2022 International Conference on Intelligent and Innovative Computing Applications
https://doi.org/10.59200/ICONIC.2022.030

Optimal Video Compression Parameter Tuning for Digital Video Broadcasting (DVB) using Deep Reinforcement Learning

William B. Motaung
Department of Electrical and Electronics Engineering,
University of Johannesburg,
Johannesburg, South Africa
wmotaung@gmail.com

Kingsley A. Ogudo
Department of Electrical and Electronics Engineering,
University of Johannesburg,
Johannesburg, South Africa
kingsleyo@uj.ac.za

Chabalala S.Chabalala
School of Electrical and Information Engineering,
University of the Witwatersrand,
Johannesburg, South Africa
chabalala.chabalala@wits.ac.za

Abstract
DVB (digital video broadcasting) has undergone an enormous paradigm shift, especially through internet streaming that utilizes multiple channels (i.e., secured hypertext transfer protocols). However, due to the limitations of the current communication network infrastructure, video signals need to be compressed before transmission. Whereas most recent research has concentrated and focused on assessing video quality, little to no study has worked on improving the compression processes of digital video signals in lightweight DVB setups. This study provides a video compression strategy (DRL-VC) that employs deep reinforcement learning for learning the suitable parameters used in digital video signal compression. The problem is formulated as a multi-objective one, considering the structural similarity index metric (SSIM), the delay time, and the peak signal-to-noise ratio (PSNR). Based on the findings of the experiments, our proposed scheme increases bitrate savings while at a constant PSNR. Results also show that our scheme performs better than the benchmarked compression schemes. Finally, the root means square error values show a consistent rate across different video streams, indicating the validity of our proposed compression scheme.

Keywords: deep learning, digital video broadcasting, multimedia streaming, reinforcement learning, video quality assessment

1. Introduction

The interest in multimedia services continues to increase at a tremendous pace, evident in the increase in data rate in 2021, and hoping to increase by a large percentage by the end of 2022 (Pocovi et al., 2017). Many of these services are offered through subscribed-internet-based media, such as YouTube, Vimeo, Netflix, and Amazon's prime video. This "paradigm shift" has influenced terrestrial satellite-based broadcasting companies to adopt internet-based multimedia services through a lighter digital video broadcasting (DVB) box or a parallel subscription that enables online streaming. A common protocol to achieve this online streaming is the dynamic adaptive streaming over HTTP (DASH), known for its simplicity and flexibility (Ekmekcioglu et al., 2016). However, these complementary options come with concurrent tasks during broadcasting. Video signal compression is a vital part of these tasks, given its correlation with video signal quality. The major problem of adopting over-the-top (OTT) multimedia signal transmission is the significantly noticeable difference in quality from the traditional satellite-based transmission. However, the limitations of the current communication network infrastructure (mostly 4G and 5G) do not permit the transmission of the full features of video signals. Hence, a need to compress (or remove some features from video signals to a certain threshold before transmitting (Santhi et al., 2003; Tse & Viswanath, 2005). Therefore, it is crucial to work on improving the compression process of video signals while satisfying customers' quality of experience (QoE).

Generally, studies have developed and adopted different standards, algorithms, or techniques for video signal compression. The H.264 and H.265 are the most widely used compression standards, particularly for motion-based codecs. Studies (Wiegand et al., 2003; Schwarz et al., 2007) have developed a video transmission framework using H.264 encoders to transform video signals. The form of video codec is reported to have some spatial redundancies since the fundamental principle of compression is to move blocks of signals alternatively. Even with the use of scalable video coding in (Kalva, 2006) to extend the capabilities of the H.264, there is still an issue of complexity due to the increased bitrate for efficiency compensation. The study in (Xu et al., 2018) hasadopted H.265 for video compression to mitigate this effect. However, there are still potential spatial and temporal redundancies, which cause poor

ISBN: 978-99949-0-888-2

video signal quality. In fact, it was concluded in (Ohm et al., 2012) that H.265 encoders, using objective and subjective metrics, perform similarly to H.264 encoders while saving 50 per cent of the bitrate`.

