Title page

Improving Visual Q/A in Autonomous Driving Scenerio's Using LLM

submitted in partial fulfillment of the requirements

for the degree of

MASTER OF TECHNOLOGY

in

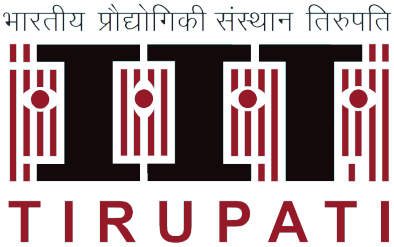
COMPUTER SCIENCE AND ENGINEERING

by

BEENA CS22M104

Supervisor(s)

Dr Chalavadi Vishnu



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY TIRUPATI

MAY, 2024

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission to the best of my knowledge. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Place: Tirupati
Date: 15-05-2024

Signature
Beena
CS22M104

BONA FIDE CERTIFICATE

This is to certify that the report titled **Improving Visual Q/A in Autonomous Driving Scenerio's Using LLM**, submitted by **Beena**, to the Indian Institute of Technology, Tirupati, for the award of the degree of **Master of Technology**, is a bona fide record of the project work done by her under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Tirupati
Date: 19-05-2019

Dr Chalavadi Vishnu
Guide
Assistant Professor
Department of Computer Science and Engineering
IIT Tirupati - 517619

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my guide, Dr. Chalavadi Vishnu, for their invaluable guidance, support, and encouragement throughout the course of this project. I am also thankful to my seniors, juniors, and other teachers for their insightful inputs and constructive feedback. Additionally, I extend my appreciation to my friends and family for their unwavering support and understanding. Their collective encouragement and guidance have been instrumental in providing me with this invaluable learning opportunity.

ABSTRACT

KEYWORDS: Visual Question Answering (VQA); Vision Transformers (ViT); Large Language Models (LLMs); Multimodal Datasets; Autonomous Driving .

In this project, we explore advancements in Visual Question Answering (VQA) within the realm of autonomous driving by synergizing Vision Transformers (ViT) and Large Language Models (LLMs). Our approach integrates ViT for in-depth visual feature extraction and LLMs for nuanced textual understanding, fostering a holistic comprehension of driving scenarios. Through pretraining on diverse multimodal datasets and subsequent fine-tuning for task-specific adaptation, our model aims to significantly improve accuracy in answering a broad spectrum of questions related to traffic signs, road conditions, and driver actions. This endeavor contributes to the evolving integration of computer vision and natural language processing, addressing the intricacies of real-world challenges in autonomous systems for enhanced interpretability and performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

(LLM)	Large language models
ViT	Visual Transformer
CNNs	convolutional neural networks
RNNs	recurrent neural networks
VQA	Visual question answering
LXMERT	Language-visual Multi-Modal BERT
ViLT	Vision-and-Language Transformer

NOTATION

The research scholar/student must explain the meaning of special symbols and notations used in the thesis. Define English symbols, Greek symbols and then miscellaneous symbols Some examples are listed here.

ρ	density, $\frac{m}{kg^3}$
r	Radius, m
θ	Angle between x and y in degrees
v	velocity of the object

CHAPTER 1

INTRODUCTION

Imagine a future where cars don't just drive themselves but also interact with you as a knowledgeable companion. That's the vision behind autonomous driving technology, where we strive to create vehicles that not only navigate roads autonomously but also understand and respond to the nuances of human communication. This project combines cutting-edge technologies in vision and language processing to make this vision a reality, specifically using Vision Transformers (ViT) and Large Language Models (LLMs).

Autonomous vehicles must comprehend vast amounts of visual and textual data to operate safely and efficiently in complex environments. Vision Transformers play a crucial role by processing visual data, helping the vehicle "see" and interpret everything from street signs and traffic signals to pedestrian movements and other vehicles' behavior. This technology enables the car to analyze and understand its surroundings with precision, ensuring safer navigation through cluttered and dynamic landscapes.

Parallelly, Large Language Models (LLMs) are employed to interpret and generate human-like responses based on the vehicle's observations and the interactions with its human occupants. These models process natural language, allowing the vehicle to understand questions and commands from passengers, and even engage in meaningful conversations. For instance, a passenger could ask, "What's the cause of the traffic jam ahead?" and the car could analyze real-time data to provide an informed response.

However, integrating these two technologies presents significant challenges. Traditional approaches often treat the visual and linguistic processing components as separate entities, which can lead to disjointed functionality where the vehicle might see obstacles but not effectively understand verbal queries about them. Our project aims to create a more holistic system where vision and language models work in concert, allowing for seamless integration of visual data interpretation with language understanding.

This integration is akin to assembling a complex puzzle where each piece represents different facets of sensory and cognitive processing. Just as a puzzle is incomplete if pieces are missing or

disconnected, an autonomous vehicle's functionality is impaired without the cohesive interaction between its 'eyes' and 'brain.' By synthesizing the capabilities of ViT and LLMs, we can equip vehicles with a comprehensive understanding of their environment, much like giving them a pair of super eyes and a highly intelligent brain.

Addressing these challenges involves developing algorithms that can effectively combine visual data with textual context in real-time. It requires advancements in machine learning, particularly in areas of multi-modal learning, where the system learns to link and interpret information from different types of data simultaneously. Moreover, ensuring that these systems can operate in real-world conditions with unfailing accuracy and reliability is paramount, as any misinterpretation or delay in processing could lead to critical failures.

As we continue to refine these technologies, the potential applications extend beyond just personal vehicles to public transport, delivery services, and even emergency response vehicles, all benefiting from autonomous technology equipped with advanced perceptual and conversational abilities. The goal is to make autonomous vehicles not only tools of convenience but also reliable partners in everyday travel, capable of understanding and interacting with their human users in a contextually relevant and meaningful way. This project isn't just about technological innovation; it's about reshaping the future of transportation to be safer, more efficient, and inherently interactive.

1.1 Problem Statement

Our mission is to transform automobiles into highly intelligent companions capable of understanding and responding accurately to queries about driving. To achieve this, we are leveraging innovative technologies like Vision Transformers (ViT) and Large Language Models (LLMs) illustrate in figure 1.1.

Vision Transformers are akin to the car's advanced eyes. They meticulously analyze the environment to detect subtle details, such as traffic signals and pedestrian actions. This technology significantly enhances the car's visual perception, which is crucial for making informed and safe driving decisions.

On the other hand, Large Language Models function as the car's brain. They are adept at

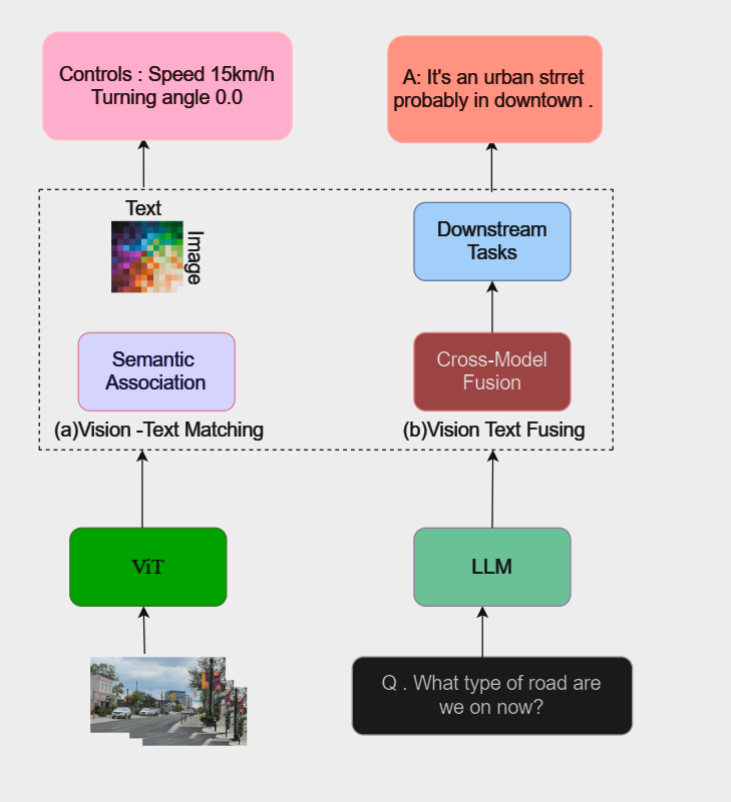


Figure 1.1: Two inter-modality connection approaches of Vision-Language Model in Autonomous Driving

processing both spoken and written queries related to driving. These models excel in understanding the context and intent behind questions, enabling the vehicle to engage in meaningful interactions. This capability allows the car to provide helpful and precise responses to the driver's inquiries.

By integrating Vision Transformers with Large Language Models, we equip vehicles with both superior sight and sophisticated comprehension. This combination not only makes autonomous vehicles safer but also transforms them into proactive assistants. They are now equipped to tackle the complexities of real-world driving with intelligence and precision, turning cars into smarter, more responsive partners on the road.

1.2 Motivation

The increasing adoption of autonomous driving technologies showcases promising progress in the automotive industry, yet it also highlights significant challenges in user acceptance and safety perception. This report draws on recent data and aims to address these critical issues by

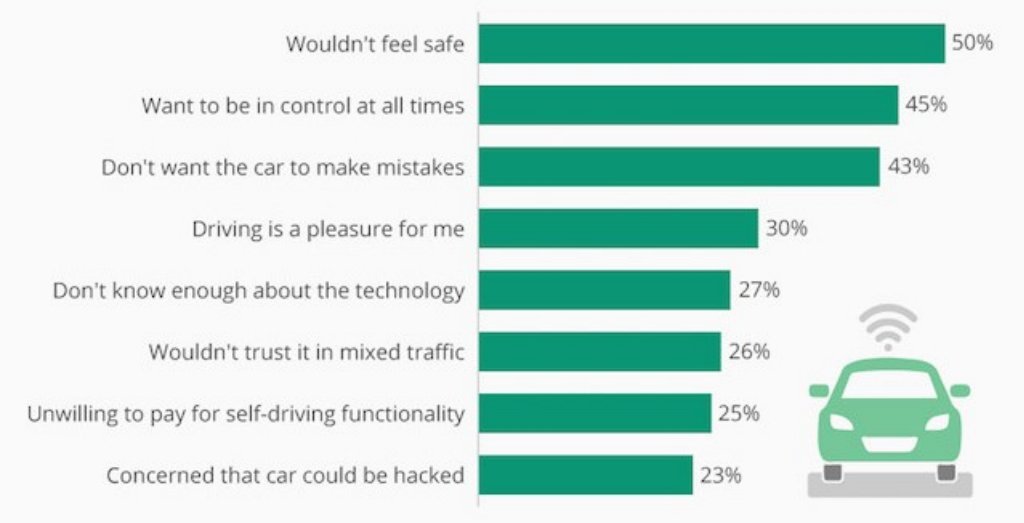


Figure 1.2: Percentage of respondents naming the following reasons for there reluctance to use self driving cars.

implementing advanced technologies like Visual Question Answering (VQA). Here's a detailed look at the current landscape and our objectives:

1.2.1 Growing Popularity

Autonomous vehicles are becoming more prevalent, with a robust 25% annual increase in users as reported by the International Association of Autonomous Vehicles (IAAV). This surge indicates a rising trust and interest in autonomous technologies among new users.

1.2.2 Persistent Safety Concerns

Despite the growth, a substantial 70% of users still harbor reservations primarily due to safety concerns, especially in areas with high pedestrian traffic. A survey by J.D. Power highlights illustrate in figure 1.2 these apprehensions, emphasizing the need for enhanced safety measures:

- **50% of respondents** do not feel safe in autonomous vehicles.
- **45% of users** prefer to maintain control over the vehicle at all times.
- **27% lack proper knowledge** about the technology, which may contribute to their hesitation.
- **26% do not trust** the technology's reliability in heavy traffic situations.

1.2.3 The Role of Visual Question Answering (VQA)

To mitigate these concerns, this report proposes the adoption of Visual Question Answering technology. VQA can play a pivotal role in enhancing user trust and safety perception by:

- **Providing Real-Time Insights**: VQA systems can analyze and interpret the vehicle's surroundings and immediately respond to the driver's queries about visible road conditions and obstacles.
- **Enhancing Situational Awareness**: By continuously assessing the environment and providing feedback, VQA helps reinforce the vehicle's capability to handle complex driving scenarios, thereby boosting the driver's confidence.
- **Educating Users**: Implementing VQA not only improves safety but also serves as an educational tool that informs users about how the vehicle makes decisions. This transparency is crucial in building trust.

1.3 Objective of This Report

The primary goal of this report is to explore how VQA technology can directly address and alleviate safety concerns associated with autonomous vehicles. By offering real-time solutions and detailed insights into the vehicle's operational dynamics, we aim to foster a safer and more widely accepted autonomous driving experience.

By addressing these key points, we aim to bridge the gap between technological advancements and user confidence, ensuring that the future of autonomous driving is not only innovative but also aligned with the expectations and safety requirements of its users.

CHAPTER 2

Literature Review

The rapid evolution of autonomous driving technologies is marked by the integration of advanced computational models that enhance vehicle perception and decision-making capabilities. Central to this advancement are transformers in multimodal models, which have revolutionized how machines understand and integrate disparate forms of data. This literature review delves into the impact of such technologies, with a focus on their application in autonomous driving scenarios, exploring various datasets and models like NuScenes, LXMERT, ViLT, and ViLBERT.

2.1 Transformers in Multimodal Models

Transformers, since their inception by Vaswani *et al.* (2017), have dramatically reshaped the landscape of machine learning. In the context of autonomous driving, they facilitate a deeper contextual comprehension and more dynamic intermodal information exchange.In Nagrani *et al.* (2021*a*) and Nagrani *et al.* (2021*b*) shows how Transformers emerge as pivotal orchestrators, enabling early fusion, late fusion,and cross-modal attention .

2.1.1 Early Fusion

In early fusion, transformers allow for the simultaneous processing of data streams. This approach integrates inputs at an initial stage, leading to joint representations of textual and visual data. By merging information early, the model can capture interactions between modalities that might be lost when processed separately.

2.1.2 Late Fusion

Contrasting with early fusion, late fusion involves processing data streams separately before their final integration. This technique allows each modality to develop independent features that maintain their unique characteristics until the final stages of processing.

