https://doi.org/10.7840/kics.2022.47.5.710

Syntax elements 기반 움직이는 객체 검출 기술

김동기•°, 로즈마리 코이카라*, 김선정**, 신소명**, 엄제원*, 김민중*, 홍주희**

Moving Object Detection Using Syntax Elements

Dong-Key Kim•°, Rosemary Koikara*, Sun-Jung Kim**, So-Myoung Shin**,
Jewon Eom*, Min-Joong Kim*, Joo-Hee Hong**

요약

비디오 스트림에서 움직이는 물체를 감지하는 것은 컴퓨터 비전에서 가장 필수적인 과제 중 하나이며, 그중에서도 배경 모델링은 모션 감지에 가장 일반적으로 사용되는 기술이다. 본 논문에서는 비디오 스트림에서 저장되는 압축 정보를 이용하여 움직이는 객체를 검출하는 기법을 제안한다. 비디오 스트림 내 syntax elements를 추출 및 분석하여 객체의 움직임 정보를 산출한다. 주요 목표는 비트스트림 영역에서 움직이는 객체를 포함하는 ROI(region of interest)를 추출하는 것이다. 제안하는 기술의 세부 내용 서술과 실험 결과 비교를 통해 이전의 연구 결과와 비교 분석하였다. 높은 연산 복잡도 문제로 인해 실시간 구동이 어려웠던 기존 연구와 달리 본 논문에서 제안하는 기술은 낮은 연산 복잡도를 통해 이를 해결하였다. 또한 움직이는 객체 검출 기술에서 발생 가능한 문제에 대해서도 자세히 설명한다.

키워드 : 교통 관제, 영상 분석, 객체 검출, 움직임 검출, 최적화
Key Words : Transportation surveillance, Video analysis, Object detection, Motion estimation, Optimization

ABSTRACT

Detecting moving objects in video streams is one of the most prominent challenges in computer vision. Background modeling and subtraction is the most commonly used technique for motion detection. This paper proposes a technique that uses information extracted from a video stream in the compressed domain to detect objects in motion. The syntax elements that store motion information are processed and analyzed. The main objective is to extract the region of interest (ROI), which are the areas of the high-resolution video bitstream that contain objects that are moving. The objects within these regions can then be classified into respective classes. Experimental results and detailed analysis compare our proposed method to previous work. Previously used techniques can be very computationally intensive and hence challenging to perform in real-time. We also discuss at length the challenges that we come across in moving object detection.

Ⅰ. Introduction

In recent times, video analytics has evolved by leaps and bounds. According to research, there are a huge amount of surveillance cameras used and this number is expected to reach 1 billion[1]. The reason for this massive increase is that there has been a massive growth in urbanization[2]. It goes without

•° First and Corresponding Author : University of Seoul, Dept of Transportation, kim650919@gmail.com, 정회원
* Pintel Ltd., rosemary@pintel.co.kr; jweom@pintel.co.kr; mjkim@pintel.co.kr, 정회원
** University of Seoul, Dept of Transportation, kimjala@naver.com; ssm0503@naver.com; bluesea4u@naver.com
논문번호 : 202112-337-A-RN, Received December 20, 2021; Revised February 24, 2022; Accepted March 7, 2022

saying that many people living within meters of each other will invariably increase crime rate. That said, CCTV camera surveillance has been used for monitoring for quite a while, be it roads, subways, retail stores, etc.

Video surveillance has never been more extensive than it is now. As a result, it is more efficient to use intelligent automated video analysis systems. The main objective of video analytics systems is to detect objects. Therefore, it is imperative to achieve high detection accuracy, with the lowest possible false alarm rates and detection misses.

In object detection, the first step is to identify objects in the video sequence and cluster pixels of these objects. A video comprises a sequence of images. There are both moving and static objects in the sequence of images. Analysts generally refer to the moving objects as foreground objects and the static part as the background objects. Moving object detection aims to identify the moving object relevant to that particular scenario[3,4].

In the past, object detection procedures were confined to frame differencing, background subtraction, and optical flow[5]. However, these techniques though easy to implement, are not efficient and cannot be adequately adapted to volatile environments[6].

