Implementation of Lossy Compression Method for Storage Saving on Fog Computing

Rachmat Indra Permadi[1], Dany Primanita Kartikasari[2], Fariz Andri Bakhtiar[3]

Fakultas Ilmu Komputer, Universitas Brawijaya

Malang, Indonesia

[1]ripermadi@student.ub.ac.id, [2]dany.jalin@ub.ac.id, [3]fariz@ub.ac.id

Abstract— **Video surveillance is a technology that uses a camera as an image receiver and a monitor or Television (TV) as an image producer to covers certain areas. In the application, video surveillance was installed for monitoring the area using a camera. The camera will record the situation that occurs periodically, then send it to the virtual storage room via internet network and then display it on a monitor/TV. However, video surveillance has weaknesses in significant data/video output. Based on that problem, the concept of fog computing is applied, using a Cloud Circuit Television (CCTV) camera to monitoring for 24 hours. However, these experiments have a large data output until it reaches 4 GB storage. To reduce the data storage, this study uses a lossy compression method. The concept of fog computing is applied to compress video using the lossy compression method. The lossy compression codec used in this study is; H.263, H.264, and MJPEG (Motion Joint Expert Group) format. The result of a video with the H.263 codec has an excellent efficiency value than other codecs, which is 66%.**

Index Terms— **Video Surveillance, Fog Computing, Lossy Compression**

I. INTRODUCTION

Cloud Computing is a service that combines computer technology with virtual internet-based development [1]. Cloud computing provides services to users for access data or applications remotely and store data without using traditional hardware. However, in its application with IoT, there are weaknesses, namely bandwidth limitations to process data transfers [2]. Fog Computing is a further development of Cloud Computing is a concept in the realm of Distributed Computing, which allows services from the cloud to be pulled to the network edge that functions for distance and latency from end-users to the cloud [3]. By combining these two technologies, all computing can be performed at the network edge (fog) for the distance and latency that must be performed on the cloud. However, there are several weaknesses in fog computing, one of which lies in the storage space. The storage space contained in fog computing devices is very limited, but the limited storage capacity is used to store large videos.

Based on the experiments that have been carried out previously, the concept of fog computing is applied, which is connected to a CCTV camera installed in a room for 24 hours. The video recorded on the CCTV camera will go to the fog point to check using a motion detection algorithm, checking is carried out to determine whether the object is in the room or not. From these experiments, the results obtained are quite large recordings of 4 GB. The large size of the video used for computing the checking process can fill the storage space of the fog node.

To overcome this problem, a lossy compression method is applied. The lossy compression method can save storage space because the burden of large video sizes is reduced by reducing the size of the video. By applying the lossy compression method to the fog node, this study makes storage efficiency in the fog node the primary goal. This research uses a lossy compression codec H.263, H.264, and MJPEG. The result of a video with the lossy compression method gets efficient results after the system performs the compression process on the video before the data is saved to the cloud.

II. LITERATURE REVIEW

A. Cloud Computing

Cloud computing is a parallel and distributed computing system of interconnected computers with a single server to provide users based on Service Level Agreement (SLA) [1]. Cloud computing provides to users to be able for access the applications, data and do storage remotely without use a traditional hardware using third-party servers or known as cloud storage. Cloud computing has two main architecture is Front-End and Back-End. The architecture is connected to each other via the internet. The Front-End can be a sensor, client, or any application that uses a cloud service. Back-End is the cloud system which consists of computer, servers and data storage [4].
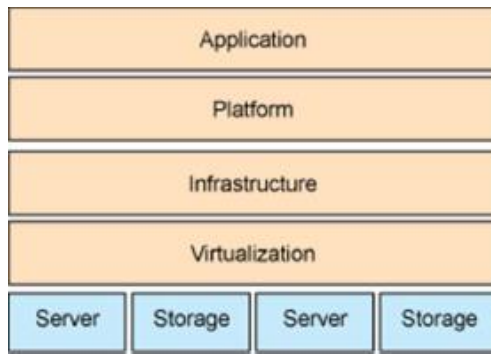
Fig. 1. Layers of cloud computing architecture

There are three types of services provided by cloud computing system is a Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). SaaS is a cloud computing service with applications provided by providers via internet. PaaS provides services in the form of cloud users can develop their own applications. IaaS provides all cloud computing service from resources, storage, networking and other to users.