Aside these compression standards, other standards have been developed, such as the texture warping and synthesis developed in (Zhang & Bull, 2011) to transform original images. The authors developed the algorithm such that the whole images are not encoded in the first phase before motion estimation. A complex wavelet transform was used to segment the image texture region focusing on the spatial and temporal properties. The artifact video metric was used to evaluate the quality of the reconstructed images, with results showing a 60 per cent bitrate saving. Kahu & Bhurchandi, (2017) developed a differential directional filter bank coding scheme to compress videos in a sequential manner. Their proposed algorithm was able to decrease redundancies in the motion compensation stage while their implementation of the adaptive rood pattern search scheme reduces the encoding time.

In the quest to improve video quality, recent studies have developed deep machine learning techniques. The deep network architecture developed in (Ballé et al., 2015) made this possible. A parametric nonlinear transformation is introduced to normalize features from natural images. It was shown that the transformed images could be differentiable and inverted, making it possible to use probability density to filter out image noise. Most importantly, it can be shown that the transformation model can be optimized using the normalization objective. This effort makes it possible to implement the optimization of the deep network architecture. Lu et al. (2020) developed the first end-to-end video compression framework using pixel-based motion compensation. The framework combines the H.264 encoder and the neural network while implementing motion compensation, motion estimation, motion compression, residual compression, and bit rate estimation. Another work on deep learning compression is done in (Wu et al., 2020), where a generative adversarial network (GAN) was used to reconstruct video frames in a surveillance video streaming framework. The GAN uses the spatial and temporal discriminator to increase the similarity between an original and the reconstructed video. The GAN was able to reduce distortion rate under a low bit-rate streaming.

The review has shown a substantial amount of work done on video compression, especially using deep learning techniques to generate quality video frames while maintaining low complexity. A major highlight is a relationship between bitrate and video quality. Finding the "right" bitrate remains a challenge for deciding video quality. Nonetheless, there has been little research on applying deep learning in a video streaming space, especially in a dynamic adaptive streaming over HTTP (DASH)-based DVB setup. As a result, this study presents a DRL system for identifying the optimum bitrate for developing video quality and analyzing efficiency and performance. The throughput is quite important since we are dealing with real-time streaming, which requires minimal delay. This work's significant benefits are presented below:

- We develop a DRL system to figure out the more suitable video bitrate for video compression in an end-to-end DASH-based online streaming DVB setup.
- We introduce a multi-objective optimization that utilizes a weighted sum aggregate approach to estimate video quality. The video quality is perceived using the PSNR, and SSIM, and the delay time is observed using network parameters in the real-time video streaming setup.
- Experiment findings indicate that the system surpasses commonly deployed encoders such as H.264 and H.265 in terms of efficiency and performance. The framework is also configured on different image transformation algorithms.

The paper is well categorized in this manner: Unit II discusses the system model; Unit III discusses the problem formulation; Unit IV describes the working of our recommended framework; Unit V displays the investigational outcomes from the framework, and Unit VI reviews the study.

2. System Model

We begin by explaining the DVB setup based on a DASH protocol, illustrated in Fig. 1. Based on a content delivery network, the incoming video stream from the server is packetized as a bitstream in a content server that is concurrently transmitted to end customers through the DVB setup. An OTT-based DASH protocol receives the stream and disseminates it to various end users. This process is ideal until issues such as intermittent buffering, outrageous lagging, and obvious low content quality start to emerge. These issues reduce the QoE, thereby causing dissatisfaction from customers. To tackle this issue, we concentrate on the compression characteristics and network parameters. Right before the video signals are transmitted to the cloud, we construct an end-to-end framework that processes the incoming real-time video.

