DEVELOPMENT OF A COMPUTER VISION-BASED SYSTEM FOR MONITORING AND DETECTING UNAUTHORISED PARKING OF TRICYCLES AND MINIBUSES IN ENUGU CITY

BY

OKENWA CHUKWUKA

GOU/U22/CSC/815

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF NATURAL SCIENCES AND ENVIRONMENTAL STUDIES, GODFREY OKOYE UNIVERSITY, ENUGU STATE. **IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE.**

SUPERVISOR: ENGR. CAMILLUS NJOKU

JUNE, 2026.

APPROVAL PAGE

This research, titled "**Development Of A Computer Vision-Based System For Monitoring & Detecting Unauthorised Parking Of Tricycles and Minibuses in Enugu City**," has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Natural Sciences and Environmental Studies, Godfrey Okoye University, Enugu, Nigeria.

Engr. Camillus Njoku	....	....
Supervisor	Signature	Date
External Examiner	....	....
External Examiner	Signature	Date
Dr. Frank Okebanama	....	....
HOD, Department of Computer Science and Mathematics	Signature	Date
Prof. I. A. Ajah	....	....
Dean, Faculty of Computing Information Technology.	Signature	Date And

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

____

OKENWA CHUKWUKA

GOU/U22/CSC/815

DEDICATION

This project is dedicated to my mother, brothers and extended family for their continuous motivation and support throughout my educational life.

ACKNOWLEDGEMENTS

First and foremost, I give all glory, honour, and gratitude to Almighty God for His grace, guidance, strength, wisdom, and protection throughout the course of this research work and my academic journey.

I sincerely express my profound appreciation to my project supervisor, **Engineer Camillus Njoku**, for the valuable guidance, constructive corrections, patience, encouragement, and support provided throughout the development of this project. Your supervision greatly contributed to the successful completion of this work.

I am also grateful to the Head of Department, **Dr. Frank Okebanama**, and all the lecturers in the Department of Computer Science, Godfrey Okoye University, for their academic guidance, knowledge, and contributions towards my growth as a student.

My heartfelt appreciation goes to my mother, **Mrs. I.H Okenwa**, and my uncles, **Engr. I.E Anizoba** and **Uncle Ogueji** for their prayers, sacrifices, financial support, encouragement, and unwavering belief in me. I am equally grateful to my brothers for their support and motivation throughout my studies.

I am equally thankful to my friends and colleagues, particularly **Okolo, Oshiore, Chibuko, Jamike, Kechere, Dukyoon, Oziri, Nsude** and **Mark** for their support, advice, motivation, and assistance throughout this academic journey. May God bless you all abundantly.

ABSTRACT

In this paper, a fully automatic vision-based detection system for detection of unauthorized parking violation by Keke NAPEP tricycles and commercial minibuses in no-parking zones in Enugu metropolis, Nigeria has been proposed and evaluated. This includes designing, developing and evaluating the performance of a combination of custom-trained YOLO11s (You Only Look Once, small variant) object detection model with Region of Interest (ROI) enforcement model based on polygon, delivered through PyQt6 desktop application. Unlike the traditional rule based/sensor-based parking monitoring systems, the present system was developed by training with custom annotated data set consisting of 175 images of Keke and 115 images of minibus, manually labeled using CVAT and curated using Roboflow, making it adaptable to the visual features of the environment in Enugu city. The YOLO11s model has been fine-tuned using transfer learning on pre-trained COCO weights over 50 epochs, batch size 16 and input resolution 640x640 pixels on Google Colab GPU machine. In violation detection, a check has been made for centroid of the bounding box of the detected vehicles whether it falls inside the polygon of the illegal zone using OpenCV polygon intersection method for a number of consecutive frames. Performance evaluation of the trained model was done based on the object detection metrics; Precision, Recall, F1-Score, Average Precision (AP) and mean Average Precision (mAP) at Intersection Over Union (IoU) of 0.5 and 0.5:0.95 (mAP@0.5 and mAP@0.5:0.95). For the validation data set, the trained model scored a Precision of 0.758, Recall of 0.832, mAP@0.5 of 0.838 and mAP@0.5:0.95 of 0.607, above the threshold of 0.80 mAP@0.5 as required by the project. For the class level results, the Keke class had an Average Precision of 0.946 and the bus class had an Average Precision of 0.716. The highest F1-Score of 0.78 was realized at confidence score of 0.539 and maximum combined recall of 0.98 at confidence score of near zero. These results show that the YOLO11s model has met a high and operationally robust standard of detection accuracy, especially for the Keke class. From these results, it is evident that it is technically feasible to deploy lightweight and real time computer vision systems for automated traffic law enforcement even in cities like Enugu.

Table of Contents

List of Tables

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Traffic monitoring has developed over the years through new technology that supports officers in monitoring (Bautista et al., 2016). A significant challenge to urban traffic flow is the absence of designated bus stops or terminals, which forces vehicles to pull over at unofficial or illegal locations along the road. This, combined with on-street parking, creates bottlenecks that disrupt the movement of other road users. The indiscriminate loading and discharging of passengers at undesignated points further compounds the problem, contributing to congestion, safety hazards, and an overall lack of order on the road network (Bashiru & Waziri, 2023). Gwilliam, (2023) stated that the increase of the urban population taken into account with the increase of economic activities which has caused a high demand of transportation architecture, with the rate being faster than the development of proper roads and public transport systems. This issue has been embodied by Enugu city, an urban powerhouse in the southeastern region of Nigeria. Being the capital of Enugu state, it has seen a positive shift in its demography which has had a negative impact on its transportation architecture (Vlahov et al., 2025). From overwhelming traffic jams, prolonged travel times, increased air pollution to an elevated risk of traffic accidents (World Health Organization, 2017).

Adetunji & Akintayo, (2026) discovered that usually, urban transport systems are made up of public and private modes; such as buses, trains, cars, bicycles as well as tricycles and minibuses which are unconventional modes that are commonly used in developing countries. Both formal and informal modes make up Nigeria's urban transportation system as efforts have been made to develop public transportation systems in both Abuja & Lagos still a substantial percentage of urban transportation challenges are addressed by informal transportation modes, made up of tricycles (keke) and minibuses. As Raji & Waziri, (2016) noted that tricycles have become a popular means of public transport in Nigerian cities. The National Poverty Eradication Programme (NAPEP) established it in 2001 as a means to create employment and provide affordable transportation. The role tricycles play in urban mobility cannot be emphasized enough. In urban and suburban areas, they are widely used for short trips and allow end-to-end accessibility. They also operate effectively in congested roads and narrow routes where larger vehicles are ineffective; making them a preferred option for numerous commuters specifically in working-class areas that are highly populated (Adewumi & Allopi, 2019). Research by Adenyi, (2017) revealed that tricycles usually carry out their activities in semi-regulated or unregulated settings, offering affordable and flexible travel methods. According to Olawole & Aloba (2018), informal transportation modes lead to safety and congestion issues as they often carry out their activities with barely any regulations. There are several limitations in Nigeria's present traffic violation system which include: time delays, high implementation cost, space occupation and lack of document integrity (Ebimobowei, 2021).

Even so, several challenges including regulatory issues, environmental impact, insufficient infrastructures are posed by the incorporation of tricycle transport into the transportation system (Cervero & Golab, 2017). Manisalidis et al. (2020), Ojojo et al. (2007), Barwari et al. (2021), all discovered that due to the absence or poor enforcement of road transportation policies, numerous tricycles operating in Nigeria are either outdated or poorly maintained. Subsequently leading to poor fuel combustion thus harmful substances including particulate matters (pm), nitrogen oxides (NO_X), sulfur oxides (SO_X), and carbon oxides (CO_X) get released into the atmosphere.). A atudy by Jayatilleke et al. (2010) found out that 53.5% of traffic violations were committed by public service vehicles among all vehicle types on the road.

As Kayami, (2019), Tahir, (2018) and Batool, (2012) all argue that inefficient enforcement of traffic regulations cam be attributed to several factors such as; poor coordination among agencies, limited capacity within enforcement bodies, conventional bribery and corruption in law enforcement, patronage culture in public transport, inadequate use of technology in enforcement, unfair processes and improper demeanor among drivers and the general public

According to Kathait et al, (2025) recent developments in the fields of computer vision and deep learning, specifically in the areas of object detection and multi-object tracking, have allowed researchers to identify vehicles and their spatio-temporal trends with high accuracy.

Existing public surveillance camera systems, available in big cities, can be used for this purpose, which eliminates any extra cost of deployment. Majority of the models which have been developed for this research study have been focused on the context of first and secondworld countries who have more standard road infrastructures and there are very few studies that have been carried out in a demographic such as the city of Enugu with its unique constraints. Therefore, an automated monitoring system that focuses on the unique constraints of Enugu is needed.

1.2 Statement of The Problem

These problems are recognized in the current means of detecting and monitoring unauthorized parking of tricycles and minibuses;

I. Limited Manpower: Enforcement officers who are deployed in congested areas are usually overloaded with too much work.

II. Bias In Officiating: Most personnels are often unjust in enforcing these rules and punishing violators as they do not see the greater effects it has on society.

III. Time Consumption: The manual approach is time consuming as traffic personnels carry out their operations in a traditional manner; which may lead to them not detecting more traffic violations.

1.3 Aim and Objectives of the Study

The study aims to develop and implement a computer vision system for monitoring and detecting unauthorized parking of tricycles and minibuses in Enugu city. To achieve the stated aim, the following specific objectives were pursued:

I. Identify the current means of detecting and monitoring unauthorized parking of tricycles and minibuses in Enugu.

II. Train a computer-vision-centered system for the detection and monitoring of unauthorized parking in tricycles and minibuses.

III. Evaluate the reliability of the proposed system through relevant metrics to assess its accuracy, stability and applicability in real world deployment.

IV. Design a web app to integrate the model for real-world applicability.

1.4 Significance of the Study

To understand the real-world significance of this project, it is essential to look at the immediate environments it aims to improve. The deployment of this technology creates a direct, positive ripple effect for the following essential groups in Enugu city:

I. Traffic Control and Law Enforcement Bodies in Enugu: For agencies such as the Enugu State Transport Company (ENTRACO) and the Enugu State Ministry of Transport, the study offers a practical guide based on evidence for managing traffic within cities. Through the implementation of automated computer vision surveillance technology, the organizations will be able to move from a reactive form of manual patrol to an automated surveillance system that will enable optimal resource allocation and identification of high violation areas within the transit zone.

II. Pedestrians and Urban Commuters: The people who have the greatest amount to gain from this on a daily basis are the pedestrians who must use the walkways in Enugu city. Through the discouragement of tricycles and minibuses from using the walkways that they are not supposed to use, this system is helping to ensure the safety of citizens by making it much less likely that there will be an accident involving pedestrians.

III. Local Shop Owners and Enterprises: For the business people who operate their businesses along the main roads of Enugu, unauthorized parking of vehicles causes economic conflict for them immediately. Parking of tricycles and minibuses directly in front of shops not only blocks the flow of customers but also makes it difficult to deliver merchandise. An organized parking system will help to create visibility and access for the customers.

1.5 Scope of the Study

The scope of this study is limited to the identification and tracking of only two types of commercial vehicles which include tricycles (Keke) and minibuses. These two types of vehicles are the main sources of localized congestion and illegal parking in major roads of Enugu. Any other type of transportation mode including but not limited to cars, heavy duty trucks or motorcycles is not included in the detection and monitoring process.

It should be noted that the scope of this detection system is limited to a detection and monitoring system only. It does not serve any law enforcement function nor does it automatically issue any penalty against offenders. The system can only detect and log cases of violations by offenders and visualize this data spatially.

The scope of the system excludes: Automatic License Plate Recognition (ALPR), fines generation, calculations and distribution and any other automated function associated with law enforcement activities against offenders.

1.6 Limitation of the Study

Limitations that are inherent to the study arise due to the fact that the dataset, flow of traffic, and environment that was used to train and test the convolutional neural network (CNN) model are limited solely to this particular area of Enugu City.

This limitation is important due to the fact that the behaviors of the transit in cities in Nigeria are not the same. Driving culture of tricycles (Keke) operators and minibus drivers, as well as roadside businesses, varies from state to state. As a result, the model developed in this study is suited specifically for Enugu City and cannot be applied in another city without further collection of data and training of the model.

1.7 Operational Definition of Terms

I. Computer Vision: An area of artificial intelligence where computers are trained to make sense of visual information through image and video analysis.

II. CNN (Convolutional Neural Network): A specific deep learning model that is used mainly for dealing with structured and grid-structured data like images and videos.

III. Tricycle: A human-powered three-wheeled vehicle commonly used in transportation.

IV. Minibus: A small bus, usually for transporting people short distances.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

It takes a multidisciplinary approach that encompasses the areas of intelligence transportation systems, computer vision, deep learning, spatial analysis, and socio-technology of mobility within developing economies to develop and deploy automated visual systems in enforcing urban traffic rules. The current chapter focuses on an in-depth, scholarly review of existing studies that relate to the development of a computer vision technology for detecting unauthorized parking of motorized tricycles (Keke) and commercial minibuses in the city of Enugu, Nigeria. The scholarly review forms the basis for understanding the theoretical background for methodological decisions underpinning this project.

The chapter will be subdivided into two main parts. The introductory part will provide an overview of the research in terms of the existing scientific literature related to the subject matter, which underscores the need for a multi-disciplinary literature review as the basis for a complex research project such as this one. The second part, titled theoretical background, will explore key concepts and research trends underlying this study. Some topics to be covered are the concepts surrounding ITS technology and its implementation difficulties in less developed countries; traffic management and parking offense in Nigerian cities; socioeconomic aspects of tricycles and minibuses transportation system in Nigeria with special regard to the city of Enugu; theoretical background of computer vision in traffic management; principles of deep learning and CNNs as an algorithmic framework for object detection; historical development and comparison of object detection methods, paying special attention to the family of YOLO models; technical specifications of YOLO11 algorithm utilized in the course of the project; theoretical background of polygon-based zone identification algorithm for parking offense detection; the need for transfer learning in specialized object detection applications; and the significance of sophisticated dataset annotation tools (CVAT and Roboflow) in generating customized datasets.

Overall, all the above sub-sections have come together to create a theoretical basis that explains the present level of understanding, the shortcomings of prior theories, and the academic and applied significance of the present system.

2.2 Theoretical Background

2.2.1 Intelligent Transportation Systems

ITS is a combination of integrated technological systems that utilize the capabilities of ICT, comprising various sensors, computing, artificial intelligence, satellite navigation, and data analysis in the design and management of road transport infrastructure. In general, the primary objective of ITS is to increase the safety, effectiveness, sustainability, and accessibility of road transportation through the use of real-time data gathering, analysis, and action. There are various subsystems associated with the ITS framework. For example, Advanced Traffic Management System (ATMS) uses surveillance cameras, loop detectors, and adaptive signal controls to ensure the optimization of traffic flow. Advanced Traveler Information Systems (ATIS) is used to provide commuters with the latest travel updates, while Advanced Public Transport Systems (APTS) improve the efficiency and predictability of bus and train transport services using vehicle tracking and passenger information integration. All of these systems attempt to ensure fewer human errors, decreased congestionrelated delays, low transport emissions, and compliance with road rules.

The implementation of the ITS has been attributed to transformative results in traffic flow, accidents prevention, and enforcement of regulations in the developed countries in North America, Western Europe, and East Asia. However, there has been increasing concern in recent academic studies about the possibility of realizing these transformative results within the totally different environments of developing countries. In their study on the successful framework of ITS implementation in sub-Saharan Africa and South Asian developing countries, Diderot et al. (2023) discovered that the technologies that have achieved notable success in high-income countries are poorly suited to the infrastructure, finance, and institutional structures of the developing world. This is according to their findings reported in the journal Procedia Computer Science. In their study, the authors found that the lack of adequate road infrastructure, lack of reliable broadband or wireless network coverage, lack of institutional capacity, the prohibitive cost of acquiring the technologies, and the lack of national ITS policies were the biggest hurdles to ITS implementation in developing countries.

More importantly, they observed that an ITS strategy in developing countries should be adapted to suit local conditions in terms of cost, complexity, and governance needs.

These constraints will have implications in the development of the current project. An automated parking enforcement system, using technologies such as inductive loop sensors, license plate recognition cameras, wireless networks, and server systems used in enforcing parking regulations, will not be feasible and cost-prohibitive in the current situation of the city of Enugu. As a result, significant efforts have been made in exploring alternative methods using less costly technologies, which utilize camera networks and standard computers. Within this line of ITS research, the current project can be placed, where it presents a ready-to-use solution, based primarily on software solutions and machine learning techniques.

2.2.2 Management of Urban Traffic and Parking Offenses in Nigeria

Management of urban traffic in Nigerian cities involves complex problems arising from the combination of the following factors: the explosion of population, inadequate road network, inadequate enforcement capabilities, and the dominance of informal and para-transit transport systems. Nigeria's major cities, including Lagos, Abuja, Kano, Port Harcourt, Onitsha, Enugu, among others, have seen dramatic increases in the number of vehicles within their jurisdiction over the past twenty years. However, the road and traffic control system of Nigeria’s major cities is lagging behind, thus causing the traffic problem to become an economic burden for these cities.

One of the most enduring causes of traffic problems in Nigeria's urban settings is that of illegal parking; a situation where commercial and private vehicles illegally park or load passengers at no-parking areas, along main roads, at crossing points, and near terminals for long periods during times of high traffic volumes. Such parking not only makes road space inadequate but also poses a threat to motorists as it interferes with their visibility at crossroads and endangers pedestrians who walk along the streets as well as other vulnerable road users. Illegal parking is made worse by the lack of adequate measures to check such activities; fines, for example, are not strictly enforced, contestable, and paid informally as bribes to traffic officers.

In all the literature related to the topic of automated parking violation detection, it is widely acknowledged that manual enforcement does not represent a viable option in terms of scalability. As noted by Kathait et al. (2025), within the discussion of their deep-learningbased parking detection system, traditional methods of enforcement which rely on the capability of law enforcement officers and members of the public to visually identify the presence of such violations are at once costly, delayed, and inadequate for the task of ensuring that violations are addressed across a sprawling city. In Nigeria, the issues of inadequate staffing, chronic underfunding, and an established pattern of transactional relationships make the problem of parking violations particularly acute, but it is also apparent that a vision-based system of automated enforcement will represent the more advanced and effective response to the issue.

The Enugu city, which is the capital of, and serves as a center of commerce in Enugu State, situated in the south eastern part of Nigeria, is a good case study of the shortcomings inherent in the traffic regulation policies implemented in most middle sized urban centers in Nigeria. The design of the roads in Enugu city, which was initially designed during the colonial era to serve much less traffic, consists of relatively few arterial routes, which cause an imbalance in traffic volume at specific nodes. Traffic generating developments such as Ogbete Market, the artisan market, hospital complexes, the University of Nigeria Teaching Hospital route and the central business district exist in a region where the design of roads is unable to cope with increased traffic volumes. Unauthorized parking of tricycles and minibuses is rampant in such areas.

2.2.3 Motorized Tricycles (Keke) and Minibuses as Urban Public Transport Vehicles in Nigeria

The motorized tricycle commonly referred to by its name as 'Keke' or 'Keke NAPEP', where the term NAPEP refers to the national poverty eradication programme, is one of the common modes of urban transportation in Nigeria. The Federal Government of Nigeria in conjunction with NAPEP established the Keke in 2001 for employment purposes as well as poverty alleviation programme in the country. It is a type of three wheeler vehicle, usually with a rear mounted petrol engine as well as a motorbike-like steering mechanism, capable of carrying two to three passengers in addition to the driver, all sitting under a roofed canopy. The advantage with this mode of transport is that it can travel on both main arteries as well as smaller local lanes in the towns and cities due to its ability to cover the first and last mile gap that buses cannot reach.

