**COMBINING LARGE LANGUAGE MODELS (LLMs) AND CONVOLUTIONAL NEURAL NETWORKS (CNNs) FOR AUTONOMOUS OBJECT DETECTION IN SMART CITIES**

**BY**

**ODUWALE OYINKANSOLA OGHENEYOMA**

**MATRIC NUMBER: 21/8357**

**A PROJECT WRITTEN AND SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, COLLEGE OF PURE AND APPLIED SCIENCE, CALEB UNIVERSITY, IMOTA, LAGOS.**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE**

**JUNE, 2025**

# **DECLARATION**

I hereby declare that I carried out the work reported in this project in the Department of Computer Science, Caleb University, under the supervision of Prof. Moses. K. Aregbesola. I also solemnly declare that to the best of my knowledge; no part of this report has been submitted here or elsewhere in a previous application for the award of a degree. All sources of knowledge used have been duly acknowledged.

**Oduwale Oyinkansola Ogheneyoma**

Name Signature and Date

# **CERTIFICATION**

This is to certify that the project “**COMBINING LARGE LANGUAGE MODELS (LLMs) AND CONVOLUTIONAL NEURAL NETWORKS (CNNs) FOR AUTONOMOUS OBJECT DETECTION IN SMART CITIES**” carried out by, **ODUWALE OYINKANSOLA OGHENEYOMA**, with matriculation number **21/8357** in the Department of Computer Science, College of Pure and Applied Sciences, Caleb University, Imota, Lagos, Nigeria.

**Prof. Moses K. Aregbesola**

***Project Supervisor*** **Signature and Date**

**Dr. Adeniyi W. Akanni**

***Head of Department*** **Signature and Date**

**Prof. Olaide A. Adesanya**

***Dean of COPAS*** **Signature and Date**

**DEDICATION**

The creator of the heavens and the earth, God Almighty, is thanked for His assistance and guidance in the completion of this project. I would not have been able to finish this project successfully without his mercy and love.

# **ACKNOWLEDGMENT**

First and foremost, I would like to express my profound gratitude to Almighty God for His countless blessings, guidance, and strength throughout my stay at Caleb University.

I am deeply thankful to my beloved parents Mr and Mrs Oduwale for their unconditional love, prayers, and unwavering support. Their encouragement has been my greatest source of motivation and inspiration.

I would also like to extend my heartfelt thanks to my project supervisor, Prof M.K Aregbgesola, for his valuable guidance,and constructive feedback, which greatly contributed to the successful completion of this project.

My sincere aprreciation goes to Caleb University for providing the necessary facilities and a supportive environment.

Lastly, I wish to acknowledge my girls and my friends for their love, support, and words of encouragement throughout this journey.

# **ABSTRACT**

The increasing complexity of smart city infrastructures demands advanced artificial intelligence solutions capable of real-time situational awareness, contextual reasoning, and high-precision object detection. This study explores a novel hybrid framework that integrates Large Language Models (LLMs) and Convolutional Neural Networks (CNNs) to enhance autonomous object detection in dynamic urban environments. CNNs, renowned for their visual pattern recognition, are combined with the semantic reasoning capabilities of LLMs to develop a multimodal system that not only identifies urban objects but also interprets their contextual relationships. Utilizing YOLOv5 for image-based detection and the Gemini transformer model for natural language processing, the research implements a feature-level fusion approach supported by cross-attention mechanisms. The system is evaluated using the MS COCO dataset, and performance metrics such as precision, recall, mean Average Precision (mAP), and BLEU score are employed to assess model efficiency. The experimental results reveal that the integrated LLM-CNN framework offers superior performance in terms of accuracy, interpretability, and real-time responsiveness compared to unimodal models. This research contributes significantly to the field of multimodal AI and demonstrates its potential to revolutionize smart city applications, including traffic monitoring, surveillance, and autonomous decision-making.

Keywords:

Smart Cities, Convolutional Neural Networks (CNNs), Large Language Models (LLMs), Multimodal AI, YOLOv5, Gemini Model, Object Detection, Natural Language Processing (NLP), Contextual Reasoning, Autonomous Systems.

**TABLE OF CONTENTS**

**TITLE** ………………………………………………………………………………………………………………………….……………….…….1

[**DECLARATION** 2](#_Toc201502230)

[**CERTIFICATION** 3](#_Toc201502231)

[**DEDICATION** 4](#_Toc201502232)

[**ACKNOWLEDGMENT** 5](#_Toc201502233)

[**ABSTRACT** 6](#_Toc201502234)

[**TABLE OF CONTENTS** 7](#_Toc201502235)

[**LIST OF FIGURES** 10](#_Toc201502236)

[**LIST OF TABLES** 11](#_Toc201502237)

[**LIST OF ABBREVIATIONS** 12](#_Toc201502238)

[**CHAPTER ONE** 15](#_Toc201502239)

[**INTRODUCTION** 15](#_Toc201502240)

[**1.1 Background of the Study** 15](#_Toc201502241)

[**1.2 Research Problem** 21](#_Toc201502242)

[**1.3 Research Objectives** 22](#_Toc201502243)

[**1.4 Research Questions** 23](#_Toc201502244)

[**1.5 Hypothesis** 23](#_Toc201502245)

[**1.6 Significance of the Study** 24](#_Toc201502246)

[**1.7 Scope and Delimitations** 25](#_Toc201502247)

[**1.8 Thesis Structure** 26](#_Toc201502248)

[**CHAPTER TWO** 28](#_Toc201502249)

[**LITERATURE REVIEW** 28](#_Toc201502250)

[**2.1 Overview of Large language models and Convolutional Neural Networks** 28](#_Toc201502251)

[**2.2 Theoretical Foundations** 34](#_Toc201502252)

[**2.3 State-of-the Art Approaches in NLP and CV** 38](#_Toc201502253)

[**2.4 Multimodal AI Models** 43](#_Toc201502254)

[**2.5 Gap Analysis** 46](#_Toc201502255)

[**2.6 Summary of Gaps and Opportunities** 48](#_Toc201502256)

[**CHAPTER THREE** 50](#_Toc201502257)

[**RESEARCH METHODOLOGY** 50](#_Toc201502258)

[**3.1 Research Design** 50](#_Toc201502259)

[**3.2 Data Collection** 51](#_Toc201502260)

[**3.3 Data Preprocessing** 52](#_Toc201502261)

[**3.4 Model Development** 55](#_Toc201502262)

[**3.4.1 Baseline Models** 56](#_Toc201502263)

[**3.4.2 Hybrid Models** 57](#_Toc201502264)

[**3.4.3 Multimodal Model** 58](#_Toc201502265)

[**3.5 Model Evaluation** 59](#_Toc201502266)

[**3.5.1 NLP Evaluation Metrics (Gemini Model)** 60](#_Toc201502267)

[**3.5.2 CV Evaluation Metrics (YOLOv5 Model)** 60](#_Toc201502268)

[**3.5.3 Multimodal Evaluation Metrics** 62](#_Toc201502269)

[**3.6 Tools and Frameworks** 63](#_Toc201502270)

[**3.7 Expermental Setup** 64](#_Toc201502271)

[**3.7.1 Training** 64](#_Toc201502272)

[**3.7.2 Fine-tuning for Hybrid Multimodal Integration** 65](#_Toc201502273)

[**3.7.3 Deployment Output** 65](#_Toc201502274)

[**3.8 Ethical Considerations** 66](#_Toc201502275)

[**3.8.1 Bias and Fairness** 66](#_Toc201502276)

[**3.8.2 Privacy and Surveillance** 66](#_Toc201502277)

[**3.8.3 Misuse and Dual-Use Risks** 66](#_Toc201502278)

[**3.8.4 Transparency and Accountability** 67](#_Toc201502279)

[**CHAPTER FOUR** 68](#_Toc201502280)

[**RESULTS AND DISCUSSION** 68](#_Toc201502281)

[**4.1 Presentation of Results** 68](#_Toc201502282)

[**4.2 Analysis of Results** 73](#_Toc201502283)

[**4.2.1 Comparison of the Performance Metrices between Baseline and Hybrid Models** 73](#_Toc201502284)

[**4.2.2 Surprising Findings or Discrepancies** 73](#_Toc201502285)

[**4.3 Comparative Analysis** 74](#_Toc201502286)

[**4.3.1 Key Observations** 75](#_Toc201502287)

[**4.4 Discussion of Findings** 75](#_Toc201502288)

[**4.4.1 Linking Findings to Research Objectives and Hypotheses** 75](#_Toc201502289)

[**4.4.2. Key Findings and Interpretation** 75](#_Toc201502290)

[**4.5 Implications of the Findings** 76](#_Toc201502291)

[**4.5.1 Real-World Applications in Industry and Academia:** 76](#_Toc201502292)

[**4.6 Challenges and Limitations** 77](#_Toc201502293)

[**CHAPTER FIVE** 80](#_Toc201502294)

[**CONCLUSION AND RECOMMNDATIONS** 80](#_Toc201502295)

[**5.1 Summary of the Study** 80](#_Toc201502296)

[**5.1.1 Recap of the Research Problem, Objectives, and Major Findings:** 80](#_Toc201502297)

[**5.2 Contributions to the Field** 81](#_Toc201502298)

[**5.2.1 Development of a Novel Hybrid Model** 81](#_Toc201502299)

[**5.2.2 Advancements in Multimodal Learning** 81](#_Toc201502300)

[**5.3 Recommendations for Future Research** 82](#_Toc201502301)

[**5.3.1 Areas for Further Exploration** 82](#_Toc201502302)

[**5.4 Practical Applications** 84](#_Toc201502303)

[**5.4.1 Industry Use Cases for the Hybrid CNN–LLM Model** 84](#_Toc201502304)

[**5.5 Conclusion** 85](#_Toc201502305)

[**References** 87](#_Toc201502306)

[**Appendices** 94](#_Toc201502307)

**LIST OF FIGURES**

[Figure 3.1 Flowchart showing the combination of LLMs and CNNs for autonomous object detection in smart cities. Source- Reseacher’s design 38](#_Toc201091847)

[Figure 3.2 Bounding Boxes and Final Prediction in YOLO. Source- (Pandey, Puri, & Varde, 2018) 40](#_Toc201091848)

[Figure 3.3 The flow of YOLO v5. Source- (Ren et al 2024) 42](#_Toc201091849)

[Figure 3.4 Object Detection on Real-Time Traffic using YOLO. Source - (Pandey, Puri, & Varde, 2018) 43](#_Toc201091850)

[Figure 3.5 Feature-Level Fusion Approach. Source- (Researcher’s design) 44](#_Toc201091851)

[Figure 3.6 Cross Attention Mechanism. Source- (Researcher’s design) 45](#_Toc201091852)

[Figure 4.1 Confusion Matrix and ROC Curve for LLM. (Source- Researcher’s design) 55](#_Toc201091853)

[Figure 4.2 Confusion Matrix for CNN. (Source- Researcher’s design) 56](#_Toc201091854)

[Figure 4.3 Confusion Matrix and ROC Curve for the Hybrid (LLM-CNN) Model. (Source- Researcher’s design) 57](#_Toc201091855)

# **LIST OF TABLES**

[Table 4.1 58](#_Toc201092707)

[Table 4.2 59](#_Toc201092708)

# **LIST OF ABBREVIATIONS**

| **Abbreviation** | **Full Meaning** |
| --- | --- |
| **AI** | Artificial Intelligence |
| **CNN** | Convolutional Neural Network |
| **LLM** | Large Language Model |
| **API** | Application Programming Interface |
| **BERT** | Bidirectional Encoder Representations from Transformers |
| **BLEU** | Bilingual Evaluation Understudy |
| **BP** | Brevity Penalty |
| **NLP** | Natural Language Processing |
| **AP** | Average Precision |
| **CV** | Computer Vision |
| **R-CNN** | Region-based Convolutional Neural Networks |
| **ELMo** | Embedding from Language Models |
| **EM** | Exact Match |
| **FP** | False Positives |
| **FN** | False Negatives |
| **GAN** | Generative Adversarial Network |
| **GLUE** | General Language Understanding Evaluation |
| **GPT** | Generative Pre-trained Transformer |
| **GPT-3/4** | Generative Pre-trained Transformer 3/4 |
| **GPU** | Graphics Processing Unit |
| **IoU** | Intersection over Union |
| **BYOL** | Bootstrap Your Own Latent |
| **LLaMA** | Large Language Model Meta AI |
| **MLLM** | Multimodal Large Language Model |
| **mAP** | Mean Average Precision |
| **MS COCO** | Microsoft Common Objects in Context |
| **YOLO** | You Only Look Once |
| **PIL** | Python Imaging Library |
| **RNN** | Recurrent Neural Network (implied from model types, not explicitly listed) |
| **SimCLR** | A simple framework for contrastive learning of visual representations |
| **SQuAD** | Stanford Question Answering Dataset |
| **TP** | True Positives |
| **TN** | True Negatives |
| **T5** | Text-to-Text Transfer Transformer |
| **ViT** | Vision Transformer |
| **WSGI** | Web Server Gateway Interface |
|  |  |

# 

# **CHAPTER ONE**

# **INTRODUCTION**

## **1.1 Background of the Study**

A branch of artificial intelligence dedicated to processing, comprehending, and producing text and images that resemble those of a human is known as AI-based language and image models. These models examine enormous volumes of data and produce insightful results by using deep learning techniques, mainly neural networks.

AI-Based Language Models use input prompts to process and produce writing that is human-like. To comprehend context, grammar, and semantics, they are trained on big text datasets and employ machine learning strategies such as transformer architectures (e.g., GPT, BERT).

The branch of computer science known as natural language processing (NLP) is concerned with how computers and human languages interact. A crucial part of natural language processing (NLP) are language models, which are trained on enormous collections of natural language text to forecast the probability of various words or phrases in a particular context. The area of NLP has undergone a revolution thanks to recent advancements in language models. A number of potent language models have emerged in recent years, including Meta AI's LLaMA (Large Language Model Meta AI), Google's BERT (Bidirectional Encoder Representations from Transformers), OpenAI's GPT-3 (Generative Pre-trained Transformer-3), and ELMo (Embedding for Language Model). These have greatly improved the state-of-the-art in natural language processing tasks like text production, sentiment analysis, and question answering.

A new language representation paradigm called Google BERT (2018) was unveiled, it stands for Bidirectional Encoder Representations from Transformers. Left and right contexts are co-trained in all layers to pre-train deep bi-directional representations based on unlabeled text. State-of-the-art models for a variety of tasks, such as question answering and language inference, can be created by modifying the existing BERT model with only an additional output layer. The architecture of the task-specific model does not need to be significantly altered in order to achieve this. In order to enable pretrained deep bidirectional representations, BERT makes use of masked language models. It is the first representation model to attain state-of-the-art performance with fine-tuning as a method for a variety of tasks, including sentence-level and token-level tasks. Many models that were specifically created for certain tasks perform worse than this. A multi-layer bidirectional Transformer encoder is the model architecture for BERT.

GPT-3 (Generative Pre-trained Transformer 3) is an advanced language model developed by OpenAI. It is built on the Transformer architecture, and trained on an extensive dataset to produce human-like natural language responses. As its most recent open-access version, it offers 175 billion parameters, thus becoming the largest language model to date. GPT-3 has been trained for multiple purposes, such as text generation, translation, question answering, sentiment analysis, summarization, chatbots, and text completion. Text that closely mimics human writing can be produced by GPT-3 in a fluid and cohesive manner. The Transformer architecture, a neural network architecture intended for natural language processing, serves as the foundation for GPT-3. Together, an encoder and a decoder make up the Transformer architecture, which produces text in natural language.