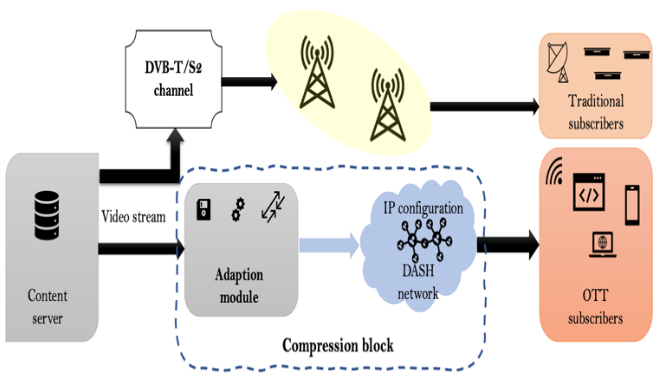

Fig 1. The DVB video streaming model.

ISBN: 978-99949-0-888-2

As demonstrated in Fig. 2, the end-to-end design incorporates encoding and decoding. The incoming stream going into the encoder is denoted as a video sequence, $V = \{f_1, f_2, \dots, f_{t-1}, f_t\} \in N$, where f_t represents a static frame of the incoming stream at time t and N denotes the overall amount of frames in the video stream. We extracted spatiotemporal features of the video stream using a recurrent neural network, yielding bitstreams $\breve{M}$ that represent the original images.

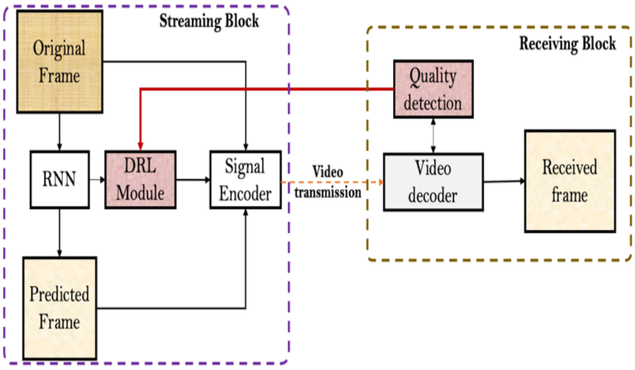

Fig 2. The DRL video compression framework.

The first step of the compression process is to quantize the video stream V. Knowing that the quantization process is not differential, thereby making end-to-end compression impossible, we draw inspiration from (Ballé et al., 2016), using the rounding operation to add uniformly distributed noise to the residual Spatio-temporal features of the incoming video stream. Therefore, having an output, $\hat{V}$. The next step is the entropy coding, where we used the arithmetic coding to encode $\widehat{m}$ and $\hat{y}$ into bits. The bitrate value is then employed to compute the encoded probabilistic model stream $\hat{y}$ and compute the entropy of the bit price. Finally, in the decoder block, the video stream with a static frame f_t is reconstructed into $\widehat{f_t}$.

3. Problem Formulation

Service providers are tasked to provide high-standard video streaming on customers' devices. However, video quality is perceived differently according to customers' bias. We will be using an objective approach to measure the video quality and quantify customers' QoE. The following metrics are adopted for this study.

A. Peak-Signal-to-Noise-Ratio

The PSNR metric is applied for computing the most attainable signal concerning the introduced noise in a reconstructed or compressed video (or a snapshot image in a video). The PSNR value quantifies the difference between the original and reconstructed video, and, since these values can have a wide range, they are counted in decibels (dB), expressed as (Huynh-Thu & Ghanbari, 2008).

$$PSNR = 10 \log_{10}\left(\frac{\mathcal{L}^2}{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[p_{mn} - q_{mn}]^2}\right) \quad (1)$$

where m and n represent the video frame dimension; p_{mn} and q_{mn} represent the original and reconstructed video respectively. Finally, $\mathcal{L}$ is the maximum dynamically possible pixel intensity of the image.