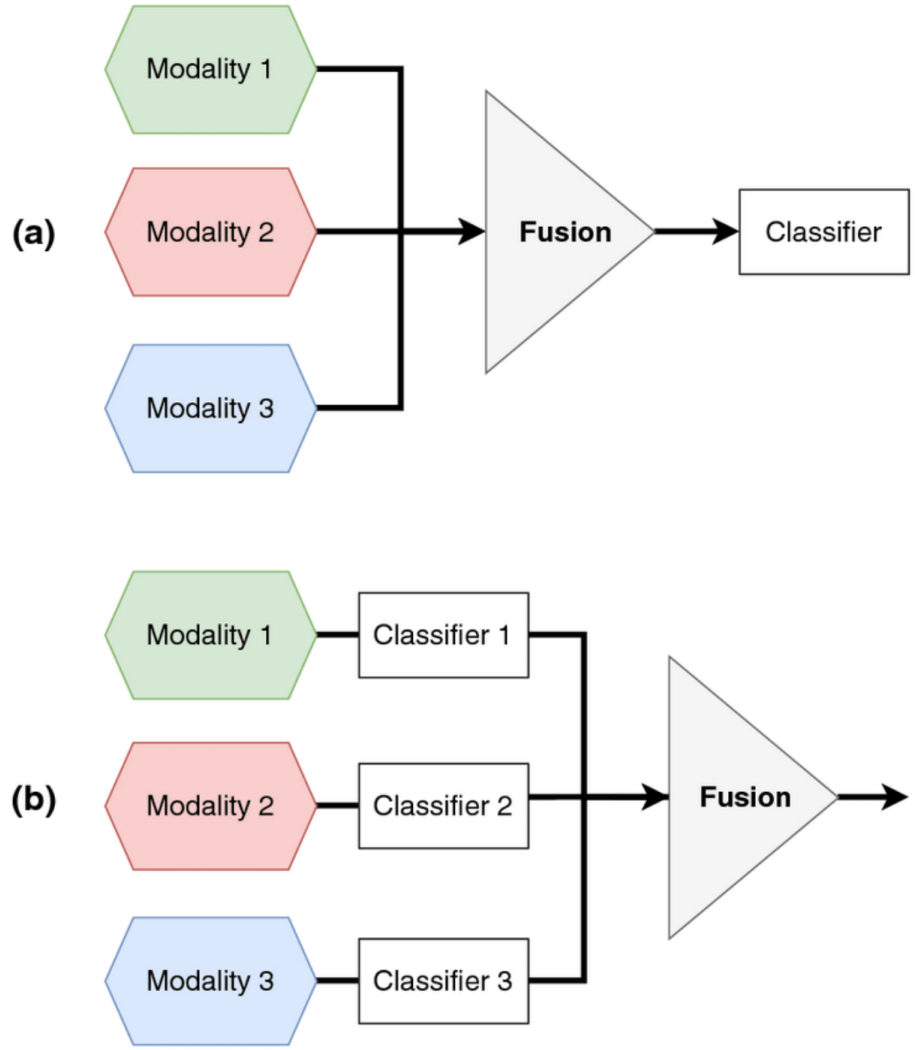


Figure 2.1: Conventional methods for multimodal data fusion: (a) Early fusion, (b) Late fusion

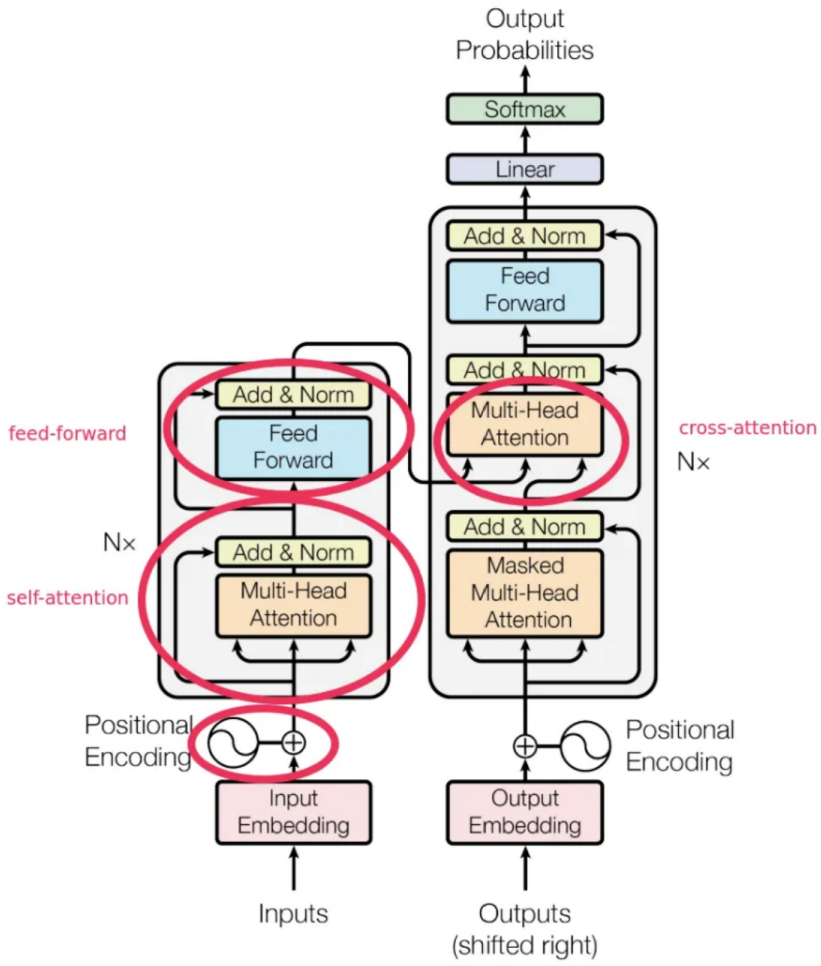


Figure 2.2: Cross attention in Transformers

2.1.3 Cross-modal Attention

Cross-modal attention acts as a conductor in the symphony of data integration, allowing the model to focus on relevant features across textual and visual inputs dynamically .In normal way in a Transformer when the information is passed from encoder to decoder that part is known as Cross Attention. Many people also call it as Encoder-Decoder Attention illustrate in figure2.2

2.2 NuScenes Dataset

The NuScenes dataset, introduced by Caesar *et al.* (2020), represents a cornerstone for training and evaluating autonomous driving systems.

Figure 2.3: An example from the nuScenes dataset. We see 6 different camera views, lidar and radar data, as well as the human annotated semantic map

2.2.1 Data Collection and Annotation

NuScenes encompasses a vast array of annotated objects, scenes, and contextual elements captured in various urban settings. The process involves equipping vehicles with an array of sensors including LIDARs, RADARs, and cameras, which collect data in diverse weather and lighting conditions shown in figure 2.3. Each object within the scene is meticulously labeled, encompassing a broad spectrum of static and dynamic elements from traffic lights and signage to pedestrians and other vehicles. This detailed curation process ensures that the dataset accurately reflects the complexity and diversity of real-world driving scenarios, providing a robust foundation for training sophisticated autonomous systems.

2.2.2 Impact on VQA Models

The depth and variety of the NuScenes dataset make it particularly valuable for training Visual Question Answering (VQA) models. These models, tasked with understanding and responding to queries about visual content, benefit immensely from the realistic and challenging scenarios presented in NuScenes. Training on such a dataset enables these models to better generalize and perform in real-world situations that autonomous vehicles will encounter, thereby improving their decision-making and interaction capabilities.

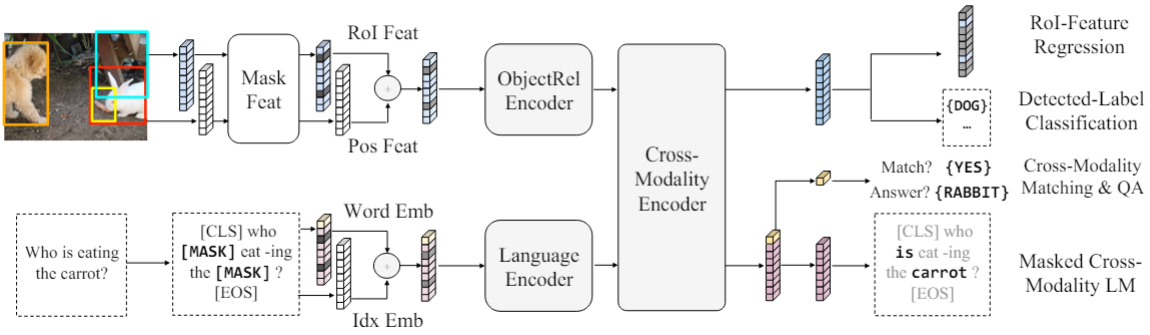


Figure 2.4: Pre-training in LXMERT. The object RoI features and word tokens are masked.Five pre-training tasks learn the feature representations based on these masked inputs. Special tokens are in brackets and classification labels are in braces.

2.3 LXMERT: Language and Vision Fusion

Developed by Tan and Bansal. (2019) , LXMERT stands out for its innovative approach to merging language and visual data.

2.3.1 Fusion Process

LXMERT uniquely integrates dual-stream architectures for processing visual and textual inputs. It employs a series of transformer encoders that separately process the inputs before they are fused for joint processing. This model is particularly adept at learning rich joint representations that encapsulate both the textual descriptions and the visual elements. Such integration allows LXMERT to achieve a more nuanced understanding of complex queries that involve both visual cues and contextual language.

2.3.2 Performance and Limitations

LXMERT has demonstrated substantial improvements in VQA tasks, showcasing its ability to interpret and answer questions about complex visual scenes accurately. However, its sophisticated architecture demands significant computational resources, which can impede deployment in real-time applications. Additionally, the performance of LXMERT hinges on the diversity and representativeness of the training data, which can be a limitation if not adequately addressed.

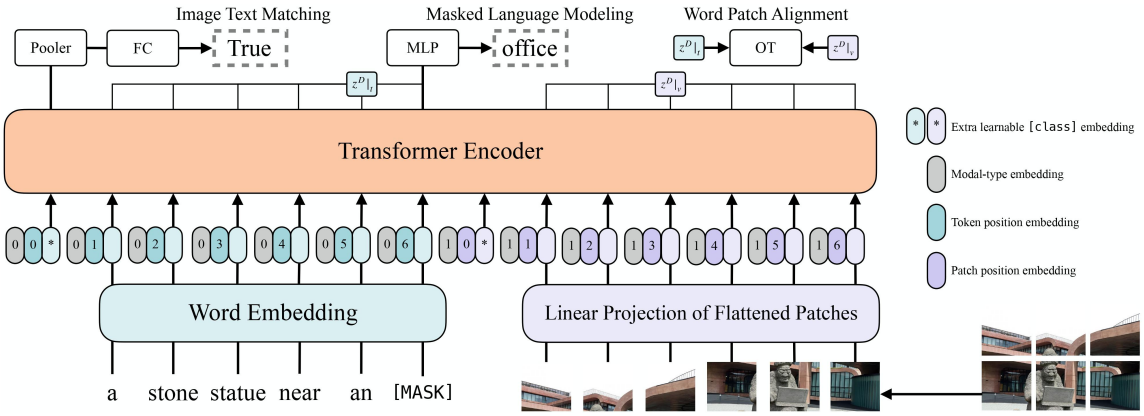


Figure 2.5: Trained On Two Tasks - [1]Masked multi-modal learning ,[2]Multi-modal alignment prediction.

2.4 ViLT and ViLBERT: Pioneers of Joint Reasoning

ViLT and ViLBERT, introduced by Kim *et al.* (2021)andLu *et al.* (2019), respectively, extend the capabilities of transformers by incorporating co-attention mechanisms.

2.4.1 Co-attention Mechanism

These models incorporate a co-attention mechanism that allows them to simultaneously attend to both textual and visual inputs. This process not only improves the alignment of language and visual data but also enables the models to construct a more coherent and contextually enriched understanding of the scene. Such capabilities are particularly crucial in autonomous driving, where interpreting complex and dynamic visual scenes alongside relevant textual information (like GPS data or user inputs) is necessary for safe operation shown in figure 2.5.

2.4.2 Challenges

Despite their advanced capabilities, ViLT and ViLBERT face challenges when deployed in highly dynamic or unpredictable environments. Their reliance on large pre-trained models and extensive training datasets can also pose scalability and adaptability issues, particularly in novel or rare scenarios not well-represented in training data.

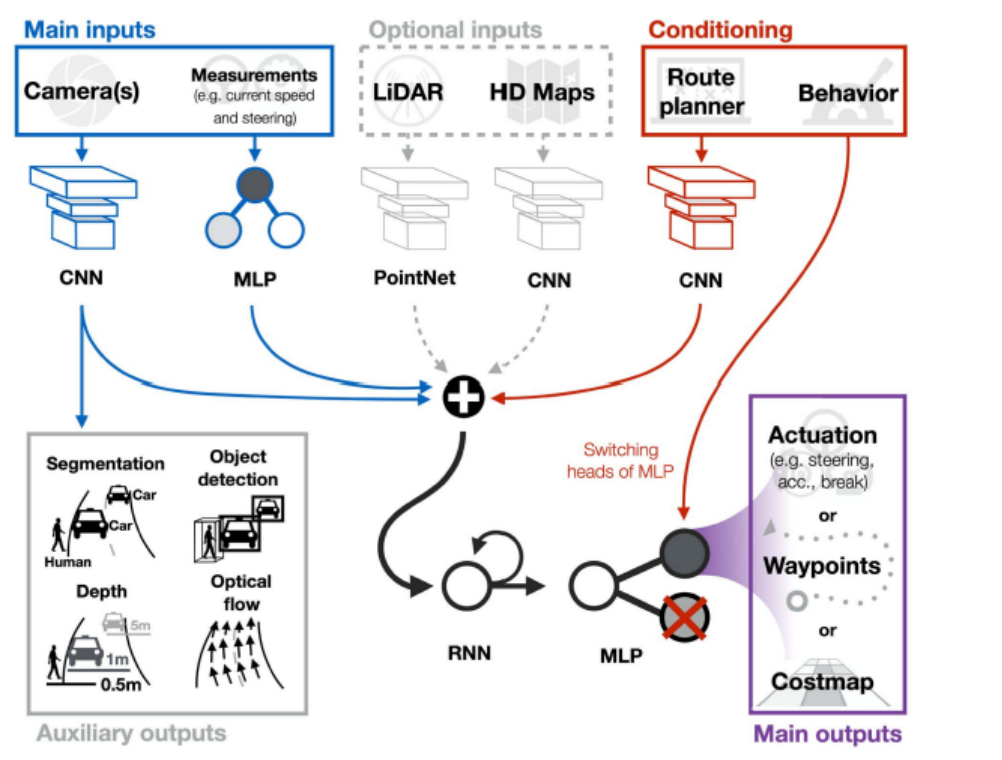


Figure 2.6: End- to -End autonomous driving system

2.5 CNN-based Approaches: The Foundation

Before the advent of transformer-based models, Convolutional Neural Networks (CNNs) were the backbone of visual processing in autonomous vehicles.In Tampuu and Muhammad (2020) the symphonyof autonomous driving evolution, CNN-based models take their place. Relying on Convolutional Neural Networks, these predecessors exhibit prowess in extracting hierarchical features from visual inputs, contributing to object detection, lane keeping, and environmental perception

2.5.1 Hierarchical Feature Extraction

CNNs are renowned for their ability to perform hierarchical feature extraction. By applying convolutional operations across visual data, these models efficiently identify and categorize spatial hierarchies and patterns. This capability makes CNNs exceptionally good at recognizing road features, detecting objects, and maintaining lane discipline—all crucial for the early stages of autonomous vehicle development shown in figure 2.6.