According to Ajiboye et al. (2020), in a comprehensive empirical analysis of tricycles in operation within the city of Minna, Niger State, Nigeria – a typical representation of the urban dynamics of northern and eastern Nigeria – it was found out that tricycles, specifically the Keke NAPEP, have become the major means of public transportation in Nigerian cities, due to the inadequacies of the formal public transit systems, which leave significant gaps for such transport means to fill. From their study, it emerged that citizens of Nigerian cities view tricycles as a cost-effective and reliable means of urban transport. Furthermore, it emerged that the ownership of a large percentage (38.2%) of the Keke NAPEP is through hirepurchase, thus putting pressure on the owner to ensure that revenue from trips is maximized daily, leading to unsafe operational practices.

The operating features of Keke in the city environment in Nigeria present some distinct problems for traffic regulation and policing. The drivers of tricycles, who are often not trained in transportation, drive with licences which they do not have through proper channels and also drive tricycles which are not registered in the transport department, tend to be associated with higher levels of violations of traffic rules. There have been reported cases of tricycles driving against traffic, making unscheduled stops and loading their passengers at forbidden places and parking in no parking zones around commercial centers, hospital entrances, and schools. This problem, as identified by studies conducted by FUDMA International Journal of Social Sciences around urban Katsina, had led to the gradual increase of traffic congestion on the roads and traffic accidents in urban centers in south-eastern parts of Nigeria like Enugu.

Another significant class of commercial transport vehicles operating in Nigerian urban settlements is the commercial minibuses. These vehicles run alongside tricycles and contribute uniquely to the issue of unauthorized parking. Commercial minibuses in Nigeria are called Danfo in Lagos and Molue in popular terminology; alternatively, they go by other names in south-eastern cities where such commercial vehicles are used. Typically, these vehicles run along semi-defined routes linking different residential areas to the market and business areas. A critical feature of the Nigerian commercial minibus is that it stops for the purpose of loading at any point along the route, without the necessity of a bus stop or a loading bay. In other words, it stops anywhere along the route when the driver or tout calls out passengers on the side of the road, leading to unauthorized parking at the highly crowded junctions, in addition to the violations committed by the tricycles.

2.2.4 Introduction to Computer Vision for Traffic Monitoring Systems

Computer vision is defined as the branch of science and technology which is concerned with the construction of computing mechanisms able to extract semantic information from digital imagery. Computer vision has emerged as one of the most rapidly developing fields in artificial intelligence research thanks to several decades of work in areas such as image processing, pattern recognition, statistics, and more recently, deep learning. In the context of traffic monitoring systems, computer vision has been applied to achieve an array of increasingly advanced functions, including vehicle detection, classification, and counting; traffic flow estimation; traffic density estimation; anomaly detection; license plate recognition; pedestrian behaviour analysis; and finally, parking violation detection.

The use of computer vision for surveillance and enforcement of traffic has received an immense boost owing to two complementary trends: the deployment of low-cost surveillance cameras connected to the Internet in urban areas and the rapid improvement in the effectiveness of deep learning models in visual recognition tasks since 2012. In a study that introduced TrafficSensor, a framework based on deep learning techniques for automated vehicle tracking and categorization through fixed-calibrated cameras on highways, Fernández et al. (2021) made a compelling case for how advances in neural network design have made it possible to use cameras to monitor traffic not just effectively but also more economically than other technologies. As was seen in their paper, an appropriately trained deep learning model on one single stream of camera data could yield accurate counts of vehicles along with their speeds and classes, setting a standard that surveillance technology like the one being discussed here aims to emulate.

In this respect, the new area of research into automated parking violations detection based on computer vision deserves special attention. Indeed, according to Kathait et al. (2025), it has become possible to perform real-time, scalable parking enforcement without any specific equipment except for regular IP cameras by creating a pipeline that combines publicly available video streams with an advanced model of object detection and a component of spatial analysis based on polygons. The authors have implemented a temporal criterion which stipulates that only those vehicles whose presence in a particular restricted zone exceeded a specified threshold would be considered violators. Such a design choice has helped reduce false positives to a minimal level by differentiating actual violations from occasional stops by passing vehicles. In view of this approach being widely recognized among scholars, the current study has chosen to implement a similar criterion in its own design.

With regard to developing countries, academic research has provided evidence that complex computer vision systems are not necessarily confined to the high-end, high-infrastructural setting. Specifically, Murindanyi et al. (2025), conducting their research in Uganda—which, like Nigeria, features limitations in its traffic surveillance infrastructure and faces deployment challenges typical of the development context—described an intelligent traffic surveillance system in which a combination of YOLO vehicle detection, license plate recognition, and automatic fines issuance were incorporated into the process. In their research, the authors were able to achieve a mean average precision of 97.9% in license plate recognition, and they confirmed that their solution was effective at functioning in real-life conditions within the Ugandan road environment, where there would be variations in lighting, varying kinds of vehicles on the roads, and low-resolution video footage.

2.2.5 Deep Learning and Convolutional Neural Networks

Deep learning, the sub-field of machine learning that involves the training of artificial neural networks made up of multiple representational layers within a hierarchical architecture, has revolutionized the upper limits of computer vision performance in the past 15 years or so. Before the advent of deep learning, computer vision systems relied on hand-crafted features, i.e., engineering mechanisms like Histograms of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), Local Binary Patterns (LBP), and Haar cascades, which were difficult to engineer and highly susceptible to environmental conditions, such as variations in illumination, viewpoint, size, and occlusion. It was not until 2012 that deep learning took off with a bang, when Krizhevsky et al. trained a deep convolutional neural network on the ImageNet Large Scale Visual Recognition Challenge dataset, beating out other classical approaches with a top-five error rate of 15.3%, while the classical approach came close behind at 26.2%.

Central to modern computer vision systems is the Convolutional Neural Network (CNN), which is an architecture that has been specially created for the task of learning hierarchies of visual features using successive layers of learnable convolutional kernels. In a comprehensive survey about various convolution techniques and their use in deep neural networks, Younesi et al. (2024) explained the working of CNNs as follows. Using successive learnable kernels on input feature maps at different spatial resolutions allows the neural network to progressively learn different visual features at different network layers. While early layers of CNNs learn low-level visual features such as edges, contrast and textures, the intermediate layers learn mid-level features such as shapes and patterns, and the deepest layers learn the high-level representation of objects and scenes. This unique ability of learning the visual features from simple to complex levels is what makes CNNs extremely effective for visual recognition tasks as they can learn both low-level and high-level visual features using a single model.

The architectural building blocks of the CNN are well-known and deserve an explanation at this point for the theoretical foundations of the current project. The convolutional layer applies locally weighted sums of the activations of the input with the learned weights of the filter of usually small dimensionality, which results in the feature map capturing local spatial structures. The pooling operation reduces the resolution of the feature map, keeping track only of the maximum activation in each pooling window, thus providing a level of translational invariance along with reduced computation cost. ReLU and related activation functions inject the necessary nonlinearity, making the network capable of approximating non-linear boundaries impossible for linear architecture. The batch normalization technique ensures that the intermediate outputs have zero mean and unit variance, thus improving stability during gradient descent, allowing the usage of larger learning rates. Classical CNNbased classification models have a fully-connected layer that combines feature maps and produces output probability distribution per each possible class; contemporary detection models replace or supplement this part with the detection head that predicts bounding boxes and their respective classes.

The current empirical performance of object detection systems based on convolutional neural networks has already achieved or surpassed the performance of humans on many wellestablished benchmarks such as COCO, Pascal VOC, and ImageNet. These results have been possible because of the synergy between three necessary conditions: the existence of annotated datasets containing many different objects, the prevalence of GPU computing equipment for making training of deep architectures feasible, and advances in algorithms like residual connections, dense connections, attention mechanisms, and network architecture search that allow the creation of more accurate networks. In this project, these technologies will be capitalized upon in order to create a detection model using CNN-based backbones.

2.2.6 YOLO Family of Object Detection Networks

The You Only Look Once (YOLO) model, proposed by Redmon et al. (2016) in their seminal paper delivered at the IEEE Conference on Computer Vision and Pattern Recognition, marked a significant departure from existing approaches to object detection by considering object detection as a one-shot approach wherein the entire image is used for detection and prediction of classes in one go through convolutional networks.

In the initial version of YOLO, the input image is subdivided into an $S \times S$ grid, where each cell predicts a predetermined number B of bounding boxes and their confidence scores along with the probabilities for object classes conditioned on the occurrence of an object within the grid cell. The whole process from the input image to output bounding boxes and classes is performed in a single end-to-end differentiable convolutional neural network, allowing for joint optimization of all the detection components via standard gradient descent. Base YOLO operates at 45 frames per second in current GPUs, while its light-weighted Fast YOLO version operates at 155 frames per second, thereby setting records for real-time performance in the detection systems with equivalent accuracy which no previous detector had yet achieved. According to Redmon et al., YOLO detects objects significantly fewer times in non-object regions compared to other detection algorithms since it considers the whole image globally and not individual region proposals.

The architecture of YOLO has seen several updates after its initial announcement, and each of them has come closer to the performance level of two-stage object detectors without sacrificing computational efficiency. Terven et al. (2023), in their extensive review of YOLO architecture models, ranging from YOLOv1 through YOLOv8 and YOLO-NAS, published in the journal Machine Learning and Knowledge Extraction, elaborately highlighted the innovations made in each version. YOLOv2 was a big step in the field, with anchor boxes (predefined bounding box shapes obtained from the clustering of training dataset bounding box shapes) that enabled accurate object localization with diverse aspect ratios, along with other improvements like batch normalization and multi-scale training; YOLOv3 used the much deeper Darknet-53 model having residual connections, along with a multi-scale detector that worked on three pyramid levels at three resolutions; YOLOv4 incorporated many training time enhancements in terms of augmented training data and improved losses like the use of CIoU, Mish activation functions, and CSP in the backbone; YOLOv5 made engineering advancements with mosaic data augmentation and anchor-based learning in the Ultralytics framework; and YOLOv8, yet again from Ultralytics, came with anchor-less detection via a decoupled head and a novel backbone C2f module.

YOLO-based architecture has been widely used and has proven itself successful in terms of detection of vehicles in urban environments and traffic violations. According to the literature reviewed by Kosasi et al. (2024) on the use of YOLO-based algorithms for vehicle detection, YOLO models, particularly, YOLOv4 and YOLOv8, have achieved vehicle detection precision higher than 90% in mean Average Precision in various environmental conditions (different illumination, partial occlusion, and diverse traffic). The authors state that YOLO family and its developments hold great promise when it comes to improving the effectiveness and accuracy of traffic monitoring in intelligent transportation systems. This serves as additional motivation behind the selection of the YOLO architecture for this study. According to Kathait et al. (2025), YOLO-based detection model is used as an object detection algorithm for detecting illegal parking, which further proves that YOLO family of algorithms is suitable for this type of application.

2.2.7 Architecture, Features, and Model Choice of YOLO11

The YOLO11 is the most recent iteration from the Ultralytics YOLO series developed by

Glenn Jocher and Jing Qiu at Ultralytics and officially launched in September 2024 via YOLO Vision 2024 conference. The architecture itself was developed with the clear intention of pushing the state-of-the-art to further optimize both accuracy and performance in parallel, as the ongoing tradeoff between precision and speed is inherent to any practically employed detector architecture. YOLO11 is built as a versatile single task architecture capable of performing object detection, instance segmentation, image classification, pose estimation, and oriented bounding box detection within one unified architecture (Jocher & Qiu, 2024).

The architectural advancements unique to YOLO11 in contrast to the preceding version, YOLOv8, are significant and require individual attention. Among the architectural advancements, C3K2, which stands for Cross Stage Partial bottleneck block and uses 3 x 3 convolutional kernel, is notable for its richer gradient flow paths and feature representation ability compared to the C2f block. A further upgraded SPPF module allows the model to incorporate contextual information effectively through pooling at different spatial levels. This function is highly essential in recognizing object at varying scales that appear within a single scene. For instance, objects at close distance alongside far objects are detected using this model. Notable among all the other improvements introduced in YOLO11 is C2PSA, a partial spatial attention mechanism that gives the network the ability to selectively attend to particular parts of the input feature map.

The benchmark results announced at the time of the YOLO11 release showed that the

YOLO11m version surpasses YOLOv8m in terms of achieving a higher mean Average Precision on the COCO object detection benchmark test, consuming 22% fewer parameters, which amounts to significant progress in terms of achieving an optimal balance between accuracy and parameter efficiency (Jocher & Qiu, 2024). These improvements have tangible application benefits: lower parameter usage leads to reduced GPU memory needs during both training and testing phases, reduced energy consumption at runtime, and improved speed on limited-computation devices, all of which have direct applicability to the current project deployment scenario. In addition, YOLO11 has been designed with cross-device deployment capability in mind, supporting efficient execution across embedded, mobile, and GPUequipped consumer devices, as well as NVIDIA Jetson boards and cloud-based inference frameworks.

In this study, the YOLO11s (small) network architecture was chosen to achieve optimal compromise between the precision of object detection and speed of inference that would be adequate for processing live video streams on typical consumer-grade devices without access to specialized high-performance GPU hardware. The small architecture represents the lower end of the range of scales available in the YOLO11 model and, compared to medium and large, provides much quicker inference while sacrificing only slightly in terms of the mAP. In a situation like this, when real-time processing of video streams is the bottleneck of the system's performance, such a solution is well justified. This decision follows the best practices for practical implementation of traffic surveillance systems that involve the allocation of computer resources among several tasks.

2.2.8 Transfer Learning for Domain-Specific Object Detection

The concept of transfer learning is described by the machine learning approach, which makes use of knowledge learned during the training of a model for the purpose of carrying out a source task using a rich and labeled dataset to aid in the faster completion of learning and performance improvement for a different target task using a limited number of labeled samples. The most common form of transfer learning in deep learning computer vision is through the process of initializing a detector or classifier using pre-trained weights from a large benchmark dataset. These include the famous ImageNet, a benchmark of more than 14 million images categorized into over 20,000 classes, or the COCO, which contains more than 330,000 images with 80 classes.

The reason why transfer learning makes sense for deep convolutional networks is that, based on empirical evidence, the early and mid layers of networks trained on large datasets of varying data have learned generic features, which means they have encoded information about lines, textures, color differences, and simple geometrical objects that can be used universally for any visual domain. On the other hand, the deep layers are highly specialized and need significant modification in order to fit the target domain in terms of the types of features it requires, as well as category distinctions and spatial arrangements. Through finetuning, it is therefore possible to gain much greater training data efficiency than through random initialization.

This particular aspect can be applied to the current problem because it requires the learning of reliable detection and localization of Keke tricycles and commercial minibuses, but in the very unique visual domain of the roadways in Enugu city for which there is not yet any annotated dataset available. Specifically, the YOLO11s model started out with its weights being pre-trained with images from the COCO dataset, which consists of 80 classes including vehicles, motorcycles, and bicycles. Then the model is trained with the newly collected dataset of the tricycles and minibuses in Enugu roadways labeled by CVAT. The approach allows the model to benefit from all the learned multi-scale representations that were learned from COCO's detection images while training the final detection network for the appearance of the particular vehicles in Enugu roadways.

2.3 Review of Related Works

Tang et al. (2020) designed an illegal parking detection system in real time through the application of deep learning. Tang et al. made advancements in the existing technique of Single Shot MultiBox Detector (SSD) through the implementation of mechanisms of contextual information delivery and spatial attention to ensure high performance in terms of vehicle detection and tracking. Their results showed better detection accuracy and real-time performance as compared to conventional SSD algorithms. The study concluded that the delivery of contextual information had helped in better identification of illegally parked cars in complex environments. But the algorithm's efficiency was hindered by heavy occlusion, poor lighting, and bad weather conditions. In their work, Jiang et al. (2020) suggested a computer vision technique for detecting parking violations using Convolutional Neural Networks (CNNs). In particular, the authors considered surveillance camera images as a source for detecting vehicles that park in restricted places. With the help of deep learningbased image recognition and object detection methods, the technique proved efficient in terms of detecting parking violations in laboratory settings. Thus, the study's findings allowed concluding that computer vision technologies have great potential in reducing manual traffic regulation and enforcement. Still, the model had significant training needs and did not work effectively under different conditions.

The study by Carrera García et al. (2022) explored an innovative intelligent parking control method based on artificial intelligence and machine learning. It comprised three stages, which included image capture, parking localization, and parking occupancy classification by using automatic learning algorithms. As a result, higher accuracy was attained for detecting empty and occupied parking spaces. Overall, the research confirmed the efficiency of computer vision in detecting parking occupancies and argued that the approach is more economical than sensor-based parking monitoring. However, the study focused only on detecting occupied parking spaces without mentioning any parking violations. Zhang et al. (2022) presented a machine vision-based system for recognizing the status of the parking space based on the YOLOv2 object detection algorithm. Deep learning algorithms were used for identifying vehicles and analyzing parking spaces using geometric mapping. As per experimental results, the training accuracy was found to be about 99.1%. This indicated the efficacy of using a YOLO-based system for parking spaces. Therefore, the findings suggested that machine vision systems offer real-time parking monitoring capabilities. Nevertheless, the system heavily depended on the presence of markings in parking spaces, leading to inaccurate results in cases where the markings were not well visible. Wang et al. (2022) suggested a curb lane monitoring framework for detecting illegal parking practices and analyzing their impact on traffic operations. The research design comprised of computer vision technologies, analysis of images taken by traffic cameras, determination of hotspots, and queueing theory. In particular, the study focused on detecting illegal parking practices and estimating traffic disruptions caused by them. The researchers concluded that computer vision technology offers opportunities for law enforcement and planning in the field of traffic. On the contrary, the developed framework was limited due to the use of traffic cameras and low-resolution videos. Berwo et al. (2023) explored a wide range of deep learning approaches to detect vehicles for classification in ITS (intelligent transportation systems). Algorithms such as CNN, YOLO, SSD, and Faster R-CNN were used to analyze the suitability of traffic surveillance systems based on deep learning methods. It became clear from this work that the models significantly outperformed conventional machine vision approaches in detecting cars and buses. The study came to the conclusion that successful vehicle detection is key to developing efficient parking violation detectors. Notably, this research was entirely theoretical in nature and did not propose a practical method for traffic violation detection. A new traffic violation detection framework supported by artificial intelligence was developed by Dede et al. (2023). This system could identify any violations related to illegal parking activities. In particular, the YOLOv5 object detection model together with StrongSORT vehicle tracking was employed to find traffic violations. The experimental results showed that traffic violations, including those related to parking, were accurately identified. Overall, the combination of object detection with vehicle tracking improved traffic violation recognition greatly. However, the described system consumed a significant amount of computing power and relied greatly on the quality of videos.

Ren (2024) created an intelligent car violation detection system by using computer vision technology and human-computer interaction. The method uses background modeling, motion analysis, and stationary objects detection to detect illegally parked cars within defined areas. This resulted in better performance, faster operation, and system reliability. Based on this result, it was revealed that monitoring vehicles in stationary mode in defined regions of interest is an effective way to detect unauthorized parking. Yet, the system still is sensitive to external changes including light variation and modification of the environment that can adversely affect background modeling.

In their research, Charde et al. (2024) designed an innovative artificial intelligence and Internet of Things based parking violation detection and notification system. Using AI, computer vision techniques, and IoT, this system allows automatic detection of parking violations and generation of enforcement notifications. The experimental results show efficient detection of illegally parked cars, accurate extraction of the vehicle's registration information. In conclusion, it was stated that artificial intelligence together with the Internet of Things improves automation in parking management systems. Still, the detection system experiences difficulties detecting blurred images, poor illumination, and partly hidden license plates.

