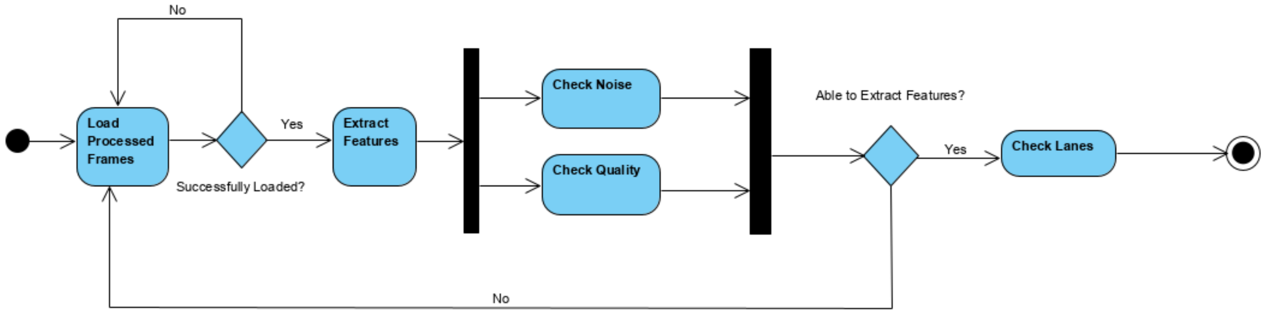


Figure 9Activity Diagram check for lanes

Visualization

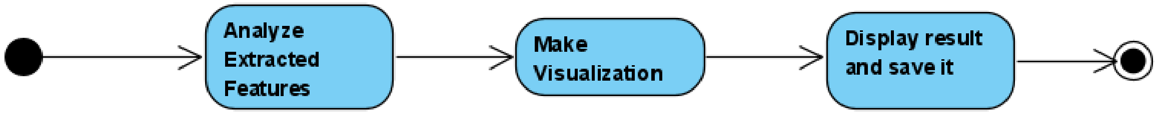


Figure 10Activity Diagram visualization

Sequence Diagram

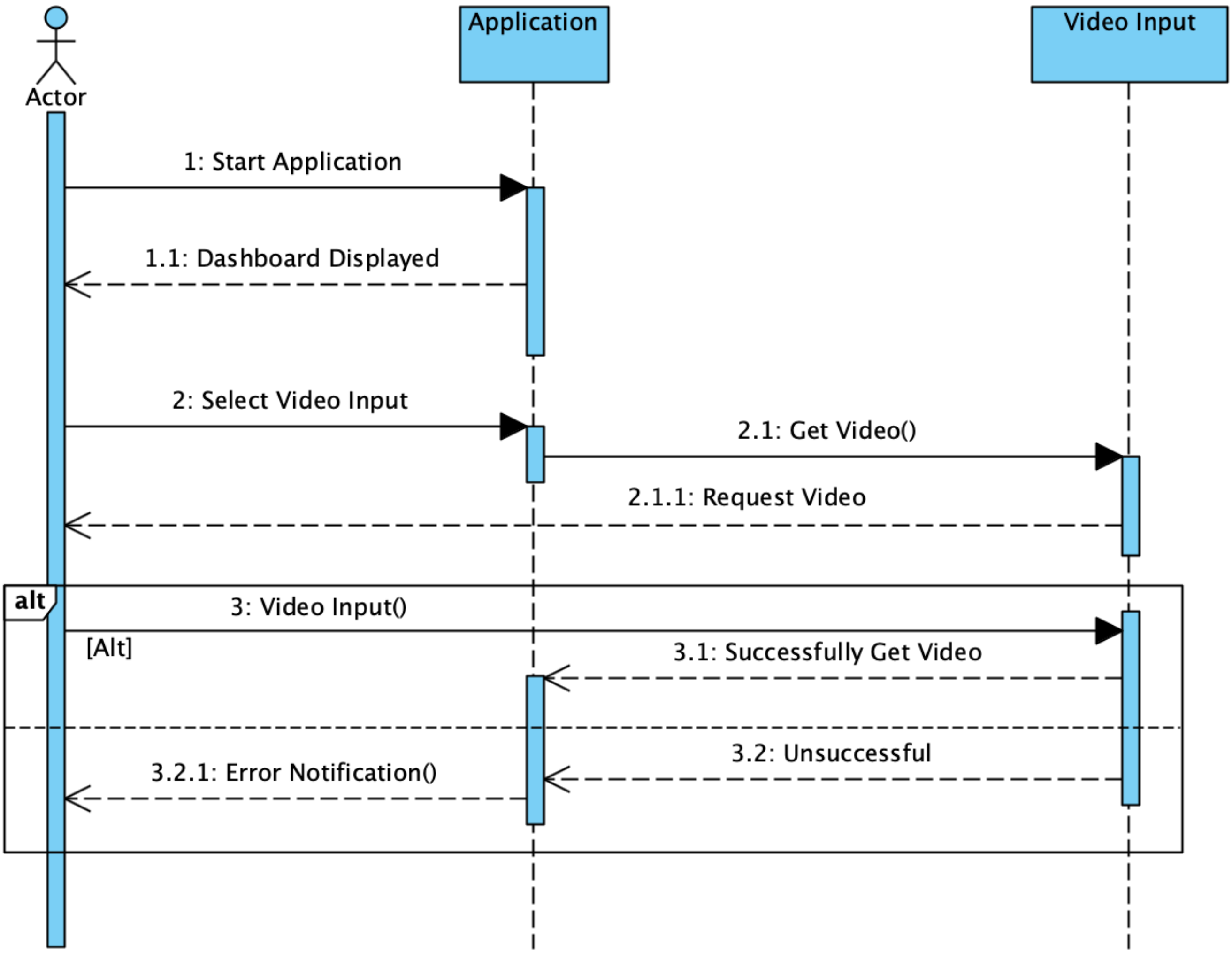


Figure 11sequence diagram video acquisition

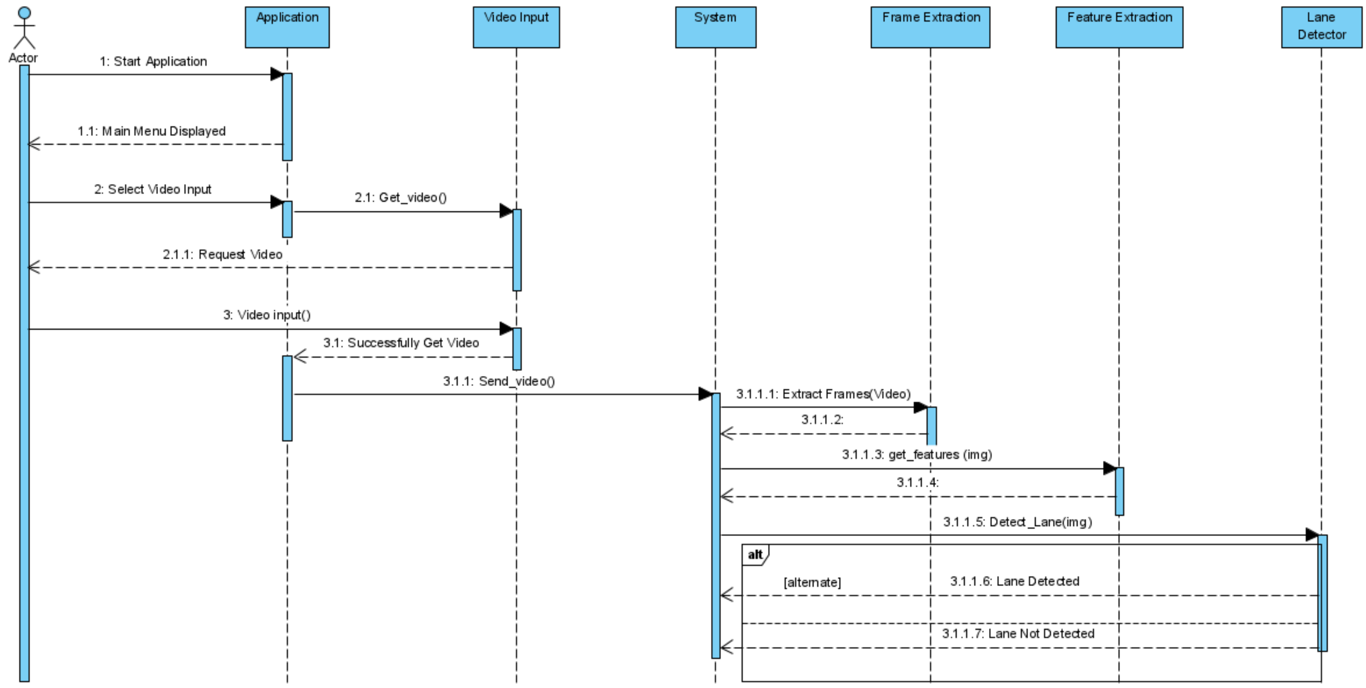


Figure 12sequence diagram preprocessing