2.5.2 Limitations

However, CNNs often struggle with capturing the broader context of the scene, which can limit their effectiveness in complex or nuanced driving scenarios. Their largely static nature also makes them less adaptable to the rapid changes that can occur in driving environments, such as sudden shifts in weather conditions or unexpected pedestrian actions.

CHAPTER 3

METHODOLOGY

3.1 Different Models Used

In the dynamic landscape of autonomous driving, the amalgamation of Vision Transformers (ViT) and Large Language Models (LLMs) stands as a transformative endeavor. Harnessing the power of ViT and LLMs, this project seeks to elevate the capabilities of Visual Question Answering (VQA) for autonomous vehicles. The symbiotic fusion of ViT's visual prowess and LLMs' linguistic acumen aims to usher in a new era of contextual understanding and decision-making in self-driving scenarios.

3.1.1 ViT-(Visual Transformer)

In Yuan *et al.* (2021) the ever-evolving field of computer vision, Vision Transformers (ViT) have emerged as a groundbreaking model architecture, challenging conventional methods of image processing. ViT represents a paradigm shift by replacing traditional Convolutional Neural Networks (CNNs) with transformers, which were initially designed for natural language processing tasks. This shift has proven remarkably successful, showcasing ViT's versatility and efficacy in handling visual data.

Working Of ViT

Image Tokenization: ViT breaks down an input image into fixed-size non-overlapping patches, treating each patch as a token.

Linear Embedding: These image patches are linearly embedded into flat vectors, preserving their spatial relationships.

Positional Encoding: To incorporate positional information, ViT adds positional encodings to the token embeddings, enabling the model to discern the spatial arrangement of the patches.

Transformer Encoder: The tokenized image is then processed through a transformer encoder,

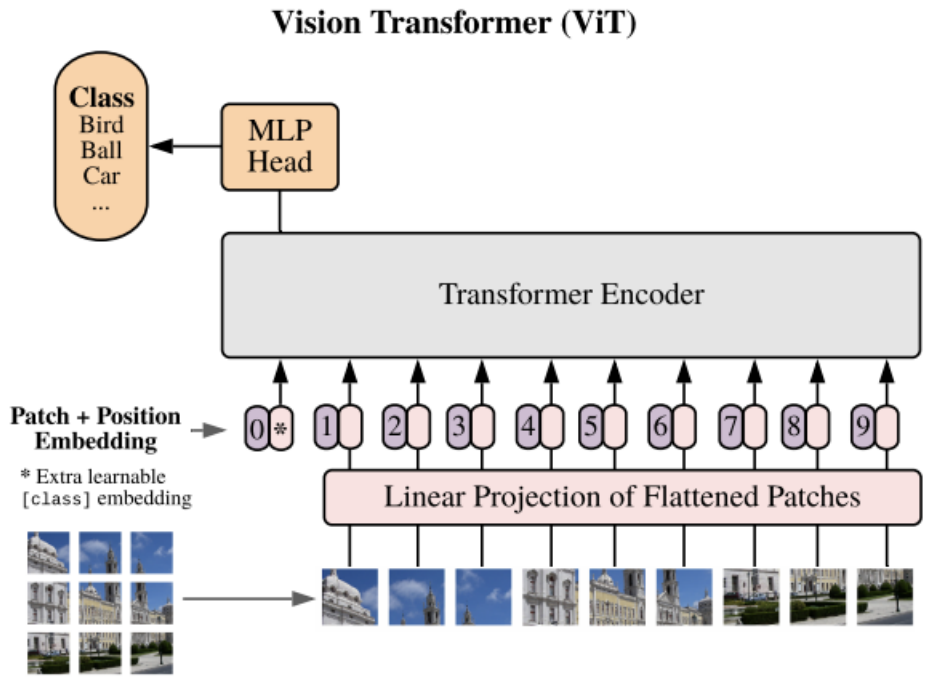


Figure 3.1: Vision transformer architecture

facilitating interactions between different parts of the image for holistic understanding.

Classification Head: The transformer's output is fed into a classification head, which produces predictions for various tasks, such as image classification or object detection.

Architecture

The architecture typically comprises in 3.1:

Input Embedding Layer: Handling the initial tokenization and linear embedding of image patches.

Transformer Encoder Blocks: Consisting of multiple layers of transformer blocks for capturing hierarchical features.

Classification Head: The final layer responsible for generating predictions based on the processed features.

ViT's strength lies in its ability to capture long-range dependencies in visual data, making it particularly effective for tasks requiring global context understanding, such as image classification and scene understanding. Its success has spurred further exploration of transformer-based architectures in various computer vision domains.

3.1.2 LLM-(Large Language Transformer)

Large Language Models (LLMs) like BERTDevlin *et al.* (2018) play a pivotal role in the intersection of language understanding and visual perception, making them integral to advancements in autonomous driving scenarios. These models, equipped with the ability to comprehend and generate human-like language, contribute significantly to the interaction between artificial intelligence (AI) systems and the complex visual inputs encountered in autonomous vehicles.

Architecture Of LLM

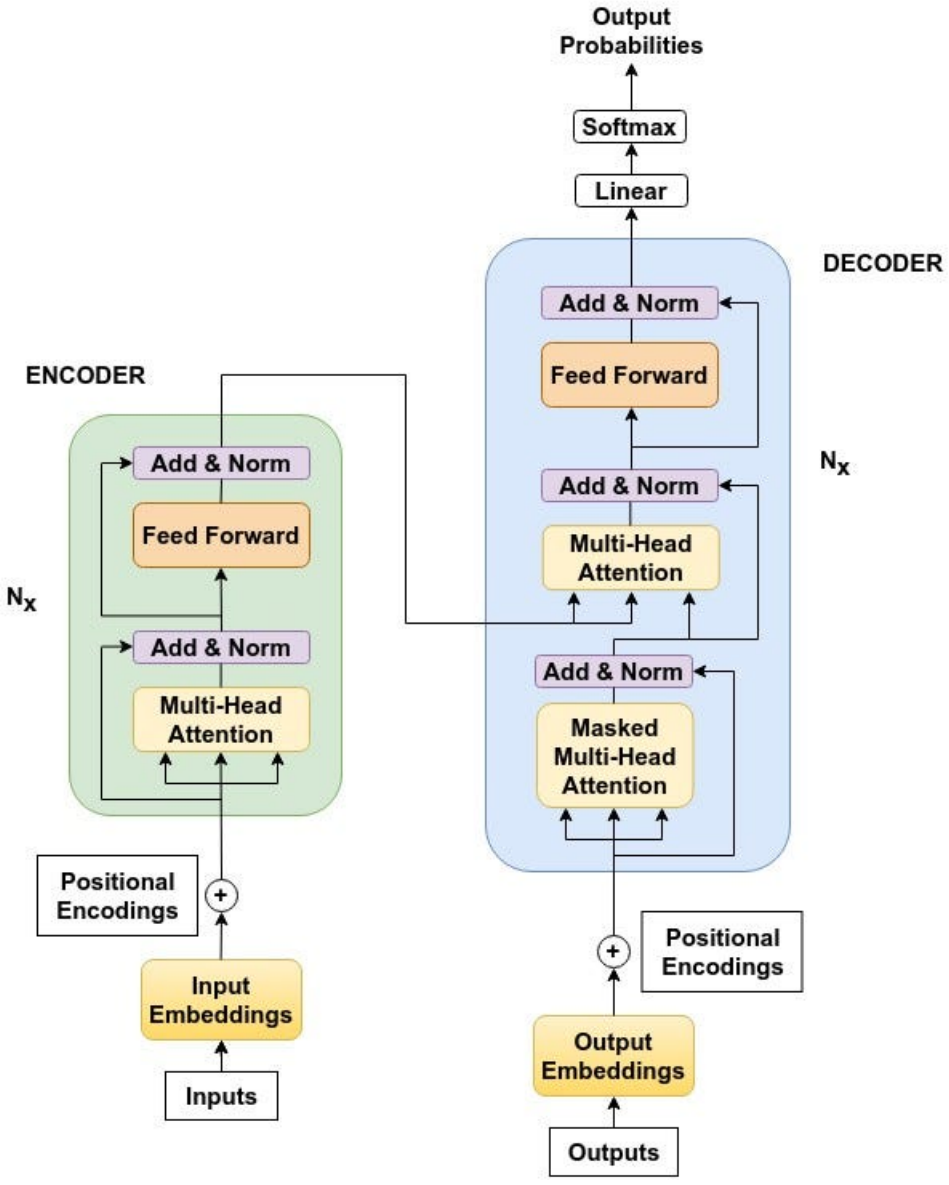


Figure 3.2: LLM transformer architecture

Input Embedding: The input text is tokenized into smaller units, often words or subwords.Each token is represented as an embedding vector.

Positional Embedding: Since transformer models don't inherently understand the order of tokens, positional embeddings are added to provide information about the position of each token in the sequence.

Encoder: Transformer Blocks: The core of the model consists of multiple transformer encoder blocks. Multihead Attention: Each block includes a multihead self-attention mechanism, allowing the model to focus on different parts of the input sequence. Feedforward Network: A feedforward neural network processes the output of the attention mechanism.

Output Embedding: The final output embeddings represent the contextualized information of each token in the input sequence. LLMs, when combined with visual encoders like ViT, provide a comprehensive understanding of both textual and visual inputs, aiding in tasks like interpreting road signs and responding to queries about the environment.

This architecture 3.2 allows LLMs to capture intricate linguistic and contextual information, making them versatile for various natural language understanding tasks in autonomous systems.

3.1.3 BERT Model

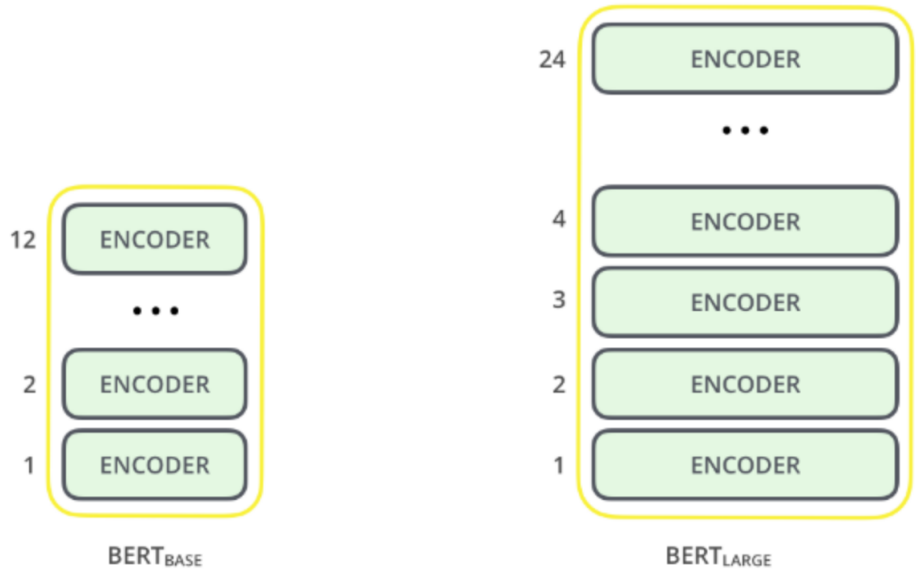


Figure 3.3: Two configrations of BERT

BERT builds on the Transformer model and is designed to deeply understand the context from both directions of text simultaneously. This is a critical enhancement over previous models that only read text from left to right or right to left. BERT is available in two configurations shown in figure 3.3:

- BERT Base: Consists of 12 layers (transformer blocks), 12 attention heads, and approximately 110 million parameters.
- BERT Large: Comprises 24 layers, 16 attention heads, and around 340 million parameters.

Both versions employ an encoder-only architecture, maintaining the same model size as OpenAI's GPT for comparative purposes.

Text Preprocessing for BERT

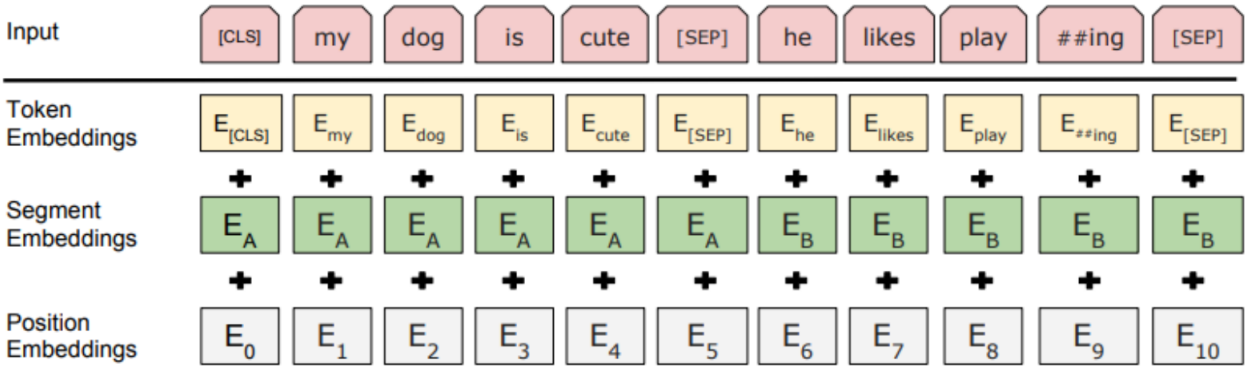


Figure 3.4: Text PreProcessing of BERT

Before feeding data into BERT, several preprocessing steps are necessary. The input for each token is represented by combining three types of embeddings shown in figure 3.4:

- **Position Embeddings:** These embeddings provide information about the positional context of words in a sentence, compensating for the Transformer's lack of sequential order capabilities.
- **Segment Embeddings:** For tasks that involve pairs of sentences, segment embeddings help the model differentiate between the two.
- **Token Embeddings:** Derived from BERT's WordPiece tokenization, these embeddings represent individual tokens.

