Smart Crosswalk Integration For Smart City Infrastructure Using AI

By

Fasih Ur Rehman

Enrollment No. 01-133222-020

Maaz Bin Rashid

Enrollment No. 01-133222-033

Supervised By

Engr Asim Altaf Shah

Session 2022-26

A Report is submitted to the Department of Electrical Engineering,
Bahria University, Islamabad.
In partial fulfillment of requirement for the degree of BS(EE).

Certificate

We accept the work contained in this report as a confirmation to the required standard for the partial fulfillment of the degree of BS(EE).

____________________.

Head of Department

____

Supervisor

____________________.

Internal Examiner

____

External Examiner

Dedication

This thesis is dedicated to our beloved parents and family members, whose unconditional support, prayers and encouragement have been the cornerstone of our academic journey. Their sacrifices, patience and tireless motivation has inspired us to do our best and to successfully complete this work.

Finally, we also dedicate this project to our friends and colleagues who provided us with moral support and encouragement during the development of this Smart Crosswalk System. Their cooperation and belief in us made us more determined to reach this milestone.

Acknowledgments

We would like to extend our sincere thanks to our highly respected supervisor, Engr. Asim Altaf Shah, for his constant guidance, valuable suggestions and constructive feedback throughout this project. I am indebted to his technical expertise, encouragement and supervision for shaping the direction and quality of this research work.

We also thank all faculty members who provided technical knowledge and support during this project. Special thanks are due to our friends and classmates for their cooperation and assistance in testing and implementing the system components.

Finally, we would like to thank our family for their patience, understanding and steadfast support in the completion of this thesis.

Abstract

Urgent attention should be paid to the safety of pedestrians at crosswalks in the face of growing traffic density and the lack of effective passive road infrastructure. This work proposes, develops and tests an edge-computed Smart Crosswalk System using real-time computer vision to adaptively control traffic intersections. The proposed system design is based on an NVIDIA Jetson Nano 4GB in combination with a YOLOv5n deep learning algorithm, with an optimised TensorRT implementation for edge computing. The computer vision system scans a Region of Interest (ROI) and identifies people and vehicles. When a perceived threat is detected, the unit activates an ESP32 microcontroller, enabling a localised warning system comprising of P5 SMD LED display panels, warning flashes and an automated alarm system.

Experiments and evaluation demonstrate that the enhanced vision system is capable of maintaining a consistent real time inference rate of 30 Frames Per Second (FPS). The object detection model obtained a mAP@0.5 of 90.9 percent with a class-wise precision of 91.1 percent for people and 90.8 percent for vehicles. The highest F1-score of 0.87 confirms the accuracy of the system in predicting the system's performance. The results show that this system is an intelligent safety system that is fast, offline and economical which can be incorporated into smart city designs.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background of the Project

The urbanization and establishment of smart cities have significantly contributed to the heavy traffic. With the rise in the number of vehicles on the roads, the problem of pedestrian safety has now become a burning issue. Research on transportation safety across the world has revealed that the majority of road accidents particularly in the developing world is due to uncontrolled pedestrian crossing since there are few advanced road traffic control systems.

The traditional zebra crossings are founded on road signs and road markings. These passive systems do not respond dynamically to people that pass by or to traffic.

The use of Artificial Intelligence (AI) in the transportation infrastructure has opened up new possibilities of intelligent safety systems. Embedded object detection is able to run directly on embedded devices without the use of the cloud and Jetson Nano is an edge AI platform [1]. A combination of computer vision, embedded systems, and smart warning systems can be used to create pedestrian crossings that are intelligent and responsive safety systems.

The project will introduce a Smart Crosswalk Integration System that will be based on real-time object detection to monitor pedestrian and vehicle traffic. It operates on live video feeds and has a lightweight deep learning model to process the video and trigger visual and audio notifications in case of dangerous scenarios.

1.2 Problem Description

While conventional zebra crossings and fixed roadside signs are widely deployed, the rate of pedestrian accidents in cities is still of concern. The main shortcomings of current infrastructure are:

The inability of passive road markings to dynamically alert distracted drivers, particularly in low-visibility or high-traffic conditions.

Fixed-timing traffic light systems that are not responsive to instantaneous changes in traffic and people.

The high cost of deployment of cutting-edge sensor networks (such as LiDAR or radar) for use across a city.

Therefore, the main engineering challenge of this study is the design of a low-cost, highly sensitive smart crosswalk system. This system must be able to detect and classify pedestrians and vehicles in real-time, operating entirely locally (without the use of cloud computing) to automatically activate localized visual and auditory alerts.

1.3 Objectives of the Study

The main goal of this work is to develop an edge-AI-based crosswalk system. The specific objectives are:

- To develop an embedded-optimized lightweight YOLOv5n object detection model.
- To obtain real-time inference (target upto 25 FPS) on the NVIDIA Jetson Nano 4GB, with TensorRT optimization.
- To develop a unifying hardware design for camera, AI inference and local physical interaction modules.

- To implement low-latency UART connection between the central processing unit and an ESP32 microcontroller for hardware triggering.
- To implement and trigger dynamic warning functions, such as a P5 LED light display and DFPlayer Mini sound module, for specific conditions.
- To assess system effectiveness through the application of conventional machine learning metrics (Precision, Recall, mAP and F1-score).

1.4 Scope of the Project

The following aspects will be included in this project:
The system consists of a single strategically located camera that can cover the crosswalk area and part of the approaching road. By carefully defining the Region of Interest (ROI), the system identifies pedestrians within the crosswalk area and vehicles approaching in a pre-defined detection area [2]. The AI model is trained on annotated datasets and optimized using TensorRT to execute effectively on the Jetson Nano 4GB platform. Output mechanisms are:

- P5 SMD LED Display (32×16 resolution) for visual warning messages.
- Flashing LED strips were installed near the crosswalk.
- DFPlayer Mini for audio announcements.
- ESP32 for handling output control logic.

The system is completely offline, privacy guaranteed and low latency.

Chapter 2

Literature Review

2.1 Introduction

Intelligent transportation systems (ITS) have evolved and transformed mobility in cities and smart city infrastructure. One of the major problems of the modern cities is the problem of the pedestrian safety in the crosswalks. Pedestrian accidents are caused by the overcrowding of roads, distracted driving and ineffective signalling systems. To address these issues, deep learning-based object detection, embedded systems, wireless communication, and smart signaling systems have been investigated by researchers [3].

The chapter is a review of six major contributions of research in the area of smart crosswalk systems. The focus of the discussion is on the methodologies, system architecture, detection, communication and implementation constraints. The purpose is to identify the gaps in the research and establish the foundation of the proposed Smart Crosswalk System.

2.2 Deep Learning-Driven Vision-Based Crosswalk Navigation with History-Based and Cascaded Control

The refrenced article proposes a vision-based crosswalk navigation system that integrates deep-learning with cascaded techniques [4]. Convolutional neural networks (CNNs) are applied by the authors to recognize the crosswalk patterns and guide pedestrian traffic. It possesses a control mechanism that is history based, thereby contributing to the stability of decision making based on the past frame information.
Cascaded control structure is good for detection in dynamic environmental scenarios like variation of illumination and occlusion. The experimental results show an improvement in localization accuracy and navigation con-

sistency. The system is not meant to regulate or enforce traffic or safety, but to assist pedestrians in navigation. It does not have vehicle detection and LED based warning systems and roadside signaling infrastructure. The approach thus improves navigation intelligence but not a complete crosswalk safety management solution.

2.3 Smart System of a Real-Time Pedestrian Detection for Smart City

To address the risks associated with city intersections, this study introduces an integrated visual monitoring system designed to detect pedestrians and prevent accidents [2]. The system architecture comprises of a camera unit, image processing unit, microcontroller, LED warning system and LCD display. Deep learning networks such as YOLO or SSD are used to recognize pedestrians on-the-fly.

When a pedestrian is detected, the LED warning lights are activated to alert the drivers, and the number of pedestrians is indicated on an LCD screen. The embedded control and computer vision combination is feasible in regard to implementation in the smart city environment.

Despite its merits, the system only deals with the detection of pedestrians. It lacks a vehicle detecting system and does not use a dual-decision system, which considers the presence of pedestrians and vehicles simultaneously. The signaling system is also not well developed, and there are simple LED warnings, and no multi-level warning systems.

2.4 Design and Implementation of Portable Smart Wireless Pedestrian Crossing Control System

The study conducted in [5] presents a mobile wireless system for pedestrian crossing control. Due to its portability and ease of deployment, the design is well-suited for temporary applications such as construction sites. It is founded on wireless communication modules to coordinate pedestrian demands with traffic signal controllers. The system can regulate the traffic by sending a message wirelessly to the traffic lights when a pedestrian presses the crossing request. The authors mention the low cost of implementation and flexibility of deployment as the key advantages. The system does not, however, have automatic pedestrian detection which is AI-based. Instead, it is manually fed through push-button mechanisms. Inability to detect objects in real-time and lack of computer vision limits the capabilities of automation in dynamic urban traffic environment.

