2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

TrafficLens: Multi-Camera Traffic Video Analysis Using LLMs

Md Adnan Arefeen[1], Biplob Debnath[2], and Srimat Chakradhar[2]

***Abstract*— Traffic cameras are essential in urban areas, playing a crucial role in intelligent transportation systems. Multiple cameras at intersections enhance law enforcement capabilities, traffic management, and pedestrian safety. However, efficiently managing and analyzing multi-camera feeds poses challenges due to the vast amount of data. Analyzing such huge video data requires advanced analytical tools. While Large Language Models (LLMs) like ChatGPT, equipped with retrieval-augmented generation (RAG) systems, excel in text-based tasks, integrating them into traffic video analysis demands converting video data into text using a Vision-Language Model (VLM), which is time-consuming and delays the timely utilization of traffic videos for generating insights and investigating incidents. To address these challenges, we propose TrafficLens , a tailored algorithm for multi-camera traffic intersections. TrafficLens employs a sequential approach, utilizing overlapping coverage areas of cameras. It iteratively applies VLMs with varying token limits, using previous outputs as prompts for subsequent cameras, enabling rapid generation of detailed textual descriptions while reducing processing time. Additionally, TrafficLens intelligently bypasses redundant VLM invocations through an object-level similarity detector. Experimental results with real-world datasets demonstrate that TrafficLens reduces video-to-text conversion time by up to $4\times$ while maintaining information accuracy.**

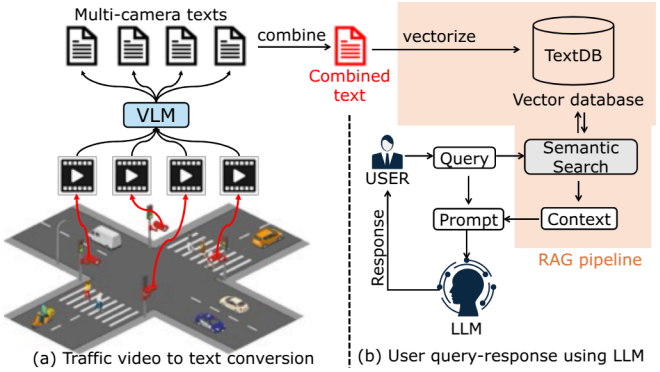


Fig. 1: An overview of the RAG-based traffic video analysis system. It operates in two phases: a) the multi-camera videos is initially converted into a text document, the combined texts from all documents are then chunked into smaller parts and stored in a vector database; b) queries are answered using a Large Language Model (LLM), leveraging query and contextual information retrieved from the vector database through semantic search.

I. Introduction

Traffic cameras have become ubiquitous in urban environments, with many cities installing hundreds to thousands of them. These cameras serve the purpose of continuously capturing video footage of traffic scenarios. The collected videos are then systematically stored for post-analysis. This extensive archive of video data offers city planners and transportation authorities a valuable resource for extracting insights, conducting investigations, preventing potential disasters, and addressing various inquiries related to traffic management. The sheer volume of archived traffic data is immense. For instance, a city with approximately one thousand of these cameras may accumulate as much as 230 terabytes of video data each month [1]. In a multi-camera setup that captures the same scene from different angles, the volume of traffic data can double or even quadruple than single camera video feeds. Analyzing such a vast amount of video feeds from traffic cameras is essential for tasks such as traffic monitoring, congestion management, and incident detection. However, this process demands a comprehensive understanding of the information embedded within the videos, highlighting the necessity for advanced analytical tools and methodologies [2]–[4].

In the domain of traffic video analysis, processing user queries through natural language processing enables direct interaction with video content. Large Language Models (LLMs), such as ChatGPT [5], have excelled in text-based interactions but face limitations when addressing data not encountered during training. The Retrieval-Augmented Generation (RAG) [6] approach has been widely adopted to augment LLMs with the ability to integrate unseen data. However, traditional RAG systems are designed to handle textual data, posing challenges when dealing with non-textual formats like videos or images common in traffic monitoring. Addressing this gap requires enhancing RAG systems with capabilities to convert these media into text-compatible formats.

Typically, Vision-Language Models (VLMs) [7]–[9] are employed to transcribe traffic videos into text, which is then processed through a RAG-based system using Large Language Models. While crucial, this conversion process often becomes a bottleneck, particularly when dealing with a large corpus of videos. It leads to delays in utilizing traffic videos promptly for generating actionable responses. This challenge is further compounded by the deployment of multiple cameras at strategic points within a traffic intersection, as depicted in Figure 1.

The multi-camera setup at a traffic intersection is designed

[1]Md Adnan Arefeen is with NEC Laboratories America, Princeton, NJ, and University of Missouri-Kansas City, MO, USA. aarefeen@nec-labs.com

[2]Biplob Debnath and Srimat Chakradhar are with the NEC Laboratories America, Princeton, NJ, USA {biplob,chak}@nec-labs.com

with redundancy in mind. Each camera complements the others by capturing events from different angles and perspectives. In situations where one camera may fail to capture a particular event due to obstructions or limitations in its field of view, neighboring cameras are strategically positioned to fill in these gaps. This coordinated arrangement minimizes the risk of incidents going unnoticed, as there is a high probability that any event overlooked by one camera will be captured by another.