Pre-training Tasks

BERT's pre-training involves two critical tasks:

Figure 3.5: Predicting the word in Sequence

- **Masked Language Modeling (MLM)** Unlike conventional language models that predict the next word in a sequence, MLM randomly masks words and predicts them based on their context shown in figure 3.5, thus learning a bidirectional understanding. During MLM pre-training:
  - 15% of the words in each sequence are masked.
  - These masked words are 80% replaced by a [MASK] token, 10% by a random word, and 10% are unchanged.
- **Next Sentence Prediction (NSP)** BERT learns to predict whether a sentence logically follows another, a vital task for understanding the relationship between consecutive sentences. This is particularly useful for applications like question answering.

3.2 Architecture For Our Multimodel Visual Q/A

The architecture of the model in 3.6 involves key components:

Vision Transformer: This module extracts crucial patterns and features from visual information in images, enhancing the model's grasp of the visual context.

Language Model: Responsible for textual information, the language encoder processes questions or queries associated with input images, bridging the gap between visual and textual

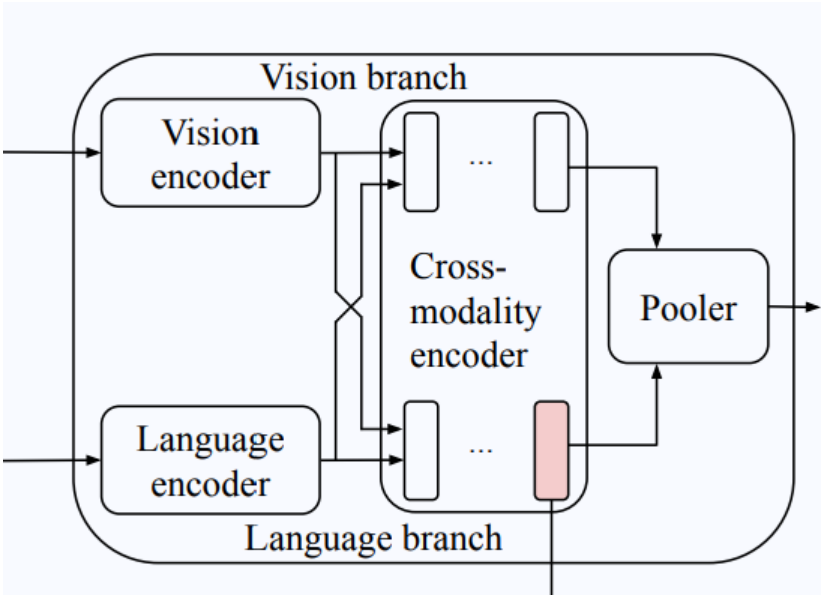


Figure 3.6: Architecture of vision branch in Autonomous

modalities.

Cross-Modality Fusion: This pivotal step combines information from both visual and textual modalities, creating a unified representation that enriches the model's overall understanding.

Pooler: Acting as a selection mechanism, the pooler efficiently extracts essential information from language and vision branches' outputs, optimizing subsequent processing.

Prediction Head: The final layer is the linchpin for generating ultimate answers based on the processed information, trained using a binary classification loss function. This intricate yet streamlined architecture ensures a comprehensive fusion of visual and textual cues for effective question answering.

3.3 Working Of Model

Input: Image and Caption Text

Image Input: The model takes an image as input. This could be in the form of pixel values representing the visual content.

Caption Text Input: Simultaneously, the model receives a caption or textual input associated with the image. This could be a question or a query related to the content of the image.

Vision Encoder :

Function: The Vision Encoder processes the image input.

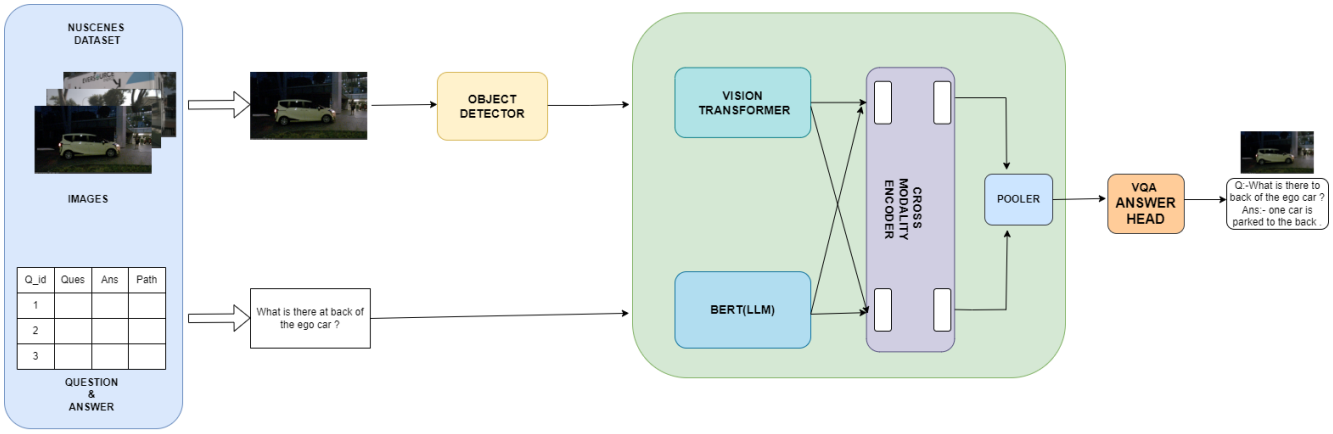


Figure 3.7: Architecture of proposed model in Autonomous

Working: Utilizes convolutional neural network (CNN) layers to capture hierarchical and spatial features in the visual information.

Output: Produces a feature representation that encodes essential visual patterns.

Language Encoder:

Function: The Language Encoder processes the caption text input.

Working: Utilizes recurrent neural networks (RNNs), transformers, or similar architectures to understand the sequential information in the textual input.

Output: Generates a feature representation that captures contextual information from the caption.

Cross-Modality Fusion:

Function: Combines information from both the visual and textual modalities.

Working: The output feature representations from the Vision Encoder and Language Encoder are combined, creating a unified representation that integrates both visual and textual cues.

Output: A fused representation that enriches the model's overall understanding of the input.

Pooler:

Function: Acts as a selection mechanism to extract essential information from the fused representation.

Working: Utilizes pooling operations (e.g., max pooling, average pooling) to downsample and focus on relevant information in a computationally efficient manner.

Output: Further refined and focused representation.

Prediction Head:

Function: The final layer responsible for generating ultimate answers based on the processed information.

Working: Involves fully connected layers and a binary classification loss function.

Output: Produces the final answer or prediction, often in the form of a binary decision (e.g., yes/no).

Training:

Objective: The model is trained using a binary classification loss function. The training process involves minimizing the difference between the predicted output and the ground truth labels associated with the input.

Inference/Test:

Application: Once trained, the model can be used for inference or testing on new, unseen data.

Prediction: Given a new image and associated caption, the model processes the input through the entire architecture and produces a final answer.

Output: Final Answer:

Interpretation: The final output is the model's answer to the question or query associated with the input image.

Example: If the question is binary (e.g., "Is there a cat in the image?"), the output could be "Yes" or "No" based on the model's prediction.

CHAPTER 4

EXPERIMENTAL DETAILS

4.1 Data Preprocessing

The initial step involved accessing the dataset stored on Google Drive. This was achieved by mounting the drive within the Colab environment, facilitating seamless data manipulation and processing.

4.1.1 Dataset Splitting

To evaluate the model effectively, the dataset was split into training and testing sets using a 75:25 ratio. This ensured that 75% of the data was used for training and the remaining 25% for testing. A random state was set for reproducibility of the split.

Steps for Dataset Splitting

- Load the original dataset from the CSV file.
- Split the data into training and testing sets in a 75:25 ratio.
- Save the train and test datasets into new CSV files for further processing.

4.1.2 Combining Datasets and Extracting Unique Answers

After splitting the dataset, the train and test datasets were combined to extract unique answers. This step was crucial for understanding the answer space and ensuring that all possible answers were considered during model training. The unique answers were then saved to a text file for future reference.

Steps for Extracting Unique Answers

- Load the train and test datasets.
- Combine the two datasets to form a complete dataset.
- Extract unique answers from the combined dataset.
- Save the unique answers to a text file for future use.

4.1.3 Further Data Processing

Additional preprocessing involved modifying the dataset to fit the model's requirements. This included extracting image paths and ensuring that each entry had the necessary information.

Steps for Data Processing

- Load the training dataset.
- Extract relevant columns such as scene token, frame token, and image paths.
- Split the combined QA pairs into separate questions and answers.
- Append the modified data to a new dataset structure.
- Write the modified dataset to a new CSV file.

4.2 Model Training

The model training phase involved several key steps to ensure the dataset was properly formatted and ready for training. These steps included creating a multimodal VQA collator and model, defining training arguments, and training the model.

4.2.1 Model Architecture

The model used for training was a multimodal model that combined text and image inputs. The text model was based on the 'roberta-base' architecture, while the image model used the 'google/vit-base-patch16-224-in21k' architecture. These models were integrated into a unified multimodal model to handle both textual and visual inputs.

Model Components

- **Text Model**: The 'roberta-base' model was used to process textual inputs.This model is a transformer-based architecture known for its strong performance on natural language processing tasks.
- **Image Model**: The 'google/vit-base-patch16-224-in21k' model was used to process visual inputs. This Vision Transformer (ViT) model splits images into patches and processes them similarly to tokenized text, enabling effective image recognition.
- **Multimodal Integration**: The text and image models were integrated to form a unified model capable of processing both types of inputs simultaneously. This integration was crucial for the Visual Question Answering (VQA) task.

4.2.2 Training Process

The training process involved defining the training arguments, initializing the model, and training it using the prepared dataset. The training was performed using the Hugging Face Trainer API, which facilitated efficient training and evaluation.

Steps for Model Training

1. Define the training arguments, including output directory, evaluation strategy, and other hyperparameters.
2. Create the multimodal VQA collator and model using specified text and image models.
3. Initialize the Trainer with the model, training arguments, datasets, and collator.
4. Train the model and evaluate its performance on the test dataset.

Algorithm for Model Training

Algorithm 1 Model Training Algorithm

1: **Input:** Training and testing datasets, model configuration
2: **Output:** Trained model, evaluation metrics
3: Define training arguments
4: Create multimodal VQA collator and model
5: Initialize the Trainer
6: Train the model using the training dataset
7: Evaluate the model using the test dataset
8: Return the trained model and evaluation metrics

4.3 Evaluation

The trained model was evaluated using the test dataset. The evaluation metrics included accuracy, precision, recall, F1-score, and the WUP (Wu-Palmer) similarity metric. These metrics provided a comprehensive understanding of the model's performance and its ability to generalize to unseen data.

4.3.1 Evaluation Metrics

- **Accuracy**: The proportion of correctly predicted instances out of the total instances.
- **Precision**: The proportion of true positive predictions out of the total positive predictions.
- **Recall**: The proportion of true positive predictions out of the total actual positives.
- **F1-Score**: The harmonic mean of precision and recall, providing a single metric that balances both.
- **WUP Similarity**: The Wu-Palmer similarity metric measures the similarity between two concepts in a taxonomy based on the depth of the concepts and their least common subsumer.

Steps for Model Evaluation

1. Evaluate the model using the test dataset.
2. Calculate accuracy, precision, recall, and F1-score.
3. Compute the WUP similarity metric to assess the semantic similarity of the predictions.
4. Visualize the results using confusion matrices and example predictions.

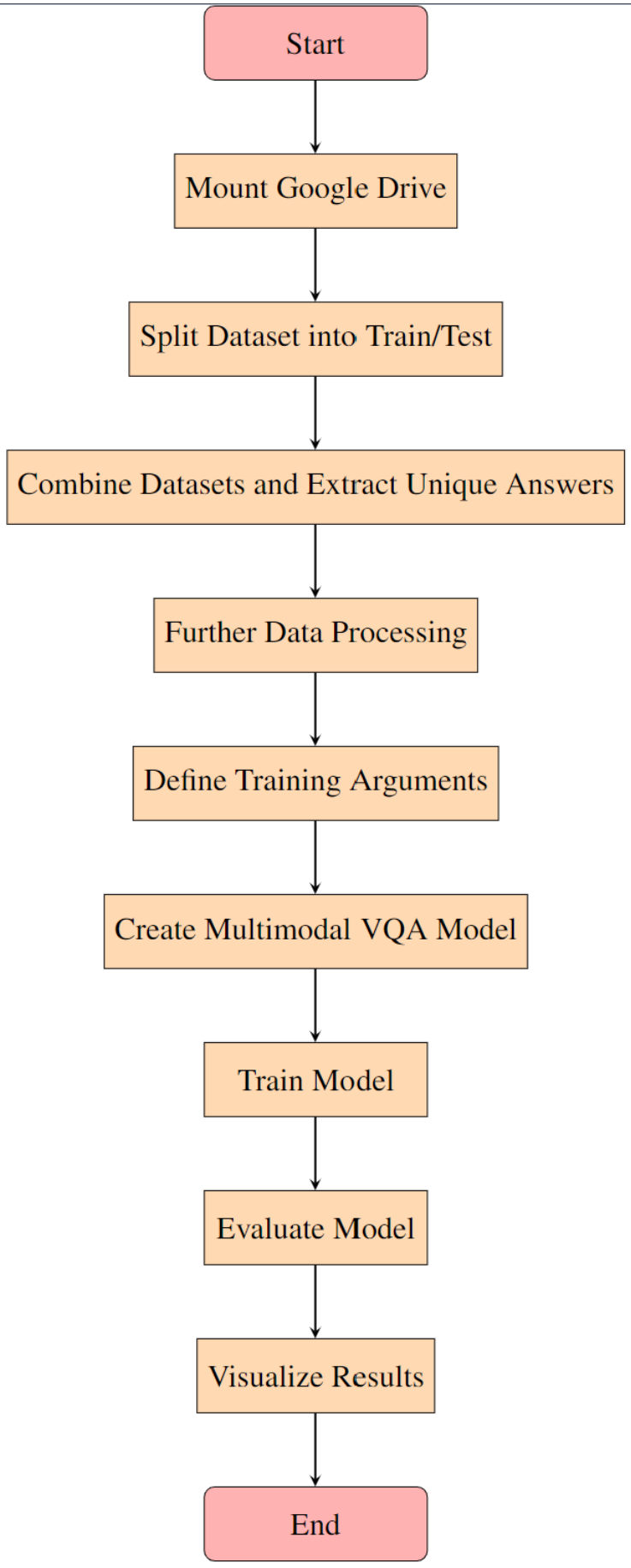


Figure 4.1: Whole Experiment FlowChart

CHAPTER 5

Experiments And Results

5.1 Traning And Testing

5.1.1 Training Pipeline

Three Parts:

Dataset Generation: This phase involves creating a dataset by combining the proposed questions and image data. The questions are carefully selected based on predefined criteria, and the dataset becomes a crucial input for training.