2.5 Embedded Real-Time Vehicle and Pedestrian Detection Using a Compressed Tiny YOLOv3 Architecture

The study presented in [6] investigates the optimization of object detection for embedded environments. By implementing a compressed version of Tiny YOLOv3, the researchers successfully achieved real-time detection of vehicles and pedestrians on devices with limited computational overhead. The computational complexity is reduced by the pruning and quantization techniques of model compression. Experimental tests demonstrate that processing speed and detection accuracy are balanced in a trade-off. The article shows that small deep learning models can be successfully implemented to embedded systems. But the research is concerned with the detection performance and to a large extent. is not a physical signaling infrastructure e.g. LED displays, warning. Light control systems, or crosswalks. It provides a good foundation on which edge AI can be deployed but does not provide a complete smart crosswalk system.

2.6 New Approach to Intelligent Pedestrian Detection and Signaling on Crosswalks

To improve intersection safety, research in [7] introduces a dual-function system that integrates visual pedestrian tracking with signaling controls. This hybrid approach allows the crossing signals to be activated automatically based on real-time vision data, streamlining the interaction between foot traffic and vehicle signals. In case of pedestrians, the traffic lights are changed to enable safe crossings. This will increase awareness of the drivers and will reduce the reliance on manual traffic control. Results of the experiment indicate that pedestrian safety is improved in various environmental conditions. However, the sophisticated vehicle detection integration and the in-built processing platforms that are founded on GPUs are not discussed in the methodology. The signaling mechanism refers primarily to the changes in traffic lights and does not include the enhanced roadside visual alert systems such as flashing LEDs or dynamic displays.

2.7 Critical Analysis of Existing Literature

Reviewing recent advancements in Intelligent Transportation Systems (ITS), a recurring conflict between computational complexity and practical hardware implementation becomes evident. Research in [1] highlights the potential of using standard CNNs for precise localization. Nevertheless, the substantial processing overhead required—combined with a focus on navigation rather than real-time threat detection—makes such systems less viable for urgent crosswalk safety applications. On the other hand, lightweight embedded algorithms like the Tiny YOLOv3 implementation in [4] offer efficient real-time pedestrian detection. However, because these systems do not incorporate a dedicated hardware component, they remain dependent on existing connected infrastructure to function.. In addition, V2P technologies and wireless push-button networks [8] require the presence of either additional connected infrastructure or manual human operation, preventing autonomous operation. The key aspect missing from existing literature is an all-encompassing system capable of executing edge-computing AI while providing a hardware-based warning signal without cloud support.

2.8 Comparative System Analysis

For a comparative perspective on the proposed smart crosswalk design, Table 2.1 is presented below with consideration to detection methods, accuracy, hardware configuration, and key drawbacks.

Table 2.1: Comparison of Recent Smart Crosswalk and Pedestrian Detection Systems

Study / Year	**Core Methodology**	**Accuracy / Target**	**Hardware Platform**	**Primary Limitation**
Lee et al. (2025) [1]	CNN with Cascaded Control	Not Quantified	Cloud/Server Infrastructure	Navigation focus only; lacks roadside physical warning integration.
Park et al. (2023) [2]	V2P Wireless Communication	High (Proximity)	V2X Modules	Requires pedestrians and vehicles to possess connected smart devices.
Zhang et al. (2023) [4]	RF Wireless Synchronization	N/A (Manual Input)	Push-button RF Transceivers	Completely manual activation; no AI vision or autonomous detection.
Falaschetti et al. (2024) [9]	Compressed Tiny YOLOv3	Trade-off dependent	General Embedded Devices	Purely software-based detection; no output warning system designed.
Proposed System	YOLOv5n + TensorRT (FP16)	90.9% (mAP@0.5)	jetson Nano 4GB + ESP32	Performance degradation under extreme meteorological conditions (heavy fog).

2.9 Novelty and Contribution of the Proposed System

The proposed system design explicitly remedies the drawbacks enumerated in Table 2.1 by creating an entirely self-contained, offline safety mechanism. The engineering novelty of this work is rooted in its hybrid design approach, which leverages edge-computing AI for high-speed, localized processing compared to the delay associated with cloud computing in ITS through TensorRT FP16 optimization for direct execution of the YOLOv5n algorithm on the 128-core Maxwell GPU of the NVIDIA Jetson Nano. Dual-Class Logic Triggering: Deploying a logic matrix that demands multiple class detections within designated spatial boundaries (Pedestrian and Vehicle) simultaneously before entering emergency conditions, significantly reducing false positives.The system manages dedicated hardware by utilizing an additional ESP32 microcontroller to handle GPIO pin operations and external hardware control. The Jetson Nano is spared from I/O interruptions in its uninterrupted 30 FPS image processing task while simultaneously activating P5 SMD LEDs, intense flashes, and DFPlayer Mini audio alarms.

Chapter 3

Requirement Specifications

3.1 Review of Current Smart Crosswalk Practices

The current smart crosswalk systems that are being used in developed urban environments are largely founded on sensor-based activation systems, which comprise infrared sensors, pressure pads, microwave radar sensors and push-button activation systems. These systems are meant to enhance safety of pedestrians by flashing lights or warning signs when pedestrians are identified. However, the existing implementations are not also sophisticated in decision-making and lack real-time visual intelligence.

Most commercial smart crosswalks are working on either by fixed-timing or simple motion sensors. These systems are not able to differentiate between the pedestrians, vehicles and environmental noise [10]. In addition, these conventional systems do not dynamically examine the location of pedestrians within a given Region of Interest (ROI). As a result, they may give false alarms or fail to operate in complex traffic scenarios.
The most recent research trends are associated with the AI-based deep learning-based pedestrian detection. The majority of these systems are however cloud-based processing which introduces latency and dependency on connectivity. Furthermore, many existing solutions are built on expensive hardware architectures, making them financially impractical for city-wide deployment. This lack of cost-efficient embedded platforms creates a significant barrier to scaling these systems across broader urban infrastructures.
The proposed Smart Crosswalk system overcomes all these limitations by integrating edge-based AI processing using NVIDIA Jetson Nano 4GB. The system is a real-time pedestrian detector which is not cloud based

and has low latency and high reliability.

3.2 Overview of the Proposed Crosswalk System

The proposed Smart Crosswalk is an artificial intelligence-based pedestrian detection and warning system which can be implemented in real-time in smart cities. The system incorporates a vision-based surveillance system and embedded edge computing to react in real-time to the presence of pedestrians and vehicles.

The crosswalk is under constant observation by a high resolution camera. The recorded video stream is then fed to a lightweight deep learning model, which is based on the YOLO model and deployed on the Jetson Nano 4GB platform. The system produces a Region of Interest (ROI) within the crosswalk space to identify the presence of pedestrians.

When a pedestrian is detected within the ROI, and a vehicle is approaching it, the system activates a number of warning systems including:

- LED Warning Screen
- Flashing LED indicators
- Backlit pedestrian signage
- Audio alert module

The system also gives real-time warning of the hazards to the driver and is also energy efficient and responsive.

3.3 System Architecture

3.3.1 High-Level Architecture

The system architecture has three layers:

1. Input Layer – Camera module to receive visual information.
2. Processing Layer - NVIDIA Jetson Nano 4GB to process AI.
3. Output Layer - LED display, flashing lights, backlit signage and audio module.

It has an architecture that is founded on an edge-computing paradigm where all inference functionality is executed locally on the embedded system and is not cloud-dependent. This reduces latency and ensures reliability in low-connectivity conditions.

3.3.2 Module Interaction

The modules are logically and sequentially interacting:

- The camera streams the video.
- Frames are sent to the AI inference engine that is operating on Jetson Nano.
- Pedestrians and vehicles are detected with the help of YOLO model.
- The decision module will examine the availability of individuals on the pedestrian crossing the ROI.
- Output modules are turned on when there are risk conditions.
- ESP32 microcontroller: The display of LEDs and the switching of warning signals is controlled by the ESP32 microcontroller.

- This scalable architecture is maintainable and scalable.

3.4 Requirement Specification

3.4.1 Functional Requirements

The main function of the Smart Crosswalk System is the continuous, real-time monitoring of the pedestrian crossing area. The functional requirements are defined as follows:

- **Real-Time Processing**: The object detection algorithm will be implemented in the vision processing unit with a minimum processing speed of 15 Frames Per Second (FPS) with a target of 30 FPS so as to ensure smooth tracking of moving targets.

- **Detection Accuracy**: The deep learning model must categorize "Pedestrians" and "Vehicles" within the specified Region of Interest (ROI) with a minimum Mean Average Precision (mAP@0.5) of 85 percent.

- **Edge Autonomy**: The system must run 100 percent offline. All AI inference and hardware triggering must be executed locally on the embedded platform without any dependence on cloud computing infrastructure.

- **Hardware Actuation**: Upon positive risk detection, the system must independently transmit serial commands to the secondary microcontroller to trigger visual (LED panel) and auditory (DFPlayer) outputs within an integrated operational loop [11].

3.4.2 Non-Functional Requirements

To ensure the system is viable for public safety infrastructure, it must meet rigid performance and environmental requirements:

- **System Latency**: The end-to-end response time—measured from the moment a camera frame is captured to the physical activation of the warning modules by the ESP32—must remain under 100 milliseconds (ms).