B. Structural Similarity Index Measure

An SSIM is considered a full reference technique for quantifying the current video frame based on the original, uncompressed video frame. This metric uses the structural properties of images for perceiving a similarity. The luminance, structure, and contrast similarity are taken for the original and compressed videos, using the circuitry from (Wang et al., 2004). The three measurements are combined using

$$SSIM_{p,q} = \frac{(2\mu_p\mu_q + g_1)(2\sigma_{pq} + g_2)}{(\mu_p^2 + \mu_q^2 + g_1)(\sigma_p^2 + \sigma_q^2 + g_2)}. \quad (2)$$

The μ_p and μ_q are the average values for the frame dimension; σ_p^2 and σ_q^2 are the variance of the dimension; σ_{pq} is the covariance of dimensions; while g_1 and g_2 are for stabilizing the denominator, computed as $0.01\mathcal{L}^2$ and $0.03\mathcal{L}^2$, respectively.

C. Time Delay

Our model deals with real-time video streaming, especially for live events. It is therefore important to measure the delay time between the sending and receiving time. Inspired by Huang et al., (2018), we used a delay gradient strategy to quantify the time transient across the two blocks, considering the mismatch in devices' clocks. The delay is expressed as

$$\mathcal{T} = \Delta F_t^R - F_t^S, \quad (3)$$

where ΔF_t^R is the change in sending time t for frame F, and F_t^S is the sending time t for frame F.

4. Reinforcement learning for video quality in DVB systems

Reinforcement learning (RL) techniques generally use value-based or policy-based approaches to learn random policies, either by directly choosing the best policy that relates to an action taken in a series of events or finding the temporal difference to quantify the future rewards, selecting the action that corresponds to the highest value function. Given the potential explosion of the state space when having continuous variables as in our case, bit rate, we use a deep neural network to represent state-actions pairs using weights. Firstly, we develop our problem using the Markov decision process (MDP).

ISBN: 978-99949-0-888-2

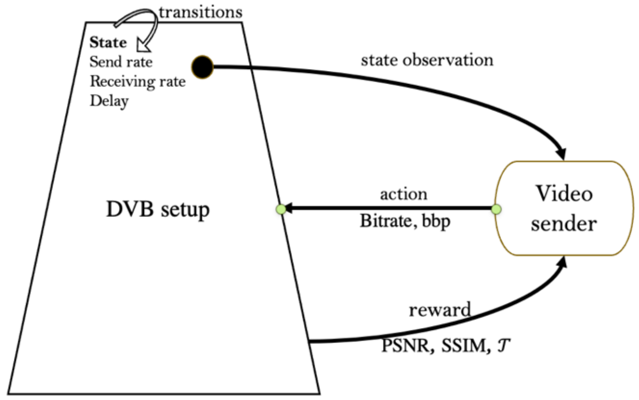

Fig 3. The reinforcement learning adoption for video quality.

D. *Markov Decision Process*

The problem is modeled as an MDP, using the notation, $M = \langle S, A, R, \gamma, P \rangle$ whereby S is the state of the environment, A represents the act by an agent in the environment, and R is the received reward after taking an action, used to determine if the quality of the action is taken according to the next stage of the environment. P is the probability distribution determining the transitions of states; and γ is the discount factor, used for leveraging the expected rewards to the present reward. Fig. 3 shows the application of MDP in our model.

State: Knowing that our environment is the DVB setup, we consider the state $S_t = \langle s, r, d, p, v \rangle$, whereby s and r are the sending and receiving video bitrate correspondingly, d is the delay between the sending and receiving time; p and v are the previous and predicted video stream quality respectively.

Action: The action taken by an agent is the sending bitrate selection at time t. Remembering the relationship between the bitrate and video quality, we aim to learn the optimal video bitrate to produce good video quality.

State Transition: This is the probability of taking an action A_t in state S_t; represented as $f(S_t, A_t)$.

Reward: When the agent takes the action A_t , it gets an immediate reward R_t. We aim to maximize QoE, we used a multi-objective-based reward framework, where three metrics from Section III, expressed as

$$R_t = w_1 PSNR + w_2 SSIM + w_3 \mathcal{T}, \qquad (4)$$

Where w_1, w_2, and w_3 are the weights assigned to each metric. The weight assignment represents the preference scale of a decision-maker.