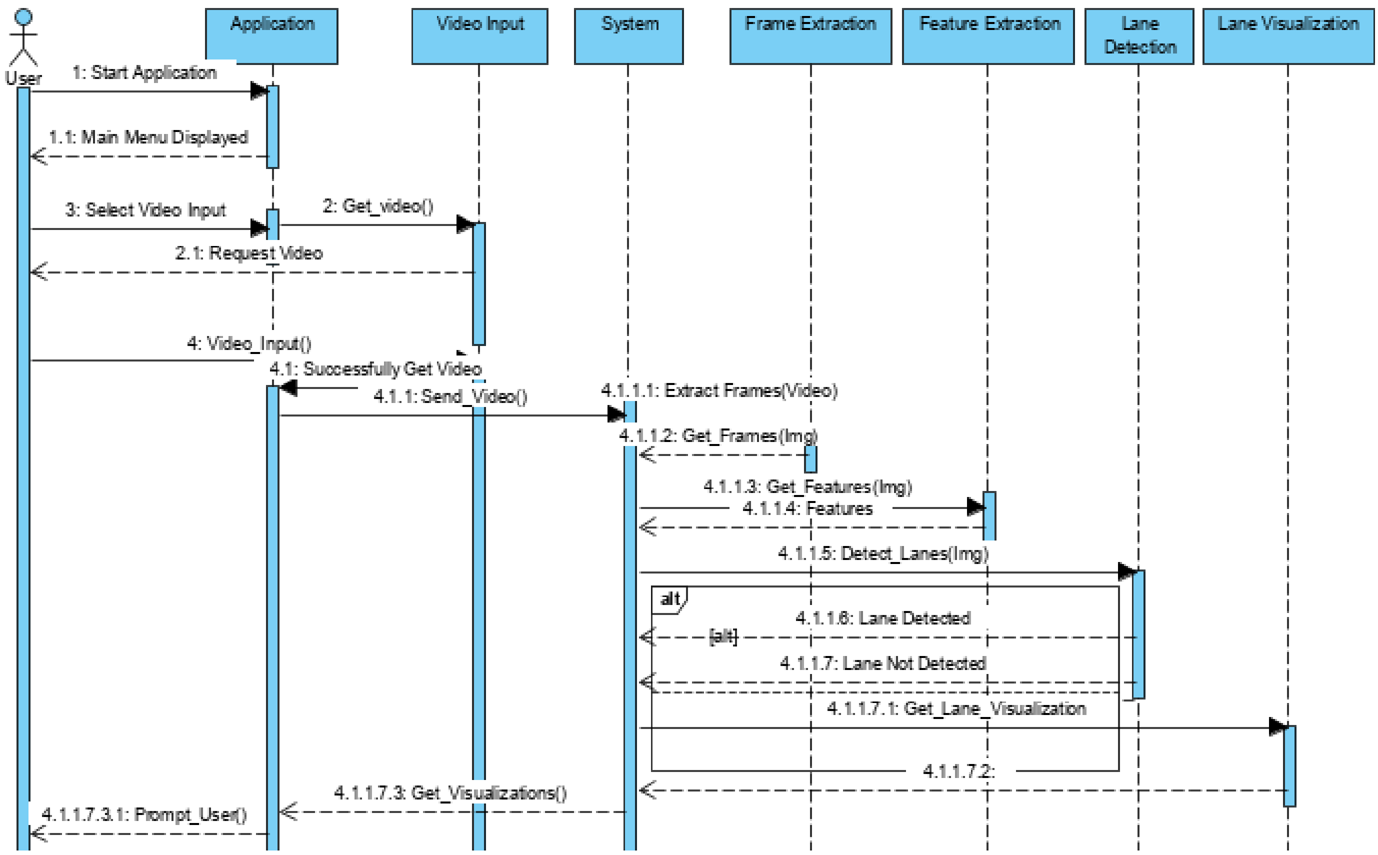


Figure 13sequence diagram video acquisition

Design Rationale

We have chosen Layered engineering this model can be used to structure programs that can be broken down into subtasks, all of which is at a particular level of reflection. Each layer offers types of assistance to the following higher layer.

Following are the reasons for Layered architecture:

The idea of layered engineering is not difficult to learn and carry out.

The layers alongside the general code association is steady across every one of the layered ventures.

Every one of the articles are kept together. Along these lines, when you need to change something in a few or every one of the objects of a specific kind, it is simpler to rapidly discover an article.

