AI-based video analysis for traffic monitoring

Bui Son Tung[1], Phung The Ngoc[1], Do Duy Thanh[1], and Nguyen Hong Thinh[1,*]

[1]*FET, VNU University of Engineering and Technology, Hanoi, Vietnam*
[*]*Corresponding author email: hongthinh.nguyen@vnu.edu.vn*

***Abstract*—Video data from these surveillance cameras is extremely huge. When it requires extracting information about an object or event from the surveillance camera without knowing the exact time, it is very challenging to immediately identify it from billions of video frames. Thus, how to effectively extract information from video surveillance has a strategically important role in practical applications. In this paper, we introduce a system to manage and retrieve surveillance videos based on the indexing of moving objects. The system contains object detection and features extraction, video indexing, retrieval processing mechanism, and fast previewing. Specifically, this system uses information from moving objects to index video content. As a result, in the retrieval phase, the system allows both text-based query and image-based query mechanisms. The system performance are evaluated on traffic surveillance videos and got promising results.**

I. Introduction

Surveillance cameras have become an indispensable part of life. It is becoming more and more popular and can be used in many different public places such as supermarkets, streets, train stations, and even private homes. The amount of video data grows at an amazing rate, requiring intelligent management that allows quick search and retrieval of information. Le et al have mentioned in [3], that to address video indexing and retrieval, both rich indexing, and flexible retrieval workflows are required. For video surveillance, the video analysis should address issues automatically, ensuring real-time and especially allowing playback to know what happen before. Thus, the recommendation system not only enriches the indexing information and supports the retrieval mechanism; but also needs to provide abstract scenes to quickly visualize the retrieval objects. Recently, deep learning based video content analysis approaches have obtained many achievements. In this paper, we present a framework for analyzing and indexing surveillance videos using several popular CNN models. The main content of this work is:

- Introduction of an analysis and indexing system for surveillance video based on deep learning algorithms
- A summary video creation algorithm from retrieved frames when we make a query on the videos database
- Evaluate the performance of the video indexing system on real datasets

The article is organized as follows: The II section provides a background on video analytics systems such as video indexing and video summarization. The recommended video analysis system is described in Section III. The IV section shows the test results and discusses the results. Finally, Section V ends the paper.

II. Background

A. Video indexing

Video indexing is the process of providing viewers with an easy way to access, search, manage, and navigate video content. The selection of indexes can be derived from the content of the video itself and from other source information such as content label, type, time,...of the original video stream. Traditional methods often use metadata such as video-attached tags to a category such as a movie, news, sports events, or online lectures,...They may also use audio files or video subtitles to extract keyword information, thereby assigning a search index. However, these methods are not suitable for surveillance videos, because the content is completely random. Therefore, only content-based methods can be used [3]. Chiang and Yang [2] proposed a method of assigning a surveillance video index based on where and when a moving object appears in the video. In this system, the authors extract all the important moving objects and then rearrange them in a compact format (video abstraction) by preserving the time coordinates and the order of the appearance of the objects. Recently, [6] proposed a method to split keyframe important frames and use deep learning algorithms to extract the object information in the frame to generate a log file describing the object profile information. This information file is used as an index file, used for quick searching when needed. However, the method's limitation is the sketchy object information (only class-id type information). When there is more than one object, the search scheme is not efficient.

B. Video summarization

Video synopsis is an effective solution to create an abstract version that summarizes video content and shortens viewing time. The method first proposed by Rav-Ache et al [9], allows for displaying all events in a shorter time period, based on the reordering of events/objects that appear in the video in both space and time. Due to the arrangement of many objects at different times into the same frame; these algorithms, in general, face the important problem of *tube collisions* when many objects can appear at the same location. Several studies have been proposed to provide the optimal sorting methods, in order to increase the time compression ratio as well as reduce the number of collisions [8], [1], [4], [7]. Nie et al [8] proposed a method to change the position in both the time domain and the space (position) of the objects by expanding the background image to accommodate multiple objects. Li et al [4], in contrast, proposes another approach to reduce tube overlap by minimizing object size. They calculate dimensionality reduction metrics for each object based on that object's interaction with surrounding objects. [7] proposes a method to create a collision-free video that will pre-compute the positions of tubes and if a collision occurs, it allows reducing object size, translating object location in both time and space. However, the biggest limitation of that method is the long computation time.