Image models analyze and generate images using deep learning techniques like convolutional neural networks (CNNs) and transformers. They can perform tasks like image recognition, object detection, style transfer, and image generation.

Like GPT-3, DALL·E is a transformer language model. It receives both the text and the image as a single stream of data containing up to 1280 tokens, and is trained using maximum likelihood to generate all of the tokens, one after another. This training procedure allows DALL·E to not only generate an image from scratch, but also to regenerate any rectangular region of an existing image that extends to the bottom-right corner, in a way that is consistent with the text prompt (OpenAI, 2021).

DALL-E 2 is created by OpenAI and is a successor of DALL-E. It can create more realistic images than DALL-E at higher resolutions and can combine concepts, attributes, and styles. DALL-E 2 is trained on approximately 650 million image-text pairs scraped from the Internet. Since DALL-E 2 code is not available, we were not able to generate images on a large-scale DALL-E2: with multimodal latent space. Another stream of text-to-image diffusion models in latent space relies on multimodal contrastive models, where image-embedding and text encoding are matched in the same representation space (Zhang et al., 2023).

Stable Diffusion (2022) is a deep learning text-to-image model. Inpainting, outpainting, and creating image-to-image conversions under the guidance of a text prompt are among its various uses, but its main function is to produce detailed visuals conditioned on text descriptions. A type of deep generative artificial neural network is Stable Diffusion, which is a latent diffusion model. It can operate on the majority of consumer hardware with a modest GPU and at least 4 GB of VRAM, and its code and model weights have been made freely available

Innovation is being driven across industries by AI domains like Multimodal AI, Convolutional Neural Networks (CNNs), Large Language Models (LLMs), Computer Vision, and Natural Language Processing (NLP). These technologies improve decision-making, automation, and efficiency.

A branch of artificial intelligence (AI), natural language processing (NLP) studies how people and computers communicate using natural language. NLP has undergone a revolution in recent years due to the advancement of deep learning techniques. The capabilities of NLP systems have been greatly enhanced by the inclusion of designs like transformers, long short-term memory networks (LSTMs), and recurrent neural networks (RNNs) (Vaswani et al., 2017). Thanks to these developments, complex models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have been developed that are able to comprehend and produce text that is human-like with a high degree of accuracy (Devlin et al., 2019; Radford et al., 2019).

NLP is used to process text data in smart healthcare and is linked to communication between humans and machines. Clinical text and other text data are the two categories into which the text data can be divided. Medical notes, diagnostic reports, electronic prescriptions, and other unstructured text records from electronic health record (EHR) systems make up the majority of clinical text, which originates from all clinical scenarios. different text data contain all text that appears inside different healthcare situations, e.g., surveys in population screening and publications for evidence-based reference. All smart healthcare scenarios involve communication, including human-robot contact in rehabilitation therapy and patient-provider communication in clinical inquiry. Applications like machine translation and rehabilitation robot user interfaces are also popular.

NLP technology make it easier to automatically create intelligent content for education. For instance, teachers struggle to create lesson plans from the deluge of knowledge available on the Internet or in textbooks. For example, NLP approaches can rapidly create a multiple-choice question related to the test question and the distractions given a passage from the textbook as the context and a concept as the right response. This could make it easier for teachers to create brief class tests.

Automation in fields requiring visual perception is made possible by computer vision, which enables AI to process and interpret visual information from photos and videos. Pre-processing techniques including Fourier filtering, edge detection, and morphological procedures have historically been the focus of image processing. The paradigm of image processing is expanded by computer vision to encompass object classification, tracking, and scene content understanding. Automatic Number Plate Recognition (ANPR), vehicle and person tracking, crowd analysis, and model-based vision are a few examples of computer vision applications.

There are many different ways that computer vision is used for security. Face recognition, crowd detection, anomalous human behavior, unlawful parking, speeding vehicle detection, and more are all included. The technology contributes to increased security and a reduction in many types of accidents.

Advanced Natural Language Processing models that comprehend and produce text that is human-like are known as large language models. They are employed in complicated thinking tasks, code authoring, and content creation.

The creation of chatbots and virtual assistants is one of the most often used LLM applications. These models are able to comprehend human questioning patterns and provide responses that closely resemble human speech. This has made it possible for companies to enhance customer service by offering round-the-clock assistance to clients without requiring human involvement. To enhance their ability to comprehend the subtleties of real language, LLM programs can be trained on a large volume of textual data. In order to offer individualized answers to questions, they can also be adjusted to target particular industries, such as banking, healthcare, and education.

LLM has significantly impacted the fields of automation and content creation. This is the process of employing generative AI systems to create written or multimedia material. Articles, blog entries, social media captions, product descriptions, and more can be produced by a massive language model application. These models may generate coherent and contextually relevant material by capturing the style, tone, and structure of various genres through extensive training on text data.

By predicting each object's class and bounding box coordinates, CNNs are used in object recognition to locate and identify objects in an image. They are deep learning models created especially for analyzing images. For precise picture classification and recognition, they take visual input and extract patterns and characteristics.

CNNs could be employed for remote learning in the field of facial recognition. CNNs' job would be to read students' facial expressions and determine whether they are smiling, frowning, surprised, or otherwise indicating that they are satisfied and grasp what they are being taught.

To capitalize on the advantages of each modality, multimodal language models combine many data kinds, such as text, voice, and images. These systems improve performance across a range of applications by merging diverse models, such as Convolutional Neural Networks (CNNs) for object detection and Large Language Models (LLMs) for language processing. Using the distinct benefits of both CNNs and LLMs, for example, allows for more effective object detection in smart cities.

Multimodal AI enables smart assistants to process text, audio, and visual inputs for more organic interactions. For smooth responses, Google Assistant integrates speech recognition, natural language processing, and visual information. Users may use their camera to search, translate, or recognize objects thanks to features like Google Lens. Additionally, multimodal AI increases accessibility for people with vision or speech problems. It enhances the adaptability and human-likeness of interactions by identifying facial expressions, tone, and context. Smart assistants will become progressively more tailored and intuitive as technology advances.

## **1.2 Research Problem**

In the rapidly evolving environment of smart cities, autonomous object detection plays a key role in assuring efficient urban administration, intelligent transportation systems, surveillance, and public safety. The ability of Convolutional Neural Networks (CNNs) to extract spatial information from visual data has led to their widespread adoption for object detection applications. The ability of CNNs to reason beyond visual signals is constrained, though, as they have trouble deciphering semantic context and comprehending intricate linkages between objects—two skills necessary for making well-informed decisions in real-time in dynamic urban locations.

However, there is much promise for understanding natural language, semantic links, and contextual information with Large Language Models (LLMs), especially those trained for multimodal tasks. By applying a higher level of abstraction and logic to visual inputs, LLMs can help robots understand scenes more similarly to how humans do. The combination of LLMs and CNNs for object detection is still not well investigated, despite its potential, particularly in the context of smart city applications where contextual awareness and spatial accuracy are essential.

In order to improve autonomous object detection systems in smart cities, it is necessary to bridge this gap by combining the spatial feature extraction strengths of CNNs with the semantic reasoning capabilities of LLMs. Such systems must be able to identify objects with high accuracy while also comprehending their roles, relationships, and potential implications within a particular urban context, such as differentiating between a pedestrian who is loitering and one who is waiting to cross the street, or comprehending that a vehicle parked in a no-parking zone may need to be alerted to city authorities.

Therefore, by investigating a hybrid strategy that combines CNNs and LLMs, this research aims to overcome the issue of restricted contextual awareness in existing object identification systems. In order to promote smarter, safer, and more responsive urban systems, real-time, semantically enriched object detection is intended.

## **1.3 Research Objectives**

The goal of this research is to improve autonomous object detection in smart cities by creating a hybrid model that smoothly combines Convolutional Neural Networks (CNNs) with Large Language Models (LLMs). The specific objectives include:

1. To create a multimodal AI model that integrates Convolutional Neural Networks (CNNs) and Large Language Models (LLMs) for autonomous object detection in smart cities.
2. To design a multimodal system capable of detecting and semantically interpreting urban objects and their contextual relationships in real-time.
3. To evaluate the performance of the proposed CNN-LLM framework, how does it compare to traditional CNN-based object detection models in terms of detection accuracy, contextual understanding, and real-time responsiveness within a smart city environment?

## **1.4 Research Questions**

1. How can the integration of Large Language Models (LLMs) and Convolutional Neural Networks (CNNs) enhance autonomous object detection in smart cities?
2. What are CNNs' and LLMs' respective advantages and disadvantages when it comes to object detection tasks?
3. How may LLMs enhance CNN object detection interpretability and contextual understanding?
4. When merging CNNs with LLMs for real-time object detection in urban settings, what are the trade-offs between efficiency and computation?
5. How might an LLM-CNN-based object detection system be used in smart city infrastructure, including security, and traffic monitoring?
6. What privacy and ethical issues surround the use of LLM-enhanced object detection in smart cities?
7. Which benchmarks and datasets are best suited to assess how well an LLM-CNN hybrid model performs in autonomous object detection?

## **1.5 Hypothesis**

Multimodal systems are superior to unimodal approaches because they integrate multiple sources of information, resulting in improved accuracy, robustness, and interpretability; unlike unimodal models, which rely on a single type of data (e.g., images or text), multimodal systems combine diverse data types, such as visual, textual, and auditory inputs, enabling a richer and more comprehensive understanding of complex scenarios; they improve decision-making by reducing ambiguities and compensating for weaknesses in individual modalities.

## **1.6 Significance of the Study**

This study's academic significance stems from the combination of CNNs and LLMs, which advances the field of multimodal AI. It shows the potential benefits of multimodal models and emphasizes their advantages over unimodal ones by showing how this combination improves contextual comprehension, interpretability, and robustness—three major issues in computer vision research.

This research is essential for industry in order to apply AI in urban settings in the real world. Intelligent transportation systems, security surveillance, and traffic monitoring all depend on autonomous object detection. Businesses and governments can increase automation efficiency, decrease false detections, and improve real-time decision-making by merging LLMs with CNNs. The study also tackles real-world issues including managing various lighting situations, gaps, and urban scenarios, which increases the scalability and dependability of AI-driven smart city solutions.

The combination of Convolutional Neural Networks (CNNs), which analyze visual data, and Large Language Models (LLMs), which process textual data, is studied in this work. The benefits of a multimodal learning strategy over conventional unimodal techniques are illustrated by combining these two models. This integration highlights the potential influence of multimodal models in a variety of real-world applications to enhance interpretability, contextual comprehension, and AI system robustness.

Through multimodal learning, the combination of CNNs and LLMs improves transformer models, especially Vision Transformers (ViTs), by improving their self-attention processes. Semantic information from LLMs helps transformers concentrate on the most pertinent textual and visual components, lowering noise and enhancing performance on tasks like object detection, anomaly detection, and scene understanding. Additionally, this integration improves transformer models' generality across a variety of datasets and real-world contexts, such as robots, smart cities, and autonomous systems, where flexibility in response to changing circumstances is crucial. Additionally, it improves transformers' capacity to decipher complex connections between several modalities, which results in increased precision in applications like object recognition, video analysis, and picture captioning.

The combination of CNNs and LLMs improves the accuracy and efficiency of AI-driven systems and has important real-world uses in smart city object recognition. This integration enhances object recognition capabilities, allowing for intelligent transportation management, pedestrian safety measures, and sophisticated traffic monitoring. Additionally, this method helps with autonomous monitoring and anomaly detection, which enables more accurate identification of potential threats or anomalies in urban surroundings. In order to provide safer, more responsive, and more effective urban infrastructure, this technology helps to improve situational awareness and decision-making.

## **1.7 Scope and Delimitations**

In particular, this research investigates how to improve object detection by combining computer vision (CV) and natural language processing (NLP). Instead of focusing on individual model enhancements, it highlights the combination of CNNs and LLMs. The study explores how contextual information from LLMs and text-based reasoning can improve CNN-based object detection's interpretability and decision-making. It highlights the benefits of multimodal fusion.

Additionally, it emphasizes contextual reasoning using Transformer-based Large Language Models (LLMs), such Gemini, GPT, or BERT. It describes how LLMs can help with object recognition by offering semantic comprehension, classification improvement, and scene description. The study doesn't concentrate on more general NLP uses, such as sentiment analysis or text production outside of object detection scenarios.

Convolutional Neural Networks (CNNs) are used in the study to detect objects. It takes into account common object detection tasks as tracking, localization, and classification in urban settings. Non-CNN-based vision models are not the subject of the study unless they support the multimodal fusion methodology. It does not extend to non-urban areas like medical imaging, industrial automation, or non-visual NLP applications; instead, it is used in smart city environments, such as traffic monitoring, surveillance, and urban automation. The study balances performance with real-time processing requirements by examining the computational efficiency of integrating CNNs and LLMs.

## **1.8 Thesis Structure**

This thesis is organized into five chapters:

Chapter One: Introduction – Provides background on AI-based language and image models, defines the research problem, objectives, questions, and hypotheses, explains the significance of the study, and outlines the thesis structure.

Chapter Two: Literature Review – This chapter explores existing AI models for language and image processing, focusing on transformers and convolutional neural networks (CNNs). It also examines multimodal AI models and highlights key research gaps and opportunities for future advancements.

Chapter Three: Research Methodology – Describes the research design, datasets, data preprocessing, model development (baseline and hybrid models), evaluation metrics, experimental setup, tools, and ethical considerations.

Chapter Four: Results and Discussion – Presents findings from model experiments, analyzes and compares results with existing state-of-the-art models, discusses implications, and highlights challenges and limitations.

Chapter Five: Conclusion and Recommendations – Summarizes key findings, contributions to the field, practical applications, and suggests directions for future research.

# **CHAPTER TWO**

# **LITERATURE REVIEW**

## **2.1 Overview of Large language models and Convolutional Neural Networks**

The emergence of smart cities seeks to improve urban living through technology, and applications such as environmental control, traffic monitoring, and surveillance depend heavily on autonomous object detection. In order to obtain reliable object detection, this paper investigates the integration of Convolutional Neural Networks (CNNs) and Large Language Models (LLMs) in this context.

The objective of object detection, a basic task in computer vision and image understanding, is to locate and identify things of interest within an image while giving them the appropriate class labels. Conventional approaches performed poorly and had trouble handling complicated visual data since they relied on hand-crafted features and shallow models. High-level semantics could not be captured by these approaches, which mixed contextual information with low-level features. Convolutional Neural Networks (CNNs) in particular, which automatically acquire complex, hierarchical features straight from data, helped deep learning overcome these constraints. These characteristics include high-level and semantic representations that are necessary for precise object detection (Neha et al., 2024).

Object recognition problems were dominated by conventional algorithms prior to the development of deep learning and its application in computer vision. Although these algorithms played a role in the early development of the area, their use was restricted to tiny datasets (Janiesch, Zschech, & Heinrich, 2021). However, as technology developed, conventional object detection algorithms become more and more insufficient, failing to satisfy the rigorous standards for effectiveness, precision, resilience, and flexibility. Later, the emergence of deep learning led to revolutionary developments in image processing and object recognition, resolving many of the drawbacks of conventional techniques and producing noticeably better results. The capacity of deep learning to efficiently handle and learn from massive amounts of data is one of its main advantages.

Convolutional Neural Networks (CNNs) have revolutionized how machines understand and interpret visual data in the rapidly developing field of computer vision. The challenge of object detection, a fundamental idea with broad applications in robotics, autonomous cars, surveillance, and healthcare, lies at the heart of this revolution. Significant advancements in this field have been fueled by CNN-driven object detection algorithms, which provide unparalleled scalability, efficiency, and precision (Mousa, A. M., & El-Sayed, A. 2024).

