Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23)
Brunel University, London, United Kingdom - August 03-05, 2023
Paper No. MVML 108
DOI: 10.11159/mvml23.108

NLP-based Traffic Scene Retrieval via Representation Learning

Touseef Sadiq[1], Christian W. Omlin[1]
[1]Center for AI Research, University of Agder
Jon Lilletuns vei 9, 4879 Grimstad, Norway
touseef.sadiq@uia.no; christian.omlin@uia.no

Abstract **-** Many automated systems require the interpretation of visual information, i.e., images, videos, and natural language input, i.e., speech or text, to comprehend their surroundings and communicate with interacting humans. One such hybrid application of computer vision using images and videos and natural language processing (NLP) recognizes traffic scenes, a crucial and challenging problem in automated transportation systems. Scene classification is just one of many areas where recent convolutional neural network (CNN) frameworks have proven to be highly effective. Still to be fully explored for application to problem-solving in the real world is CNN's impressive, truly representative learning capability. However, newer CNN implementations, such as YOLO and DeepSort, show promise for object detection. The BERT model is the benchmark for text embeddings and the most efficient method currently available. Hence, we aim to retrieve the vehicles from the traffic videos using natural language-based description, i.e., text. The paper proposes a novel approach that combines YOLOv7, the recent version of YOLO, DeepSort algorithms for object detection, i.e., detecting the vehicles from the traffic scene from the frames of the videos and the transfer learning model, i.e., BERT model for text embeddings. Additionally, a Kalman filter is utilized to track the cars by providing the id and will retain them in the other frames of the videos. The machine learning model performs the similarity checking, i.e., siamese neural networks. The experiments are performed on the standard dataset of AI city challenge 2022. Moreover, the results depict that the proposed approach achieves 28.49 % of Recall@5, 42.08 % of Recall@10, and 20.73 % of MRR, indicating the proposed method's effective approach.

Keywords**:** Computer vision, BERT, neural networks, CNN, YOLO, and Deep Sort.

1. Introduction

Traffic videos are a valuable resource for urban city planning because they can be used to investigate and comprehend the correlations of traffic. Nevertheless, because of the exponential growth of the data, storing traffic videos can be a difficult task. As a result, extracting the traffic video activity from the huge-scale stream is difficult. The remarkable ability of human beings to categorize complex traffic scenes accurately and quickly is crucial for inferring the traffic situation and navigating the dynamic driving environment. For this reason, it will be a significant step forward to develop automatic traffic scene recognition software that can mimic human performance. Such a system will significantly benefit numerous applications, including autonomous vehicles, traffic mapping, and surveillance. Improving traffic flow during peak hours is just one example of how the automatic acquisition of data from real-world traffic scenes will play a crucial role in the future of traffic management [1].

Although image-based vehicle retrieval systems are the norm, text-based traffic scene retrieval technology has received significantly more attention in the research community. The target of interest can be described using easily accessible natural descriptions in text-based retrieval systems rather than image-based frameworks, which necessitate at least an image of the target. Text queries are not nearly as effective as image queries describing subtle differences in appearance. Furthermore, they are less complicated, more intuitive, and easier to use, and they can rapidly provide additional layers of characterizations like shape, appearance, location, and relevance to another target. However, image queries excel at describing broad visual characteristics. [2].

More than two decades have been devoted to studying image representation, during which several practical feature extraction algorithms have been proposed for extracting image features. The rich variations hidden in the data cannot be reflected by hand-crafted features, despite their limited success.