Videos generated by CCTV surveillance devices generally undergo compression to reduce bandwidth and storage during video surveillance. In this paper, we consider the H.264/AVC[7] video coding standard introduced by the Moving Picture Experts Group (MPEG) and the Video Coding Experts Group (VCEG). Techniques involving video analysis generally decode the H.264 bitstream before being analyzed. However, over the years, researchers have made efforts to recognize moving objects immediately on the compressed video stream to eliminate this decoding stage.

Refs.[8-15] obtained motion vector (MV), DCT coefficient, and MB partition information from the bitstream to detect moving objects. However, these methods raise concerns where shifts in illumination and motion in the background result in false object detections. Most of these methods assume that there is no background motion and no illumination shifts.

Nevertheless, most surveillance videos will not be devoid of the above factors. Hence, these methods are not practical. This means that actual applications cannot be completely automated. Some form of human intervention will be required.

In this paper, we propose a real-time object detection technology that is lightweight and fast and that can perform detection on a live stream for the purpose of surveillance. To achieve this, the following are the relevant areas regarding the performance we are concerned about:

1) Computation cost: When developing a detector, consideration of only the accuracy is insufficient. We require the computational cost to be low while maintaining the required accuracy. Hence, our developed system is lightweight.
2) Speed: The detector developed must work in real-time, as we require it to detect motion in a live stream. In order to achieve this, all the critical operations have to be time-sensitive.
3) ROI: A bounding box has to be drawn around the appropriate ROIs. These are the resulting objects that have some motion.

Also, our goal is the commercialization of the product and hence is not restricted to a laboratory environment.

The rest of the paper goes into the details of the technique proposed in this paper. Chapter two discusses the related work. The proposed system is described in chapter three. In chapter four, the experimental results are shown. Finally, it is concluded in chapter five.

Ⅱ. Related Work

Video object detection is a fundamental problem in computer vision and has a broad spectrum of applications. Research has been done on both using the motion information is a video as well as using deep networks on still images. Applying deep learning object detectors to each frame of a video as you would for still images causes redundant computational cost. Accordingly, a frame work that can use temporal information for real-time detection is crucial.

2.1 Motion Information for Object Motion Detection

Prior to 2012, analysts have performed substantial research to implement moving object detection using bitstream and in the H.264 compressed domain[7-9, 11-17]. However, with the onset of machine learning and deep learning, researchers have concentrated on using them to solve object detection problems. Hence, following this, there has been a significant reduction in research in object motion detection using parameters extracted directly from the bitstream.

Ref.[16] introduced using a spatiotemporal graph for detection and tracking objects using H.264 bitstream. But their technique is based on grid calculation and has a high computational cost. Though Ref.[8] used texture, form, motion dissimilarity energies, the resultant detection was not suitable because of the noise generated by the technique. Ref.[9,17] used a combination of data obtained from compressed domain and pixel domain to determine the moving object. These techniques unfortunately work only with fixed GOP and are computationally intensive. In Ref.[11] the P-frame and I-frame are analyzed separately. Their use of background modeling leads to a higher computation cost.

2.2 Real-time motion detection H.264 compressed domain for surveillance application[18]

In the technique proposed in this paper the video stream is analyzed in the compressed domain and features are then combined after which the genetic algorithm is utilized. The genetic algorithm has been included for instances when there is a large movement. Finally, Gaussian filtering is performed followed by object segmentation.

2.3 H.264/AVC

The H.264/Advanced Video Coding is the most widely used industry standard for video coding. The main challenge for video transmission over a network has been the limited channel capacity and the unexpected channel behavior[19]. The video encoders exploit the spatial and temporal redundancies between the frames to overcome the problems mentioned above.

In the H.264/AVC coding standard, the input image is divided into fixed-size blocks as the basic unit of encoding, and it is called a macroblock, which includes a luma block and two chroma blocks. The size of the luminance block is 16×16. For simplicity, the block sizes described in the article are all the size of the luma block. If 4:2:0 sampling is used, the chroma block size is half the size of the luma block. In the prediction process, the macroblock is further divided into small blocks for prediction according to different prediction modes. In intra prediction, the macroblock can be divided into small 16×16, 16×8, 8×16 and 8×8 blocks. If the division method is 8×8, the divided four blocks are called sub-macroblocks, and the sub-macroblocks can be divided again 8×8, 8×4, 4×8, and 4×4 small blocks. Each small block can independently perform motion estimation, but the small block in each sub-macroblock can only refer to the same reference frame.

