Research Article

Embedded System-Based Image Processing Methods for Detection of Forensic Events in CCTV Videos

Irfan Kilic, Emre Orhan, Enes Kurtulan, Muhammed Enes Ozel, Gozde Arslan and Orhan Yaman

Abstract— **Nowadays, there is no place where security cameras (CCTV) are not used. Security cameras play a huge role in solving criminal cases. However, a lot of time is spent examining these camera recordings. This situation causes the incidents to not be resolved and causes delays. This study, it is aimed to use machine learning to increase the size of security camera recordings with efficient algorithms that can work on devices with low processing power such as embedded systems. Within the scope of the study, an experimental environment was created by installing a security camera system. Fast and effective video reduction algorithms have been developed on videos collected in different scenarios. New approaches called hopscotch and lens algorithms have been presented for video reduction. These approaches are aimed to obtain rapid results by applying them to security camera videos. It is thought that the developed video reduction approaches will lead to the creation of applicable prototypes on embedded cards such as Raspberry Pi. PSNR (Peak Signal to Noise Ratio) metric was used to compare the images. Real-time results were obtained with our approaches applied to images.**

Index Terms—**Embedded system, Forensic investigation, Video reduction, Image processing, CCTV, PSNR.**

İrfan Kılıç, is with Information Technologies Department, Firat University, Elazığ, Turkey,(e-mail: irfankilic@firat.edu.tr).
https://orcid.org/0000-0001-5079-2825

Emre Orhan, is with Department of Digital Forensic Engineering, Firat Universit, Elazig, Turkey, (e-mail: emre2344orhan@gmail.com).
https://orcid.org/0009-0000-2214-2518

Enes Kurtulan, is with Department of Digital Forensic Engineering, Firat Universit, Elazig, Turkey, (e-mail: 215509032@firat.edu.tr).
https://orcid.org/0009-0003-1431-7834

Muhammed Enes Özel, is with Department of Digital Forensic Engineering, Firat Universit, Elazig, Turkey, (e-mail: muhammedenes.ozel1@gmail.com).
https://orcid.org/0009-0001-0556-7757

Gözde Arslan, is with Department of Digital Forensic Engineering, Firat Universit, Elazig, Turkey, (e-mail: gozde.arslann0@gmail.com).
https://orcid.org/0009-005-8608-400X

Orhan Yaman, is with Department of Digital Forensic Engineering, Firat Universit, Elazig, Turkey, (e-mail: muhammedenes.ozel1@gmail.com).
https://orcid.org/0000-0001-9623-2284
Manuscript received Jun 15, 2024; accepted Dec 30, 2024 .
DOI: 10.17694/bajece.1501656

I. Introduction

THE DEVELOPMENT of technology and the decrease in production costs continue to positively affect our lives. These developments and changes affect every aspect of our lives, including security. A large workload is needed to monitor the current or historical records of the numerous cameras owned by security forces such as the police and gendarmerie. In addition to security forces, universities, airports, terminals, hospitals, etc. Camera systems are needed to ensure security in civilian living spaces. Hundreds/thousands of cameras are needed to ensure effective security, especially in areas with high human density. This situation reveals how important camera systems are for both security forces and civilian life safety. Table I shows the number of cameras per 1000 people in some countries and cities.

TABLE I
NUMBER OF CAMERAS PER 1000 PEOPLE BY COUNTRY AND CITY [1]

Country	City	Number of CCTV	Population	Number Of Cameras Per 1000 People
China	Chongqing	2.579.890	15.354.067	168.03
China	Shenzen	1.929.600	12.128.721	159.09
England	Londra	627.707	9.176.530	68.40
UAE	Abu Dhabi	20.000	1.452.057	13.77
Georgia(USA)	Chicago	35.000	2.679.044	13.06
Russia	Moskova	146.000	12.476.171	11.70
Germany	Berlin	39.765	3.556.792	11.18
India	Yeni Delhi	179.000	18.600.000	9.62
Türkiye	İstanbul	109.000	15.190.336	7.18

Table I summarizes the number of cameras, especially in crowded cities. It is seen that the number of cameras per 1000 people is significant. Especially in China, there is almost 1 camera for every 6 people [1]. These figures once again show the importance of camera systems in our lives. In addition to traditional camera systems, smart security solutions are being developed. Especially in security cameras, many scenarios such as smart monitoring, motion detection, suspicious package detection, and alarm generation are carried out by recording devices and cameras.

