



# Video coding deep learning-based modeling for long life video streaming over next network generation

Mohammad Alsmirat<sup>1,2</sup> · Yousef Sharrab<sup>3,4</sup> · Monther Tarawneh<sup>3</sup> · Sana'a Al-shboul<sup>5</sup> · Nabil Sarhan<sup>4</sup>

Received: 14 April 2022 / Revised: 6 December 2022 / Accepted: 9 December 2022 / Published online: 3 January 2023  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Availability is one of the primary goals of smart networks, especially, if the network is under heavy video streaming traffic. In this paper, we propose a deep learning based methodology to enhance availability of video streaming systems by developing a prediction model for video streaming quality, required power consumption, and required bandwidth based on video codec parameters. The H.264/AVC codec, which is one of the most popular codecs used in video steaming and conferencing communications, is chosen as a case study in this paper. We model the predicted consumed power, the predicted perceived video quality, and the predicted required bandwidth for the video codec based on video resolution and quantization parameters. We train, validate, and test the developed models through extensive experiments using several video contents. Results show that an accurate model can be built for the needed purpose and the video streaming quality, required power consumption, and required bandwidth can be predicted accurately which can be utilized to enhance network availability in a cooperative environment.

**Keywords** Video communication · Video streaming · Perceptual video quality modeling · Encoding power consumption modeling · Video communication systems · Machine learning · Deep learning · Artificial neural networks

## 1 Introduction

Video streaming has become increasingly more popular, especially after the recent COVID-19 epidemic, due to the rising needs for distance learning, working from home, business and government virtual meetings, and one-to-one video calling. Moreover, video surveillance systems are mostly relying on digital communication nowadays and this communication may share other communication types as what happen in home networks. Video data generally consumes extremely high network and servers bandwidth. This situation become worst when multiple parties, who share the same network, need to use video communication at the same time, which may over-saturate the network and may cause interruption in the communication over the network. Moreover, some devices are battery powered which introduces the need for power management to guarantee the longest availability of the video streaming systems. System availability is very important for smart networks services and one of the three main goals of cyber security [1, 2]. To prevent these problems, next generation

---

✉ Mohammad Alsmirat  
malsmirat@sharjah.ac.ae; masmirat@just.edu.jo  
Yousef Sharrab  
sharrab@iu.edu.jo; yousef.sharrab@wayne.edu  
Monther Tarawneh  
Mtarawneh@iu.edu.jo  
Sana'a Al-shboul  
smalshboul16@cit.just.edu.jo  
Nabil Sarhan  
nabil.sarhan@wayne.edu

<sup>1</sup> Computer Science Department, University of Sharjah, Sharjah, UAE  
<sup>2</sup> Computer Science Department, Jordan University of Science and Technology, Irbid, Jordan  
<sup>3</sup> Faculty of Information Technology, Isra University, Amman, Jordan  
<sup>4</sup> Deep Learning Lab, ECE Department, Wayne State University, Detroit, MI, USA  
<sup>5</sup> Computer Engineering Department, Jordan University of Science and Technology, Irbid, Jordan

networks and smart video sources that may utilize machine learning are required [3–5].

Video compression is generally incorporated with video streaming to reduce the required streaming transmission bandwidth. Video compression relies on similarity (spatial and temporal) in video frames and it only codes the differences within a frame (spatial) and/or between consecutive frames (temporal). This encoding process consumes the most energy compared to all other phases of transmissions, such as capturing, wired or wireless transmission, decoding, up-scaling, and displaying [6–8]. Video compression can ideally provide any needed output video playback rate. One of the most popular encoders in video communication is H.264, which employs many features for more efficient compression and better flexibility [9, 10]. In this paper, we present H.264 as a case study.

