RETRIEVING VEHICLES THROUGH NATURAL LANGUAGE QUERIES

Professional Practice/Seminar (IT890) Project Report
Submitted in partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

CHIRAG BAVISHI (222IT005)

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE -575025

APRIL, 2023

DECLARATION

I hereby *declare* that the *Professional Practice/Seminar (IT890) Project Work Report* of the M.Tech.(IT) entitled ***'Retrieving vehicles through natural language queries'*** which is being submitted to the National Institute of Technology Karnataka Surathkal, in partial fulfillment of the requirements for the award of the Degree of Master of Technology in the department of Information Technology, is a ***bonafide report of the work carried out by me***. The material contained in this project report has not been submitted to any University or Institution for the award of any degree.

(Chirag Bavishi-222IT005)

____

Department of Information Technology

Place : NITK, SURATHKAL

Date :

CERTIFICATE

This is to *certify* that the Professional Practice/Seminar (IT890) Project Work Report entitled '***Retrieving vehicles through natural language queries'*** submitted by Chirag Bavishi(222IT005) as the record of the work carried out by him/her, is *accepted as the Professional Practice/Seminar (IT890) Project Work Report* submission in partial fulfilment of the requirements for the award of degree of Master of Technology in the Department of Information Technology.

Dr. Sowmya Kamath

____

ABSTRACT

Retrieving vehicles through natural language queries involves the development of sophisticated systems that can understand and process human language requests for information about vehicles. These systems employ advanced natural language processing techniques such as syntactic and semantic analysis, named entity recognition, and machine learning to identify and extract relevant information from a user's query. The use of natural language queries provides users with a more intuitive and convenient way to search for information about vehicles, as they can express their queries in a more conversational way. Such systems have important applications in industries such as automotive sales and marketing, customer service, and transportation logistics, where they can improve efficiency and streamline communication. As the technology continues to develop, the potential applications of retrieving vehicles through natural language queries are likely to expand, with the possibility of integrating these systems with voice assistants and other smart devices becoming increasingly feasible.

CONTENT

Chapter 1: Introduction

1.1 Introduction

The development of natural language processing technology has transformed the way we interact with computers and machines, making it possible to communicate with them in a more intuitive and conversational way. This technology has been applied to a wide range of domains, including customer service, healthcare, education, and finance, among others. In the automotive industry, natural language processing has significant potential to enhance the way we retrieve information about vehicles. Traditionally, searching for information about a vehicle has involved using specific keywords or phrases, which can be limiting and time-consuming.

However, with the development of systems that can understand and process natural language queries, users can now ask questions in a more conversational way, making the search for information about vehicles more intuitive and efficient. The goal of retrieving vehicles through natural language queries is to provide users with a more personalized and convenient experience, allowing them to access the information they need quickly and easily. This technology has important applications in industries such as automotive sales and marketing, customer service, and transportation logistics, where it can improve efficiency and streamline communication. As the technology continues to advance, the potential applications of retrieving vehicles through natural language queries are likely to expand, with the possibility of integrating these systems with voice assistants and other smart devices becoming increasingly feasible.

There are several natural language processing techniques available for retrieving vehicles through natural language queries, including syntactic and semantic analysis, named entity recognition, and machine learning.

Syntactic analysis involves identifying the grammatical structure of a sentence to understand the relationships between words and phrases. This technique is particularly useful for identifying the subject, verb, and object of a sentence, which can help determine the user's intent and the information they are looking for.

Semantic analysis, on the other hand, focuses on understanding the meaning of words and phrases, taking into account the context in which they are used. This technique can help identify synonyms and related concepts, allowing for a more nuanced understanding of the user's query.

Named entity recognition involves identifying specific entities, such as vehicle makes and models, mentioned in a user's query. This technique can help narrow down the search results to only those that are relevant to the user's request.

Deep learning models can be trained on large amounts of data, allowing them to learn complex patterns and relationships between words and phrases. This can enable the system to generate more accurate and personalized responses to user queries. Additionally, deep learning models can be designed to handle more complex queries and to identify subtle nuances in language, which can further improve the accuracy of the system.

One example of a deep learning-based technique for retrieving vehicles through natural language queries is the use of neural language models, such as BERT (Bidirectional Encoder Representations from Transformers)[1]. These models are designed to understand the context and meaning of words and phrases, allowing them to generate more accurate responses to user queries. By training these models on large datasets of natural language queries and vehicle information, they can be used to accurately and efficiently retrieve information about vehicles in response to user requests.