- **Power Consumption**: The system delivers superior energy efficiency to ensure compatibility with smart city power infrastructures. The central processing unit (Jetson Nano) must operate within a strict 5W to 10W power, and the complete integrated circuit must function reliably on a standard 5V, 10A DC switch-mode power supply.

- **Environmental Robustness**: The physical compound and camera module must maintain operational integrity across moderate environmental variations, reducing false positive detections caused by static infrastructure.

3.5 Visual and Structural Components

3.5.1 Block Diagram

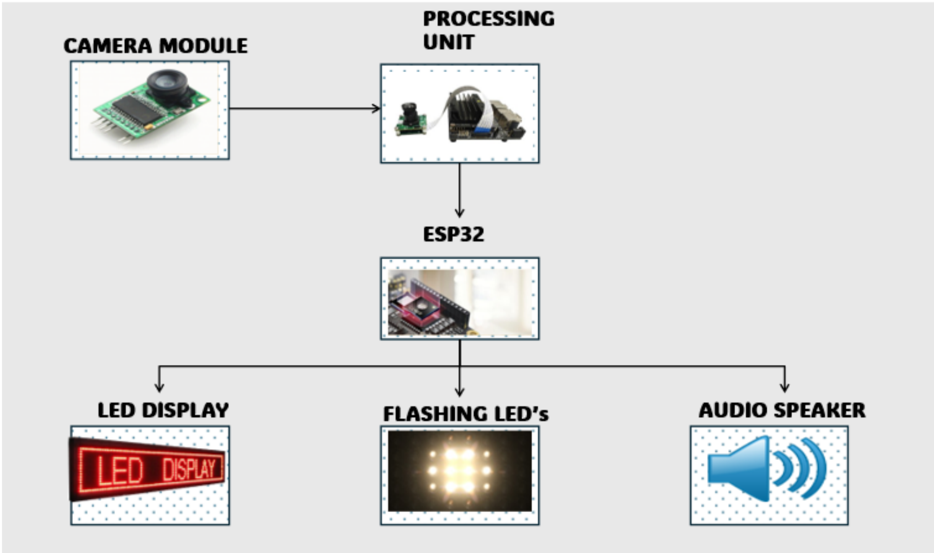

Figure 3.1: Block Diagram Of Smart Crosswalk Integration For Smart City.

This block diagram depicts a pedestrian warning and alert system which employs vision. A camera module records a video feed that is transmitted to a Jetson Nano 4GB processing unit. In Jetson Nano, an object detection algorithm (YOLOv5) is implemented to process each frame and detect pedestrians and set the appropriate system state. The system detects three states of the situation, depending on the logic of detection: Idle (safe to drive), Warn (slow down), or Stop (pedestrian ahead). The Jetson Nano then sends state commands via UART communication to an ESP32 microcontroller which is the controller. The ESP32 translates these commands into physical outputs: it turns on flashing LEDs to show warning/stop, turns on an audio module to generate alerts (typically when in the stop state), and updates a P5 SMD LED display to display the right messages [12].

The camera and Jetson Nano perform intelligent vision processing and the ESP32 performs real-time control of visual and audio output devices to inform the driver.

3.5.2 3D Layout

Below is the computer-Aided Design (CAD) images of the proposed Smart Crosswalk system. The CAD model shows the physical positioning of the

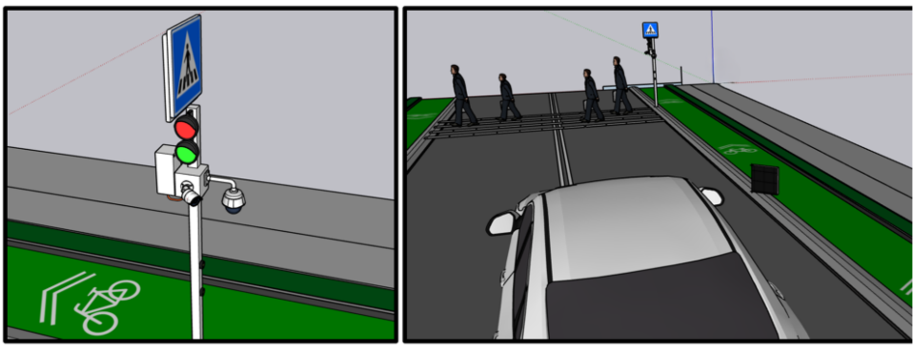

Figure 3.2: 3D layout of smart crosswalk made on sketchup.

components like camera module on a pole on the traffic signal, pedestrian crossing signs, LED warning display panel and roadside installation of warning modules:

- The angle of the camera position to achieve the maximum ROI.
- LED positioning screen to view the car.
- Backlit signs at the driver's level.
- Crosswalk marking alignment
- Stepwise Operational Visualization

The CAD simulation illustrates the work of the system step-by-step:

1. Step 01 - Pedestrian approaches crosswalk.
2. Step 02 - The camera takes the shot of the pedestrian.
3. Step 03 – Detection confirmation
4. Step 04 - Warning light on.
5. Step 05 - LED panel alerts of approaching car.
6. This 3D model affirms the feasibility of the real world and installation.

3.5.3 System Process Flow

Below is the system's process flow.

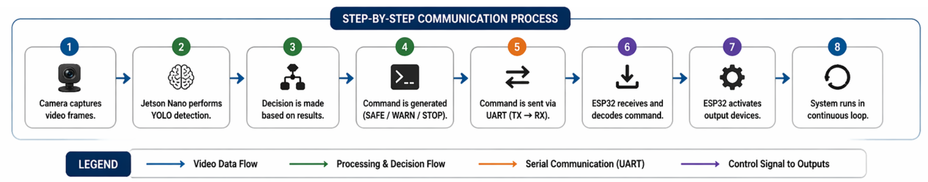

Figure 3.3: Flow chart of the system logic.

This process flow describes the logic of the Smart Crosswalk system in a logical, systematic flow of work. This begins with booting the system and the Jetson Nano installing all the required software, including the object detecting model and communication interfaces. Once the Jetson Nano has been started, this device links to the pedestrian and vehicle camera modules to continue the video streaming. The two cameras then begin real time surveillance of their respective field of view with predetermined Regions of Interest (ROI) that are focused on the crosswalk and the lanes coming towards it.

The video frames are analyzed during processing by the detection algorithm. When a pedestrian is detected within the designated ROI of the crosswalk area, the system will automatically change to the warning mode and abandon the monitoring mode. In this state, several safety outputs are activated simultaneously: the LED warning screen is activated, and an alert message is displayed on the screen, the flashing LED system is activated, and high-visibility visual warning signals are given, the backlit crosswalk sign is activated, and more attention is given to the approaching drivers. When a car is also seen coming to the crosswalk at the same time, the system maintains the backlit sign and warning signs to implement the alert condition and ensure that the driver attention is not lost in the period when the pedestrian is on the crosswalk.

The system revises the scene on real time basis. Once no pedestrian is detected in the ROI, all warning outputs are switched off, and the system automatically returns to default monitoring mode. This endless loop of detection, decision-making and output control will ensure responsive, real-time functionality and minimize unwarranted alerts. Overall, the flowchart is an excellent illustration of the conditional branching logic model, whereby the system outputs are triggered based on the real-time detection findings to ensure that the pedestrians are safer at the crosswalk [13].

Chapter 4

Hardware Design and Specifications

4.1 Introduction

This chapter describes the hardware design and specification of the Smart Crosswalk System. The performance of the proposed system depends on the selection of the suitable hardware components. Thus all the components are selected carefully based on its processing capacity, compatability, cost effectiveness and its capability to work in real time. The hardware modules include processing unit, camera module, microcontroller, display unit, power supply and audio system. All of these components are part of one system that is capable of detecting pedestrians and vehicles and issuing appropriate warnings. This chapter also contains specification sheets of all the components and why they have been selected.

4.2 Jetson Nano 4GB

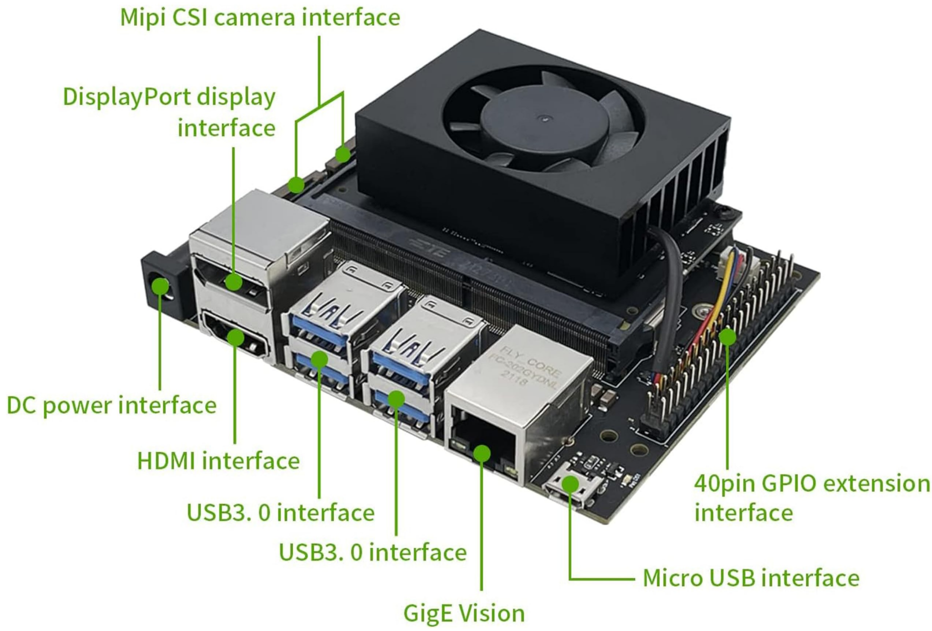

Figure 4.1: Jetson Nano.