An adaptive video communication system can achieve network availability by adapting user video transmission rate [11]. Many things should be done for such systems to work correctly. One thing is to measure network congestion (network capacity) which help in specifying the possible rate of the video stream that is being transmitted. Another thing is to force the video compressor to produce the suitable video rate to guarantee smooth video transmission and playback. Device energy consumption should be also taken into consideration and should be reduced to maximize the overall system availability especially if some devices are battery powered. This can be considered as a control problem with two inputs: current device energy level and network congestion level, and one output which is the video transmission rate. Video transmission rate can be then fed to the video compressor to compress the video with the needed rate. If all video sources apply this optimization continuously, system availability is guaranteed. We can look at the problem from a different perspective in which we want to predict the availability of the streaming system. In this perspective, the input would be some encoder parameters and the output would be the power needed for the encoding process beside the resulting rate of the video stream. In this study, we are interested in the later perspective and we build the prediction system using deep learning. In previous studies [11–14], researchers tried to provide solutions for adaptive video streaming. But most solutions were only limited to specific types of networks or they did not take into consideration all related system's parameters. To be more specific, only few studies [6, 7, 15] tried to model the impact of video encoders related parameters on the power consumption and the resulting bit-rate of the encoder. And according to our knowledge, none of the studies used deep learning in the modeling process.

This paper contributes to the Video communication systems' availability by modeling the relationship between

the systems' availability related parameters and the video encoders' parameters. In specific, we study the relationship between (the video streaming related power consumption, perceptual video quality at the destination, and the required communication bandwidth) and the video encoder resolution and quantization parameters. We study and analyze the predictability of the video communication related power consumption, perceptual video quality at the destination, and the required communication bandwidth based on the encoder resolution and quantization parameters using deep learning algorithms. All data needed have been collected in a lab environment as described in Sect. 3. The developed deep learning models are based on more than 1, 000 different experiments of video encoding and simultaneously measuring consumed power. The perceptual video quality and required bandwidth are measured at the end of the encoding process. The developed models can be used to assess the impacts of various video encoding parameters, and thus can help in the dynamic control of various source settings, including but not limited to resolution and quantization, to achieve the optimal power consumption, bandwidth, video quality, and computer vision accuracy in robotics [11–14]. Although we consider the popular H.264 standard, this study can help in deriving models for other encoders, such as *High Efficiency Video Encoding* (HEVC) and VP9, by following the same procedure. The results show that the proposed models are very accurate in predicting the video communication related power consumption, perceptual video quality at the destination, and the required communication bandwidth based on the encoder resolution and quantization parameters.

The rest of the paper is organized as follows. Section 2 discusses background information and related work. Section 3 develops various models and discusses the experimental setups and modeling methodology. Section 4 presents the validation results and provides an overall analysis. Finally, conclusions are drawn in the last section.

## 2 Background information and related work

A video is a sequence of frames (images) captured at a rate equal or faster than the eye perception (frame rate). Consecutive frames are usually very similar to each other and the similarity is called temporal redundancy. Each frame consists of number of pixels and the number of pixels in a frame is known as frame spatial resolution. Higher-resolution indicates higher clarity. A video file in the original format (storing all pixel information) would be tremendous in size, which requires very high bandwidth and storage. Encoding techniques compress these large video files into much smaller files that could be transmitted without noticeable loss or delay. This, along with the availability of

high-speed communication networks facilitated video communication as a new means of communication.

The most popular video encoder nowadays is H.264, which implement both spatial and temporal compression. H.264 has high computational complexity, mainly due to its motion estimation, complex prediction, and rate distortion optimization [7, 10, 15–17].

The video encoding process can generally be divided into the following three high-level stages: *Intra- and Inter-Prediction (Estimation) Stage*, *Transformation, Quantization and Their Inverse Stage*, and *Entropy Coding Stage*. In the estimation stage, both intra-prediction and inter-prediction are used to reduce the spatial and temporal redundancies in the video, respectively [7, 17, 18]. Streaming video quality depends on video encoding process and the amount of bandwidth required for it to be viewed properly. Video data contains spatial and temporal redundancy. Similarities can thus be encoded by just considering differences within a frame (spatial), and/or between frames (temporal). Intra-prediction utilizes spatial correlation in each frame to reduce the amount of data necessary to represent the image.

Some of the common forms of video communication are video calling and video conferencing. Video conferencing technology has enabled us to seamlessly connect face-to-face with people all over the world. Video conferencing technology has revolutionized education and schools by bringing the best learning opportunities from around the world to classrooms and homes. Video Communication process is shown in Fig. 1,

Power consumption, bandwidth requirements, and video quality are major concerns in video communication systems especially in smart systems that utilize mobile devices and wireless communications. In this paper, we develop deep learning-based predictive models for power consumption, required video bandwidth, and perceived video

quality in a typical video communication system. The model capture the impacts of various video encoding parameters, and thus can assists in the dynamic control of various camera/sensor settings to achieve the optimal power consumption, bandwidth, and video quality.