Camera manufacturers are constantly doing R&D to increase recording times, lower energy consumption, and reduce costs for competition. It also develops artificial intelligence-based techniques for easy use and analysis of camera systems. In recent years, scenarios such as motion detection, suspicious package detection, and alarm generation have become classic features in almost all camera systems. Camera manufacturers are developing smart analysis approaches using artificial intelligence methods. The main motivation when developing these methods is to solve users' problems.

A. Literature Background

Studies on reducing video images have been carried out from two different perspectives. These; 1. Video resolution reduction and 2. Video frame skipping.

If we take a look at the studies done in recent years on video resolution reduction; In the study conducted by Y. Huang et al., it was aimed to improve the reduction and increase of resolution as a common task by rescaling video images. [2]. A method to measure user experiences when reducing the resolution of EEG-based VR video images was proposed by Z. li et al. [3]. In the study by D. Hazra et al., a neural network was proposed to obtain a high-resolution sampling of low-resolution CCTV images using Generative Adversarial Networks [4].

If we examine the recent studies on skipping video frames; In the study conducted by M. Adnan Arefeen and his colleagues, the FrameHopper algorithm was proposed in systems with edge-cloud collaboration. This study presented an oracle solution by applying FrameHopper on filtered images using reinforcement learning. [5]. A study by T. Wang et al. proposed a Frame Skipping mechanism that actively manages frames within the decoder queue, effectively reducing queuing latency. [6]. In a study by J. Cempron et al., frame skipping was modeled empirically and a model was proposed to simulate frame skipping with the help of high-resolution videos. In this way, the effect of frame skipping on object tracking was measured [7].

B. Motivation of Study

New datasets were obtained and used from the images we obtained within the scope of the study. Lightweight methods have been developed and tested with data sets so that they can work on development boards such as Raspberry Pi. The main motivation of the study is to develop video reduction algorithms for fast and high-accuracy monitoring of CCTV camera recordings in IoT-based and embedded systems. Considering the cost of such systems, this study offers a lower-cost solution.

With study;

- Making both time and human workforce more useful by making the time spent on problems more efficient,
- Preventing the problem of problems being clarified late, as important details may be overlooked in cases that are more difficult to solve,
- Increasing the rate of solving crimes by reducing the time spent in investigating incidents,
- Recommending optimized algorithms that can run on devices with low processing power to keep costs low,
- Thanks to video reduction, large-volume videos are shortened and less space is kept in memory.
- It is aimed to create a data set from the images to be collected and to share this data set.

II. Materials and Methods

Techniques that vary depending on the type of video vary depending on the type of camera. Features such as whether the camera is moving or not may also cause changes in the techniques used. Within the scope of the study, a feasibility study was conducted on existing data sets and new approaches were proposed. The approaches recommended at the end of the feasibility study are listed below;

- Hopscotch Algorithm
- Lens Algorithm
- Hybrid Method (Hopscotch + Lens Algorithm)

In this study, experiments were carried out on our data set for the "Hopscotch Algorithm", "Lens Algorithm" and our Hybrid method.

A. Hopscotch Algorithm

Each second of a video usually consists of 24 frames. Using 24 frames per second is the minimum number of frames for the video to appear fluid to the human eye. Having 24 frames per second means that there are thousands of frames in an hours-long video. Therefore, the video analysis algorithm will need a lot of time and a lot of processing power to analyze all of these frames one by one. With the Hopscotch algorithm, instead of examining 24 frames of each second, you can skip frames and see 3,6,12, etc. in a second. It aims to examine the square. 3,6,12 etc. per second. Although the square is not a fluid image for the human eye, it may be sufficient for computers to analyze the image. The hop-hop algorithm will examine the frames in the video by skipping them and is expected to increase the performance of the analysis process by 2 to 8 times. With this algorithm, it is shown how much difference the number of frames in a 1-hour video makes when examined with the hop-by-hop algorithm. This difference is thought to contribute greatly to both faster analysis and optimized operation on devices with lower processing power. The general flow diagram of the Hopscotch algorithm is given in Fig. 1. The pseudocode of the Hopscotch algorithm is given in Table II.

The number of frames obtained and processed when using the Hopscotch algorithm, whose pseudo-code is shown in Table II, is calculated in Table III.

Table III lists the total number of frames of an original video one hour long, 24 fps (frames per second), and the total frames of videos edited with the Hopscotch algorithm. When these calculations are examined, if the video is updated to 6 fps with the Hopscotch algorithm, the total number of frames decreases 4 times compared to the original video. This situation positively affects the speed of the method.