In the transform and quantization process, the macroblock is divided into 4×4 and 8×8 small blocks, and the prediction residuals in each small block are transformed and quantized respectively to obtain the quantized coefficients.

Each coded macroblock consists of the following information:

1) MB type: I (intra-coded), P (inter-coded from one reference frame) and B (inter-coded from one or two reference frames)
2) Prediction information
3) Coded Block Pattern (CBP)
4) Quantization Parameter (QP)
5) Residual data

2.4 One-stage and Two-stage detectors

In this paper, we compare our system to two-stage detectors, because though not wholly similar, these are the state-of-the-art detectors that come closest to our detector to be on an even ground while comparing. The two-stage approach to detectors[20-22] first generate candidate regions, i.e., regions with the highest probability of containing objects. Then, in the second stage, the candidate regions are passed to the upper layers that generate the bounding boxes and classify the objects.

Ⅲ. Proposed Work

In any object detection system, the first step is to identify the location of objects in a video stream. Most CNN-based models are expensive in computation, occupy a lot of CPU memory. Though recent developments in deep learning technology have significantly improved the object detection rate, the need for downsampling for image analysis drastically influences the computation speed. Also, this leads to the quality of the images to reduce, which hampers the detection of small objects. This research focuses on developing a fast and lightweight object detection technology that uses syntax elements from H.264 compressed domain bitstream generated from high-resolution video.

3.1 Challenges

The following are the challenges that we had to overcome while implementing the detector proposed in this paper:

1) Noise: While extracting motion information, a lot of the noise has to be removed.
2) Illumination variation: There might be constant changes in the lighting of the scene, which results in the algorithm observing motion information alterations.
3) Parameter correlation: There has to be a correlation between the extracted elements to detect accurate object motion.
4) Complex background: We have had to ensure that there was minimal movement in the background and that still cameras were used for recording the video.

3.2 Syntax Elements

In the previous chapter, we have discussed H.264/AVC standard for video compression.

The H.264 compressed domain video sequences consist of all the information required to reproduce the respective videos. The following parameters that have named ‘syntax elements’ are extracted from the bitstream and used for developing the detector.

1) Macroblock Size (MB): The macroblock size is the size in bits that a macroblock occupies. When analyzing an H.264/AVC compressed bitstream it has been observed that the MB size is an indication of motion within that macroblock.
2) Motion vector: This vector indicated a displaced region that will be used for motion compensation. A reference frame and the current frame are considered for calculating the motion. A region in the current frame is located and compared to the corresponding similar region in the reference frame. The motion vector information is the offset between the reference frame and the current frame. The motion vector is a combination of two values: Motion magnitude and Motion angle.
3) Quantization Parameter: A block of residual samples is transformed using a 4×4 or 8×8 integer transform, an approximate form of the Discrete Cosine Transform (DCT). The transform outputs a set of coefficients; each is a weighting value for a standard basis pattern. The output of the transform, a block of transform coefficients, is quantized, i.e., each coefficient is divided by an integer value. Quantization reduces the precision of the transform coefficients according to a quantization parameter (QP). Typically, the result is a block in which most or all coefficients are zero, with a few non-zero coefficients. Setting QP to a high value means that more coefficients are set to zero, resulting in high compression at the expense of poor decoded image quality. Setting QP to a low value means that more non-zero coefficients remain after quantization, resulting in better-decoded image quality but lower

compression.

4) Residual Data: The encoder processes a frame of video in units of a macroblock. It forms a prediction of the macroblock based on previously-coded data, either from the current frame (intra prediction) or from other frames that have already been coded and transmitted (inter prediction). Finally, the encoder subtracts the prediction from the current macroblock to form the residual data.

The H.264 compressed domain video sequences consist of all the information required to reproduce the respective videos. We manipulate the elements as mentioned above to obtain the required ROIs. This is better employed in real-time or fast object detection and tracking systems.

3.3 Methodology

Figure 1 shows the basic schematic diagram of our proposed moving object detection process. This detector analyzes the encoded H.264 bitstream to detect the presence of objects. The syntax elements are used to detect objects in real-time. The H.264 bitstream, as shown above, contains information such as motion vectors, residual data, and macroblock types which can be used for object detection and tracking.