Thus, here comes the role of deep learning (DL), one of the machine learning (ML) approaches that has garnered much interest in recent years for applying it to real-world problems, is deep learning. A DL framework is a group of ML algorithms that takes its cues from the structure of the human brain. These algorithms manage data by transforming it through various representations and organizational structures. The application of deep learning algorithms in the NLP-based image retrieval domain to close the semantic chasm is motivated by their success in other areas (such as object recognition). DL can map input and output data without human-provided features [3]. CNN performs outstandingly in computer vision applications such as face recognition, object detection, and semantic segmentation [4]

Hence, nowadays, deep learning-based algorithms like convolutional neural networks (CNN) have made great strides in automatically extracting the features from images and proving to be the state-of-the-art algorithm for representing images in computer vision tasks. Because CNNs are trained with multiple convolution layers in an end-to-end architecture, they can recognize complex features. This ability makes CNN indispensable for many computer vision tasks, including scene recognition. Many standard examples, such as image classification using the ImageNet dataset, have demonstrated superior performance to previous work that used hand-crafted features [5]. ML and DL are widely accepted in healthcare [6], watermarking [7], image classification [8], indexing [9], and many more.

The various challenges faced by the scene retrieval automated systems are.

1.The variety of textual information gathered in the wild can be quite large. Textual information may be straightforward to understand for humans, but it can be very challenging for machines to differentiate between varying kinds of the same vehicle (for example, “A car is going straight” and “The car is going to head forward”). It appears that the problem in learned models is made worse by the limited amount of training data.

2.Second, the availability of higher training data is severely constrained. Text-to-image vehicle retrieval is still in its infancy, so there are not as many samples available for feature training through manual annotations as in the case of standard datasets like ImageNet. Models are expected to be more successful when they extensively use pre-trained parameters and only require a small number of labels for fine-tuning.

Thus, this paper proposes a novel vehicle retrieval approach through traffic video queries. To achieve the goal, this paper utilizes YOLOv7 and DeepSort algorithms for object detection, while the Kalman filter is used for tracking. BERT, a transfer learning model, is used to generate text embeddings for the queries. Also, the siamese neural network is used for similarity checking between the text and videos to retrieve the desired object from the videos.

The paper is further organized into related work and its analysis in section II. Section III describes the detailed proposed methodology. Section IV discusses the simulation results and their analysis. Finally, the conclusion and future work is presented in section V.

2. Related Work

Various research has been conducted and focuses on traffic scene retrieval and vehicle retrieval by combining computer vision and natural language processing (NLP). As the dataset for this task is the AI city challenge, all the researchers used this data and proposed novel solutions by achieving the ranks in the challenge.

Zhang [10] proposed a multi-granularity retrieval system comprised of three parts. First is the module for analysing textual descriptions of vehicles to extract useful information (such as colour, type, and speed, among other things). The second module is the language-augmented multi-query vehicle track retrieval module, which is used as a starting point for combining information from various imperfect queries. Third, the module for improving target vehicle attributes explicitly combines static and dynamic target vehicle characteristics to produce a final retrieval result. The proposed system won the sixth annual AI City Challenge by achieving the Mean Reciprocal Rank (MRR) of 66.06% on the private test set.

To combat the domain bias issue brought on by hypothetical situations and multiple camera angles, Le [2] creates a reliable natural language-based vehicle retrieval system. To facilitate descriptive representation learning, authors implement CLIP [11] to extract visual and textual representations efficiently. Additionally, a cutting-edge Domain Adaptive Training method creates pseudo labels from the annotated data to transfer the knowledge to the unanticipated data, which can then be applied to new scenarios in the test set. With this straightforward approach, the gap between the domains of the training set and the test set is closed with minimal computational expense and data while still outperforming state-of-the-art methods. To resolve the problem of uncertainty and get rid of the wrongly retrieved vehicle track, the proposed model employed a context-aware post-processing technique. As evidence of its efficacy, the suggested approach attained third place at the AI City Challenge 2022, achieving an MRR accuracy of 47.73% on the private test set.

Moreover, Li [12] proposed a system that uses grayscale video to perform multi-view temporal action localization to achieve action recognition while driving. For this purpose, SwinTransformer, a feature extractor, is used as a unified framework to identify both boundaries and classes. Also, the multiple loss functions for embedded features with unambiguous limitations are enhanced. The proposed system achieves an F1-score of 0.3154 on the A2 dataset.