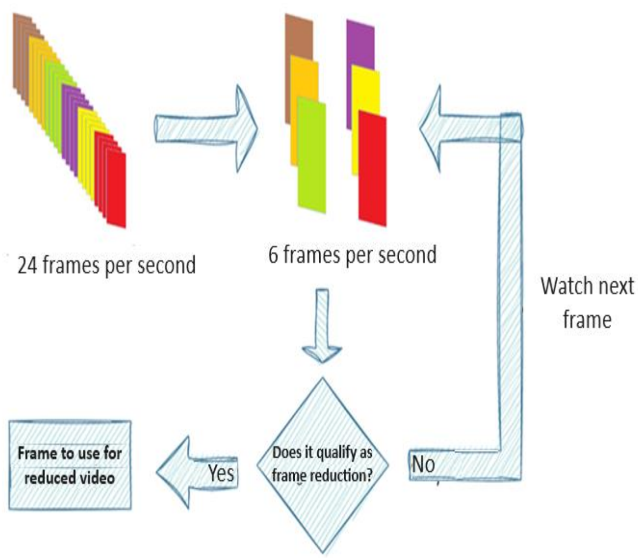


Fig. 1. Hopscoth algorithm flow diagram

TABLE II
HOPSCOTCH ALGORITHM PSEUDOCODE

Step No	Steps
00	Get 24fps video
01	Take an image frame by skipping 1 in 2
02	If the received frame is suitable, go to step 03, otherwise go to step 01
03	Run color-based HSV algorithm on imported frame
04	If the video is not finished, go to step 01
05	Finish

TABLE III
NUMBER OF FRAMES IN THE ORIGINAL VERSION OF THE VIDEO AND THE NUMBER OF FRAMES AFTER EDITING BY THE HOPSCOTCH ALGORITHM

Video Duration	Frames Per Second	Total Frames
1 Hour	24	86.400
1 Hour	20	72.000
1 Hour	15	54.000
1 Hour	10	36.000
1 Hour	6	21.600
1 Hour	4	14.400

B. *Lens Algorithm*

The purpose of the lens algorithm, as in previous algorithms, is to ensure that the video summarization algorithm runs faster and more efficiently on devices with lower processing power. This algorithm performs video summarization by reducing the resolution of the video. The reason for reducing the quality of the video is that the frames in the videos are examined faster by allowing the algorithm to process fewer pixels. The time information of the keyframes detected in the low-resolution version of the video will be recorded and marked in the original version of the video and presented to the user. The resolution values of security cameras can go up to 5MP, but generally, 2MP resolution is preferred in security cameras. A 2MP security camera has a resolution of 1920 * 1080 pixels. With the lens algorithm, this value is reduced to 1280 * 720 or 320 * 240 pixels, allowing processing on fewer pixels. In this way, the algorithm is expected to produce faster results by consuming fewer resources. The general flow diagram of the lens algorithm is given in Fig. 2. The pseudocode of the lens algorithm is given in Table IV.

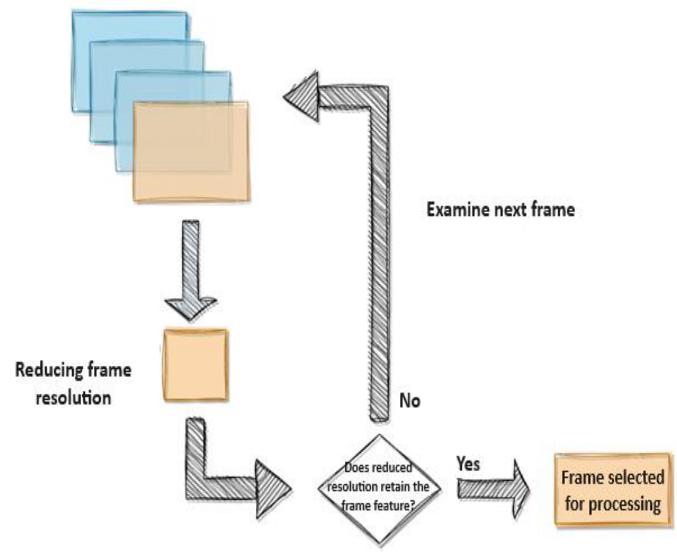


Fig. 1. Lens algortihm flow diagram

TABLE IV
LENS ALGORITHM PSEUDOCODE

Step No	Steps
00	Get high-resolution video frame by video
01	Reduce frame resolution by half (row, column)
02	If the received frame is suitable, go to step 03, otherwise go to step 01
03	Run color-based HSV algorithm on imported frame
04	If the video is not finished, go to step 01
05	Finish

TABLE V
HYBRID METHOD PSEUDOCODE

Step No	Steps
00	Get video frame by video
01	Take an image frame by skipping 1 in 2
02	Reduce frame resolution by half (row, column)
03	If the received frame is suitable, go to step 04, otherwise go to step 01
04	Run color-based HSV algorithm on imported frame
05	If the video is not finished, go to step 01
06	Finish

In this study, two different methods are proposed for rapid analysis of sample security camera videos. The proposed methods were applied to the security camera videos to be collected and the performance results were compared.