Data Design

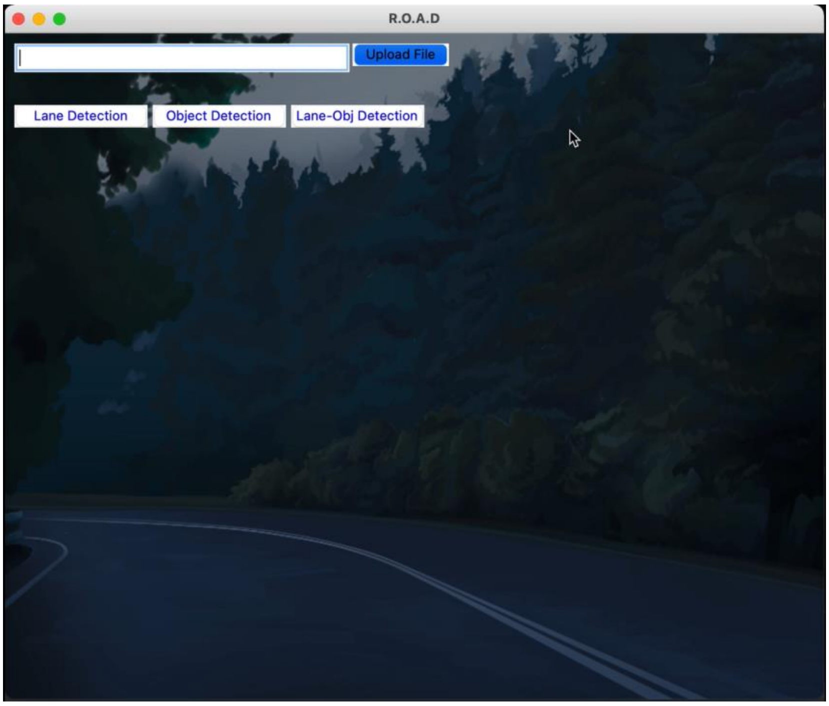
Data Dictionary

Dataset Name	Details	Source
Urban Unmarked	98 – training 100 – testing	http://cvlibs.net/datasets/kitti/eval_road.php
Urban Marked	95 – training 96 – testing	http://cvlibs.net/datasets/kitti/eval_road.php
Urban Multiple Marked Lanes	96 – training 94 – testing	http://cvlibs.net/datasets/kitti/eval_road.php
Urban	combination of the three above	http://cvlibs.net/datasets/kitti/eval_road.php
Weather and light Conditions	17.4% were crisp evening driving, 16.4% were blustery daytime driving, and 66.2% were shady evening time driving.	https://www.dropbox.com/s/rrh8lrdclzlnxzv/full_CNN_train.p?dl=0

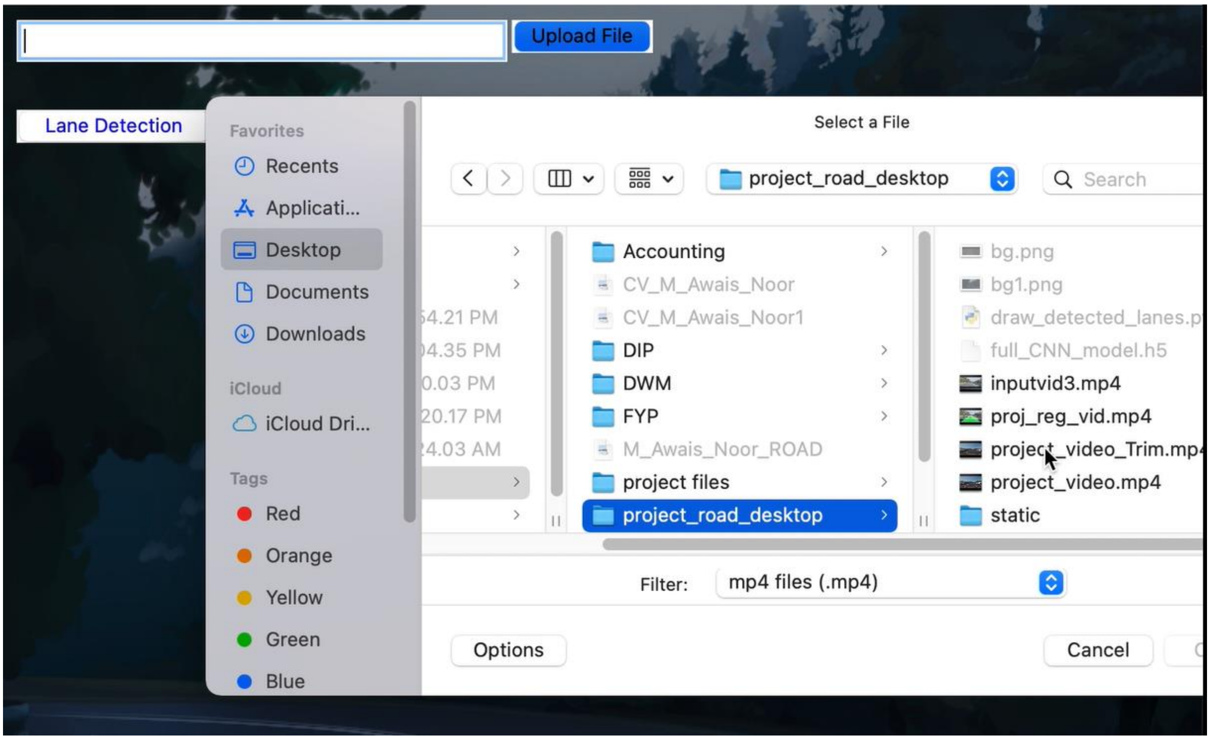
User Interface

Following visualization shows user interface for R.O.A.D application dashboard and video upload.