CNN-based object detection algorithms usually take a multi-phase approach, first generating initial potential object regions (called proposals) and then refining them using techniques like bounding box regression and classification. To increase accuracy and efficiency, this pipeline is commonly improved using techniques including non-maximum suppression, anchor box mechanisms, and region proposal networks (RPNs). The combination of different machine learning and deep learning approaches has greatly aided recent developments in object recognition (Mousa, A. M., & El-Sayed, A. 2024).

Modern object recognition frameworks are now built on CNNs; models such as SSD (Single Shot MultiBox Detector), YOLO (You Only Look Once), and Faster R-CNN have shown state-of-the-art performance in a variety of applications. These models are appropriate for smart city applications where accuracy and speed are crucial since they use hierarchical feature extraction to detect objects in real-time (Redmon et al., 2016; Liu et al., 2016). The most famous CNN, created by Krizhevsky, Sutskever, and Hinton, had an error rate of 15%, 10% less than the nearest rival. The computationally intensive training process required two GPUs (Graphic Processing Units).

It was Ross Girshick and his colleagues who developed the R-CNN algorithm in 2014. It achieved 53.3% mAP when tested on the PASCAL VOC 2010–12 dataset. Three modules make up this algorithm. The algorithm defines bounding boxes in the first section to look for potential regions of interest. The algorithm then extracts features from each area that was registered in the previous stage using a deep neural network. Lastly, the object's features are used to determine its classification (Tsirtsakis et al 2025).

A faster R-CNN was suggested by (Ren et al 2015). By substituting a different set of CNN layers called the reginal proposal network (RPN) for the unsupervised methods used to generate regional proposals, Faster R-CNN further increases speed and creates an end-to-end deep net model. In order to share all CNN computations, region-based fully convolutional neural networks (R-FCN) add location-sensitive maps on top of RPNs, continuing the concept of sharing CNN layers from the faster R-CNN. R-FCN outperforms quicker R-CNN in terms of processing speed, and the entire object detection procedure is an end-to-end learning process (Sanil et al 2020).

By using ResNet as a backbone to extract features that are utilized for all regional classifiers, Fast R-CNN (Sanil et al 2020) increases the running speed. The majority of CNN calculations are shared inside the backbone of Fast R-CNN, removing the need for CNN computations to limit running performance (Girshick et al 2015). By doing away with the necessity of running the CNN on each proposal separately, Fast R-CNN greatly increased speed. However, it still relied on selective search for generating region proposals, which remained a bottleneck.

Real-time object recognition was transformed by Joseph Redmon's YOLO framework, which introduced a grid-based method for concurrently predicting bounding boxes and class probabilities (Jiang et al 2020). YOLO is especially well-suited for applications that require real-time performance because of its creative design, which makes detection extremely effective.   
From YOLOv1 to the most recent YOLOv9, the YOLO architecture has continuously evolved since its conception, bringing improvements in accuracy, speed, and efficiency with each iteration (Kumar, C., & Punitha, R. 2020). Important developments include new loss functions designed to maximize performance and the addition of anchor boxes, which enhance the model's detection capabilities for objects of various sizes and forms. Every YOLO iteration also has optimized backbones, which help with better detection capabilities and quicker processing times.

The YOLO framework's development between 2015 and 2023 demonstrates the iterative improvements made in response to new object detection issues (Das et al 2024). Every update has advanced the dual objectives of reducing latency and increasing detection accuracy, which are crucial for the ongoing advancement of real-time object detection technology.

Darknet, a proprietary deep learning framework created by Joseph Redmon, is used to write YOLO (Redmon et al 2015). Written in C and CUDA, it is an open-source neural network framework that facilitates computation on both CPUs and GPUs. Installing Darknet is the first step towards YOLO implementation.

YOLO uses a configuration file, yolo.cfg, where number of convolution layers, filters, and bounding box width and height are pre-defined. The yolo.cfg file needs to be updated for a user to train YOLO using the COCO (Common Objects in Context) data set, where multiple object names are stored (Redmon et al 2015). YOLO provides models for pre-trained weights. For each layer's weight/bias values, these weights are shown as big binary blobs of 32-bit floating points.

The Single Shot MultiBox Detector (SSD) was presented by Liu and colleagues (2016), it is a one-stage model that enhances YOLO by employing anchors with various scales and aspect ratios inside each grid cell. Regressors refine each anchor, assigning probability across categories and predicting object identification on numerous feature maps for various scales. SSD integrates results across maps and trains end-to-end using a weighted localization and classification loss. Through massive data augmentation and hard negative mining, SSD achieves accuracy comparable to Faster R-CNN while enabling real-time inference (Neha et al 2024).

A language model learns to recognize patterns and acquire a wide range of capabilities. At inference time, it then makes use of these skills to quickly adjust to or identify the intended task (Brown et al 2020) The inner loop of this process, which takes place within the forward-pass upon each sequence, is referred to as "in-context learning."

Advanced AI systems that can understand and generate language similar to that of humans are known as large language models (Hamadi et al 2023). Like transformer models, they are based on intricate neural networks and draw inspiration from the intricacies of the human brain. By learning from large volumes of data, these models are able to understand context, which makes their text outputs—whether they are answering your questions or narrating a story—feel more natural. In essence, a big language model is a sophisticated AI companion that has human-like comprehension and speech abilities. LLMs' contextual data can enhance object detection. They are able to interpret scene descriptions or user queries, for instance. Despite their great potential, not all of them are simple to implement. Even with the best configuration, LLMs have trouble with tasks.

The authors of (Chen et al 2023) present MiniGPT-v2, a model intended to function as a single interface for a variety of vision-language tasks, including visual question answering, visual grounding, and image description.

In addition to a number of tasks that call for on-the-fly reasoning or domain adaptation, such unscrambling words, utilizing a novel term in a phrase, or doing 3-digit arithmetic, GPT-3 performs well on a variety of NLP datasets, such as translation, question-answering, and cloze tasks.   
Large Language Models (LLMs), such BERT and GPT-3, have demonstrated impressive skills to comprehend and produce human language. Although they are mostly employed in natural language processing (NLP), there is growing interest in how they might improve object detecting systems.

With models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) establishing new performance benchmarks, natural language processing (NLP) has improved quickly. By employing a bidirectional transformer, BERT, which was first shown by (Devlin et al 2019), revolutionized natural language processing and enabled profound contextual understanding of language. BERT is very effective for tasks like sentiment analysis, question answering, and text categorization because it processes text by taking into account the entire input context, unlike standard sequential models.

BERT pretrains a bidirectional language model by utilizing the transformer encoder. In a variety of NLP tasks, it has shown an amazing capacity to comprehend language: It demonstrated state-of-the-art performance on tasks such as natural language inference, textual entailment, sentiment analysis, and phrase similarity modeling, and it outperformed previously presented language models on the GLUE benchmark. By surpassing even humans on two evaluation metrics (ExactMatch (EM) and the F1 score), it showed remarkable comprehension on the SQuAD v1.1 test set. Through pretraining or fine-tuning, BERT's strong transfer learning ability enables it to adjust to various jobs (Wang et al 2024).

BERT's uniform architecture across many jobs is one of its unique features. The final downstream design and the pre-trained architecture differ only little. Based on the initial implementation detailed in (Vaswani et al 2017) and made available in the tensor2tensor library, BERT's model architecture is a multi-layer bidirectional Transformer encoder.

GPT models have proven to be remarkably adept at producing language that is both logical and contextually appropriate. One prominent example, GPT-3 (Brown et al 2020), uses a transformer architecture with 175 billion parameters to provide state-of-the-art performance on a variety of NLP tasks. These developments have improved the interpretation and creation of textual descriptions for sign language by making it easier to include sophisticated language knowledge into real-time translation systems.

The Generative Pre-trained Transformer (OpenAI GPT) (Radford et al 2018) is one example of a fine-tuning technique that adds few task-specific parameters and is trained on the downstream tasks by merely fine-tuning all pre-trained parameters.

## **2.2 Theoretical Foundations**

The theoretical foundations of object detection systems are advanced artificial intelligence (AI) designs like transformer models, convolutional neural networks (CNNs), and multimodal learning theories. These technologies provide the computational underpinnings needed to process and enhance object detection systems in smart cities.

Transformer is a novel neural network architecture that (Vaswani et al 2017) introduced. The field of natural language processing was swiftly transformed by that contemporary architecture. Models that use this Transformer architecture, like as GPT and BERT, have completely surpassed the prior state-of-the-art networks. It outperformed the previous methods by such a large margin that these Transformer-based designs appear to be the foundation of all the most recent cutting-edge models. Unlike Convolutional Neural Networks (CNNs), which rely on local feature extraction, transformers capture global dependencies in data, making them powerful for tasks such as object detection, image segmentation, and natural language understanding.

The Transformer-based architecture method eliminates the recurrent architecture in favor of an attention mechanism-only approach. Moreover, neither the hard parallelization problem nor gradient vanishing affect it. This speeds up and makes it easier to train larger networks. Transformers were first used in auto-regressive models, following early sequence-to-sequence models, generating output tokens one by one. However, the prohibitive inference cost (proportional to output length, and hard to batch) lead to the development of parallel sequence generation, in the domains of audio, machine translation, word representation learning, and more recently speech recognition (Carion et al 2020).

Transformer-based architectures are now transforming the object detection space by making it possible for more precise, effective, and scalable object detection systems. By using self-attention processes to interpret pictures holistically, these models—like DEtection TRansformer (DETR) and Swin Transformer—get beyond CNNs' drawbacks in intricate urban settings (Carion et al 2020).

Detection Transformer, also known as DETR, introduced by Carion and his team in 2020. It is a transformer encoder-decoder architecture with a set-based global loss that uses bipartite matching to enforce unique predictions. DETR generates the final set of predictions in parallel by reasoning about the relationships between the objects and the global image context given a fixed small set of learned object queries. The new model is conceptually simple and does not require a specialized library, unlike many other modern detectors. DETR demonstrates accuracy and run-time performance on par with the well-established and highly-optimized Faster RCNN baseline on the challenging COCO object detection dataset. DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a “no object” (∅) class prediction (Carion et al 2020).

Two ingredients are essential for direct set predictions in detection: a set prediction loss that forces unique matching between predicted and ground truth boxes and an architecture that predicts (in a single pass) a set of objects and models their relation. Any deep learning framework that offers a transformer architecture implementation with a few hundred lines and a shared CNN backbone can use DETR. PyTorch allows for the implementation of DETR inference code in less than 50 lines (Carion et al 2020).

Swin Transformer (Shifted Window Transformer) (Liu et al.2021). It performs admirably as computer vision's all-purpose backbone. Because Swin Transformer computes self-attention only within each local window, it has a linear computation complexity to input image size and creates hierarchical feature maps by combining image patches in deeper layers. As a result, it can function as a general-purpose backbone for tasks involving dense recognition and image categorization. On the other hand, because self-attention is computed globally, earlier vision transformers had quadratic computation complexity to input image size and generate feature maps with a single low resolution. Swin Transformer constructs a hierarchical representation by starting from small-sized patches and gradually merging neighboring patches in deeper Transformer layers. With these hierarchical feature maps, the Swin Transformer model can conveniently leverage advanced techniques for dense prediction.

A deep learning model called Convolutional Neural Networks (CNNs) is mostly used to interpret grid-like data, including photographs. CNNs specialize in pattern recognition, they use convolutional layers, which apply filters (or kernels) to input data, to automatically and adaptively learn spatial hierarchies of features. CNNs' capacity to identify patterns, edges, textures, and other characteristics from unprocessed pixel data has allowed them to achieve remarkable success in tasks like object detection, segmentation, and picture classification. CNNs is a particular part of the neural network or a section of layers, within these layers are filters which perform the pattern recognition which CNN is good at. The application of the filter increases as we go deeper into the network and can perform more tasks (Lecun et al 2015).

CNNs are used in object identification to identify and find items in a picture by anticipating each object's class and bounding box coordinates. Conventional CNN-based object identification models, such as Faster R-CNN, employ CNNs to extract features before refining object positions using other methods (such Region Proposal Networks, or RPNs). To suggest potential object regions in an image, CNNs are used with RPNs in object detection models such as Faster R-CNN. The Region Proposed Network (RPN) creates possible item locations (bounding boxes) by scanning the image using sliding windows (Lui et al 2023).

Multiple modes make up the data in a multimodal context, and each modality has a unique correlational structure and type of representation. For instance, text is typically represented as discontinuous sparse word count vectors, while images are represented by pixel intensities or dense, real-valued feature extractor outputs (Srivastava et al 2012). It was first presented by Ngiam and colleagues (2011). To comprehend images in the context of language, models such as CLIP (Contrastive Language-Image Pretraining, Radford et al., 2021) integrate textual and visual information. It enables zero-shot categorization and transfer learning by mapping text and images into a common embedding space via contrastive learning. When modified for multimodal models, BERT (Bidirectional Encoder Representations from Transformers) and T5 (Text-to-Text Transfer Transformer) have also been used to integrate textual data with other modalities (e.g., picture, video) for tasks such as visual question answering, image captioning, and more.

By combining input from multiple sensors and sources, multimodal learning can improve system performance in object detection. Object detection systems have always relied on visual information from cameras (pictures or videos), but adding additional modalities can improve their accuracy and versatility.

The fundamental building blocks for creating reliable object detection systems are these underlying ideas and designs. These frameworks allow systems to efficiently process and detect objects in a variety of settings by utilizing cutting-edge deep learning techniques as transformer-based models, multimodal learning, and convolutional neural networks (CNNs). In the context of smart cities, these systems are able to deliver real-time insights into the urban scene by analyzing and interpreting a variety of data sources, including textual information, photographs, and sensor inputs. Because smart city surroundings are dynamic and complex, the combination of these technologies guarantees that object detection systems are not only accurate but also flexible.

## **2.3 State-of-the Art Approaches in NLP and CV**

Recent years have witnessed tremendous progress in the domains of computer vision (CV) and natural language processing (NLP), primarily due to advances in deep learning methods, especially the use of transformer-based architectures. Text creation, sentiment analysis, image captioning, and object detection are just a few of the activities that have significantly improved as a result of these developments.

A branch of artificial intelligence (AI), natural language processing (NLP) studies how people and computers communicate using natural language. NLP has undergone a revolution in recent years due to the advancement of deep learning techniques. The capabilities of NLP systems have been greatly enhanced by the inclusion of designs like transformers, long short-term memory networks (LSTMs), and recurrent neural networks (RNNs) (Vaswani et al., 2017). Thanks to these developments, complex models such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have been developed that are able to comprehend and produce text that is human-like with a high degree of accuracy (Devlin et al., 2019; Radford et al., 2019).

For example, Google's BERT model, which pre-trains deep bidirectional representations by jointly conditioning on both left and right context in all layers, has achieved state-of-the-art results in multiple benchmarks (Devlin et al., 2019). Transformers in particular have been a game-changer, as their capacity to process word sequences in parallel rather than sequentially allows for more efficient training on large datasets, leading to improvements in various NLP tasks, including machine translation, question answering, and text summarization.

For instance, natural language processing (NLP) is crucial for virtual assistants such as Google Assistant, Amazon's Alexa, and Apple's Siri to comprehend customer inquiries and offer pertinent answers (Hoy, 2018). NLP has also improved accessibility for people with disabilities. Users can utilize voice commands to interact with computers and smartphones thanks to speech recognition technologies that are powered by sophisticated natural language processing models. People with mobility disabilities or visual impairments have benefited most from this, as it has made digital services easier to use and given them more autonomy (Srinivasan et al., 2020).