C. Hybrid Method

In the study, a hybrid method was proposed by combining the Hopscotch and Lens algorithms and making both skipping and resolution changes on the videos. The pseudocode of this method is given in Table V.

It is thought that our Hybrid method will be suitable for working particularly well on IoT and embedded system-based devices such as Raspberry Pi, Odroid, and UP2 [8], [9], [10].

III. Experimental Results and Discussion

Experiments were conducted for our algorithms using 780 frames of CCTV video with a resolution of 1920x1080 (2 MP). Intel i7 12th generation 2.3 GHz processor and 32 GB memory were used for the experimental environment. In the experiments, object detection and object tracking were performed on videos. Fig. 3 shows the bounding box and location tracking graph of the object detected in the video in different frames. An HSV color-based method was used for object detection and tracking [11].

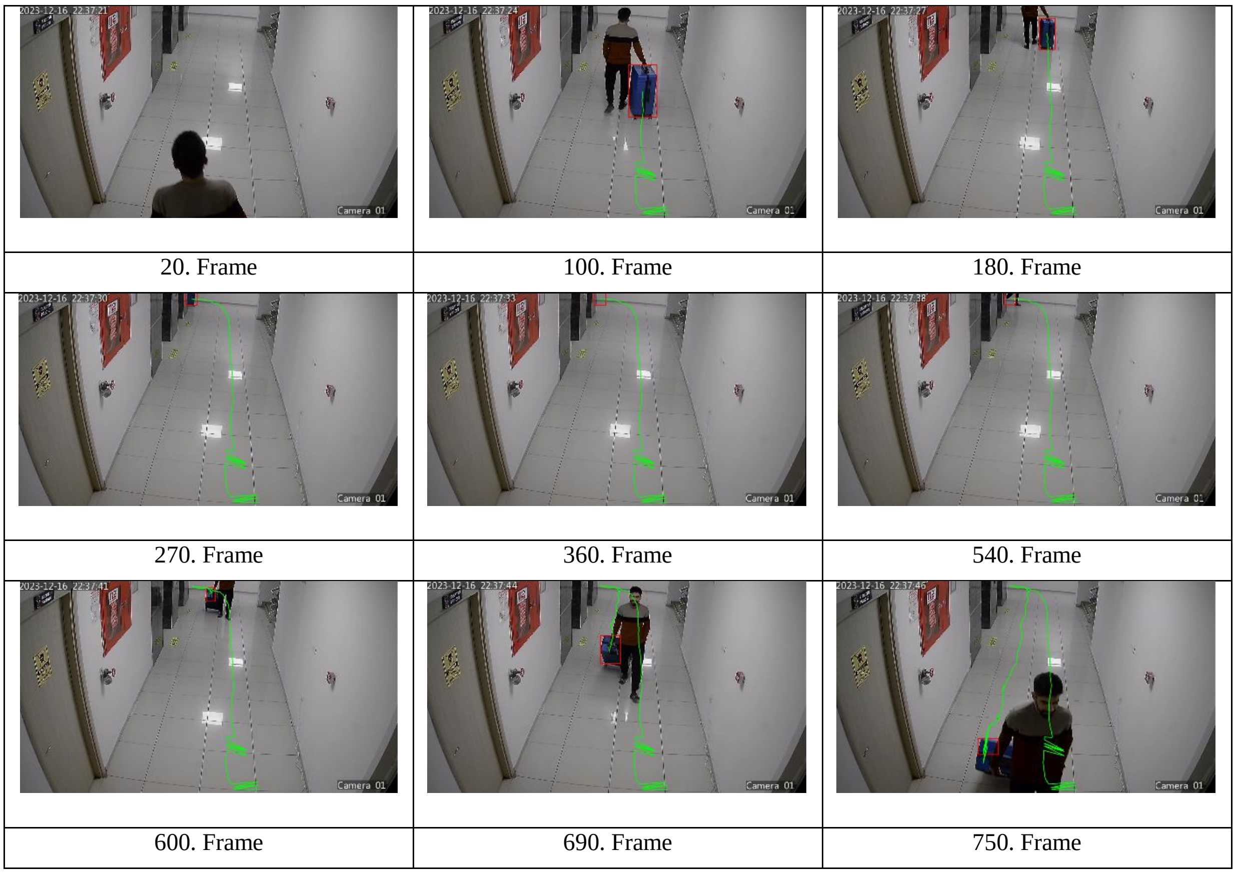


Fig. 3. Object detection and location tracking graph on video

When Fig. 3 is examined, it is seen that the image is in a dim (slightly dark) environment. It can be seen that the object is detected and tracked correctly in different frames. Fig. 4.