B. Fog Computing

Fog computing is a computing services system that has set of connected devices located at the edge of network with support from cloud service providers to provide convenience in user storage and communications [5]. Fog computing and cloud computing have difference in computing process, where fog computing do at the edge of network and cloud computing do at the server center.
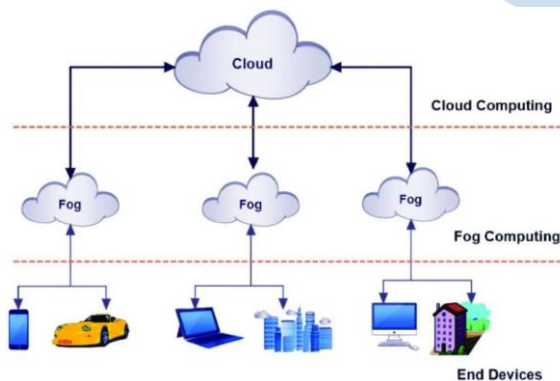

Fig. 2. Fog computing architecture.

The concept of fog computing is to provide network services near nodes. Fog has the responsibility to provide compute, storage and networking services between end nodes in IoT and cloud environments. Figure 2 shows some architectures in fog computing that provide services in an IoT environment using networks system such as Wireless Sensor Network (WSN), Virtual Sensor Network (VSN), VANET, and Private Area Network (PAN) [6].

C. Multimedia Network

Multimedia network has 2 different meanings, namely network and multimedia. A network (computer network) is a collection of computers and other supporting computer devices connected to each other in a single unit. Computer network media is divided into 2, namely, via cable (wired) or wirelessly (wireless) so that it allows users to exchange information with each other, such as sending documents and data.

While multimedia comes from 2 words, namely 'multi' which means many, and 'media' which means something that is used to convey or carry something. It can be interpreted that multimedia is the use of computer devices to display various kinds of information such as sentences or text, images, audio, video, and others. Based on the two definitions above, it can be concluded that Multimedia Network is the use of computers connected to each other to combine text, images, audio, video, and others to interact with each other [7].

D. Real-Time Streaming Protocol

RTSP (Real Time Streaming Protocol) is a application level protocol for controlling real time restricted data transmission over a network. RTSP can transmit audio and video data in real time controlled and on demand, provide services such as fast forward, rewind, stop, play and other control functions. RTSP do not transmit streaming data itself unless using RTP (Real-time Transport Protocol). The advantage of RTSP is that it provides a state-of-the-art connection between the server and client, allowing the user to pause or replay a position in the stream. In its implementation, RTSP can be connected with compression services such as MPEG-4 and H.264, where users can view video with sound [8].

E. Lossy Compression

Lossy Compression is when the data compressed. Lossy Compression is based on the assumption the data hold lot of information than humans being perceived. Thus, irrelevant data can be deleted [9]. In other words, this method is uses for reduce size of data for content, handling, storage and delivery. One of the advantages is the application of multimedia files such as video (MPEG-4, MKV and WMV) which can make the file size much smaller than the original size using the Discrete Cosine Transform algorithm.

F. Discrete Cosine Transform

Discrete Cosine Transform (DCT) is common lossy compression technique for processing compression of image and audio. DCT is used to convert data to summation a series of cosine waves that oscillate at different frequencies [9]. DCT involves using the Cosine function and real coefficients, whereas the Fourier Transform using the sine and cosine functions is much more efficient because fewer functions are required to approximate the signal. Both Fourier are spatial domains into frequency domains, and their respective functions convert something back. The stages in this DCT method are video data is processed starting from the prediction stage, then

transform & quantization and entropy encoding then comes out as a codec sequence. Furthermore, the results of the codec sequence will be reprocessed through the stages of entropy encoding, rescale & inverse transform, and reconstruction, so that it becomes a video. The stages of the process are displayed in Figure 3. Encode stage is the first step input into a context vector it's a codec and then a decode is a context vector to generate the output of the video.
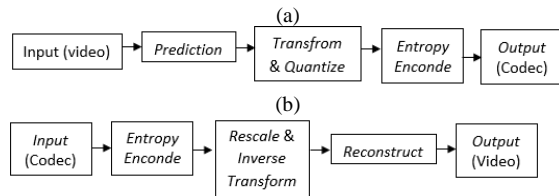

Fig. 3. (a) Stages of Encode and (b) Decode DCT Method.