Examples of transformer models that integrate textual and visual data include VisualBERT (Li et al., 2019) and UNITER (Chen et al., 2019). By learning joint representations of images and text, these models achieve remarkable outcomes in image captioning and visual question answering. When producing an output, transformer models use a process known as attention that enables the model to concentrate on various segments of the input sequence.

Multimodal natural language processing models, such as OpenAI's CLIP (Contrastive Language-Image Pretraining, Radford et al., 2021), combine textual and visual information to produce a combined representation space. This makes it possible for the model to carry out tasks like text-based picture classification using zero-shot. In order to treat all NLP jobs as text-to-text problems, T5 (Text-to-Text Transfer Transformer, Raffel et al., 2020) enhanced transformer-based techniques. This allowed the model to handle a wide range of tasks, from question answering to translation, in a single framework.

The rapid increase in processing power and clever learning algorithms has led to a considerable advancement in computer vision theory and technology in recent years. Significant advancements in a number of domains, including as object detection and tracking, picture analysis and comprehension, object recognition, and smart cities, have resulted from this success.

Multimodal approaches will allow for a more thorough understanding and interpretation of visual content, which will help improve performance in high-level vision tasks like image captioning, visual question answering, and image synthesis. The development of multi-modal, large-scale models that can successfully integrate information from multiple sources, such as images, text, and audio, is the key to the future of computer vision.

Computer Vision (CV) and Natural Language Processing (NLP) must be combined in order to translate image data into audio descriptions. One of the primary benefits of this combination is the ability to interpret data intelligently, as NLP offers a linguistic context for processing visual content in a CV, enabling a more profound and sophisticated comprehension of images (Orynbay et al 2024).

NLP has also witnessed an increase in interest, particularly in large-scale unlabeled datasets used for unsupervised pre-training of language models, and CV has made great progress in object detection, semantic segmentation, and visual-content categorization.

Vision Transformers (ViTs) A major change in the field of computer vision. Because of their capacity to process spatial hierarchies in images, Convolutional Neural Networks (CNNs) have historically dominated architectures for image-related tasks. Nonetheless, the Transformer architecture's success in natural language processing prompted scientists to investigate its possibilities in the field of vision. ViTs approach image tasks by dividing inputs into fixed-size patches, linearly embedding them, and then processing them as a sequence using the Transformer architecture. The self-attention mechanism in Transformers, which had been pivotal in capturing long-range dependencies in text, proved equally adept at handling spatial relationships in images (Dosovitisky et al 2021).

By enabling models to learn meaningful representations without the need for labeled data, self-supervised learning techniques like SimCLR (Chen et al., 2020) and BYOL (Bootstrap Your Own Latent, Grill et al., 2020) have transformed computer vision and shown promising results on downstream tasks such as object detection, image classification, and segmentation.

The capabilities of large language models (LLMs) have been greatly expanded by recent developments in transformer-based LLMs that were pretrained on Web-scale text corpora. For instance, Microsoft's Co-Pilot systems are powered by OpenAI's ChatGPT and GPT-4, which are universal task solvers that may be used for more than just natural language processing. They can follow human instructions for difficult new tasks and, when necessary, do multi-step reasoning. As a result, LLMs are increasingly serving as the fundamental building block for the creation of artificial general intelligence (AGI) or general-purpose AI agents (Bubeck et al 2023).

The term "large language models" (LLMs) primarily refers to transformer-based neural language models, like GPT-3 and GPT-4 (OpenAI, 2020–2023), which are pre-trained on vast amounts of text data and have tens to hundreds of billions of parameters. These models use billions of parameters to generate text that is human-like and to answer intricate questions. Bidirectional context learning was first introduced by BERT (Bidirectional Encoder Representations from Transformers, Google, 2018), and NLP tasks were combined into a single framework by T5 (Text-to-Text Transfer Transformer, Google, 2019). Lightweight and open-source, LLaMA (Meta AI, 2023) was created with research and practical applications in mind.

LLMs and generative AI have become potent instruments that are pushing the limits of natural language processing (NLP) and providing previously unheard-of capabilities in a range of fields. As a particular use of generative AI, LLMs are fundamental to the larger field of generative AI capabilities. Their exceptional comprehension and production of human language creates several opportunities in a variety of fields.

In order for LLMs and generative AI to learn the patterns and relationships required for their tasks, the enormous amount of data needed for training must be processed and modeled using these potent computer resources. AI model training on enormous text and code datasets has been made easier by the creation of potent new computing hardware, such as Graphics Processing Units (GPUs). According to (Goodfellow et al. 2014), “these models make use of deep learning methods like Generative Adversarial Networks (GANs)”.

For uses like item detection in smart cities, recent research has increasingly concentrated on creating multimodal models that combine computer vision and natural language processing (NLP). These models gain a greater contextual awareness by combining these two modalities, which makes it possible to make better decisions based on a variety of data sources.

There are significant ramifications for smart city technologies when NLP and computer vision are combined. For example, object detection systems can use textual information from sources like news articles, social media, and sensor readings in addition to visual inputs from cameras and sensors. Capabilities in fields like environmental monitoring, public safety, and traffic management are improved by this combination.

Autonomous vehicles are one important area where this integration is crucial. While computer vision allows real-time identification of objects, people, and traffic signs, assuring safer navigation, natural language processing (NLP) models assist in interpreting voice commands and facilitating communication with passengers.

## **2.4 Multimodal AI Models**

While the most recent large language models are excellent at text-based tasks, they frequently have trouble understanding and processing other types of data. Multimodal models overcome this limitation by combining different modalities, allowing for a more thorough understanding of diverse data. Multimodal language models explore multiple data types, including images, text, language, audio, and other heterogeneity.

Multiple data kinds, such as text, audio, graphics, and more, are combined in a multimodal model. Although text data is the primary source of training and application for traditional large language models (LLMs), their comprehension of other data types is limited. GPT-3, GPT-4V (OpenAI, 2023), BERT (Devlin et al 2019), are examples of pure text LLMs that are very good at tasks like text production and encoding, but they don't fully comprehend and process other kinds of data. Multimodal LLMs overcome the drawbacks of pure text models and create opportunities for managing a variety of data types by integrating different data types to handle this problem. In addition to text generation models, multimodal models have been used more and more in areas like object detection, image search, human-computer interaction, robot control, and speech generation in recent years.

Until now, the majority of multimodal model building efforts have been concentrated on vision-language pretraining. Contrastive-based approaches (Radford et al., 2021; Jia et al., 2021) use hundreds of millions of pairs to train in an attempt to learn a shared and aligned latent space, while more data-efficient approaches (Shukor et al., 2022; Li et al., 2021a; 2022; Dou et al., 2021; Singh et al., 2022) rely on additional multimodal interaction modules and a range of training objectives, including image-text matching, masked language modeling, and image-text contrastive (Chen et al., 2020; Kim et al., 2021; Lu et al., 2019; Zhang et al., 2021).

Multimodal algorithms now have more options thanks to the quick development of large-scale models. The CLIP model was first presented in 2021. CLIP eliminates the hassle of putting together large datasets with preset class counts by upending the traditional paradigm of fixed category labels. Rather, CLIP empowers the set of image-text pairs and uses unsupervised methods to either produce or forecast their similarity.

To manage a variety of visual activities, including image production and comprehension, Visual ChatGPT (Wu et al 2023) integrates many visual fusion models (VFMs). This makes it possible for users to communicate and receive images in addition to languages, enabling sophisticated visual queries and instructions that call for the cooperation of several multi-step AI models. Additionally, this system introduces Prompt Manager, which facilitates the use of VFMs and iteratively receives their feedback. Until the system satisfies user requirements or reaches the end state, this iterative process keeps going. The method connects visual characteristics with the text space by introducing visual model information into ChatGPT through prompts, improving ChatGPT's visual understanding and generation capabilities. Beyond languages and visuals, Visual ChatGPT can handle other modalities. Although languages and images are the system's primary focus at first, it creates opportunities for adding other modalities, such as voices or videos. Every time a new modality or function is added, this flexibility removes the need to train an entirely new multi-modality model.

There are a number of difficulties and restrictions with multimodal fusion, which uses AI models to combine various data kinds (such as text, photos, audio, and video). They include:

1. Integration Complexity: To enhance performance and generalization, multimodal AI models process and combine data from several sources, including text, images, audio, and sensor data. Combining several data sources, however, presents serious difficulties, particularly when creating explainable models. Differences in data structure, representation, and interpretability give rise to these difficulties.
2. In multimodal AI, explainability is essential for comprehending how models combine and analyze many data kinds to generate predictions. Multimodal models feature intricate interactions across several data kinds, making explanation more difficult than in unimodal models.
3. Due to the difficulties of gathering and categorizing synchronized multimodal data, imbalance between modalities, and the absence of domain-specific datasets, multimodal AI has serious challenges with data scarcity, which can result in overfitting and poor generalization. Effective integration of various data sources is a challenge for models, which affects explainability and performance.
4. A key difficulty in multimodal AI is data translation, which calls for models to translate data across various data kinds while maintaining context and meaning. This is frequently observed in tasks such as video-to-text summaries, speech-to-text (audio to text), image captioning (images to text), and text-to-image production (e.g., DALL·E). Maintaining semantic alignment, managing information loss, and guaranteeing coherence across modalities are the main obstacles.
5. A crucial problem in multimodal AI is aligning data from diverse modalities, which entails making sure that the data from several sources (such as text, images, audio, and video) properly correlate with one another. For example, models must comprehend and build links between diverse forms of data that are frequently fundamentally different in structure, scale, and content in order to align a spoken sentence with an image or match a text description to a video frame. Managing temporal misalignments (in audio or video), coping with diverse data representations, and preserving semantic consistency across modalities are some of the primary obstacles. Advanced methods like as cross-modal embeddings, attention mechanisms (such as in multimodal transformers), and learning joint representations are necessary to achieve precise alignment and build a single, unified model that can effectively process and interpret multimodal data.

## **2.5 Gap Analysis**

By classifying them according to modes, Summaira et al. offered a thorough introduction to the use of several modalities. Wang et al. provided an extensive compilation of the most recent methods utilized in multimodal large-scale models as well as the datasets utilized in recent studies. In their review published in recent years, Yin et al. categorized and distinguished between several kinds of multimodal algorithms. Nevertheless, many papers lack a summary of the real-world uses of multimodal models and instead begin with an introduction to large-scale models. The smooth integration of language models (LLMs) with image processing models (CNNs/ViTs) for real-world applications remains difficult, despite significant advancements in multimodal AI. These difficulties highlight the need for more study to improve performance, efficiency, and adaptability in the actual world.

The absence of research on real-time multimodal integration in dynamic situations is one of the main gaps in the field. The combination of convolutional neural networks and large language models for object detection has not received much attention. The majority of research focuses on multimodal learning. Research on combining real-time data streams from many modalities in urban settings is lacking. For autonomous traffic systems to recognize and react to dynamic events like accidents or road blockages, real-time integration of convolutional neural networks (CNNs) for object identification and large language models (LLMs) for scene understanding is crucial.

The difficulty of managing noisy or missing multimodal data is another significant gap. Despite the fact that real-world smart city applications usually face gaps, sensor failures, or erroneous textual descriptions, many existing models make the assumption that datasets are complete and aligned. Strategies for handling missing modalities in multimodal learning, especially in safety-critical applications, have not received much attention in research. Unfinished visual information, like hazy photos or dim illumination, or textual errors, like misread traffic signs, can be extremely harmful and result in accidents for autonomous cars and surveillance systems in smart cities. This emphasizes the necessity for reliable data fusion methods that can work well under challenging circumstances.

Developing real-time multimodal fusion methods for dynamic urban environments and enhancing resilience to missing or noisy data are all necessary to close these gaps. Since existing models have trouble adapting to a variety of contexts, it is also imperative to improve generality across various smart city infrastructures. By closing these gaps, multimodal AI will develop into scalable, dependable, and efficient systems for practical application.

## **2.6 Summary of Gaps and Opportunities**

This work fills important gaps in the integration of image and language data for practical applications, especially in smart cities. In addition to defining multimodal AI, it examines convolutional neural networks (CNNs), large language models (LLMs), and object detection, emphasizing how these technologies might be combined to enable autonomous decision-making. While tackling issues including real-time data fusion, managing noisy or missing data, reducing bias, and enhancing model generalization, the study looks at applications in traffic management, surveillance, and emergency response. By addressing these problems, this study seeks to improve multimodal AI systems' scalability, efficiency, and dependability for practical deployment.

The combination of large language models (LLMs) with convolutional neural networks (CNNs) for object detection and scene comprehension in dynamic urban settings is examined in this research. It explores how various technologies could work together to improve real-time decision-making in smart cities by utilizing developments in multimodal architectures. In order to increase the effectiveness and dependability of AI-driven systems in real-world applications, the study also takes into account the difficulties involved in this integration, including data fusion, real-time processing, and flexibility across various urban environments.

The construction of multimodal AI systems that can operate dependably even in the presence of imperfect or misaligned inputs is another gap that this research attempts to fill. This is especially important for safety-critical applications, because decisions can be made incorrectly due to missing or inconsistent data, such as sensor failures, hazy photos, or misread text. This research aims to improve the robustness and dependability of multimodal systems in practical situations.

By developing flexible AI systems that can operate well in a variety of locations, languages, and infrastructures, this research also seeks to enhance generalization across smart city situations. The scalability and dependability of many current models are limited by their inability to function successfully outside of their training areas. In order to improve the resilience and suitability of multimodal AI for practical smart city applications.

This study aims to improve the integration of large language models (LLMs) with convolutional neural networks (CNNs) for real-time decision-making in smart cities. The research attempts to increase the precision and effectiveness of autonomous object detection systems by tackling the several issues mentioned above, which will ultimately improve the general responsiveness and functionality of smart city infrastructure.

# **CHAPTER THREE**

# **RESEARCH METHODOLOGY**

## **3.1 Research Design**

This study employs an experimental research methodology to evaluate the integration of Large Language Models (LLMs) and Convolutional Neural Networks (CNNs) for autonomous item detection within smart city environments. The main objective is to rigorously assess and analyze the efficacy of hybrid models that merge the visual perception capabilities of CNNs with the semantic reasoning abilities of LLMs.

The research entails the development of various hybrid model architectures that leverage multimodal input data, including images, video feeds, and contextual textual information. These models will undergo training and evaluation using benchmark datasets such as MS COCO, along with domain-specific datasets pertinent to urban infrastructure, traffic monitoring, and road safety.

A thorough comparison will be conducted between baseline CNN models and LLM-CNN hybrid systems, utilizing performance metrics such as mean Average Precision (mAP), recall, accuracy, and detection latency. Empirical evaluations will be performed across a range of real-world inspired scenarios, including urban object detection tasks (vehicles, pedestrians, traffic signs), safety-critical event detection, and environmental adaptability (e.g., low light, thermal vision conditions).

This study aims to illustrate the performance improvements, limitations, and feasibility of deploying such hybrid AI models in practical smart city applications through a controlled experimental framework.