We manipulate the motion magnitude, motion angle, macroblock size, quantization parameter, and residual data values to obtain the required ROIs. This works seamlessly for real-time or fast object detection and tracking systems. As mentioned in the previous section, this technology presented us with some challenges. The algorithm we have employed gets rid of these challenges and removes noise.

The basic process is:

1) Extract syntax elements from H.264 compressed domain bitstream.
2) Process the following parameters:
   (1) Motion magnitude
   (2) Motion angle
   (3) Macroblock size
   (4) Quantization parameter
   (5) Residual data
3) Determine *n* frames
4) Aggregation of parameters by *n*
5) Extract ROIs
6) Draw bounding boxes around moving objects

The regions that have been determined to contain moving objects can then in the future be classified. Figure 2 shows the proposed detector with a classifier.

On extraction of the syntax elements, each of them contain some noise. The noise is gotten rid of by setting a threshold for each syntax element. Most related research work analyze the parameters for

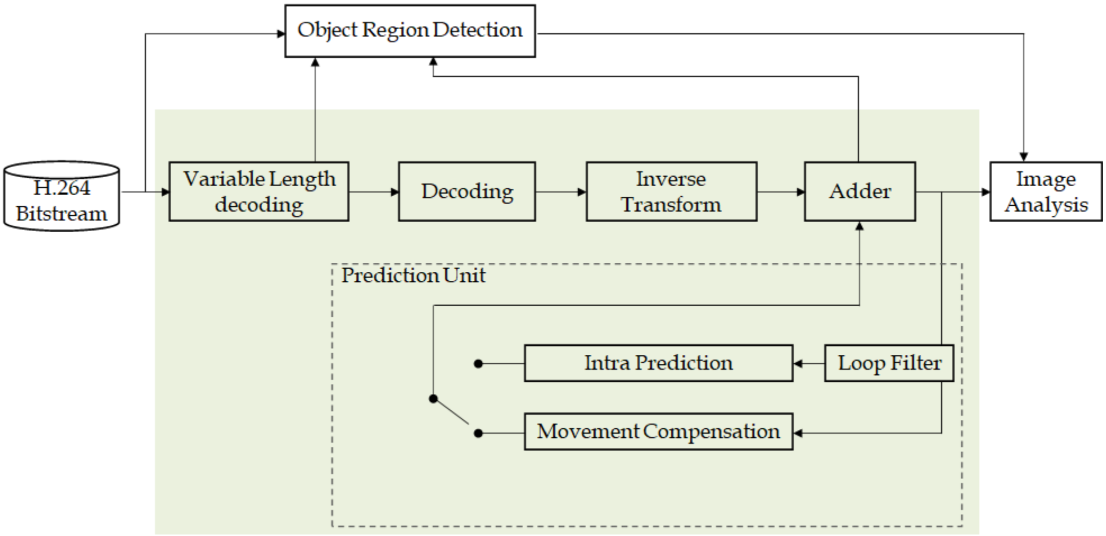

Fig. 1. The Architecture of Proposed System

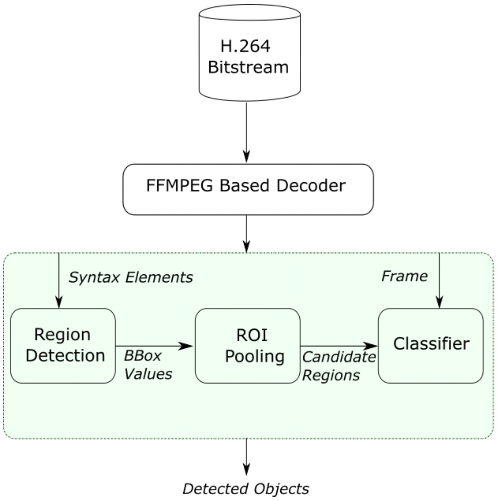

Fig. 2. Proposed Detector with Classifier

each of the frames separately and determine the object region for each of those frames. In our proposed algorithm we analyze the syntax elements on each of the frames, but the determination of the bounding box is done by the aggregation of parameters over a set number of frames. Now this set number of frames will change depending on the characteristics of the scene and the environment in which the videostream is collected from. The basic flow of the proposed algorithm is shown in figure 3.