This study presents the use of Q-learning methodology for solving the bitrate decision issue. The effect of the sender (the agent) taking an action A on state S, following a policy ω is computed using the sum of the discounted expected rewards, shown in (4),

$$Q^{\omega}(S_t, A_t) = \mathcal{E}^{\omega}\left[\sum_{k=0}^{K} \gamma^k R_{t+k}\right], \qquad (5)$$

where γ balances the short- and long-term reward over the expectation $\mathcal{E}$. The optimal policy, ω^* which maximizes the QoE is expressed as

$$Q^*(S_t, A_t) = \max_{\omega} Q^{\omega}(S_t, A_t). \qquad (6)$$

We update the state-action pairs, $Q_t\ (S_t, A_t)$ by using the bellman's equation, described in (7). The next Q-value is determined by the old Q-value and the probability to move to the next state, learned by the α parameter. The new (or next) Q-value is learned as

$$\begin{aligned} Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) \\ + \alpha_t[Q_t(S_{t+1}, A_{t+1}) \\ - Q_t(S_t, A_t)]. \end{aligned} \qquad (7)$$

The Q-learning approach uses a look-up table to estimate the action-value pairs, but our state space consists of continuous variables. The continuous variables cause a large Q-table which could make the problem intractable. We address this problem by implementing a deep neural network (DNN) to appraise the action-value utility.

The traditional Q-learning algorithm will take the states of the DVB setup and feed them into a deep Q-network. We dub the algorithm as deep reinforcement learning video compression (DRL-VC). Algorithm 1 shows the DRL-VC's training process. The algorithm starts with the initialization of the DQN and target network parameters, θ and $\breve{\theta}$. Given several epochs, a **for** loop is created to update the DQN parameters based on the observed reward at each episode.

Algorithm 1 DQN Training process

Initialize the DQN and target network parameters; $\theta, \breve{\theta}$, episodes, $\mathcal{E}$
for all $\mathcal{E}$ **do**
  obtain state observation, using state, S_t
  for each time t
    select an action vector, A_t using a random parameter
    perform action A_t and receive the reward R_t
    transition to the next state, S_{t+1} using $f(S_t, A_t)$
    store the experience as a vector; (S_t, A_t, R_t, S_{t+1}) in a buffer, $\mathcal{D}$
    sample a batch of U samples from buffer $\mathcal{D} = (S_t, A_t, R_t, S_{t+1})_t^U$
    calculate $\breve{y}$ using the Q-value in (5) from the DQN target

$$\breve{y}_t = R_t + \gamma \max_{A_t} Q(S_{t+1}, A_{t+1} | \theta)$$

    Update the DQN's weight from the main network

$$L_\theta = \frac{1}{U} \sum_t \left(\breve{y}_t - Q(S_t, A_t;\ \theta_t)\right)^2$$

    Update the DQN parameters using a gradient approach $\theta \leftarrow \theta - \sigma \nabla L_\theta$
  end for
end for

ISBN: 978-99949-0-888-2

5. Results and Discussion

We discuss the evaluation of our proposed real-time video streaming compression scheme. The performance is evaluated using the KonVid test video database, streamed at 30, 40, and 60 frames per second at a varying time duration. The DVB setup was later analyzed with the proposed framework using live videos. The PSNR and SSIM values were compared to the bits per pixel (BPP) settings using each adopted video database. Fig. 4 shows the video quality using the PSNR for the average BPP, comparing our proposed DRL-VC to other compression schemes. It is observed that there the DRL-VC performs better than the benchmarked schemes, especially in the range of 0.15 to 0.2 bpp, our scheme performs significantly better than the H.264 and H.265 encoders, while slightly surpassing the DVC scheme by some margin.