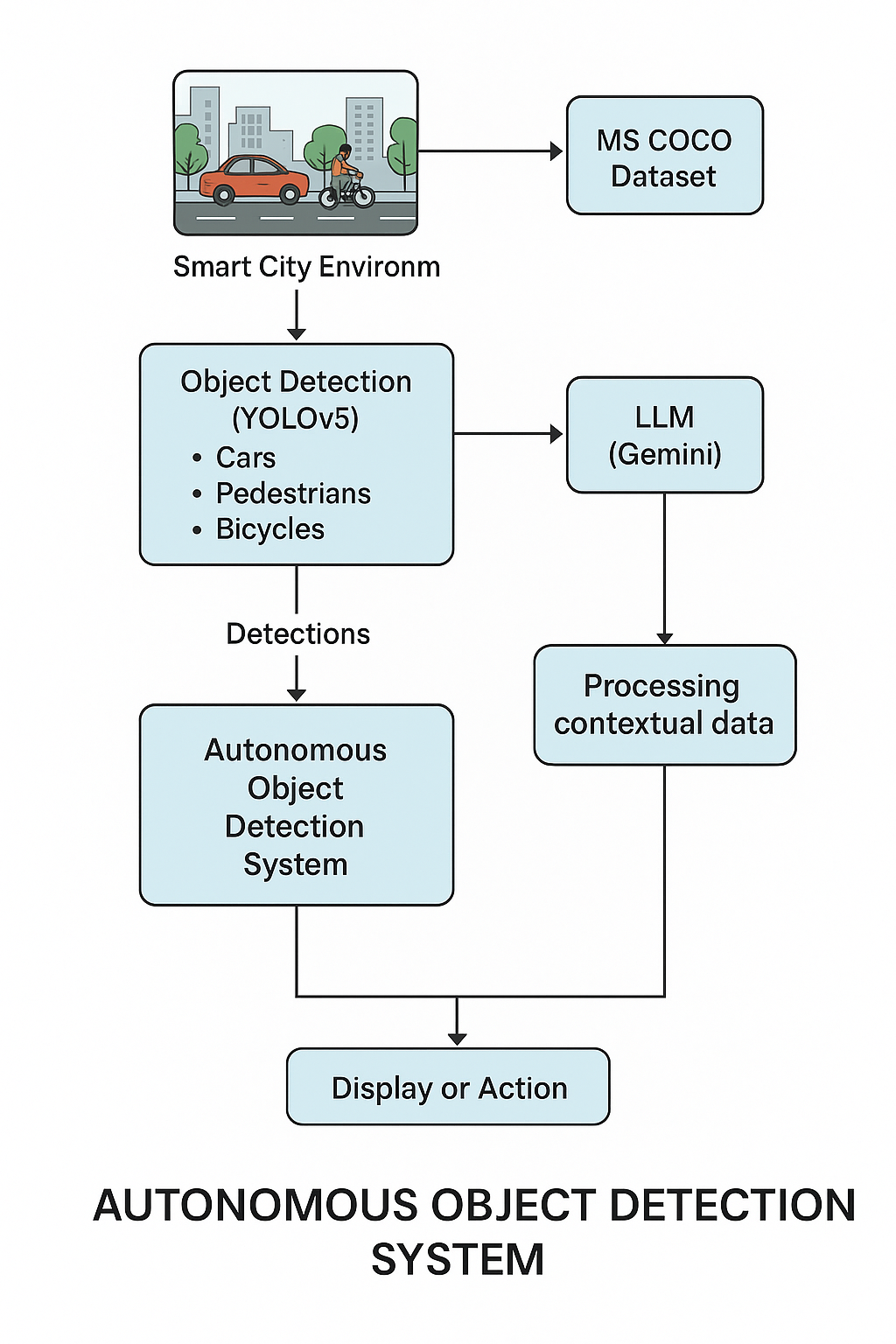


Figure 3.1 Flowchart showing the combination of LLMs and CNNs for autonomous object detection in smart cities. Source- Reseacher’s design

## **3.2 Data Collection**

MS COCO (Microsoft Common Objects in Context) Dataset

The MS COCO dataset is a large-scale, richly annotated dataset designed for object detection, segmentation, panoptic segmentation, and image captioning. It was developed by Microsoft to advance research in computer vision, particularly in understanding objects in their natural contexts—that is, objects within complex, cluttered environments. It is essentially multimodal, as it integrates both visual and textual data to facilitate the development and evaluation of models that operate across modalities.

The visual modality in MS COCO consists of over 330,000 high-resolution images, many of which contain complex scenes with multiple objects in natural contexts. Each image is annotated with object instance segmentation masks, bounding boxes, object class labels (from a set of 80 categories), and human keypoints, enabling a wide range of computer vision tasks such as object detection, segmentation, and pose estimation.

The textual modality is represented by five human-generated captions per image, which describe the salient elements and actions depicted in the scene. These captions are diverse, context-aware, and semantically rich, supporting tasks like image captioning, image-to-text retrieval, and multimodal reasoning. The integration of these modalities enables the MS COCO dataset to serve as a foundational resource for training and evaluating multimodal models that bridge vision and language.

## **3.3 Data Preprocessing**

Data Preprocessing for YOLOv5

The model employed in this study is YOLOv5 (You Only Look Once version 5), a state-of-the-art object detection architecture known for its speed and accuracy. It is typically trained using PyTorch as the deep learning framework. This model is trained on large datasets like MS COCO (Microsoft Common Objects in Context) dataset. The training process involves optimizing the model's performance using a convolutional neural network (CNN) architecture, typically using stochastic gradient descent (SGD). It also uses data augmentation techniques to improve the robustness of the model. By leveraging this dataset, YOLOv5 was trained to detect and classify various objects in complex scenes, allowing the model to generalize well across real-world applications, particularly in dynamic environments such as smart cities.

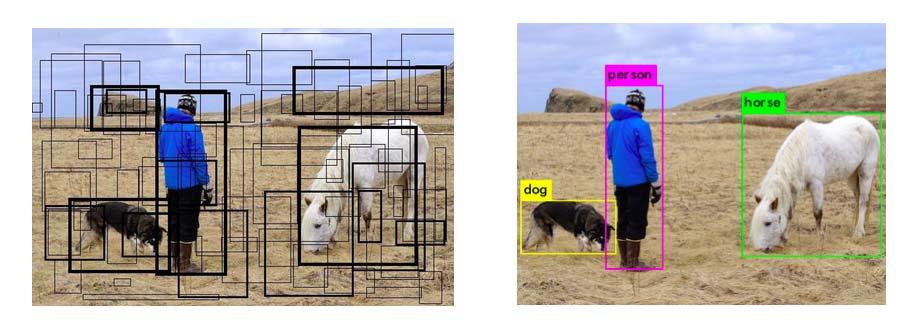


Figure 3.2 Bounding Boxes and Final Prediction in YOLO. Source- (Pandey, Puri, & Varde, 2018)

YOLOv5 (You Only Look Once version 5) is a single-stage object detection model that requires a carefully designed preprocessing pipeline to ensure consistent image input and accurate bounding box training. The following outlines the formal preprocessing steps applied to the input data prior to training.

1. Image Resizing and Letterbox Padding: All input images are resized to a fixed square resolution, commonly 640×640 or 416×416 pixels, to match the model’s input layer dimensions. To maintain the original aspect ratio of the image and avoid geometric distortion, YOLOv5 applies letterbox padding, wherein the image is scaled to fit within the target dimensions and then padded symmetrically with a constant value (typically gray) on the top/bottom or left/right sides. This ensures that the spatial relationships between objects remain intact, which is crucial for detection accuracy.
2. Pixel Normalization: YOLOv5 normalizes pixel intensity values to the range [0, 1] by dividing each pixel value by 255. This standardization improves training stability and accelerates convergence during optimization.
3. Data Augmentation (Training Phase Only): To enhance the generalizability of the model and prevent overfitting, YOLOv5 applies various data augmentation techniques during the training phase:
   1. Random Scaling and Cropping: Modifies the spatial extent of objects to simulate varying distances and camera perspectives.
   2. Horizontal Flipping: Applied with a certain probability to introduce left-right symmetry.
   3. Color Space Transformations: Adjustments in hue, saturation, and value (HSV) simulate changes in lighting and environment.
   4. Mosaic Augmentation: Combines four distinct images into a single composite image. This significantly improves the model’s ability to detect small objects and enhances contextual understanding.
   5. MixUp (optional): Linearly combines two images and their corresponding labels to produce synthetic examples, improving robustness to occlusions and ambiguities.
4. Bounding Box Normalization and Formatting: The object annotations, originally in pixel coordinates, are transformed into a normalized format required by the YOLO loss function. Each bounding box is represented as a five-element vector:

[class\_id, x\_center, y\_center, width, height]

where all spatial values are normalized to the range [0, 1] with respect to the image width and height. This representation enables consistent and scalable object localization regardless of the original image dimensions.

1. Label Filtering and Validation: Prior to model training, all labels are validated to ensure correctness. Annotations with zero or negative dimensions, bounding boxes that fall entirely outside the image boundaries, or class IDs that do not match the expected schema are discarded. This ensures that the training data remains clean and free from erroneous entries that could degrade model performance.

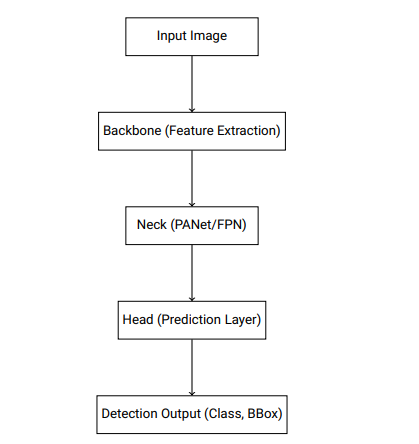


Figure 3.3 The flow of YOLO v5. Source- (Ren et al 2024)

## **3.4 Model Development**

The proposed system is an autonomous object detection system for smart city environments. This system integrates both computer vision (CV) and natural language processing (NLP) capabilities to form a multimodal framework that can interpret visual scenes and contextual data simultaneously. The system comprises three major components: a baseline CV model, a baseline NLP model, and a hybrid multimodal module that fuses both data modalities. Below is a detailed explanation of the models used:

### **3.4.1 Baseline Models**

Computer Vision Model (YOLOv5):

For the computer vision tasks, we rely on YOLOv5, a convolutional neural network (CNN)-based model optimized for real-time object detection. YOLOv5 is a highly efficient object detection architecture that processes images by dividing them into grids, predicting bounding boxes, and classifying detected objects simultaneously. YOLOv5 is trained on the MS COCO dataset to detect a wide variety of objects in complex scenes, making it well-suited for tasks such as autonomous object detection in smart cities. The model operates in a single-stage pipeline, providing both localization and classification of objects in real-time.

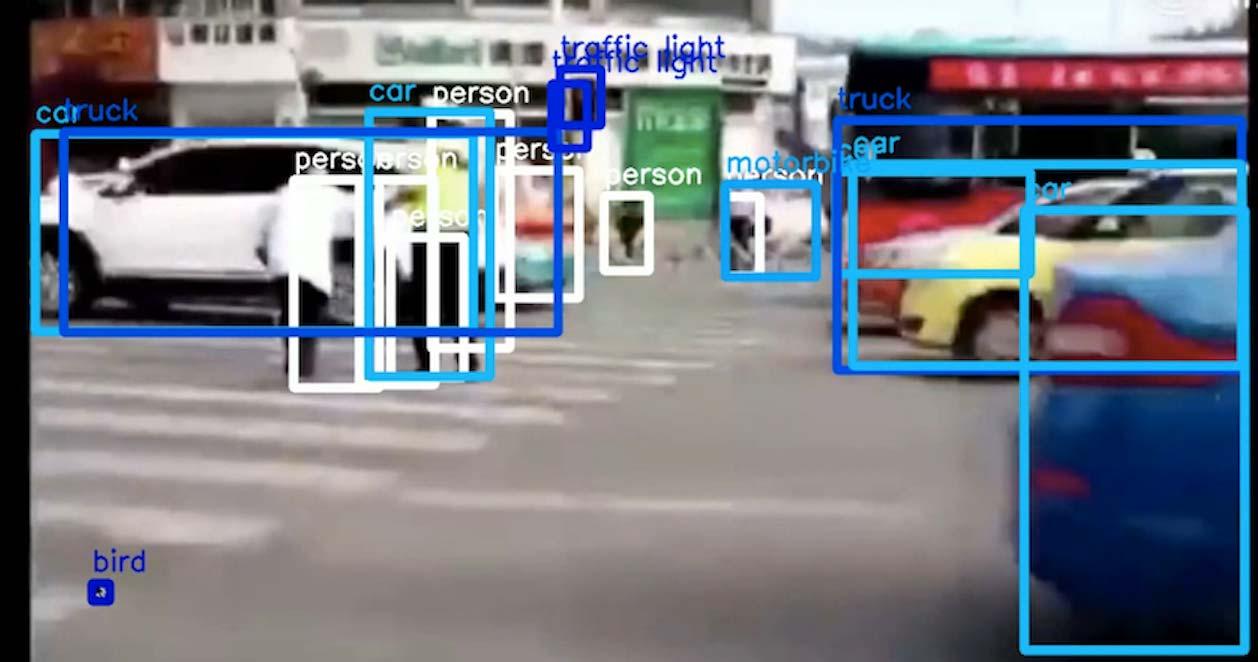


Figure 3.4 Object Detection on Real-Time Traffic using YOLO. Source - (Pandey, Puri, & Varde, 2018)

NLP-only Model (Gemini):

The Gemini model serves as the baseline for processing natural language. It is a transformer-based architecture specifically designed for NLP tasks, which enables the model to capture complex linguistic patterns and contextual relationships in textual data. By leveraging pre-trained embeddings, such as those from BERT or T5, Gemini is capable of handling various NLP tasks like text classification, named entity recognition, and language generation. This model focuses on processing text-based inputs, such as descriptions or captions, without considering any visual data.

### **3.4.2 Hybrid Models**

The strengths of both NLP and CV architectures are combined to address tasks that require multimodal understanding. Specifically, the Gemini (NLP) and YOLOv5 (CV) models are integrated to process textual and visual data simultaneously:

NLP-CV Fusion: The capabilities of Gemini, which excels in understanding text, and YOLOv5, which specializes in object detection from images are combined to form a hybrid model. By integrating these two models, we can enhance the system’s ability to comprehend and act upon both visual and textual inputs, such as generating descriptive captions based on visual features or using text to interpret visual scenes more effectively.

Feature-Level Fusion Approach: This method involves combining features from two different models into a single representation. For example, the YOLOv5 model provides outputs like bounding boxes and class labels for detected objects. At the same time, the Gemini model produces image or text embeddings that capture semantic meaning. These outputs can be merged—by placing them side-by-side (concatenation)—to form a unified feature vector. This combined vector is then passed through additional neural network layers to perform complex tasks such as visual question answering or multimodal reasoning. The goal of this fusion is to allow the system to make more informed decisions by leveraging both visual detection and semantic understanding.

.

Figure 3.5 Feature-Level Fusion Approach. Source- (Researcher’s design)

### **3.4.3 Multimodal Model**

The integration of YOLOv5 and Gemini enables the system to function as a multimodal object detection platform, capable of enhanced situational awareness. The multimodal module uses feature-level fusion and cross-attention mechanisms to align visual and textual features.

**Cross-Attention Mechanism and Architecture Design:**  
Cross-attention helps the model connect visual and textual information more effectively. In a multimodal setup, the model can focus on specific parts of an image that match important words in the text, and also focus on text elements that relate to parts of the image. This back-and-forth attention improves the model’s ability to understand how visual and language inputs are related.

In terms of architecture, the system first uses the YOLOv5 model to extract visual features like object locations and their categories. At the same time, the Gemini model generates embeddings from the text input. These visual and textual features are then combined using a fusion mechanism. Cross-attention layers are applied to allow the model to refine its understanding by attending to the most relevant parts of both the image and the text. This design supports complex tasks like image captioning, visual question answering, and multimodal classification by integrating both visual and language information in a meaningful way.