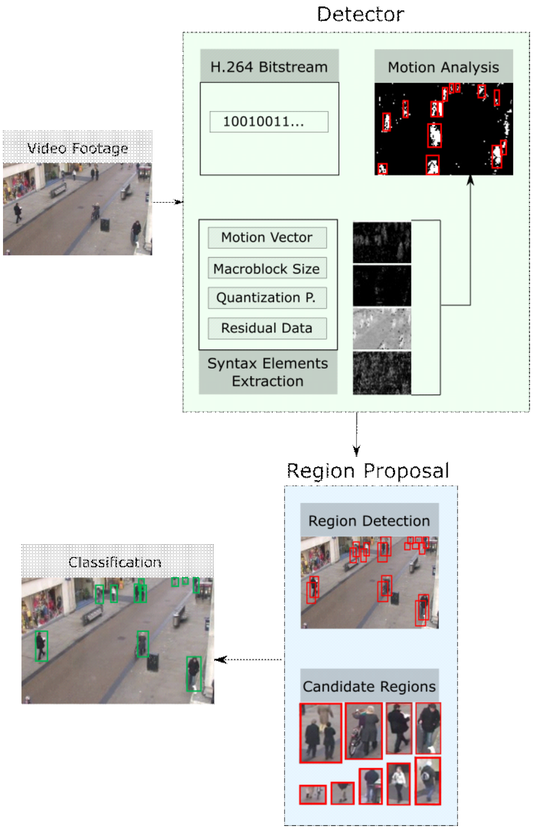

Fig. 3. Architecture of Proposed Detector

Ⅳ. Experimental Results

Current technology cannot work directly with videos of high resolution. Over time, the quality of videos from CCTV or other surveillance sources is improved. We are now at a point where 4K resolution is widely used. However, because of computational speed, constraints on hardware resources, and limited storage space, the analysis cannot be done on high-resolution videos in real-time. High-resolution videos are downsampled before it is analyzed.

In this proposed work, an analysis of the video is done without lowering the resolution of the video. Now, the encoded bitstream that has been transmitted directly by the surveillance device is used to analyze the video. We extract specific parameters that we call syntax elements from the bitstream; these are then processed to analyze the video. This significantly reduces the computation time compared to using conventional deep learning. Also, there is no pixel quality loss as downsampling is being avoided when using this technique.

4.1 Evaluation Specifications

We use the well-known PASCAL VOC dataset [23] with 20 object classes, training is performed on the VOC 2012 trainval (10K images). We only report mAP using IoU at 0.5 on a video dataset with resolution 1920×1080. We use the same training and testing configurations for a fair comparison.

For testing, the evaluation was done on a platform with Windows 10 as the operating system, Intel(R) I9-10900X FPU @3.70GHz, 128GB RAM and a NVIDIA GeForce GTX 1650 SUPER GPU.

4.2 Comparison with other Research Work

Table 1 shows the comparison between the decoding time taken by a regular CNN detector versus the entire detection process of the proposed bitstream detector. The values in the table shows the time taken by a CNN detector only for the decoding process. After the decoding is done the frames then have to be analyzed for objects.

As discussed previously, the detector proposed in this paper is a two-stage detector. Hence, we compare it with other state-of-the-art two-stage detectors. The main objective of this research work is to analyze high-resolution videos to detect motion objects for the purpose of surveillance. In order to do this we needed to consider appropriate research that would provide a fair comparison. Table 2 compares the mAP over IOU of the proposed detector with Faster-R-CNN[21], FPN[22] and Ref.[18] also two-stage detectors. GES (Global Effective Stride) is the factor by which an input image is downscaled with respect to the feature map that is the input to the region proposal method. A GES of 16 means that a 16×16 object is represented by just one pixel in that feature map. As shown in Table 2, the proposed detector has a GES of 1. This is because the input to the detector is not downscaled. Because we do not downscale the video the integrity of video is maintained.

