JURNAL INFOTEL
Informatics - Telecommunication - Electronics
Website: http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN: 2085-3688; e-ISSN: 2460-0997

Literature study of learning-based video compression

Kholidiyah Masykuroh[1,*], Eueung Mulyana[2]
[1]Doctoral Program of Electrical and Informatics Engineering School, Bandung Institute of Technology
[2]Electrical and Informatics Engineering School, Bandung Institute of Technology
[1,2]Jl. Ganesha No. 10, Bandung 40132, Indonesia
[*]Corresponding email: 33222009@mahasiswa.itb.ac.id

Received 5 April 2023, Revised 26 June 2023, Accepted 14 August 2023

Abstract — Developments in telecommunications technology today, such as cellular with the fifth generation (5G), the development of IoT prototypes, and the migration of analog TV to digital TV starting in 2022. The development of various research using machine learning. The problem with video format information is that the video file size is quite large, so the transmission process requires a large bandwidth. In addition, sharing services such as video on demand (VoD) and Video Broadcasting are sensitive to delay. In comparison, the transmission media has limited capacity, such as terrestrial TV, Ethernet/Fast Ethernet, and wireless cellular data such as 2G, 3G HSPA, 4G, *etc*. Based on reports from Cisco, the development of internet users has increased by 10 % per year, with 80 % of total traffic using video. Developments in various video compression standards, such as the most recent H.264 and H.265, produce high-quality, low-bitrate video. Much research has been carried out with various proposed compression methods based on machine learning. Either uses singular block learning based or end-to-end. This research focuses on the literature study of video compression with machine learning.

Keywords – end-to-end, machine learning, video, video compression

I. Introduction

As an example of the telecommunications process, at one time, a person wanted to contact another person, and the information medium chosen to communicate was video. The problem faced when using video is its large size. The information is sent through a channel or information media that has limited capacity. In addition, various technological developments are currently happening on the network side. Today's technology development is from the cellular communication system migrating from 4G/LTE to 5G. In addition, the migration from analog TV to digital TV is underway in Indonesia. In recent years, the internet of things (IoT) has been utilized to develop various products. Additionally, in data processing, different kinds of research are being conducted in artificial intelligence, machine learning, deep learning, neural networks, *etc*. [1].

Based on information obtained [2], the number of internet users in 2023 will increase by around 10 percent per year in the Asia Pacific (APAC) region. In addition, Yang *et al.* [3] and Chen *et al.* [4] states that about 80 % of internet traffic uses video. Vashistha [5] describes several video analytics applications, as shown in Fig. 1. These applications include media, security & surveillance, autonomous car, smart cities, smart home, and retail businesses.

Increasing use of video in telecommunications traffic as predicted by Cisco. This condition is the basis for the importance of studying video compression. There are two main benefits of video compression, the first is for storage needs, and the second is to reduce the bit rate to meet capacity standards in telecommunications networks. The current development of video compression standards has succeeded in reducing the bitrate with good video quality. The video compression standards currently being developed are H.264 and H.265, or HEVC. In addition, mobile edge computing (MEC) by ETSI has now introduced the development of the telecommunications network itself. MEC is a cloud server that works on the edge of the mobile network. One of the use cases of MEC is video analytics which can be used for various applications such as video surveillance, traffic detection, public security, to smart cities.

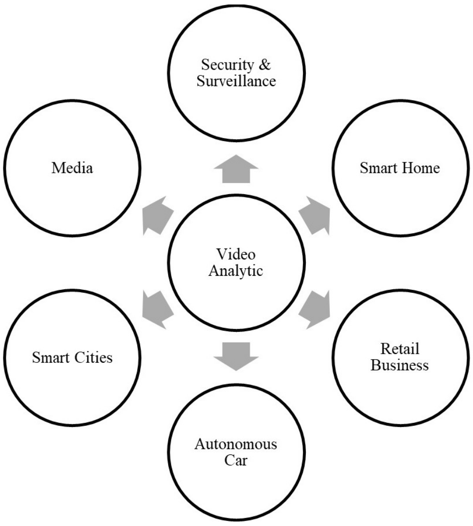

Fig. 1. Video analytics applications [5].

The MEC has computing capabilities that support learning-based video compression. The development of learning-based video compression is based on developing the H.264 standard by optimizing one block and a combination of several blocks from the H.264 architecture. In addition, research [6] has been carried out by developing digital video coding (DVC) by optimizing motion estimation using CNN and arithmetic entropy coding. Therefore, learning-based video compression can be developed to optimize the end-to-end combination of architectural blocks.

This paper results from a literature study starting from the development of learning-based video compression, the video compression architecture used, and development opportunities. This information is extracted from various literature that has been read and discussed with experts. This paper is structured into five sections. They are the introduction, the research method, the learning-based video compression development, the discussion, and the conclusions.

II. Research Method

This section discusses basic of Shannon information theory, the development of video compression standard, and MEC.

A. Basic of Shannon Information Theory

Information theory was introduced by Shannon about 50 years ago. Shannon's Information Theory is used to determine entropy values. The entropy value can be calculated from the value of the probability distribution. The entropy value is the sum of the probability of information appearing multiplied by that probability's base two logarithm value. The entropy value is the lower limit value for representing information. In other words, the entropy value indicates the lower limit to which information can be compressed. After the emergence of Shannon's information theory, various compression techniques emerged. We know and are still studying several compression techniques: Shannon, Shannon-Fano, Huffman, Arithmetic, Binary Arithmetic, etc. In addition, compression techniques based on block-based coding, transformation coding, and predictive coding have also been developed [7].