There are many factors responsible for causing unauthorised parking in urban environments. The fundamental reason for this problem lies in the imbalance between the number of cars and the amount of parking spaces available. In the fast-growing metropolises of developing countries, where urban development and infrastructure construction have not kept pace with car population, the problem of unauthorised parking is very severe. For example, in Hanoi, Vietnam—a metropolis representing urban development in developing countries—Doan and Hiep (2017) found that the main cause of illegal parking is a lack of parking facilities together with inadequate spatial information provided to drivers. It is worth noting that in developing countries like Nigeria, the problem of illegal parking exists to a much larger extent due to improper urban structure and land use, because urban planners are unable to foresee the parking needs of future land uses (Doan & Hiep, 2017). Cultural issues and weak regulation, along with poor traffic enforcement, also contribute to the problem of unauthorised parking in some way. In their paper, Nguyen and Sartipi (2024) explored a smart-camera parking management system using the PakLoc and PakSta models for automatic identification and analysis of parking spaces occupancy. Their methodology was built around using computer vision algorithms for automatic recognition of parking spaces and detection of their occupancy status. They reported a 94% reduction in the number of parking spaces labeled manually. As a result, they concluded that automation of the parking spaces identification process greatly increases the scalability and feasibility of intelligent parking systems. Yet, this study focused on occupancy monitoring and did not address unauthorized parking detection in particular.

A comprehensive review of computer vision-based smart parking systems by Hadi, George, and Ahmed (2024) included an evaluation of existing datasets, image processing, deep learning approaches, and applications of computer vision in parking management. As a result of their analysis, they reported that camera-based systems and deep learning approaches became prevalent in the development of intelligent parking systems. Finally, they stated that computer vision is a promising scalable and affordable technology for use in parking management and traffic monitoring. Nevertheless, being a literature review, it did not provide any practical experience.

2.3.1 Summary of Literature Review

AUTHOR(S) & YEAR	WORK DONE	METHODOLOGY	RESULTS	FINDINGS	LIMITATIONS
Tang et al. (2020)	Developed a real-time illegal parking detection system using deep learning.	Enhanced Single Shot MultiBox Detector (SSD) with contextual information transmission and spatial attention mechanisms for vehicle detection and tracking.	Achieved high realtime detection performance and improved accuracy compared to conventional SSD models.	Contextual information significantly improved illegal parking recognition in complex scenes.	Performance decreased under severe occlusion, poor lighting, and adverse weather conditions.
Jiang, Feng & Li (2020)	Proposed a CNN-based illegal parking detection framework for urban management applications.	Used Convolutional Neural Networks (CNNs) and machine vision techniques to identify illegally parked vehicles from surveillance images.	Demonstrated effective detection of parking violations in controlled environments.	Deep learning can automate parking violation monitoring and reduce manual enforcement efforts.	Required large labeled datasets and struggled with environmental variations.
Garcia et al. (2020)	Designed an intelligent parking monitoring system using artificial vision.	Combined automatic learning techniques with computer vision to identify parking occupancy and vacant spaces.	Improved parkingspace localization and occupancy recognition accuracy.	Computer vision can support intelligent parking management with minimal sensor deployment.	Focused mainly on occupancy detection rather than unauthorized parking enforcement.

Zhang et al. (2022)	Developed a parking-status recognition system based on machine vision.	Applied YOLOv2 object detection to identify vehicles and parking spaces, followed by geometric analysis for occupancy determination.	Achieved approximately 99.1% training accuracy.	YOLO-based methods are effective for real-time parking monitoring applications.	Performance may degrade when parkingspace markings are unclear or occluded.
Wang et al. (2022)	Proposed a curb-lane monitoring and illegal parking impact estimation framework.	Combined computer vision, traffic-camera analysis, hotspot detection, and queueing theory to monitor illegal parking activities.	Successfully estimated the traffic impacts caused by illegally parked vehicles.	Computer vision can support both violation detection and traffic-management analysis.	Depended on fixed camera infrastructure and lowresolution video sources.
Berwo et al. (2023)	Conducted a survey of deeplearning techniques for vehicle detection and classification.	Reviewed CNN-, YOLO-, SSD-, and Faster RCNN-based approaches for vehicle detection in Intelligent Transportation Systems.	Identified deeplearning models as state-of-theart solutions for traffic surveillance tasks	Accurate vehicle detection is the foundation for illegal and unauthorize d parking detection systems.	Survey work; no implementatio n or experimental parkingviolation system.

Dede et al. (2023)	Developed an AI-assisted traffic violation detection system capable of detecting illegal parking.	Utilized YOLOv5 for vehicle detection and StrongSORT for object tracking across video frames.	Successfully detected multiple traffic violations, including illegal parking.	Combining detection and tracking improves the reliability of parking violation identification.	Computationally intensive and dependent on high-quality video streams.
Ren, (2024)	Developed an intelligent vehicle violation detection system integrating computer vision and humancomputer interaction.	Used background modeling, motion analysis, and stationary-object detection to identify illegally parked vehicles.	Improved detection accuracy, stability, and processing speed.	Illegal parking can be effectively detected by identifying stationary vehicles within defined regions of interest.	Environmental changes can affect background modeling accuracy.
Charde et al. (2024)	Proposed an AI- and IoTbased parking violation detection and alert system.	Combined computer vision, license-plate recognition, and IoT technologies for automatic violation reporting.	Successfully detected parking violations and captured vehicle license plates for enforcement.	Integration of AI and IoT enables automated monitoring and notification systems.	Sensitive to motion blur, poor lighting, and obstructed license plates.

Nguyen & Sartipi (2024)	Developed a smart-camera parking system with automatic parking-spot detection.	Proposed PakLoc and PakSta models for automatic parking-space localization and occupancy analysis using computer vision.	Reduced manual parkingspace labeling effort by over 94%.	Automated parkingspace identificatio n improves scalability of intelligent parking systems.	Focused primarily on occupancy management rather than unauthorized parking enforcement.
Hadi, George &	Presented a comprehensive	Examined datasets,	Identified major trends	Computer vision offers	-
Ahmed (2024)	review of computervision-based smart parking systems.	detection algorithms, and parkingmanagement applications.	toward deep learning and camerabased monitoring systems.	costeffective and scalable parking management solutions.	

Table 1: Summary of Literature Review

2.4 Research Gap

Despite the considerable amount of research discussed above in the previous sections, the analysis shows that there are certain identifiable, logical, and highly practical gaps within the existing body of knowledge that the current project is designed to fill. First, the most obvious gap relates to the dearth of computer vision papers that specifically discuss parking detection or surveillance of illegal parking activities in either Nigeria or the rest of West Africa. In particular, the vast majority of prior research has been directed at detecting either speed violation or license plate recognition, rather than addressing the issue of parking as a separate problem area. Secondly, the particular types of vehicles that generate the parking offenses in cities like Enugu namely, Keke tricycles and commercial minibus taxis have never been addressed in the computer vision literature in the context of parking or traffic enforcement. The second and highly related gap relates to the availability of training datasets with annotations. The widely used benchmark datasets in parking and vehicle detection studies, such as PKLot, CNRPark, and the MS COCO vehicle classes, were created based on Western/East Asian roads and do not contain any samples of the unique Keke/Ngwe-nge types of vehicles that are predominant in Nigeria's cities. It is due to the distinctive nature of Keke vehicles with three wheels and canopies which cannot be identified through Westernstyle vehicle-only datasets, and there is no publicly available annotated dataset for the Keke vehicles yet, according to the reviewed literature. Thus, the current project aims to fill in this gap by creating an annotated dataset of Enugu roadway images annotated in CVAT and processed in Roboflow.

Third, although polygon-based zone enforcement has been applied in a variety of new systems, there is no literature showing that it has been integrated into the YOLO system. Moreover, the application of this approach in the enforcement of roadside no-parking zones in an urban setting in Africa has not yet been conducted. Polygon-zone systems have been tested only in Western nations using designated parking areas, but not in irregular road shoulders and intersections where unauthorized parking occurs. It is in this area where informal means of transportation park their vehicles without proper permits. By applying polygon-zone enforcement in informal roads using customized training of YOLO11 for vehicle detection, the current study contributes an entirely new approach not covered in the review.

Lastly, a systemic gap exists in terms of the inclusion of representations of African urban road environments in computer vision studies; and in more ways than the one pertaining specifically to parking management. The preponderance of datasets used in training, evaluation, and deployment of vehicle detection algorithms has involved imagery collected in North America, Western Europe, and East Asia; and the choice of architectures and datasets that prevails within the literature reflects the characteristics of traffic, road infrastructure, and the vehicles found in these places. African urban environments are distinctive in that they feature heterogeneous compositions of formal and informal motor vehicle fleets, congested road spaces, diverse lighting and road conditions, and significant proportions of unique vehicle types. In this way, the current project makes its own small contribution to addressing this gap in representational capacity through building an image-based surveillance system based upon a dataset and architecture derived entirely from local images and vehicle types.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

This chapter provides an analysis and design of the proposed computer vision based detection and monitoring system for tricycles (referred to as "Keke" locally) and minibuses in Enugu metropolis. The chapter starts with an analytical assessment of the current approach adopted in monitoring and enforcement of traffic regulations with regards to unauthorized parking of the vehicles and identifies the limitations associated with the existing system. This is then followed by defining the objectives of the proposed system and development of a System Requirements Specification (SRS), which include both functional and non-functional requirements. The chapter goes further to provide the design approach selected and rationale behind it along with methods used for collecting data. The systems modeling in UML is done through use case diagram, activity diagram and class diagram. The chapter ends with the design of the system including input design, output design, database design and system architecture design.

With the increasing number of people living in the urban areas of Enugu State, there has been a considerable rise in the numbers of Keke (tricycles) and minibuses plying the city roads. While providing an important means of transportation to the residents of the urban cities of Enugu state, the vehicles have been identified as sources of traffic chaos because of the common practice of unlawful parking at unauthorized places. In light of the ineffectiveness of manual efforts in addressing this challenge, there arises the need for implementing a technology-based solution. The proposed solution employs the use of YOLO (You Only Look Once) family of real-time object detection models, with YOLO11s, that use the custom image dataset of roads in Enugu State.

3.2 System Analysis

The process of system analysis forms the basic building block of the software development process life cycle, which provides a clear structure through which to analyze the operating environment in which the system will operate. The analysis of the current state of traffic control in Enugu is conducted to highlight the shortcomings in the current system and thus identify the goals of the proposed system. This is done by conducting field studies, consultations with traffic control personnel, and literature research on traffic control in Nigerian cities.

3.2.1 Analysis of the Existing System

The present system that is used for the surveillance and enforcement of traffic rules on the unapproved parking of tricycles and minibuses in Enugu city depends wholly on the actual deployment of road and traffic control personnel. Traffic officers of organizations like ESURA and Road Safety Corps are deployed in different identified hotspots within the city where their work involves watching out, detecting, and apprehending the operators of vehicles that park illegally or operate against the specified traffic routes. In the process of carrying out their work, some of these officers make use of their personal phones in capturing videos of offenders while they are positioned in different blind spots where they cannot be seen by the vehicle operators.

The entire process of enforcement is reactive in nature rather than being proactive. The patrol officers conduct patrol around the designated routes or stay stationed at certain fixed points when traffic is at its peak, usually in the morning and evening times. Once a violation has been noticed by the officers, then they approach the driver in person and either impose the fine on the spot, seize the vehicle or take the driver to the nearest point of enforcement. The whole process relies completely upon the presence of the officers at that particular place and time of violation.

However, the information collected from these activities through data collection includes the number of vehicles seized, location of the most common violation incidents, and time at which violation is mostly committed, which is not analyzed or tracked properly. This makes enforcement decisions arbitrary, without taking into consideration historical trends or geographical analysis. There is no automation feature, remote access, or even any integration with digital media to make management decisions based on evidence collection for traffic violations. Using human beings as the only tool for detecting and enforcing compliance in this system makes it vulnerable to some limitations that will be highlighted in the following section.

3.2.2 Limitations of the Existing System

Several limitations affect the man-based approach to traffic control in the city of Enugu, limiting its effectiveness and making it less sustainable in the long run. These limitations are as follows:

i. Human Tiredness and Distracted Attention: Fixed location traffic control officers can suffer from exhaustion, boredom, and distraction as the job requires continuous observation through the visual sense. Manual surveillance is mentally tiring, hence the decrease in the awareness of the officers over time leads to the overlooking of traffic offences. Contrary to technology, humans are not able to maintain a constant level of accuracy throughout their shift.

ii. Geographical Restriction: The manpower used is limited due to the limitation in the workforce and available resources. Therefore, it can cover only a restricted number of points. Zones that do not have any traffic control officer are left unsupervised and therefore vehicles can park illegally within these areas.

iii. Susceptibility to Corruption and Compromise: Face-to-face enforcement transactions expose the process to corruption. Drivers can pay informal bribes to the enforcement officers to evade being sanctioned, which negates the entire enforcement process. The lack of an objective and automated means of recording any violations makes it difficult to address such corrupt practices.

iv. Lack of Digital Records and Evidence Management: In the current system, there are no digital records for violation, information about offending vehicles, or any enforcement transaction. Lack of a data infrastructure makes it hard to analyze trends or identify repeat offenders and even compile any report that could be used to develop policies or conduct legal proceedings. Informal information gathered using one’s personal phone cannot be used as structured evidence in an enforcement context.

v. Susceptibility to Environmental Factors: Manual monitoring is limited by poor weather conditions such as heavy rains and fog, which are common occurrences in the tropical environment of Enugu. Enforcement activities also become limited at night because of low visibility.

vi. Necessity for Scalability and Lack of Ability for Real-Time Situational Awareness Monitoring: This manual process is not scalable without proportional increase in number of staffs when there is an increase in volume of traffic as well as in the extent of urban area being monitored. Also, the system lacks any ability for real-time monitoring.

3.2.3 Objectives of the Proposed System

In light of the weaknesses and inefficiencies of the current manual detection and monitoring system, the objective of developing the computer vision-based detection and monitoring system is aimed at achieving the following:

i. Iidentification and characterization of how tricycles and minibuses parked without permission in Enugu city are currently detected and monitored, thus serving as the benchmark for comparison of the performance of the proposed system.
ii. Development of a detection and monitoring model of the vehicle types using YOLO11s with an annotated set of Keke and minibus images captured on the roadsides of Enugu.
iii. Rigorous evaluation of the reliability of the developed model using established performance evaluation metrics like Precision, Recall, Mean Average Precision at Intersection over Union=0.5 (mAP@0.5) and Mean Average Precision at Intersection over Union =0.5:0.95 (mAP@0.5:0.95).
iv. . Development of a web-based application with the detection model integrated into the same, allowing authorized personnel to upload images/video frames to the system and receive results of the detection.

3.3 System Requirements Specification (SRS)

The SRS specifies the functional and nonfunctional requirements that the proposed system should satisfy in an elaborate manner. The SRS acts as a reference document for the design and implementation of the system. This ensures that all the needs of the stakeholders are incorporated into the design process and thus provide the basis for evaluating the developed system. The requirements specified in this section have been obtained from the system analysis performed in Section 3.1, the interactions with traffic officers and urban planners and good practices in designing such systems.

3.3.1 Functional Requirements

The following functional requirements have been put forward for the designed detection and monitoring system:

- Image and Video Upload: This system should have a user interface that facilitates the process of uploading either an image file in formats such as JPEG and PNG or video files in formats like MP4 or AVI to be analyzed through the detection model.
- Vehicle Detection: On getting the uploaded image or video file, this system should automatically run the detection model called YOLO11s and automatically detect all the Keke and minibuses from the file.

- Visualization through Bounding Box: All detected vehicle instances should be annotated in the form of bounding boxes drawn around the objects detected along with their respective classes (Keke and bus) and detection confidence scores.
- Output of Detection Summary: This system should give an output detection summary including the number of detections per class, confidence scores and the spatial coordinates of the bounding boxes.
- Detection Log Management: The system shall maintain a log of all detection incidents, tracking metadata such as filename, upload date and time, detections per class, and path to the output file that has been generated, making it possible to retrieve these records for later reference.
- Downloadable Outputs: The system shall allow for downloading the annotated output image or video, which can be used for enforcement, reporting, and evidentiary purposes.
- User Authentication: The system shall have an authentication process through which only authorized individuals will be allowed to use the detection capabilities of the system.
- Detection History Access: Authorized users shall be able to access the history of previous detection events, optionally filtered by date.

3.3.2 Non-Functional Requirements

Non-functional requirements specify those quality criteria, performance parameters, and system behaviors which need to be satisfied, regardless of actual functional features. Below, the following non-functional requirements are determined for the developed system:

- Performance and inference speed: The system will be able to analyze the standard resolution picture (640×640 pixels) and provide the results of detection within the reasonable period of time, aimed at being shorter than five seconds when implemented on standard server hardware, to provide near-real-time analysis.
- Accuracy of detection: Trained detection model will have the mAP (mean average precision) metric equal or higher than 0.80 for IoU 0.5 on validation dataset.
- Usability: The interface of the web application is supposed to be user-friendly and require no additional technical knowledge from the user. All control elements will be properly labeled and feedback messages will be clear. The system should be usable by traffic officers with the basic computer literacy.
- Reliability and availability: The system should be designed for the uninterrupted work during regular business hours. It should have availability equal or higher than 95%. The system will be equipped with error handling mechanism that will help to manage invalid or corrupted input files.
- Security: Password hashing and session management will be used for the authentication of the users to secure the access to the system. Detection logs and media files that have been uploaded will be secured and can only be accessed by authorized users.
- Scalability: The system architecture should support the scaling of the user load and data storage, while maintaining good performance. The model inference system should be modular to make the system scalable in case of future replacement with another improved version of the model.

- Portability: The web deployment architecture should be designed in such a way that makes the system available on any device having an up-to-date web browser installed.
- Maintainability: The code should be developed using modules to ease the future maintainability of the system, updating the model and adding features to it.

3.4 System Design Methodology

The design approach defines the organizational structure that controls the way the system will be developed, from its analysis to testing and implementation. This choice of the proper design approach depends on the specifics of the project, the necessity of iterations, and the validation of each element of the system, especially the machine learning model.

3.4.1 Justification for Methodology

Agile Development approach was selected for implementing the system under consideration. It involves structuring the development process through a number of iterations, during which the system increment becomes available and can be analyzed, tested, and improved in preparation for the next iteration. The iteration includes all development activities – from requirements engineering to testing, performed on a certain part of the system functionality.

It seemed especially relevant to apply this approach to development of the proposed system due to several factors. Firstly, the machine learning part of the system requires a lot of iteration, since the performance of the algorithm (in our case – YOLO11s) needs to be checked after each training session, and accordingly, the parameters of training need to be tuned. In other words, the process of machine learning is an iteration process in itself. The linear approach, such as Waterfall, would be inappropriate in this case due to lack of flexibility.

Second, an aspect of the system that can be incrementally delivered through its web application part is the following. Key functions like uploading images and displaying detections can be implemented during initial iterations, and more advanced functions like detection history handling, user authentication, and report downloading will be added in further iterations. As a result, there will always be opportunities for user feedback and quality assessment during development, instead of waiting for it all to be done at once during acceptance testing. The iterative approach will therefore be in line with both research and engineering components of the project.

3.5 Method of Data Collection

The data collection process for the current research project included two main sources of data: data needed to train and validate the YOLO11s object detection model, and qualitative data that would be useful in formulating the system requirements.