Figure 1 illustrates a RAG-based system designed to address questions related to multi-camera traffic videos. To initiate the process, the video RAG system converts the videos into text, dividing it into non-overlapping clips. Each clip undergoes analysis by a Vision-Language Model (VLM) [10]–[13], recording the output in text format. The video feed from each camera is processed by a VLM. The outputs from all VLM cameras are combined to generate the final textual description of events captured at the traffic intersection clip by clip basis. Collating text information from all clips generates a long document for the multi-camera videos, which is then segmented into chunks. Each chunk is embedded into a vector by an embedding model and subsequently stored in a vector database. Once the video-to-text conversion is complete, upon receiving a query, the video RAG system embeds the query, and conducts a semantic search using the embedding vectors of chunks to retrieve relevant chunks from the vector database. These retrieved chunks form the context, which is combined with the query to generate a prompt. Finally, the prompt is fed into a Large Language Model (LLM) to generate a response for the query.

While a RAG-based system for traffic videos can address a wide range of queries, a major challenge lies in the time required for a VLM model to generate textual descriptions from video clips. For instance, processing a frame using the InternLM-XComposer2 [9], with a maximum token limit of 64, takes an average of 3.8 seconds on a server equipped with an NVIDIA GeForce RTX 3090 GPU. Thus, it would take more than a day for this VLM model to analyze a 24-hour traffic video from one camera and generate text descriptions, assuming it processes one frame every three seconds. Now, if converting 24-hour traffic videos from one camera takes one day, and an intersection is monitored using four cameras, then it would take more than 4 days just to extract text from the video footages before starting analysis through LLMs. This poses a significant challenge for various applications, such as preventing law enforcement agencies from conducting timely analyses of criminal incidents captured in the traffic videos.

Vision-Language Models (VLMs) essentially function as predictive models for the next token in a sequence. They take input in the form of images or video clips, along with a text prompt, and produce text as output. Within the model, inputs are tokenized initially, and these tokens can be processed in parallel, meaning that the length of the input tokens does not significantly affect latency. However, the length of the generated output significantly affects latency due to sequential token generation. The size of the generated output is influenced by the prompt. Moreover, VLMs offer a maximum output token limit parameter to regulate the amount of generated information, thereby controlling inference speed. In this paper, our aim is to accelerate the video-to-text conversion process by refining the text prompt and adjusting the maximum token limit parameter of VLMs, leveraging the unique attributes of multi-camera setups deployed at traffic intersections.

To accomplish this goal, we propose TrafficLens , an innovative algorithm designed to quickly produce textual descriptions of video clips using VLM models for monitoring traffic intersections equipped with multiple cameras. It employs a sequential approach, capitalizing on the overlapping coverage areas of the cameras at these intersections. Initially, TrafficLens applies VLM to the video clip from one camera, prompting it to generate detailed descriptions while employing a higher token limit. Subsequently, TrafficLens utilizes the resulting output as a prompt for the next camera, instructing VLM to include additional details not initially covered, while enforcing a lower token limit. This iterative process continues for subsequent cameras, with each iteration incorporating further details from previous cameras and reducing the token limit. Furthermore, it can bypass subsequent VLM calls when it detects a high degree of similarity among the video feeds from different cameras.

In summary, we make the following contributions:

- We present TrafficLens , a novel algorithm for accelerating video-to-text conversion using VLMs for traffic intersections equipped with multiple cameras. It employs a higher token limit for the first camera to extract comprehensive text, while applying a lower token limit for subsequent cameras to capture objects undetected by the preceding cameras.
- TrafficLens utilizes intelligent prompt engineering to adjust the token limit in subsequent camera videos during the video-to-text conversion process. Additionally, it reduces conversion time by eliminating redundant clips from subsequent cameras through object-level similarity detection.
- Our experimental evaluation using the StreetAware dataset [14], which covers traffic intersections in New York City, demonstrates that TrafficLens can accelerate the video-to-text conversion time by up to 4× while maintaining information accuracy.

II. Related Work

Vision-Language Models (VLMs) represent a crucial advancement in the fields of Natural Language Processing (NLP) and Computer Vision (CV). By integrating textual and visual data, they enable a deeper understanding of multimodal content. Through VLMs, transportation systems are able to deeply understand real-world environments, thereby improving driving safety and efficiency. To build an interactive system for advanced traffic video understanding, along with VLMs, large language models are also necessary to query on textual description of traffic video data. Using retrieval augmented generation (RAG) [6], [12], [15],

video data can be incorporated as text to execute interactive query by the users. A survey is conducted by Zhou et al. [16] to explore the application of VLMs in intelligent transportation systems. VLMs concerning traffic data are primarily categorized into three types: Multimodal-to-Text (M2T) [17], Multimodal-to-Vision (M2V) [18], and Vision-to-Text (V2T) [7], [19]. While V2T models take images or videos as inputs and generate textual descriptions as outputs, the M2T models take inputs in the form of image-text or video-text pairs. They analyze both the visual and textual components and generate textual descriptions as output. For example, when provided with an image depicting traffic congestion alongside its corresponding textual description, an M2T model can produce a detailed textual narrative of the scene, encompassing elements such as traffic flow, weather conditions, and levels of road congestion.