- **Video Acquisition:**

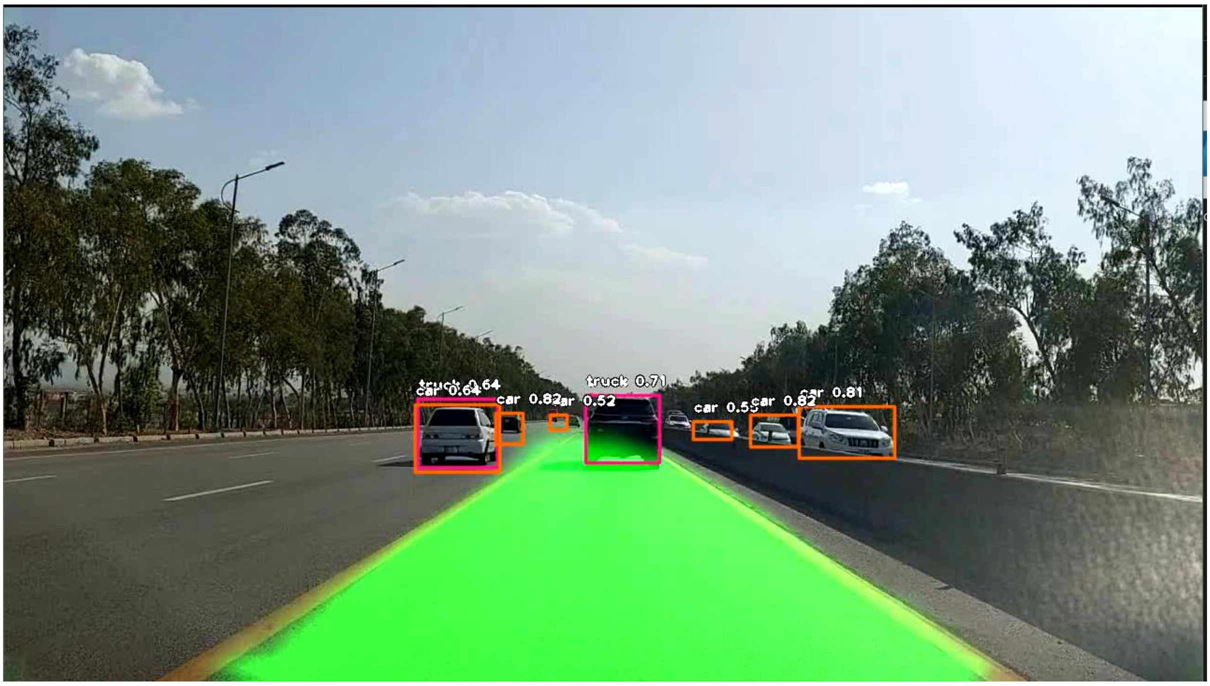


- **Select Video from Directory:**



- **Lane Detection:**

- **Object detection:**



Pseudo Code

- ***Pseudo code for main program***

```
CLASS Lanes()
FUNCT main() {
      LOAD CNN_model
      CREATE Lanes() Object lanes
      INPUT  path_input_video
```

```
      Clip1 = Video_acquisition( path_input_video)

      Img = SPLIT  TO FRAMES (clip1)

      CALL Funct Road_lines( img )

      OUTPUT Video (path_output_video)

}

FUNCT Road_lines( img ) {

      RESIZE img

      UNNORMALIZE img

      PREDICT.CNN_model

      APPEND prediction to List

      CALCULATE Average Detection


      CREATE Lane_markings with RGB Dimensions

      PADD R,B dimensions

      STACK G dimension

      RESIZE img


      Result = MERGE Lane_markings WITH img

      RETURN result

}
```

- ***Pseudo code for cnn model***

```
FUNCT Create_model (input_shape, pool_size) {

      Model = CREATE NeuralNetwork
```

```
      NORMALIZE input_shape

      CNN_Layer(model)

      POOLING(model)

      DECONVOLUTION(model)

      UPSAMPLE(model)


RETURN model

}


FUNCT main() {

      LOAD dataset

      LOAD labels

      NORMALIZE dataset

      SPLIT INTO training/testing sets

      INT Batch_size =128

      INT epoch = 10

      Pool_size = (2,2)

      Input_shape = X_train.shape[1 :]

      CREATE Model (CNN_model)

      WHILE training = FALSE DO

TRAIN model

      COMPILE model

      SAVE model

}
```

Requirement Matrix

Functional Requirement ID#	Functional Requirement / Use Case	Priority	Test Case ID#	Execution Status	Defect#
FR_1	Video Acquisition	High	T 001	Complete	2
FR_2	Video Processing	High	T 002	Complete	4
FR_3	Lane Detection	High	T 003	Complete	6
FR_4	Lane Visualization	High	T 004	in progress	-

CHAPTER 6 SYSTEM TESTING

Introduction

This test plan file describes the appropriate techniques, method and methodologies used to plot, execute, and manage checking out of the "ROAD". The test plan will ensure that ROAD meets the driver requirements at an accredited level.