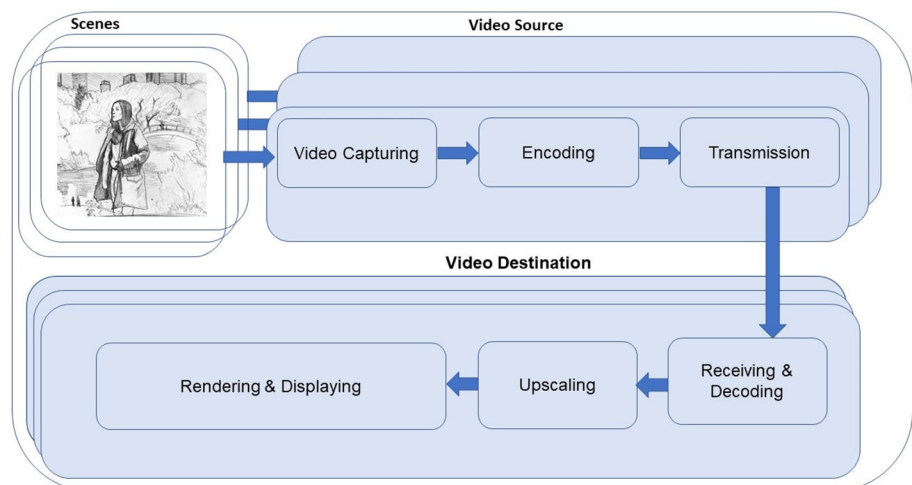
### 3 Building the model and experimental setups

Since 2012, Deep Learning (DL) has witnessed great success in computer vision and other disciplines such as speech recognition, natural language processing and modeling [19]. The success of DL is based on the recent availability of big data, high computational power, and the utilization of the powerful Artificial Neural Network (ANN) algorithms. A popular example of Deep Learning networks is the feed-forward Deep Network or Multi-Layer perceptron (MLP).

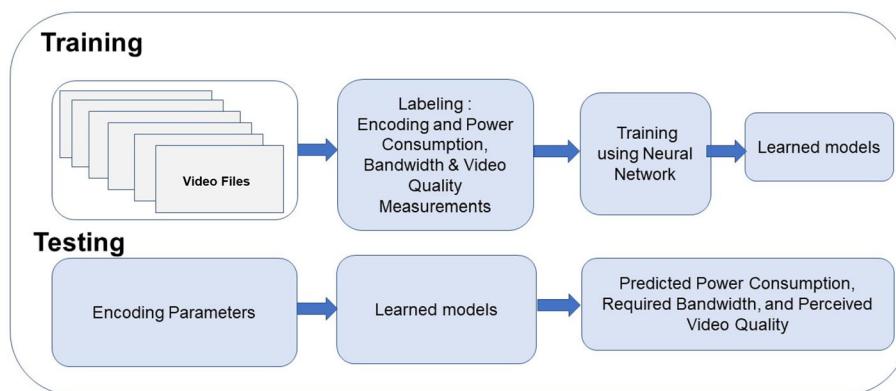
These models use multiple layers, one input layer that represents the input independent vector, one or more hidden layers, and one output layer that represent the output dependent vector. Nodes in each layer are linked to all nodes in the next layer. These links hold the intermediate weights. The weight on each link defines how a first layer influences the second layer and so on. During the training process, the weights are updated in each iteration in order to obtain the lowest error in the output. Gradient descent back-propagation is often used as the learning algorithm to minimize the output error. Calculating the optimal weights is the main issue of deep learning and machine learning training process. Nonlinearities are represented in the neural network by activation functions in each node such as ReLu and Softmax [20].