In this paper, our focus is on designing a large-scale traffic video analysis system utilizing multimodal-to-Text (M2T) models and Large Language Models (LLMs). Current M2T models operate on image or video clips spanning several seconds. However, we are dealing with longer videos. Therefore, we use M2T models as foundational components to process longer videos by dividing them into smaller clips. Nonetheless, these models take longer to process images or video clips into textual form. Hence, we leverage the distinctive features of multi-camera setups deployed at traffic intersections to expedite the text conversion process. While some works exist related to the multi-camera setup [20], [21], they primarily focus on object detection and tracking. In contrast, our focus lies in generating a textual description by combining the information observed by individual cameras.

III. TrafficLens

A. Motivation

Multiple cameras provide broader coverage of the traffic intersection, reducing blind spots. If one camera fails, others can still capture necessary footage. Figure 2 shows views from two cameras at a traffic intersection [14], capturing simultaneous moments. This multi-camera setup reveals additional details, such as a person not visible from the right camera, while both cameras capture the white and black cars. To enhance efficiency in multi-camera traffic video analysis, merging overlapping information with unique details from each camera only once can significantly reduce the video-to-text conversion process.

We have identified three key strategies to accelerate video-to-text conversion in a multi-camera traffic video analysis system utilizing large language models:

1) **Efficient Use of VLMs:** Unlike traditional machine learning models with fixed processing times, the inference time of a Vision-Language Model (VLM) [7]–[9] varies depending on the complexity of the prompt and the size of the generated output. Adjusting the *maximum token limit* and refining prompts can help reduce processing times.
2) **Addition of Unique Information Across Cameras:** Using specific prompts such as "Describe the undetected objects only" instead of a general narrative such as "Compose a descriptive narrative" can prevent redundant processing across cameras, further reducing conversion time as VLM inference depends on the number of output tokens.
3) **Elimination of Overlapping Information:** A similarity detector can prevent reprocessing video footage from subsequent cameras that duplicate information from earlier feeds. Additionally, it can help reduce the hallucination problem in VLMs [22].

(a) Left camera (b) Right camera

Fig. 2: Left and right camera view in StreetAware [14] dataset. Unique information of a person wearing a plaid shirt is absent from right camera view.

To illustrate the effect of the *maximum token limit* parameter (which controls the amount of text generated by the VLM models) on inference time, let us examine the image depicted in Figure 3, which shows a street scene in New York City sourced from the StreetAware dataset [23].

Fig. 3: A sample image from the StreetAware [23] dataset.

We generate textual descriptions using the InternLM-Xcomposer2 VLM model (with 1.8 billion parameters) [9] with the prompt "Compose a descriptive narrative" by setting the maximum output token limit parameter to 128. It generates the following output:

InternLM-Xcomposer2-1.8b:
Output Tokens: 16, 32, 64, 128
The image captures a bustling city intersection, where a man in a blue shirt is crossing the street, a bicycle rider in the background, and a white SUV parked on the side. The intersection is marked by a crosswalk, and a bike lane is also visible. The backdrop of the scene is a mix of urban architecture, including buildings and trees, and a clear blue sky.

The total number of output tokens generated above is 77. Different colors indicate the additional details added

as the token limit increases. It is important to note that the maximum output token limit serves as guidance for the VLM to control the length of the generated text. However, depending on the input image and prompt, it may produce shorter outputs.

To demonstrate the relationship between inference latency and output token size in VLMs, we conducted experiments with various maximum output token limits: specifically, 16, 32, 64, 128, and 256. We tested these limits on VLM models including InternLM-Xcomposer2 (with 1.8 billion and 7 billion parameters [9], LLAVA-1.5-7b model [24], and MobileVLM with 1.7 billion parameters [25]. This range of limits enables us to observe how each model responds to different token limit parameters.

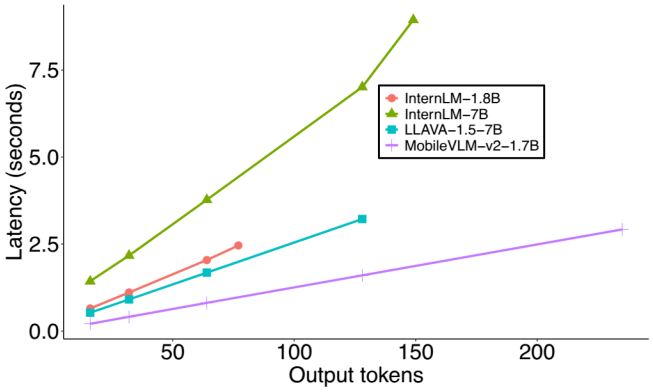


Fig. 4: Latency of VLM output generation increases with the increase in number of output tokens.