Overall, deep learning techniques have the potential to significantly improve the accuracy and efficiency of systems for retrieving vehicles through natural language queries, making them an important area of research and development in this field.

1.2 Motivation

The motivation for developing systems for retrieving vehicles through natural language queries is to provide users with a more intuitive and convenient way to access information about vehicles. Traditionally, searching for information about a vehicle has involved using specific keywords or phrases, which can be limiting and time-consuming. However, with the development of systems that can understand and process natural language queries, users can now ask questions in a more conversational way, making the search for information about vehicles more intuitive and efficient.

The use of natural language queries can provide a more personalized experience for users, allowing them to access the information they need quickly and easily. For example, a user may ask a system for recommendations on the best SUV for a family of five, and the system can generate a list of relevant options based on the user's preferences and needs.

In addition to improving the user experience, systems for retrieving vehicles through natural language queries have important applications in industries such as automotive sales and marketing, customer service, and transportation logistics. These systems can improve efficiency and streamline communication, allowing for faster and more accurate responses to user queries. For example, in automotive sales and marketing, a system that can accurately recommend vehicles based on a user's needs and preferences can lead to increased customer satisfaction and sales.

Furthermore, as the technology continues to advance, the potential applications of retrieving vehicles through natural language queries are likely to expand. For example, with the integration of voice assistants and other smart devices, users may be able to access information about vehicles through natural language queries without the need for a separate system or interface. This could have significant implications for the automotive industry, as it would allow for more seamless and convenient access to information about vehicles.

Chapter 2: Literature Survey

2.1 Outcome of Literature Survey

Retrieving vehicles through natural language queries has become an increasingly important problem in recent years due to the rise of voice assistants and chatbots. Traditional keyword-based search systems have limitations in understanding natural language queries and often produce inaccurate results. Deep learning techniques have shown great promise in improving the accuracy and efficiency of vehicle retrieval. This literature survey explores the different deep learning techniques used in this area and recent advancements in the field.

Neural language models have been shown to be effective in understanding natural language queries. BERT[1], GPT-2[3], and RoBERTa[2] are popular models used in this area. These models are trained on large datasets of natural language queries and vehicle information to accurately retrieve relevant vehicles in response to user requests.

Embedding techniques such as word2vec[4] and GloVe[5] are also used in this area. These techniques represent words and phrases in a high-dimensional vector space, allowing for more accurate understanding of natural language queries.

Convolutional neural networks (CNNs) have been used to extract relevant features from textual data and improve the accuracy of vehicle retrieval. Recurrent neural networks (RNNs) are also used in this area to capture temporal relationships between words and phrases and improve the accuracy of vehicle retrieval.

Attention mechanisms and transformer models have shown promise in improving the accuracy and efficiency of vehicle retrieval through natural language queries. These models allow the system to focus on the most relevant parts of the input and make better predictions.

Hao et al. [6] has proposed solution which transformer to generate synthetic data. In postprocessing stage, for each test picture, they have extracted features from the actual image (CVF2) with the compressed version (CFV2-C). Then rotated each of these photos horizontally and obtain two features. After this step we can obtain four characteristics altogether and then aggregate those to obtain the ultimate ReID feature. In the training part, they have implemented 2 stages i.e., CNN based model or Transformer based model and then training on those models separately and compared with each other. For CNN based model they have focused on ResNet-IBN, ResNet and ResNext-IBN.

Zhao et al. [7] propose a Symmetric Network with Spatial Relationship Modeling (SSM) method for vehicle retrieval using natural language. Which basically works on cross model i.e., combining of visual features of images or frames of vehicles and text features give as input. They have achieved 43.92% MRR accuracy in the AIcity contest. In first stage, Image augmentation is done, which will generate motion image by pasting cropped image of individual frames. In second stage, text augmentation is performed, which includes subject augmentation, motion augmentation and location augmentation. For vehicle characteristic learning they have used encoder like efficientNetB2, ibn-ResNet101-a pretrained over ImageNet dataset. Also used RoBERTa for finding text embedding which represents more values to the natural language text. This implementation got first rank in AICity competition with MRR score of 0.4392.