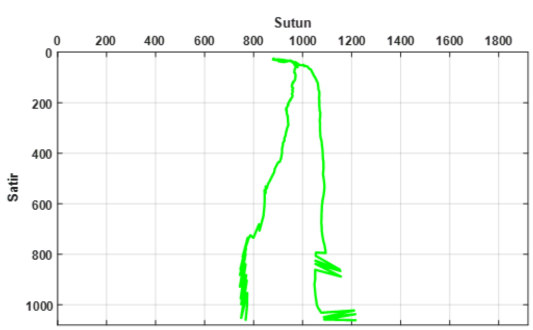
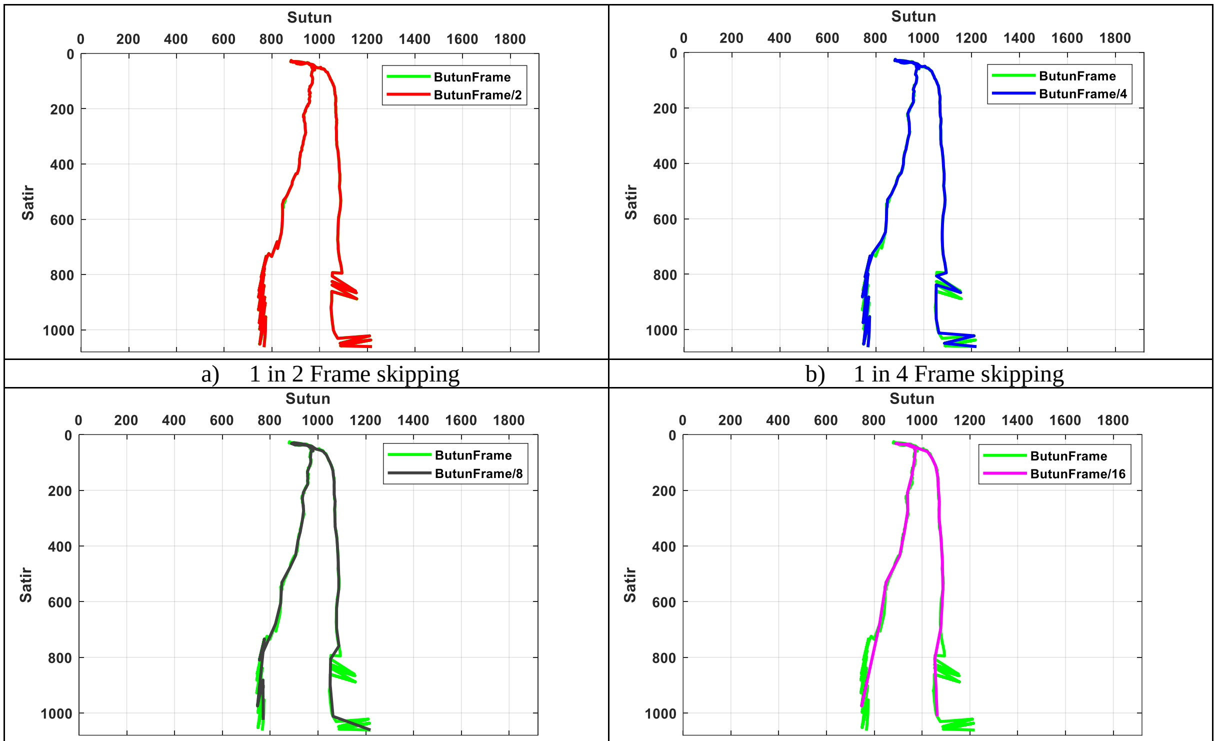