In Figure 4, we observe a clear relationship between the maximum token limit and the inference latency across various VLMs. As the maximum token limit increases, the time taken for inference also consistently increases across all VLMs. This trend indicates that as VLM models are tasked with generating longer textual descriptions, they require more time to complete the inference process. These observations motivate us to build an interactive system, TrafficLens, that can efficiently convert multi-camera videos to text for further analysis with large language models.

B. Proposed Method

Multiple cameras covering the same view from different positions provide unique perspectives. While there is some overlap in the scenes captured, each camera may also record objects that are only visible from its specific vantage point. Consequently, in a multi-camera setup at a traffic intersection, extracting information independently from each camera can lead to the accumulation of redundant data, which in turn impacts the efficiency of video ingestion. To eliminate redundancy from accumulating information from all cameras, TrafficLens implements an incremental approach. It first considers a camera as base and extracts the scene information.

1) Base-camera Ingestion: TrafficLens starts with selection of a base camera video ingestion. At first, TrafficLens runs a scene detector to identify the non-overlapping clips from the video corpus. With an open source vision language model (VLM), TrafficLens runs the following prompt to extract the details :

> **Prompt 1:**
> Compose a descriptive narrative.

2) Camera-n ingestion: After completing the ingestion from the base camera, TrafficLens obtains the base text for each clip from the base camera. Subsequently, TrafficLens employs Prompt 2 to delve deeper and extract additional details from the clips of the next camera with the same time frame as base-camera. In this way, unique information from all cameras will be covered. Since multiple cameras cover the same scene from various angles, this approach leverages the unique positioning of each camera to enrich the overall context and understanding of the scenario.

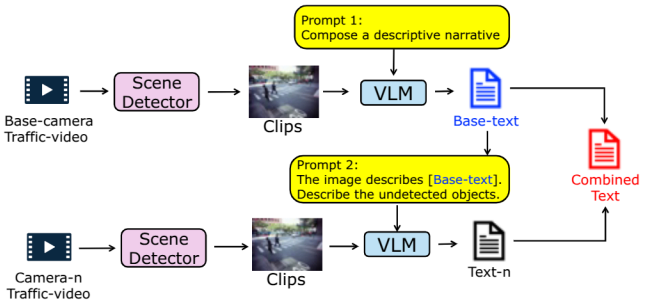
> **Prompt 2:**
> The image describes [Base-text]. Describe the undetected objects.

Figure 5a depicts the accelerated video-to-text conversion workflow through prompt refinement of TrafficLens. Varied prompts are employed to progressively capture textual information across cameras. Prompt 2 is specifically designed to compel the VLMs to extract information solely about objects that were not detected by previous camera feeds. This approach significantly reduces redundancy in subsequent camera feeds.

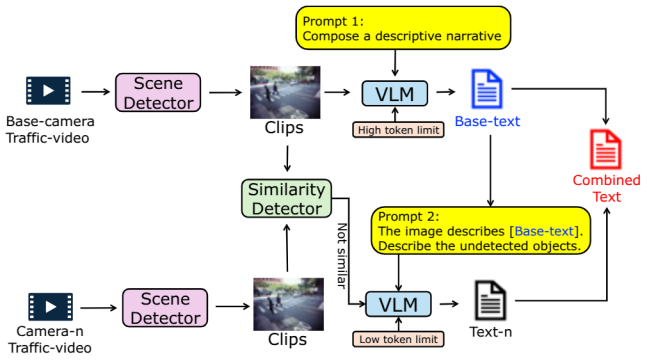
We have observed that when using Prompt 1, the base camera text generally captures most of the necessary information. Consequently, when Prompt 2 is applied to subsequent cameras, it often redundantly identifies objects that have already been detected. An example is shown below, where repeated information is indicated in red.

> **Base-text**: The image captures a moment on a city street, where a pedestrian is seen crossing the road. The pedestrian, dressed in a black shirt and shorts, carries a backpack, suggesting they might be a student or a commuter. The street is marked with a bike lane, indicating a focus on eco-friendly transportation. The background reveals a building with a large window, and a car parked nearby, adding to the urban setting. The perspective of the image is from the side of the road, providing a clear view of the pedestrian and the surrounding environment.
>
> **Camera-n text**: The undetected objects in the image include a building with a large window, a car parked nearby, and a pedestrian crossing the road.

Based on this observation, TrafficLens introduces two



(a) Different prompts are employed to incrementally capture textual information across cameras.



(b) Token limits are adjusted, and a clip similarity detector is utilized to further reduce the time required for textural information extraction.

Fig. 5: Accelerated video-to-text conversion workflow of TrafficLens.

additional techniques to enhance the ingestion capability of VLMs for the multi-camera setup at the traffic intersections: Clip Similarity Detector and Reduced Token Limit. Figure 5b illustrates the workflow for the enhanced video-to-text ingestion process of TrafficLens.