Like the above approach, Nguyen-Ho [13] proposed a multi-modular architecture that provides reliable outcomes while while also being explainable and scalable. Also, a rule-based representation of events considers the entities adjacent to the the mentioned object and thus represents events through their interdependencies. The authors incorporated post-processing processing methods from HCMUS's strategy for the AI City Challenge 2021 with their modified model of Alibaba's solution solution to increase the interpretability of the suggested retrieval method. Due to the vehicle-centric nature of traffic data, two language and image modules are employed to conduct an in-depth analysis of the input and derive both the context's global properties and the vehicle's internal attributes. To achieve the best possible match with the query's core features, a novel dual-training method is introduced that treats each representation vector individually. With an MRR of 36.11 %, it ranked in the top four for fifty % of the test and all training data.

Similarly, Xu [14] proposed a solution for cross-modal vehicle retrieval that takes advantage of the interaction between visual and natural language representations. The source video must first be pre-processed to generate local and global motion semantics. There are two approaches to pre-processing pertinent description sentences: Textual Local Instance Semantics Extraction and Textual Local Motional Semantics Extraction. In order to extract the features from the images and embeddings from the text, a two-stream architecture model was used, which included four encoders for both text and image processing. When it came time to perform retrieval, a fusion of visual features and text embeddings was created by first arranging the importance of each input along the feature channel. In the end, the suggested method attained an MRR of 33.20%, putting the research in seventh place in the AI City Challenge 2022.

Alike, Du [15] proposed a novel framework, OMG, for the vehicle retrieval task based on natural language processing, which detects multiple levels of granularity in visual representation, textual representation, and optimization algorithm. Furthermore, the target's features and the camera's and the environment's motion are all encoded independently in the visual representation. Semantic features are represented at varying levels of granularity through one global embedding, three local embeddings, and a colour-type prompt embedding derived from the textual representation. Furthermore, the overall structure is optimized by utilizing a cross-modal, multi-granularity contrastive loss function. The suggested approach ranks ninth in the most recent AI City Challenge with an MRR of 30.1%.

In contrast to the above vehicle retrieval or traffic scene retrieval based on NLP, various researchers have proposed object retrievals other than vehicles. Rohrbach [16] suggested a strategy for teaching grounding that involves reconstructing a text using an attention mechanism that can be dormant or optimized effectively. The proposed method trains on a recurrent network language model to encode the text, and then it learns to pay attention to the proper region of the image in order to reproduce the input phrase. The efficacy of the proposed method with varying degrees of supervision, from unsupervised to partially supervised to fully supervised, on the Flickr 30k Entities and ReferItGame datasets is presented. The supervised variant outperforms both datasets by a wide margin.

Similarly, Karpathy [17] addressed the problem of retrieving images and sentences in both directions. The proposed method implemented the neural network model that learns a multi-modal embedding space to understand the hidden inter-modal alignment between image and text fragments. By breaking down sentences and images into smaller parts for analysis, the authors developed the Fragment Alignment Objective as an additional term to the more common Global Ranking Objective. Moreover, the proposed model demonstrated a notable improvement in retrieval rate on image sentence retrieval tasks compared to prior work. Additionally, the proposed model generates predictions that are open to interpretation.

Recent works have tried to visually represent data in images and encode text queries using a language feature extractor. An encoder is typically used for an image retrieval framework to learn global and local contexts, combining an attention mechanism and convolution techniques. Primarily, the research has used the AI challenge dataset for NLP-based vehicle or traffic scene retrieval. The highest MRR achieved is 66.06%. Even the accuracy achieved by the proposed approaches is not promising. Hence, there is still enormous scope for improvement in the result using the surveyed methods like encoders, transformers, CNN, RNN, and many more.

3. METHODOLOGY

This paper aims to detect objects, i.e., vehicles, in traffic videos using the queries. This paper is about the hybridization of computer vision and the NLP field. Furthermore, a novel approach for retrieving the traffic scenes from the videos is proposed. The block diagram representing the proposed methodology is given in Fig. 1.