In this paper, a deep learning approach, mainly supervised learning of artificial neural networks, is proposed to

**Fig. 1** The process of video communication



**Fig. 2** Illustration of the deep learning process



**Table 1** Properties of the standard dataset used in the training

Video segment	Time duration (s)	Frame size in pixels	Number of frames
Akiyo	10	352 × 240 (CIF)	300
Silent	10	352 × 240 (CIF)	300
SignIrene	18	352 × 240 (CIF)	540
Deadline	45.8	352 × 240 (CIF)	1374

develop prediction models for video encoding quality and required resources based on input encoder parameters. Regression by artificial neural networks is chosen for its high accuracy in prediction based on labeled data.

Figure 2 shows the deep learning process we followed in this paper. Raw video sequences are collected and encoded using different resolutions and quantization parameters. During and after encoding, each encoded video sequence is labeled by the required power, required bandwidth, and the resulted quality of the sequence. After that, the sequences were divided into training and testing subsets. A deep neural network model is then developed and trained on the training subset of the sequences. Upon completing the training, the trained model is tested on the unseen test subset of the sequences to predict the video encoding power consumption, the required bandwidth, and the perceptual video quality for these videos. The predicted data is then compared with the actual data to mature the model accuracy.

We encoded sequences by changing both the resolution and the quantization parameter and we measure encoding power consumption, resulting bit-rate, and the perceived video quality. The utilized sequences are the popular Akiyo, Silent, SignIrene, and Deadline video sequences downloaded from VIPs Lab<sup>1</sup>. More information about these sequences are listed in Table 1. We use these sequences because they are available in raw (uncompressed) format. We change the resolution by down-scaling each video sequence from the original size of (352 × 240) down to 10% of that or (35 × 24) (specifically, we consider the

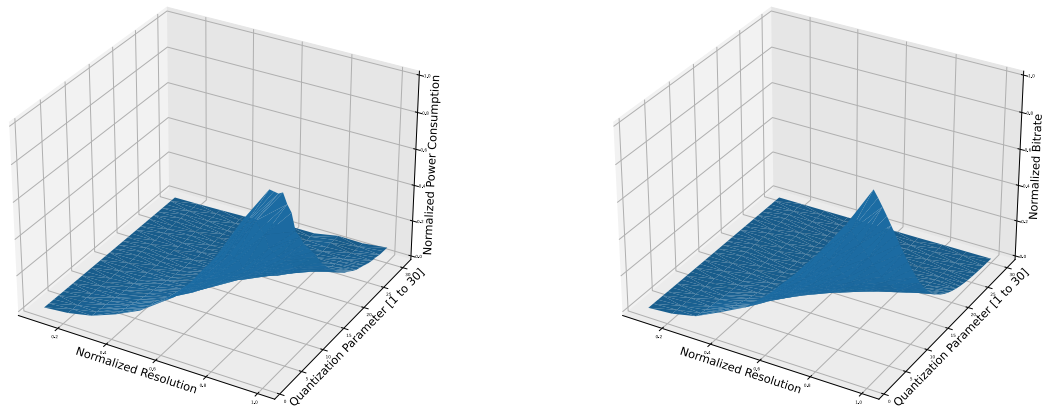
original itself 100% and (90%, 80%, ..., and 10% of the original size). For each of these sizes, we also produce different video quality levels using the quantization parameter (from 1 to 30) of the encoder.

We measure the power consumption while encoding as explained in [8] and then we find the bit-rate of the encoded video from the encoded video file size. We use the decoded video frames to measure the perceived video quality compared to the original video frames. As a metric for perceived video quality, we utilize *Structural SIMilarity Index* (SSIM) [21] between decoded and the original frames.

We prepare a table that contains two input columns: a combination of the percentage of downsizing the frames (1.0, 0.9, 0.8, ....., 0.2, 0.1) and the quantization parameter (1,2,3,....., 29,30), and three output columns: the power consumption, the bit-rate, and the SSIM quality. Figure 3a–c plots the collected data.