Supervised Representation Learning: In this stage, the model undergoes training using the meticulously generated dataset. The training process focuses on enhancing the model's ability to understand and represent features from both questions and images.

Self-Explanatory Heatmaps Generation: Leveraging the cross-attention module, attention scores are extracted. These scores are utilized to generate self-explanatory heatmaps, providing insights into the model's focus areas during the processing of visual information.

5.1.2 Testing Pipeline

Four Parts:

Dataset Splitting: Reserved for evaluating the model's performance on new and unseen data, this step involves splitting the dataset into a portion dedicated to testing.

Evaluation: The model is assessed on the testing set to gauge its generalization capabilities. This phase provides valuable insights into how well the model performs on previously unseen data.

Performance Metrics: Various metrics are employed to measure the model's effectiveness on the testing data. These metrics offer quantitative assessments of its performance, aiding in the identification of strengths and potential areas for improvement.

Adjustments and Fine-Tuning: Based on the testing results, the model or its hyperparameters may undergo adjustments and fine-tuning. This iterative process aims to enhance the model's overall performance by addressing identified shortcomings or limitations.

5.2 Metrics

5.2.1 WUPS Score:

The WUP (Wu-Palmer) similarity score is a metric used to measure the similarity between two words based on their depth in a hierarchical structure such as WordNet. It is often used in natural language processing tasks to assess the semantic similarity between words.
The WUP similarity score is calculated using the following formula:

$$\text{WUP Score} = \frac{2 \times \text{Depth of the Lowest Common Subsumer (LCS)}}{\text{Depth of Word 1} + \text{Depth of Word 2}}$$

Where:

- The "Depth of the Lowest Common Subsumer (LCS)" is the depth of the lowest common ancestor in the hierarchical structure.
- The "Depth of Word 1" and "Depth of Word 2" are the depths of the words in the hierarchy.

Usage of metrics in Evaluation

1. Semantic Similarity Measurement:WUP also used to quantify the semantic similarity between words. Higher scores indicate greater similarity.

2. Word Sense Disambiguation:In applications where understanding the specific meaning of a word is crucial (e.g., question answering), WUP score can aid in word sense disambiguation.

3. Evaluation Metric: most importantly WUP score can serve as an evaluation metric for tasks like question answering or information retrieval, where understanding the similarity between predicted and ground truth answers is essential.

5.2.2 Accuracy

Accuracy is a measure of the overall correctness of a classification model. It is calculated as the ratio of correctly predicted instances to the total instances.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100$$

5.2.3 F1 Score

The F1 score is a metric that combines precision and recall into a single value. It is useful when there is an uneven class distribution.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

where

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

CHAPTER 6

Phase 1 Results for Indoor dataset

6.1 Metrics Evaluations Results

In the evaluation of our model on a subset of 10 samples from the test set, we compared the predicted labels against the ground truth. The predicted labels, such as "photo," "table," "lamp," and others, were assessed for correctness. The WordNet-based similarity (WUP) scores 6.1 were computed for each prediction, indicating the semantic similarity between the predicted and true labels. Scores of 1.0 signify perfect matches, while lower scores indicate less semantic similarity.

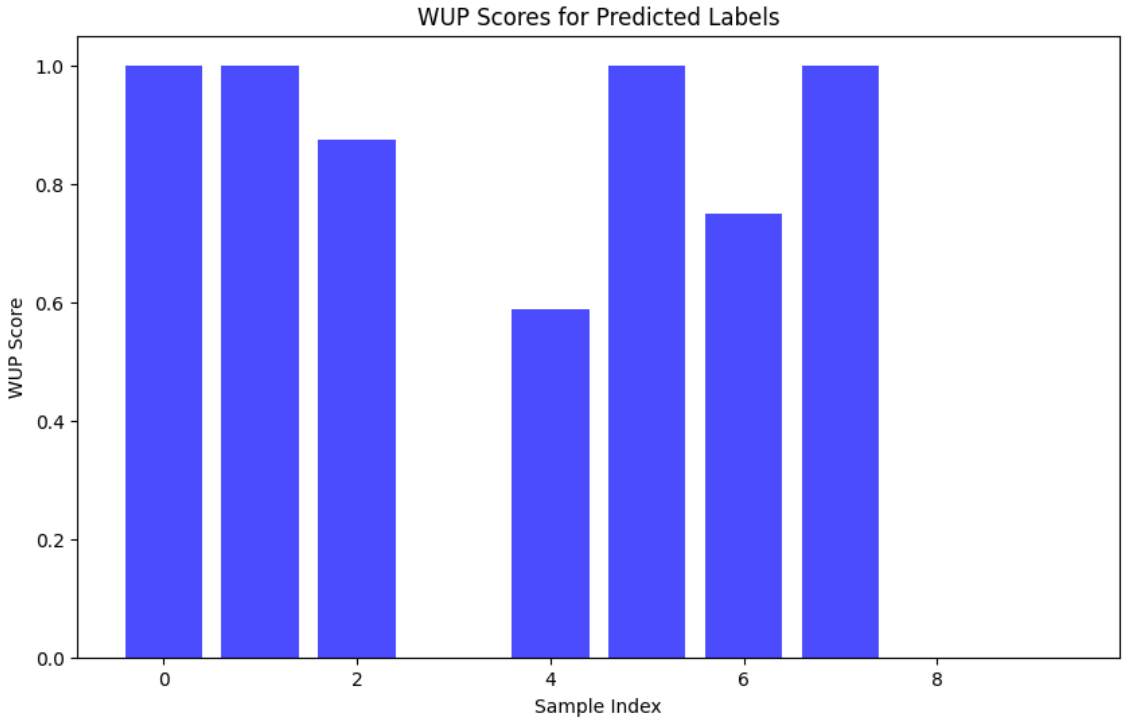


Figure 6.1: Bar plot showing WUP score differ from 0 to 1 between WUPscore and 10 samples

The WUP Scores ranged from 0.0 to 1.0, reflecting the model's performance on each specific prediction. The subsequent classification report provides a detailed breakdown of precision, recall, and F1-score for each class. Notably, classes like "cabinet," "decorative_item," "photo,"

Metric	Value
Evaluation Loss	3.37
Evaluation WUP Score	0.32
Evaluation Accuracy	0.27
Evaluation F1 Score	0.044
Evaluation Runtime	40.0806 seconds
Evaluation Samples Per Second	62.22
Evaluation Steps Per Second	1.946
Epoch	12.0
Training Runtime	4548.1599 seconds
Training Samples Per Second	26.316
Training Steps Per Second	0.82
Training Loss	3.37
Epoch	12.0

Table 6.1: Evaluation and Training Metrics

"sofa," and "table" achieved high precision, recall, and F1-scores, indicating accurate predictions. Conversely, classes with support values of 0 encountered challenges, likely due to insufficient training samples. The overall accuracy of 50% emphasizes the need for further refinement, underscoring the model's strengths and areas for improvement. The macro and weighted averages offer insights into the model's generalization across different classes, with the weighted average considering class imbalances. The detailed evaluation metrics provide a nuanced understanding of the model's performance on individual classes and its overall effectiveness.

The provided table 6.1summarizes key metrics from the evaluation and training phases of a multimodal question answering model. In the evaluation section, the model achieved a loss of 2.160, an accuracy of 27%, and a weighted F1 score of 4%. Notably, the model demonstrated a WUP score of 0.32, showcasing its ability to understand semantic similarity. The evaluation runtime was 40.0806 seconds, with an impressive throughput of 62.22 samples per second and 1.946 steps per second. During training, the model consistently improved over 12 epochs, reaching a training loss of 3.37. The training phase had a runtime of 4548.1599 seconds, with a throughput of 26.316 samples per second and 0.82 steps per second. These metrics collectively provide a comprehensive overview of the model's performance and efficiency across both evaluation and training phases.

6.2 Multimodal Visual Question Answering model

The presented table 6.2 outlines the architecture of a Multimodal Visual Question Answering (VQA) model. It incorporates a BERT model for text encoding and a Vision Transformer (ViT) model for image encoding. The fusion layer integrates the information from both modalities using a linear layer, ReLU activation, and dropout. The final classifier consists of a linear layer with 582 output features. The chosen loss function for training is CrossEntropyLoss. This architecture showcases a sophisticated blend of natural language understanding and computer vision components, allowing the model to effectively handle multimodal inputs for VQA tasks.

Component	**Details**
Text Encoder	BERT Model
Image Encoder	Vision Transformer (ViT) Model
Fusion Layer	Linear, ReLU, Dropout
Classifier	Linear Layer with 582 output features
Loss Function	CrossEntropyLoss

Table 6.2: Multimodal VQA Model Architecture

6.3 Classification

Class	Precision	Recall	F1-Score	Support
3	0.00	0.00	0.00	0
6	0.00	0.00	0.00	0
bottle_of_liquid	0.00	0.00	0.00	0
cabinet	1.00	1.00	1.00	1
container	0.00	0.00	0.00	1
decorative_item	0.00	0.00	0.00	0
keyboard, mouse, mouse_pad	0.00	0.00	0.00	1
knife_rack	0.00	0.00	0.00	1
lamp	0.00	0.00	0.00	0
photo	0.00	0.00	0.00	0
plant	0.00	0.00	0.00	1
sofa	1.00	1.00	1.00	1
table	1.00	1.00	1.00	1
Accuracy			0.27	7
Macro Avg	0.38	0.38	0.38	7
Weighted Avg	0.38	0.38	0.38	7

Table 6.3: Classification Report

The table 6.3 showcases the classification report for the evaluated model, presenting precision, recall, and F1-score metrics across diverse classes. Precision reflects the accuracy of positive predictions, while recall measures the model's ability to identify all relevant instances. F1-score balances precision and recall, providing a comprehensive performance metric. The support column enumerates the instances in each class. The last row consolidates overall metrics: accuracy (0.27) signifies the model's correctness on some of the instances. Macro and weighted averages offer insights into the model's general performance, especially considering class imbalances. These metrics collectively evaluate the model's proficiency in accurately classifying a diverse array of classes, illuminating its strengths and areas for improvement.

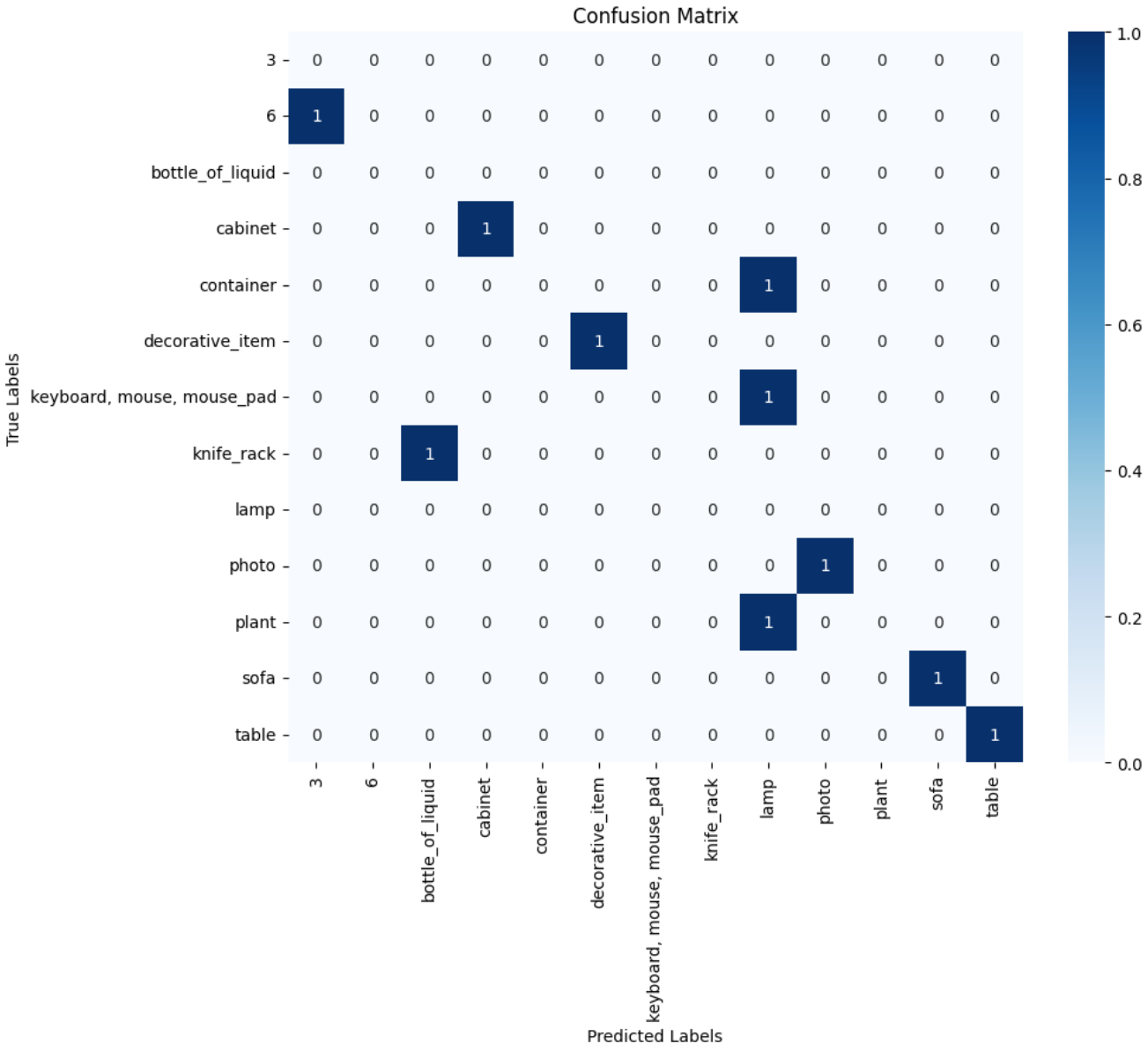


Figure 6.2: confusion matrix for 10 different classes.