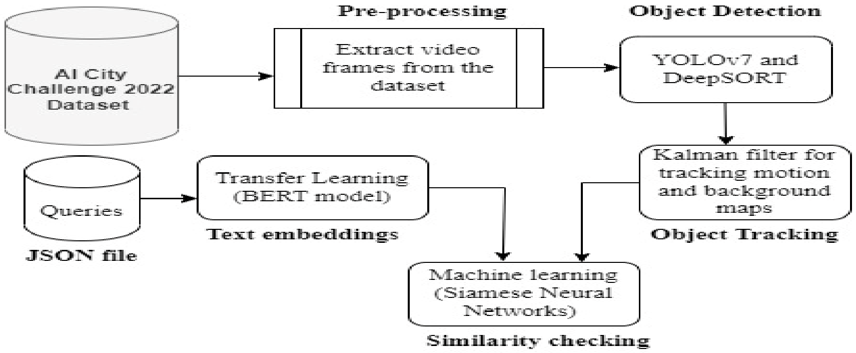
Pre-processing
Object Detection
AI City Challenge 2022 Dataset
Extract video frames from the dataset
YOLOv7 and DeepSORT
Kalman filter for tracking motion and background maps
Object Tracking
Queries
JSON file
Transfer Learning (BERT model)
Text embeddings
Machine learning (Siamese Neural Networks)
Similarity checking

Fig. 1: Block diagram of the proposed methodology

The various modules in the block diagram are given below.

3.1. Pre-processing

To process the videos for retrieving the scenes through the NLP queries, extracting the frames from the videos and providing it to the next module, i.e., Object detection for detecting the vehicles in the videos.

3.1.1. Object Detection

Once the frames are retrieved from the videos, they are provided as input to the object detection module. Two state-of-the-art algorithms, YOLOv7 and DeepSORT, are utilized for detecting vehicles in the videos.

In YOLOv7 [19], the transition layer's architecture remains unchanged despite the extensive renovations made by E-ELAN to the computational block. It employs expand, shuffle, and merge techniques, improving the network's learning capacity while preserving its initial gradient path. Group convolution increases the channel size and the number of individual computational blocks within a given layer by multiplying each block by the same value for the group parameter. After that, the feature maps generated by the individual computational blocks are rearranged and then appended to one another. Thus, the original architecture's number of channels will be reflected in the number of channels in each set of feature maps. At last, combine these feature map sets into one big pile. E-ELAN also improved its ability to learn new and different kinds of features.

Model re-parameterization and dynamic label assignment are two strategies for network optimization that are new to the literature. For the first case, the author thinks that providing access to the sequence of ResNet [20] or DenseNet [21] will give more gradient variability for slightly different graphs, which will break the network structure because RepConv's [22] is connected to its own identity. As a result, the author severed the identity connection in RepConv and created the desired reparameterized convolution, enabling him to effectively combine re-parameterized convolution with various networks. The latter is accomplished through the application of Deep supervision [23], with the addition of an auxiliary head structure in the network's intermediate layer serving as an auxiliary loss to direct the external network's weight. For this arrangement, the authors develop a brand-new system for assigning labels.

The DeepSORT [24] algorithm employs a straightforward application of the Kalman filter to manage the correlation of frame-by-frame data, and it employs the Hungarian algorithm to quantify the correlation. The performance of this algorithm is satisfactory even at a very high frame rate. Since SORT does not care about how the detected target looks, it is only applicable when the uncertainty in estimating the target's state is small. In addition, if a target is not matched in each frame, SORT deletes it. This helps to increase the efficiency with which tracking is performed. Nevertheless, this results in the issue of an ID switch, which indicates that the ID allotted to the target is susceptible to easy and frequent changes. Hence, DeepSORT uses appearance information and the ReID framework to extract appearance features, reducing ID switches by 45%. It also converts the IoU cost matrix-based matching mechanism of SORT into a combined Matching Cascade and IoU matching mechanism. Regarding long-term occluded targets, Matching Cascade's central idea is to prioritize track matching for the targets that emerge more frequently. Using this technique, one can successfully match previously occluded targets. In the final phase of matching, DeepSORT uses IoU matching to reconcile discrepancies between unmatched tracks and detection targets, which can mitigate significant shifts due to apparent mutations or partial occlusion. Furthermore, it takes inspiration from the ReID model in that it relies on an object detection network's output feature embedding to determine similarity.