For the model dataset, data collection was performed through direct observations and video/images taken of vehicles on Enugu metropolitan roads. Video and still image recording was done using a mobile phone equipped with a camera to capture images/footage of Keke (tricycles) and minibuses in various environments including running cars, parked cars on roadsides, intersection traffic jams, and motor parks. Video recordings were taken at various times during the day, such as day time and evening, to add variability in terms of light intensity. Video recordings were performed from different perspectives, such as roadside observation, moving car observation, and high places observation to make sure the dataset included the different perspectives from which the detection model would be expected to work in the actual situation. Video frames were then selected at certain intervals and manually annotated using Roboflow annotation tools where bounding boxes were drawn for each Keke and bus that could be seen. This dataset was called Keke-bus-1 and was saved in YOLO format in Roboflow.

The secondary data gathering, which was carried out in the course of determining system requirements and analyzing the existing system, was done via formal and informal interviews with the road traffic enforcement personnel who were working in the city of Enugu. This helped understand the actual conditions of the manual system, the difficulties encountered by the personnel on duty, and the functional requirements that the personnel would expect from the proposed system. The secondary data was also gathered from the related literature on traffic management in Nigeria.

3.5.1 Dataset Description and Preprocessing

The training and evaluation set for YOLO11s were based on a completely custom-built dataset. The reason for creating a customized dataset instead of modifying an existing one is the lack of an existing annotated dataset that includes the particular vehicles, road markings, building surroundings, and lighting features found in an urban street scenario in Nigeria. There is no existing vehicle dataset like those used for benchmarking purposes internationally that contains these particular vehicle types. The creation of a new dataset is therefore essential to the contribution made in this project.

Data acquisition was achieved by filming video and capturing pictures at various places in the Enugu metropolis using a mobile phone. Filming took place at different times of the day to allow for diversity in the lighting conditions, covering bright sunlight in the middle of the day, cloudy weather conditions, and evening lighting. Various recording points were used such as static points at the roadside, video footage taken from inside a moving car, and footage taken from elevated points, all aimed at achieving diversity in the perspective and scale at which the target cars would be filmed.

Image annotation was done through the use of the CVAT (Computer Vision Annotation Tool) software, which is an advanced level of annotation software that has been made available for use through open source and also supports the YOLO bounding box format. This involved the annotation of all images where Keke (tricycles) and minibuses were marked by bounding rectangles for each one, where the two classes (Keke, with class index 0 and bus with class index 1) were identified. The annotated images were exported to the YOLO format and uploaded to Roboflow under the Keke-bus-1 project.

As shown by the label distribution plot in Figure 4.1 below, the training split comprised roughly 175 Keke labels and 115 bus labels, representing a somewhat imbalanced class distribution towards the Keke class. This imbalance is reflective of the actual ratio of vehicles on the road in Enugu, since there are far more tricycles than minibuses on the roads in this region. As shown in Figure 4.2 below, the correlogram for the bounding box width-height pair indicates that there is a strong linear correlation between width and height within each class, implying that the annotations have similar aspect ratios and that the detections cluster around the bottom centre part of the image frame.

Data preprocessing for the training set consisted of downsizing images to 640 x 640 pixels in size using the letterbox technique that maintains the aspect ratio and pads images with grey spaces, not altering their content. The following data augmentations were performed using the configuration parameters from the Ultralytics training pipeline: horizontal flip (probability of 0.5), HSV colour jittering of independent channels (hue, saturation and value) of the image, random scale change (up to 50 percent) and random image translation (up to 10 percent). The mosaic data augmentation was switched off only for the last 10 training epochs, while for the rest of the time the data augmentation was performed at maximum intensity in order to train the model to learn on different pictures without any transformations. Also, the random erasing with a probability of 0.4 was applied in order to make the model more resilient to partial occlusions.

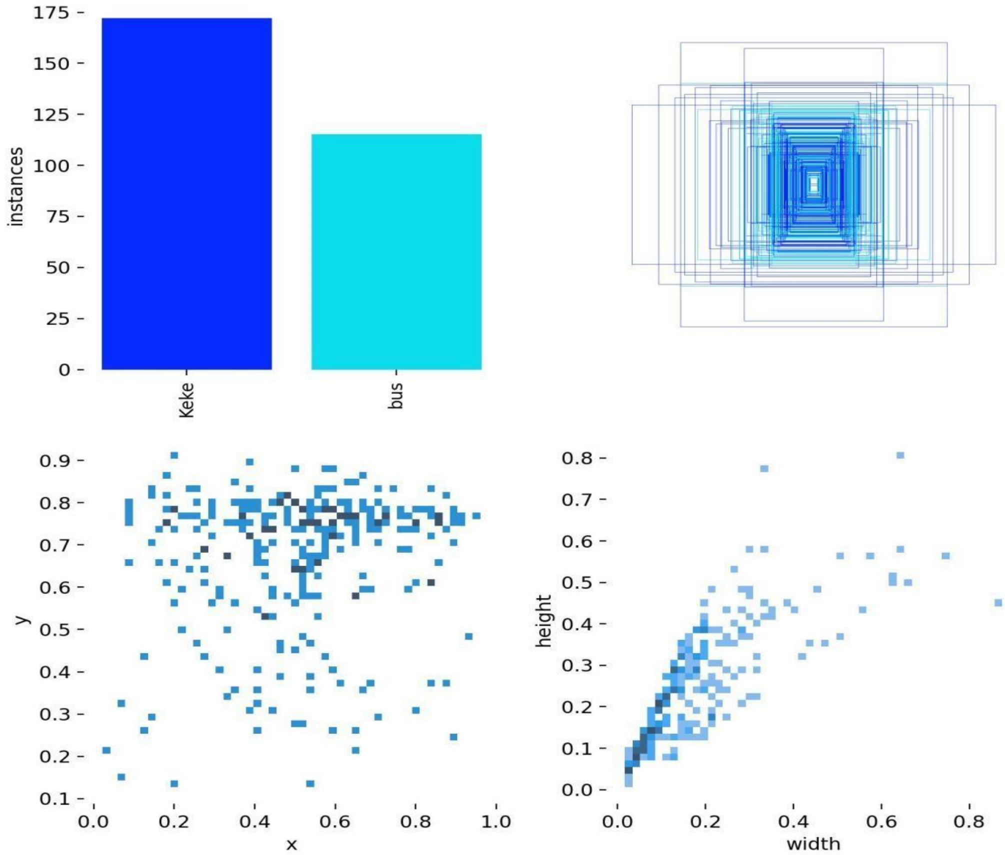

Figure 1: Dataset Label Distribution — Instance Counts per Class and Bounding Box Spatial Distribution

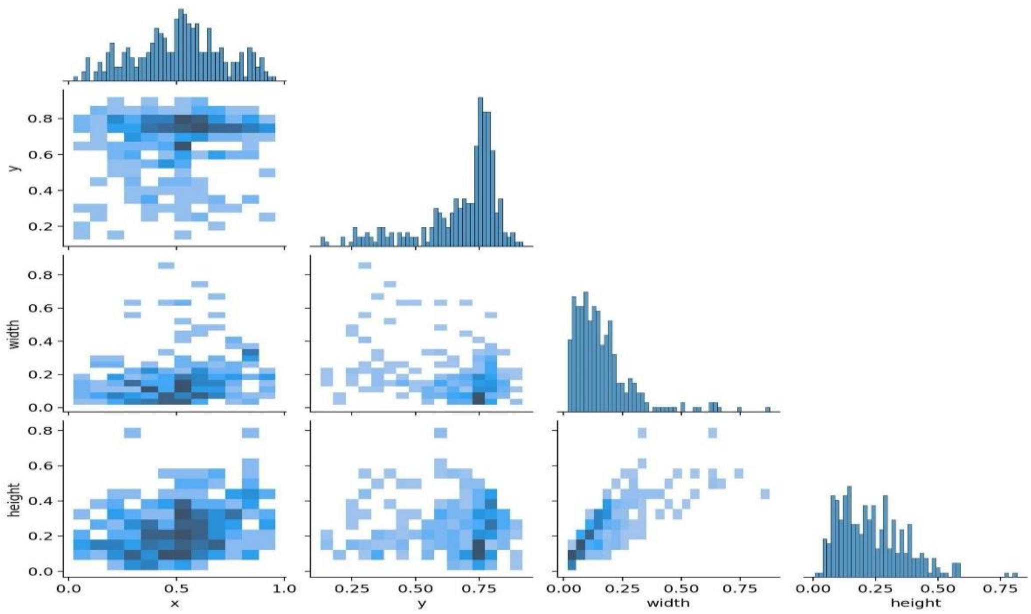

Figure 2: Label Correlogram — Pairwise Relationships Between Bounding Box Spatial and Dimensional Attributes

3.6 System Modeling Using UML

Unified Modelling Language (UML) refers to a standardized modelling language that is commonly used in software engineering to describe, visualize, and document the artifacts of a software system. In this part, three main UML diagrams will be discussed that are relevant to the design of the proposed system, which include use case diagram, activity diagram, and class diagram. The three diagrams are meant to give the behavior, processes, and organization of the system, which form the basis of implementation as discussed in chapter four.

3.6.1 Use Case Diagram

Use case diagram shows the functional interactions that take place between the system actors and the functionality of the system. In the proposed detection and monitoring system, two system actors can be distinguished: the Traffic Enforcement Officer (main system actor) and the System Administrator.

The Traffic Enforcement Officer is the main user of the system. This system actor uses the system through the interaction with it using the web application to log in to the system, upload the image or video file for analysis, start detection process, view annotated output with detection summary, download output for further documentation, and check detection history log. All these interactions form the main use cases of the system.

The System Administrator is the actor responsible for the maintenance of the platform. His/her use cases include user accounts management, monitoring of the health and performance of the system, checking comprehensive detection logs, and updating/replicating the detection model when the new version becomes available.

The use case diagram for the proposed system is illustrated in Figure 3.1. The system boundary represents all system use cases, whereas the two actors are placed outside the system boundary and connected to their use cases via associations. The use case “Login” is connected to both actors. The use cases “Upload Media”, “Run Detection”, “View Detection Output”, “Download Results” and “View Detection History” are connected to the actor
Traffic Enforcement Officer. The use cases “Manage Users”, “View All Detection Logs” and

“Update Detection Model” are connected to the actor System Administrator. The use case “Run Detection” inherits from the “Upload Media” use case because detection is conditionally invoked after the file upload and validation.

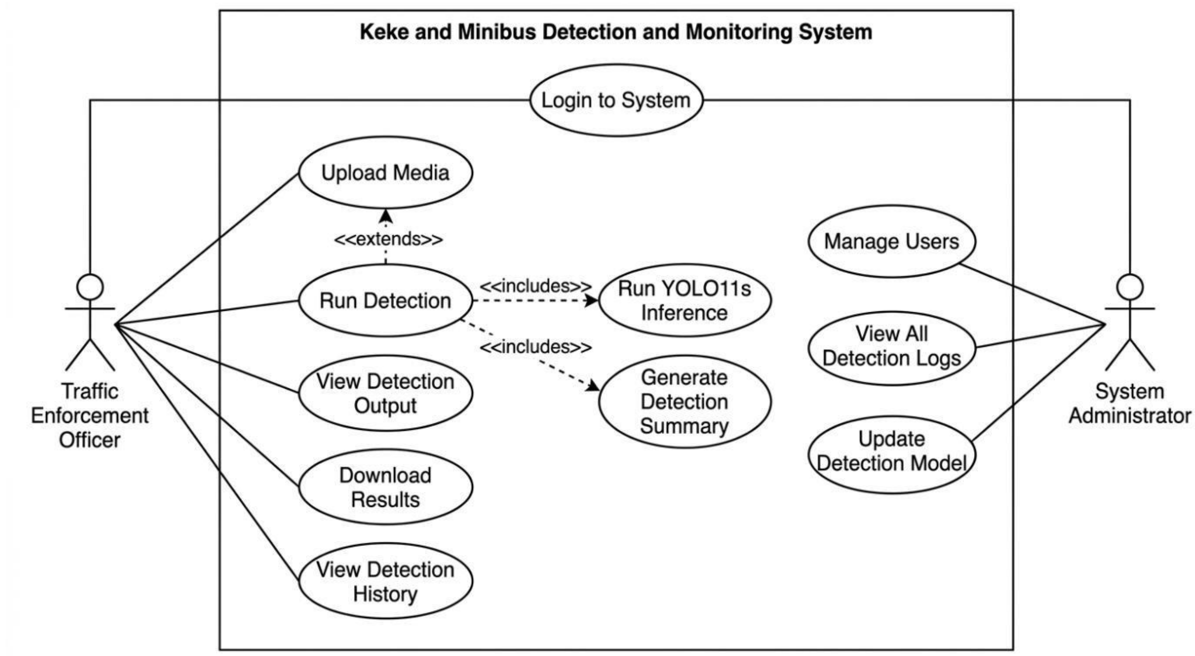

Figure 3: Use Case Diagram of the Proposed Keke and Minibus Detection System

3.6.2 Activity Diagram

The activity diagram illustrates the dynamic flow of events within the system through modeling of the sequence of events and decisions that take place in the major process of operations which are the uploading of images or videos and detection of vehicles. Figure 3.2 below is the activity diagram illustrating the flow of events within the system from the point of view of the Traffic Enforcement Officer using the web application.

The activity flow starts when the traffic enforcement officer accesses the web application and gets to the login screen. Once the login details have been entered and verified, the system automatically redirects to the main dashboard where the officer picks the option of uploading media file and selects the file. Before the file is processed by the system, the system will validate the file to check whether the file is in the correct format either JPEG, PNG, MP4 or AVI as well as whether it exceeds the maximum allowed file size.

The detection engine runs the uploaded image or the individual frames of the uploaded video through the trained model to detect and localize all occurrences of Keke and bus class objects. Bounding boxes labeled with their class names and confidence score are added to the processed data. Once the inference is completed, the detection data, which includes the number of classes, confidence score, bounding box coordinates, and time stamp, is saved to the detection log database. Finally, the annotated data is displayed on the result page of the web application for inspection by the officer, and he may download the annotated output file if necessary.

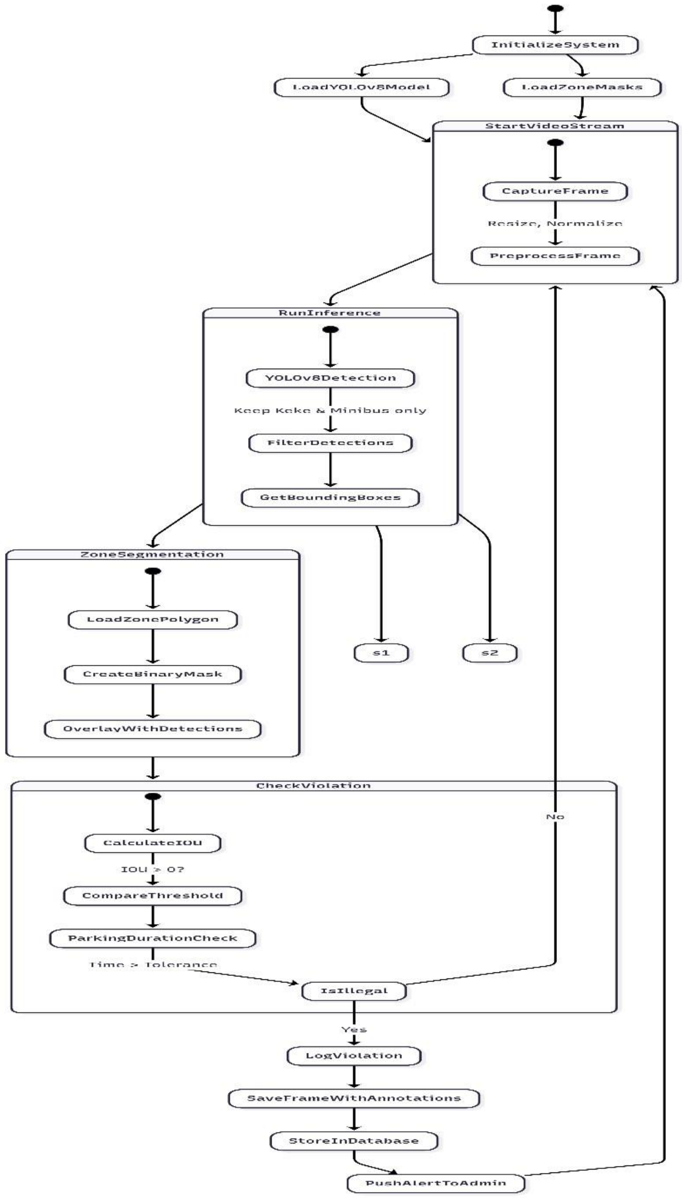

Figure 4: Activity Diagram for the Image Upload and Vehicle Detection Process

3.6.3 Class Diagram

Class diagram is a static model of system structure showing all important classes and their associations. Figure 3.3 depicts class diagram of the suggested system which involves the following key classes.

User class encapsulates all registered users of the system and includes the following attributes: userID (integer, primary key), username (string), email (string), passwordHash (string), role (enumeration: officer/admin), and createdAt (datetime). Methods of the class include login(), logout(), and updateProfile().

UploadedFile class encapsulates a media file uploaded by a user to be analyzed. The attributes of the class include fileID (integer, primary key), userID (integer, foreign key referring to User class), fileName (string), fileType (enumeration: image/video), filePath (string), fileSize (float), and uploadedAt (datetime). Methods of the class include validateFile() and deleteFile(). UploadedFile class has a many-to-one association with User class because one user can upload several files.

Detection class represents a detection generated by the YOLO11s inference engine on processing an uploaded file. It has the following attributes: detectionID (int, primary key), fileID (int, foreign key to UploadedFile), totalKeke (int), totalBus (int), processingTime (float), outputFilePath (string), and detectedAt (datetime). It has two methods called runInference() and generateSummary(). There is a one-to-one relationship between the Detection and the UploadedFile classes.

BoundingBox class represents a single detected object from a Detection, containing the details of the bounding box. It has the following attributes: boxID (int, primary key), detectionID (int, foreign key to Detection), classLabel (string), confidenceScore (float), xMin (float), yMin (float), xMax (float), and yMax (float). There is a many-to-one relationship between BoundingBox and Detection classes as a single detection may result in generating multiple bounding boxes.

DetectionLog class acts as an audit trail of all the detection operations done in the system. It has the following attributes: logID (int, primary key), detectionID (int, foreign key to Detection), userID (int, foreign key to User), actionType (string), and timestamp (datetime).

There is a many-to-many relationship between the Detection and the User classes through the DetectionLog table.