B. Development Video Compression Standard

The first video encoder was MPEG-2 in 1994, while MPEG-4 was developed in 1999. The next encoder was designed as advanced video coding (AVC) in 2003. Finally, a new high-efficiency video coding standard called high-efficiency video coding (HEVC) was formed in 2012, and a new technology called future video coding (FVC) was developed starting in 2021 [8].

The video compression standards developed by ISO and ITU are the MPEG series and the H.26h series. Table 1 describes the differences between the MPEG and H.26h series standards. MPEG standard video compression is used for entertainment services. Some of its uses are for media storage and broadcasting. Meanwhile, the H.26h series video compression standard is used for communication services, namely video conferencing. ISO and ITU developed these two video compression standards into MPEG-4, which can be used for multimedia compression services, website authoring, and wireless videophones [9]. The latest video compression standard is H.265 or HEVC, used in DVB-T2 technology.

The movement of objects between temporally adjacent frames causes a large variation in frames. If the number of objects that move between the current frame and the previous frame can be determined, the difference can be calculated by aligning the objects in the current frame with the previous frame. This determination would probably produce a zero error in principle. However, this comparison did not happen for several reasons. First, object movement uses a variation of the pixel value. Because a 2D image is a projection of a 3D image, determining whether a rigid or deformed object causes motion is difficult. Deformable body movement implies that different body parts can move to varying degrees. As a result, one can appreciate the difficulty in estimating the temporal movement of objects between frames of contiguous image sequences [10].

C. Mobile Edge Computing (MEC)

MEC is a cloud server that works on the mobile network's edge and performs special tasks that traditional network infrastructure cannot [11].

ETSI promotes the MEC architecture. MEC provides low latency, high bandwidth, and context-aware video streaming services. MEC resources, such as base stations (BSs), access points (APs), and radio access networks (RAN), are deployed at the network's

Table 1. Video Compression Standards Comparison

	MPEG-1	**MPEG-2**	**MPEG-4**	**H.263**	**H.261**
Dates of Standardization	11/92	11/94	1/99 Version 1 1/00 Version 2	5/96 Version 1 1/98 Version 2	12/90 Version 1 5/94 Revised
Primary Application	Digital storage media	Broadcast/ DVD/ HDTV	Web authoring, Multimedia compression, Wireless videophone	Desktop/ Wireless videoconferencing	Wireline video conferencing
Typical Video Bitrates	1.5 Mbps	4-6 Mbps	20 Kbps - 6 Mbps	20-384 Kbps	128-384 Kbps
Typical Video Frame Dimensions	352 × 240 (SIF)	720 × 480 (Rec. 601) 720 × 480 (601)	176 × 144 (QCIF) 352 × 288 (CIF)	176 × 144 (QCIF) 352 × 288 (CIF)	176 × 144 (QCIF) 352 × 288 (CIF)
Typical Associated Audio Quality	Stereo CD quality	Surround sound	Speech / Music / Stereo CD / Surround	Speech	Speech

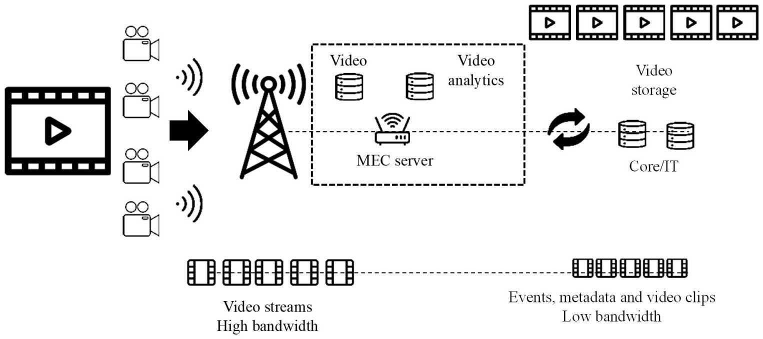

Fig. 2. Video Analytics [11].

edge. MEC's applications and use cases include intelligent video acceleration, video streaming analysis, augmented reality service, and connected vehicles [12].

MEC's use cases include active device location tracking, augmented reality content delivery, video analytics, RAN-aware content optimization, distributed content and DNS caching, and application-aware performance optimization. For example, one of MEC's use cases, video analytics, is depicted in Fig. 2. Video streams received from cameras via LTE uplink are transcoded and stored by the video management application. The video analytics application analyzes video data to detect and notify users of specific configurable events such as object movement, lost children, abandoned luggage, etc. Furthermore, the application sends low-bandwidth video metadata to the central operations and management server for database searches. As a result, safety and public security applications, as well as smart cities, are possible [11].

III. Learning-based Video Compression Development

There are two compression methods in general, namely lossless and lossy. Lossless compression is a method that is used to minimize the number of bits in representing an image or video by having the ability to reconstruct it back into the original data. Meanwhile, lossy compression is a method for maintaining acceptable quality from the results of data reconstruction [9].