Clip Similarity Detector. In most multi-camera setups, there is an overlapping region where the same objects are captured by all cameras. Additionally, during periods of no traffic (i.e., when roads are empty), all cameras capture no objects, thus sharing similar information. Taking this into account, TrafficLens implements a similarity detector to identify similarities in multi-camera scenes. It uses object-level similarity as a metric for similarity detection, employing the Intersection over Union (IoU) score for quantification. When the IoU score exceeds a specified threshold, the VLM is not invoked for that clip to avoid generating redundant textual information already obtained from the base camera. This approach enables a more refined analysis of visual data across different camera feeds.

Eliminating similar clips also contributes to reducing the hallucination problem [22] in VLMs. Specifically, when Prompt 2 is invoked to generate additional descriptions about undetected objects, the VLM may hallucinate (i.e., produce fabricated information). By avoiding the use of the VLM with similar clips, TrafficLens not only reduces video-to-text conversion time but also mitigates the hallucination problem.

Reduced Token Limit. We discuss in Section III-A that the latency involved in converting video feeds to text is influenced by the number of output tokens produced by VLMs. To reduce ingestion time, we can limit the number of tokens generated during this conversion process. For a multi-camera setup in TrafficLens , when calling the VLM n times for n cameras, we first generate a detailed description of the base camera feed using Prompt 1 with a higher token limit. Subsequently, we use a tailored Prompt 2 to generate additional information for other camera feeds, specifically targeting details that may have been missed by the base feed. When invoking Prompt 2, TrafficLens reduces the token limit. This strategy significantly reduces the time required to process multi-camera video feeds at traffic intersections by cutting down on the number of generated tokens and eliminating redundant information.

IV. Evaluation

For evaluation of TrafficLens , we consider StreetAware [23] dataset that focuses on observing pedestrian movement in intersections using REIP sensors. The description of the dataset is shown in Table I. We consider videos of two cameras from the StreetAware dataset from two camera positions i.e. left, and right. Each video is 46 minutes 44 seconds long.

TABLE I: Description of datasets with ingestion time

Camera	Model	Video Duration (hh:mm:ss)	Ingestion time (hh:mm:ss)
Left	InternLM-1.8B	00:46:44	00:28:54
	LLAVA-7B	00:46:44	00:30:22
Right	InternLM-1.8B	00:46:44	00:27:19
	LLAVA-7B	00:46:44	00:30:57

TABLE II: Token generation statistics under maximum token limit

Camera	Model	Max. token limit	Max. output tokens	Min. output tokens	Avg. output tokens
Left	InternLM-1.8B	256	254	53	113.14
	LLAVA-7B	256	213	77	142.18
Right	InternLM-1.8B	256	231	53	105.65
	LLAVA-7B	256	214	77	144.82

We consider two Vision-Language Models (VLMs) for experimental analysis: InternVLM-1.8B [9] and LLAVA-7BV [24]. We run each model independently using the Prompt 1 on each of the video feeds and compute the total ingestion time. We refer this as the *baseline* method.

TABLE III: Comparison of ingestion time between TrafficLens and the baseline approach.

No of cameras	Total video duration	Model	Total ingestion time (hh:mm:ss)				
				TrafficLens			
			Baseline	$T_r = 80$, $T_\ell = 32$ w/o Similarity Detector	$T_r = 80$, $T_\ell = 32$ w Similarity Detector	$T_r = 50$, $T_\ell = 32$ w/o Similarity Detector	$T_r = 50$, $T_\ell = 32$ w Similarity Detector
2	01:33:28	InternLM-1.8B	00:56:13	00:28:32	00:25:16	00:21:09	00:18:07
		LLAVA-1.5v-7B	01:01:19	00:25:42	00:22:21	00:19:44	00:16:17