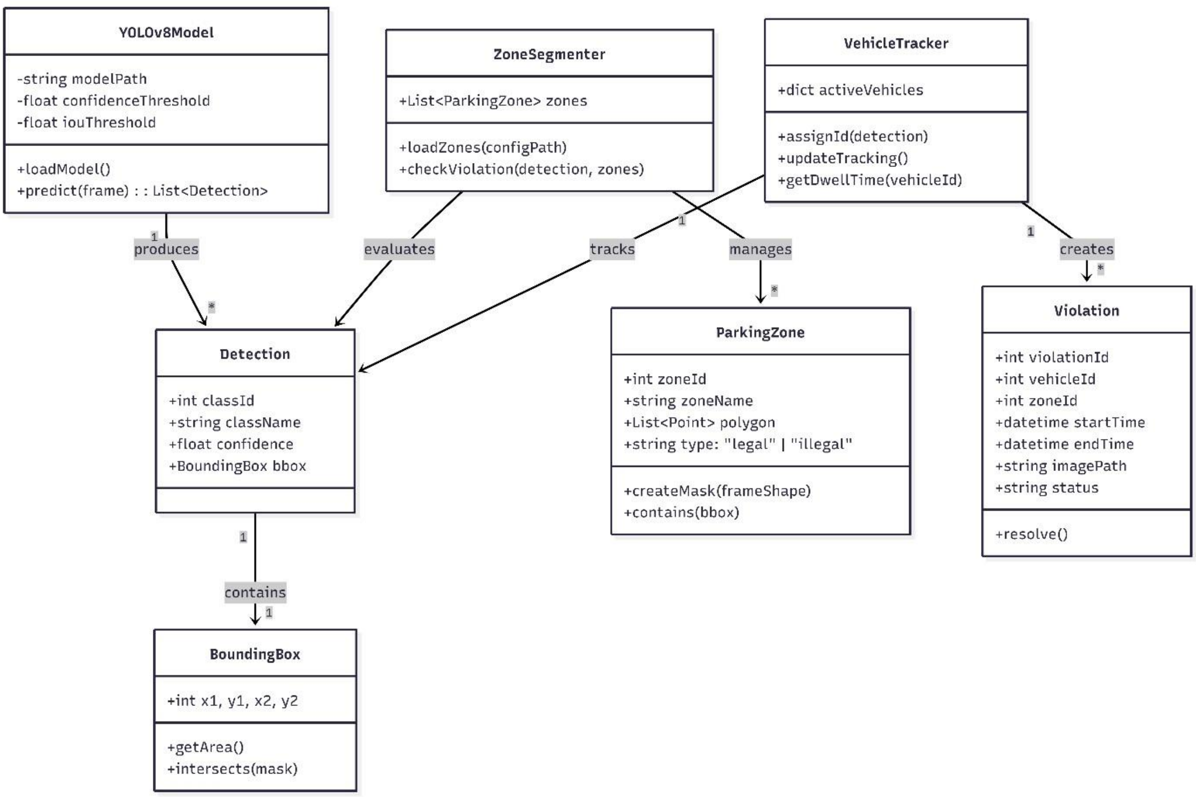

Figure 5: Class Diagram of the Proposed Detection and Monitoring System

3.7 System Design

The System Design process entails the transformation of the requirements and models created in the previous chapters into detailed specifications of the input, output, database, and architecture of the system under consideration. The design process outlined in this chapter informed the implementation process of the system as seen in Chapter Four.

3.7.1 Input Design

Input design involves defining the way the data will enter into the system, making sure that all input methods are intuitive, validated, and appropriate for the needs of the targeted users.

The input for the user authentication involves two fields within the login form: a field for the username or the email and the field for the password. Both fields are mandatory. The system validates that none of the fields is left blank and that the input credentials correspond to any record in the database. The passwords are hashed during the validation process, which guarantees that no plain text credentials are saved. The error messages that indicate a failure during the log-in procedure do not mention which field failed.

Input for the media upload is obtained from the user via a file upload form displayed on the main detection web page after the process of authentication. The acceptable file formats in this case include JPEG, JPG, PNG, MP4, and AVI. An initial client-side validation takes place before the submission of the file to the server to weed out incompatible file formats and files that exceed a certain size limit. After the completion of client-side validation, the file is sent to the server where a subsequent server-side validation occurs to determine the file format by inspecting its MIME type as well as checking whether the file is corrupt and empty.

3.7.2 Output Design

Output design describes how the output generated through the system's processing will be provided to the user such that the information related to the detection activity is conveyed to the user in an appropriate way, which means it should be accurate and easy to understand.

The major output from the system will be in the form of annotated detection images or videos. When an image input is supplied to the system, then it will produce the annotated version of the input image wherein the bounding boxes will be drawn around the detected objects, such as Keke and bus, and each object will be assigned a unique colour depending on the class. The class name and confidence score will also be mentioned as a text above the bounding box. The same process will be applied when a video is input.

Besides the annotated detection result, the system generates a detection summary which is included on the results page in tabular form. The details provided in the detection summary include the number of Keke vehicles detected, the number of bus vehicles detected, the date and time of detection, and the name of the uploaded file. A list of individual detections will also be provided showing the class label, confidence score, and bounding box of the detection. This information helps in quality assurance purposes and allows the user to pick out detections that might need manual verification.

The detection history section also provides an overview of all previous detection events for the authenticated user. It takes the form of a paginated table showing the filename, upload date, number of detections by classes, and a link to re-visit the detection result.

3.7.3 Database Design

The database design is responsible for defining how the persistent data will be stored within the database. This project uses a relational database management system (RDBMS), which is currently being developed using SQLite; however, there are plans to move it onto a more advanced database like PostgreSQL or MySQL. There are four main tables within the database.

Table 2: Users Table

Field Name	Data Type	Key	Nullable	Description
userID	INTEGER	PRIMARY KEY	NOT NULL	Unique user identifier
username	VARCHAR(100)	UNIQUE	NOT NULL	Unique username
email	VARCHAR(200)	UNIQUE	NOT NULL	User email address
passwordHash	VARCHAR(255)	-	NOT NULL	Hashed password
role	VARCHAR(20)	-	NOT NULL	officer or admin
createdAt	DATETIME	-	NOT NULL	Account creation timestamp

Table 3: UploadedFiles Table

Field Name	Data Type	Key	Nullable	Description
fileID	INTEGER	PRIMARY KEY	NOT NULL	Unique file identifier

userID	INTEGER	FOREIGN KEY	NOT NULL	References Users.userID
fileName	VARCHAR(255)	-	NOT NULL	Original file name
fileType	VARCHAR(10)	-	NOT NULL	image or video
filePath	VARCHAR(500)	-	NOT NULL	Server storage path
fileSize	FLOAT	-	NOT NULL	File size in MB
uploadedAt	DATETIME	-	NOT NULL	Upload timestamp

Table 4: Detections Table

Field Name	**Data Type**	**Key**	**Nullable**	**Description**
detectionID	INTEGER	PRIMARY KEY	NOT NULL	Unique detection ID
fileID	INTEGER	FOREIGN KEY	NOT NULL	References UploadedFiles
totalKeke	INTEGER	-	NOT NULL	Count of Keke detected

totalBus	INTEGER	-	NOT NULL	Count of buses detected
processingTime	FLOAT	-	NULL	Inference time in seconds
outputFilePath	VARCHAR(500)	-	NULL	Path to annotated output
detectedAt	DATETIME	-	NOT NULL	Detection timestamp

Table 5: BoundingBoxes Table

Field Name	**Data Type**	**Key**	**Nullable**	**Description**
boxID	INTEGER	PRIMARY KEY	NOT NULL	Unique bounding box ID
detectionID	INTEGER	FOREIGN KEY	NOT NULL	References Detections
classLabel	VARCHAR(20)	-	NOT NULL	Keke or bus
confidenceScore	FLOAT	-	NOT NULL	Model confidence (0-1)
xMin	FLOAT	-	NOT NULL	Left edge of bounding box

yMin	FLOAT	-	NOT NULL	Top edge of bounding box
xMax	FLOAT	-	NOT NULL	Right edge of bounding box
yMax	FLOAT	-	NOT NULL	Bottom edge of bounding box

Relational integrity within the database is enforced through foreign keys between the Users, UploadedFiles, Detections, and BoundingBoxes tables. A cascading delete has been implemented to ensure that when a file entry is deleted, all related detection entries along with all related bounding boxes will be deleted as well, to maintain referential integrity. Indexes have been used on the userID field in the UploadedFiles table and the fileID field in the Detections table to enhance the efficiency of queries used for frequent tasks such as viewing history and generating reports.

3.7.4 System Architecture Design

The design architecture of the proposed system has been developed according to the ThreeTier Architecture, which means that the application has been divided into three different tiers logically. These include the Presentation Tier, Application Logic Tier and Data Tier. The Three-Tier Architecture makes the system more modular, scalable and manageable and is a common choice when developing applications through web based interfaces. Figure 3.4 describes the architecture of the system.

The Presentation Tier forms the front-end tier of the system and deals with all user interaction. This has been implemented using a web application which runs in the user's browser and includes HTML5, CSS3 and JavaScript. This tier consists of the login interface, file uploading interface, detection results page displaying output annotations and detection summary, and the detection history interface. The communication between the presentation tier and the application logic tier occurs via HTTP requests and the file uploads are made using multipart form data and detection results in the form of JSON responses displayed on the results page.

The Application Logic Layer is the main processing layer where all the business logic is contained. The layer uses the Python Flask web framework to do routing, session management, user authentication, and file management. In this layer, the media files uploaded in the presentation layer are received, processed for server-side validation, called upon the YOLO11s inference engine using the Ultralytics Python framework, and coordinated for storing the outputs in the data tier. Model weights of the YOLO11s are loaded into the memory during startup in order to reduce inference latency per request. This layer exposes RESTful API endpoints that can be used by future clients if the system is extended outside the web-based interface.

The Data tier deals with data persistence and access. This tier comprises a relational database (SQLite in the development environment) and the file storage folder where the uploaded media files and annotated output files are stored. Communication between the Flask app and the database takes place through SQLAlchemy Object Relational Mapping (ORM), which allows abstracting from SQL communication and enables the migration to a full-fledged RDBMS at some point in the future without any changes to the application itself. File paths in the database correspond to the files of physical media that are stored on the file system of the server.

The three-tier approach allows separating concerns effectively: alterations in the user interface layer do not necessitate any changes in the detection logic, and the detection engine can be updated independently from the database structure. Such separation of layers was implemented on purpose in view of further development of the system that includes real-time processing of a CCTV stream, geolocation of the violations and multi-user administrative dashboards for the city-scale implementation.

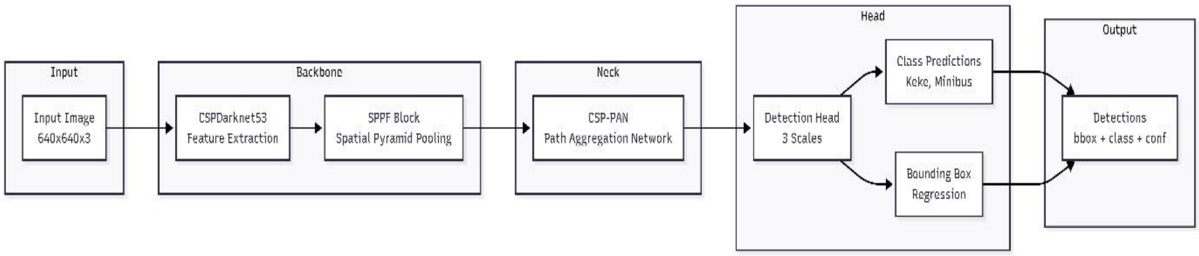

Figure 6: Three-Tier System Architecture of the Proposed Detection and Monitoring System

CHAPTER FOUR

IMPLEMENTATION, TESTING, AND RESULTS

4.0 Introduction

This chapter presents a comprehensive account of the practical implementation, testing, and evaluation of the computer-vision-based detection and monitoring system for Keke (tricycles) and minibuses in Enugu metropolis. The chapter commences with a description of the development environment, the programming language, libraries, and hardware infrastructure employed during both the model training and application development phases. It proceeds to document the composition and preprocessing of the custom dataset used to train the YOLO11s object detection model, followed by a detailed account of the model's architecture and training configuration. The chapter then describes the system's modular implementation, encompassing the homepage, login, and dashboard components of the PyQt6-based desktop application. Evaluation methodology is presented, including the metrics used to assess model performance, after which the results of system testing and a walkthrough of the user interface are provided. At the end of the chapter, there is an analysis of the results of detecting the model, aided by the evaluation plots, confusion matrix, and the annotated inference results of training and validation.

The system designed in the current project consists of the combination of a trained YOLO11s neural network, a polygon-based spatial enforcement algorithm, and a PyQt6 desktop application to provide an end-to-end solution for automated vehicle detection. Once implemented, the system takes image input from the video feed of the camera facing the specific area allocated for parking, analyzes the video using YOLO11s to detect all the Keke and bus class cars in the picture, and decides whether the vehicle falls into the previously established legal polygon for parking. Any car not falling into the polygon range is considered to be violating the rules and is logged into the system's JSON administration database.

4.1 Development Environment and Tools

The construction of the proposed system was carried out using two separate computational systems: a cloud-based GPU system for training models and a personal computer for developing the application. This approach was chosen on purpose because of the computational complexity of the deep learning training process, which demands hardware acceleration beyond what is available on a standard development machine.

4.1.1 Programming Language and Libraries

The programming language that was chosen to develop the entire project is Python 3.10 which involves dataset processing, training the models, integration of the inference engine, application logic, and the UI. The wide range of scientific computing libraries and the machine learning libraries, along with Python’s easy-to-read code and community support, makes it an ideal language for computer vision tasks.

The main libraries and frameworks that were used to develop the system include the following. Ultralytics library version 8.x has been used as the framework in implementing the YOLO11s object detection model, which provides a higher-level API for training, validating, testing, and exporting the model. In addition, Ultralytics framework makes use of PyTorch deep learning framework but makes the whole process easier by providing an abstraction of the complexity and making the training process reproducible using a simple configuration process. OpenCV library (Open Source Computer Vision Library) has been used to read images and video frames, convert color spaces, and draw geometric overlays including the legal parking zone polygon. PyQt6 was the graphic user interface module that provided the desktop application layer through which the system was accessed by the administrator. PyQt6 is the Python binding for Qt6 and was selected due to the wide array of widgets that it supports as well as its signal-slot event mechanism and support for embedding OpenCV videos into the native desktop window. The supervision library was utilized in the process of annotation and post-processing stages to provide structured manipulation of the bounding boxes and rendering of the detection in a standardized format that is suitable to the needs of the system. CVZone2 was utilized alongside OpenCV for additional utility functions such as text rendering.

The dependencies for the Python implementation were handled using a requirements file to ensure repeatability of the development environment setup. For the lightweight database used to store the violation logs, the JSON library that comes with the standard Python library was used.

4.1.2 Development Platform and Hardware Requirements

Training of the model was carried out using Google Colaboratory (Google Colab), a cloudbased Jupyter notebook that offers on-demand use of computing power with GPU acceleration. In our case, the training employed a Tesla T4 GPU with 15 GB of GPU RAM, which offered hardware acceleration required to perform 50 epochs of training in a reasonable amount of time. Training of the model was performed in Google Colab as the
Keke-bus-1 dataset was stored in the Roboflow platform, from which it was used directly in Google Colab through Roboflow API. Model configuration, as described in the args.yaml file, had a batch size of 16, image resolution of 640x640, and YOLO11s pretrained weights were used as initialisation checkpoints.

Application development, which included development of the PyQt6 GUI, the enforcement algorithm for the polygons, and the JSON logger module, was done on a local machine with Visual Studio Code as the IDE. Python 3.10 was installed via Anaconda, and all necessary libraries were installed into an isolated environment in order to avoid any potential library conflicts. Application testing was done on the local machine with still images and video recordings of the roads in Enugu as inputs, while the trained YOLO11s model weights were loaded from a local path.

4.2 Model Architecture and Training

The object detection algorithm used as a backbone of the proposed approach is the YOLO11s (You Only Look Once, version 11, small version) architecture, which represents the latest development within the YOLO series of architectures implemented by Ultralytics. The YOLO11s is classified as a one-stage detector since it conducts object classification and bounding box localization in one inference step compared to the two-stage detectors, like Faster R-CNN, that use separate stages for generating region proposals. Thus, the YOLO11s architecture can be considered the best fit for the proposed application due to its real-time inference capabilities.

YOLO11s is an implementation of a network comprising the backbone for extracting visual features based on CSPDarknet, the neck based on the PANet for fusing multiscale features and decoupled detection head for independently predicting box coordinates and class scores at three different scales of detection. The network is initialized using the pretrained weights which are the result of training the network on the COCO dataset (yolo11s.pt) and then adapting the detection head to the two classes (Keke and bus) of the current problem using transfer learning. Pretrained weights were used to ensure the network makes use of the highlevel visual features acquired from training on a large dataset, hence achieving faster convergence.

Training was carried out using the Ultralytics training API with the setup saved in the args.yaml file. Hyperparameters used included 50 training epochs, a batch size of 16 images, an input size of 640x640 pixels, and the auto optimizer choice method, which resulted in using Stochastic Gradient Descent (SGD) with the momentum of 0.937 and weight decay of 0.0005. Learning rate was 0.01 initially but was planned to decay to 0.01 using the cosine annealing-style learning rate scheduling throughout the training process. Warmup strategy included 3 epochs during which the learning rate was gradually increased from the bias of 0.1 to its initial value. For NMS during validation, the IoU threshold was 0.7 and up to 300 detections per image were allowed. The three parts of the loss function included: CIoU (bounding box regression loss) with the weight of 7.5, classification loss with the weight of
0.5, and DFL (distribution focal loss) with the weight of 1.5.

However, the training loss curves, obtained from the results.csv log file, show steady and reliable convergence during the 50-epoch training session. The training box loss is seen to have fallen from an initial 1.119 at epoch 1 to 0.571 at epoch 50, whereas the training classification loss is seen to fall sharply from 5.775 to 0.427 during the same time interval, showing that the learning process became increasingly precise in distinguishing between the Keke and bus objects. The training distribution focal loss also fell from 1.165 to 0.909. On the other hand, the validation box loss is seen to decrease from 1.078 at epoch 1 to 1.008 at epoch 50, whereas the validation classification loss falls significantly from 5.500 to 0.685, thus confirming that there was no overfitting during the training and that the generalisation was true.

The detection accuracy of the model on the validation data set gradually increased during the training process. From epoch 13, an increase in mAP@0.5 accuracy was noticeable and attained a value of 0.646. The performance of the model improved all through to the middle epochs, where mAP@0.5 value was 0.800 for epoch 31, finally stabilizing at 0.838 for epoch 50. The final mAP@0.5:0.95, which measures the accuracy of average precision between a threshold range of 0.50 to 0.95, and thus measures localization accuracy more accurately, had its final value of 0.607 at epoch 50. This shows that the detection accuracy of the model attained a satisfactory level for the proposed system application.

4.3 System Implementation and Modules

The entire system is developed as a desktop application developed using PyQt6, which includes the usage of the YOLO11s trained model for making predictions, OpenCV for handling the camera frames and polygon drawing, and a database using JSON files for storing violation logs. This system is organized into three major modules; these include the Homepage Module, the Login Module, and the Dashboard Module. Each of these modules consists of one screen within the application and represents a particular responsibility within the system process.

The Homepage Module makes up the initial page for the application to start from. Once the application starts, the user will be directed to a homepage which explains what the application is all about, the intended use of the application as being a monitoring system for Keke and minibus, and a call-to-action button that redirects the user to the login page. The homepage is designed in such a way that it is simple to avoid confusion of the user and to create an administrative setting.

The Login Module collects and authenticates the entered credentials of the administrator, which are provided via the Username and Password input fields. Upon entry and submission, these credentials are checked against the administrator credentials saved in the JSON configuration file. If the credentials authenticate successfully, the application proceeds to use the Dashboard Module. Otherwise, an error message is shown, and the login screen is refreshed. All passwords are stored in their hashed forms in the JSON configuration file in order to protect any plain text credentials from being leaked if somebody gains access to the database file.

The Dashboard Module represents the working component of the system and consists of the complete workflow of vehicle detection and violations tracking. At the start of the dashboard, the administrator is shown either a live or a static video stream in the form of frames obtained from the camera feed displayed via the PyQt6 widget with the help of OpenCV frame capturing. The dashboard offers the administrator an interactive polygon zone configuration tool for drawing the authorized parking zone by clicking the frame to create vertices of the polygon. The vertices are joined sequentially to form a closed polygon indicating the authorized parking zone on the frame.