Based on the results of a literature survey conducted in [13], this survey describes the various video compression and machine learning techniques that are being used. The conventional learning-based cooperation approach and the learning-based end-to-end compression approach are the two techniques for learning-based video compression methods. The conventional-learning-based cooperation approach includes four techniques: intra-prediction using CNN [14], [15], MSCNN [16], GAN [17], and RAN [18], [19]. Inter-prediction and post-processing using NN interpolation [20], dictionary learning [21], CNN [22]. In-loop filter using adaptive sample offset (SAO) [23], adaptive loop filtering [24], and in-loop filter [25] with machine learning techniques such as CNN [26], DRN [27], and Deep CNN [28]. Deep CNN learning-based guidance of a conventional codec [29].

Meanwhile, there are two techniques in the learning-based end-to-end compression approach, they are predictive video coding and generative video coding. Predictive video coding has three methods: learning-based flow estimation with machine learning techniques such as CNN [6] and deep CNN [30]. Second, the motion compensation becomes a warping function using CNN [31], [32]. Third, the GAN-based learning-based frame synthesis model will perform the reconstruction task [33], [34]. Then generative video coding has two methods. First, the variational autoencoder network using CNN [35]–[37]. Second, the training frames and autoregressive dynamics number of priors using soft-then-hard quantization [38].

According to Hoang and Zhou [13], several methods have been developed for Intra prediction: block-wise

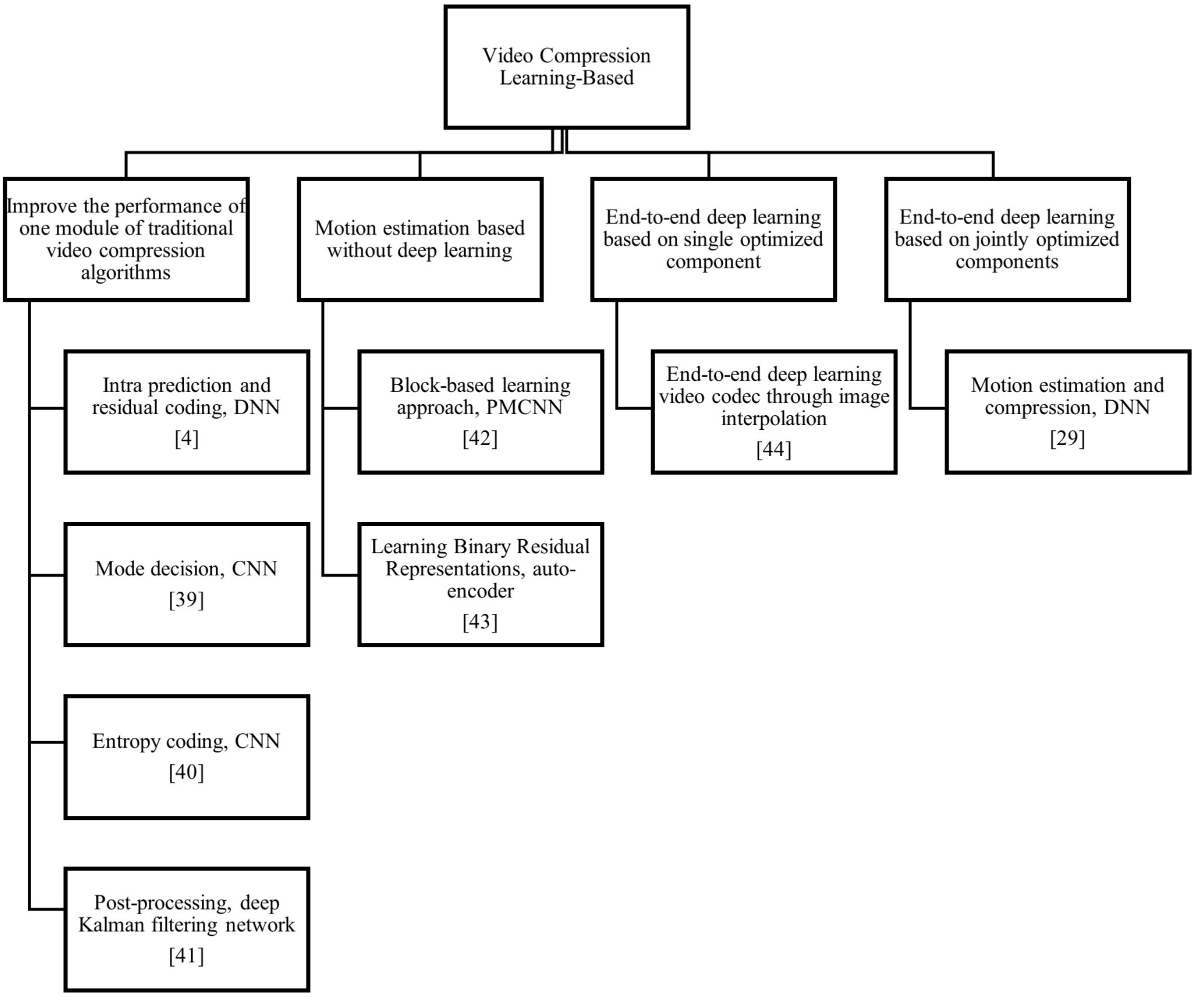
Video Compression Learning-Based
Improve the performance of one module of traditional video compression algorithms
Motion estimation based without deep learning
End-to-end deep learning based on single optimized component
End-to-end deep learning based on jointly optimized components
Intra prediction and residual coding, DNN [4]
Mode decision, CNN [39]
Entropy coding, CNN [40]
Post-processing, deep Kalman filtering network [41]
Block-based learning approach, PMCNN [42]
Learning Binary Residual Representations, auto-encoder [43]
End-to-end deep learning video codec through image interpolation [44]
Motion estimation and compression, DNN [29]