G. H.263 (MPEG-4 ASP)

The H.263 video codec is the standard for compression of video at low bit rates. The H.263 is primary objective for encode video at very low bit rates in applications such as public telephone, narrowband integrated service digital and cellular networks. H.263 has seven basic types of frames of which only the first two are mandatory, which are independently encoded Intra also called I-frames. The I frames are encoded without reference to the other frames being encoded relative to the previous or I P frames, whereas the B frames are predicted bidirectional, i.e. they are predicted concerning the past and future frames. The compression given is the highest in the B-frame and the least in the I-frame. In addition to this PB, Increase in PB, EI and EP [10].

H. H.264 (MPEG-4 AVC)

H.264 is an standard video encoding also known as a MPEG AVC, represents an advanced video compression technology. This standard was developed in collaboration with the International Organization for Standardization's MPEG (ISO/IEC/SG29/WG11) group and the International Telecommunication Union's video coding expert group (VCEG, ITU-T/SG16/Q.6). H.264 codec for application in low bitrate video to high definition video.
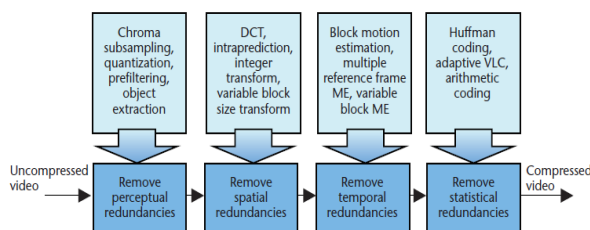

Fig. 4. Compression video stages.

Fig. 4 shows the various techniques that video encoding algorithms use to reduce redundancy. Video compression algorithms have various techniques that can be used to reduce redundancy in some ways, these techniques are applied. The H.264 video codec is standardization for decoding process not a encoding process.
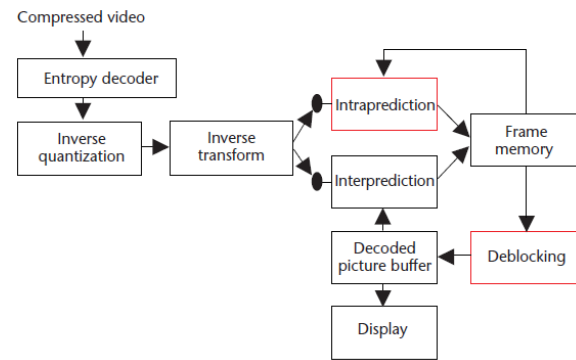

Fig. 5. H.264 codec decompression stages.

Fig. 5 shows the key components of the H.264 decoder. This decoding structure, without the new Intra-prediction and Deblocking components in H.264, like most video codecs in use today [11].

I. MJPEG (Motion JPEG)

MJPEG is a video compression format in which each video frame or interlaced plane of a digital video sequence is compressed separately as a JPEG image. MJPEG is recently used by video recording devices such as digital cameras, IP cameras, and webcams; and with a non-linear video editing system. The MJPEG format has the disadvantage that it requires higher bandwidth than H.264, MPEG-2 and MPEG-4. However, in the case of vehicles, MJPEG is preferred because video is usually used to store event-oriented still images or to transmit them to other devices [12].

MJPEG (Motion JPEG) is widely used in the field of non-linear editing. MJPEG can be used for frame editing and multi-layer image processing. This method treats a motion video sequence as a continuous still image. This compression method compresses each frame individually and completely. In the editing process, each frame can be saved randomly and can be edited according to the accuracy of the frame. Since MJPEG only compresses spatial redundancy within frames and does not compress redundancy differences between frames, the compression efficiency is not high.