In a surveillance situation, to visualize and reconfirm an event; users need to review and compare retrieval results from time to time if multiple search results are returned. For that reason, in this study, we used the idea of video synopsis and propose a simple preview solution for our video management system.

III. Proposed video management system

In surveillance video sequences, the objects play an important role. Thus, it is possible to store surveillance video information through the information of objects in the video. In addition, to be able to easily filter or search video content, detailed object characteristics will be used as index information. For the traffic monitoring video processing problem, the main objects are people and vehicles circulating on the road. Therefore, we initially define index information such as information related to the object such as object type, color, Timestamp, and moving directions...; then from which to select the suitable algorithms for extracting such information. Based on the index information definition, our framework is divided into two main stages: (i) object detection and tracking: detecting moving objects and tracking those objects during the movement in the video and (ii) extracting information about the objects: object classification, color classification ...of the object being tracked. The detail of the processing is shown in Fig 1

A. Object detection and tracking

1) Active frame selections: Surveillance video has the property that moving objects do not always appear in the frame. To reduce the execution time of object decomposition algorithms on all frames, we implement a background subtraction algorithm to remove frames without motion. The background frame is updated based on static frames obtain every ten minutes. In this way, the background is adaptive to the variation of weather or light conditions...As a result, the processing time for object detection or tracking is significantly reduced. To reduce the effects of light or weather, the background frame will be regularly re-evaluated and updated based on the previous motionless frames based on the Median filter algorithm.

2) Object detection, segmentation and tracking: There are many methods of object detection, segmentation, and tracking that can be applied, but the methods that apply deep learning algorithms bring outstanding results and the computation time is also greatly improved. Based on the research results of Vishal Mandal and Yaw Adu-Gyamfi [5] which surveyed and evaluated the accuracy of a number of several ways to combine famous object detection and tracking algorithms. They test the combination approaches on real vehicle monitoring video data obtained from traffic surveillance cameras in the US. The result shows that using CenterNet+DeepSORT gave the best results.

B. Features extraction

We use different sets of visual features at different levels to distinguish moving objects. Such information is extracted from video frames and from detected objects using AI algorithms. The relative importance of this feature method compared to another feature method may vary according to the application or the point of view of the surveillance camera. The use of multiple features also allows detailed object indexing information, or retrieval back when needed. The following descriptors have the highest overall performance for both search testing and concept modeling:

- Object Class-ID which can be Vehicle type (such as car, truck, bus, motorbike) or pedestrian. Most CNN-based detection algorithms allow locating and classifying objects. The information is obtained from object detection and tracking step
- Timestamp: The time the object first appears (GMT-time) in the video frame, can be obtained from the video sequence.
- Color: Color information of the object. This information can be extracted using some simple color recognition network or based on some color features such as color histogram, and color moment. In this paper, we build a small CNN network to recognize 10 basic colors of vehicles such as black, blue, brown, golden, green, grey, white, yellow, red, and silver.
- License Plate: This information can be extracted using two CNN networks: license plate detection network and OCR recognition network. However, the vehicle's license plate information is not always verifiable, due to the viewing angle and quality of the surveillance camera.
- Moving direction: the direction of the object's movement: This is also important information that allows the search to retrieve information quickly. Movement direction can be verified through tracking. Based on the change of the object's position over time in the tracklet, we can determine the direction of the object's movement.
- Embedded feature vector: Recently, the problem of finding and relocating objects in many cameras has been interesting. The encoding of the object information that allows searching at different angles is therefore necessary. Embed features vector is an object coding vector; extracted using the Car-ReID model; for the purpose of allowing object searching across multiple cameras.