Fig. 3. Video compression learning-based method [6].

prediction, coding tree unit (CTU), and angular prediction. At the same time, the parameters achieved are the average BD-BR (Bjøntegaard delta bit rate). BD-BR is the difference value of the bit rate obtained based on two test scenarios using HM and VTM software.

According to Lu *et al.* [6] as shown in Fig. 3, learning-based video compression has four main methods. First, one module of traditional video compression algorithms was improved, second by motion estimation based without deep learning, thirdly by end-to-end learning based on single optimized components, and fourthly by ent-to-end learning based on jointly optimized components. Four techniques are used to improve the performance of one module of traditional video compression algorithms, namely intra-prediction and residual coding using DNN [4], mode decisions using CNN [39], entropy coding using CNN [40], and post-processing using Kalman filtering network [41]. Motion estimation based without deep learning involves two block-based learning approach techniques using pixel motion CNN (PMCNN) [42] and learning binary residual representations using autoencoder [43]. End-to-end deep learning video codec through image interpolation is a technique from deep end-to-end learning based on a single optimized component [44]. And motion estimation and compression are techniques for end-to-end learning based on jointly optimized components [6].

According to Lu *et al.* [6], there have been many learning-based studies by optimizing one or more blocks with or without end-to-end. They [6] optimizes motion estimation and compression using CNN and entropy coding, combining two optimization components. In addition, it also implements an end-to-end video compression method. The test parameters are PSNR to bits per pixel (bpp) & multiscale structural similarity index (MSSIM).

Based on the literature survey that has been carried out, there is a development of the method used for learning-based video compression. The first is by extending one of the blocks from a standard video compression architecture. Then it develops by optimizing more than one block, or it is called jointly. Lu *et al.* [6] also shows that learning development using RNN aims to get progressive image compression. CNN is used to obtain a learning-based scheme by autoencoder. Existing research aims to minimize distortion using the mean square error (MSE) parameter. The MSE value is obtained by calculating the difference in error between

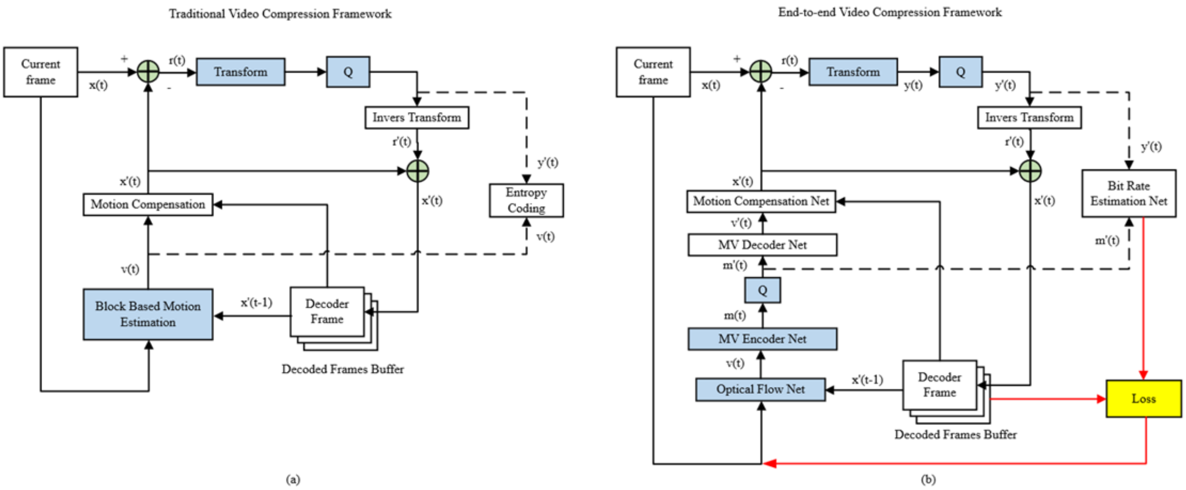

Fig. 4. (a) Traditional H.264 or H.265 predictive coding architecture (b) Scheme for neural network compression [6].

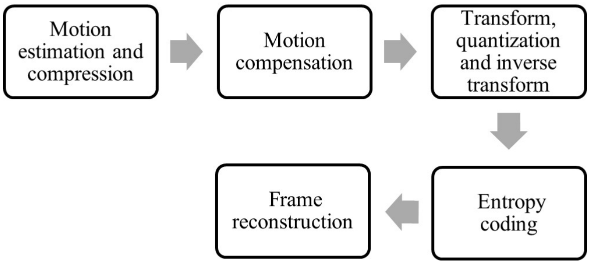
Motion estimation and compression
Motion compensation
Transform, quantization and inverse transform
Entropy coding
Frame reconstruction

Fig. 5. Learning-based video compression process [6].

the original image and the reconstructed image. And finally, is rate-distortion optimization (RDO) by getting a higher bit compression efficiency value, for example, using an arithmetic procedure.