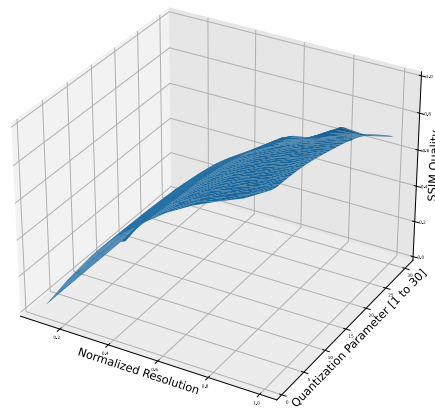
Figure 3a, b, and c represent the impact of varying the video encoding parameters (quantization parameter and spatial resolution) on the encoder power consumption, required video transmission bandwidth, and encoded video quality respectively. By combining the use of the quantization parameter and frame size in pixels, in H.264 encoding, the required bandwidth can be reduced to as low as one percent of the bandwidth at the highest setting and the encoder power consumption can be decreased to as low as 1/25 of the power consumption with full resolution at quantization parameter of 1. Also, we can see from the figures that determining the output combination of power consumption, bitrate, and SSIM for a specific combination

<sup>1</sup> <https://see.xidian.edu.cn/vips/index.html>.



(a) encoder power consumption

(b) required video transmission bandwidth (Bitrate)



(c) encoded video quality (SSIM)

**Fig. 3** Impact of varying quantization parameter and spatial resolution**Fig. 4** The ANN model code

```
# Defines "deep" model and its structure
model = Sequential()
model.add(Dense(7, input_shape=(2,), activation='relu'))
model.add(Dense(7, activation='relu'))
model.add(Dense(7, activation='relu'))
model.add(Dense(7, activation='relu'))
model.add(Dense(7, activation='relu'))
model.add(Dense(3,))
model.compile(Adam(lr=0.003), 'mean_squared_error')

# Pass several parameters to 'EarlyStopping' function and assigns it to 'earlystopper'
earlystopper = EarlyStopping(monitor='val_loss', min_delta=0, patience=15, verbose=1, mode='auto')

# Fits model over 2000 iterations with 'earlystopper' callback, and assigns it to history
history = model.fit(X_train, y_train, epochs = 2000, validation_split = 0.2, shuffle = True, verbose = 0,
                    callbacks = [earlystopper])

# Plots 'history'
history_dict=history.history
loss_values = history_dict['loss']
val_loss_values=history_dict['val_loss']
plt.plot(loss_values, 'bo', label='training loss')
plt.plot(val_loss_values, 'r', label='training loss val')

# Runs model with its current weights on the training and testing data
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Calculates and prints r2 score of training and testing data
print("The R2 score on the Train set is:\t{:0.3f}".format(r2_score(y_train, y_train_pred)))
print("The R2 score on the Test set is:\t{:0.3f}".format(r2_score(y_test, y_test_pred)))
```

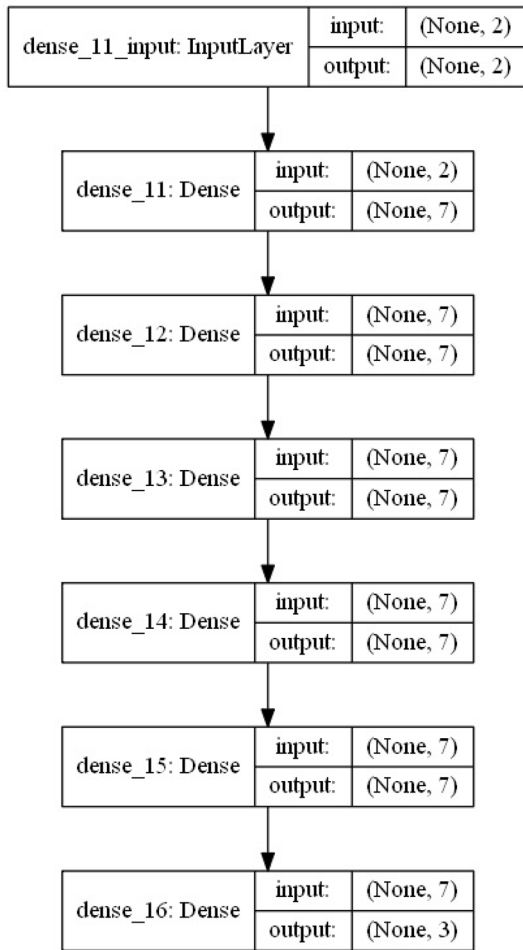


Fig. 5 ANN model structure

of resolution and quantization parameters is not an easy task and requires a sophisticated prediction algorithm.