Shankaranarayan et al. [8] represents various methods available for vehicle retrieval using natural language. Generally, steps include in the problem are data preprocessing, feature learning, Metrix learning and post processing. He has mentioned different models available for each step and which provides better performance. During the feature learning phase for vehicle reidentification, models emphasizing both local and global differentiating factors fared significant on reidentification. It has also been discovered that models that include spatial and temporal features for vehicle retrieval outperform models that only use visual characteristics.

Shuai et al. [9] used 2 encoders for image feature extraction, one is for global feature extraction and second one is for local feature extraction. Global feature is like sequence of frames and how a vehicle is moving over the given environment and considering movement of nearby object. While local features are focused on particular vehicle, i.e., color of the vehicle, size of it, etc. another encoder is used for text. They have used RoBERTa as text encoder and ResNet101 as Image encoder. The proposed architecture is not giving expected output, getting 18.69% MRR accuracy only.

Author/Year	Methodology	Advantages	Limitation
Luo, Hao, Weihua Chen/ 2021	Identifying influence node by using local and global structure	Both cropping training data and using synthetic data can help the model learn more discriminative features.	Model structure is very complex, may take more time to train and validate.
Zhao, Chuyang, Haobo Chen/2022	Implemented Symmetric Network with Spatial Relationship Modelling (SSM) method for NL-based vehicle retrieval.	Then a spatial relationship modelling methods is proposed to make better use of location information.	-
Shankaranarayan, N., S. Sowmya Kamath/2022	A comprehensive review of multiple models addressing the vision-based vehicle retrieval is presented.	Pre-processing, Feature learning, Metric learning and post-processing can provide more efficiency.	-
Xu-Hua YangZhen Xiong/2021	Text encoder, local image encoder, global image encoder.	Used global encoder because natural language sentences not only contain the information of vehicle appearance but also describe the trajectories and background.	Could have used other than BERT model.

Table 1: Summery of the Literature Survey

2.2 Problem Statement

The problem is to develop a system that can accurately and efficiently retrieve vehicles in response to natural language queries, allowing users to easily find the vehicle they are looking for.

2.2.1 Objectives

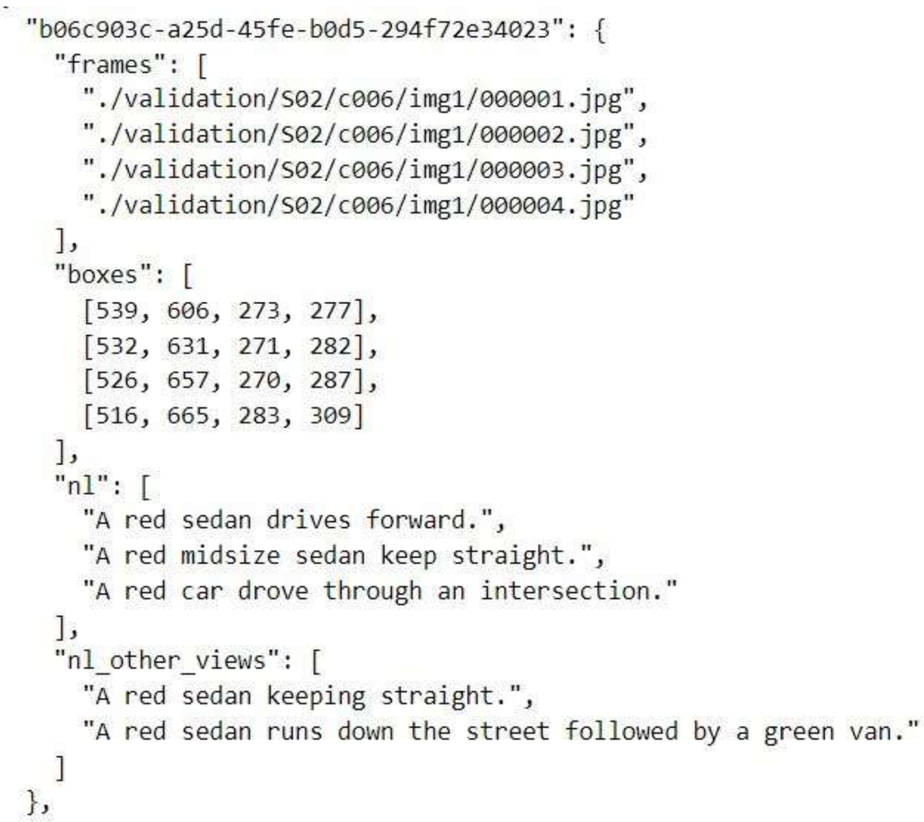
- Data preprocessing.
- To implement EfficientNet B0 to find image features.
- To extract NL features using ALBERT.
- To evaluate performance of the model with the base paper.