IV. Discussion

Fig. 4 is an end-to-end compression video model combining motion estimation and compression optimization using CNN with entropy coding. An explanation of the series of processes carried out in the test is described in Fig. 5, starting from motion estimation and compression trained using CNN, followed by motion compensation, followed by the transformation, quantization, and inverse processes of the transformation, then entered into the entropy coding process using arithmetic and finally the frame reconstruction process [6].

A. *Motion-Compensated Prediction*

The movement of objects between temporally adjacent frames causes a large variation in frames. Suppose the number of objects moving between the current and previous frames can be determined. The difference can be calculated by aligning the objects in the current frame with those in the previous frame. This determination would probably produce a zero error in principle. However, this comparison did not happen for several reasons. First, object movement uses a variation of the pixel value. Because a 2D image is a projection of a 3D image, it is difficult to judge whether a rigid or deformed object causes motion. Finally, deformable body movement implies that different body parts can experience movement to different degrees. As a result, one can appreciate the difficulty in estimating the temporal movement of objects between frames of contiguous image sequences [10].

The movement of objects can be considered as translations, namely from left to right and up to down, and rigid bodies cause this movement. As a result, the boundaries of objects within the frame must first be determined to determine the movement of objects within the current frame and reference. As a result, rectangular blocks of the same size are used. The next step is to estimate the translational motion of the rectangular block between the current frame and the reference frame, referred to as motion estimation. This calculation will yield a motion vector with horizontal and vertical components. Once the motion vector has been determined, the current frame's rectangular blocks can be aligned with the reference frame to find the corresponding differential pixels. Motion-compensated prediction refers to aligning and distinguishing objects between frames [10].

Trials have been carried out regarding motion-compensated prediction using two methods. The first method is an Exhaustive or Full Search to find matched motion blocks using windows and blocks as described before. The second method is Pyramidal or Hierarchical Motion Estimation, which reduces the number of searches by estimating the motion amount from low-resolution to high-resolution frames. The test results can be seen in Fig. 6 and Fig. 7.

The frames compared are frames A and B, extracted from the video. Three frame reconstruction results are distinguished based on the observed pixels. This experiment uses a 16 × 16 window and an 8 × 8 motion

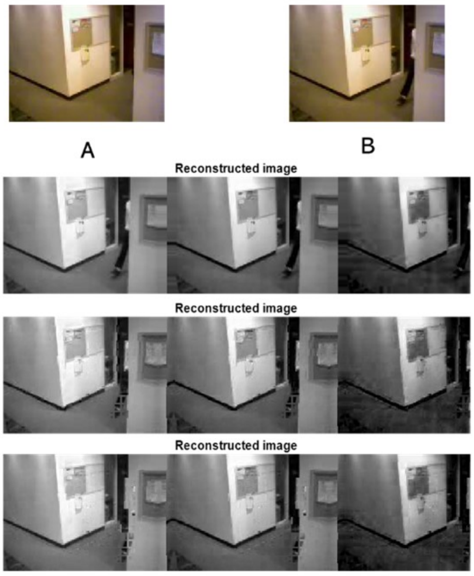

Fig. 6. Exhaustive or full search.

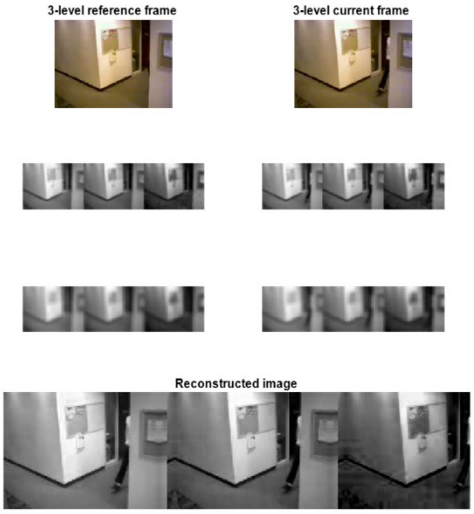

Fig. 7. Pyramidal or Hierarchical Motion Estimation.

block. The first reconstructed frame is full prediction with MSE with MC = 49,800 and MSE without MC = 147,994. The second reconstructed frame is a half prediction with a value of MSE with MC = 37,877 MSE without MC = 147,994. The third frame resulting from the reconstruction is the quarter prediction with MSE with MC = 27,743 and MSE without MC = 147,994. This experiment is explained in Fig. 5.

Fig. 7 is an experimental day of Pyramidal or Hierarchical Motion Estimation. The level is determined using a subsample of 2, namely 1 for the full native resolution of 1/4 and 1/16 of the original frame. Each block is filtered using a Gaussian low-pass filter.

The two methods are exhaustive or full search and pyramidal or hierarchical motion estimation. Both methods can predict frames based on previous and current frame information.