To build the ANN model, we use Keras on top of TensorFlow and Python environment. Figure 4 shows the code for the developed ANN model. Utilizing Keras/

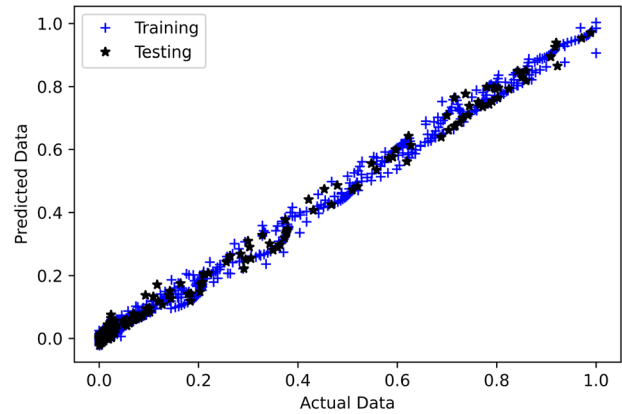
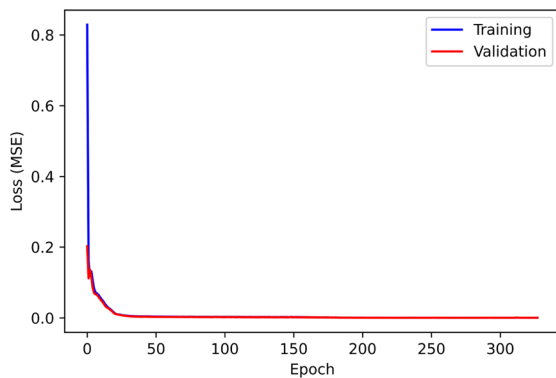


Fig. 7 Overall actual vs predicted for the testing dataset

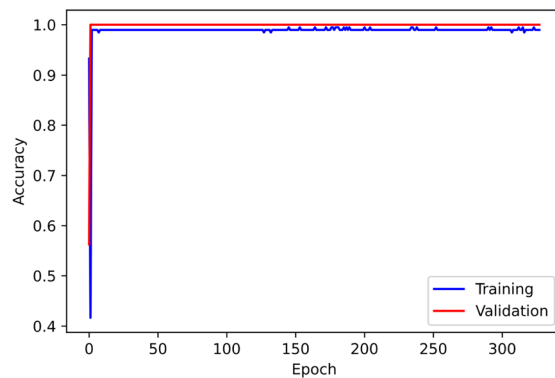
TensorFlow, the code define the network to be sequential and define the number of nodes in the input, the output and the hidden layers. Adam optimizer is selected to minimize the mean squared error (MSE)

between the actual and predicted values. The “Early-stop” function is used to stop training automatically. During training, the MSE is measured at end of each epoch and if the loss is no longer decreasing by a value more than *min\_delta* for *patience* consecutive epochs, the training terminates.

With “Earllystop”, the direction (weather to stop when monitored quantity decreasing or increasing) is automatically inferred from the name (type) of the monitored quantity (loss should decrease and accuracy should increase). Training is invoked by *model.fit()* function. We set the maximum number of epochs in the training process to be 2000 if early stopping function did not work. The fit function splits the training dataset randomly into 20% for validation and 80% for training. Finally, we measure the goodness of the fit by  $R^2$ , where if  $R^2 = 1$  indicates a perfect fit.



(a) Loss Function (MSE)