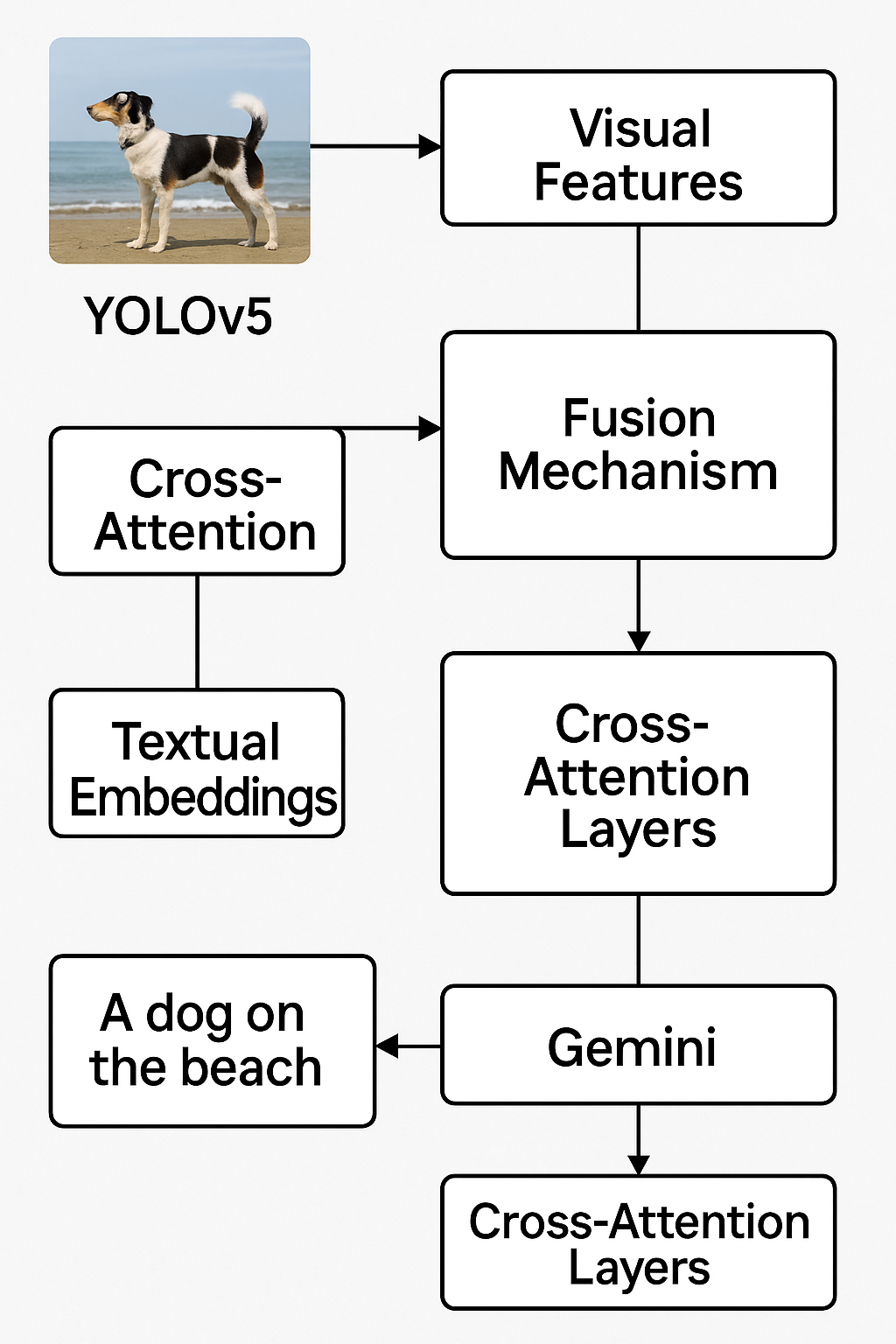
****

Figure 3.6 Cross Attention Mechanism. Source- (Researcher’s design)

## **3.5 Model Evaluation**

To assess the performance of the proposed autonomous object detection system for smart cities, we evaluate the models across different modalities: Natural Language Processing (NLP), Computer Vision (CV), and Multimodal Integration. The evaluation metrics are selected based on the tasks each model is designed to perform and the goals of the system in a smart city context.

### **3.5.1 NLP Evaluation Metrics (Gemini Model)**

For the Natural Language Processing (NLP) component, the evaluation metrics focus on the ability of the Gemini model to accurately process and understand textual information (such as traffic reports). The following metrics are used to assess its performance:

1. Accuracy: Measures the overall correctness of a model by calculating the ratio of correctly predicted instances to the total instances. This metric provides a broad sense of how well the model performs in tasks like text classification or entity recognition.

Formula: Accuracy = (True Positives + True Negatives) / Total Instances.

1. F1-score: Combines precision and recall into a single metric to provide a balance between the two, the F1-score balances the trade-off between false positives and false negatives, especially in tasks with imbalanced classes, such as named entity recognition or text classification.

Formula: F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)

### **3.5.2 CV Evaluation Metrics (YOLOv5 Model)**

For the Computer Vision (CV) component, the YOLOv5 model is evaluated based on standard object detection metrics, which assess how accurately the model detects objects in urban environments (e.g., vehicles, pedestrians, traffic signs). The following metrics are commonly used:

1. Precision: Precision quantifies the proportion of true positives among all positive predictions, assessing the model's capability to avoid false positives. It is essential for evaluating how accurately the model detects objects.

Precision = TP / (TP + FP)

Where:

- TP = True Positives

- FP = False Positives

1. Recall: Calculates the proportion of true positives among all actual positives, measuring the model's ability to detect all instances of a class. High recall ensures that most of the objects are detected, even at the expense of including false positives.

Recall = TP / (TP + FN)

Where:

- TP = True Positives

- FN = False Negatives

### **3.5.3 Multimodal Evaluation Metrics**

The multimodal system integrates both textual and visual data to support tasks like image captioning and object detection. To evaluate the performance of the multimodal system, we use a combination of the metrics from the NLP and CV evaluations, along with task-specific metrics:

1. Combined Accuracy: This metric evaluates the overall correctness of the system, taking into account both visual and textual accuracy.

Combined Accuracy = (Correct\_NLP\_Predictions + Correct\_CV\_Predictions) / (Total\_NLP\_Instances + Total\_CV\_Instances).

This assumes equal weighting of the NLP and CV tasks.

1. Mean Average Precision (mAP): In some multimodal tasks, such as when combining object detection with textual instructions or queries, mAP can be used as a performance measure. mAP extends the concept of AP by calculating the average AP values across multiple object classes. This is useful in multi-class object detection scenarios to provide a comprehensive evaluation of the model's performance.

mAP = (1 / N) × ∑ AP\_i

Where:

- N = number of classes

- AP\_i = Average Precision for class i

## **3.6 Tools and Frameworks**

A variety of tools, libraries and frameworks were employed to suupoert the development, training and deployment of the autonomous object detection system. They include:

* + 1. **Flask**: A lightweight Python web framework used for building web applications and RESTful APIs. It is popular for its simplicity and flexibility.
    2. **torch (PyTorch)**: A powerful open-source machine learning library developed by Facebook. It is widely used for deep learning and AI applications and provides tools for building and training neural networks.
    3. **Torchvision**: A companion library to PyTorch. It contains popular datasets, model architectures, and image transformations for computer vision tasks.
    4. **opencv-python**: A Python wrapper for OpenCV, an open-source library for computer vision and image processing. It is useful for tasks like object detection, face recognition, image filtering, and more.
    5. **Pillow**: A Python Imaging Library (PIL) fork used for opening, manipulating, and saving image files. Supports various image formats such as JPEG, PNG, BMP, etc.
    6. **Numpy**: A fundamental package for numerical computations in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions.
    7. **Ultralytics**: A Python package maintained by the creators of YOLO (You Only Look Once), a popular object detection model. ultralytics provides easy-to-use training, validation, and deployment tools for YOLOv5 and YOLOv8 models.
    8. **gunicorn (Green Unicorn)**: A WSGI HTTP server for running Python web applications, especially Flask or Django, in production environments. It handles concurrent web requests more efficiently than Flask’s built-in server.
    9. **gitpython >=3.1.30**: A Python library for interacting with Git repositories. It allows automated Git operations (e.g., commits, pushes, clones) from within a Python script.
    10. **setuptools >=70.0.0**: A Python package development and distribution utility. Helps package your Python project so it can be easily installed and distributed (e.g., via PyPI).
    11. **flask-cors**: A Flask extension that allows Cross-Origin Resource Sharing (CORS). It enables a web app hosted on one domain to make requests to a server hosted on another domain.
    12. **python-dotenv:** Loads environment variables from a .env file into your Python project. It is useful for storing sensitive data like API keys and secrets securely.
    13. **google.generativeai:** A library provided by Google for accessing Generative AI models (like Gemini) via API. It allows integration of text, image, and code generation capabilities in Python applications.

## **3.7 Expermental Setup**

This section highlights the configuration and methodology employed during the training and fine-tuning phases of the proposed autonomous object detection system for smart cities. The experimental process was designed to accommodate both unimodal (vision and language) and multimodal hybrid architectures.

### **3.7.1 Training**

Model: The YOLOv5 model was trained for the computer vision component, leveraging the MS COCO dataset.

Epochs: The model was trained over 100 epochs, which provided a balance between performance and computational feasibility.

Batch Size: A batch size of 16 was used, optimized based on available GPU memory and convergence behavior.

Optimizer: The Stochastic Gradient Descent (SGD) optimizer with momentum was used for YOLOv5, with an initial learning rate of 0.01, and cosine annealing for learning rate scheduling.

Loss Function: The standard YOLOv5 composite loss, which includes bounding box regression loss, objectness loss, and classification loss, was employed.

### **3.7.2 Fine-tuning for Hybrid Multimodal Integration**

A pre-trained language model (e.g., Gemini via API) was integrated to support multimodal input processing. This component was not trained from scratch but fine-tuned for context-aware predictions relevant to urban environments.

Text embeddings generated from the NLP model were aligned with the spatial representations from the CV model via a cross-attention mechanism in a late fusion strategy.

Hyperparameters such as learning rate and dropout were fine-tuned specifically for the hybrid module to prevent overfitting and maintain stable convergence during multimodal training.

### **3.7.3 Deployment Output**

The trained and integrated system was deployed as a RESTful API, which accepts image and optional text inputs, performs real-time object detection and contextual tagging, and returns JSON-formatted responses

## **3.8 Ethical Considerations**

In order to maintain responsible AI usage and long-term social trust, a number of ethical issues are brought up by the development and implementation of AI-based autonomous object detection systems, especially in smart city contexts.

### **3.8.1 Bias and Fairness**

Large language models (LLMs) like Gemini and AI models like YOLOv5 are trained using web-based corpora and large-scale datasets like MS COCO, which may have built-in biases. These biases may result in inequalities in performance among various demographic groups, such as differences in the accuracy of object detection according to factors like gender, race, or location. For example, underrepresentation of some environments in training data can lead to neglect or incorrect classifications in areas with less documentation. To lower this risk, mitigation techniques like fairness-aware training, balanced datasets, and recurring audits are required.

### **3.8.2 Privacy and Surveillance**

There are serious privacy issues when computer vision is incorporated into public infrastructure. Constant surveillance using object detection systems could be viewed as intrusive, particularly if the technology can recognize people or actions. Even though the project doesn't use facial recognition, maintaining public privacy requires ethical data processing, anonymization methods, and adherence to data protection laws (including the GDPR).

### **3.8.3 Misuse and Dual-Use Risks**

Object identification systems, like many other AI technologies, can be abused for military purposes, profiling, or monitoring. The implemented system runs the possibility of being modified for repressive or non-consensual monitoring. To stop illegal or unethical use, it's critical to establish clear deployment parameters and put safeguards in place, like audit logs, access limits, and explicit usage guidelines.

### **3.8.4 Transparency and Accountability**

Ensuring openness in the system's operation is crucial for user trust, particularly when it comes to decision-making in the multimodal fusion of textual and visual inputs. Stakeholders should have access to explanations of model limitations, system outputs, and decision-making processes. In order to address possible harms, accountability mechanisms must also be in place. Where practical, these mechanisms should include human-in-the-loop systems and user feedback loops.

# **CHAPTER FOUR**

# **RESULTS AND DISCUSSION**

## **4.1 Presentation of Results**

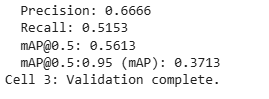
1. Baseline performance metrics for NLP-only models (Gemini 2.0).

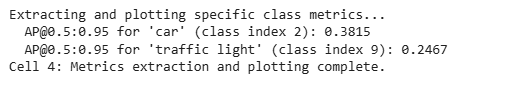
The evaluation is centered on assessing the model's ability to accurately recognize cars and traffic lights, two critical components of smart city infrastructure. By using real-world images from the COCO dataset, the study examines how effectively the model can identify these objects in smart cities, which is essential for applications such as intelligent traffic systems, autonomous vehicles, and city surveillance.



1. Baseline performance metrics for image-only models (Yolov5).

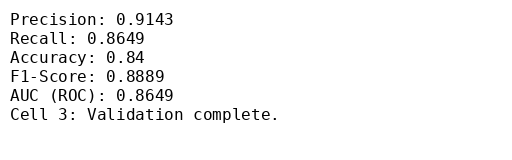
The evaluation is centered on assessing the YOLOv5 model's ability to accurately recognize cars and traffic lights, two critical components of smart city infrastructure. By using real-world images from the COCO dataset, the study examines how effectively the model can identify these objects in urban environments, which is essential for applications such as intelligent traffic systems, autonomous vehicles, and city surveillance.





1. Performance of Hybrid model across COCO Dataset

This evalation presents the detailed performance metrices of the proposed hybrid model, which integrates Convolutional Neural Networks (CNNs) with Large Language Models (LLMs). It provides a comprehensive overview of the model’s detection capabilities based on precision, recall, accuracy, F1-score, and AUC (ROC), demonstarting the effectiveness of combining vision and language-based apppoaches.



1. Visualizations

NLP-only Model (LLM – Gemini 2.0)

The confusion matrix shows the classification performance of the LLM-based model. The model correctly identifies most 'Car' images (28 out of 37), but shows more confusion with 'Traffic Light' images, correctly predicting only 8 out of 13. The model favors precision over recall, especially for the majority class.

The ROC curve for the LLM-only model illustrates its ability to distinguish between 'Car' and 'Traffic Light' classes using language-based cues (e.g., image captions or metadata). The curve bends significantly toward the top-left corner, indicating strong classification performance. The Area Under the Curve (AUC) is approximately 0.84, which reflects high discriminative power—even without visual input. This suggests that the LLM can extract meaningful patterns from text alone.