Since the combination of YOLO and DeepSORT performs as admirably as many researchers have claimed [25], the recent version of the YOLO series, i.e., YOLOv7 and DeepSORT are ensembled to detect objects, such as vehicles, in videos.

3.1.2. Object Tracking

Once the objects have been identified, monitoring their motion and background maps is necessary. As a result, the proposed method uses the Kalman filter to predict and update the position of vehicle given in a video scene and detection in each of the video frames.

The Kalman filter, also known as linear quadratic estimation (LQE), remains one of the most important and widely used algorithms for fusing data from multiple sensors. The low computational cost, well-designed recursive properties, representation of the optimal estimator for one-dimensional linear systems under the assumption of Gaussian error statistics, and amenability to real-time implementation all contribute to the Kalman filter's efficacy [26]. The Kalman filter estimates model parameters that can be considered an expansion of Gauss's original least squares method. Until the middle of the twentieth century, most systems were unchanging, and measurements were not based on how much time had passed. However, the system may continuously evolve with measurements from mobile platforms in autonomous navigation and other applications. As a result, the Kalman filter is the optimal algorithm for this estimation method because it estimates system parameters across time epochs and links measurements.

The characteristics of the Kalman filter are that they are discrete, using measurements taken at regular intervals. Kalman filters are recursive, meaning the current state is used to inform future predictions (position, velocity, acceleration, etc.). It also makes an educated guess about external factors influencing the situation. Kalman filters make predictions. They do this by taking measurements, such as with sensors, and then using both predictions and measurements to make an updated estimate of the state [26].

3.2. Text embeddings

For converting the queries stored in the json file, the text is converted into word embeddings using the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model.

Google's 2018 Model for NLP Tasks, BERT [27], is a pre-training model that shows promise in several NLP-related tasks, such as text classification, question and answer, text summarization, and many more. BERT's functional architecture is split into two stages. In the first pre-training stage, unsupervised learning is used to acquire knowledge from data that has not been labelled, in this case, linguistic data. In supervised learning, a model that has been learned and pre-trained can be fine-tuned and adapted to the new challenge by employing transfer learning. Even with a limited amount of data, it delivers improved accuracy. Pre-trained model, BERT for NLP, was inspired by other transfer learning models available for computer vision tasks, such as VGG16, Inception v1, etc. Applying learned models to unrelated data is known as transfer learning. Each of the 12 blocks of transformers in the BERT-base architecture is equipped with 12 self-attention heads. There are a total of 768 pieces hidden. The model represents the sequence by using 512-word embeddings as input. Segmentation is accomplished with the help of the special token [SEP]. The [CLS] embedding is used initially when classifying texts. Softmax classifier is used for text classification by receiving embeddings of [CLS] tokens from the last layer.

3.3. Similarity Checking

After the word embeddings have been generated, the results of the tracked objects are fed to the similarity-checking module that also uses the word embeddings as input to compare the queries against the tracked objects and return the requested traffic scene. The siamese neural network is utilized for this purpose.

Siamese neural networks (SNNs) are a type of neural network architecture that includes subnetworks with two or more identical components. Here, "identical" means that their configurations share the same values for all parameters and weights. The two separate networks share the same updated settings. The networks are used in various contexts because of their ability to discover similarities between inputs through feature vector comparison.