4.2.1 Description

The Jetson Nano 4GB is the central processing unit of the system. It is particularly targeted at AI-based applications and provides the acceleration of deep learning models by using the GPU. It will be responsible in this project to implement the YOLO model and real-time object detection [14].

4.2.2 Specification Sheet

Table 4.1: Jetson Nano Specification Sheet

Parameter	**Specification**
Processor	Quad-core ARM Cortex-A57
GPU	128-core Maxwell GPU
RAM	4GB LPDDR4
Storage	microSD card
Operating System	Linux (JetPack SDK)
Power Consumption	5W – 10W
Connectivity	USB, HDMI, Ethernet
AI Framework Support	TensorFlow, PyTorch

4.2.3 Justification for Selection

Jetson Nano is selected due to its high-performance computer and ability to run deep learning models. It can also accelerate the graphics processor (GPU) as compared to the conventional microcontrollers, which is crucial in the real-time recognition of objects. Moreover, it is not expensive like other AI platforms and can support different AI frameworks, which is why it is suitable in this project.

4.3 Camera Module

Figure 4.2: Camera module.

4.3.1 Description

The camera module is used for capturing the video of crosswalk in real-time. It provides the necessary visual data to identify objects. The position of the camera is important so that the road and the pedestrian area can be covered as much as possible [15].

4.3.2 Specification Sheet

Table 4.2: Camera Module Specification Sheet

Parameter	**Specification**
Resolution	720p / 1080p
Frame Rate	30 FPS
Interface	USB / CSI
Lens Type	Wide-angle
Focus Type	Fixed
Power Supply	5V

4.3.3 Justification for Selection

The use of the camera module was due to its ability to provide high-quality video input. A clear video feed is required to make the correct detection. The selected camera is cheap, compatible with Jetson Nano, and capable of capturing detailed images that are required to carry out the detection process [14].

4.4 ESP32 Microcontroller

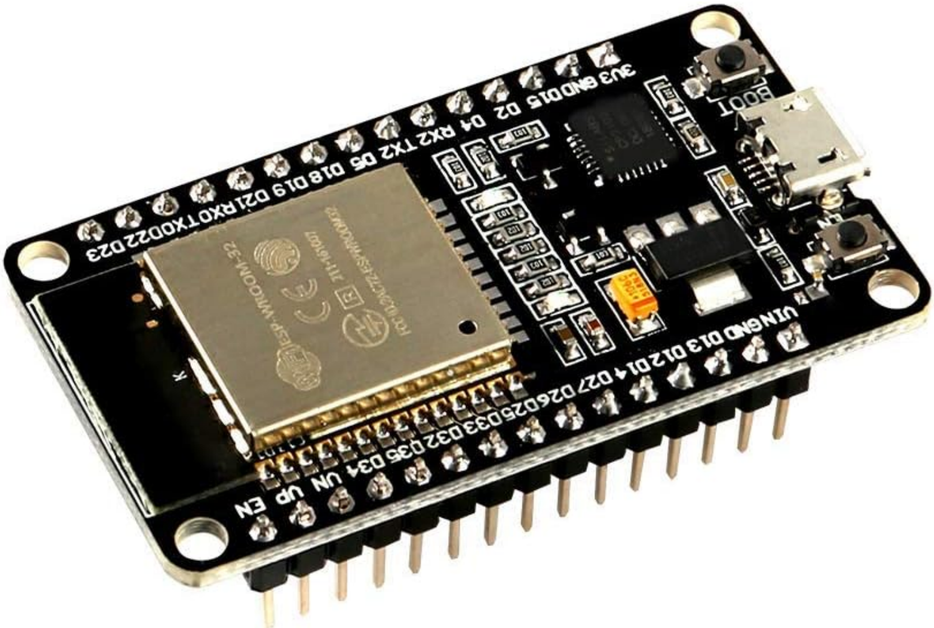

Figure 4.3: ESP32.

4.4.1 Description

ESP32 microcontroller is used as a control unit to the output devices. It receives the signals of the Jetson Nano and turns on the warning systems respectively.

4.4.2 Specification Sheet

Table 4.3: ESP32 Specification Sheet

Parameter	**Specification**
CPU	Dual-core Xtensa
Clock Speed	Up to 240 MHz
RAM	520 KB SRAM
Wi-Fi	Yes
Bluetooth	Yes
GPIO Pins	34 pins
Operating Voltage	3.3V

4.4.3 Justification for Selection

ESP32 was selected as it is extremely rapid, inexpensive and simple to combine. It is wireless communication-enabled and has sufficient GPIO pins to drive different devices. It is low power consuming and hence can be utilized in continuous operation.

4.5 SMD LED Display

Figure 4.4: SMD Display.

4.5.1 Description

SMD LED display has been used to indicate warning messages to the drivers [16]. It provides visual indications such as stop or slow down.

4.5.2 Specification Sheet

Table 4.4: SMD Display Specification Sheet

Parameter	**Specification**
Type	P5 SMD LED Module
Resolution	32×16
Voltage	5V
Brightness	High
Scan Type	1/4 Scan

4.5.3 Justification for Selection

The LED display is selected because of its brightness and the possibility to be observed in the outdoor setting. It ensures that the warning messages are easily visible even in the daytime.

4.6 Audio Module and Speaker

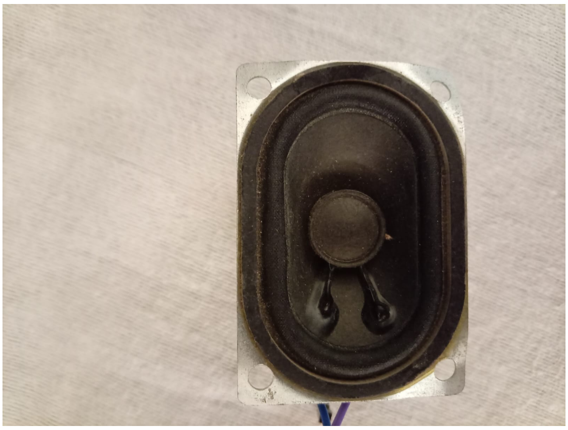

Figure 4.5: Speaker.

4.6.1 Description

The audio unit generates sound alerts to warn the drivers in case pedestrians are detected.

4.6.2 Specification Sheet

Table 4.5: Speaker Specification Sheet

Parameter	**Specification**
Model	DFPlayer Mini
Audio Format	MP3
Voltage	3.2V – 5V
Output	Speaker Interface

4.6.3 Justification for Selection

The choice of the audio module was due to its simplicity and lack of costliness. It provides a feasible solution to generating audible alerts without necessarily having complex hardware.

4.7 Power Supply

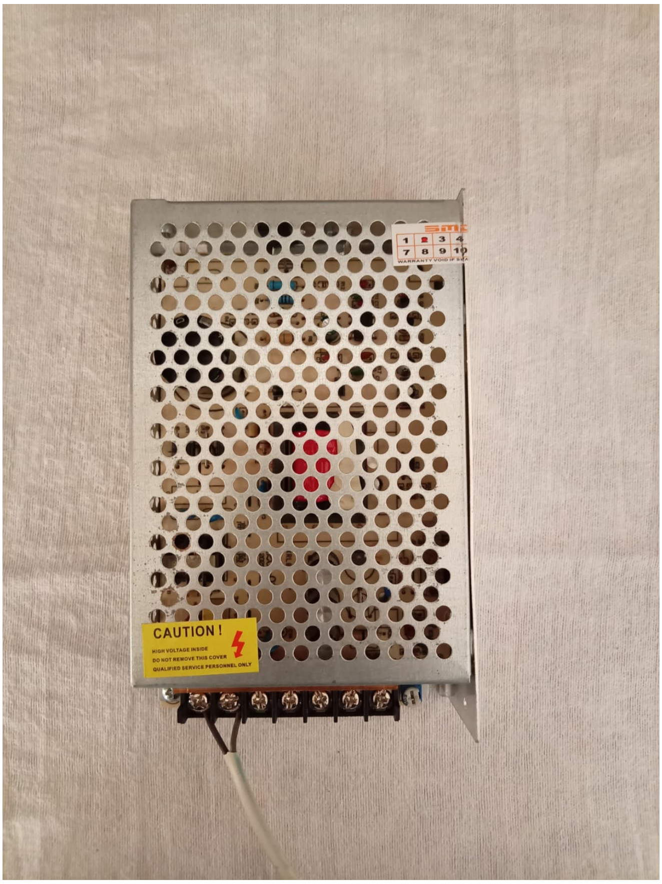

Figure 4.6: Power supply.