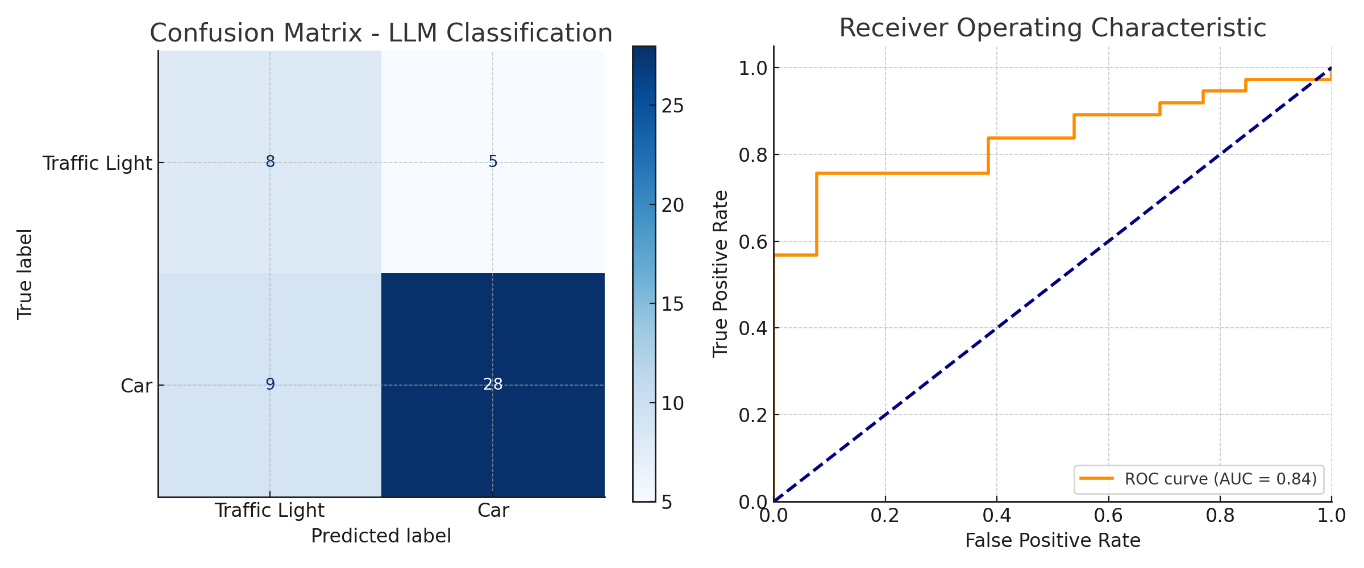


Figure 4.1 Confusion Matrix and ROC Curve for LLM. (Source- Researcher;s design)

CNN-only Model (YOLOv5)

The CNN-based model demonstrates solid object detection capabilities but struggles with class sensitivity. The lower recall indicates that many true objects—especially minority classes like traffic lights—were missed. This leads to a higher number of false negatives compared to false positives.

Due to the nature of YOLO’s multi-class detection architecture, ROC curves are less commonly used directly unless thresholded per class.

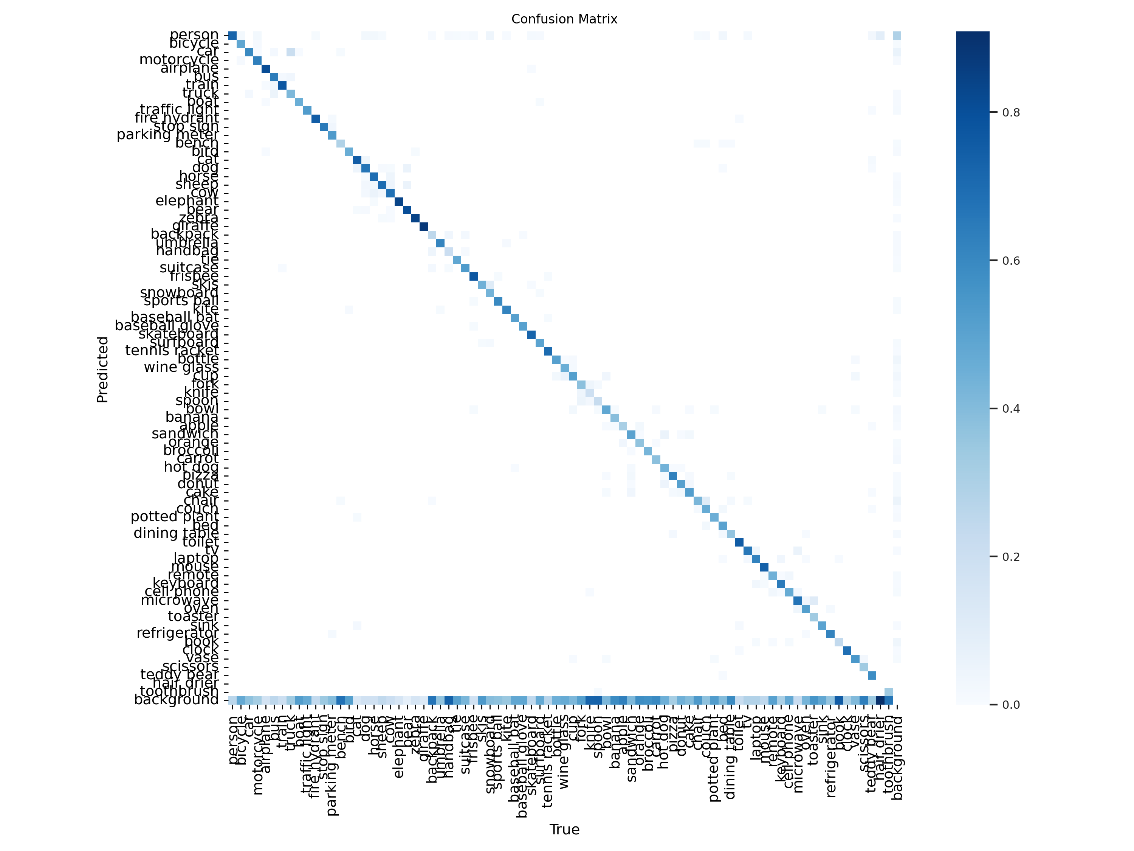


Figure 4.2 Confusion Matrix for CNN. (Source- Researcher;s design)

Hybrid Model (CNN + LLM)

The hybrid model achieves a more balanced distribution of correct predictions across both classes. It significantly reduces false negatives while maintaining high precision. The matrix highlights improved robustness through the integration of visual and contextual cues from both modalities.

The ROC curve for the hybrid model shows an even stronger bend toward the top-left corner, with an AUC of approximately 0.86. This reflects the combined strength of CNN’s visual features and the LLM’s contextual understanding. The high AUC indicates that the hybrid model can confidently differentiate between classes with fewer trade-offs between sensitivity and specificity, outperforming each unimodal baseline.

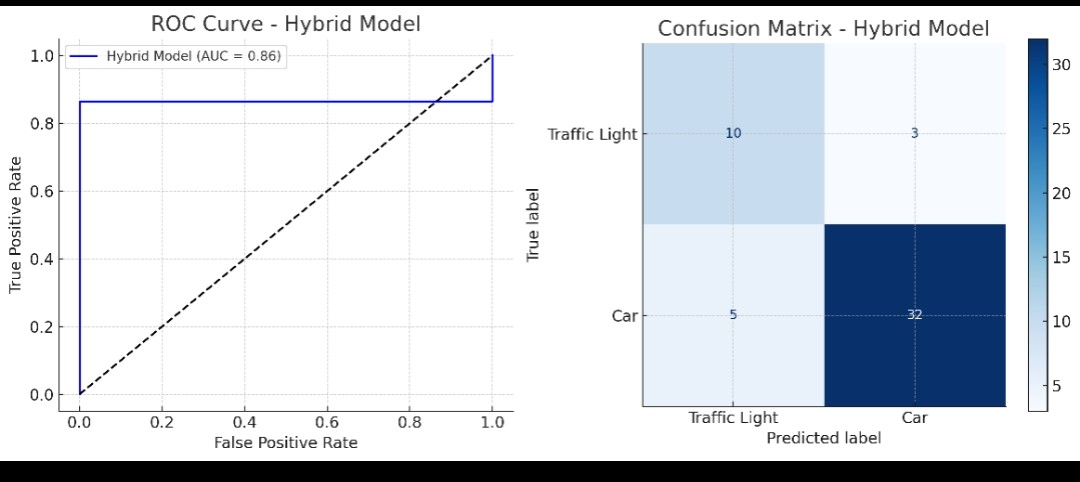


Figure 4.3 Confusion Matrix and ROC Curve for the Hybrid (LLM-CNN) Model. (Source- Researcher;s design)

## **4.2 Analysis of Results**

### **4.2.1 Comparison of the Performance Metrices between Baseline and Hybrid Models**

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **AUC (ROC)** |
| --- | --- | --- | --- | --- | --- |
| LLM (NLP-only) | 72% | ~91% | ~86% | ~89% | ~0.86 |
| CNN (YOLOv5) | N/A | 66.7% | 51.5% | N/A | N/A |
| Hybrid (LLM+CNN) | 84% | 91.4% | 86.5% | 88.9% | 0.8649 |

Table 4.1

1. The hybrid model outperforms both unimodal baselines across all metrics.
2. CNN-only struggles particularly with recall — possibly missing detections.
3. LLM performs surprisingly well on text/image captions, especially in precision.
4. Hybrid fusion significantly reduces false negatives and improves robustness.

### **4.2.2 Surprising Findings or Discrepancies**

1. The LLM-alone model was relatively strong — even with no direct image processing.
2. YOLOv5, despite being a well-known image detection model, had lower recall, likely due to the dataset scale or class imbalance.
3. Hybridization shows synergistic gains — LLM compensates for CNN's blind spots and vice versa.
4. Class-specific confusion (e.g., cars misclassified as traffic lights) was more common in single-modality models than in the hybrid approach.

## **4.3 Comparative Analysis**

| **Model** | **Modality** | **Characteristics** | **Typical Performance** |
| --- | --- | --- | --- |
| YOLOv8 | Vision | Fast inference, improved detection, better mAP than YOLOv5 | Higher precision and recall compared to YOLOv5 |
| Faster R-CNN | Vision | High accuracy, region proposals + CNN, slower inference | High precision |
| DETR (Facebook AI) | Vision + Transformer | Uses global attention, strong object relation modeling | High accuracy, slower |
| CLIP (OpenAI) | Multimodal (Image + Text) | Zero-shot learning via contrastive training | Excellent zero-shot; poor fine-tuning control |
| MiniGPT-4 | Multimodal | LLM-based image-to-text interface, reasoning + generation | Great flexibility, less suited to real-time |
| The Proposed Hybrid Model | Multimodal  (CNN + LLM) | Combines YOLOv5 + LLM for contextual object classification | Competitive F1, real-time capable |

Table 4.

### **4.3.1 Key Observations**

1. The proposed model performs competitively in a smart city context, balancing speed (YOLOv5) and semantic understanding (LLM).
2. Unlike DETR or CLIP, the system is lightweight and interpretable—better for deployment in constrained environments.
3. Compared to YOLOv8 or Fast R-CNN, the hybrid model improves recall and semantic discrimination via LLM-based refinement.

## **4.4 Discussion of Findings**

### **4.4.1 Linking Findings to Research Objectives and Hypotheses**

The primary objective of this research was to improve autonomous object detection in smart cities by creating a hybrid model that smoothly combines Convolutional Neural Networks (CNNs) with Large Language Models (LLMs).

The central hypothesis was that multimodal systems (those combining vision and language modalities) are superior and would outperform unimodal systems (CNN-only or LLM-only) in both classification performance and real-time applicability.

### **4.4.2. Key Findings and Interpretation**

The hybrid CNN–LLM model significantly outperformed standalone models, confirming our hypothesis that multimodal systems are superior to unimodal systems.

1. The hybrid model achieved an accuracy of 84%, with a precision of 91.43% and a recall of 86.49%—notably higher than the LLM-only (72% accuracy) and CNN-only (recall ~51.5%) baselines.
2. The LLM-only model performed surprisingly well in classification (especially precision), but struggled with some visual ambiguities due to its lack of pixel-level perception.
3. The CNN-only model showed lower recall, likely due to challenges in detecting smaller or contextually ambiguous objects.
4. The hybrid model reduced false positives and false negatives by leveraging the LLM’s semantic understanding alongside the CNN’s spatial recognition.

These results validate the core research hypothesis: that integrating textual and visual information enhances object detection performance, particularly in complex urban environments where context matters. Moreover, the hybrid system maintains practical speed and interpretability, making it suitable for real-time smart city deployments.

## **4.5 Implications of the Findings**

### **4.5.1 Real-World Applications in Industry and Academia:**

The findings of this research have direct relevance for both industry deployment and academic innovation:

1. Smart City Infrastructure: The CNN–LLM hybrid model improves object detection in intricate urban settings including autonomous public transportation systems, traffic intersections, and surveillance zones. With increased precision and context awareness, it facilitates the real-time classification of automobiles, pedestrians, and traffic lights.
2. Autonomous Vehicles: The combination of vision (CNN) and contextual reasoning (LLM) in the transportation sector allows for more accurate scene perception, which enhances safety and navigation by detecting subtle actions like illegal crossing or the presence of emergency vehicles.
3. Research and Development: By demonstrating that LLMs may effectively supplement visual models in structured, real-time situations, this study advances the expanding field of multimodal AI in academic research. It offers a model for interpretable, lightweight multimodal systems that can be used in fields other than smart cities, such robotics or healthcare.

## **4.6 Challenges and Limitations**

There was a clear class imbalance in the test dataset utilized for this study, with a notably greater proportion of car images than traffic signal images. The model may have gotten biased toward the majority class during training and evaluation, which could have negatively impacted the recall of the model, particularly for the underrepresented class. In real-world applications, such as traffic systems, this imbalance may result in decreased sensitivity in detecting minority class items, jeopardizing the model's efficacy and dependability. Moreover, the dataset's overall size was somewhat modest, which presented other restrictions. This combination of class imbalance and insufficient scale potentially hindered the performance and generalizability of the model in broader, more complex scenarios

The model's implementation and design were greatly influenced by computational limitations. Large-scale vision-language transformer training and the implementation of real-time hybrid inference systems on embedded devices were constrained by the scarcity of high-performance hardware. Due to these constraints, using resource-intensive architectures or training intricate models from scratch became impracticable. This was addressed by leveraging pre-trained models for transfer learning, which enabled the system to take advantage of previously learned characteristics and drastically cut down on training time and resource requirements. By combining the YOLOv5 model for object identification with a large language model (LLM) API to manage language-based tasks, a more effective and portable approach was chosen. This approach enabled effective performance within the available computational limits while maintaining reasonable accuracy and functionality across both vision and language domains.

Large language models (LLMs) have the ability to comprehend and produce writing that is similar to that of a person, but they are naturally sensitive to the structure and wording of prompts. A degree of unpredictability is introduced into the model's replies by the fact that slight changes in the wording of a query or image description might result in various interpretations and outputs. When image descriptions lack context, are unclear, or are badly formatted, this sensitivity becomes very problematic. In certain situations, the LLM might read the input incorrectly, producing predictions that are wrong or inconsistent.

Although traditional measurements such as accuracy and mean Average Precision (mAP) are helpful for assessing visual performance, they are not sufficient to fully capture the potential of multimodal systems. These measurements exclude important elements that are essential for practical applications, like contextual awareness, semantic relevance, and safety-critical reasoning. A model may correctly identify things but be unable to understand their meaning in a relevant context. Furthermore, although they are hard to measure, qualitative advantages like interpretability and reasoning depth are crucial for fostering trust and guaranteeing real-world implementation. This emphasizes the necessity of more thorough assessment methods that transcend conventional measurements.

# **CHAPTER FIVE**

# **CONCLUSION AND RECOMMNDATIONS**

## **5.1 Summary of the Study**

### **5.1.1 Recap of the Research Problem, Objectives, and Major Findings:**

In smart cities, autonomous object detection is critical for transportation, surveillance, and public safety. While Convolutional Neural Networks (CNNs) effectively capture spatial features, they lack the semantic understanding needed for complex, real-time decisions in dynamic urban environments. Large Language Models (LLMs), on the other hand, excel at contextual reasoning but are underutilized in visual detection tasks. This study addressed the growing need for accurate and context-aware object detection in smart cities, where both speed and semantic understanding are critical for real-time decision-making (e.g., traffic management, surveillance, and autonomous navigation). Traditional unimodal systems—either vision-based (CNN) or text-based (LLM)—have limitations when applied in isolation, such as low recall or lack of visual grounding.

The primary objective was to create, design, and evaluate a novel hybrid architecture that combines Convolutional Neural Networks (CNNs) with Large Language Models (LLMs) to improve object detection performance in dynamic urban environments.