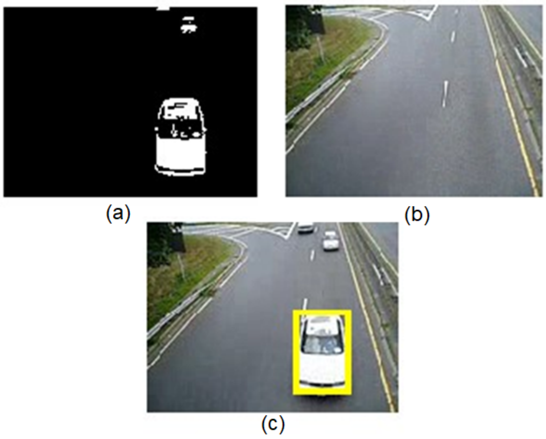

Fig. 8. Object detection: (a) Background, (b) Foreground, and (c) Bounding box.

The video used in the experiment is vipmen.avi. The experiment was carried out by extracting the frame and taking two frames, one as a reference frame (Frame A) and the second as the current frame (Frame B). Tests are carried out based on pixel information from the two frames. The algorithm can work to estimate the frames that are between the two.

B. Object Detection

Object detection is a process for finding objects in images or videos using background reduction, feature extraction, statistical methods, *etc*. [45]. The results of the experiments that have been carried out by separating the background and foreground are shown in Fig. 8.

The video used in the experiment is traffic.avi. This video is a recording from a surveillance camera of traffic conditions on the highway.

This object detection technique can also be used in self-driving by separating the background and foreground. However, in self-driving, the number of objects that need to be detected is more diverse. These objects include roads, sidewalks, trees, traffic lights, other vehicles, pedestrians, *etc*.

An example of object detection in self-driving can be seen in Fig. 9. The video used in the test is 01_city_c2s_fcw_10s.mp4. The experimental results show that several objects, such as roads, billboards, other vehicles, and trees, were detected. However, the disadvantage of this technique is that if there are objects with a dark or black color, they cannot be identified exactly. In addition, distinguishing one object from another still has limitations, so the entire object cannot be detected perfectly.

The video used in the trial is a video that comes from the library in Matlab. The trials were carried out

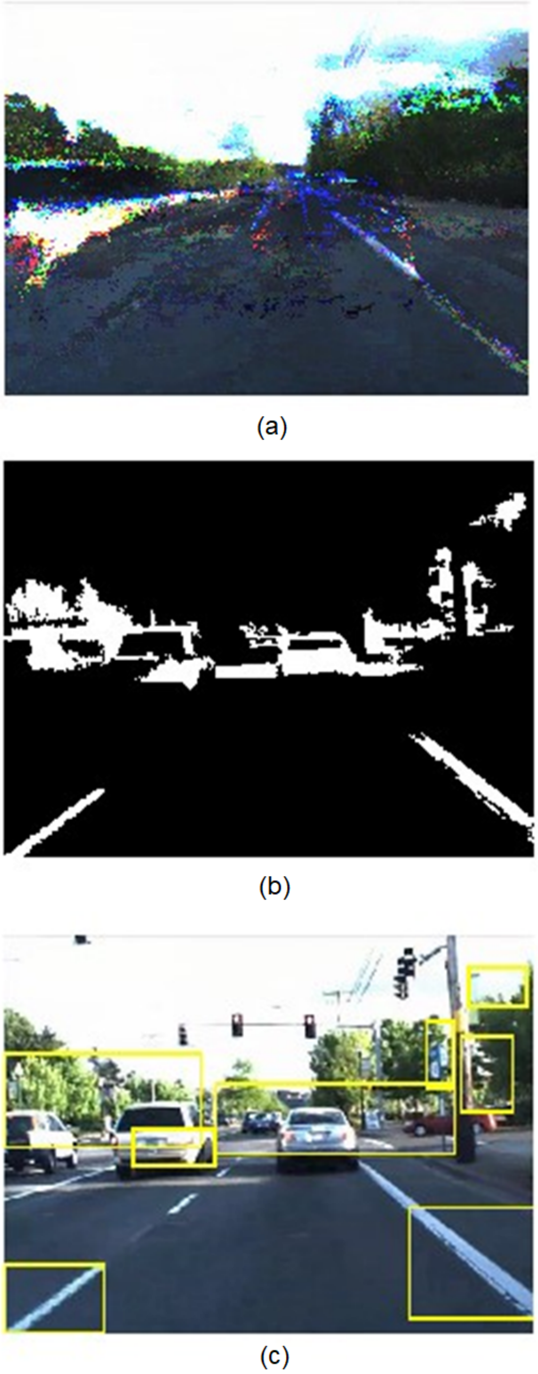

Fig. 9. Self driving: (a) Background, (b) Foreground, and (c) Bounding box.

using a simulation program that was run on Matlab R2020b.

V. Conclusion

Many studies have examined video compression with deep learning with optimization in the estimation section, so research development might be carried out on how to design a video compression model with a combination of optimization of the components, not only motion estimation using deep learning. Development opportunities can be carried out on motion compensation, entropy coding, quantization, *etc*. The parameters to be achieved in this study are quality through PSNR, bit rate, and compression ratio. As comparison data using video compression standards such as H.264 or H.265.

Compressed video can be used in various applications, such as multimedia communication systems using the motion compensation prediction method. In object detection by separating between background and foreground, and so on. The results of object detection that have been tested are influenced by the color components on the background and objects. If the background color and the object are not in contrast, then the object cannot be detected perfectly.