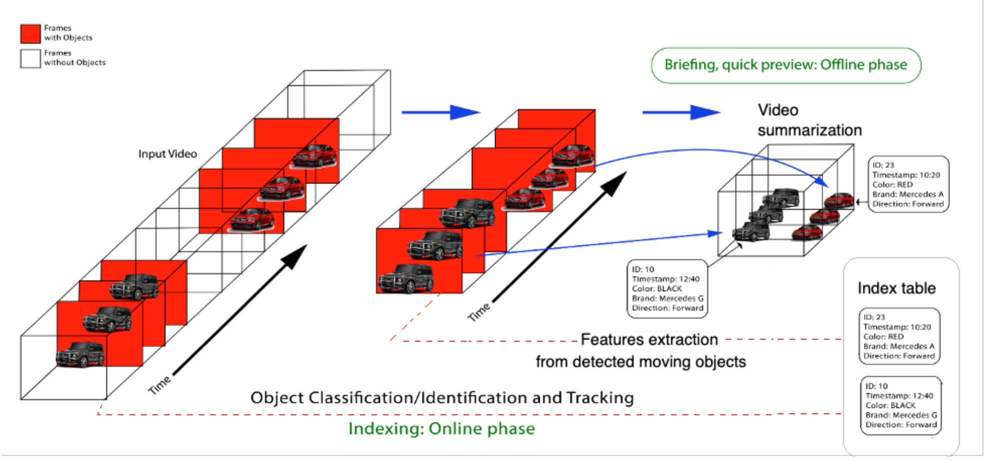

Fig. 1. Video indexing system architecture

C. Indexing

After extracting information of objects in the sequence, each object appearing in the surveillance video will be assigned some basic information such as ID (based on track-id), vehicle type (obtained from the detection algorithm and object separation); tracklet-id (sequence of the object's position in the video); movement direction (based on object's position in consecutive frames), color, license plate ... The information is stored as a text database, so the size of data is very small compared with the video's original size. In addition, to facilitate the retrieval and search of information in the video, an index matrix is also created for all objects in the video sequence, where each row has the index information of an object; the column will contain the corresponding information of features (as vehicle type, color ...) so that object can be extracted when needed.

D. Smart query and retrieval

Our system provides the ability to allow users to query and retrieve the desired object based on the basic properties of the object. Usually, the search is performed by the observer. By storing the index as a numeric matrix above, searching and filtering information will be done quickly based on computational techniques on the matrix. The system allows searching based on descriptive keywords and also based on visual content (query by image or custom region)

1) Query by Keyword: For example, if you need to find a red car that appears in the period from 2:30 to 3, the process is simply based on filtering the information in the matrix above. First of all, because the process of storing and processing information is continuous, the rows in the matrix are ordered in time. We use the prior time information to limit the number of rows to search; then find the rows of the matrix where the 4th column is 3 and the 5th column is 8. The result returns a list of rows with matching object information. Once we have the object's ID information, we can observe them based on the frame number information in which it appeared.

2) Query by an image or a region: When taking the image as a template for searching, the input image will be passed through a series of object decomposition and feature extraction algorithms to create a search vector in index vector format. The Euclidean distance function (L2) is used to measure the difference between the feature vectors and the index matrix. Top-k closest approximations will be returned; based on the list of results, we can display the same information as when searching with keywords. Further, we can search by embedded feature vector by applying a trained Car-ReID network.

E. Visualization the retrieval results

In general, even after filtering out the object information, the re-verification step is also quite time-consuming and requires manual operations, due to having to review each frame of moving objects. If the returned candidates list contains many objects, the process is even longer. To reduce playback and manipulation time, a method of creating video snapshots is required. A lot of methods have been proposed to create an optimal video synopsis (as the review in Sec II) in terms of shortest time viewing time or least tube collisions. After surveying and running experiments in the real system, we found that implementing an optimal algorithm takes expensive computation and processing time. Therefore, we set the minimum system requirements to meet. Those requirements are as follows: (i) Enough (contains enough information about the objects), (ii) Short (the quick video is as short as possible); (iii) No anomalies (objects disappear, object positions are wrong, objects are overlapped) and (iv) Fast (generate video summaries quickly). For that purpose, we introduce the algorithm, which is a more flexible version of [4] and [7]. The detail of the method is described below:

1) Resize object: During the experiment, we noticed that the collision between objects occurs because the size of the bounding box is too large and takes up a lot of space in the frame. It also becomes difficult to arrange the appearance order of the vehicles to avoid conflicts difficult because each vehicle has a different travel speed. Therefore, we used the idea from the study of Li et al [4] to reduce the objects appearing in the frame. However, unlike [4], the object is only shrunk when a frontal collision is detected, here before placing an object in the static background, the size of the object will be reduced by a distance by multiplying with a predefined parameter; thus the object will be scaled from the beginning. Due to the advantage of having information available from the object's index vector, the reduction does not lose information.

2) Spatial and temporal shifting: Another way can solve the collision problem is to allow the object to be moved not only in the time domain but also in the space domain. Nie et al [7] proposed a method to change the object's initial position in the time domain to reduce the length of the summary video and translate it into the space domain to avoid collisions with other vehicles. However, this method is only effective in cases where the vehicles are only moving in a single lane. We apply the main idea of [7], expand the real road from one lane to three parallel lanes, and move the subject from the middle lane to the sides. In addition, the object is also scaled down so, as a result, the object density can be doubled.