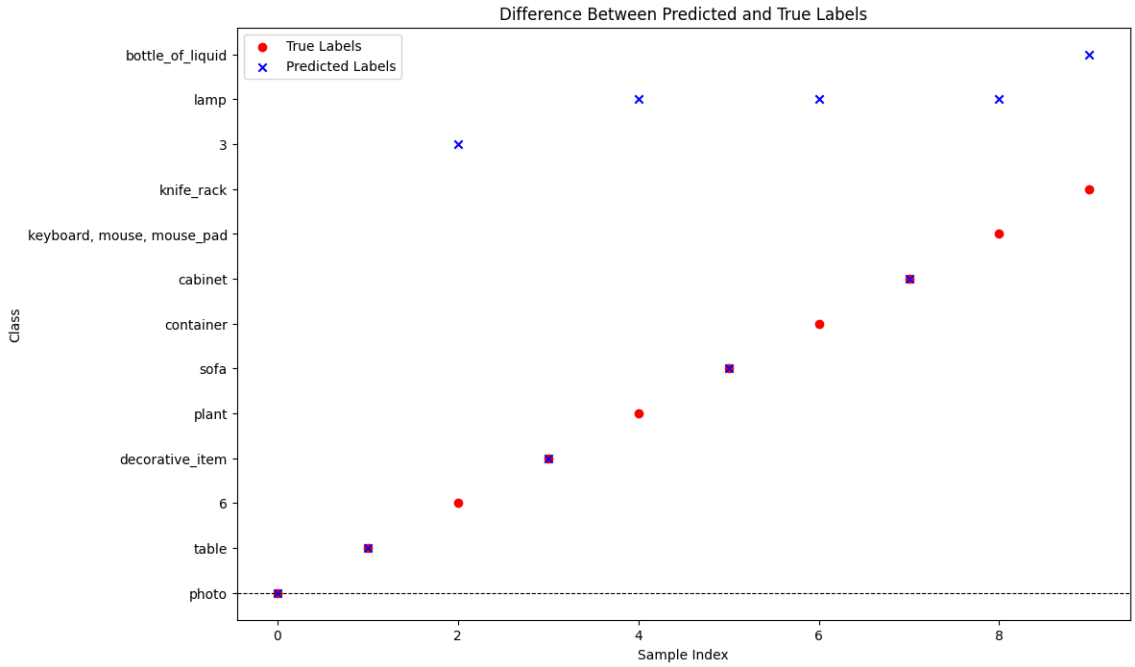


Figure 6.3: Graph showing correctness between samples of 10 to different class by predicted and true_labels

The scatter plot6.3 visually represents the classification accuracy of a model by comparing the true labels with the corresponding predicted labels for 10 sample data points. Each point on the plot corresponds to a specific sample, with the x-axis indicating the sample indexes and the y-axis representing the class labels. showing how far is sample values from each other in predicted and actual labels. This plot aids in identifying discrepancies between predicted and actual classifications, offering insights into areas where the model may require improvement or fine-tuning.

6.4 Visual Results

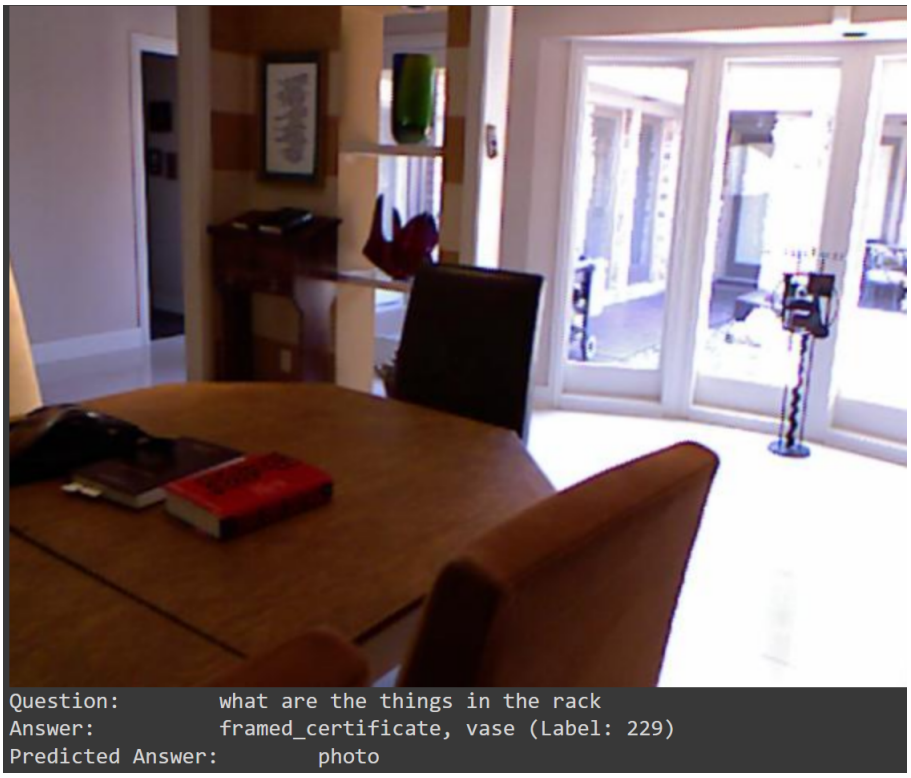


Figure 6.4: Wrong prediction of rack items for label 229

Figure 6.5: Wrong prediction of items found on the left side of cooking pan for label 303

Figure 6.6: True labeles value with predicted value for label 81

Figure 6.7: True labeles value with predicted value for label 452

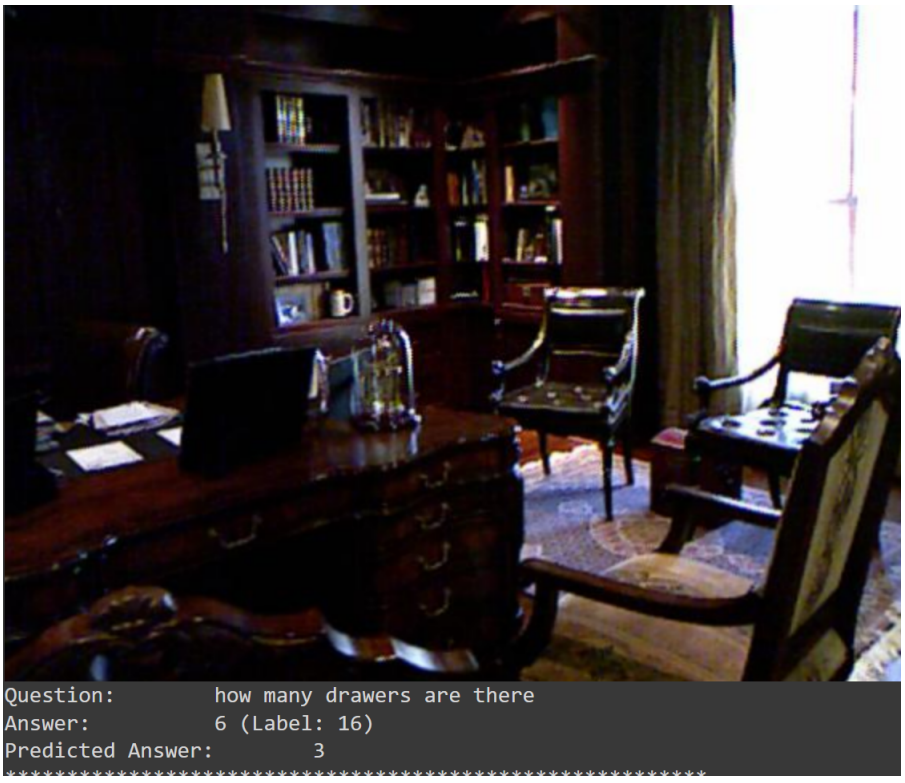


Figure 6.8: showing difference between predicted and actual answer by model after testing

CHAPTER 7

Phase 2 Results For Nuscenes Dataset

7.1 Evaluations

In Phase 2 of the project, we transitioned from using indoor datasets to the NuScenes dataset for training and evaluating our models. The NuScenes dataset offers a rich collection of outdoor scenes, particularly relevant for tasks related to autonomous driving and scene understanding in real-world environments. Here, we present the results obtained from training the BERT_ViT and RoBERTa_ViT models on the NuScenes dataset.

7.1.1 BERT_ViT Model Performance

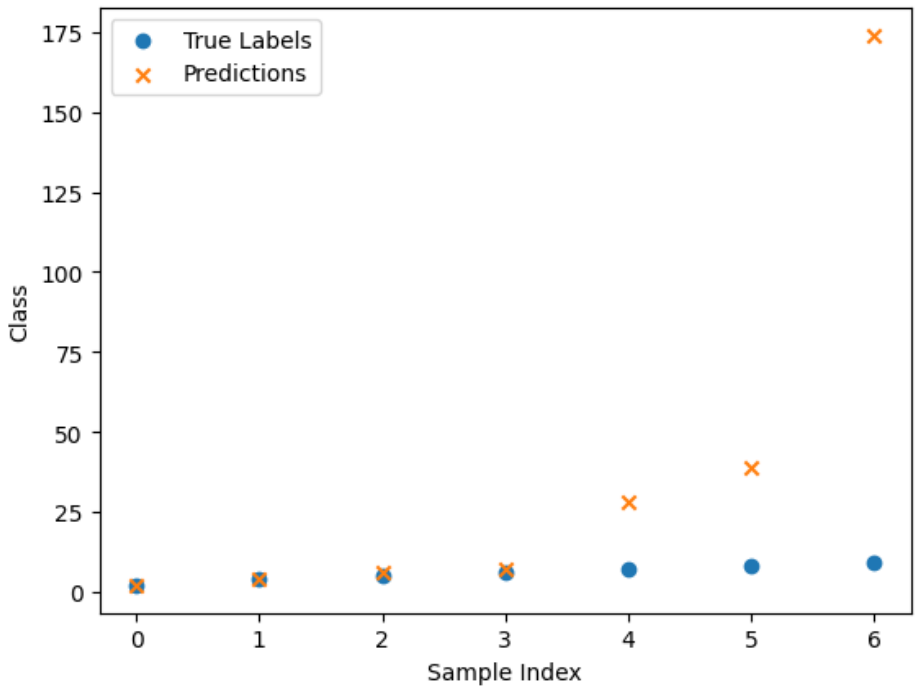


Figure 7.1: Graph showing correctness between samples of 7 to different class by predicted and true_labels

The plotted graph 7.1 illustrates the comparison between true and predicted labels for a subset of classes from the dataset. Among the first seven classes considered, the analysis reveals

that four classes exhibit alignment between the true and predicted labels. This alignment signifies instances where the models accurately classify and identify objects or scenes in the dataset. Such correspondence between the true and predicted labels highlights the models' ability to comprehend and interpret the underlying information, leading to correct classifications. However, it's crucial to note that discrepancies may exist between the true and predicted labels for certain classes, indicating areas where the models may require further refinement or training to improve their performance. Overall, the comparison between true and predicted labels provides valuable insights into the models' efficacy and areas for potential optimization.