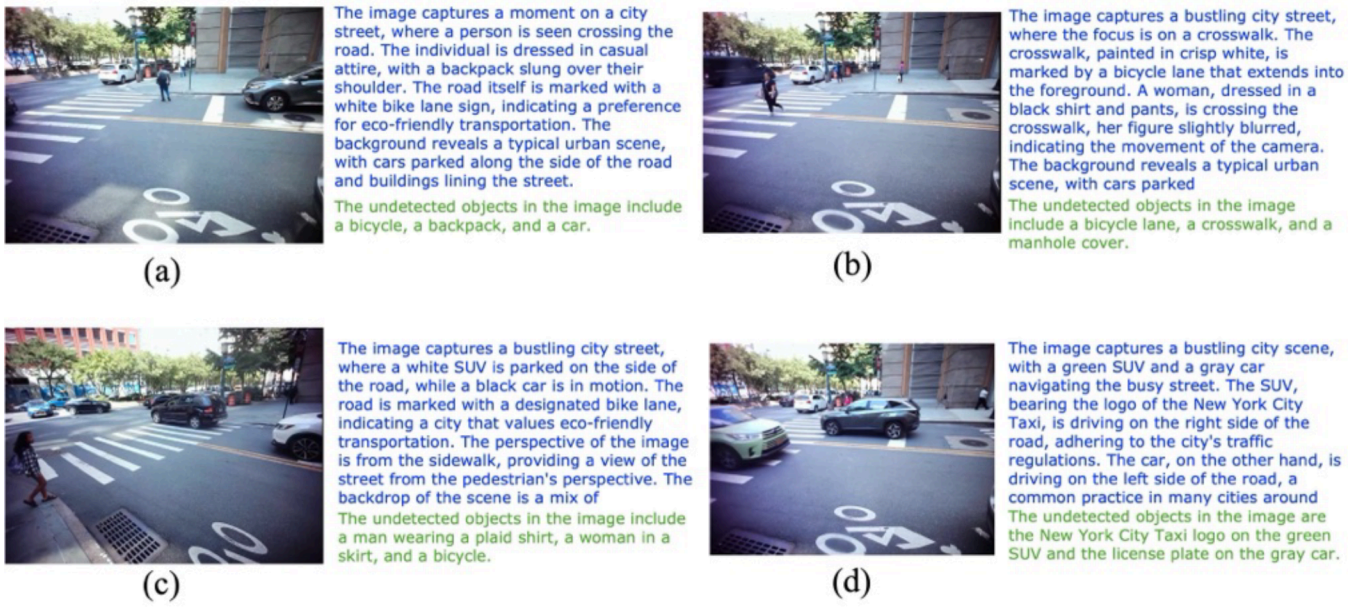


Fig. 6: Video clips with descriptions using two prompts with token limit 80 (right camera, referred as base, text shown in blue color) and 32 (left camera text shown in green color). The addition of new information is visible from the text descriptions: (a) "backpack" is absent in right camera output; (b) "manhole-cover" can not be detected in the right camera text; (c) "man wearing a plaid shirt" is not visible from the right camera; and (d) a license plate is detected in the left camera.

We observe that each model takes approximately 28 to 31 minutes to convert the 46-minute and 44-second long videos to text. Therefore, for a multi-camera setup using 2 cameras, the total ingestion time rises to between 56 minutes and 1 hour, which is time-consuming.

We also observe the output token statistics for each model in Table II. On average, the InternLM-1.8B model generates between 105 and 113 tokens, while the LLAVA-1.5v2-7B model produces between 142 and 144 tokens. These figures inform the token limits that should be set for text generated by the base camera and other cameras. Following this, in TrafficLens , we consider the right camera as base and convert the video to text using Prompt 1 with token limits (T_r =80, and 50 respectively). Subsequently, for the left camera, we utilize Prompt 2 to extract additional details with token limits $T_\ell = 32$. Figure 6 displays clips with text from the right camera, limited to 80 tokens (shown in blue), along with additional text from the left camera, limited to 32 tokens (shown in green). It is evident that utilizing textual information from both cameras enhances the accuracy of information compared to relying solely on a single camera, underscoring the value gained from a multi-camera setup.

Table III demonstrates the speed-up in video-to-text conversion time achieved by TrafficLens compared to the baseline. For InternLM-1.8B, TrafficLens achieves approximately $2\times$ to $3\times$ time reduction, and for LLAVA-7B, TrafficLens achieves approximately $3\times$ to $4\times$ time reduction compared with the baseline approach. For InternLM-1.8B (with $T_r = 80$ and $T_\ell = 32$), adjusting the token limit reduced the conversation time from approximately 56 minutes to 21 minutes. Additionally, the similarity detector further reduced it to 18 minutes. For LLAVA-7B under the same settings, adjusting the token limit reduced the conversation time from approximately 61 minutes to 29 minutes, with the similarity detector further reducing it to 16 minutes. In all these experiments, the similarity threshold has been set to 0.21, as determined by the ablation results described in section V.

The addition of new information in subsequent VLM calls as shown in Figure 6 suggests a dissimilarity between the left camera (i.e., camera-n) text and the right camera (i.e., base) text. A greater degree of text dissimilarity increases the likelihood of uncovering new details about the traffic intersection. To quantify this new information, we employ metrics such as the BERT score [26] and ROUGE score [27].

TABLE IV: Ingestion information variation among cameras

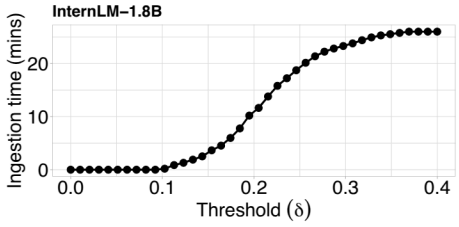
Model	Method	BERT score	ROUGE-L
LLAVA-7B	Baseline	0.96	0.43
	TrafficLens ($T_r = 80, T_\ell = 32$)	**0.85**	**0.26**
InternLM-1.8B	Baseline	0.95	0.42
	TrafficLens ($T_r = 80, T_\ell = 32$)	**0.85**	**0.32**

Lower BERT and ROUGE scores indicate a higher diversity of information. Our observations, as shown in Table IV, reveal that the scores decrease with the use of targeted prompts and reduced token limits, confirming that text dissimilarity corresponds to the addition of new contextual information.