Fig. 4.. Location tracking graph of the detected object

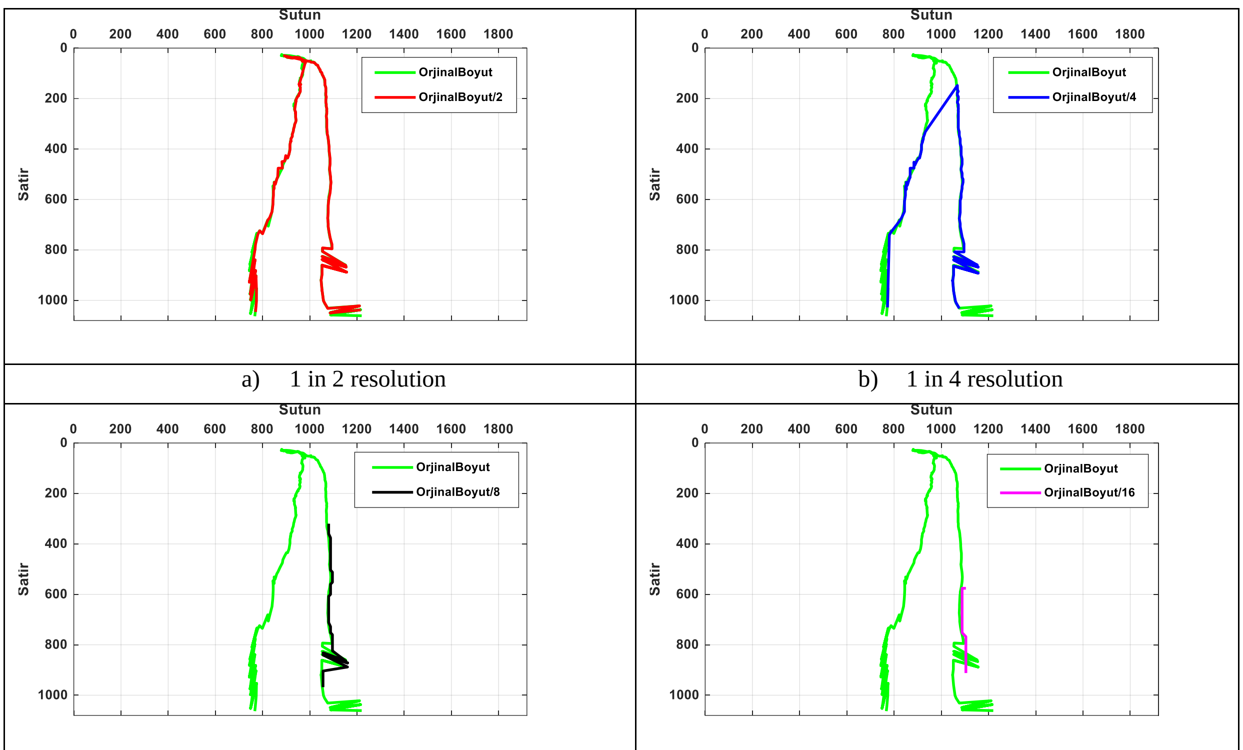
The object was detected and positioned in 494 frames on the 780-frame image. All 780 frames have been processed. The total processing time for 780 frames is 51.79 seconds. Fig. 5 shows the object position graphs detected using 1 in 2, 1 in 4, 1 in 8, and 1 in 16 Frames, according to the actual video, using the Hopscotch algorithm. When the graphics in Fig. 3 are examined, it is seen that the results in the videos with 1 in 2 and 1 in 4 skipping are very close to the actual image. Fig. 6 shows the location tracking graphs of the object detected using the Lens algorithm with lower resolutions (1 in 2, 1 in 4, 1 in 8, 1 in 16) compared to the original video.

When the graphics in Fig. 6 are examined, it is seen that the results for a resolution of 1 in 2 are very close to the actual video. In Table VI, the time spent and PSNR (Peak Signal to Noise Ratio) scores for the Hopscotch algorithm, Lens algorithm, and our Hybrid approach are given for our example scenario. PSNR is generally used to compare areas such as signal, graph, and bounding box.



a) 1 in 2 Frame skipping
b) 1 in 4 Frame skipping
c) 1 in 8 Frame skipping
d) 1 in 16 Frame skipping

Fig. 5. Hopscotch algorithm results according to different frame skips



a) 1 in 2 resolution
b) 1 in 4 resolution
c) 1 in 8 resolution
d) 1 in 16 resolution

Fig. 6.. Lens algorithm results according to different resolutions

When the graphics in Fig. 6 are examined, it is seen that the results for a resolution of 1 in 2 are very close to the actual video. In Table VI, the time spent and PSNR (Peak Signal to Noise Ratio) scores for the Hopscotch algorithm, Lens algorithm, and our Hybrid approach are given for our example scenario. PSNR is generally used to compare areas such as signal, graph, and bounding box.

PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of disruptive noise that affects the fidelity of its representation.

Most commonly used to measure the reconstruction quality of lossy compression codecs (for example, for image compression) [12], [13], [14].

PSNR works roughly as follows:

Original Signal: A clean, noise-free video. This is the original signal.

Noisy Signal: Let's assume that the image is corrupted by some noise such as compression artifacts, pixelation, or various reasons as in our methods. This is a noisy signal.

PSNR Calculation: PSNR is calculated by comparing the original and noisy signals. It takes the difference (error) between two signals, takes its Frame, averages it over the entire signal, and then divides it by the variance of the original signal. Finally, the result is converted to decibels (dB). PSNR calculation is given in equation (1).

$$SNR = 10.log_{10}\frac{MAX}{MSE^2} \quad (1)$$

Here, the MAX value refers to the maximum pixel value. For an 8-bit image, this value is 255. The MSE (Mean Squared Error) value is the average of the Frame differences between the corresponding pixels of the original and reconstructed images. MSE calculation is given in equation (2).

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{I_i} - \mathrm{K_i})^2 \quad (2)$$

In equation (2), the N value is the total number of pixels, I_i value is i in the original image. pixel value intensity, K_i in the modified image i. The pixel value gives the density.

A higher PSNR value indicates a better-quality reconstruction. A general interpretation of PSNR values is given below:

- PSNR > 40 dB: Excellent quality, almost indistinguishable from the original signal.
- PSNR 30-40 dB: Good quality, noticeable but slight distortion.
- PSNR 20-30 dB: Acceptable quality, distortion becomes visible.
- PSNR < 20 dB: Low quality, significant distortion.

TABLE VI
HOPSCOTCH, LENS, AND HYBRID METHOD PERFORMANCE RESULTS ACCORDING TO THE EXAMPLE SCENARIO

	Frame size	Total number of frames	Frame ratio	Size ratio	Number of frames processed	Object Number of frames detected	Time (sec) for the number of frames processed	PSNR Score (dB)
Example scenario	1080x1920	780	1/1	1/1	780	494	51.79	-
Proposed HopScotch (Frame hopping) algorithm	1080x1920	780	1/2	1/1	390	247	**26.33**	**30.08**
	1080x1920	780	1/4	1/1	195	124	13.36	27.07
	1080x1920	780	1/8	1/1	97	60	6.75	25.02
	1080x1920	780	1/16	1/1	48	30	3.37	23.26
Proposed Lens (Resolution reduction) Algorithm	540x960	780	1/1	1/2	780	385	**20.59**	28.09
	270x480	780	1/1	1/4	780	218	17.52	23.54
	135x240	780	1/1	1/8	780	91	14.57	22.46
	68x120	780	1/1	1/16	780	43	12.61	21.94
Proposed Hybrid Method	540x960	780	1/2	1/2	390	193	**12.34**	**28.13**
	540x960	780	1/4	1/2	195	93	**6.27**	25.75

When Table VI is examined, it is seen that as the frame skipping rate decreases for our sample scenario of the Hopscotch algorithm (object detection and tracking), the time taken for the entire image decreases by half (26.33, 13.36, 6.75, 3.37) for each step. It is seen that ½ skip rate gives better results (30.08) compared to other methods, especially in terms of PSNR value. It can be seen that it takes less time (20.59) due to the lower number of pixels processed for the ½ ratio in the lens algorithm. It has been observed that our Hybrid method, in which both methods are used together, gives the best results in terms of time.

IV. Conclusion

Considering the experimental results, it is seen that the three methods we recommend give results (29.62 Frames/second, 37.88 Frames/second, 63.2 Frames/second) above real-time values (24 Frames/second). It is understood from this that the desired result was achieved even with only ½ Frame ratio. Considering the performance of a desktop computer in today's conditions, it will be seen that the Hybrid method we propose can easily work in real-time in an IoT-based or embedded system. It is aimed to obtain results by using the Grid method in future studies. In addition, results will be obtained, and deep learning in addition to machine learning for object detection and tracking used in our scenario will expand the study.

Acknowledgment

This study was supported by TÜBİTAK 2209A project number 1919B012218080.