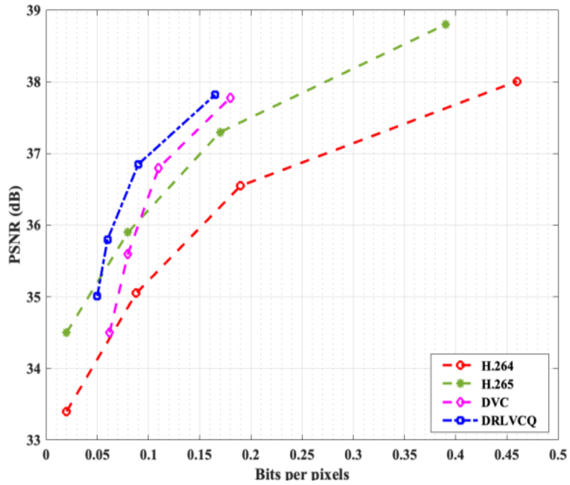

Fig 4. Video quality assessment for KonVid database using PSNR values

In Fig. 5, it is seen that the DRL-VC also, by some margin, outperforms the DVC, but is significantly better than other schemes. It is worth noting that the DRL-VC has a higher average performance than the DVC with the PSNR and SSIM metrics.

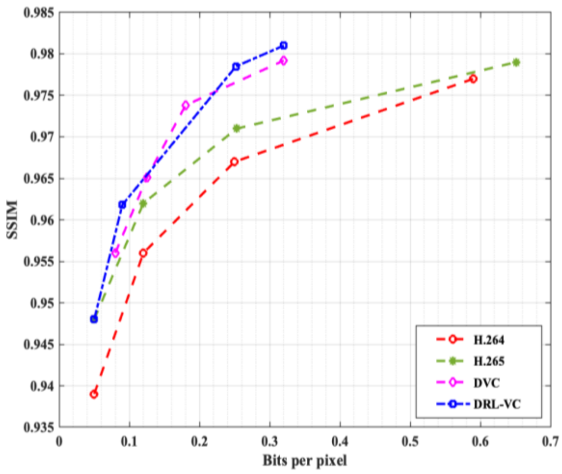

Fig 5. Video quality assessment for KonVid database using SSIM values

An ablation study is carried out on our scheme, starting with the impact of decision-making preference. The BDBR and BD-PSNR were adopted to assess video compression schemes' performance. BDBR is the bit rate savings percentage using a baseline compression scheme at the same PSNR, while the BD-PSNR depicts the performance gain using a baseline compression scheme at the same bit rate. The weight for each metric is varied to increase the preference rate and we pegged the favored metric to have 50 per cent weighting, meaning that the 0.5 is assigned to the metric of interest, while the remaining 0.5 is split between the remaining metrics. It is to note that the initial weighting system assigns equal weights to all metrics. Table I shows the results for each metric. It is seen that the best values are obtained when the SSIM is favored. The SSIM-focused scheme has a bit savings of 21.2 per cent while the PSNR-focused scheme can only produce a saving of 17.5 per cent. This outcome conforms to works in literature that report the significance of the SSIM metric in assessing video quality and its superiority over the PSNR. It is seen that focusing on the delay metric will not necessarily improve video quality; hence the poor values.

TABLE I. COMPARISON OF DIFFERENTLY WEIGHTED METRICS USING THE BDNR AND THE BD-PSNR

Weighted metric		Vid_Str1	Vid_Str2	Vid_Str3	Vid_Str4
PSNR (50%)	**BDBR (%)**	-25.51	-17.24	-18.16	-32.02
	BD-PSNR	0.75	0.51	0.64	1.12
SSIM (50%)	**BDBR (%)**	-28.59	-23.33	-19.27	-33.61
	BD-PSNR	1.22	0.66	0.92	1.42
DELAY (50%)	**BDBR (%)**	-21.02	-17.08	-13.85	--25.05
	BD-PSNR	0.71	0.37	-0.52	0.89

Using different network characteristics, we analyze the delay time using three configurations of our proposed compression scheme. We separately integrate three encoders in our scheme and assess the effect on the delay time. DRL-VC-1, DRL-VC-2, and DRL-VC-3 utilized the Arithmetic-based X.264, Hyperprior-based H.265, and the RNN encoder respectively. From Fig. 6, it is observed that the other encoders use a more complex process, delaying the video transmission. The RNN proves to be less complex.