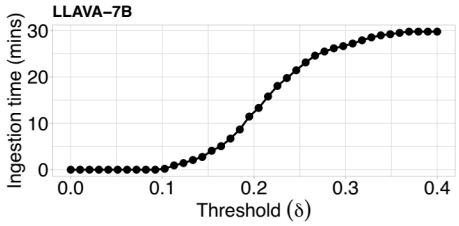
To evaluate the query response capabilities of the Large Language Model (LLM) through textual information generated by TrafficLens , we present three complex questions in Table V. These questions go beyond simple object identification, such as finding a "car", and include queries for specific details like "a white SUV", "person with backpack", and "woman with a black bag". These queries require the sophisticated capabilities of the VLM to accurately extract the necessary information. The context being retrieved by the RAG from the text produced by TrafficLens based on each query. For all these queries, TrafficLens despite reduced text information, successfully generated correct response.

V. Ablation Study

We conduct an in-depth analysis of the relationship between the similarity threshold, denoted as δ, and the ingestion time required by our system. The similarity threshold δ plays a crucial role in determining which video clips are processed by the Vision-Language Model (VLM). Specifically, δ sets a cutoff point, where only clips with a similarity score lower than δ are passed to the VLM for further processing.



(a) InternVLM-1.8B [9]



(b) LLAVA-7BV [24]

Fig. 7: The impact on ingestion time when skipping clips from subsequent cameras during processing with different VLMs using the similarity detector of TrafficLens.

TABLE V: Query-response from StreetAware [14] video dataset.

Q1	Is there any white SUV?
RAG Context	00:07:23 : The image captures a bustling city street, where a black SUV and a white SUV are parked on the side of the road. The black SUV is positioned on the left side of the image, while the white SUV is on the right. The street is lined with trees and buildings, adding to the urban atmosphere. A crosswalk is also visible in the foreground, indicating a pedestrian-friendly area. The undetected objects in the image are the black SUV and the white SUV.
Response	Yes, there is a white SUV. It is mentioned as being parked on the side of the road.
Q2	Is there any person with a backpack?
RAG Context	00:00:48 : The image captures a moment on a city street, where a person is seen crossing the road. The individual is dressed in casual attire, with a backpack slung over their shoulder... . 00:02:05 : The image captures a moment on a city street, where a pedestrian is seen crossing the road. The pedestrian, dressed in a black shirt and shorts, carries a backpack, suggesting they might be a student or a commuter... . 00:00:35 : The image captures a moment on a bustling city street. A man, clad in a white shirt and blue jeans, strides purposefully towards the camera. He carries a backpack, ... 00:05:27 : The image captures a moment on a city street, where a man is crossing the road. He is dressed in a blue shirt and black pants, and he carries a backpack ...
Response	Yes, there is a person with a backpack. Multiple instances in the provided context describe individuals carrying backpacks while crossing the street.
Q3	Is there any woman with a black bag?
RAG Context	00:00:16 : The image captures a moment on a bustling city street. A woman, dressed in a vibrant red shirt and blue jeans, is seen walking with a black bag. She is crossing the street, adhering to the pedestrian crossing markings. The street is lined with trees, providing a touch of nature amidst the urban setting. Cars are parked along the side of the road, hinting at the. The undetected objects in the image are the woman's black bag and the cars parked along the side of the road.
Response	Yes, there is a woman with a black bag. She is described in the context at 00:00:16:000 as wearing a vibrant red shirt and blue jeans, walking with a black bag while crossing the street.

As illustrated in Figure 7, our findings reveal a clear trend: as the value of δ increases, the ingestion time also increases. This is because a higher δ broadens the range of clips deemed dissimilar enough to require processing by the VLM. Consequently, more clips are subjected to detailed analysis, leading to longer overall ingestion times. On the other hand, when δ is set to a lower value, the threshold for processing is more stringent, and fewer clips fall below this threshold. As a result, the number of clips requiring VLM processing decreases, thereby reducing the ingestion time. This inverse relationship between δ and ingestion time is critical for optimizing system performance.

Through extensive analysis of average ingestion times across various threshold values, we have determined that a δ of 0.21 strikes a balance between processing thoroughness and efficiency. Therefore, this value was selected for all subsequent experiments described in Section IV, as it provides a practical trade-off, ensuring that the system effectively processes necessary clips without incurring unnecessary delays.

VI. Conclusion

In this paper, we introduce TrafficLens , a novel algorithm designed to enhance rapid video-to-text conversion through Vision-Language Models (VLMs) in multi-camera setups at traffic intersections. TrafficLens efficiently processes video information by first extracting detailed textual data using a VLM from a base camera with a high maximum token limit. It then applies a lower token limit to gather additional information from other cameras, optimizing the overall video-to-text conversion time. TrafficLens also skips text extraction from other cameras when there is a high degree of similarity between the video feeds. This approach ensures quicker data processing while eliminating redundant information extraction. Finally, texts from all cameras covering the same time frames are concatenated and used as a knowledge base within a Retrieval-Augmented Generation (RAG) framework for a Large Language Model (LLM) to generate responses based on user queries. Experimental results demonstrate that TrafficLens significantly accelerates video-to-text conversion time without sacrificing information accuracy.