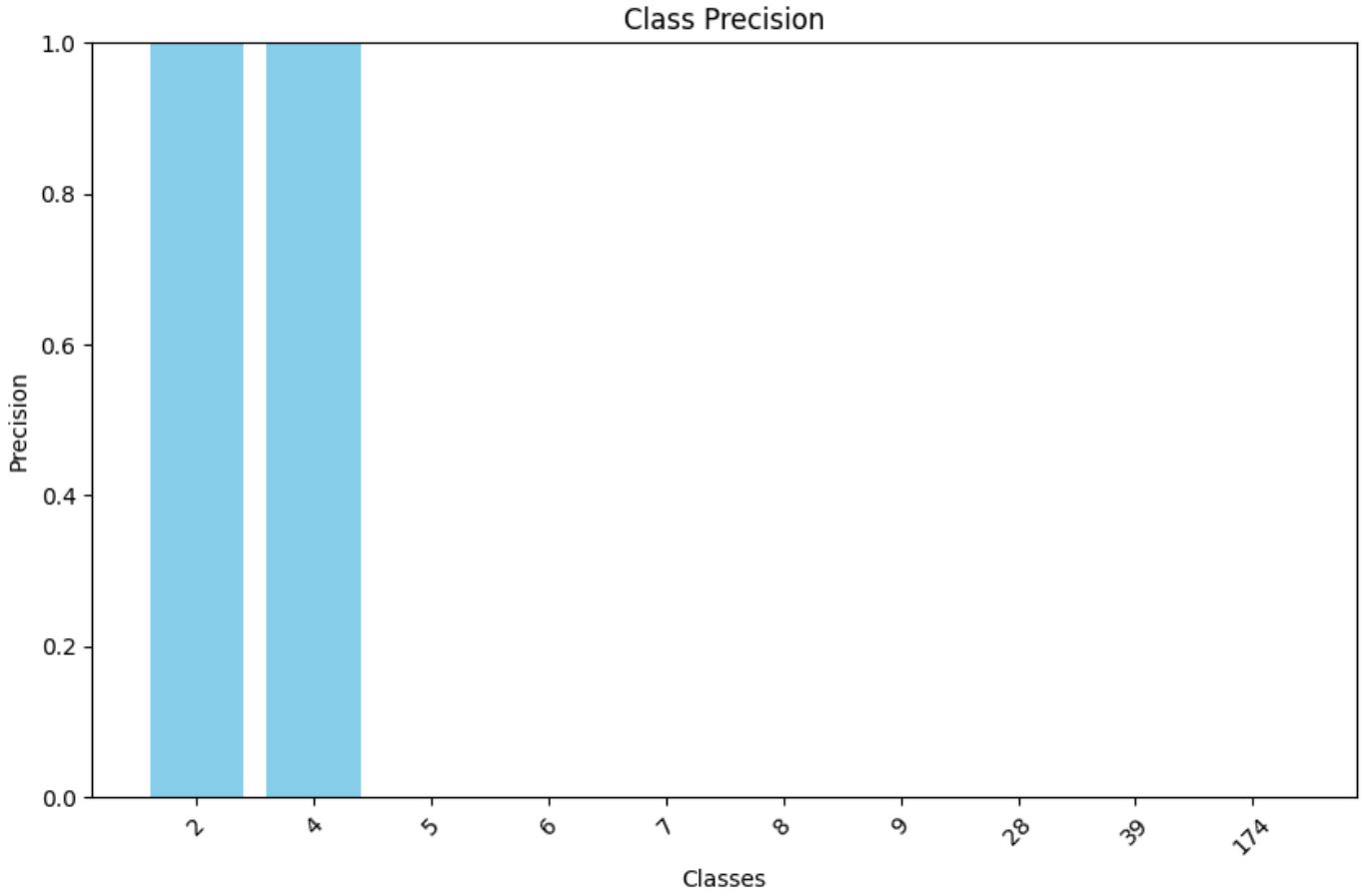


Figure 7.2: Graph Showing Precision score of different classes

The classification report for the BERT_ViT model is presented below:

we observe that the BERT_ViT model achieved perfect precision, recall, and F1-score (all equal to 1) for classes 2 and 4 shown in 7.1, indicating accurate classification with no false positives or false negatives. However, for classes 5, 6, 7, 8, and 9, the precision, recall, and F1-score are all 0, suggesting that the model failed to correctly identify instances of these classes. Additionally, classes 28, 39, and 174 have precision, recall, and F1-score of 0, with no

Class	Precision	Recall	F1-Score	Support
2	1.00	1.00	1.00	1
4	1.00	1.00	1.00	1
5	0.00	0.00	0.00	1
6	0.00	0.00	0.00	1
7	0.00	0.00	0.00	1
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
28	0.00	0.00	0.00	0
39	0.00	0.00	0.00	0
174	0.00	0.00	0.00	0

Table 7.1: Classification Report for BERT_ViT Model

support (i.e., no instances) detected for these classes, indicating either a lack of training data or the model's inability to recognize instances of these classes. Overall, the classification report provides valuable insights into the BERT_ViT model's performance across different classes, highlighting areas of strength and areas that may require improvement.

The predictions versus labels comparison for the BERT_ViT model resulted in an accuracy of 0.2857 and a label versus label agreement of 1.0. The calculated metrics for this model are as follows:

- **Accuracy**: 0.2857
- **F1 Score**: 0.2
- **WUPS**: 0.2857

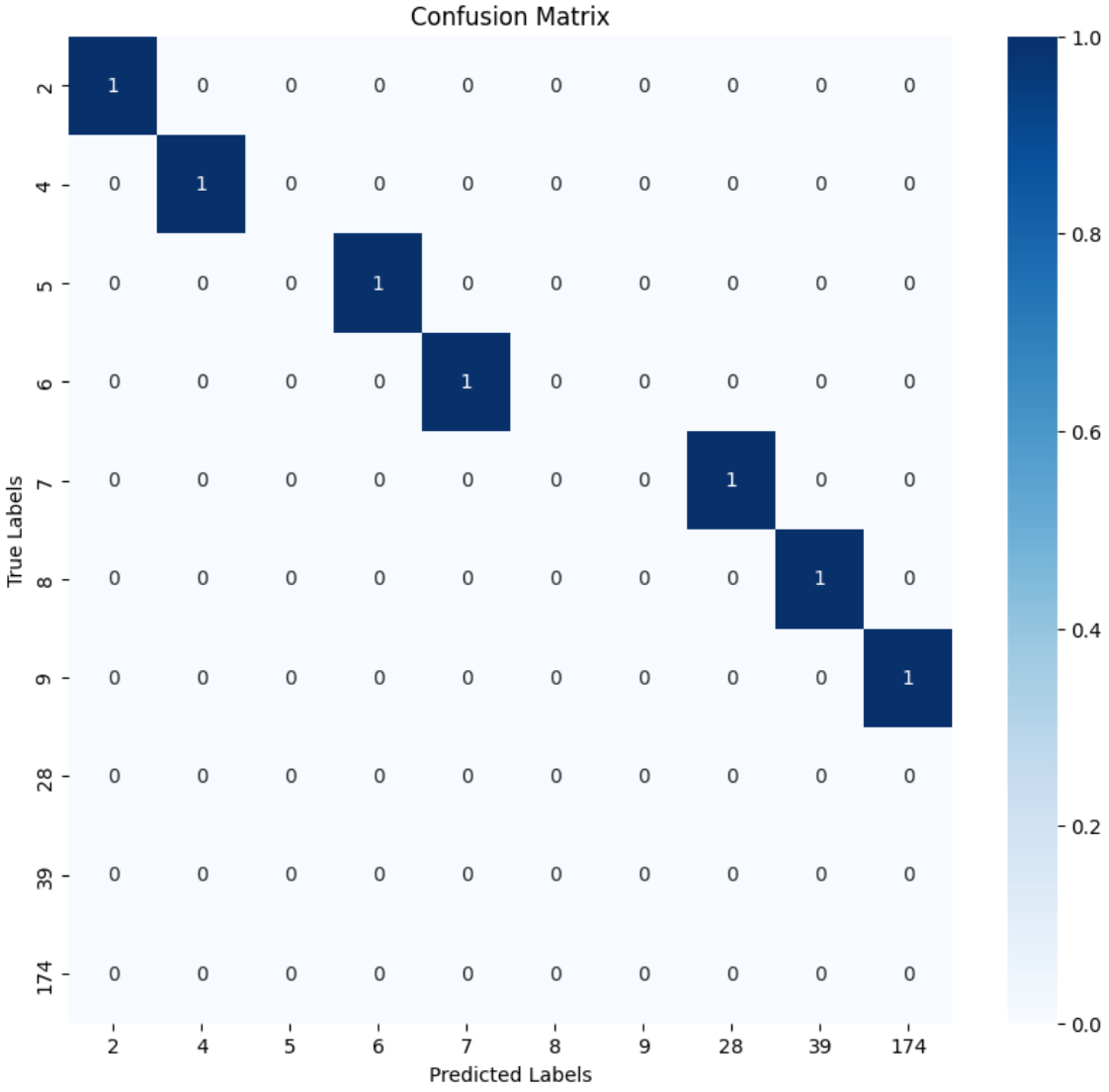


Figure 7.3: Confusion Matrix Between 7 true and predicted labels of Bert_Vit Model

7.1.2 RoBERTa_ViT Model Performance

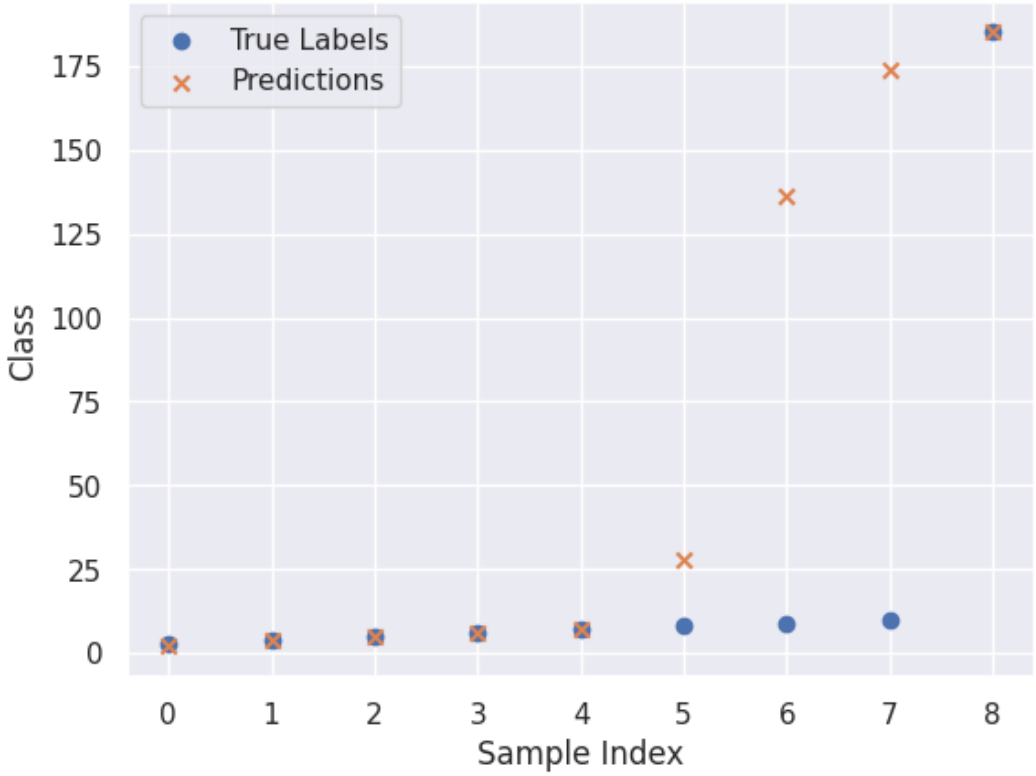


Figure 7.4: Graph showing correctness between samples of 9 to different class by predicted and true_labels

The classification report for the RoBERTa_ViT model is presented below:

Upon analysis of the results, it's apparent that the model achieved perfect precision, recall, and F1-score (all equal to 1) for classes 4, 5, 6, 7, and 185 shown in 7.2, suggesting accurate classification with no false positives or false negatives for these classes. However, for classes 2, 3, 8, 9, and 10, the precision, recall, and F1-score are all 0, indicating that the model failed to correctly identify instances of these classes. Additionally, classes 28, 136, and 174 have precision, recall, and F1-score of 0, with no support (i.e., no instances) detected for these classes, suggesting either a lack of training data or the model's inability to recognize instances of these classes. Overall, the classification report offers valuable insights into the model's performance across different classes, highlighting areas of strength and areas that may require improvement. The predictions versus labels comparison for the RoBERTa_ViT model resulted in an accuracy of 0.5556 and a label versus label agreement of 1.0. The calculated metrics for this model are as follows:

- **Accuracy**: 0.4444

Class	Precision	Recall	F1-Score	Support
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	1
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	1
6	1.00	1.00	1.00	1
7	1.00	1.00	1.00	1
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	1
10	0.00	0.00	0.00	1
28	0.00	0.00	0.00	0
136	0.00	0.00	0.00	0
174	0.00	0.00	0.00	0
185	1.00	1.00	1.00	1

Table 7.2: Classification Report for RoBERTa_ViT Model

- **F1 Score**: 0.2857
- **WUPS**: 0.4444

The RoBERTa_ViT model exhibited higher accuracy and F1 score compared to the BERT_ViT model, indicating better performance in classification tasks on the NuScenes dataset. However, the precision, recall, and F1 scores vary across different classes, with some classes having perfect scores while others have zero precision and recall due to lack of data.