(b) Accuracy

Fig. 6 ANN model training monitoring

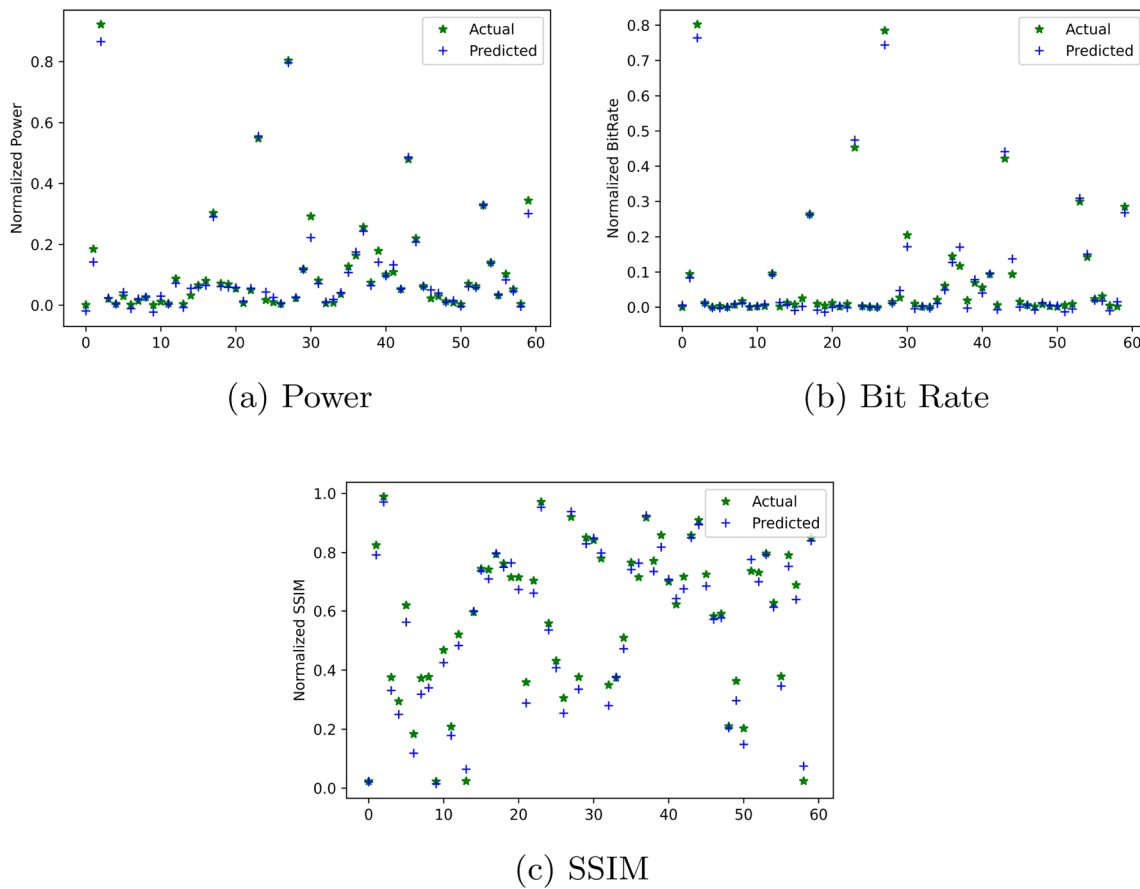


Fig. 8 Actual vs predicted for each output in the testing dataset

Table 2 R2 results

Data	R2
Power	0.989
BitRate	0.990
SSIM	0.981
Overall testing	0.987
Overall training	0.990

Figure 5 shows the neural network model, with inputs being video resolution as a percentage of the original frame size and the quantization parameter. The network has four hidden layers and three output layers: video power consumption, required bandwidth, and perceptual video quality (measured in SSIM).

### 4 Model validation results and analysis

Figure 6a shows the progress of the cost function (Validation loss) used in the training of the ANN model. In this paper, we use Mean Squared Error (MSE) which is the average (mean) loss over the entire training dataset of the

squared of the difference between the actual and the predicted values. From the figure, we can see that the MSE is close to zero after training the model for 292 epochs.

Figure 6b shows the progress of the accuracy of the model over the training and the validation data. The figure shows quick convergence of the training process without showing any overfitting.

Figure 7 shows the overall relation between the actual data and the predicted data from the neural network model. Since the prediction has high accuracy, the relation is linear with actual and predicted data are very close to each other.

To details the results shown in Figs. 7, 8 plots that actual data and the predicted data for each point for each output in the testing dataset. Since the prediction has high accuracy, the values are very close to each other for all outputs.

To provide numerical values that represent the closeness of the predicted data and the actual data plotted in the previous figure, We calculated  $R^2$  for each output separately and for all outputs together for both the training and testing data. The results are shown in Table 2. As shown in the table,  $R^2$  is close to 1 for both the training and the testing data for all outputs.