4.7.1 Description

Power supply ensures a constant voltage and current to the system components.

4.7.2 Specification Sheet

Table 4.6: Power Supply Specification Sheet

Parameter	**Specification**
Type	SMPS
Output Voltage	5V
Output Current	10A
Input Voltage	220V AC

4.7.3 Justification for Selection

The justification behind the SMPS power supply is that it can provide constant power to different components at a specified time. It ensures the stable operation of the system.

Chapter 5

Implementation of Smart Crosswalk System

5.1 Introduction

This chapter is about the full implementation of the proposed Smart Crosswalk System. It encompasses the decision and design of the hardware, preprocessing of the data, training of the deep learning model, and system logic design. The aim of this chapter is to demonstrate how conceptual design as described in the earlier chapters is put into a working system [17].

5.2 Hardware Implementation

The hardware selection was based on the performance, cost, and compatibility with AI-based applications. The system is modular in nature where each component has a specific task to undertake.

5.2.1 Jetson Nano 4GB

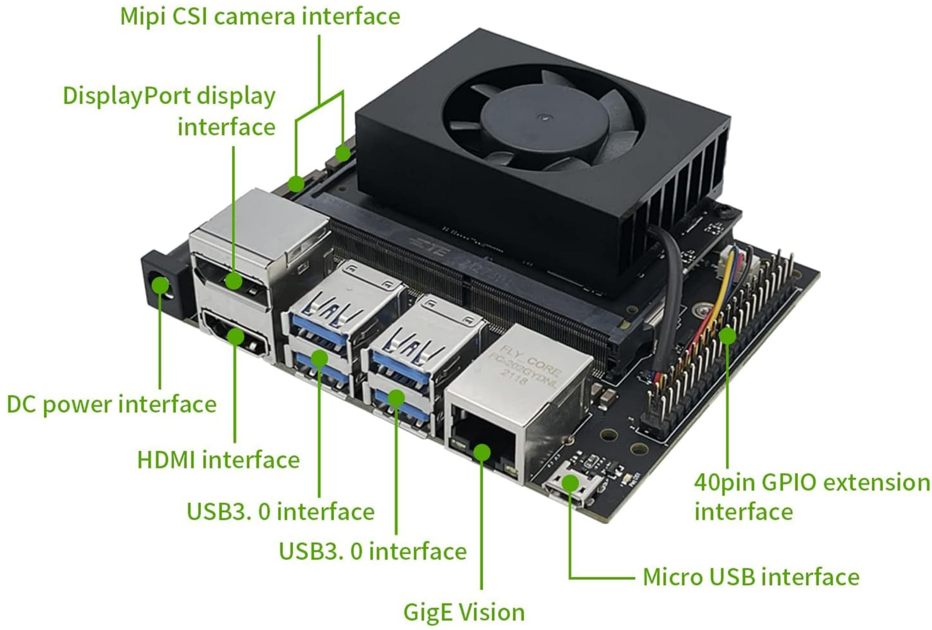

Figure 5.1: Jetson Nano.

The main processing unit is Jetson Nano 4GB. It is designed to be applied in edge AI applications and provides deep learning model acceleration through the utilization of a GPU. Although the model training is done on a computer-based system, the Jetson Nano is expected to apply the trained model to identify in real-time.

5.2.2 Camera Module

Figure 5.2: Camera module.

The camera module is a live video input of the crosswalk environment. The place where the system is placed is very important to its efficiency. The camera must be set at a height and angle which will allow them to see clearly the pedestrians and vehicles heading towards them.

5.2.3 ESP32 Controller

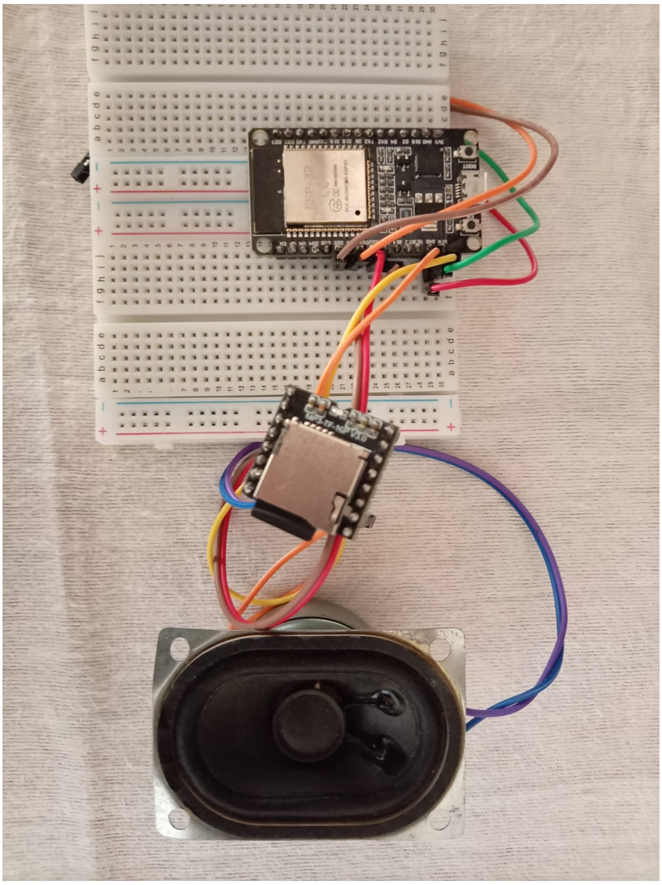

Figure 5.3: ESP32 controller.

The ESP32 microcontroller is used to control output devices such as audio alerts and LEDs. It accepts orders of the processing unit and executes them efficiently.

5.2.4 Output Devices

Figure 5.4: SMD LED display showing pedestrian warning message.

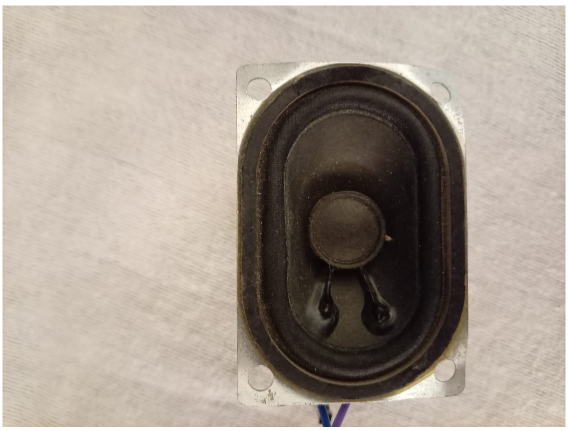

Figure 5.5: Audio speaker module for voice alerts.

The system has a number of output devices to provide warnings: SMD LED Display of visual messages. Flashing LEDs as attention alerts. Sound warning audio module and speaker. These devices are turned on depending on the system condition.

5.3 Software Implementation

This section defines how we used software to train our model.

5.3.1 Developing Environment

The system is developed using Python programming language. The processing of images is carried out with the assistance of such libraries as OpenCV, and the training of the YOLO model is carried out with the assistance of deep learning frameworks [18].The training is done on a computer system due to its high capability to compute.

5.4 Dataset Preparation

The preparation of the dataset is one of the most important steps of the system development. Several datasets which are publicly available on Kaggle [19] were used, rather than one dataset. According to these datasets, relevant images containing pedestrians and vehicles were selected.
It was carried out in the following steps:
A collection of data sets of different sources.

- Relevancy image filtering to crosswalk situations.
- Removal of undesired or bad quality images.
- Choosing images to be manually labeled.
- This was done to ensure that the data set was tailored to the Smart Crosswalk System.

5.4.1 Data Filtering and Selection

The datasets received were very numerous and many irrelevant images. In this way, the pictures that did not contain any pedestrians or cars were removed. Only images of real road situations were selected.