Table 1. Comparison of Decoding Speeds by Channels

Channels	Video	CNN (full-frame decoding) [(ms/ch)/fps]	Syntax (frame and syntax decoding) [(ms/ch)/fps]
1 channel	8988	72,234 / 122.7	105,991 / 84.8
2 channels	8988*2	79.481 / 113.1	112,444 / 79.9
4 channels	8988*4	92,759 / 96.9	129,293 / 69.5
5 channels	8988*8	132,080 / 68.0	184,997 / 48.6

Table 2. Performance Comparison

Method	GES	mAP, 0.5	fps
Faster-R-CNN	16	29.3	18.5
Faster-R-CNN	8	45.9	17.8
FPN	4	48.5	40.9
Kim et al.[18]	1	39.71	131.4
Proposed Detector	1	49.24	136.6

4.2.1 Faster R-CNN[21]

Faster R-CNN detection system is composed of two modules. The first module is a deep, fully convolutional backbone network which outputs global feature maps. An RPN takes images of any size as input and outputs a set of rectangular object proposals, each with a confidence score of it being an object.

The drawback of Faster R-CNN is that it does not detect small objects as well because of the GES factor and the fact that the anchor boxes are manually predefined.

4.2.2 FPN[22]

Faster R-CNN has inspired many works. Feature pyramid networks (FPN) were proposed to crop RoI features from different levels depending on the scale.

Though the FPN does provide an improvement over the Faster R-CNN, it has a GES factor of 4 that leads to a significant memory overhead and puts a strain on the GPU.

4.2.3 Real-time motion detection in H.264 compressed domain for surveillance application[18]

This algorithm uses the following features to extract motion: luma DCT coefficients, chroma DCT coefficients, motion vectors and macro blocks. The use of DCT coefficients means that the video bitstream has to be partially decoded and hence increases the computational complexity. This algorithm has a limitation in that it does not detect the precise movement by using macroblocks in the compressed domain.

4.3 Observations

Using macroblock information extracted from a H.264 bitstream for motion detection guarantees detection speed. This detector is an advancement over previous related work. We observe the following advantages in our scheme over the previous research:

1) Industrial level usage: The main objective of this system is to develop a multichannel live stream motion detection system that can be used at an industrial level.
2) Computational cost: From Table 1 and Table 2 we can see that the computational cost of proposed scheme is lower. Most of the related research available has contributed to the likes of calibration and filtering background modeling. Our objective is to improve motion detection; we have not included an extra module for calibration, filtering, or background modeling to hamper the computational cost. Also, a reason for the reduction in computational cost is the aggregation module that combines the results of a set number of frames to determine the ROI.
3) High-Resolution analysis: An objective of the algorithm is to develop a technology that performs object detection in real-time on high-resolution videos without downsampling.
4) Real-time detection: The related research in this area mainly focuses on the detection system's accuracy. Speed, computational cost, and accuracy are significant factors in industrial applications. This is because motion detection has to be performed on a live stream. From Table 2 it can be discerned that the frames processed per second is much higher than Faster-R-CNN and slightly higher than FPN detection systems.
5) Minimal operations: The motion is detected by extracting only the syntax elements. We have compared and analyzed the motion in the compressed bitstream; this implies that decoding is avoided. Therefore, an accurate analysis is performed at high speed without loss of pixel information.

Figure 4 and 5 show the detection results of the proposed scheme.