A neural network is typically trained to make predictions across multiple classes. This causes issues whenever new data classes need to be added or deleted. It is necessary to retrain the neural network with the most recent data to fix this. In addition, a large amount of data must be available for training deep neural networks. Nevertheless, SNNs learn a similarity function to use in their predictions. As a result, it can be learned to determine whether or not two images are equivalent. By doing so, the network's classification skills can be applied to previously unseen data types without retraining them.

3.4. Performance metrics

Recall and mean recall rate (MRR) are the performance metrics used when evaluating the performance of the suggested methodology.

Recall

This metric indicates the proportion of the query's actual relevant results displayed. Mathematically, it is given as given in eq. 1.

$$Recall = \frac{True\ Positive+False\ Negative}{True\ Positive} \quad (1)$$

Mean Reciprocal Rank (MRR)

MRR is useful when the most relevant item returned by the system is to be displayed higher. Mathematically MRR is given in eq.2.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \quad (2)$$

Where $|Q|$ = number of queries; $rank_i$ =rank of first i relevant results.

4. Simulation Results and Analysis

The experiments are performed using python. The experiments are performed on Amazon Web Server (AWS). The graphical processing units, i.e., A100 GPU for training, are utilized, and it took about 6.5 hours and required 50 epochs for convergence.

4.1. Dataset description

The dataset used is the AI city Challenge 2022 [18] for the experimentations. The dataset, CityFlowV2, consists of the footage gathered from 46 cameras at 16 pedestrian crossings throughout a mid-sized American city, totalling 3.58 hours (215.03 minutes) of video. At 4 kilometres, the maximum separation between the two most distant cameras operating in tandem is impressive. Intersections, roadway segments, and highways are just some of the many different locations represented in the dataset. The data set is broken up into six different possible outcomes. Three are used for training, two for validation, and one for testing. The dataset has 313931 bounding boxes, representing 880 unique vehicle identities. Only vehicles captured by at least two cameras have their data annotated. Video quality is consistently high across the board, with a median frame rate of 10 fps and a minimum resolution of 960p. Every possible scenario also provides access to the video's offset from the beginning, which can be used to synchronize the videos.

This dataset was created using the CityFlow Benchmark and includes free-text annotations about vehicles. There are 2498 vehicle tracks in this data set, each of which is described in three different ways using natural language. This task consists of 530 different vehicle tracks and 530 different query sets, each of which has three different descriptions. This data set is meant to retrieve and rank vehicle paths for every query. The dataset consists of three files: train-tracks.json, test-tracks.json, and test-queries.json.

4.2. Quantitative result analysis

The various performance measures used for evaluating the performance of the proposed model are given below.

- Recall@5, i.e., recall for top 5 results, means taking the sum of the samples in the top five and dividing it by the total number of samples.
- Recall@10, i.e., recall for top 10 results, means taking the sum of the samples in the top ten and dividing it by the total number of samples.
- MRR

The results are evaluated by utilizing a dictionary stored in the files; the key contains the user's query to the model, and the value stores the model's ranking of the vehicles in response to that query (key). The proposed model iterates through the queries one at a time in the notebook, picking the best car based on its rank (the first one) and locating the vehicle in question. Also, the proposed model first checks to see if the car's rank is below 5, and if it is, the recall@5 is boosted, while if the rank is below 10, we boost the recall@10; and finally, we calculate the MRR is calculated based on the results acquired. The predicted model results contain a dictionary, the key contains the entered query to the model and value of the dictionary contains the ranks of the retrieved cars based on query description. Here we are sliding the queries one by one, chooses best matched vehicle tracks. If the rank of the tracked vehicle is below 5 then we increase the recall. The results of the proposed model are given in table 1.

Table 1: Results of the proposed model

Recall@5	**Recall@10**	**MRR**
0.2849	0.4208	0.2073

The results for the various defined queries are given in fig. 2, fig.3 and fig.4.