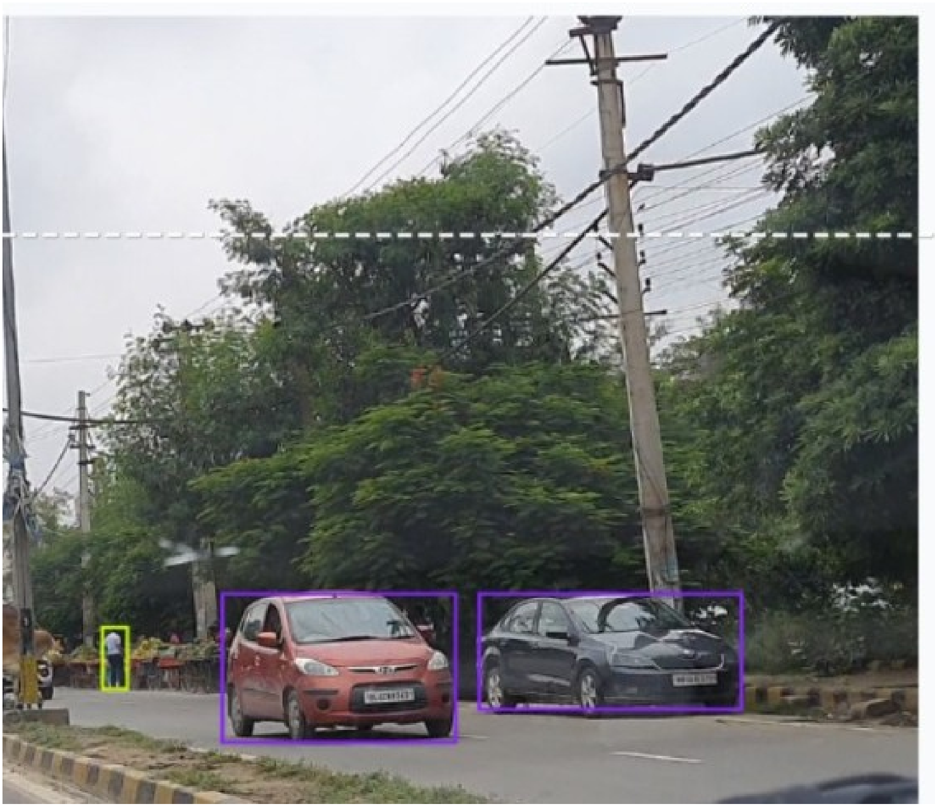

Figure 5.6: Selected data.

Figure 5.7: Raw data.

5.4.2 Data Annotation (Labeling)

Figure 5.8: Annotated dataset samples showing bounding boxes for vehicle and pedestrian detection.

The images were filtered and annotated manually post-filtering. The pedestrians and vehicles were defined by drawing a bounding box and each object was assigned a class label using Roboflow [20]. Labeling is required because the model is trained to identify objects based on these labels.

5.5 Model Training

The trained model based on YOLO was trained on the prepared dataset. The training process involves feeding the model with labeled images and optimizing the parameters to minimize errors.

5.5.1 Training Performance

Figure 5.9: Model inference performance illustrating real-time detection of pedestrians and vehicles with associated confidence scores.

This is because the model performance improves with time in the training process, as it is capable of recognizing objects more accurately.

5.6 Detection Logic

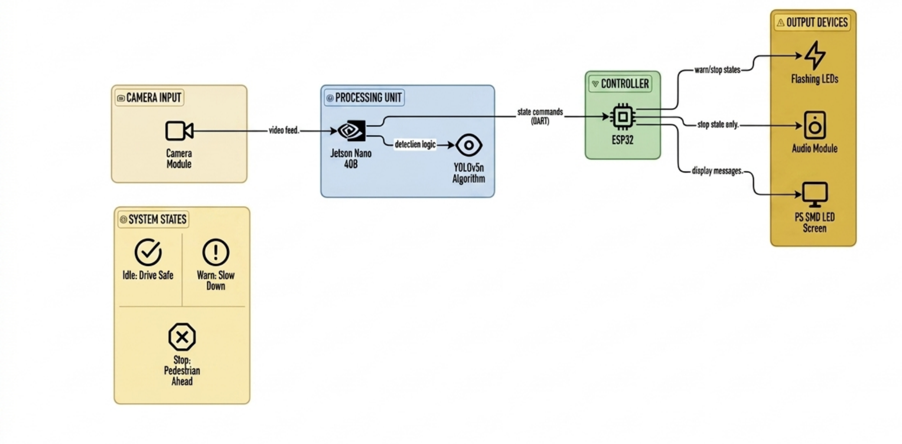

Figure 5.10: System architecture and detection logic flow including processing unit and output device integration.

The detection system works based on processing each frame and object detection. The system decides what to do depending on the outcomes of the detection. The logic includes: The system fails in the event that there is no pedestrian detected. There is a warning in case a pedestrian is identified. A stop signal is activated when both the pedestrian and vehicle are detected.

5.7 Communication Design

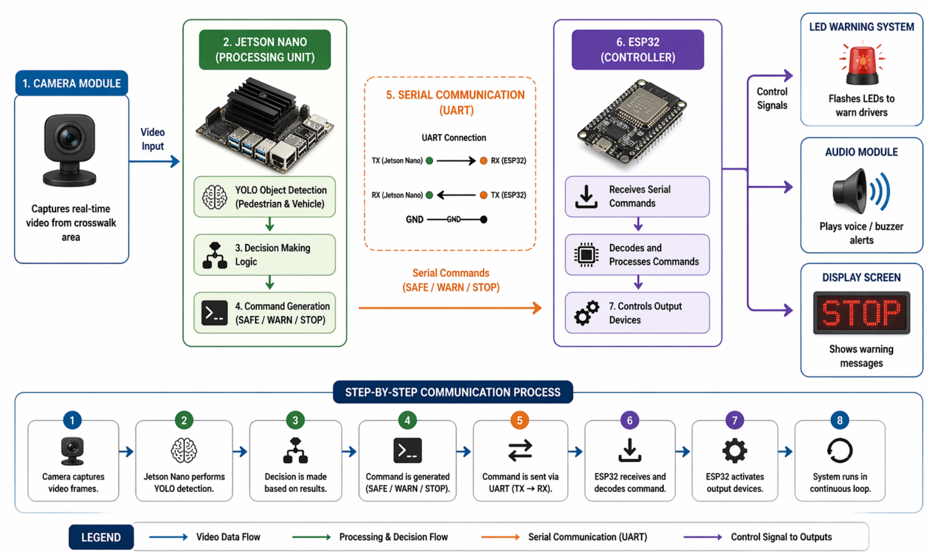

Figure 5.11: Communication flow and step-by-step process between Jetson Nano and ESP32.

Serial communication aids in the development of processing unit and controller communication. Commands are sent depending on the outcome of detection.

Chapter 6

System Testing, Results and Analysis

6.1 Introduction

The chapter describes the testing procedure and performance evaluation of the proposed Smart Crosswalk System. The system was tested with the trained YOLO based object detection model to detect the pedestrians and vehicles in real-time scenarios. The system was tested for accuracy, reliability and efficiency under different environmental conditions. The model performance was evaluated by different evaluation measures like precision, recall, F1-score, and confusion matrix [5]. The results show the efficiency of the system in object detection and response to the objects. The analysis also discusses strengths and weaknesses of the proposed system. Testing Results The results are in graphical form for clear view of system performance. The results confirm the practical feasibility of the developed system.

6.2 Testing Methodology

The system was evaluated on a database of images with pedestrians and vehicles under various conditions such as day time, night and traffic congestion. The dataset was divided into training and validation set to evaluate the model performance. The model was trained and tested. To find things in pictures that we can't see so that we can generalise. The system performance was assessed by means of standard measures such as Precision, Recall and F1 Score. These tests are useful for determining how accurate the model is at identifying objects and for avoiding false identifications. We also calculated the confidence of the predictions to determine the optimal level at which we can determine the presence of a certain object. The graphs used to visualize the model behavior were precision-recall curves and F1-confidence curves. This will give full test of the ability of the system to detect. For research based projects this chapter should include full description of evaluation metrics, analysis/discussion of evaluation results.

6.3 Model Inference and Performance Metrics

For determining the performance of the deployed edge computing YOLOv5n algorithm for real-time detection, various machine learning metrics and training hyperparameters were collected. The metrics that have been used in the project to evaluate the performance of the model have been described in Table 6.1 below.

Table 6.1: Summary of System Evaluation Metrics and YOLOv5n Training Parameters.

Metric	Project Value
Precision	91.1% (Pedestrian) 90.8% (Vehicle)
Recall	91.0% (Pedestrian) 89.0% (Vehicle)
F1 score	0.87
mAP@50	90.9%
mAP@50-95	~58% - 60%
Epochs	100+
Learning Rate	0.01 (Standard YOLOv5)

6.3.1 Analysis of Inference Speed and Accuracy

As shown in Table 6.1, the performance of the YOLOv5n model, fine-tuned using TensorRT for edge computing on the NVIDIA Jetson Nano platform, was highly effective. After compiling the .engine file, the device achieved a consistent inference rate of 30 frames per second (FPS). This performance is ideal for practical smart city applications, as it ensures minimal latency between the object detection process and the response of the hardware warning system.

The overall detection accuracy was reported at 90.9 percent mAP@0.5. A class-wise analysis revealed that pedestrian detection accuracy was 91.1 percent, while vehicle detection accuracy stood at 90.8 percent. Moreover, the F1 score of 0.87 indicates that the model maintains a good balance between precision and recall, which helps minimize both false positives and false negatives in a real-time crosswalk environment.