The inference engine produces bounding boxes, class labels, and confidence levels for all the detected Keke and buses in each frame. The Dashboard Module uses the zone intersection algorithm to determine if the centroid or lower end of each bounding box of the detected vehicles is located inside the predefined legal polygon. The polygon check is done using the pointPolygonTest method provided by OpenCV library. The pointPolygonTest returns a positive value if the test point is inside the polygon and negative otherwise. Vehicles whose test point is outside the polygon are considered to have committed a violation. In case of violation events, the system saves the current annotated frame with class label, confidence level, violation notice message displayed using CVZone2 library, and the path to the saved frame in the output folder. Log information on the violation events such as timestamp, class of the offending vehicle, confidence level, bounding box coordinates, and path to the saved violation frame are simultaneously written into the JSON administrative database.

4.4 Testing and Evaluation

Testing and evaluation of the system were performed to evaluate the performance of the trained YOLO11s model according to the benchmarks of object detection, to check the correct functioning of the components of the system, and to test the usability of the PyQt6based interface. Two types of testing have been performed to evaluate the system – quantitative model evaluation according to the metrics provided by the Ultralytics validation pipeline, and functional system testing using test cases for the desktop application.

4.4.1 Evaluation Metrics

The metrics for evaluating the detection model include Precision, Recall, F1-Score, Average Precision (AP), and Mean Average Precision (mAP) at intersection over union (IoU) values of 0.5 and 0.5:0.95. This is common in object detection studies, and they offer a full characterization of the detection model's performance on the parameters of confidence, localization, and classification.

Precision determines the fraction of positive detections that are true positives and is indicative of the accuracy of the model to predict only those detections that are accurate. The Precision is calculated by taking the ratio of True Positive (TP) divided by the total number of True Positives and False Positives (TP+FP). Recall determines the fraction of actual positives that are accurately predicted by the model and represents the sensitivity of the model. The Recall score is determined by calculating the ratio of True Positives and the total number of

True Positives and False Negatives (TP+FN). The F1-Score is the Harmonic Mean of Precision and Recall, and provides an excellent insight into the performance of the model in case the two scores vary significantly. The F1-Confidence Score Plot is plotted by varying the confidence threshold.

For a particular class, the average precision (AP) is calculated as the area under the curve formed between precision and recall, integrating the precision across all recall values. The mean average precision (mAP@0.5) calculates the average of APs for all the classes calculated using an intersection over union (IoU) threshold of 0.5, where a detected bounding box needs to have at least 50 percent overlap with the true bounding box to be regarded as true positive. Similarly, mAP@0.5:0.95 takes the average of AP at ten different IoU thresholds between 0.50 and 0.95 in increments of 0.05.

Confusion Matrix offers a breakdown of the decisions made by the model in class by class manner where true positives, false positives, false negatives and the background predictions for each class are provided. The raw confusion matrix as well as the normalized confusion matrix, where each row is normalized with respect to the total number of ground truth per class, are presented.

4.4.2 System Testing

System testing was carried out to determine whether individual components of the desktop application were functioning correctly when executed separately and in conjunction with one another. The testing was done through a systematic process in which specified test data was used to test individual components and system output was then checked against expected results based on the system requirements.

Homepage module was checked through the process of running the program and checking that the homepage appeared properly, all interface elements such as title, description, and call-to-action button appeared properly formatted, and the click on the call-to-action button navigated the program to the login page.

Login module was checked through three tests. The first one implied entering valid credentials of administrator and the system should have switched to the dashboard page. The second one involved entering an invalid password together with the valid name and getting the message about authentication error appearing and staying on the same page. The third test entailed leaving the fields blank and triggering the submit button and ensuring that there was validation of blank input and proper notification without making any database requests. All three tests were successful.

The Dashboard module was evaluated with a set of recorded static image data and video frame sequence from Enugu. It was established that the camera feed is properly displayed in the PyQt6 widget, that the Polygon Zone interface properly captures the clicks on the vertices and renders the polygon overlay on the frame, that the YOLO11s inference model properly detects Keke and bus objects in the test frames with the proper confidence score label, that

the zone intersection algorithm properly categorizes the detected objects as being either in or out of the polygon, and that the violation events led to proper logging and saving of the frames. The violation records in the JSON log file were checked for completeness after each test run.

4.4.3 User Interface Testing and Walkthrough

Tests were carried out on the user interface of the PyQt6 desktop application to assess its navigability, visual appeal, and functionality from the point of view of the administrative enduser. The walkthrough begins right from the point when the application is opened up until the review of the violations log page.

When the application is started, the first screen that appears before the user is the Homepage. This page gives information about the name of the system and the purpose of the system which is for Keke and minibus monitoring and enforcement. There is a very conspicuous 'call-to-action' button titled 'Proceed to Login' on the homepage. There is not much of information on this homepage.

In the Login screen, the admin user inputs their username and password in the available input box. The password input hides characters for security reasons. By pressing the 'Login' button, the system will authenticate the credentials based on the admin record that exists in the system. Authentication success leads to transition of the user to the Dashboard screen, an animated process giving feedback to show loading of the new screen.

In the Dashboard screen, the administrator sees the live video feed from the camera feed showing in the main pane of the screen. There is a side panel that gives control of zone configuration and detection management. In order to set up the authorized zone, the administrator presses the 'Set Parking Zone' button that initiates the polygon drawing mode. The administrator proceeds by placing the vertices of the polygon in the corners of the authorized zone of the camera feed frame. For each vertex, the system connects the vertices using lines while displaying the indicator of each vertex. After placing the last vertex and closing the polygon, there will be a colour overlay of the authorized zone on the camera frame.

After setting the zone, the administrator presses the 'Start Detection' button. Inference engine YOLO11s starts working and starts processing the frames. The Keke and bus instances that are detected are marked using bounding boxes and confidence scores labels drawn by CVZone2. Vehicles detected within the legal polygon get a green bounding box around them, whereas vehicles that are detected beyond the boundaries of the polygon get a red bounding box around them along with 'VIOLATION' text written on them. Every time there is a violation, there will be an entry made in the admin log panel visible on the side. The log panel will scroll down by itself whenever there is a new entry of violation. The administrator can also click on any log entry to see the frame of the violation in the popup preview window. The 'Stop Detection' button stops the inference engine, while 'Export Log' button exports the session's violation records into the JSON database file.

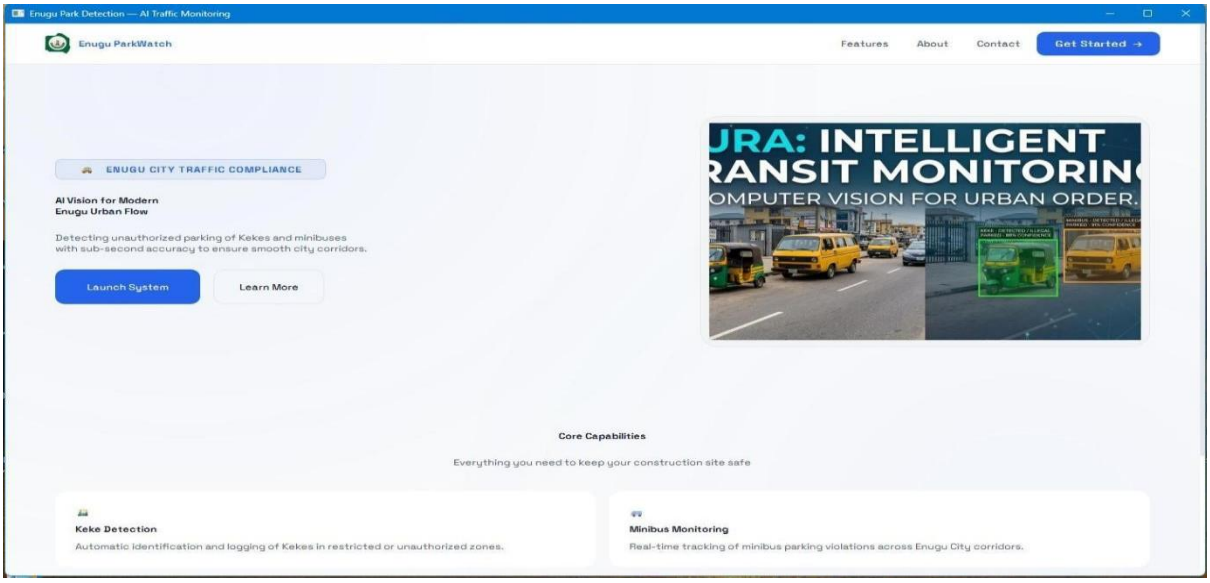

Figure 7: Homepage Screen of the Keke and Minibus Monitoring System

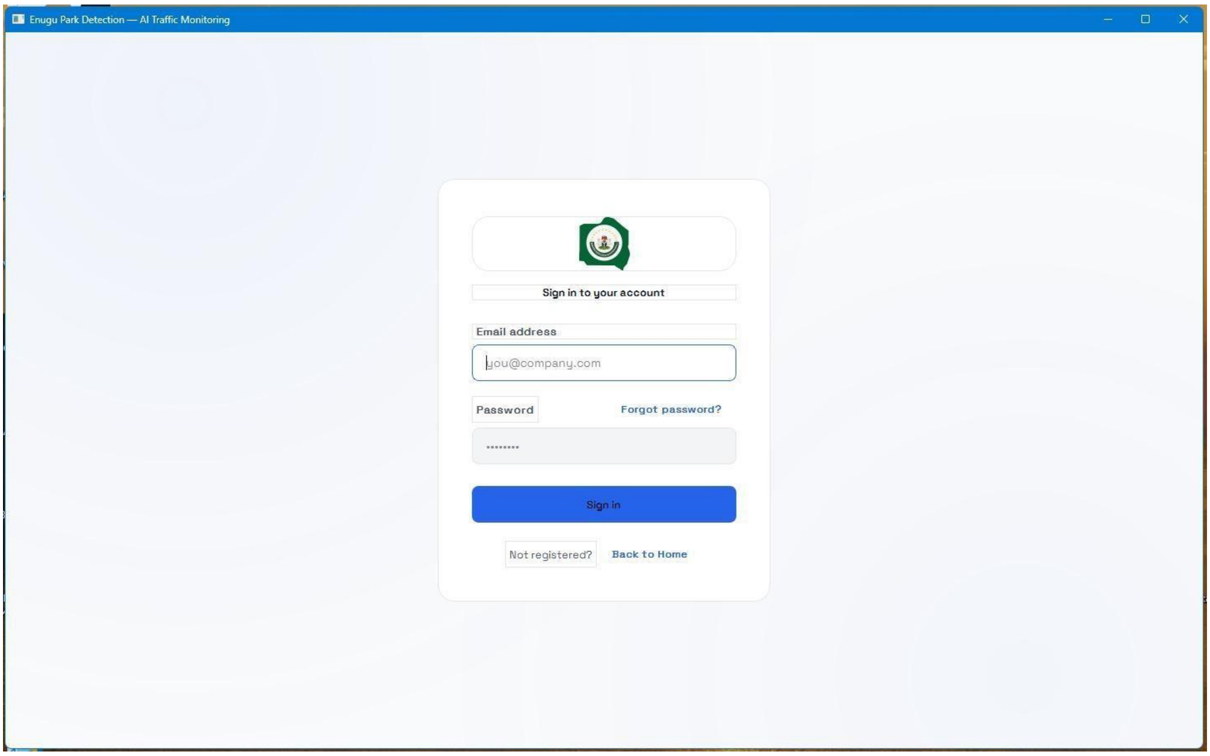

Figure 8: Login Screen Showing Administrator Credential Entry Interface

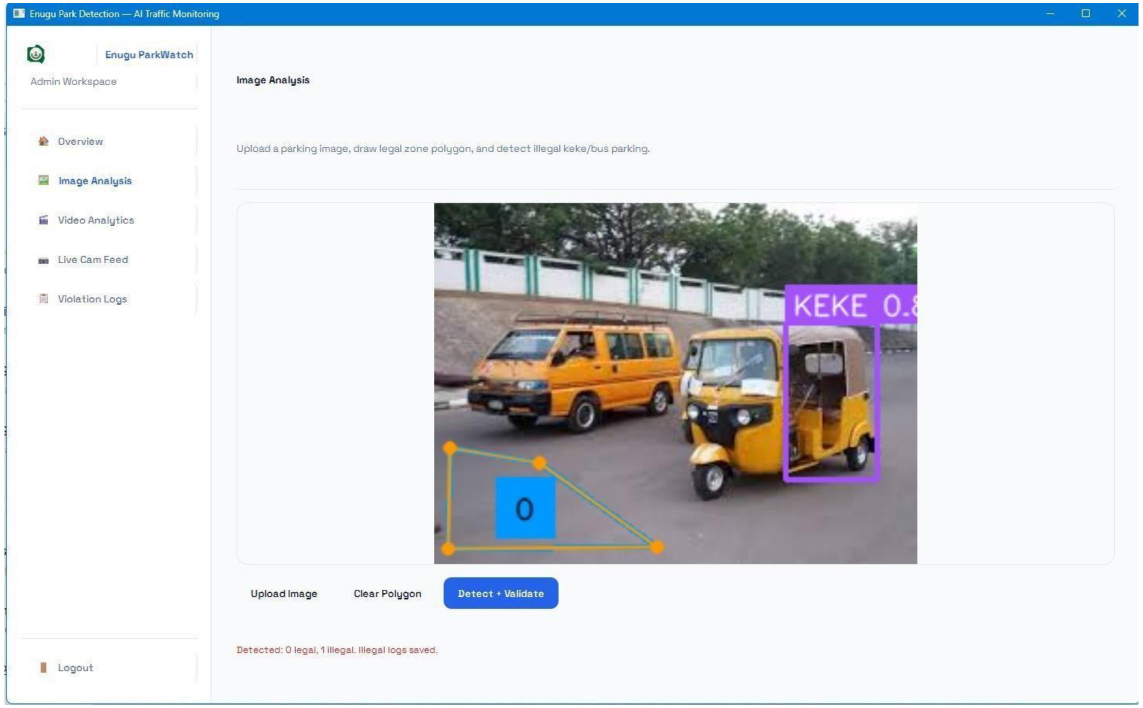

Figure 9: Dashboard Screen Showing Parking Zone Configuration and Active Detection

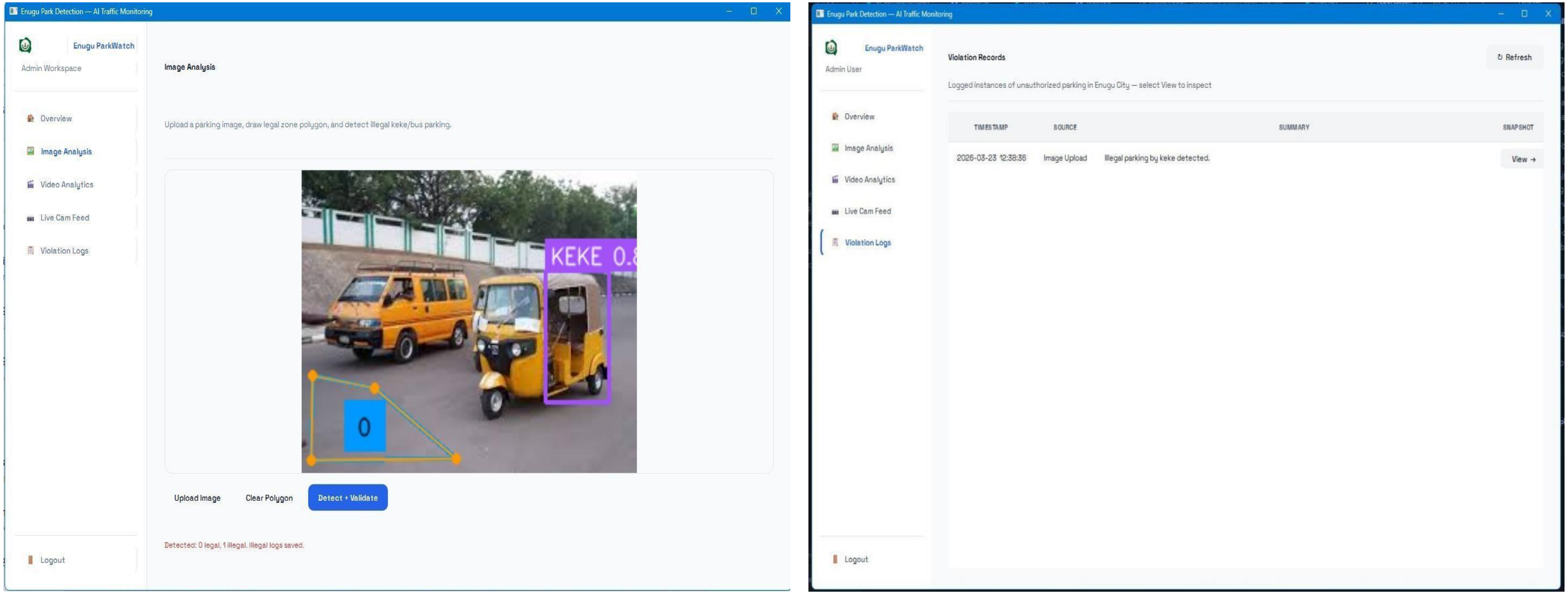

Figure 10: Dashboard Screen Showing Violation Detection with Bounding Box Annotations and Log Entries

4.5 Results Presentation and Analysis

This chapter discusses and analyzes the quantitative evaluation results generated by the Ultraalytics' validation pipeline after the completion of the 50 epochs training process. The results have been discussed using a range of artifacts including tables with metric results, precision versus recall curve, F1 vs confidence curve, recall vs confidence curve, confusion matrix, and training/validation batch examples.

4.5.1 Output Samples and Interface Screens

Figure 4.7 displays a mosaic of batches of training examples in the initial epoch of training (batches 0, 1, and 2) that show the annotated bounding boxes superimposed on the training examples from the Keke-bus-1 training dataset. From the images displayed, the varied nature of the training examples is evident by the presence of different Keke and bus objects in various lighting conditions, perspectives, distances, and degrees of occlusion. The classes index numbers (0 for Keke, 1 for bus) appear next to each image. The training batch mosaics display the effectiveness of the use of mosaic augmentation technique, where four training images are used to form one training example.