J. Motion Detection

Motion detection is the process of detecting the change or moving of an object. It can be processed mechanical or electrical method. Motion detection is crucial for image processing in the system [13]. It has been widely used in computer vision applications such as MPEG4 image compression, target tracking, video retrieval, human recognition and visual surveillance. The motion detection has an objective for locating foreground objects in the scene to further analysis [14]. The video can be captured using a stationary or moving camera such as IP Camera [15].

K. OpenCV

OpenCV (Open-Source Computer Vision) is a library that is used to process images and videos so that we can get the information in them. OpenCV has more than 2500 optimized algorithms, including a range of machine and machine learning algorithms. This algorithm can detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, etc.

The advantage of using OpenCV is that it can make more use of time and resources in image processing and is easy to translate. In addition, OpenCV is open source, so it is easy to obtain. In its application, there are 3 ways to perform motion detection with OpenCV, including the following:

- Frame differencing: The frame differencing method uses two or three adjacent frames based on time series images to reduce and obtain the different images.
- Background subtraction: Commonly used methods for segmenting motion in static images. It will detect moving areas by subtracting the current pixel-by-pixel image from the reference background image created by averaging the image over time in the initialization period.
- Optical flow: The optical flow method uses a motion target's characteristic vector that changes over time to detect areas of motion in a sequence of images [13].
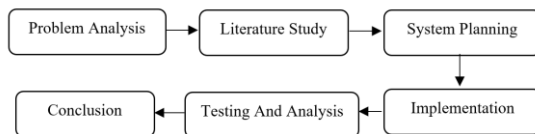
III. RESEARCH METHODS

Fig. 6. Research Methods.

The first step for this research is to conduct a literature study. This step is done to collect all references related to the research. The reference collection stage in this literature study also includes an analysis of problems and system requirements. The collected references include a description of the theory to the chosen method to solve the problem.

When all references and requirements for the system have been obtained, the next step is system design. This stage is done by designing the form of the system that will be used according to the needs analysis. After completing the system design, the system will be implemented at the implementation stage. The implementation stage is carried out the functions of the requirements that have been designed. The results of the implementation stage will be tested at the system testing stage. System testing is carried out to measure the success of the system based on predetermined parameters. After the test results are obtained, conclusions can be drawn from the research conducted and suggestions for possible future research.

IV. SYSTEM PLANNING

A. System Topology

In the design of the topology will be made 3 nodes. The first thing that needs to be done is to design the topology of sending video from the camera to the fog node which is then forwarded to the cloud server.
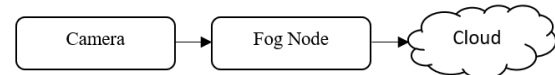

Fig. 7. System Topology

In Figure 7 it is explained that the camera will stream to the fog node. Then the fog node is in charge of performing video compression, computing, and uploading videos. While the cloud is in charge of storing the videos that have been computed.

B. System Flowchart

The system flowchart explains the design of the running process of the system that will be used in the implementation of the lossy compression method in fog computing. The system flowchart will be shown in Figure 8 below.
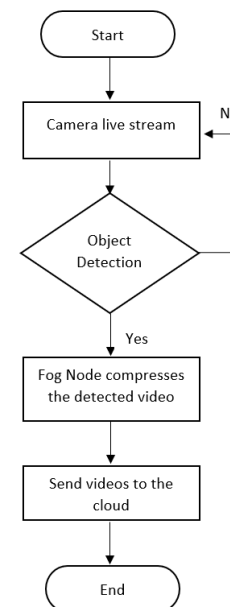

Fig. 8. System Flowchart

C. System Architecture

System architecture is a description of the system that will be implemented in the system. The system architecture on the system will be shown in Figure 9. below.
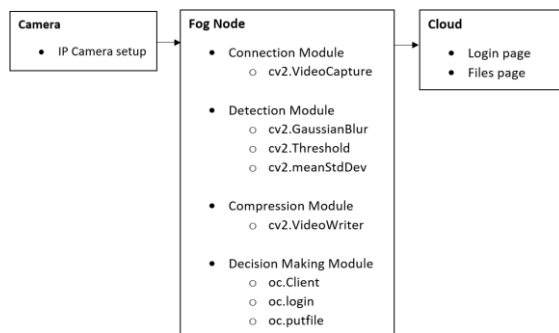
Fig. 9. System Architecture

The modules used in this system are libraries belonging to OpenCV (cv2) and OwnCloud (oc).