Manual Testing will be followed which incorporates testing a product physically, i.e., without utilizing any robotized device or any content. In this kind, the analyzer assumes control over the job of an end-client and tests the product to distinguish any sudden conduct or bug. Every Unit will be tried independently and afterward will be coordinated with different units; along these lines, Unit Testing and Integration testing will be followed. For every unit, Black box Testing is done and for consolidated Unit Acceptance Testing is finished.

The test scope incorporates the Testing of all useful, application execution and use cases prerequisites recorded in the necessity report.

Programming testing, contingent upon the testing technique utilized, can be carried out whenever in the improvement cycle. Nonetheless, most of the test exertion happens after the prerequisites have been characterized and the coding interaction has been finished.

Approach

After all test cases are executed, a test report will be summarized to show the quality of ROAD. Following test approaches will be used in test execution:

1. **Unit test**. Developers are chargeable for unit testing. The implementation of each module and individual factor can be proven one by one.

2. **Integration test**. After the unit test is exceeded above the described satisfactory threshold, testers will execute the integration check cases. After all the modules are included, it's miles crucial to test the product as a black box.

3. **Positive and negative testing design technique:** Here we have to design specific test cases in certain situations. This will in turn show if all the functional requirements are being fulfilled. Invalid situations will also be tested to see if the system can judge them.

Features to be tested:

Following are the features that need to be tested:

1. Only mp4 video format videos must be submitted Video resolution must be of 1280 x 720 or lower.

2. Pre-processing is applied on the video frames.
3. Lane detection model is applied, and the results are generated.
4. Object detection model is applied, and the results are generated.

Pass/fail criteria:

Details of the test cases are specified in phase Test Deliverables. Following the ideas outlined under, a check item would be judged as skip or fail.

1. Preconditions are met.

2. Inputs are finished as detailed.

3. The result works as what laid out in output => Pass.

Testing tasks

- Develop test cases.

- Execute assessments based totally on the evolved test cases for the software.

- Report defects from the executed check instances if any.

- Provide whole check report.
- Incorporate or manage changes later in the level of the project development.

Training needs

Basic information on testing procedures and strategies is required for the testing of the undertaking.

Methods, for example, Black Box testing, coordination testing ought to be known to designers. Every one of the engineers will test each other's work and will be effectively partaking in the turn of events and testing of the undertaking all the while.

Risks and Contingencies

Schedule Risks:

Proper WBS should be followed as given in the document along with proper management of time or the project will be at risk of falling behind schedule.

Operational Risks:

Proper meetings and goals should be designed in accordance with schedule in order to make sure that testing goes according to plan. There should not be any communication gap.

Technical Risks:

Expertise of group members will be taken into account in order to minimize technical risks.

Programmatic Risks:

The scope of the project will be kept in accordance with the degree limitations to manage this.

Test Cases

Table 6. 1 Video Acquisition

Test Case Number	01
Test Case Name	Video Acquisition
Description	User uploads a video in the application.
Testing Technique	Component testing, Black Box Testing
Preconditions	Application should be open
Input Values	Enter video directory(path) and click “Upload”
Steps	1. Open the Application. 2. Main Screen is open. 3. Enter video directory(path) 4. Click on ‘Upload’ button.
Expected output	Video should be uploaded
Actual output	Video is uploaded successfully.
Status	Test case passed successfully.

Table 6. 2 Video Processing

Test Case Number	02
Test Case Name	Video Processing
Description	This feature involves sending the video to the model where it is divided into frames.
Testing Technique	Component testing, Unit Testing
Preconditions	Video path should be selected, and upload button should be clicked.
Input Values	User clicks the upload button for the selected video
Steps	1. Open the Application. 2. Main Screen is open. 3. Enter video directory(path) 4. Click on ‘Upload’ button.
Expected output	Video should be able to be processed in the model
Actual output	Video is successfully Processed.
Status	Test case passed successfully.

Table 6. 3 Lane Detection

Test Case Number	03
Test Case Name	Lane Detection
Description	The video frames will be analyzed, and features data will be extracted from these frames.

Testing Technique	Unit testing
Preconditions	The frame images are successfully loaded into the model.
Input Values	User clicks the upload button for the selected video
Steps	1. Enter video directory(path) 2. Click on ‘Lane Detection’ button.
Expected output	Model should be able to detect lanes.
Actual output	Model is able detects lanes.
Status	Test case passed successfully.