Chapter 3: Methodology

3.1 Dataset Detail

Dataset used for the current project is CityFLow-NL[10] taken from AICity competition 2023. The dataset includes approximately three hours of synchronized high-definition footage captured from 40 cameras stationed across 10 junctions, with a maximum distance of 2.5 kilometers between any two cameras recording simultaneously. Dataset comprises over 5,000 distinct and precisely NL descriptions of vehicle targets, which makes it the first multi-target multi-camera tracking dataset to our knowledge.

The dataset is divided into 2 parts, one is train and another is validation. In validation folder, there are total 2156 queries given. Each query is in JSON format containing some number of frames, and bounding box given respectively and 3 natural language sentences. It also has provided multiple extra descriptions for each trajectory, which are labelled "nl_other_views."



```
"b06c903c-a25d-45fe-b0d5-294f72e34023": {
  "frames": [
    "./validation/S02/c006/img1/000001.jpg",
    "./validation/S02/c006/img1/000002.jpg",
    "./validation/S02/c006/img1/000003.jpg",
    "./validation/S02/c006/img1/000004.jpg"
  ],
  "boxes": [
    [539, 606, 273, 277],
    [532, 631, 271, 282],
    [526, 657, 270, 287],
    [516, 665, 283, 309]
  ],
  "nl": [
    "A red sedan drives forward.",
    "A red midsize sedan keep straight.",
    "A red car drove through an intersection."
  ],
  "nl_other_views": [
    "A red sedan keeping straight.",
    "A red sedan runs down the street followed by a green van."
  ]
},
```

Fig 1: Representation of the Dataset

3.2 Architecture of proposed model

3.2.1 Data preprocessing

Dataset will have video along with all queries. The first task will be to extract all frames per queries given. As I mentioned earlier, there are bunch of frames given in every queries. So, we'll read the video from the path given in frame and convert it into list of frames.

In given example of the dataset, video path will be ". /validation/S02/c006", which is mentioned in the frames. After converting the video into list of frames, we will read frame number which are mentioned in the queries. In our example those will be 1, 2, 3 and 4.

We will need individual images of particular vehicle a query has mentioned. Dataset provides bounding box for the vehicle it is referring. So, we will crop that image from all frames for all queries.

3.2.1 Architecture of the model

Model will take 2 separate inputs; one is visual input which is sequence of frames and their cropped images of given vehicle. Another input is natural language. The output will be distance between visual features and text feature. The less difference is, the more closed related the vehicle is with the given text triplet