D. Evaluation Method

The evaluation method is a stage that is carried out after implementation is complete. In the evaluation method, there is a testing stage which will later prove that by using the compression method, the resulting video will be minimized, so that it does not burden the storage on the fog node. In this study, there are several scenarios that exist to test the effect of the compression method on the fog node and analyze the results obtained including CPU usage and memory usage. All tests were recorded at 20 fps and run for 1 hour (3600 seconds) to obtain similar results. There are parameters used in this test, including; Compression results and fog node performance are based on CPU Usage and Memory Usage. In this study, three test scenarios will be used. The test scenario can be seen in Table 1.

TABLE 1. TESTING SCENARIOS

No.	Scenario	Explanation
1.	Scenario 1	The OpenCV implementation uses the H.263 lossy compression codec for 1 hour duration of recorded.
2.	Scenario 2	The OpenCV implementation uses the H.264 lossy compression codec for 1 hour duration of recorded.
3.	Scenario 3	The OpenCV implementation uses MJPEG lossy compression codec for 1 hour duration of recorded.

V. EXPERIMENT AND RESULT

A. Camera Implementation

In this research, an IP Camera has a specification with is 3 megapixel and FHD 1080P maximum video output resolution. The IP Camera can operate using the wireless network with 3 antennas behind. It will be used to record video then be streamed to the fog node. On the camera used there is an RTSP protocol that is used to stream. The data streamed here are all frames recorded by the camera. Later these frames will be processed by the fog node before being sent to the cloud.

Fig. 10. IP Camera

B. Fog Node Implementation

In this research, fog node uses Raspberry PI 4 as a microcontroller that will perform computation, compression and video upload. When the streaming process takes place, computations will be carried out in the form of motion detection to determine whether there are moving objects in the video. The duration of this experiment is 1 hour in 2x3 room with 1 detected person. After the computing process is complete, the compression process will be carried out to compress the video. After the compression process, the video will be forwarded to the cloud. The compression method used is the lossy compression method. The lossy compression method aims to produce videos that are smaller but still usable. The codecs used are H263, H.264, and MJPEG with .mp4 container. In the lossy compression method, the resulting video is in .mp4 format with a resolution of 640x480p. Pseudocode lossy compression can be seen in Table 2.

TABLE 2. PSEUDOCODE LOSSY COMPRESSION

```
1   START
2      SET threshold value
3      GET start time
4      SET fourcc lossy codec
5      SET output video
6      DEF distMap (frame1, frame2)
7         GET frame1 distance value
8         GET frame2 distance value
9         CALCULATES  different  between
10  frame
11         CALCULATES  absolute  difference
12  norm
13         CALCULATES  normalize  distance
14  value
15         RETURN distance
16      SET stream source
17      GET check1
18      GET check2
19
20      WHILE True
21         GET check3
22         IF check3 = True
23            GET rows and columns
24            CALCULATES distance frame1 and
25  frame3
26            SET GaussianBlur smoothing
27            GET threshold value
28            CALCULATE standardDeviation
29         IF stDev > sdThresh
30            CREATE FILE
```

```
31        GET endTime
32          SET durasi
33          IF durasi > durasi_rekam
34             SET selesai = True
35     RELEASE camera
36     RELEASE output
37     PROGRAM upload1
38     GET destroyAllWindows
39     PROGRAM lossy
40  END
```

4	58.759 KB	165.691 KB	303.269 KB
5	57.391 KB	166.455 KB	303.988 KB
6	57.087 KB	167.411 KB	303.442 KB
7	58.802 KB	165.652 KB	304.547 KB
8	56.447 KB	167.137 KB	304.202 KB
9	56.058 KB	165.783 KB	303.596 KB
10	57.184 KB	165.887 KB	302.178 KB