Table 6. 4 Object Detection

Test Case Number	05
Test Case Name	Object Detection
Description	The system shall detect objects in the video as well along with the lanes.
Testing Technique	Integration testing, Black Box Testing, Unit testing
Preconditions	Detect Lane and Object button clicked
Preconditions	The frame images are successfully loaded into the model.
Steps	1. Enter video directory(path) 2. Click on ‘Object Detection’ button.
Expected output	Processed video should be able to detect objects
Actual output	The processed video detects objects.
Status	Test case passed successfully.

Table 6. 5 Video Visualization

Test Case Number	04
Test Case Name	Video Visualization
Description	The system shall give a final visualization of lanes detection in video.
Testing Technique	Integration testing, Black Box Testing
Preconditions	Features data per frame is generated from the previous stage and analyzed.
Input Values	Click on ‘Lane Detection’. Click on ‘Object Detection’.
Steps	1. Open the Application. 2. Main Screen is open. 3. Enter video directory(path) 4. Click on ‘Upload’ button. 5. Click on ‘Lane Detection’. 6. Click on ‘Object Detection’.
Expected output	Processed video should be able to run lane detection and object detection.
Actual output	Lane detection is successfully detected in the new processed video. Object detection is successfully detected in the new processed video.
Status	Test case passed successfully.

CHAPTER 7 ANALYSIS

R.O.A.D was aimed at simplifying the automated lane and object also making it user friendly. At the same time, in such systems, accuracy has to be taken into account as it is of the key importance. For this, thorough quality and accuracy check was performed regularly on our dataset so that we get the best possible accuracy.

Here we implemented two different Convolutional Neural Networks for lane and object detection. For lane detection we used a Neural Network which was based on the architecture of UNET. UNET architecture yielded the best accuracy and validation results as compared to other well-known algorithms which are applied for lane detection.

Algorithm	Epochs	Accuracy	Mean Square Error
UNET	20	87%	0.0046

The above matrix shows that our lane model achieved the minimum acceptable accuracy measure which was set at 85%.

CHAPTER 8 FUTURE WORK

A system of this magnitude always needs continuous work to evolve. There are a lot of possible changes and additions that can be done to the system to improve its performance and functionalities. The system has been made in a modular fashion which enables integrating new features very easy.

Project R.O.A.D can be extended to include the following features:

- Real-time detection could be implemented.
- To better suit Pakistani traffic scenarios, the object detector could be modified to detect rickshaws and donkey-carts separately as well.
- Roads with a large number of bumps, holes and cracks can also be considered while detecting lanes.
- An 'over-take detector' can be added, which can help to detect any possibility of an over-take by observing the driving trends of other cars in front due to the unpredictable driving of Pakistani people.

CHAPTER 9 CONCLUSION

The purpose of the application is to provide assistance in self-driving cars, by not only detecting the current lane of the driver but also the surrounding objects on the road and observable-environment such as pedestrians, vehicles and traffic lights etc. It will also help the drivers to drive safely. Thus, the driver will be able to overcome different difficulties while driving and improve the journey experiences. The system consists of a desktop application which provides efficient detection of lanes and the objects, which are encountered during driving. Applying image processing and object detection to distinguish lanes, vehicles and alert the driver. R.O.A.D will provide a deeper insight to image processing and computer vision and enabling different aspects of lane and object detection and improving the results.

BIBLOGRAPHY

[1] https://www.ripublication.com/irph/ijeee_spl/ijeeev7n6_09.pdf [15/10/20]

[2] https://keras.io/[20/10/20]

[3] https://ieeexplore.ieee.org/record/9045804 [23/10/20]

An Overview of Recent Progress of Lane Detection for Autonomous Driving - IEEE Conference Publication

IEEE Xplore, conveying full content admittance to the world's greatest specialized writing in designing and innovation. | IEEE Xplore

[4] https://ieeexplore.ieee.org/record/6491712 [30/10/20]

Path identification dependent on order of path mathematical model - IEEE Conference Publication

IEEE Xplore, conveying full content admittance to the world's greatest specialized writing in designing and innovation. | IEEE Xplore

[5] https://ieeexplore.ieee.org/record/4580573 [10/11/20]

Ongoing path identification for self-governing vehicles - IEEE Conference Publication

IEEE Xplore, conveying full content admittance to the world's best specialized writing in designing and innovation. | IEEE Xplore

[6] https://ieeexplore.ieee.org/report/8807727 [16/11/20]

Path Detection and Tracking For Intelligent Vehicles: A Survey - IEEE Conference Publication

IEEE Xplore, conveying full content admittance to the world's best specialized writing in designing and innovation. | IEEE Xplor

[7] https://www.Cs.Cmu.Edu/~efros/publications/LBMV07/Papers/viola-cvpr-01.Pdf

Rapid Object Detection the usage of a Boosted Cascade of Simple Features by means of Paul Viola & Michael Jones

[8] Cho, Jae-Hyun, Young-Min Jang, and Sang-Bock Cho. "Lane popularity algorithm the usage of the Hough transform with implemented accumulator cells in multichannel ROI." In Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on, pp. 1- 3. IEEE, 2014.