ISBN: 978-99949-0-888-2

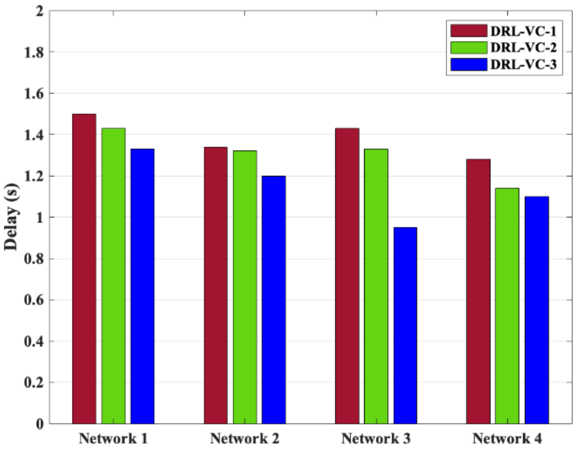

Fig 6. Delay time for different encoder-based configurations.

To further validate our proposed scheme, we ran an error test across different video streams using the root mean square error (RMSE). An RMSE can measure the correlation of errors across multiple outputs and explain the hidden variance across a dataset. From Fig. 7, the RMSE is plotted across different video streams. The video streams are characterized by their frame rates and dimensions. Knowing that the lower RMSE values indicate better and, with bright yellow being the lowest in the figure, it is seen that our framework performs significantly well across different video streams.

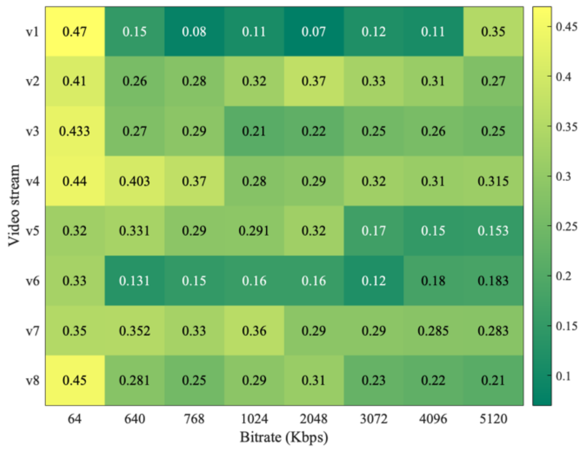

Fig 7. RMSE values for different video streams.

Finally, we analyze the multi-objective aspect of our proposed scheme. Pareto optimal solutions were generated using a variety of weights. Fig. 8 shows the Pareto front for PSNR and SSIM. The circled data point shows the compromise solution from other potential solutions. The solution shows an SSIM value of 0.9702 and a PSNR value of 37.92 dB.

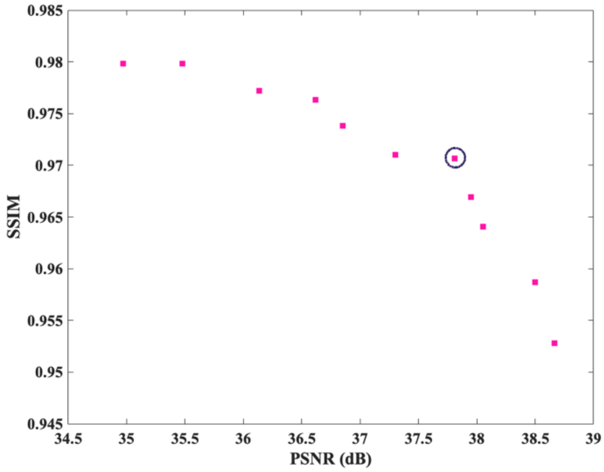

Fig 8. Pareto front for PSNR ans SSIM.