6.4 Training Performance Analysis

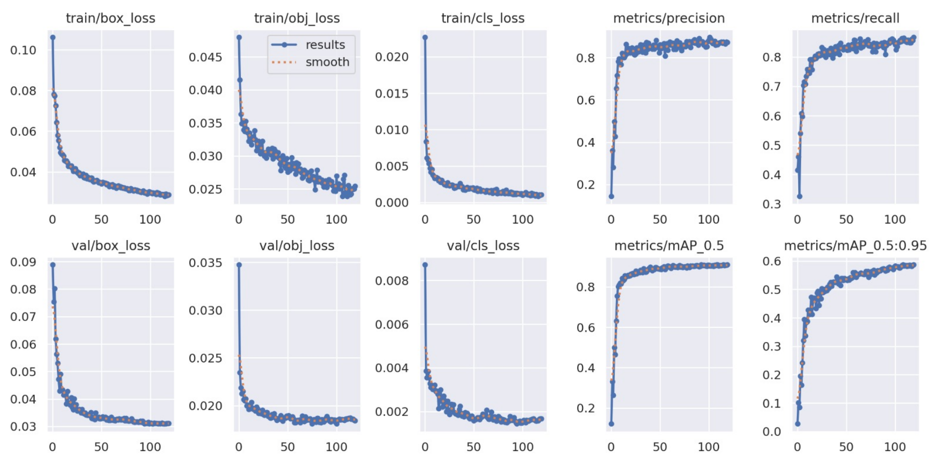

Figure 6.1: Training and validation loss curves alongside precision, recall, and mAP metrics over 100+ epochs.

6.4.1 Training Loss Analysis

The training graphs show that the model is effective in time-based learning. The training box loss starts with a high value of approximately 0.105 and decreases gradually to approximately 0.028, which implies that the bounding box accuracy is enhanced. Similarly, object loss is reduced to about 0.025 compared to 0.047, that shows that the model is more effective in establishing the presence of an object in an image. The classification loss also decreases significantly to approximately 0.022 to nearly 0.002, which means that the model is more efficient in the accurate classification of pedestrians and vehicles. The gradual reduction of all training losses indicates that there are no fluctuation and instability in learning.

6.4.2 Validation Loss Analysis

The validation losses are also on a downward trend, which is an affirmation of good generalization performance. The error of the validation box is minimized to about 0.031, which means that the model can estimate the position of objects on the hidden data. The loss of the validation object is reduced to approximately 0.018 compared to 0.035, indicating that the object detection is more confident. The validation classification loss is minimized to about 0.009 to 0.0015 which is a good classification ability [21]. The similarity of the training and validation curve shows that the model is not overfitted and can be applied to both training and validation data.

6.4.3 Precision and Recall Analysis

Good detection is shown by the accuracy and recall curves. The accuracy increases rapidly, beginning with approximately 0.15 and reaching approximately 0.88-0.90, implying that most of the objects identified are correct.

The recall rises to approximately 0.87-0.89, i.e. the model recognizes most of the actual objects in the images. The two curves stabilize after several epochs signifying that the model has achieved optimal performance. High precision reduces false detections, and high recall ensures that no pedestrians and vehicles are missing.

6.4.4 Mean Average Precision (mAP) Performance Analysis

The mAP @0.5 curve shows a steady increase and reaches the maximum at approximately 0.90, indicating that the detection accuracy is extremely high at the common IoU threshold. A stricter measure of evaluation, the mAP 0.5:0.95, is gradually increasing to about 0.58-0.60. This means that the model can be successfully applied even in strict evaluation conditions. The progressive rise and stabilization of these measures is the evidence that the model has learned the localization and classification successfully.

6.4.5 Model Convergence Discussion

The model convergence is suitable as shown by the general training behavior. All the values of the losses are decreasing smoothly and such performance measures as precision, recall and mAP are increasing. The model achieves a stable performance after approximately 80-100 epochs and no additional improvement is significant [21]. There are no sudden spikes and irregularities in the graphs, which is evidence of the stable and reliable training. This demonstrates that the model is well trained and can be applied in real-time pedestrian and vehicle detection.

6.5 Precision-Recall Evaluation

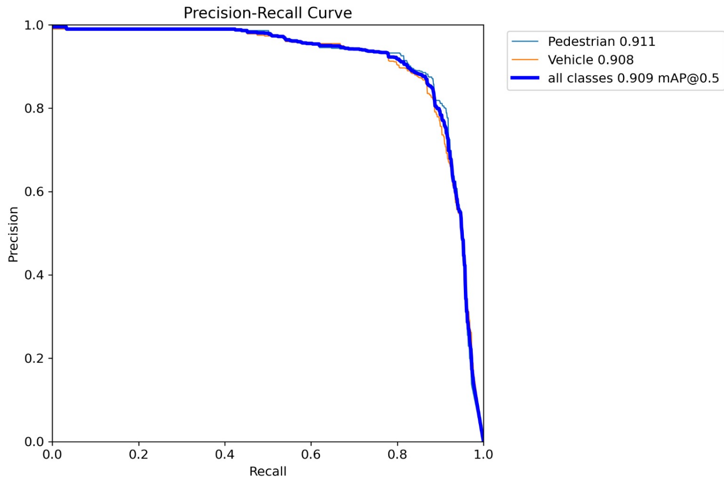

Figure 6.2: Precision-Recall curve for pedestrian and vehicle classes at mAP@0.5.

The Precision-Recall curve depicted in Figure 6.2 shows the performance of the suggested model in terms of pedestrian and vehicle detection. It is clear from the chart that there is a strong dependency between the two variables, implying that the detection is performed efficiently. Pedestrian class receives a precision of about 0.911; similarly, vehicle class receives approximately 0.908 of precision. The entire performance of the model results in the mAP@0.5 metric of 0.909. The graph implies that the precision is extremely high (almost equal to one) regardless of the level of recall from the interval of $[0.0 - 0.8]$ [22]. It is obvious that almost no false positives are detected by the suggested model, at least in most cases. At the same time, when the value of recall tends to become higher (approaching the value of 1.0), some decrease in precision occurs, which is usual for such systems. It is evident from the Precision-Recall chart that the curve is relatively smooth, and that pedestrian and vehicle performances are rather close, thus implying that the model detects objects from both classes equally accurately and reliably. This fact is crucial for real-world crosswalk safety system application.

6.6 F1 Confidence curve

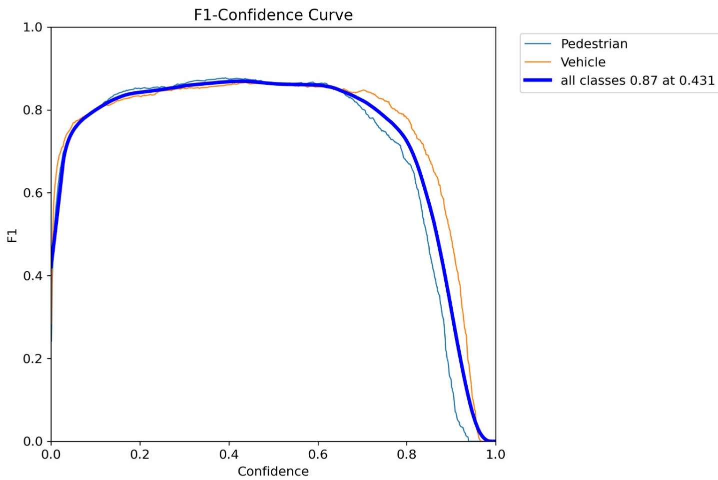

Figure 6.3: F1-Confidence curve showing the optimal threshold for pedestrian and vehicle detection.

As can be observed from Figure 6.3, the F1-Confidence curve demonstrates the dependency of the confidence level on the F1-score of the suggested object detection algorithm. The F1 score represents the harmonic mean of the precision and recall metrics. From the graph provided, one can conclude that the model reaches its peak at the F1-score of about 0.87, which corresponds to the confidence level of 0.431 and is its optimal operating point. Specifically, the F1-score experiences a sharp increase at the confidence level between 0.0 and 0.2 because the model detects all objects with high recall but low precision [12]. On the contrary, at the confidence level between 0.2 and 0.7, the F1-score remains stable within the range of 0.80 to 0.87 and does not experience significant fluctuations, which is evidence of good model performance. At the confidence levels exceeding 0.8, however, the F1-score drops sharply due to the increased stringency of the model in making predictions and, thus, rejects all true detections, which results in a lower recall rate. Interestingly, one can observe that the trends for pedestrians and vehicles are similar to each other. The graph also shows that both pedestrian and vehicle classes follow similar trends, indicating consistent performance across different object categories.

6.7 Confusion Matrix Analysis

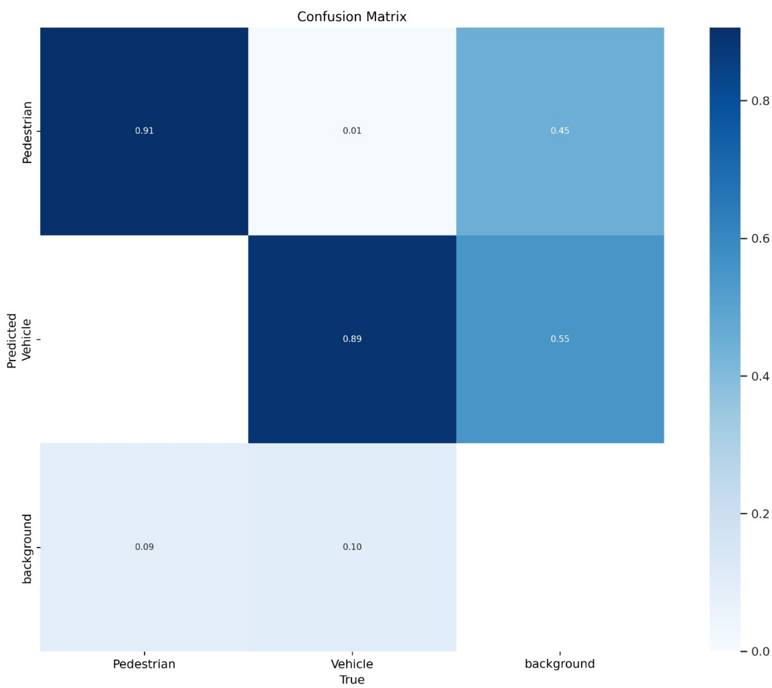

Figure 6.4: Confusion matrix illustrating the classification accuracy for pedestrian and vehicle detection.