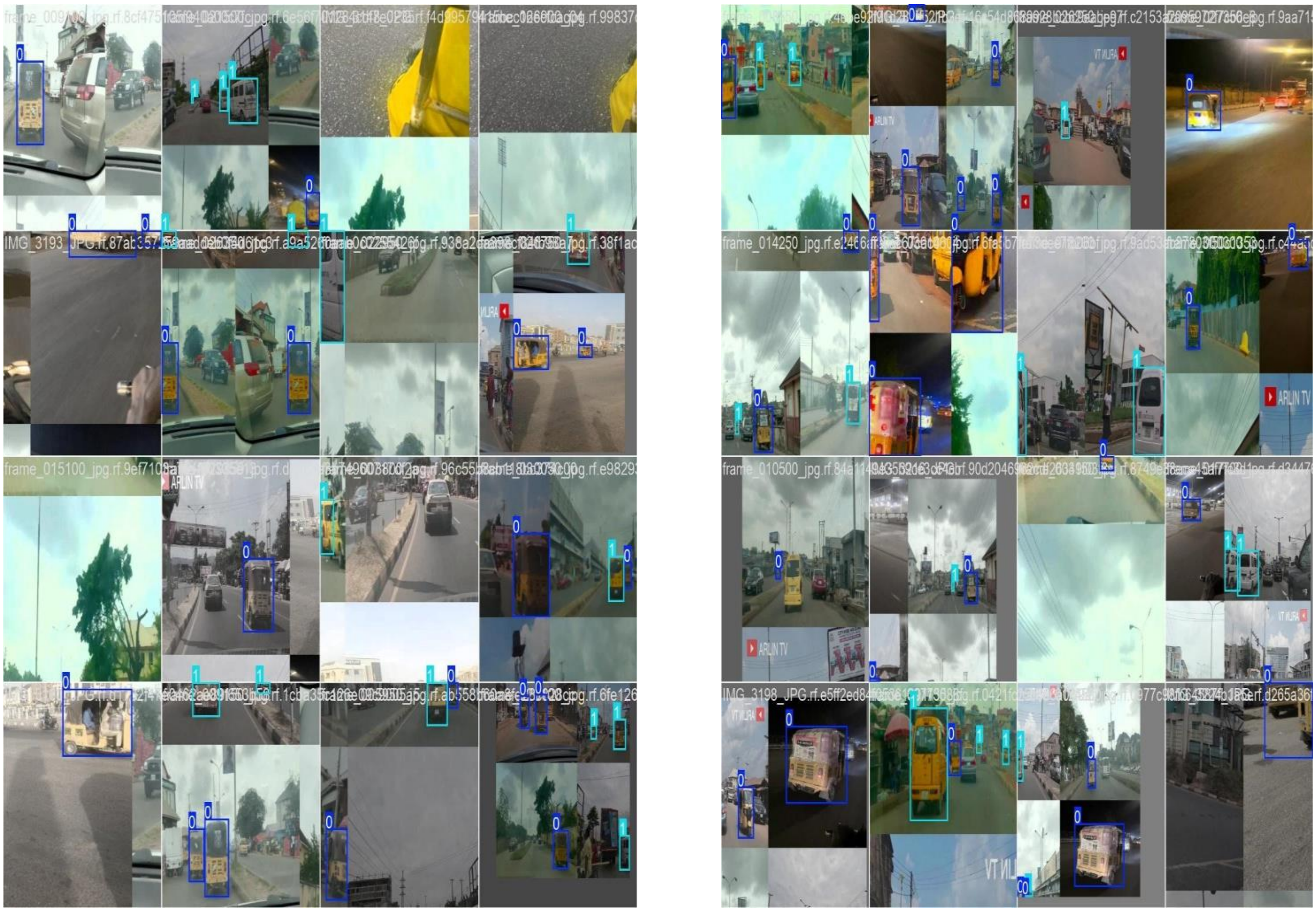

Figure 11: Training Batch Samples (Early Epochs) Showing Ground Truth Annotations on Augmented Training Images

Training batches for the latter epochs (441 and 442) are shown in Figure 4.8, at which time the mosaic augmentation is turned off because of the close_mosaic parameter value being set at 10. These latter epoch training batches illustrate how the model learns on clean individual images, thus enabling it to make accurate predictions based on unaugmented input towards the end of training. The annotations used for these frames cover a variety of complex detection tasks, such as distant, partially occluded and border region vehicles.

Figure 12: Training Batch Samples (Late Epochs) Showing Ground Truth Annotations on Clean Unaugmented Images

Figure 4.9 and 4.10 illustrate the validation batch label samples containing the true annotations of the validation images. The images above clearly indicate that the validation set is comprised of a representative distribution of the target vehicles among diverse road conditions. Figure 4.11 illustrates the corresponding validation batch prediction samples where the predicted bounding boxes and class labels are superimposed onto the validation images. A comparative study between the true labels and the model predictions clearly indicates high accuracy for most of the cases where the model accurately predicts and localises most of the Keke and bus objects contained in the validation set images. Misclassification is mainly due to the absence of very small or occluded vehicle objects, which is expected due to the drawbacks of the single-stage detectors at small object scales.

Figure 13: Validation Batch Ground Truth Labels

Figure 14: Validation Batch Ground Truth Labels (Second Batch)

Figure 15: Validation Batch Predicted Detections with Class Labels and Confidence Scores

4.5.2 Discussion of Results and System Performance

Quantitative evaluation of the trained YOLO11s model for the Keke-bus-1 dataset validation subset resulted in obtaining the metrics shown in Table 4.1 below. In general, the model showed the following performance metrics - 0.758 of Precision, 0.832 of Recall and 0.838 of mean Average Precision with Intersection over Union threshold 0.5 (mAP@0.5) for both classes. Mean Average Precision mAP@0.5:0.95 over the whole range of IoU thresholds was 0.607. All the mentioned above indicates the high operational performance of the model regarding object detection and good recall rate which shows that the model detected a large number of all objects present in validation images.

Table 6:Summary of YOLO11s Model Evaluation Metrics on Validation Set

Metric	Keke (Class 0)	Bus (Class 1)	All Classes
Precision	0.930 (norm.)	0.860 (norm.)	0.758

Recall	0.930 (norm.)	0.860 (norm.)	0.832
AP@0.5	0.946	0.716	0.831
mAP@0.5	-	-	0.838
mAP@0.5:0.95	-	-	0.607
F1 Score (all)	-	-	0.78 at 0.539

There is a significant variance in terms of per-class performances. The AP for the Keke class was 0.946, performing much better than the AP for the bus class at

0.716. There are two reasons why there is an imbalance in terms of per-class performance; one is that in the training dataset, there were more Keke than buses by a 1.5:1 ratio, and the other is the high variability of minibuses relative to tricycles on the roads in Enugu. The minibuses in Enugu range widely in size, colour, and body shape – from Hiace-type vans to commercial buses, while the Keke tricycles have the consistent three-wheeled structure with the distinct yellow and black colours of Enugu State.

Precision-Recall graph (Figure 4.12) is a class-wise graph depicting the trade-off between precision and recall at all confidence levels. As can be seen, the curve for the Keke class stays on high precision values over a prolonged period of recall, which indicates that the model has good detection capacity when it comes to detecting Keke objects without having a very large number of false positives. Therefore, the Area Under Curve (AP = 0.946) is large, which means comprehensive detection. On the other hand, the bus class curve declines more quickly with increasing recall due to the increased complexity of detecting both precision and recall for the bus class. The total mAP@0.5 of 0.831 from the PR graph matches the value from the last epoch of training.

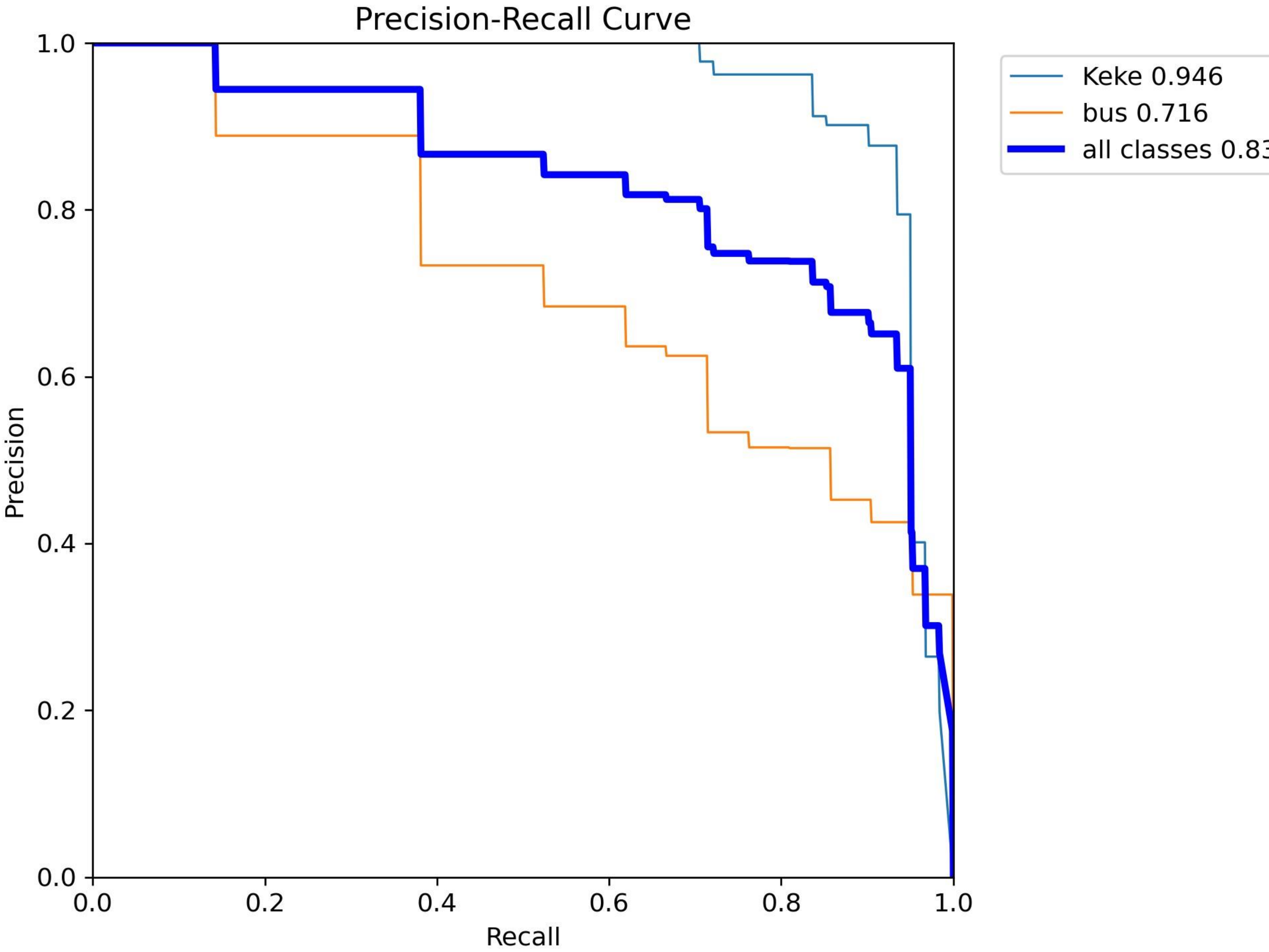

Figure 16: Precision-Recall Curve for Keke and Bus Classes (Keke AP = 0.946, Bus AP = 0.716, mAP@0.5 = 0.831)

The F1-Confidence graph (Figure 4.13) shows the F1-Score in relation to the confidence detection threshold level for both individual classes and all the classes combined. The best F1-Score achieved for all classes was 0.78, at the confidence threshold of 0.539. This particular score has been used as the confidence threshold recommendation for the installed system, since it is the point where the maximum trade-off between precision and recall is reached. For the confidence threshold below 0.539, the model detects more detections, though with many false positives, and vice versa, with a confidence threshold above 0.539. In addition, the F1 curve for Keke class detection is significantly higher compared to the F1 curve for bus class detection, with the Keke class maintaining an F1-score above 0.85 in a relatively wider range of confidence intervals from around 0.2 to 0.8

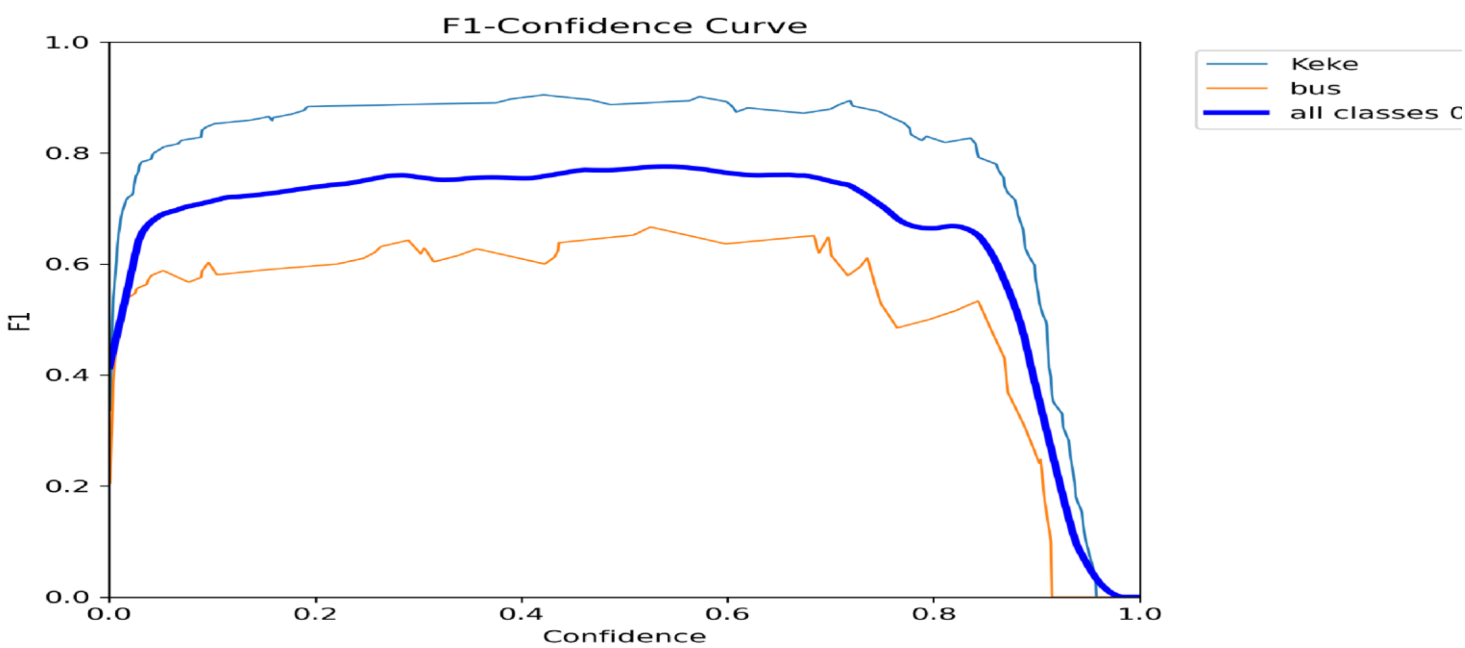

Figure 17: F1-Confidence Curve Showing Optimal F1 of 0.78 at Confidence Threshold 0.539

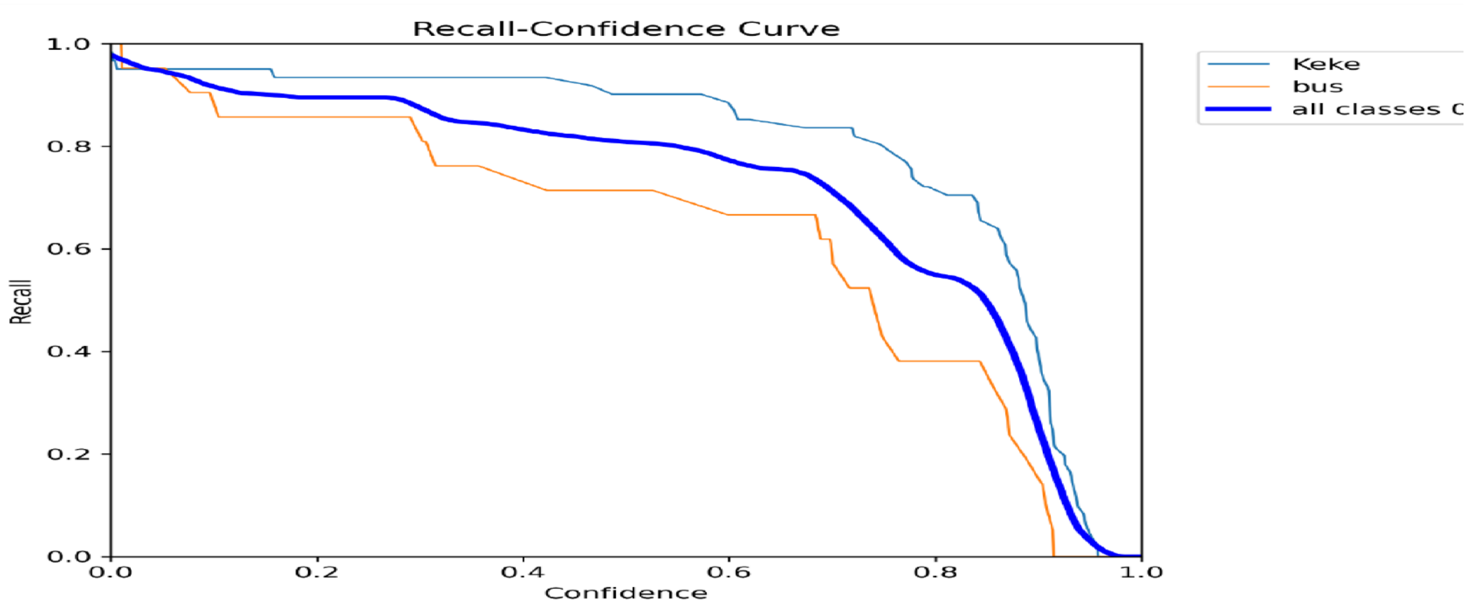

Figure 18: Recall-Confidence Curve Showing Maximum Combined Recall of 0.98

Recall-confidence graph (refer Figure 4.14) clearly shows that the recall of all the classes taken together is 0.98 at a confidence threshold of near zero, i.e., at confidence threshold of zero, the model detects 98 percent of the objects present in ground truth without applying any confidence threshold on evaluation. With

increase in confidence threshold, the recall drops; however, the recall of Keke class remains higher than that of bus class. This particular graph is helpful in such situations where, while using the model, maximizing detection sensitivity is more important than precision, like when the downstream violation log is reviewed by humans, in which case a smaller threshold value can be used.

The confusion matrix (Figure 4.15) gives a quantitative insight into the classification performance of the model on the Keke, bus, and background categories within the validation data set. According to the matrix, out of 61 ground truth cases of Keke, 57 were accurately predicted as Keke while 4 were wrongly predicted as background (miss detections). Out of the total number of buses evaluated, 19 were correctly predicted as buses, 18 were predicted as background (misses) while 0 instances of Keke were wrongfully predicted as buses. It is interesting to note that 10 background instances were wrongly predicted as Keke. These are false detections where the model wrongly identifies the background instances as tricycles. The normalized confusion matrix (Figure 4.16) represents the above figures in percentages of the total number of ground truths in each category. The Keke category had a normalised true positive rate of 0.93 thereby proving the recall level for this category. On the other hand, the Bus category had a normalised true positive rate of 0.86 based on the correct bus predictions and the total number of ground truths for the bus instances. The relatively high percentage of buses categorized as background (0.14) shows the difficulty of detecting minibuses.