The challenge faced in developing learning-based video compression is bitrate reduction with good quality, which is very competitive with developing video compression using H.264 or H.265. Development of learning-based video compression by optimizing a combination of two architectural compression blocks using learning methods other than CNN, for example, RNN and entropy coding using other methods such as Huffman coding.

References

[1] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," *arXiv*, Jan. 13, 2020. Accessed: Oct. 14, 2022. [Online]. Available: http://arxiv.org/abs/2001.03569

[2] Cisco, "Cisco annual internet report (2018–2023)," *Cisco public*, 2020.

[3] Y. Yang, R. Bamler, and S. Mandt, "Improving inference for neural image compression," *arXiv*, p. 12.

[4] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, "DeepCoder: A deep neural network based video compression," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL: IEEE, Dec. 2017, pp. 1–4. doi: 10.1109/VCIP.2017.8305033.

[5] V. Vashistha, "Computer vision - object detection on videos - deep learning," Mar. 03, 2023. [Online]. Available: www.udemy.com

[6] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10998–11007. doi: 10.1109/CVPR.2019.01126.

[7] T. M. Cover and J. A. Thomas, "Elements of information theory, 2nd ed. Hoboken," New Jersey: John Wiley & Sons, Inc, 2006.

[8] D. Karwowski, T. Grajek, K. Klimaszewski, O. Stankiewicz, J. Stankowski, and K. Wegner, "20 years of progress in video compression – from MPEG-1 to MPEG-H HEVC. General view on the path of video coding development," in *Image Processing and Communications Challenges 8, R. S. Choraś, Ed., in Advances in Intelligent Systems and Computing*, vol. 525. Cham: Springer International Publishing, 2017, pp. 3–15. doi: 10.1007/978-3-319-47274-4_1.

[9] A. Katsaggelos, "Fundamentals of digital image and video processing," Northwestern University. [Online]. Available: https://www.coursera.org/

[10] K. S. Thyagarajan, "Still image and video compression with MATLAB," 1st ed. Wiley, 2010. doi: 10.1002/9780470886922.

[11] M. Patel, Y. Hu, P. Hede, J. Joubert, C. Thornton, B. Naughton, J. R. Ramos, C. Chan, V. Young, S. J. Tan, D. Lynch, N. Sprecher, T. Musiol, C. Manzanares, U. Rauschenbach, S. Abeta, L. Chen, K. Shimizu, A. Neal, P. Cosimini, A. Pollard, and G. Klas, "Mobile-edge computing – Introductory technical white paper," no. 1, Sep. 2014.

[12] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 2, pp. 871–903, 2021, doi: 10.1109/COMST.2021.3065237.

[13] T. M. Hoang and J. Zhou, "Recent trending on learning based video compression: A survey," *Cogn. Robot.*, vol. 1, pp. 145–158, 2021, doi: 10.1016/j.cogr.2021.08.003.

[14] I. Schiopu, H. Huang, and A. Munteanu, "CNN-based intra-prediction for lossless HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2019, doi: 10.1109/TCSVT.2019.2940092.

[15] Z.-T. Zhang, C.-H. Yeh, L.-W. Kang, and M.-H. Lin, "Efficient CTU-based intra frame coding for HEVC based on deep learning," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Kuala Lumpur: IEEE, Dec. 2017, pp. 661–664. doi: 10.1109/APSIPA.2017.8282116.

[16] Y. Wang, X. Fan, S. Liu, D. Zhao, and W. Gao, "Multi-scale convolutional neural network based intra prediction for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2020, doi: 10.1109/TCSVT.2019.2934681.

[17] L. Zhu, S. Kwong, Y. Zhang, S. Wang, and X. Wang, "Generative adversarial network-based intra prediction for video coding," *IEEE Trans. Multimed.*, vol. 22, no. 1, pp. 45–58, Jan. 2020, doi: 10.1109/TMM.2019.2924591.

[18] Y. Hu, W. Yang, S. Xia, W.-H. Cheng, and J. Liu, "Enhanced intra prediction with recurrent neural network in video coding," in *2018 Data Compression Conference*, Snowbird, UT: IEEE, Mar. 2018, pp. 413–413. doi: 10.1109/DCC.2018.00066.

[19] Y. Hu, W. Yang, S. Xia, and J. Liu, "Optimized spatial recurrent network for intra prediction in video coding," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, Taichung, Taiwan: IEEE, Dec. 2018, pp. 1–4. doi: 10.1109/VCIP.2018.8698658.

[20] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 6420–6428. doi: 10.1109/ICCV.2019.00652.

[21] J. Schneider, J. Sauer, and M. Wien, "Dictionary learning based high frequency inter-layer prediction for scalable HEVC," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL: IEEE, Dec. 2017, pp. 1–4. doi: 10.1109/VCIP.2017.8305019.

[22] J.-K. Lee, N. Kim, S. Cho, and J.-W. Kang, "Deep video prediction network-based inter-frame coding in HEVC," *IEEE Access*, vol. 8, pp. 95906–95917, 2020, doi: 10.1109/ACCESS.2020.2993566.

[23] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012, doi: 10.1109/TCSVT.2012.2221529.