IV. Experiments

A. Training the deep-learning models

The core of the system is the algorithms for extracting video content information, which focuses on algorithms for object detection, segmentation, object tracking as well as information extraction from objects (color recognition, OCR...). All these algorithms use deep learning methods. To ensure performance, we trained the network on published datasets such as the AI-city challenge 2019 traffic monitoring video dataset, and the VERI dataset and finetune the model on Traffic surveillance camera video obtained on Vietnamese roads. In this study, we use Nvidia GTX 2080TI GPU to perform training, train the models, and use fine-tune technique to shorten the training time. Detailed information about the CNN networks used for each task is described in Table I.

B. Evaluation

1) Indexing performance: To evaluate the performance of the indexing step, we use several widely used statistical evaluation metrics, including classification accuracy (ACC) [for classification task], Multiple Objects Tracking Accuracy [MOTA], Number of switching-id (ID Switched-IDS), and frame-rate (number of processing frame per second) [for real-time evaluation] [10]. The performance of the system is evaluated on short videos, 6 minutes long each, obtained in Hanoi, Da Nang, and Nha Trang, with different views and conditions. The results are shown in Tables III andII. It can be seen in II that the tracking algorithms are reliable, and can extract all the objects in the surveillance video. Using a time-stamp to filter objects is not afraid of being missed. Table III shows the reliability of the information used to index objects in the video. As can be seen, the color and number plate are less reliable. This is due to changing shooting conditions resulting in poor color recognition; as well as vehicle angles and movements leading to incompatible license plate recognition models. Adding more data or increasing the quality of surveillance cameras can improve the model.

2) Performance of Query and Retrieval step: We also tested the system's performance when using the search for index information stored in the system's database. We perform a search by keyword vehicle type and vehicle color, as well as query by image, which is a frame taken from a surveillance video. In the case of a query by keyword, we filter the information directly from the database. When searching by keyframe, we re-apply the system pipeline with detection to compute the

TABLE I
Networks use in our system

Task	Method
Vehicle Detection	CenterMask
Vehicle Tracking	DeepSORT
Color Recognition	MobileNet-v2
License plate detection	Yolo-v4
License plate recognition	EasyOCR
ReID	Car-ReID

TABLE II
Accuracy of tracking algorithm and processing times

Sequences	Properties	MOTA	IDS	Frame-rate
Video1	Left views,205 objects	99.03 %	2	18.6
Video2	Right views,229 objects	98.7 %	3	18.6
Video3	Center views,266 objects	99.62%	1	19.3

TABLE III
Reliability of the information fields in the index vector in our system

Indexing field	Precision
Object class (type)	99 %
Object's timestamp	100 %
Object's colors	89.97 %
License plate recognition	88 %

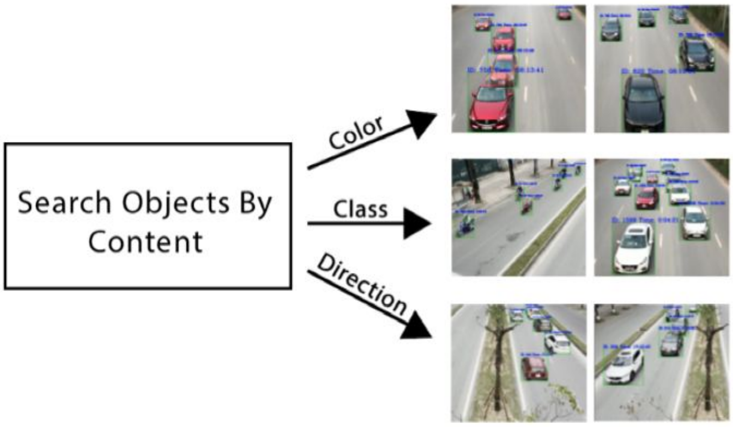

Fig. 2. Results of video summarization algorithms

embedded vector and use this feature vector to compute and generate the potential list. Based on the top-k images retrieved, we calculate the accuracy. The results are presented in Table IV. The obtained results show that the search for information based on the index has high accuracy. Accepting top-k with a large k will improve the result. However, it also takes more time for users to filter themselves based on observations. Search based on object type gives good results; vice versa based on color characteristics is not so accurate, since color changes with different illumination conditions. In contrast, an image-based search gives fairly reliable results (90%).