[9] Low, Chan Yee, HairiZamzuri, and SaifulAmriMazlan. "Simple Robust Road Lane Detection Algorithm”. IEEE, 2014.

[10] https://www.Hindawi.Com/journals/am/2018/8320207/

[11] C. Kreucher, S. Lakshmanan, and K. Kluge, “A motive force caution device primarily based on the LOIS lane detection set of rules,” in Proceedings of IEEE International Conference on Intelligent Vehicles, 1998, pp. 17–22.

[12] M. Chen, T. Jochem, and D. Pomerleau, “AURORA: a imaginative and prescient-primarily based roadway departure warning device,” in 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95. “Human Robot Interaction and Cooperative Robots”, Proceedings, 1995, vol. 1, pp. 243–248 vol.1

[13] https://www.Sciencedirect.Com/technological know-how/article/pii/S240595952030240X#b11

APPENDIX A

PALAGARISM REPORT

thesis

ORIGINALITY REPORT

20%	16%	5%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	www.diva-portal.org Internet Source	2%
3	fics.nust.edu.pk Internet Source	2%
4	medium.com Internet Source	2%
5	Submitted to Higher Education Commission Pakistan Student Paper	2%
6	www.irjet.net Internet Source	1%
7	naadispeaks.wordpress.com Internet Source	1%
8	hackernoon.com Internet Source	1%
9	www.opfro.org Internet Source	1%

10	Submitted to Australian Institute of Higher Education	Student Paper	<1%
11	Submitted to Visvesvaraya Technological University, Belagavi	Student Paper	<1%
12	irep.iium.edu.my	Internet Source	<1%
13	Arthur Ahmad Fauzi, Fitri Utaminingrum, Fatwa Ramdani. "Road surface classification based on LBP and GLCM features using kNN classifier", Bulletin of Electrical Engineering and Informatics, 2020	Publication	<1%
14	Submitted to NCC Education	Student Paper	<1%
15	{ HYPERLINK "http://www.ijritcc.org/" \h }	Internet Source	<1%
16	Submitted to Oakland Community College	Student Paper	<1%
17	{ HYPERLINK "http://www.hindawi.com/" \h }	Internet Source	<1%
18	{ HYPERLINK "http://www.ijariit.com/" \h }	Internet Source	<1%
19	Submitted to The Millennium Universal College - TMUC		<1%

Student Paper

20	Submitted to Nizwa College of Technology Student Paper	<1%
21	downloads.hindawi.com Internet Source	<1%
22	ieeexplore.ieee.org Internet Source	<1%
23	Submitted to Colorado Technical University Online Student Paper	<1%
24	Lipismita Panigrahi, Kesari Verma, Bikesh Kumar Singh. "Hybrid segmentation method based on multi – scale Gaussian kernel fuzzy clustering with spatial bias correction and region – scalable fitting for breast US images", IET Computer Vision, 2018 Publication	<1%
25	Submitted to Manipal University Student Paper	<1%
26	Submitted to The University of Manchester Student Paper	<1%
27	A. Al Mamun, P. P. Em, J. Hossen. "Lane marking detection using simple encode decode deep learning technique: SegNet", International Journal of Electrical and Computer Engineering (IJECE), 2021 Publication	<1%

28 doaj.org <1%
Internet Source

29 hdl.handle.net <1%
Internet Source

30 Khalid Isa. "Experimental Studies on Dynamics Performance of Lateral and Longitudinal Control for Autonomous Vehicle Using Image Processing", 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, 07/2008 <1%
Publication

31 { HYPERLINK "http://www.instructables.com/" \h } <1%
Internet Source

32 { HYPERLINK "http://www.studymode.com/" \h } <1%
Internet Source

33 Submitted to Liverpool John Moores University <1%
Student Paper

34 Submitted to University College London <1%
Student Paper

35 lorien.die.upm.es <1%
Internet Source

36 Tommy Koens, Erik Poll. "Chapter 1 Blockchain Utility in Use Cases: Observations, <1%

Red Flags, and Requirements", Springer Science and Business Media LLC, 2021
Publication

---

37 Gang-Len Chang, Thanavat Junchaya, Alberto J. Santiago. "A REAL-TIME NETWORK TRAFFIC SIMULATION MODEL FOR ATMS APPLICATIONS: PART I—SIMULATION METHODOLOGIES", I V H S Journal, 1994 <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		