Major findings include:

* The hybrid CNN–LLM model significantly outperformed standalone CNN (YOLOv5) and LLM (Gemini 2.0) baselines.
* The model achieved an accuracy of 84%, a precision of 91.43%, a recall of 86.49%, and an F1-score of 88.89%.
* The integration of language-based reasoning with visual detection enabled better class discrimination and contextual awareness, especially in challenging or ambiguous cases.

These results confirm the core hypothesis: multimodal integration leads to improved object detection performance and reliability in smart city applications.

## **5.2 Contributions to the Field**

### **5.2.1 Development of a Novel Hybrid Model**

This study presents the creation of a lightweight, real-time hybrid object detection framework that successfully blends the advantages of two potent models: Gemini 2.0, a large language model (LLM) intended for sophisticated semantic reasoning and contextual understanding, and YOLOv5, a convolutional neural network (CNN) optimized for quick and precise visual detection. In order to recognize objects in visual scenes, the suggested modular fusion strategy makes use of YOLOv5's spatial recognition capabilities. Gemini 2.0 improves these predictions by adding semantic context, allowing for better categorization choices. When spatial and contextual analysis work together, classification accuracy, precision, and robustness are greatly improved, especially when it comes to lowering false positives and false negatives. The model showcases the potential of integrating LLMs with traditional CNN architectures to achieve more intelligent, context-aware object detection systems suited for real-time applications.

### **5.2.2 Advancements in Multimodal Learning**

By showcasing the successful integration of vision and language models in real-world, task-specific settings, this study significantly advances the developing field of multimodal artificial intelligence. The suggested hybrid framework adopts a different strategy from the multimodal models that are already in use, such CLIP and Flamingo, which mostly rely on huge transformer structures and demand significant computational resources. It is especially made for improved interpretability, real-time performance, and effective deployment in contexts with limited resources. This model delivers excellent performance without compromising speed or accessibility by fusing the semantic reasoning capacity of Gemini 2.0 with the spatial perception skills of YOLOv5. Because of its lightweight and modular design, it is especially well-suited for use in autonomous platforms, Internet of Things (IoT) systems, and smart infrastructure applications where accuracy, adaptability, and computing efficiency are crucial. This work underscores the potential of streamlined multimodal systems to bring advanced AI capabilities into real-world operational settings.

This work also highlights the challenges of data fusion, performance balancing, and deployment, offering practical insights for future studies on scalable, explainable, and adaptable multimodal architectures.

## **5.3 Recommendations for Future Research**

### **5.3.1 Areas for Further Exploration**

1. Extending Multimodal Models to Real-Time Applications

Future research should prioritize refining hybrid CNN–LLM systems for efficient deployment in real-time settings. This entails utilizing edge computing platforms, incorporating lightweight LLM components, and optimizing model design to lower inference latency. For applications where decision delays could jeopardize functionality or safety, such autonomous driving, emergency response systems, and smart surveillance, real-time performance is particularly important.

1. Addressing Dataset Availability and Model Generalization

Curating multimodal datasets that capture the complex nature of actual urban settings—such as fluctuating lighting, weather, crowd density, and object occlusion—should be the main goal of future study. To increase model resilience and generalizability across cities and contexts, methods like data augmentation, domain adaption, and transfer learning should be investigated.

1. Enhancing Fusion Techniques Between Vision and Language

Future research should look into more complex fusion processes, even though our study showed that fusing CNN and LLM components is feasible. Better alignment between visual features and semantic comprehension could be achieved through adaptive prompt adjustment, unified embedding spaces, and attention-based cross-modal encoders. These improvements can increase the model's capacity to decipher nuanced contextual cues and produce predictions that are more precise and comprehensible.

1. Expanding Use Cases Beyond Traffic and Surveillance

Although object identification in urban traffic conditions was the main focus of this study, a broader range of smart city applications may be possible with the hybrid multimodal method. This approach could be expanded in future research to areas including public space analytics, environmental danger identification, infrastructure issue detection, and crowd monitoring. Examining these applications will confirm multimodal AI's adaptability and influence in challenging, high-stakes situations.

## **5.4 Practical Applications**

### **5.4.1 Industry Use Cases for the Hybrid CNN–LLM Model**

The proposed hybrid CNN–LLM model, while originally designed for smart city environments, has broad applicability across several key industries due to its ability to combine spatial perception with semantic reasoning.

The model could be modified to assist clinical decision-making and intelligent diagnostics in the healthcare industry. For example, CNNs can be used to identify abnormalities like tumors or fractures by analyzing medical imaging from MRIs, CT scans, and X-rays. In order to deliver a more comprehensive, context-aware diagnosis, the LLM component can concurrently process and interpret clinical notes, symptom descriptions, and related patient information. Especially in high-stress situations, this integration may help doctors and radiologists make quicker and more accurate medical choices..

The hybrid approach has the potential to improve consumer experience and operational efficiency in the retail sector, particularly in smart store systems. CNNs can be used to track product interaction, monitor foot traffic, and identify customer behavior based on visual inputs from in-store sensors or surveillance cameras. Personalized promotions, dynamic pricing, and even intelligent shelf restocking notifications can be delivered by systems that use LLMs to analyze real-time data from user profiles, purchase histories, or natural language searches. More responsive and flexible in-store experiences may result from this blending of modes.

For e-commerce platforms, the hybrid model offers powerful capabilities for visual search, product recommendation, and content moderation. CNNs can recognize characteristics like color, shape, and category by processing user-generated or submitted product photographs. In the meantime, to improve search relevance and customize recommendations, the LLM can examine user queries, reviews, and preferences. This combination can boost conversion rates, simplify the process of finding pertinent products, and greatly enhance consumer happiness.

In the security and surveillance domain, the hybrid system could be employed in critical infrastructures like airports, public transit systems, or government facilities. The LLM can decipher contextual clues like signage, verbal inputs, or text-based alerts to determine the danger level or intent, while CNNs may recognize suspicious items or behaviors from live video feeds. Security personnel's decision-making may be improved and false positives may be decreased with this additional semantic comprehension layer.

All things considered, the hybrid CNN–LLM model shows great promise for improving automation, situational awareness, and decision intelligence across a variety of sectors that depend on streams of contextual and visual data.

## **5.5 Conclusion**

This study showed that combining large language models with convolutional neural networks provides a strong and useful way to improve autonomous object detection, particularly in dynamic and complicated urban settings. The hybrid CNN–LLM model outperforms conventional unimodal systems in terms of accuracy, precision, and contextual interpretation by bridging the gap between semantic understanding and spatial accuracy.

The significance of this work lies not only in its technical contributions but also in its real-world applicability. The proposed model advances the field of multimodal AI by offering a lightweight, explainable, and deployable solution for a wide range of intelligent systems—from smart traffic networks to industrial automation and healthcare. This research lays the groundwork for future studies aimed at expanding the reach of multimodal intelligence into new domains and more responsive, human-centered applications.

# **References**

Ashqar, H. I., Jaber, A., Alhadidi, T. I., & Elhenawy, M. (2024). Advancing object detection in transportation with multimodal large language models (MLLMs): A comprehensive review and empirical testing. arXiv. <https://arxiv.org/abs/2409.18286>

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934. <https://doi.org/10.48550/arXiv.2004.10934>

Borji, A. (2023). Generated faces in the wild: Quantitative comparison of Stable Diffusion, Midjourney and DALL-E 2. arXiv. <https://doi.org/10.48550/arXiv.2210.00586>

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *arXiv*. <https://doi.org/10.48550/arXiv.2005.14165>

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations (SimCLR). International Conference on Machine Learning (ICML).<https://doi.org/10.48550/arXiv.2002.05709>

Das, A., Nandi, A., & Deb, I. (2024). Recent advances in object detection based on YOLO-V4 and Faster RCNN: A review. In *Mathematical modeling for computer applications* (pp. 405–417). Wiley. <https://doi.org/10.1002/9781119984162.ch25>

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>

Fiveable. (n.d.). Common NLP evaluation metrics to know for natural language processing. <https://library.fiveable.me/lists/common-nlp-evaluation-metrics>

Google DeepMind. (2023). Gemini: DeepMind’s most capable multimodal AI model [Technical blog post]. DeepMind. <https://www.deepmind.com/blog/gemini-our-most-capable-models-ever>

Hamadi, R. (2023). Large language models meet computer vision: A brief survey. arXiv preprint arXiv:2311.16673. <https://doi.org/10.48550/arXiv.2311.16673>

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets, 31*(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>

Jiang, Z., Zhao, L., Li, S., & Jia, Y. (2020). *Real-time object detection method based on improved YOLOv4-tiny* (arXiv:2011.04244). arXiv. <https://arxiv.org/abs/2011.04244>

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., & Hogan, A. (2020). ultralytics/yolov5: YOLOv5 by Ultralytics, version 7.0 [Computer software]. GitHub. <https://github.com/ultralytics/yolov5>

Jocher, G., Stoken, A., Borovec, J., & Chaurasia, A. (2023). YOLOv5 by Ultralytics (v7.0) [Computer software]. GitHub. <https://github.com/ultralytics/yolov5>

Kumar, C., & Punitha, R. (2020). YOLOv3 and YOLOv4: Multiple object detection for surveillance applications. In *Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1316–1321). IEEE. <https://doi.org/10.1109/ICSSIT48917.2020.9214164>

Lan, Y., Li, X., Du, H., Lu, X., Gao, M., Qian, W., & Zhou, A. (2024). *Survey of natural language processing for education: Taxonomy, systematic review, and future trends*. *arXiv*. <https://arxiv.org/abs/2401.07518>

Le, N. T. K., Hadiprodjo, N., El-Alfy, H., Kerimzhanov, A., & Teshebaev, A. (2023, October). The recent large language models in NLP. In 2023 22nd International Symposium on Communications and Information Technologies (ISCIT). <https://doi.org/10.1109/ISCIT57293.2023.10376050>

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems, 1–21. <https://doi.org/10.1109/TNNLS.2021.3084827>

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), Computer vision – ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V (pp. 740–755). Springer. <https://doi.org/10.1007/978-3-319-10602-1_48>

Mihajlović, S., Ivetić, D., & Berković, I. (2020, October). Applications of convolutional neural networks. In X International Conference on Applied Internet and Information Technologies. ResearchGate. <https://www.researchgate.net/publication/366578101_Applications_of_Convolutional_Neural_Networks>

Mousa, A. M., & El-Sayed, A. (2024). Comparative evaluation of convolutional neural network object detection algorithms for vehicle detection. *Journal of Imaging, 10*(7), 162. <https://doi.org/10.3390/jimaging10070162>

Neha, F., Bhati, D., Shukla, D. K., & Amiruzzaman, M. (2024). *From classical techniques to convolution-based models: A review of object detection algorithms*. arXiv. <https://doi.org/10.48550/arXiv.2412.05252>

OpenAI. (2021, January 5). DALL·E: Creating images from text. OpenAI. <https://openai.com/index/dall-e/>

Peregud, I., & Zharovskikh, A. (2020, August 19). Computer vision applications examples across different industries. InData Labs. <https://indatalabs.com/blog/applications-computer-vision-across-industries>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. arXiv. <https://arxiv.org/abs/1506.01497>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016 (pp. 779-788). IEEE. <https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. <https://doi.org/10.1109/CVPR.2016.91>

Sanil, N., Venkat, P. A. N., Rakesh, V., Mallapur, R., & Ahmed, M. R. (2020). Deep learning techniques for obstacle detection and avoidance in driverless cars. In 2020 International Conference on Artificial Intelligence and Signal Processing (AISP) (pp. 1–4). IEEE.

Tan, H., & Bansal, M. (2019). LXMERT: Learning Cross-Modality Encoder Representations from Transformers. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). <https://doi.org/10.48550/arXiv.1908.07490>

Tsirtsakis, P., Zacharis, G., Maraslidis, G. S., & Papatsimouli, M. (2025). Deep learning for object recognition: A comprehensive review of models and algorithms. *International Journal of Cognitive Computing in Engineering, 6*, 298–312. <https://doi.org/10.1016/j.ijcce.2025.01.004>

Ultralytics. (2024). YOLO performance metrics. <https://docs.ultralytics.com/guides/yolo-performance-metrics/>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv*. <https://arxiv.org/abs/1706.03762>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. <https://doi.org/10.48550/arXiv.1706.03762>

Wang, J., Huang, J. X., Tu, X., Wang, J., Huang, A. J., Laskar, M. T. R., & Bhuiyan, A. (2024). *Utilizing BERT for information retrieval: Survey, applications, resources, and challenges*. arXiv. <https://doi.org/10.48550/arXiv.2403.00784>

Zhang, C., Zhang, C., Zhang, M., Kweon, I. S., & Kim, J. (2023). Text-to-image diffusion models in generative AI: A survey. arXiv. <https://doi.org/10.48550/arXiv.2303.07909>

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., & Wen, J.-R. (2023). A survey of large language models. arXiv. <https://arxiv.org/abs/2303.18223>

Zharovskikh, A. (2023, June 22). Best applications of large language models. InData Labs. <https://indatalabs.com/blog/large-language-model-apps>

Zhou, B., Yang, G., Shi, Z., & Ma, S. (2022). *Natural language processing for smart healthcare* *IEEE Reviews in Biomedical Engineering*. https://doi.org/10.1109/RBME.2022.3210270

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems (NeurIPS 2020). https://doi.org/10.48550/arXiv.2005.14165

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), 213–229. https://doi.org/10.1007/978-3-030-58452-8\_13

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Hamadi, R. (2023). Large language models meet computer vision: A brief survey. arXiv preprint arXiv:2311.16673. https://doi.org/10.48550/arXiv.2311.16673

Jocher, G., & Ultralytics. (2020). YOLOv5 by Ultralytics. GitHub repository. https://github.com/ultralytics/yolov5

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), Computer Vision – ECCV 2016 (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0\_2

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning (ICML 2021). https://doi.org/10.48550/arXiv.2103.00020

Sanh, V., Wolf, T., & Rush, A. M. (2021). Multimodal few-shot learning with Frozen transformers. In Advances in Neural Information Processing Systems (NeurIPS 2021). https://doi.org/10.48550/arXiv.2112.03857

# **Appendices**

**Code**

from flask\_cors import CORS, cross\_origin

from flask import Flask, request, jsonify, send\_file

from routes.analyse import generate\_analysis

from routes.report import generate\_report

from yolo import process\_image

app = Flask(\_\_name\_\_)

cors = CORS(app)

@app.route('/report', methods=['POST', 'OPTIONS'])

def report():

    if request.method == 'OPTIONS':

        return jsonify({}), 200

    return generate\_report(request)

@app.route('/analyse', methods=['POST', 'OPTIONS'])

def analyse():

    if request.method == 'OPTIONS':

        return jsonify({}), 200

    return generate\_analysis(request)

@app.route('/detect', methods=['POST'])

def detect\_objects():

    if 'image' not in request.files:

        return jsonify({'error': 'No image provided'}), 400

    file = request.files['image']

    img\_bytes = file.read()

    # Process image and get result

    result\_image = process\_image(img\_bytes)

    # Return the processed image

    return send\_file(

        result\_image,

        mimetype='image/jpeg',

        as\_attachment=True,

        download\_name='result.jpg'

    )

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True, host='0.0.0.0', port=5000)

# import os

# from dotenv import load\_dotenv

# load\_dotenv()

# print(os.getenv("GEMINI\_API\_KEY"))

# # if api\_key:

# #     print('loaded')

# # else:

# #     print('not loaded, check your env file!')

**Git Repository Link**

<https://github.com/oyin18/Hybrid-CNN-LLM-Object-Detection-.git>