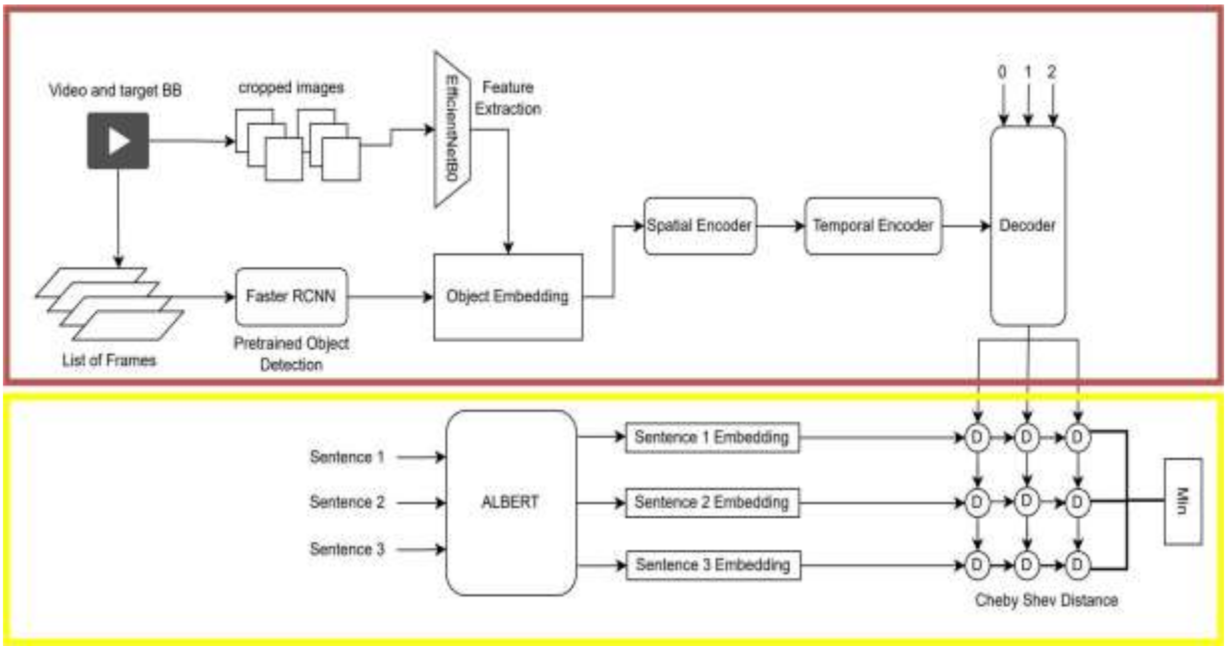


Fig 2. The main pipeline of the proposed method

In shorts, there are separate module which can run simultaneous. One performing Video Processing and another one is Natural Language Processing. Video Processing performs finding embedding of each frame of the video and extract feature by 2 encoders one by one. Both encoders extracting features like spatial features and temporal features. Spatial features are those which are related to the specific object, like vehicle's color, shape, etc. temporal features are taking consideration with respect to the time. By considering past and future positioning of the object. For example, if vehicle is turning left or right. If head light is turning on and off, etc. then comes transformer decoder which takes NL queries as input and provides 3 different embeddings.

On the other hand, Natural Language Processing will be handling feature extraction of NL queries. First of all, it will find text embedding and map to visual embeddings. At the end, I'll compare distance between 2 feature set. If the distance is less that means the given is image or frame is very much relatable to the NL queries.

We'll explore each component with more detail.

3.2.3 Computing Image Embedding

We implied Faster R-CNN for finding embedding of each frame provided by uuids of the training set. Faster R-CNN (Region-based Convolutional Neural Network)[11] is a deep learning algorithm for object detection in images. Faster R-CNN builds on the earlier R-CNN and Fast R-CNN models and improves the speed and accuracy of object detection by introducing a region proposal network (RPN) that shares convolutional layers with the detection network. The RPN generates a set of region proposals, which are candidate object bounding boxes, and these proposals are fed into the detection network to classify and refine the bounding boxes.

ResNet-50 is used as backbone of faster R-CNN. It achieves state-of-the-art accuracy on several benchmark datasets such as COCO and PASCAL VOC. This dataset is having 80 classes. In our case, we will select only few classes which are related to our work. Also, object with below 0.85 confidence score will not be selected.

This stage will take input as frames for all queries. End output of this stage will be vector of size (5 + 256). Where 5 represents bounding box of size 4 and class type of size 1. Rest 256 values represents feature of the object detected.

3.2.4 Convolutional Backbone

The convolutional backbone is simple CNN based on ResNet models. Before preceding to this step, we'll crop particular image of vehicle defined by boundary box in the training dataset for each frame. These set of cropped images will be given as input to this model. Each M cropped images will be set to fix size before passing it.

Given the sequence of the M sampled frames of a generic tracking sequence, the region delimited by the tracking bounding box is cropped from each frame, and the crops are resized to a common fixed size.

Once the patches are prepared, they are fed into the convolutional backbone, which is a ResNet-family convolutional network. The convolutional backbone processes each frame of the sequence individually along the temporal axis, meaning that it treats each frame as a separate input to the network.

To accomplish this, the frames are stacked along the temporal axis, forming a tensor of shape (M, 3, W, H), where M is the number of frames in the sequence, 3 is the number of color channels (e.g., RGB), and W and H are the width and height of the resized patches. Here we have resized the cropped images to size of (90, 110). The temporal axis is treated as the batch dimension, which means that each frame is processed individually as if it were a separate input in a batch of size M.