7. Conclusion

In this study, we presented a novel multi-objective deep reinforcement learning video compression framework for real-time applications in a digital video broadcasting setup. The study formulates a multi-objective problem, considering PSNR, SSIM, and delay time. We developed our scheme to start with an RNN-based encoder, followed by the DRL. The DRL was able to learn the optimal bitrate for producing video quality, considering network variability. The proposed scheme proves to perform better than benchmark schemes, using the PSNR and SSIM metrics. The bitrate savings is increased by a large margin, especially when the SSIM is given the most preference. The scheme was also validated using the RMSE to measure the error rate among different video streams, which prove effective with a minimal deviation across all streams. Using the weighting method, we were able to generate Pareto optimal solutions for the PSNR and SSIM. For further studies, we will introduce a technique for selecting actions in the DRL framework, hoping to generate better policies for video compression. Furthermore, the motion compensation part of the compression framework will be studied. We will also include the effect of heterogenous devices in the compression framework, making it more robust for implementation.

8. References

Ballé, J., Laparra, V., & Simoncelli, E. P. (2015). Density modeling of images using a generalized normalization transformation. arXiv preprint arXiv:1511.06281.

Ballé, J., Laparra, V., & Simoncelli, E. P. (2016). End-to-end optimized image compression. arXiv preprint arXiv:1611.01704.

Ekmekcioglu, E., Gurler, C. G., Kondoz, A., & Tekalp, A. M. (2016). Adaptive multiview video delivery using hybrid networking. IEEE Transactions on Circuits and Systems for Video Technology, 27(6), 1313-1325.

ISBN: 978-99949-0-888-2

Huang, T., Zhang, R. X., Zhou, C., & Sun, L. (2018, June). Delay-constrained rate control for real-time video streaming with bounded neural network. In Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video (pp. 13-18).

Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of validity of PSNR in image/video quality assessment. Electronics letters, 44(13), 800-801.

Kahu, S. Y., & Bhurchandi, K. M. (2017). A low-complexity, sequential video compression scheme using frame differential directional filter bank decomposition in CIE la* b* color space. IEEE Access, 5, 14914-14929.

Kalva, H. (2006). The H. 264 video coding standard. IEEE multimedia, 13(4), 86-90.

Lu, G., Zhang, X., Ouyang, W., Chen, L., Gao, Z., & Xu, D. (2020). An end-to-end learning framework for video compression. IEEE transactions on pattern analysis and machine intelligence, 43(10), 3292-3308.

Ohm, J. R., Sullivan, G. J., Schwarz, H., Tan, T. K., & Wiegand, T. (2012). Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). IEEE Transactions on circuits and systems for video technology, 22(12), 1669-1684.

Pocovi, G., Soret, B., Pedersen, K. I., & Mogensen, P. (2017, May). MAC layer enhancements for ultra-reliable low-latency communications in cellular networks. In 2017 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1005-1010). IEEE.

Santhi, K. R., Srivastava, V. K., SenthilKumaran, G., & Butare, A. (2003, October). Goals of true broad band's wireless next wave (4G-5G). In 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484) (Vol. 4, pp. 2317-2321). IEEE.

Schwarz, H., Marpe, D., & Wiegand, T. (2007). Overview of the scalable video coding extension of the H. 264/AVC standard. IEEE Transactions on circuits and systems for video technology, 17(9), 1103-1120.

Tse, D., & Viswanath, P. (2005). Fundamentals of wireless communication. Cambridge university press.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. IEEE Transactions on circuits and systems for video technology, 13(7), 560-576.

Wu, Y., He, T., & Chen, Z. (2020, October). Memorize, Then Recall: A Generative Framework for Low Bit-rate Surveillance Video Compression. In 2020 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

Xu, J., Zhou, B., Zhang, C., Ke, N., Jin, W., & Hao, S. (2018, September). The impact of bitrate and GOP pattern on the video quality of H. 265/HEVC compression standard. In 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC) (pp. 1-5). IEEE.

Zhang, F., & Bull, D. R. (2011). A parametric framework for video compression using region-based texture models. IEEE Journal of Selected Topics in Signal Processing, 5(7), 1378-1392.