TABLE IV
RETRIEVAL ACCURACY OF INDEXING SYSTEM

Query by	top-1			top-5		
	Precision	Recall	F1	Precision	Recall	F1
Type	0.95	0.97	0.96	0.98	0.98	0.97
Color	0.82	0.7	0.76	0.93	0.87	0.9
Image	0.9	0.84	0.87	0.97	0.93	0.95

C. Video summarization

We evaluate the performance of our proposed summarization approach based on some metrics. Firstly, we calculate the Compression Ratio (CR) by the following equation:
$CR = \frac{f_{original}}{f_{sum}}$, which $f_{original}$ is refers to the total frames of the original video and f_{sum} is the total frames of summarization video. The higher the index, the better the system's ability to summarize information. We also calculate the sum of frames in which the overlap problem occurred called total collision frames, which represents the visual quality of the summary video. The smaller this number, the better the summary video to observe. In our study, we examine our proposed methods with the same traffic videos which are recorded on the highway. Most videos contain a crowded density of objects so that we cannot reduce entirely the overlap while keeping an acceptable compression ratio. The performance results of the proposed algorithm are shown in Table V. The result shows that reducing and translating objects can improve results significantly. Besides, we also compare the proposed algorithm with the algorithm introduced in [7], in terms of processing time. It can be seen in Table VI, that when the number of objects used to create surveillance videos is large, the proposed algorithm gives good results

TABLE V
VIDEO SUMMARIZATION PERFORMANCE

Video	Total	Object resize		Object resize+shift	
	frames	CR	Collision frames	CR	Collision frames
#1	8874	5.33	0	8.73	0
#2	8670	4.79	104	7.92	72
#3	7680	9.64	19	15.33	8

After retrieval, we can create summary videos from received frames to observe the results quickly. The example results are shown in Fig. 2

V. CONCLUSIONS

In this paper, we presented a framework for indexing and retrieving surveillance video content based on deep learning algorithms. This work is only described as a simple prototype system so that a real system can be developed. The obtained results show that the system works well, meets the actual requirements, and is feasible to build in real applications. The performance of the system can be further improved by modifying advanced CNN models and training on more diverse video data.

TABLE VI
PROCESSING OF VIDEO SUMMARIZATION STEP

Video seq	No of Tubes	Method [7]	Our
video1	10 %	43	47
video2	20 %	58	76
video3	30 %	215	**111**

REFERENCES

[1] Kemal Batuhan Baskurt and Refik Samet. Video synopsis: A survey. *Computer Vision and Image Understanding*, 181:26–38, 2019.

[2] Cheng-Chieh Chiang and Huei-Fang Yang. Quick browsing and retrieval for surveillance videos. *Multimedia Tools and Applications*, 74(9):2861–2877, 2015.

[3] Thi-Lan Le, Monique Thonnat, Alain Boucher, and François Bremond. Surveillance video indexing and retrieval using object features and semantic events. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1439–1476, 2009.

[4] Xuelong Li, Zhigang Wang, and Xiaoqiang Lu. Surveillance video synopsis via scaling down objects. *IEEE Transactions on Image Processing*, 25(2):740–755, 2015.

[5] Vishal Mandal and Yaw Adu-Gyamfi. Object detection and tracking algorithms for vehicle counting: a comparative analysis. *Journal of Big Data Analytics in Transportation*, 2(3):251–261, 2020.

[6] Duaa Mohammad, Inad Aljarrah, and Moath Jarrah. Searching surveillance video contents using convolutional neural network. *International Journal of Electrical and Computer Engineering*, 11(2):1656, 2021.

[7] Yongwei Nie, Zhenkai Li, Zhensong Zhang, Qing Zhang, Tiezheng Ma, and Hanqiu Sun. Collision-free video synopsis incorporating object speed and size changes. *IEEE Transactions on Image Processing*, 29:1465–1478, 2019.

[8] Yongwei Nie, Chunxia Xiao, Hanqiu Sun, and Ping Li. Compact video synopsis via global spatiotemporal optimization. *IEEE transactions on visualization and computer graphics*, 19(10):1664–1676, 2012.

[9] Alex Rav-Acha, Yael Pritch, and Shmuel Peleg. Making a long video short: Dynamic video synopsis. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 435–441. IEEE, 2006.

[10] Yi Zou, Weiwei Zhang, Wendi Weng, and Zhengyun Meng. Multi-vehicle tracking via real-time detection probes and a markov decision process policy. *Sensors*, 19(6):1309, 2019.