By processing each frame individually, the convolutional backbone is able to extract features that are specific to each frame of the sequence, which can be used to track the object across frames.

In the given experiment, I have used EfficientNetB0. EfficientNets[12], as the name implies, are extremely efficient computationally and achieved 84.4% top-1 accuracy on the ImageNet dataset. There are other model of EfficientNetB series which are from B1 to B7. The baseline model is EfficientNetB0 only.

In this experiment, we will compare performance of EfficentNetB0 with respect ResNet18. EfficientNetB0 is a larger model compared to ResNet18. The number of parameters in EfficientNetB0 is about 5 times more than ResNet18. EfficientNetB0 has shown better accuracy than ResNet18 on various image classification benchmarks, such as ImageNet.

3.2.5 Transformer: Spatial-Temporal Encoder

In this stage, we will use 2 identical transformer encoders sequentially for extracting spatial and temporal features from the embedding we got from previous stage. The first encoder combines the (O + 1) object embeddings from each of the M frames to form a single visual embedding, to offer an in-depth frame-level illustration. The another one combines information across the time axis to determine the sequential meaning of the picture shown.

First encoder, namely **spatial encoder** takes input as visual embedding produced in previous stage. As it is already mentioned, output of embedding is 261 (5+256). So we have to resize the input vector to 256 so it is divisible by the number of MHSA Heads used in the subsequent attention mechanism. So, the input will be (Bs, O, M, 256) where Bs is mini batch size and number of frames(M) will be set as additional batch size. O is object embedding with vector size of 256. The mean between the O's embeddings is computed to produce a single embedding for each of the M timesteps. In simpler terms, the Spatial Encoder takes in the visual input and applies a linear layer to reduce its dimensionality. It then applies an attention mechanism to extract the most important information from the input, while ignoring padding values. Finally, it computes the mean of the embeddings to produce a compact representation for each timestep.

Another encoder, **temporal encoder** takes input from previous encoder's output. The Temporal Encoder operates almost identically to the Spatial Encoder, but with one main difference: the definition of the Positional Encoding. Positional Encoding is a mechanism used in the Transformer architecture to inject information about the position of each element in the sequence. In the case of the Temporal Encoder, the position refers to the timestep or frame in the video sequence. Since the number of frames in a video sequence can be very large (up to 3620 frames), but the number of sampled frames used in the model is limited to a maximum of 80, the Temporal Encoder uses a modified version of Positional Encoding. Instead of the canonical sequence of incremental integers (0, 1, ..., M−1), the indices of the sampled timesteps are used as input to the positional encoding. The Temporal Encoder module helps the model learn and encode the temporal information in the video sequence, allowing it to better recognize and track objects across multiple frames.

3.2.6 Natural Language Embedding

This stage is responsible for encoding the textual descriptions of the objects in the video sequence and mapping them to the same latent space as the visual embeddings. To compare the visual input (the video sequence) with the textual description, we need to encode both in a common format so that we can compute a distance measure between them.

We will be using ALBERT[13], which is widely used for natural language processing tasks. It will provide natural language embedding with respect to the NL triplets. ALBERT (A Lite BERT) is a transformer-based language model proposed in 2019 as an improved version of BERT (Bidirectional Encoder Representations from Transformers), which is a widely used pre-trained language model that has achieved state-of-the-art results in many NLP tasks. The main idea behind ALBERT is to reduce the number of parameters in BERT while maintaining its performance or even improving it. To achieve this, ALBERT uses a factorized embedding parameterization technique and cross-layer parameter sharing, which enables sharing of parameters across layers of the model.

The factorized embedding parameterization technique reduces the number of parameters in the embedding layer by factorizing the large embedding matrix into two smaller matrices, one for word embeddings and the other for positional embeddings. This reduces the memory footprint of the model, which enables training on larger batch sizes and accelerates the training process.

In this experiment, we will compare both model and how to performance varies in this vehicle retrieval problem.

3.2.7 Distance Function

This is the final stage, which is trained to find optimal distance between 2 embedding i.e., visual embeddings and NL embeddings. Once after training, weight be set so that most likely embedding have less distance. The more relatable text to the frame, the less distance is.

In this experiment, distance is measure by **Chebyshev Distance**[14]. Chebyshev distance is defined as the maximum absolute difference between the coordinates of two points. In other words, it is the length of the shortest path between the two points if only horizontal, vertical and diagonal movement is allowed.