C. Cloud Server Implementation

The cloud server used in this study is Amazon Web Service which uses a web-based storage application to store video data recorded from the camera node. The storage application used is OwnCloud. AWS is used because it can provide highly reliable, scalable and low cost of infrastructure in the clouds. Its very easy to access servers, storage, and a broad set application services in over internet [17]. OwnCloud has a freely available software which seen a rapid development and one of the open source STaaS products. The OwnCloud appears to be geared towards running on other web servers and operating system does exist. Other benefit of using OwnCloud, client can make standard file upload, download, delete, rename, capabilities, default web interface allows user to play a media file, viewing images in the gallery and maintain a calendar and contact list [18].
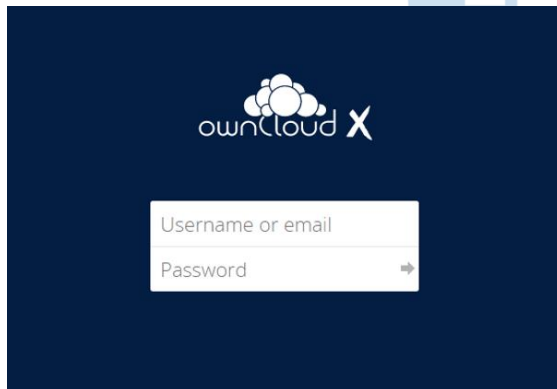
The results of the compression ratio will be displayed in graphical form in the following figure.
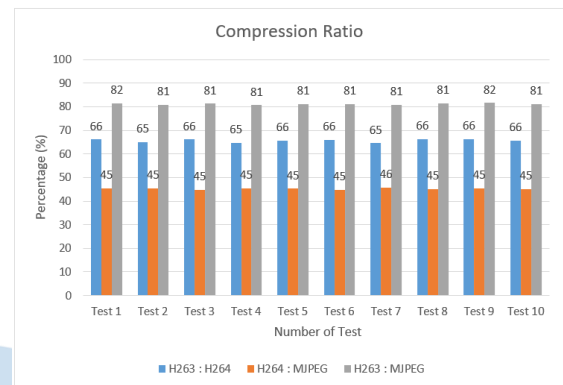

Fig. 12. Compression ratio.

The result of the compression space saving is H.263 codec has the highest compression is 58.802 KB at Trial 7 and lower compression is 56.058 KB at Trial 9. The H.264 codec has the highest compression is 167.452 KB at Trial 3 and the lowest compression is 165.691 at Trial 4. The MJPEG codec has the highest compression 304.752 at Trial 1 and the lowest compression is 302.178 at Trial 10. Based on the result in Table 3, is H.263 codec has a smaller compression file size than H.264 and MJPEG codec.

E. CPU Usage Utility

CPU Usage is the amount of time for which CPU for processing an instruction of computer program, the example is waiting input/output data. This parameter is used to see CPU Usage performance on the fog node when motion detection is running. CPU Usage is monitored using the formula in Table 2 for ten trials. The following are the results of the CPU Usage test results can be seen in Table 4.

Fig. 11. OwnCloud login page.

D. Compression Ratio

This parameter is used to calculate the relative reduction in data size represented by different compression algorithms. By using the Space Saving method by determining the cut in size that is proportional to the uncompressed size using equation (1).

$$Space\ Saving = 100x(1 - \frac{compressed\ video\ size}{original\ video\ size})(1)$$

Based on the test scenarios that have been carried out, the following results were obtained:

TABLE 3. COMPARISON OF COMPRESSION RESULTS

Trial	Lossy Compression		
	H.263	H.264	MJPEG
1	56.347 KB	166.248 KB	304.725 KB
2	57.986 KB	165.788 KB	303.030 KB
3	56.763 KB	167.452 KB	302.571 KB

TABLE 4. COMPARISON OF CPU USAGE ON FOG NODE

Trial	CPU Usage (%)		
	H.263	H.264	MJPEG
1	22.1	19.8	22.2
2	24.1	25.2	24.0
3	24.7	23.9	26.3
4	23.3	25.1	23.0
5	23.5	26.8	25.8
6	24.2	26.5	25.7
7	23.3	26.3	25.6
8	24.8	27.8	26.3

| 9 | 24.5 | 26.9 | 24.7 |
| 10 | 24.1 | 27.8 | 24.6 |

F. Memory Usage Utility

Memory Usage Utility is the amount of memory available when the applications running on your system. This parameter is used to see the performance of Memory Usage on the fog node when motion detection is running. Memory Usage is monitored using the formula in Table 2 for ten trials. The results of the Memory Usage test on the fog node can be seen in Table 5.