References

[1] F. Kossmann, Z. Wu, E. Lai, N. Tatbul, L. Cao, T. Kraska, and S. Madden, "Extract-transform-load for video streams," *Proceedings of the VLDB Endowment*, vol. 16, no. 9, May 2023.

[2] U. Mittal, P. Chawla, and R. Tiwari, "Ensemblenet: A hybrid approach for vehicle detection and estimation of traffic density based on faster r-cnn and yolo models," *Neural Computing and Applications*, vol. 35, no. 6, 2023.

[3] Y. Tang, J. Bi, S. Xu, L. Song, S. Liang, T. Wang, D. Zhang, J. An, J. Lin, R. Zhu, *et al.*, "Video understanding with large language models: A survey," *arXiv preprint arXiv:2312.17432*, 2023.

[4] Y. Shibata, Y. Kawashima, M. Isogawa, G. Irie, A. Kimura, and Y. Aoki, "Listening human behavior: 3d human pose estimation with acoustic signals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 323–13 332.

[5] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[7] Q. M. Dinh, M. K. Ho, A. Q. Dang, and H. P. Tran, "Trafficvlm: A controllable visual language model for traffic video captioning," *arXiv preprint arXiv:2404.09275*, 2024.

[8] X. Tian, J. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," *arXiv preprint arXiv:2402.12289*, 2024.

[9] X. Dong, P. Zhang, Y. Zang, Y. Cao, B. Wang, L. Ouyang, X. Wei, S. Zhang, H. Duan, M. Cao, *et al.*, "Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model," *arXiv preprint arXiv:2401.16420*, 2024.

[10] Showlab, "VLog: Video As a Long Document," https://github.com/showlab/VLog, 2023, gitHub Repository.

[11] K. Lin, F. Ahmed, L. Li, C.-C. Lin, E. Azarnasab, Z. Yang, J. Wang, L. Liang, Z. Liu, Y. Lu, C. Liu, and L. Wang, "MM-VID: Advancing Video Understanding with GPT-4V(ision)," *arXiv preprint arXiv:2310.19773*, 2023.

[12] M. A. Arefeen, B. Debnath, M. Y. S. Uddin, and S. Chakradhar, "iRAG: An Incremental Retrieval Augmented Generation System for Videos," *arXiv preprint arXiv:2404.12309*, 2024.

[13] M. A. Arefeen, B. Debnath, M. Y. Uddin, and S. Chakradhar, "ViTA: An Efficient Video-to-Text Algorithm using VLM for RAG-based Video Analysis System," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2266–2274.

[14] Y. Piadyk, J. Rulff, E. Brewer, M. Hosseini, K. Ozbay, M. Sankaradas, S. Chakradhar, and C. Silva, "StreetAware: A High-Resolution Synchronized Multimodal Urban Scene Dataset," *Sensors*, vol. 23, no. 7, 2023.

[15] M. A. Arefeen, B. Debnath, and S. Chakradhar, "Leancontext: Cost-efficient domain-specific question answering using llms," *Natural Language Processing Journal*, vol. 7, p. 100065, 2024.

[16] X. Zhou, M. Liu, B. L. Zagar, E. Yurtsever, and A. C. Knoll, "Vision language models in autonomous driving and intelligent transportation systems," *arXiv preprint arXiv:2310.14414*, 2023.

[17] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 910–919.

[18] X. Wang, Z. Zhu, G. Huang, X. Chen, and J. Lu, "Drivedreamer: Towards real-world-driven world models for autonomous driving," *arXiv preprint arXiv:2309.09777*, 2023.

[19] C. Liu, X. Zhang, F. Chang, S. Li, P. Hao, Y. Lu, and Y. Wang, "Traffic scenario understanding and video captioning via guidance attention captioning network," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[20] K. Shim, S. Yoon, K. Ko, and C. Kim, "Multi-target multi-camera vehicle tracking for city-scale traffic management," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021.

[21] J. Müller, A. Fregin, and K. Dietmayer, "Multi-camera system for traffic light detection: About camera setup and mapping of detections," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017.

[22] Z. Bai, P. Wang, T. Xiao, T. He, Z. Han, Z. Zhang, and M. Z. Shou, "Hallucination of multimodal large language models: A survey," *arXiv preprint arXiv:2404.18930*, 2024.

[23] Y. Piadyk, J. Rulff, E. Brewer, M. Hosseini, K. Ozbay, M. Sankaradas, S. Chakradhar, and C. Silva, "Streetaware: A high-resolution synchronized multimodal urban scene dataset," *Sensors*, vol. 23, no. 7, p. 3710, 2023.

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[25] X. Chu, L. Qiao, X. Zhang, S. Xu, F. Wei, Y. Yang, X. Sun, Y. Hu, X. Lin, B. Zhang, *et al.*, "MobileVLM-V2: Faster and Stronger Baseline for Vision Language Model," *arXiv preprint arXiv:2402.03766*, 2024.

[26] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[27] L. Chin-Yew, "Rouge: A package for automatic evaluation of summaries," in *Proceedings of the Workshop on Text Summarization Branches Out, 2004*, 2004.