Formally, if we have two points (x1, y1) and (x2, y2), then the Chebyshev distance between them is given by:

D(x1, y1, x2, y2) = max(|x1 - x2|, |y1 - y2|)

Chebyshev distance can be a better choice in situations where we want to measure the distance between two points in a grid-like system, such as in image processing, board games, or graph theory. It can also be useful in situations where we want to find the closest point to a given query point in a dataset. One of the advantages of Chebyshev distance is that it takes into account all possible moves in a grid-like system. For example, if we want to measure the distance between two points in a chessboard, the Chebyshev distance will consider all possible moves a chess piece can make, such as moving diagonally or horizontally.

However, one limitation of Chebyshev distance is that it does not take into account the actual distance between two points. For example, the Chebyshev distance between two adjacent cells in a grid is the same as the Chebyshev distance between two cells that are far apart diagonally. Therefore, it may not be a good choice in situations where the actual distance between two points is important. So, we'll be comparing with Euclid distance in this experiment.

3.3 Evaluation Metrics

MRR is a metric that measures the quality of the ranked list of retrieved vehicles. It takes into account the rank of the first relevant vehicle in the list. The reciprocal rank of the first relevant vehicle is calculated and averaged over all the queries in the test set. MRR gives an idea of how well the system is able to retrieve the most relevant vehicle for a given query. A higher MRR score indicates better performance.

The vehicle retrieval by NL descriptions tasks generally utilizes the Mean Reciprocal Rank (MRR) as the standard metrics. The formula is given below:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i},$$

Where |Q| is the number of sets of text descriptions. Rank_i refers to the rank position of the right track for the i^{th} text description.

Recall is a commonly used evaluation metric in information retrieval and machine learning tasks. It is a measure of the fraction of relevant items that are retrieved by a system, out of all the relevant items in the system.

Recall @5 is a metric that measures the proportion of relevant vehicles that are retrieved in the top 5 results of a query. In other words, it measures how well the system is able to retrieve at least one of the top 5 relevant vehicles for a given query. Recall @5 is important because users typically only look at the top few results of a search. If the system can retrieve relevant vehicles in the top 5 results, it is more likely to be useful to the user.

Recall @10 is similar to Recall @5, but it measures the proportion of relevant vehicles that are retrieved in the top 10 results of a query. It provides a broader view of the system's ability to retrieve relevant vehicles, and is particularly important for queries that have a large number of relevant vehicles.

Chapter 4: Experiments and Results

We experimented the project under CPU configured machine. System specification is given in below table.

Specification	Value
Processor	Intel core i7, 3.00GHz x8
RAM Size	32GB
Disk Size	512GB (SSD)
OS Name	Ubuntu 20.04.6 LTS
OS Type	64-bit

Table 2: System specification

We have divided training data in 90:10 ratio for training and testing. Testing dataset have 215 different queries. we first compute all the possible visual embeddings Vi and language embeddings Ti for i in (0, 215). Since the two branches of the model are independent, the number of inference steps grows linearly in the number of sequences to be tested.

I have used batch size of 8 while training the model. Also, I have used Adam optimizer in the convolution module and Learning rate of 0.001.

For each visual embedding V^{i1} we can compute the distance $d(V^{i1}, T^{i2})$ for each i_2 in the dataset. At this point, for each Vi we sort the distance to the descriptions in descending order. In this way, it is possible to compute the Mean Reciprocal Rank (MRR) metric. Where d is defined as Chebyshev distance of 2 embeddings.

4.1 Base paper implementation

The base paper[15] have implemented the model with BERT to extract NL embeddings. They also have used resnet18 as convolution backbone which is little heavier that EfficientNetB0 model. They have used Euclid Distance for comparing two embeddings i.e., visual embeddings and NL embeddings.

They have trained model of 680 epochs on 2 Nvidia A100 GPUs with a mini-batch size of 48 per GPU. Learning rate, they have taken is 3.5e^{-5}.

4.2 Performance Analysis

In Table 2 we have mentioned MRR, Recall @5 and Recall @10 for few different cases. Where first row is base model with convolution backbone of resnet18, BERT for natural language embeddings and distance function of Euclid distance. We have compared base model with other models and performance metrics is given in the same table.