The confusion matrix provides the breakdown of the correct and incorrect model predictions. It shows the results of the classification of pedestrian, vehicle, and background classes. The values on the diagonal are the correct predictions and the values on the off diagonal are the misclassifications. The matrix shows that the model can identify pedestrians and vehicles with an accuracy of approximately 0.91 and 0.89 respectively. The classification performance of these values is good. Objects and background are somewhat misclassified, as is common in complex scenes. The confusion of the background means that the model occasionally recognizes irrelevant areas as objects. However, the overall performance is good due to the good detection accuracy of the primary classes. This demonstrates that the system can distinguish between pedestrians and vehicles effectively.

6.8 Detailed Discussion of Results

Our developed intelligent system is working perfectly. It can detect pedestrians and vehicles properly. After examining our results, we have identified the capabilities and limitations of our system. The graphs shows that the system was trained very well. They show that the system learned well and made few mistakes. This means that the training was successful and the system is working properly. As the performance of the system did not drop suddenly. We can conclude that training was stable. If we look at the Precision-Recall chart we can see that the smart crosswalk system is very precise. It means that the system does not make false detections. This is very important because false detections can make the system less effective in real world. The F1-Confidence chart shows that the smart crosswalk system works best when it is 43 percent confident. The smart crosswalk system is very good at detecting pedestrians and vehicles. It has precision and recall which means it is very effective. Despite the remarkable performance of the system, there are certain limitations that may be identified based on the analysis. When the threshold of confidence grows, the recall starts decreasing significantly, which means under strict detection conditions, the algorithm would not be able to detect all the objects. Furthermore, testing accuracy may be compromised by environmental factors like fog or heavy traffic conditions. Based on the confusion matrix analysis result, it can be concluded that there are misclassified instances, their number is small and that the overall accuracy of the system is rather high. Nevertheless, since this phenomenon is normal for object recognition algorithms, we need to keep working on it to make it even better. Summing up the findings, we can say that the developed system proved to be highly accurate and stable under typical conditions; moreover, it demonstrated its real-time nature, which made it suitable for real-life usage in smart crosswalks [10].

6.9 Limitations of the System

Despite the good performance of the system, it has its limitations. The detection accuracy may be low in extreme weather conditions such as heavy rain or fog. Low light conditions can also affect the camera and detection model performance. The system may provide false detections when the objects are partially covered or overlapping. In addition, the quality and variety of training data also affect the model performance. The absence of variation in the data can reduce the generalization power. The processing speed can also be affected when processing a few objects simultaneously. These limitations define the aspects of future work improvement.

Chapter 7

Conclusion and Future Recommendations

7.1 Conclusion

The main purpose of this work was to design and build a pedestrian and vehicle detection system using a deep learning approach and a dedicated processor. This goal has been successfully achieved due to the use of a YOLO-based model in combination with the processing and control unit.
It has been proven that a pedestrian and vehicle detection system can detect objects in an image with a high level of reliability. The performance of the system is demonstrated by the high mAP value 0.5 equal to 0.90, as well as excellent values of precision and recall. Both Precision–Recall and F1-Confidence analysis showed that the proposed model works efficiently under any conditions and maintains a good balance between accuracy and robustness. In addition, the model can recognize objects and generate relevant alarms based on the situation.
As for the hardware part, the use of a processing module with a dedicated controller allows for effective management of all communication processes and output devices, such as lights, sound, and display. Video recognition, identification of objects, and the transmission of instructions via serial communication have been successful. It means that the implementation of the project in practice is feasible.
Moreover, it is worth noting that another significant contribution achieved in this project lies in the cost-effective nature of the developed intelligent pedestrian safety system. Being composed of inexpensive hardware components including Jetson Nano, ESP32, and output modules, the system represents a cost-efficient alternative to currently available solutions to smart traffic management.
Nevertheless, it should be mentioned that the intelligent pedestrian safety system being developed does have several disadvantages. Thus, detection

performance will drop somewhat when faced with unfavorable circumstances like bad weather conditions, severe traffic congestion, and occlusion. Apart from that, it should be noted that the presented system detects only two types of objects.

Thus, it can be concluded that the intelligent pedestrian safety system being developed proved to be successful in terms of reliability and accuracy. It is also a low-cost alternative to existing smart traffic control methods.

7.2 Future Recommendations

Although the proposed Smart Crosswalk System has performed well, there are several ways in which the work can be enhanced in the future. The accuracy of the system can be further improved with more advanced deep learning models such as more recent versions of YOLO or transformer-based detectors. The size and diversity of the training data will also be expanded to improve the performance under the unfavorable conditions such as low light, rain, and crowded places. The other notable enhancement would be to incorporate vehicle speed and distance estimation which would allow the system to make wiser decisions and provide more precise warnings.

In addition, hardware upgrades such as more powerful embedded devices and more complex models may be used to increase the processing speed as well. In addition, the system can be connected with the existing traffic infrastructure and may be energized using renewable energy sources such as solar energy to enhance efficiency and sustainability. These extensions will make the system stronger, scalable and fit the real-world application.

References

[1] Y. S. Lee, W. C. Yeh, and C. Y. Chen, "Deep learning-driven vision-based crosswalk navigation with history-based and cascaded control," in *Proceedings of the IEEE International Conference on Advanced Intelligent Systems*, (Delft, Netherlands), July 2025.

[2] H. S. Park, J. K. Kim, and P. K. Sharma, "Smart crosswalk management with vehicle-to-pedestrian communication," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, (Hong Kong), Oct. 2023.

[3] W. K. Saad, Y. Hashim, and W. A. Jabbar, "Design and implementation of portable smart wireless pedestrian crossing control system," *IEEE Access*, vol. 8, pp. 106109–106120, 2020.

[4] F. Zhang, L. Wei, and H. Abdullah, "Design and implementation of portable smart wireless pedestrian crossing control system," in *Pro-*

ceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), (Bilbao, Spain), Sept. 2023.

[5] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, 2016.

[8] A. M. Al-Humaidi, S. Rahman, M. Khan, and J. Smith, "Smart system of a real-time pedestrian detection for smart city," in *Proceedings of the IEEE Conference on Smart Cities and Urban Informatics*, (Florence, Italy), June 2023.

[9] L. Falaschetti, L. Manoni, L. Palma, P. Pierleoni, and C. Turchetti, "Embedded real-time vehicle and pedestrian detection using a compressed Tiny YOLO v3 architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 12, pp. 19399–19414, 2024.

[10] NVIDIA Corporation, *NVIDIA Jetson Nano Developer Kit User Guide*. NVIDIA Corporation, 2020.

[11] R. Singh and P. Kumar, "Vision-based pedestrian detection for smart cities using deep learning," in *IEEE Smart Cities Conference*, 2023.

[12] Y. Liu and Q. Chen, "Vehicle-to-pedestrian communication for smart crosswalk safety," in *IEEE Vehicular Networking Conference*, 2022.

[13] S. Saponara and A. Elhanashi, “Real-time video surveillance for smart cities using deep learning,” *IEEE Access*, vol. 8, pp. 123456–123470, 2020.

[14] C. Chen, B. Li, and Y. Zhang, “Real-time pedestrian detection using yolo-based deep learning model,” *IEEE Access*, 2021.

[15] D. Kim and S. Lee, “Intelligent traffic control system for pedestrian safety using ai,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[16] T. J. Mateo Sanguino, J. M. Lozano Domínguez, M. J. Redondo González, and J. M. Davila Martin, “New approach to intelligent pedestrian detection and signaling on crosswalks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 12, pp. 21416–21428, 2024.

[17] K. Patel and R. Mehta, “Enhancing road safety using ai-based pedestrian detection systems,” *IEEE Transactions on Vehicular Technology*, 2023.

[18] L. Garcia and M. Torres, “Embedded system for real-time pedestrian detection and warning,” in *IEEE Embedded Systems Conference*, 2021.

[19] Kaggle Inc., “Kaggle: Your machine learning and data science community.” `https://www.kaggle.com`. Accessed: 2026-05-01.

[20] Roboflow Inc., “Roboflow.” `https://roboflow.com`. Accessed: 2026-05-01.

[21] J. Wang, H. Liu, and X. Zhao, “Smart crosswalk system based on iot and computer vision,” in *IEEE International Conference on Internet of Things*, 2022.

[22] Z. Ali and M. Hassan, "Iot-based smart pedestrian crossing system for urban roads," *IEEE Access*, 2022.