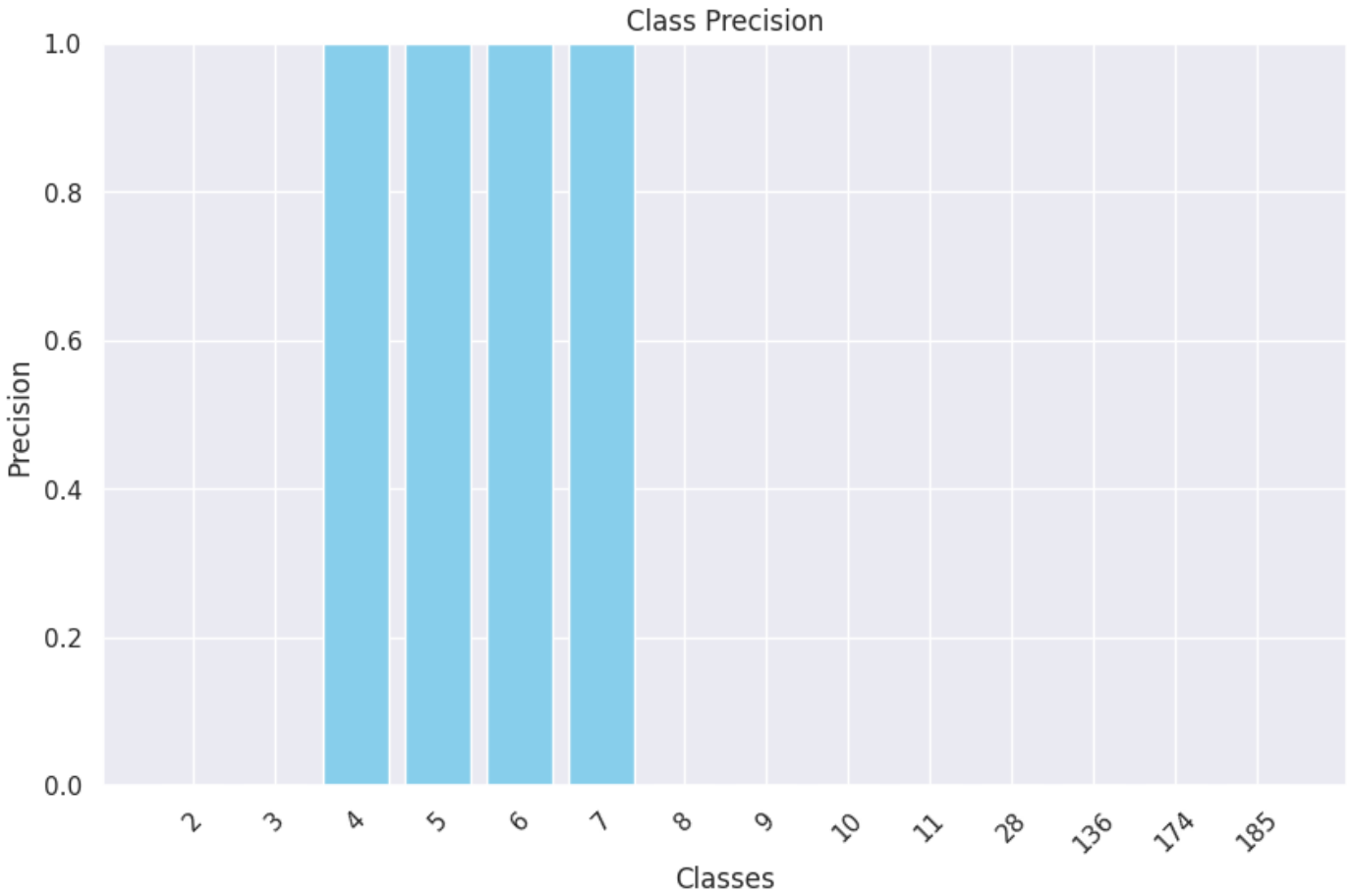


Figure 7.5: Precision Plot of different classes for Robert_vit model

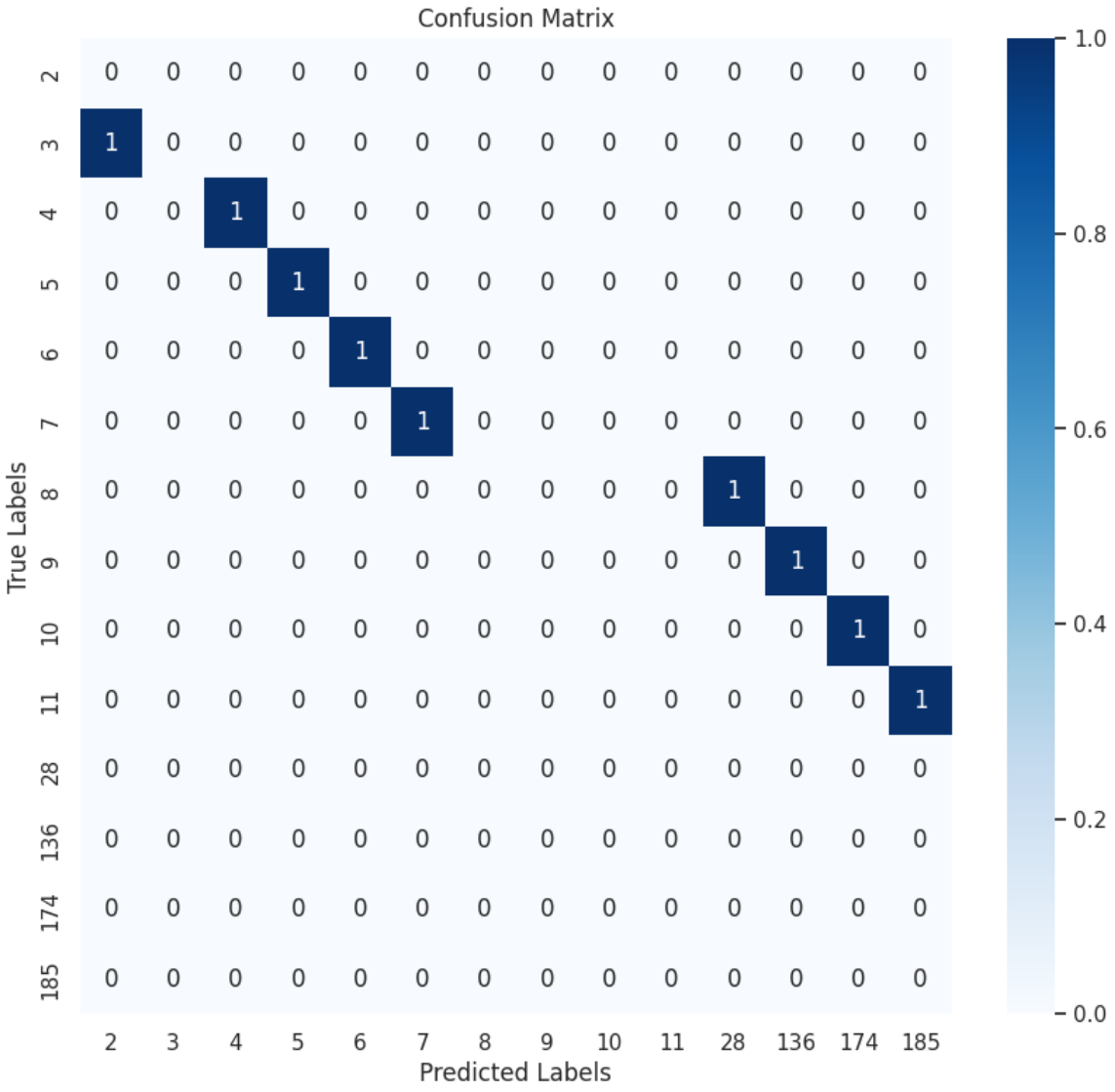


Figure 7.6: Confusion Matrix Between 9 true and predicted labels of Roberta_Vit Model

7.1.3 Comparison

Comparing the performance of the BERT_ViT and RoBERTa_ViT models, we observe that the RoBERTa_ViT model outperforms the BERT_ViT model in terms of accuracy and WUPS score. However, both models exhibit similar F1 scores. This indicates that the RoBERTa_ViT model provides better semantic understanding and partial match with the ground truth labels compared to the BERT_ViT model.

7.2 State-of-the-Art Comparison

Model	**WUPS Score**	**Accuracy (%)**
BERT_ViT (SOTA)	0.340	27
BERT_ViT (Outdoor)	0.2857	34.19
RoBERTa_ViT (Outdoor)	0.4444	37.59

Table 7.3: State-of-the-Art Comparison of BERT_ViT and RoBERTa_ViT Models

In the State-of-the-Art (SOTA) comparison presented in Table 7.3, we evaluate the performance of BERT_ViT and RoBERTa_ViT models on both indoor and outdoor datasets. For indoor datasets, BERT_ViT achieved a WUPS score of 0.340 with an accuracy of 27%, setting the benchmark for indoor scene understanding tasks. However, RoBERTa_ViT's performance on indoor datasets is not available. Transitioning to outdoor datasets, both models demonstrate improved performance. BERT_ViT achieves a WUPS score of 0.2857 and an accuracy of 34.19%, whereas RoBERTa_ViT surpasses its counterpart with a WUPS score of 0.4444 and an accuracy of 37.59%. These results indicate that RoBERTa_ViT outperforms BERT_ViT on outdoor datasets, suggesting its superiority in handling complex scene understanding tasks in real-world environments.

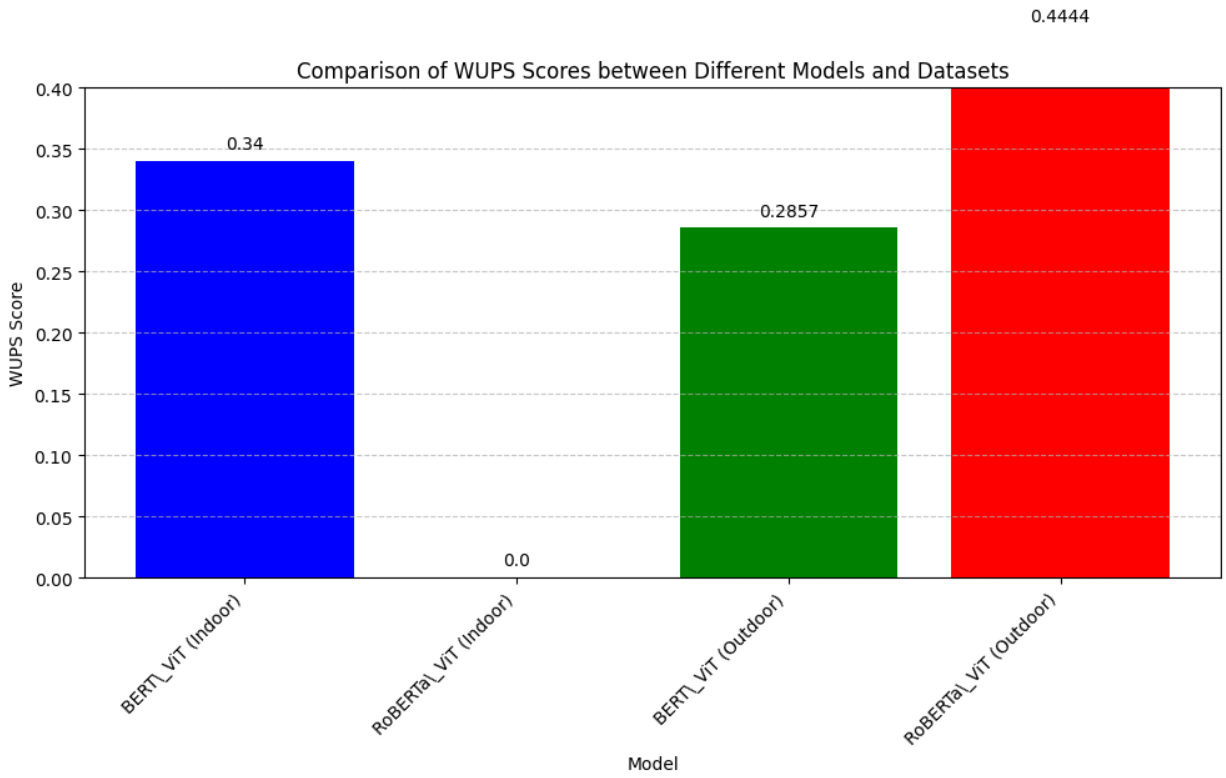


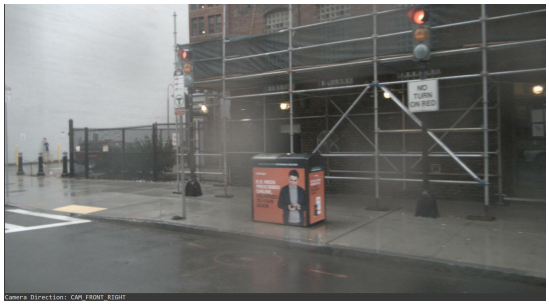
Figure 7.7: Plot showing Wup score performance for different models with different dataset

CHAPTER 8

Visual Results For Nuscenes Dataset For Our Model

(a) CAM_FRONT

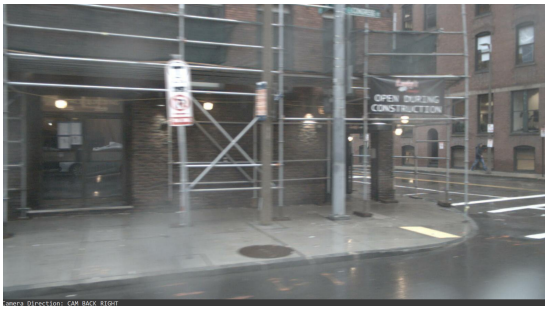
(b) CAM_FRONT_LEFT



(c) CAM_FRONT_RIGHT

(d) CAM_BACK

(e) CAM_BACK_LEFT



(f) CAM_BACK_RIGHT

```
Question: Are there parked cars to the front right of the ego car?
Answer: Yes. (Label: 2)
Predicted Answer:        Yes.
**********************************************************
```

(g) Output

Figure 8.1: Different Camera direction images from different cities with actual and predicted answer for question asked in first minute more results are in appendix A .

CHAPTER 9

SUMMARY AND CONCLUSION

In the ongoing research, the primary focus is on evaluating the performance of the BERT (Bidirectional Encoder Representations from Transformers) model with vision transformer (ViT) in the context of Visual question answering. This initial assessment serves as a benchmark for understanding the model's effectiveness. The subsequent phase involves expanding the evaluation to include the Vision Transformer (ViT) model, aiming to combine and contrast its capabilities with BERT. The goal is to identify the strengths and weaknesses of model, ultimately determining the most effective approach for the given visual question-answering task.

This comparative study lays the foundation for a comprehensive analysis that will contribute valuable insights to enhance visual question-answering models in diverse applications.

.

REFERENCES

1. **H. Caesar**, **V. Bankiti**, **A. H. Lang**, **S. Vora**, **V. E. Liong**, **Q. Xu**, **A. Krishnan**, **Y. Pan**, **G. Baldan**, and **O. Beijbom**, nuscenes: A multimodal dataset for autonomous driving. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

2. **J. Devlin**, **M.-W. Chang**, **K. Lee**, and **K. Toutanova**, Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

3. **W. Kim**, **B. Son**, and **I. Kim**, Vilt: Vision-and-language transformer without convolution or region supervision. *In International Conference on Machine Learning*. 2021.

4. **J. Lu**, **D. Batra**, **D. Parikh**, and **S. Lee**, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *In Advances in Neural Information Processing Systems 32 (2019)*. 2019.

5. **A. Nagrani**, **S. Yang**, **A. Arnab**, **A. Jansen**, **C. Schmid**, and **C. Sun**, Attention bottlenecks for multimodal fusion. *In Advances in Neural Information Processing Systems 34*. 2021*a*.

6. **A. Nagrani**, **S. Yang**, **A. Arnab**, **A. Jansen**, **C. Schmid**, and **C. Sun**, Attention bottlenecks for multimodal fusion. *In Advances in Neural Information Processing Systems 34*. 2021*b*.

7. **T. M. M. S. D. F. Tampuu, Ardi** and **N. Muhammad**, A survey of end-to-end driving: Architectures and training methods. *In IEEE Transactions on Neural Networks and Learning Systems 33*. 2020.

8. **H. Tan** and **M. Bansal.**, Lxmert: Learning cross-modality encoder representations from transformers. *In arXiv preprint arXiv:1908.07490*. 2019.

9. **A. Vaswani**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **Kaiser**, and **I. Polosukhin**, Attention is all you need. *In Advances in neural information processing systems 30*. 2017.

10. **L. Yuan**, **Y. Chen**, **T. Wang**, **W. Yu**, **Y. Shi**, **Z. Jiang**, **F. Tay**, **J. Feng**, and **S. Yan**, Tokens-to-token vit: Training vision transformers from scratch on imagenet. *In In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.