The defined queries for testing the proposed model are

queries = {"q1": ["A Blue SUV. A white SUV drives towards the intersection.", "A Blue SUV. A white SUV runs down the street.", "A Blue SUV. A white SUV runs down the street.", "A blue SUV. A white SUV. A white SUV."],

"q2": ["A vehicle runs down the street.", "A vehicle takes a U-turn", "A vehicle runs up the street."],

"q3": ["A Blue SUV, runs down the street", "A Blue SUV, crossing the intersection", "A Blue SUV drives towards the intersection.", "A Blue SUV. A blue SUV"]}

Fig. 2. depicts the red car running down the street is detected according to the query "A vehicle runs down the street". The red bounding box is the prediction, while the blue bounding box is the ground truth label for the frame. The difference between the predicted and the ground truth bounding box is performed using cosine similarity metrics, indicating that the difference is very low, as can be visible from the figs. 3,4, and 5.

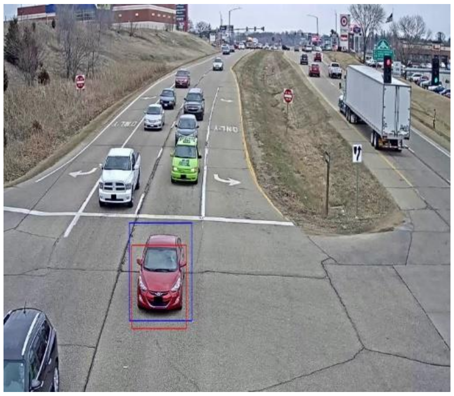

Fig.2: Result 1 for the queries described

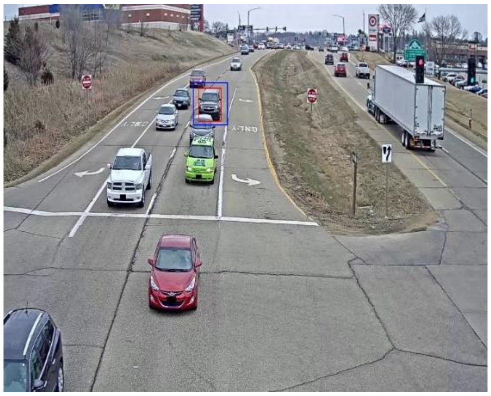

Fig.3: Result 2 for the queries described

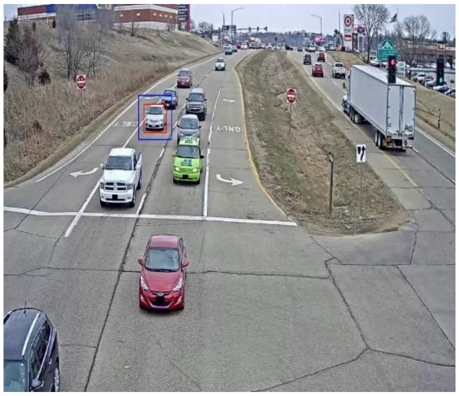

Fig.4: Result 3 for the queries described

Fig 3 depicts the blue SUV running down the street according to the q3: "A Blue SUV runs down the street." The prediction and the ground truth bounding boxes are nearly equal, depicting the efficiency of the proposed algorithm. Similarly, fig. 4 depicts the white car running down the street detected based on the query q2: "A vehicle runs down the street". Also, the predicted and the ground truth bounding boxes are nearly the same, indicating the effectiveness of the proposed algorithm.

5. Conclusion and Future Work

This paper proposes an approach that retrieves the results based on queries about objects and their spatial relationships within video frames. Visual information is retrieved by locating specific features of the image and describing them in detail. Even though the proposed approach was trained on data with a global view, the proposed approach's strength was shown by how often it gave correct visual traffic instances to the queries about local spatial relationships. The proposed approach uses YOLOv7 and DeepSort for object detection and transfers learning model BERT for generating the text embeddings. Also, the tracking of the vehicles in the video frames is performed by the Kalman filter. Finally, the Siamese neural network is used for similarity checking. From the experimental results, it can be concluded that the proposed approach provides effective results by achieving promising results on the benchmark dataset. Even though the results obtained by the proposed approach are promising, there is still a need for improvement. Hence, this research can be extended further to track the vehicles and classify them based on natural language description by exploring certain algorithms such as attention-based models for vision language alignment in common representation space or learning direct mapping between visual and language embeddings which may sidestep some of the challenges of feature space alignment.