REFERENCES

[1] Paul Bischoff, "Surveillance camera statistics: which are the most surveilled cities?," 2023. Available: https://www.comparitech.com/vpn-privacy/the-worlds-most-surveilled-cities/. [Accessed: Mar. 30, 2024]

[2] Y. C. Huang, Y. H. Chen, C. Y. Lu, H. P. Wang, W. H. Peng, and C. C. Huang, "Video Rescaling Networks with Joint Optimization Strategies for Downscaling and Upscaling," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021. doi: 10.1109/CVPR46437.2021.00353

[3] Z. Li, B. Geng, X. Tao, Y. Duan, T. Li, and J. Huang, "EEG-Based VR Video Quality Measurement for Resolution Reduction," in *2023 IEEE International Conference on Communications Workshops: Sustainable Communications for Renaissance, ICC Workshops 2023*, 2023. doi: 10.1109/ICCWorkshops57953.2023.10283518

[4] D. Hazra and Y. C. Byun, "Upsampling real-time, low-resolution CCTV videos using generative adversarial networks," *Electron.*, 2020, doi: 10.3390/electronics9081312

[5] M. Adnan Arefeen, S. Tabassum Nimi, and M. Yusuf Sarwar Uddin, "FrameHopper: Selective Processing of Video Frames in Detection-driven Real-Time Video Analytics," in *Proceedings - 18th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2022*, 2022. doi: 10.1109/DCOSS54816.2022.00033

[6] T. Wang, Z. Meng, M. Xu, R. Han, and H. Liu, "Enabling high frame-rate UHD real-time communication with frame-skipping," in *HotEdgeVideo 2021 - Proceedings of the 2021 3rd ACM Workshop on Hot Topics in Video Analytics and Intelligent Edges*, 2021. doi: 10.1145/3477083.3481582

[7] J. P. C. Cempron and J. P. Ilao, "Network Video Frame-Skip Modeling and Simulation," in *2021 27th International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2021*, 2021. doi: 10.1109/M2VIP49856.2021.9664996

[8] M. Ajmal, M. H. Ashraf, M. Shakir, Y. Abbas, and F. A. Shah, "Video summarization: Techniques and classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7594 LNCS, pp. 1–13, 2012, doi: 10.1007/978-3-642-33564-8_1

[9] "Rasperry Pi," 2023. Available: https://www.raspberrypi.org/. [Accessed: Dec. 15, 2023]

[10] "Odroid," 2023. Available: https://www.hardkernel.com/. [Accessed: Dec. 15, 2023]

[11] "UP2 Board," 2023. Available: https://up-board.org/about-up/. [Accessed: Dec. 15, 2023]

[12] H. C. Kang, H. N. Han, H. C. Bae, M. G. Kim, J. Y. Son, and Y. K. Kim, "Hsv color-space-based automated object localization for robot grasping without prior knowledge," *Appl. Sci.*, 2021, doi: 10.3390/app11167593

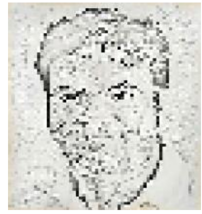
[13] "PSNR Mathworks," 2023. Available: https://www.mathworks.com/help/images/ref/psnr.html. [Accessed: Dec. 20, 2023]

[14] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?," in *2012 4th International Workshop on Quality of Multimedia Experience, QoMEX 2012*, 2012. doi: 10.1109/QoMEX.2012.6263880

[15] Al Boivik, *HANDBOOK OF IMAGE AND VIDEO PROCESSING*, First Edit. Canada: Academic Press, 2000.

BIOGRAPHIES

İrfan Kılıç After graduating from Fırat University Computer Engineering in 2004, I am working as a System and Network administrator in the Computer Science Department of the University. I completed my master's degree at İnönü University Computer Engineering on "Analysis of DNA Sequencing with Graph Techniques" in 2016. I have a PhD degree in Computer Engineering in 2023.I have studied in the field of deep learning, and machine learning. During my doctorate, I worked as a researcher in the research and development project "DEĞİRMEN".

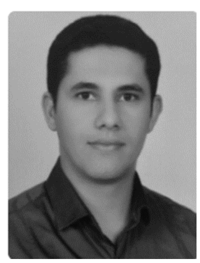


Emre Orhan He is an undergraduate student in Digital Forensic Engineering.

Enes Kurtulan is an undergraduate student in Digital Forensic Engineering.

Muhammed Enes Özel He is an undergraduate student in Digital Forensic Engineering.

Gözde Arslan, She is an undergraduate student in Digital Forensic Engineering.



Orhan Yaman Orhan Yaman (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer engineering from Firat University, Elazig, Turkey, in 2012, 2014, and 2018, respectively., He is currently an Assistant Professor with the Department of Computer Technology, Firat University. His research interests include image processing, fault diagnosis, railway inspection systems, and fuzzy systems.