[24] C.-Y. Tsai, C.-Y Chen, T. Yamakage, I. S. Chong, Y.-W. Huang, C.-M. Fu, T. Itoh, T. Watanabe, T. Chujoh, M. Karczewicz, and S.-M. Lei, "Adaptive loop filtering for video coding," *IEEE J. Sel. Top. Signal Process.*, vol. 7, no. 6, pp. 934–945, Dec. 2013, doi: 10.1109/JSTSP.2013.2271974.

[25] X. Zhang, R. Xiong, W. Lin, J. Zhang, S. Wang, S. Ma, and W. Gao, "Low-rank based nonlocal adaptive loop filter for high efficiency video compression," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2016, doi: 10.1109/TCSVT.2016.2581618.

[26] B. Wohlberg, "Convolutional sparse representations as an image model for impulse noise restoration," in *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, Bordeaux, France: IEEE, Jul. 2016, pp. 1–5. doi: 10.1109/IVMSPW.2016.7528229.

[27] Y. Wang, H. Zhu, Y. Li, Z. Chen, and S. Liu, "Dense residual convolutional neural network based in-loop filter for HEVC," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, Taichung, Taiwan: IEEE, Dec. 2018, pp. 1–4. doi: 10.1109/VCIP.2018.8698740.

[28] S. Kuanar, C. Conly, and K. R. Rao, "Deep learning based HEVC in-loop filtering for decoder quality enhancement," in *2018 Picture Coding Symposium (PCS)*, San Francisco, CA: IEEE, Jun. 2018, pp. 164–168. doi: 10.1109/PCS.2018.8456278.

[29] H. Lin, X. He, L. Qing, Q. Teng, and S. Yang, "Improved low-bitrate HEVC video coding using deep learning based super-resolution and adaptive block patching," *IEEE Trans. Multimed.*, vol. 21, no. 12, pp. 3010–3023, Dec. 2019, doi: 10.1109/TMM.2019.2919433.

[30] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 1647–1655. doi: 10.1109/CVPR.2017.179.

[31] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, and S.-M. Hu, "Deep online video stabilization with multi-grid warping transformation learning," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2283–2292, May 2019, doi: 10.1109/TIP.2018.2884280.

[32] M. Zhao and Q. Ling, "PWStableNet: Learning pixel-wise warping maps for video stabilization," *IEEE Trans. Image Process.*, vol. 29, pp. 3582–3595, 2020, doi: 10.1109/TIP.2019.2963380.

[33] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18-23 June 2018, Salt Lake City, UT, USA, pp. 8798–8807.

[34] C. Jia, X. Zhang, S. Wang, S. Wang, and S. Ma, "Light field image compression using generative adversarial network-based view synthesis," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 9, no. 1, pp. 177–189, Mar. 2019, doi: 10.1109/JETCAS.2018.2886642.

[35] A. Habibian, T. V. Rozendaal, J. Tomczak, and T. Cohen, "Video compression with rate-distortion autoencoders," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Oct. 2019, pp. 7032–7041. doi: 10.1109/ICCV.2019.00713.

[36] J. Luo, S. Li, W. Dai, Y. Xu, D. Cheng, G. Li, and H. Xiong, "Noise-to-compression variational autoencoder for efficient end-to-end optimized image coding," in *2020 Data Compression Conference (DCC), Snowbird, UT, USA: IEEE*, Mar. 2020, pp. 33–42. doi: 10.1109/DCC47342.2020.00011.

[37] J. Han, S. Lombardo, C. Schroers, and S. Mandt, "Deep generative video compression," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019, p. 12.

[38] Z. Guo, Z. Zhang, R. Feng, and Z. Chen, "Soft then hard: Rethinking the quantization in neural image compression," in *Proceedings of the 38 th International Conference on Machine Learning, PMLR 139*, 2021, p. 10.

[39] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5088–5103, Nov. 2016, doi: 10.1109/TIP.2016.2601264.

[40] R. Song, D. Liu, H. Li, and F. Wu, "Neural network-based arithmetic coding of intra prediction modes in HEVC," in *2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL: IEEE*, Dec. 2017, pp. 1–4. doi: 10.1109/VCIP.2017.8305104.

[41] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, "Deep kalman filtering network for video compression artifact reduction," in *Computer Vision – ECCV 2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11218. Cham: Springer International Publishing, 2018*, pp. 591–608. doi: 10.1007/978-3-030-01264-9_35.

[42] Z. Chen, T. He, X. Jin, and F. Wu, "Learning for video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 566–576, Feb. 2020, doi: 10.1109/TCSVT.2019.2892608.

[43] Y.-H. Tsai, M.-Y. Liu, D. Sun, M.-H. Yang, and J. Kautz, "Learning binary residual representations for domain-specific video streaming," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.12259.

[44] C.-Y. Wu, N. Singhal, and P. Krähenbühl, "Video compression through image interpolation," in *Computer Vision – ECCV 2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., in Lecture Notes in Computer Science, vol. 11212. Cham: Springer International Publishing*, 2018, pp. 425–440. doi: 10.1007/978-3-030-01237-3_26.

[45] M. Kaushal, B. S. Khehra, and A. Sharma, "Soft computing based object detection and tracking approaches: State-of-the-art survey," *Appl. Soft Comput.*, vol. 70, pp. 423–464, Sep. 2018, doi: 10.1016/j.asoc.2018.05.023.