References

[1] T. Huang, "Automatic symbolic traffic scene analysis using belief networks," in AAAI, 1994, vol. 94, pp. 966–972.

[2] H. D.-A. Le, "Tracked-vehicle retrieval by natural language descriptions with domain adaptive knowledge," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3300–3309.

[3] J. Wan, "Deep learning for content-based image retrieval: A comprehensive study," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 157–166.

[4] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," Comput Intell Neurosci, vol. 2018, 2018.

[5] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos, "Scene classification with semantic fisher vectors," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2974–2983.

[6] P. Israni, "Breast cancer diagnosis (BCD) model using machine learning," Int. J. Innov. Technol. Exploring Eng., vol. 8, no. 10, pp. 4456–4463, 2019.

[7] D. Israni and M. Bhatt, "Embedding Color Video Watermark in Image using Orthogonal and Bi-orthogonal Wavelet Transform," in Proc. of International Conference on Advances in Computer Science and Application, 2013.

[8] M. Dave, A. Ganatra, and D. Israni, "Evaluating classifiers and feature detectors for image classification bovw model: a survey," International Journal of Computer Engineering & Applications, vol. 12, pp. 1–7, 2017.

[9] P. Israni and D. Israni, "An indexing technique for fuzzy object oriented database using R tree index," in 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp), 2017, pp. 1–5.

[10] J. Zhang, "A Multi-Granularity Retrieval System for Natural Language-Based Vehicle Retrieval," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3216–3225.

[11] A. Radford, "Learning transferable visual models from natural language supervision," in International Conference on Machine Learning, 2021, pp. 8748–8763.

[12] W. Li, S. Chen, J. Gu, N. Wang, C. Chen, and Y. Guo, "MV-TAL: Mulit-view temporal action localization in naturalistic driving," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3242–3248.

[13] T.-L. Nguyen-Ho, "Text query based traffic video event retrieval with global-local fusion embedding," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3134–3141.

[14] B. Xu, Y. Xiong, R. Zhang, Y. Feng, and H. Wu, "Natural Language-Based Vehicle Retrieval With Explicit Cross-Modal Representation Learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3142–3149.

[15] Y. Du, B. Zhang, X. Ruan, F. Su, Z. Zhao, and H. Chen, "OMG: Observe multiple granularities for natural language-based vehicle retrieval," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3124–3133.

[16] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele, "Grounding of textual phrases in images by reconstruction," in European Conference on Computer Vision, 2016, pp. 817–834.

[17] A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," Adv Neural Inf Process Syst, vol. 27, 2014.

[18] "AI City Challenge ." https://www.aicitychallenge.org/2022-data-and-evaluation/#:~:text=This%20dataset%20contains%202498%20tracks,json%2C%20test%2Dtracks. (accessed Nov. 27, 2022).

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv:2207.02696, 2022.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[22] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.

[23] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in Artificial intelligence and statistics, 2015, pp. 562–570.

[24] B. Veeramani, J. W. Raymond, and P. Chanda, "DeepSort: deep convolutional networks for sorting haploid maize seeds," BMC Bioinformatics, vol. 19, no. 9, pp. 1–9, 2018.

[25] F. Yang, X. Zhang, and B. Liu, "Video object tracking based on YOLOv7 and DeepSORT," arXiv preprint arXiv:2207.12202, 2022.

[26] B. Alsadik, Adjustment models in 3D geomatics and computational geophysics: with MATLAB examples. Elsevier, 2019.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.