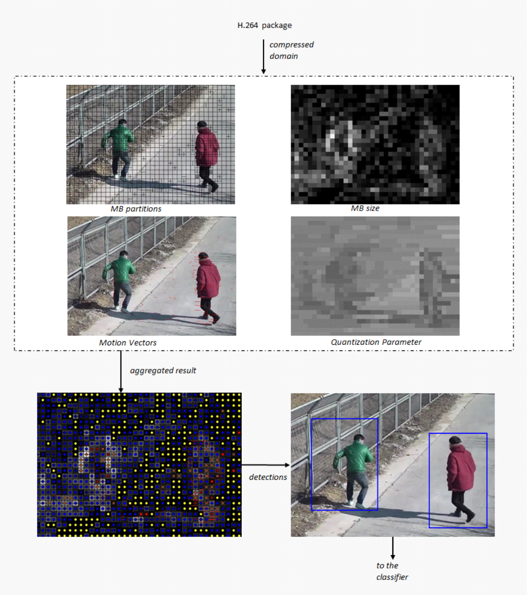

Fig. 4. Visualization of the detection process

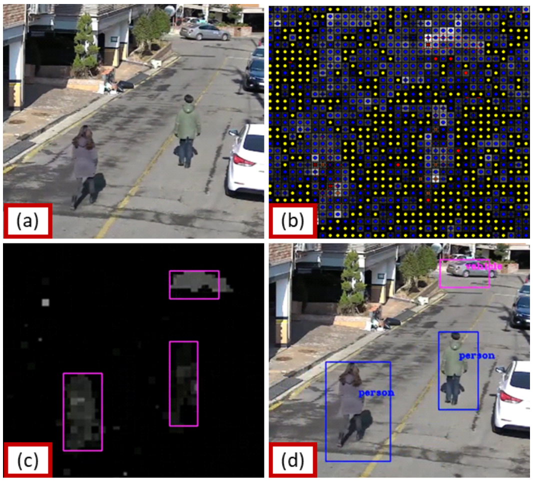

Fig. 5. Real-time Detection: a) CCTV footage b) Normalized values c) Final classification d) Aggregated result

Ⅴ. Conclusions

This paper presents an innovative approach to object motion detection using the syntax elements extracted from the H.264 bitstream. The main focus of this paper is analyzing high-resolution transport and surveillance CCTV videos. As described in the paper, we can observe that this results in some challenges like the change in illumination being detected as object motion. In order to eliminate this, we have included an error elimination module. Our H.264 bitstream-based detector is an advancement over previous related research in terms of the accuracy and the number of frames detected within a timespan. Furthermore, the proposed work considers speed and computational cost as significant factors; this enables us to use it for live-streamed video with less resource consumption.

The proposed technique shows excellent performance because it analyzes the screen and maintains the quality of the frames. We have shown that the analyzing speed is significantly lower. This is because the analysis is done on compressed information instead of de-compressed information. This causes a reduction in the amount of time required for analysis. Also, since downsampling is not done, the ROI that has been extracted does not undergo any loss of pixel information; the original quality is maintained.

Future work: Currently, the detection is being done on the international standard for video compression, that is, H.264/AVC. We are developing it to be adapted to future international standards, including H.265 and H.266.

References

[1] P. Bischoff, *Surveillance camera statistics: Which cities have the most CCTV cameras?*(2021), Retrieved Dec. 14, 2021, from https://www.comparitech.com/.

[2] G. W. Leeson, "The growth, ageing and urbanisation of our world," *J. Population Ageing*, vol. 11, no. 2, pp. 107-115, Springer, 2018.

[3] Y. Pang, L. Ye, X. Li, and J. Pan, "Incremental learning with saliency map for moving object detection," *IEEE Trans. Cir. and Syst. for Video Technol.*, vol. 28, no. 3, pp. 640-651, 2016.

[4] S. -H. Yoo, "A study on counting number of passengers by moving object detection," *J. Internet Comput. and Serv.*, vol. 21, no. 2, pp. 9-18, 2020.

[5] S. H. Shaikh, K. Saeed, and N. Chaki, *Moving Object Detection using Background Subtraction*, Springer, 2014.

[6] R. S. Rakibe and B. D. Patil, "Background subtraction algorithm based human motion detection," *Int. J. Sci. and Res. Publications*, vol. 3, no. 5, pp. 2250-3153, 2013.

[7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Cir. and Syst. for Video Technol.*, vol. 13, no. 7, pp. 560-576, 2003.

[8] W. You, M. H. Sabirin, and M. Kim, "Moving object tracking in H. 264/AVC bitstream," *Int. Wkshp. Multimedia Content Anal. and Mining*, Springer, 2007.

[9] W. You, M. H. Sabirin, and M. Kim, "Real-time detection and tracking of multiple objects with partial decoding in H.264/AVC bitstream domain," *Real-Time Image and Video Process.*, vol. 7244, 2009.

[10] C. Kas, M. Brulin, H. Nicolar, and C. Maillet, "Compressed domain aided analysis of traffic surveillance videos," *2009 Third ACM/IEEE ICDSC*, 2009.

[11] C. Poppe, S. De Bruyne, T. PAridaens, P. Lambert, and R. Van de Walle, "Moving object detection in the H. 264/AVC compressed domain for video surveillance applications," *J. Visual Commun. and Image Representation*, vol. 20, no. 6, 2009.

[12] S. K. Kapotas and A. N. Skodras, "Moving object detection in the H.264 compressed domain," *2010 IEEE Int. Conf. Imaging Syst. and Techniques*, 2010.