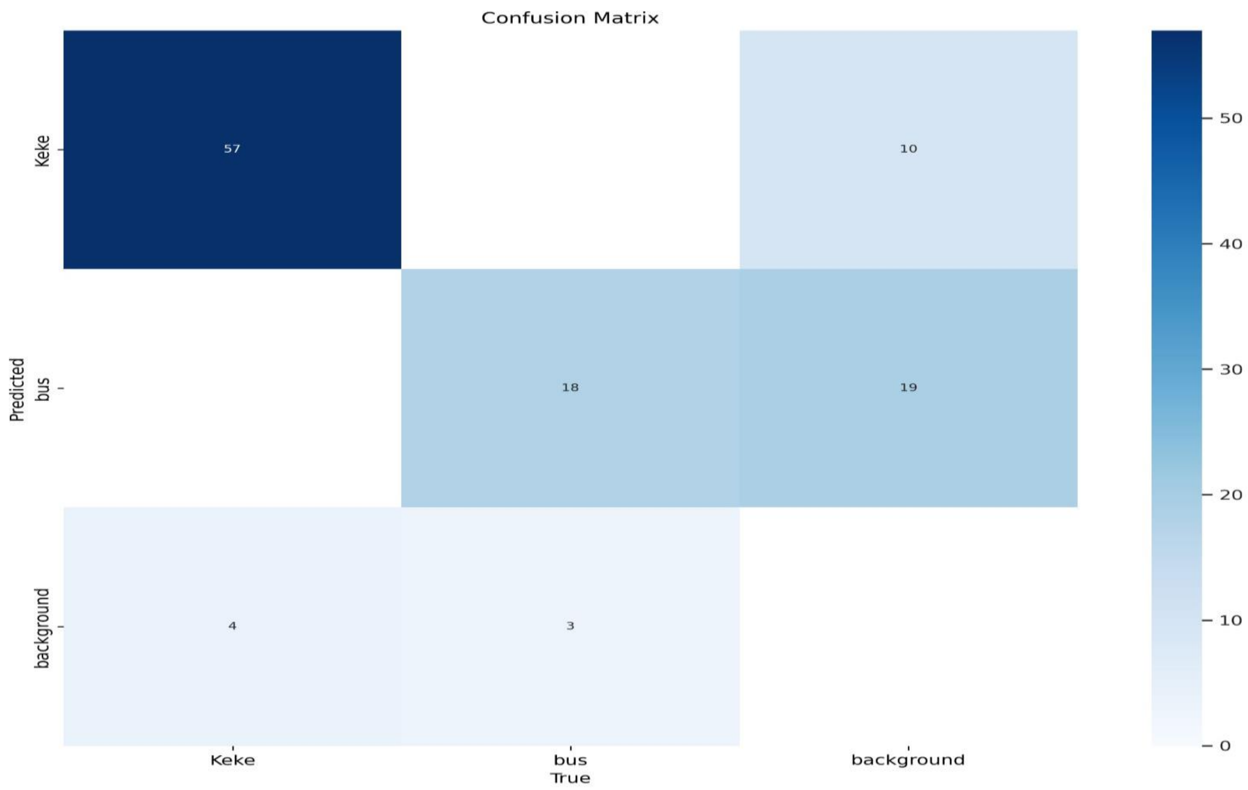

Figure 19: Confusion Matrix — Raw Detection Counts Across Keke, Bus, and Background Classes

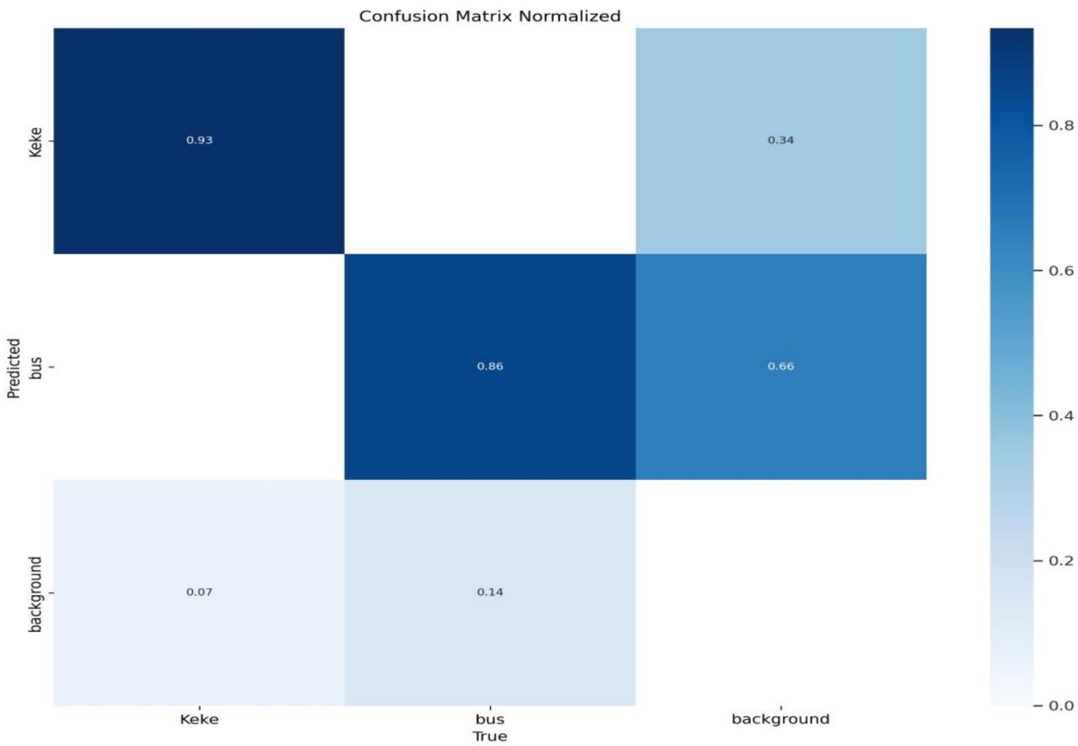

*Figure 20: Normalised Confusion Matrix — Row-Normalised True Positive Rates per Class*81

Overall, the outcome of the evaluations shows that the trained YOLO11s meets the set minimum performance requirement of mAP@0.5 $\geq$ 0.80 as required by the system requirements, having attained a final mAP@0.5 of 0.838. The performance of the model is impressive for the Keke class while the performance for the bus class is operationally satisfactory. The difference in performance in each class is a result of the nature of the dataset that can be remedied in future by collecting more data on the bus class. The training loss plot indicates the stability in convergence with no indication of overfitting since the validation loss was decreasing in tandem with the training loss during training. As a whole, the system of YOLO11s detection model, polygon zone enforcement algorithm, and PyQt6 interface is technically feasible and applicable for automated traffic enforcement using computer vision-based vehicle monitoring in Enugu metropolis.

CHAPTER FIVE

SUMMARY, RECOMMENDATIONS, AND CONCLUSION

5.0 Introduction

In this chapter, the conclusion, recommendations, and findings for the project "Computer Vision-based Detection and Monitoring of Tricycles and Minibuses in Enugu City" will be stated. This chapter is going to highlight the main problem statement that served as motivation behind this research, discuss the adopted methods, and highlight the key results obtained throughout the course of this project. The future
recommendations and suggestions that should be considered in order to make this system more valuable will be provided. These improvements will include the ways through which this system can become more efficient and effective.

5.1 Summary

This problem emerged due to the continuous occurrence of unauthorized parking of vehicles referred to as Keke (tricycles) and minibuses in the metropolis of Enugu, which largely results in traffic chaos and disarray. Currently, the only means through which this is controlled is through the deployment of traffic officers who suffer from fatigue, low spatial coverage, corruption, inability to generate digital records, and are environmentally exposed. So far, no technical measures have ever been developed to solve this particular problem in Enugu before, making this project highly pertinent and novel.

In order to address the identified problem, the following key objectives were formulated: document current methods for detecting and monitoring unauthorized parking of tricycles and minibuses in Enugu; train computer vision models for automatic detection and localization of said vehicles in image data; evaluate the models' performances according to the object detection metrics; develop web and desktop applications using the trained models.

For the achievement of the first objective, an analysis of the current manual enforcement system was done through the observation of fieldwork and the interview of traffic enforcement officers in Enugu. This investigation validated dependence on visual surveillance and established six distinct types of limitations, which were discussed in Chapter Three to set up a baseline metric for the comparison with the system. custom annotated dataset of Keke and minibus images from the roads of Enugu was compiled using direct data gathering methods and manually annotated through CVAT tool. The obtained dataset consisting of about 175 Keke samples and 115 bus samples was uploaded to Roboflow and then used for fine-tuning the YOLO11s detector utilizing transfer learning on COCO pre-trained weights. Training of the detector took place for 50 epochs on Google Colab GPU machine with a batch size of 16 and 640×640 input resolution.
The evaluation stage carried out on the validation subset resulted in mAP@0.5 equaling 0.838 and mAP@0.5:0.95 reaching 0.607, which is well above the minimum performance level requirement of 0.80. For Keke class Mean Precision equals 0.946, and Mean Precision for bus class is 0.716. The F1-Score value for both classes together equaling 0.78 was reached at the confidence threshold of 0.539, while maximum recall is 0.98 for zero confidence threshold.

To meet the fourth objective, a PyQt6 Desktop Application was created and implemented using the trained detection model, OpenCV, Supervision, and CVZone2. The application consists of an administrative GUI with three major modules; namely, Homepage Module which introduces the application, Login Module which handles administrator authentication, and Dashboard Module where the stream is received from the camera and allows setting of polygonal legal parking zones, conducts real-time YOLO11s predictions in each frame, determines the position classification of vehicles through zone intersection process, and stores all the violations recorded together with their respective frames in a JSON database.

Overall, it can be concluded that the implementation of a computer vision model based on the YOLO11s network trained on a custom dataset consisting of locally collected images is capable of delivering a functionally adequate level of detection ability regarding Keke and minibuses within Enugu conditions. The ability to detect these vehicles is well suited to be combined with a spatial algorithm for enforcement along with a proper administration panel to deliver a working solution.

5.2 Recommendations

Taking into account the limitations in the current implementation, the results achieved through the project, and observations made throughout system testing, the following suggestions for the future can be made. Currently, the configuration module responsible for generating polygons for the legal parking zone configuration supports only static images. Each new input requires the creation of a new zone manually, and the system itself cannot remember previous zones. Therefore, the most important suggestion for further development can be considered an expansion of functionality in the polygon zone configuration module to cover video and camera streams in real time. This will provide the possibility to use one zone configuration for multiple frames of an active stream and thus ensure continuous observation without interruption. This task is to be implemented by integrating a video input loop within the PyQt6 dashboard and launching it on a separate thread not to block the interface during inference processing.

Another suggestion for future work would be implementing an alert notification system in real time. Presently, violation occurrences are recorded in the JSON database in a passive manner, where administrators have to manually look through the log panel. In future implementations, push notifications or sound-based alerts in case of violations can be considered, as this would ensure that enforcement agents receive alerts about violations without having to keep their eyes fixed on the monitoring dashboard all the time. This would add greater value to this tool especially if there is one person monitoring different camera feeds.

It is worth noting that while the dataset collected for model training was adequate enough to represent road conditions in Enugu, it was small in number and mostly consisted of daytime data from select locations.
Future research can focus on acquiring a larger dataset with an increased number of bus class instances. Class imbalance is what caused the AP score of the bus class to be lower than that of the Keke class. Nighttime data would also add value, as well as employing data augmentation techniques such as adding simulated noise or adjusting the gamma levels to create more nighttime images.

In contrast to the currently employed JSON database which suffices for development purposes only, future iterations should transition towards employing relational databases for production usage such as PostgreSQL which would allow concurrent access by multiple administrative users and structuring queries and reporting on violation data, as well as storing data that can be proven to be valid legally. Transitioning to web-based deployment as opposed to the current desktop application would improve accessibility by allowing enforcement administrators to check their feeds and logs through any internet-enabled device without the need for installing additional software.

The current implementation lacks vehicle tracking capabilities between inference frames, which results in the same vehicle being flagged multiple times in the violation logs due to persisting in the prohibited area during inference frames. In this regard, future improvements should consider implementing a multiple object tracking model, for example, the BoT-SORT model described in the Ultralytics configuration file (tracker: botsort.yaml) which enables assigning unique identities for tracked vehicles during all inference frames. Such capability would allow counting a single vehicle once within each inference frame as a single violation and, therefore, generating an improved log file containing violations.

5.3 Conclusion

It is clear from this project that computer vision technology, including the single-stage YOLOv1 object detection architecture, can be successfully used to detect and keep track of Keke tricycles and minibuses in the road environment of Enugu. This was made possible through the creation of a customized dataset consisting of images collected through the conduct of field work in Enugu and labeled using quality annotation software to capture the visual features of the target vehicles. Training of a YOLO11s model on this data set yielded a mean Mean Precision of 0.838 with IoU = 0.5, a Keke class Mean Precision of 0.946, and a maximum combined recall of 0.98, thus validating the use of the model for its intended application.t6 desktop administrative interface has been built such that it takes data from a stream of video feeds from cameras, detects the target vehicle, checks whether it is parked within the designated legal space, and logs all cases where violations occur in a sustainable database. In doing so, it covers the six types of limitations in the current manual monitoring system by being automatic, objective, scalable, and free of fatigue and human errors associated with deployment positions.

Challenges such as those faced in the implementation of the project, particularly the challenge of creating a large enough dataset to work with, due to the non-existence of freely available labelled data sets for Nigerian road vehicles, underscore the need for grassroots-level data gathering to advance the development of AIpowered transport systems in Africa. The data set developed in the course of this study is a valuable resource for further investigations in the field. This data set offers a technical base for implementing automatic vehicle monitoring in the city of Enugu, contributing not only a working prototype, but also a tested detection algorithm, and a methodology that can be used in further computer vision research in Nigeria. With the improvements proposed in section 5.3, the tool described in this paper has the potential to be improved into a reliable enforcement mechanism that would help combat unauthorised parking, improve traffic circulation, and bring Nigerian transportation management to the technological age.

REFERENCES

Adeniyi, O. (2017). Operational characteristics and regulatory challenges of informal transport systems in developing countries. *Journal of Urban Mobility and Transport*, *14*(3), 45–58.

Adetunji, M. O., & Akintayo, O. A. (2026). Dynamics of urban transport modes: Integrating formal and informal transport systems in Nigerian cities. *African Journal of Infrastructure and Development*, *8*(1), 112– 126.

Adewumi, A. O., & Allopi, D. (2019). The role of tricycles (keke) in urban passenger transportation: A case study of selected Nigerian cities. *International Journal of Traffic and Transportation Engineering*, *9*(2), 89– 101.

Azfar, T., Li, J., Yu, H., Cheu, R. L., Lv, Y., & Ke, R. (2022). *Deep learning based computer vision methods for complex traffic environments perception: A review* (arXiv Preprint No. arXiv:2211.05120). arXiv. https://arxiv.org/abs/2211.05120

Barwari, A. M., Al-Saffar, M. A., & Hussein, S. K. (2021). Environmental and health implications of poor vehicle emissions maintenance in developing urban regions. *Journal of Environmental Studies and Pollution Research*, *28*(4), 512–524.

Bashiru, A. R., & Waziri, O. O. (2023). Assessment of the causes and effects of traffic congestion on road users in major road intersections. *Federal University of Technology Minna Journal of Research*, *5*(1), 31–45.

Batool, S. (2012). Bribery and institutional failures in traffic enforcement mechanisms. *Journal of Public Policy and Governance*, *19*(3), 142–155.

Bautista, J. R., Gorostiza, A. M., & Ramos, L. C. (2016). Technological advancements in traffic monitoring systems: A review of global enforcement tools. *International Journal of Computer and Systems Engineering*, *10*(4), 310–318.

Berwo, M. Y., Al-Husainy, M. A. F., & Mohammed, M. A. (2023). Deep learning approaches for vehicle detection and classification in intelligent transportation systems: A review. *Sensors, 23*(10), 4832. https://doi.org/10.3390/s23104832

Cervero, R., & Golab, J. (2017). Informal transport in the developing world: Regulatory challenges and infrastructural integration. *Transportation Research Part A: Policy and Practice*, *103*(2), 220–235.

Charde, P., Sharma, R., Kumar, A., & Singh, V. (2024). AI and IoT based parking violation detection and alert system. *International Journal of Advanced Research in Computer Science and Engineering Technology, 13*(4), 115–123.

Carrera García, D., Vázquez-Arellano, M., & Ruiz-Garcia, L. (2022). Intelligent parking monitoring system based on artificial vision and machine learning. *Multimedia Tools and Applications, 81*(22), 31823–31847.

Dede, D., Sarsil, M. A., Shaker, A., Altintas, O., & Ergen, O. (2023). *Next-gen traffic surveillance: AIassisted mobile traffic violation detection system* (arXiv Preprint No. arXiv:2311.16179). arXiv. https://arxiv.org/abs/2311.16179

Ebimobowei, A. (2021). Limitations and document integrity issues in contemporary manual traffic violation systems. *Nigerian Journal of Technological Research*, *16*(2), 67–75.

Gwilliam, K. (2023). *Cities on the move: A World Bank urban transport strategy review*. World Bank Publications.

Hadi, A. S., George, L. E., & Ahmed, M. A. (2024). Computer vision-based smart parking systems: A comprehensive review. *Iraqi Journal for Electrical and Electronic Engineering, 20*(1), 1–15.

Jayatilleke, S., Perera, M., & Bandara, S. (2010). Analysis of traffic rule violations among public service vehicles in developing urban frameworks. *Journal of South Asian Transportation Studies*, *15*(1), 53–62.

Jiang, Y., Feng, X., & Li, X. (2020). *Research on illegal parking detection based on convolutional neural networks*. In *Proceedings of the International Conference on Artificial Intelligence and Smart Manufacturing*. Atlantis Press.

Kathait, S. S., & Kumar, A. (2025). Computer vision and deep learning based approach for violations due to illegal parking detection. *International Journal of Computer Applications*, *186*(70), 1–10.

Kayami, H. (2019). Inefficient enforcement of traffic regulations: Factors contributing to regulatory lapses in developing nations. *Global Transport and Infrastructure Journal*, *31*(2), 88–97.

Manisalidis, I., Stavropoulou, E., Stavropoulos, A., & Bezirtzoglou, E. (2020). Environmental and health impacts of air pollution: A review on the combustion of fossil fuels. *Frontiers in Public Health*, *8*(14), 1–12.

Nguyen, H., & Sartipi, M. (2024). *PakLoc and PakSta: Smart camera-based parking space localization and occupancy detection using computer vision* (arXiv Preprint). arXiv.

Ojojo, P. O., Alao, O. A., & Bello, A. I. (2007). Exhaust emissions from poorly maintained commercial tricycles and their effect on urban air quality. *Nigerian Journal of Environmental Management*, *12*(3), 204– 215.

Olawole, M. O., & Aloba, O. (2018). Informal public transport and urban functionality in Nigeria:

Operational attributes, challenges, and safety hazards. *Journal of Transport Geography*, *69*, 182–195.

Raji, B. A., & Waziri, O. O. (2016). Tricycles (keke) as an alternative mode of urban public transport in Nigeria: The National Poverty Eradication Programme (NAPEP) perspective. *Journal of Sustainable Development in Africa*, *18*(4), 45–59.

Ren, Y. (2024). Intelligent vehicle violation detection system under human–computer interaction and computer vision. *International Journal of Computational Intelligence Systems, 17*(1), 40. https://doi.org/10.1007/s44196-024-00427-6

Tahir, M. (2018). Institutional coordination and bribery cultures in transport law enforcement agencies.

Journal of Development Administration, *24*(1), 102–115.

Tang, H., Peng, A., Zhang, D., Liu, T., & Ouyang, J. (2020). *SSD real-time illegal parking detection based on contextual information transmission*. *Computers, Materials & Continua, 62*(1), 293–307. https://doi.org/10.32604/cmc.2020.06427 Zhang, Y., Wang, H., Liu, X., & Chen, J. (2022). Parking status recognition based on machine vision and YOLO object detection. *Procedia Computer Science, 201*, 450– 457.

Vlahov, D., Galea, S., & Ompad, D. C. (2025). Urbanization, demographic shifts, and the deterioration of transport architecture in sub-Saharan African powerhouses. *Journal of Urban Health*, *102*(1), 14–28.

World Health Organization. (2017). *Global status report on road safety: Time for action*. World Health Organization.

Wang, Z., He, S. Y., & Leung, Y. (2022). Estimating the impacts of illegal curb parking using computer vision and traffic surveillance data. *Transportation Research Part A: Policy and Practice, 163*, 180–196.