TABLE 5. COMPARISON OF MEMORY USAGE ON FOG NODE

Trial	Memory Usage (%)		
	H.263	H.264	MJPEG
1	22.3	22.6	23.1
2	24.5	25.4	25.6
3	22.2	23.0	25.2
4	23.5	24.2	23.9
5	26.9	31.5	25.1
6	23.6	26.1	26.3
7	23.0	25.2	26.8
8	22.8	25.2	23.0
9	23.1	25.9	25.3
10	22.4	30.5	27.0

VI. CONCLUSION

Based on the results of the tests conducted by comparing three different types of lossy compression codecs with 1 hour duration of recorded. It can be seen that the H.263 codec produces a greater efficiency value than other codecs, which is 66%. Codec H.263 produces an average video size of 57,282 KB. Then, the H.264 codec produces an average video size of 166,350 KB. While the MJPEG codec produces an average video size of 303,555 KB. The H.263 codec has a greater efficiency because of the standard video compression is at very low bit rates. This codec commonly can be used in application such as cellular network, public telephone networks and narrow band integrated service digital network.

The utility value of the fog node does not have a significant difference using three different types of lossy compression codecs. Codec H.263 produces an average utility value of 23.8% and 23.4%, respectively. Then, the H.264 codec produces an average utility value of 25.6% and 25.9%. While the MJPEG codec produces an average utility value of 24.8% and 25.1%.

REFERENCES

[1] I. Odun-Ayo, O. Ajayi, B. Akanle and R. Ahuja, "An Overview of Data Storage in Cloud Computing," IEEE, Jammu, 2017.

[2] M. R. Raza, A. Varol and N. Varol, "Cloud and Fog Computing: A Survey to the Concept and Challenges," Beirut, 2020.

[3] M. Chiang, S. Ahn and M. Gorlatova, "Leveraging fog and cloud computing for efficient computational offloading," in *IEEE MIT Undergraduate Research Technology Conference (URTC)*, Cambridge, 2017.

[4] H. M. Musse and L. A. Alamro, "Cloud Computing: Architecture and Operating System," IEEE, Sousse, 2016.

[5] S. Yi, Z. Hao, Z. Qin and Q. Li, "Fog Computing: Platform and Applications," Washington DC, 2015.

[6] M. Aazam, S. Zeadally and K. A. Harras, "Fog Computing Architecture, Evaluation, and Future Research Directions," in *IEEE Communications Magazine*, T. S. El-Bawab, Ed., Jackson, IEEE, 2018, pp. 46-52.

[7] Farhat, "Jaringan Multimedia," Universitas Gunadarma, Jakarta, 2005.

[8] D. Chu, C.-h. Jiang, Z.-b. Hao and W. Jiang, "The Design and Implementation of Video Surveillance System Based on H.264, SIP, RTP/RTCP and RTSP," Hangzhou, 2013.

[9] K. Ramya and M. Pushpa, "A Survey on Lossless and Lossy Data Compression Methods," *International Journal of Computer Science and Engineering Communications,* vol. 4, no. 1, pp. 1277-1280, 2016.

[10] I. U. Khan, "Performance Analysis of H.264 Video Coding Standard and H.263 Video Coding Standard," *VSRD Technical & Non-Technical Journal,* vol. 2, no. 1, pp. 8-14, 2011.

[11] H. Kalva, "The H.264 Video Coding Standard," *IEEE MultiMedia,* vol. 13, no. 4, pp. 86-90, 2006.

[12] D. Kim, K. Hong, B. Jung and J. Kim, "A Real-Time Streaming System for a Multifunctional Vehicle Information Display Based on TMO," Ho Chi Minh City, 2015.

[13] M. Sahasri and C. H. Gireesh, "Object Motion Detection and Tracking for Video," *International Journal of Engineering Trends and Technology,* no. Special, pp. 161-164, 2017.

[14] fourcc.org, "FOURCC.org - Video Codecs and Pixel Formats," 2011. [Online]. Available: https://www.fourcc.org/fourcc.php. [Accessed 25 September 2019].