[13] S. De Bruyne, C. Poppe, S. Verstockt, P. Lambert, and R. Van de Walle, "Estimating motion reliability to improve moving object detection in the H.263/AVC domain," *2009 IEEE Int. Conf. Multimedia and Expo*, 2009.

[14] R. C. Moura and E. M. Hemerly, "A spatiotemporal motion-vector filter for object tracking on compressed video," *2010 7th IEEE Int. Conf. Advanced Video and Sign. Based Surveillance*, 2010.

[15] C. Kas and H. Nicolas, “An approach to trajectory estimation of moving objects in the H.264 compressed domain,” *Pacific-Rim Symp. Image and Video Technol.*, Springer, 2009.

[16] H. Sabirin and M. Kim, “Moving object detection and tracking using a spatio-temporal graph in H.264/AVC bitstreams for video surveillance,” *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 657-668, 2012.

[17] M. Brulin, C. Kaes, H. Nicolar, and C. Maillet, “Compressed domain aided analysis of traffic surveillance videos,” *ICDSC-IEEE*, 2009.

[18] Y. K. Kim, Y. G. Jeon, and S. H. Shin, “Real-time motion detection in H.264 compressed domain for surveillance application,” *J. Physics: Conf. Series*, vol. 1780, no. 3, 2021.

[19] N. Özbek and T. Tunali, “A survey on the H. 264/AVC standard,” *Turkish J. Electr. Eng. & Comput. Sci.*, vol. 13, no. 3, 2005.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2014.

[21] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in NIPS*, vol. 28, pp. 91-99, 2015.

[22] R. Girshick, “Fast r-cnn,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015.

[23] M. Everingham, L. Van Gool, C. KI Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” vol. 88, no. 2, pp. 303-338, Springer, 2010.

김 동 기 (Dong-Key Kim)

1988년 2월 : 홍익대학교 토목공학과 졸업

1996년 12월 : F.I.U (Florida International University) 토목공학과 석사

2019년 3월~현재 : 서울시립대학교 교통공학과 박사과정

<관심분야> 교통공학, 교통안전, 딥러닝, AI

[ORCID:0000-0003-2485-5209]

로즈마리 코이카라 (Rosemary Koikara)

2013년 6월 : Assam Don Bosco University, Computer Science and Engineering. 졸업

2015년 3월 : Christ University Faculty of Engineering, School of Computer Science and Engineering. 석사

2021년 2월 : 경북대학교 컴퓨터학부 박사

2020년 12월~현재 : 주식회사 핀텔, 선임 연구원

<관심분야> 정보 보안, 딥러닝, 컴퓨터 비전

[ORCID:0000-0001-7506-8657]

김 선 정 (Sun-Jung Kim)

1998년 2월 : 목원대학과 도시공학과 졸업

2001년 2월 : 목원대학과 도시공학과 석사

2008년 3월~현재 : 서울시립대학교 교통공학과 박사과정

<관심분야> 교통공학, 교통안전

[ORCID:0000-00001-9540-0274]

신 소 명 (So-Myoung Shin)

2017년 2월 : 경기대학교 도시교통공학과 졸업

2019년 2월 : 서울시립대학교 교통공학과 석사

2019년 3월~현재 : 서울시립대학교 교통공학과 박사과정

<관심분야> 교통공학, 교통안전

[ORCID:0000-0002-0888-2349]

엄 제 원 (Jewon Eom)
2018년 2월 : 한양대학교 융합전자공학부 졸업
2020년 2월 : 한양대학교 전자컴퓨터통신공학과 석사
2019년 3월~현재 : 주식회사 핀텔, 연구원
<관심분야> 딥러닝, 컴퓨터 비전, 영상 처리
[ORCID:0000-0002-3370-3293]

김 민 중 (Min-Joong Kim)
2015년 8월 : THAMMASAT University, Industrial Engineering졸업
2019년 2월 : 울산과학기술원 인간공학과 석사
2019년 8월~현재 : 주식회사 핀텔, 연구원
<관심분야> VR, UX, AI, 컴퓨터 비전

홍 주 희 (Joo-Hee Hong)
2002년 2월 : 한양대학교 교통공학과 졸업
2008년 8월 : 시립대학교 교통관리학 석사
2013년 3월~현재 : 시립대학교 교통공학과 박사과정
<관심분야> 교통안전, 빅데이터, 영상솔루션
[ORCID:0000-0002-9549-8141]