#Epoch	NLP Emb.	Convolution Backbone	Distance Function	MRR	Recall@5	Recall@10
2	BERT	Resnet18	Euclid	0.0287	0.0233	0.0372
2	ALBERT	Resnet18	ChebyShev	0.0331	0.0233	0.0651
2	BERT	EfficientNetB0	ChebyShev	0.0317	0.0326	0.0651
2	ALBERT	EfficientNetB0	ChebyShev	0.0335	0.0186	**0.0651**
4	ALBERT	Resnet18	ChebyShev	**0.0362**	**0.0372**	0.0605

Table 3: Results obtained by various models

As we can see in the above table, ALBERT is providing more accuracy (MRR) as compare to BERT. The key difference between BERT and ALBERT is in their architecture. BERT uses a fixed architecture where all the layers have the same number of parameters. On the other hand, ALBERT uses a parameter-sharing approach where the model shares parameters between layers. This results in a more efficient and compact model, which requires fewer parameters and less computational resources than BERT.

We also compared the change in performance if convolution backbone is changed, i.e., instead of using resnet18, we have used EfficientNetB0 which is lighter than ResNet18. EfficientNetB0 is a smaller and more efficient model than ResNet18, with fewer parameters and faster inference times.

In all model, I have used distance function as Chebyshev distance. In base paper model, they have used Euclid distance as distance function.

It is clearly showing that ALBERT and EfficientNetB0 is performing better than base model. MRR is higher than base model’s MRR. We can improve the model accuracy by training for more epochs.

Chapter 5: Conclusion and Future Work

In conclusion, the use of natural language to retrieve vehicles is a promising area of research that can greatly enhance the user experience in the automotive industry. The combination of visual embeddings and natural language embeddings using state-of-the-art models such as EfficientNetB0 and ALBERT can significantly improve the accuracy of vehicle retrieval based on user queries. EfficientNetB0 can be used to extract visual embeddings from vehicle images, while ALBERT can be used to generate natural language embeddings from user queries. These embeddings can then be used to match the user query with the closest vehicle images in the database.

Future work in this area can focus on further improving the accuracy of vehicle retrieval using more advanced techniques such as multi-modal embeddings that incorporate both visual and natural language information. Additionally, integrating other types of data, such as user preferences and location, can further enhance the relevance of the retrieved vehicles. We can opt for other version of EfficeintNet B series, which are heavier model than B0.

Chapter 6: References

[1] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[2] Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

[3] Budzianowski, Paweł, and Ivan Vulić. "Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems." *arXiv preprint arXiv:1907.05774* (2019).

[4] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[5] Huynh, Su V. "A strong baseline for vehicle re-identification." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4147-4154. 2021.

[6] Luo, Hao, Weihua Chen, Xianzhe Xu, Jianyang Gu, Yuqi Zhang, Chong Liu, Yiqi Jiang, Shuting He, Fan Wang, and Hao Li. "An empirical study of vehicle re-identification on the AI City Challenge." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4095-4102. 2021.

[7] Zhao, Chuyang, Haobo Chen, Wenyuan Zhang, Junru Chen, Sipeng Zhang, Yadong Li, and Boxun Li. "Symmetric network with spatial relationship modeling for natural language-based vehicle retrieval." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3226-3233. 2022.

[8] Shankaranarayan, N., and S. Sowmya Kamath. "Deep Vision based Vehicle Retrieval for Automated Smart Traffic Surveillance Systems." In *2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pp. 1-8. IEEE, 2022.

[9] Bai, Shuai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Hongxia Yang, and Yi Yang. "Connecting language and vision for natural language-based vehicle retrieval." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4034-4043. 2021.

[10] Feng, Qi, Vitaly Ablavsky, and Stan Sclaroff. "Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions." *arXiv preprint arXiv:2101.04741* (2021).

[11] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[12] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International conference on machine learning*, pp. 6105-6114. PMLR, 2019.

[13] Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).

[14] Coghetto, Roland. "Chebyshev distance." *Formalized Mathematics* 24, no. 2 (2016): 121-141.

[15] Scribano, Carmelo, Davide Sapienza, Giorgia Franchini, Micaela Verucchi, and Marko Bertogna. "All you can embed: Natural language based vehicle retrieval with spatio-temporal transformers." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4253-4262. 2021.