## 5 Conclusions and future work

We have developed a deep learning-based power consumption, required bandwidth, and SSIM quality prediction model for a video communication system. The model can help in the automatic control of various camera/sensor parameters setting, including frame size and quantization, to achieve the optimal outcomes in terms of video perceived quality, encoding power consumption, and the required bandwidth. By combining H.264 encoding quantization parameter and frame size settings, the required bandwidth can be reduced to as low as one percent of the original requirements and the power consumption can be lowered to as low as 1/25 of the full resolution at quantization parameter of 1. We notice that the neural network model is very accurate. It achieves prediction with high similarity to the original labeled data as measured by R squared goodness of fit measure. In future work, we will build a neural network model that predict the required resolution and quantization parameter (output) based on the encoding power consumption, required bandwidth, and the video quality (input). We will also experiment with higher resolution videos.

**Author contributions** MA is the main author and he has finalized the writing, model building and implementing, and the experiments. YS provided the data, and initialize the writing, model building and implementation, and experiment design and implementation. MT helped in the writing. SA helped in the implementation of the model and the generation of the figures. NS was the PhD supervisor of YS when YS started collecting the data.

**Funding** This work is not funded.

**Data availability** Some of the data used in this research is publicly available as referenced within this paper. Newly generated data are not public as it is still under study by the authors.

## Declarations

**Competing interests** The authors have not disclosed any competing interests.

**Informed consent** All authors are informed about this submission.

## References

- Satam, S., Satam, P., Pacheco, J., Hariri, S.: Security framework for smart cyber infrastructure. *Clust. Comput.* **25**(4), 2767–2778 (2022). <https://doi.org/10.1007/s10586-021-03482-2>
- Tiefenau, C., Häring, M., Krombholz, K., von Zezschwitz, E.: Security, availability, and multiple information sources: exploring update behavior of system administrators. In: Sixteenth Symposium on Usable Privacy and Security ({SOUPS} 2020), pp. 239–258 (2020)
- Jedari, B., Prensankar, G., Illahi, G., Francesco, M.D., Mehrabi, A., Ylä-Jääski, A.: Video caching, analytics, and delivery at the wireless edge: a survey and future directions. *IEEE Commun. Surv. Tutor.* **23**(1), 431–471 (2021). <https://doi.org/10.1109/COMST.2020.3035427>
- Luo, J., Yu, F.R., Chen, Q., Tang, L.: Adaptive video streaming with edge caching and video transcoding over software-defined mobile networks: A deep reinforcement learning approach. *IEEE Trans. Wireless Commun.* **19**(3), 1577–1592 (2020). <https://doi.org/10.1109/TWC.2019.2955129>
- Saltarin, J., Bourtsoulatze, E., Thomos, N., Braun, T.: Adaptive video streaming with network coding enabled named data networking. *IEEE Trans. Multimed.* **19**(10), 2182–2196 (2017). <https://doi.org/10.1109/TMM.2017.2737950>
- Sharrab, Y., Alsmadi, I., Sarhan, N.: Towards the availability of video communication in artificial intelligence-based computer vision systems utilizing a multi-objective function. *Clust. Comput.* **25**, 1–17 (2022). <https://doi.org/10.1007/s10586-021-03391-4>
- Sharrab, Y.O., Sarhan, N.J.: Aggregate power consumption modeling of live video streaming systems. In: Proceedings of the 4th ACM Multimedia Systems Conference, pp. 60–71 (2013)
- Sharrab, Y.O., Sarhan, N.J.: Modeling and analysis of power consumption in live video streaming systems. *ACM Trans. Multimed. Comput. Commun. Appl.* **13**(4), 54–15425 (2017). <https://doi.org/10.1145/3115505>
- Agrawal, P., Zabrovskiy, A., Ilangovan, A., Timmerer, C., Prodan, R.: Fasthttps: fast approach for video transcoding time prediction and scheduling for http adaptive streaming videos. *Clust. Comput.* **24**, 1–17 (2021). <https://doi.org/10.1007/s10586-020-03207-x>
- Sharrab, Y., Alsmirat, M., Hawashin, B., Sarhan, N.: Machine learning-based energy consumption modeling and comparison of h.264/avc and google vp8 encoders. *Int. J. Electr. Comput. Eng.* **11**(2), 303–310 (2021)
- Alsmirat, M., Sarhan, N.J.: Intelligent optimization for automated video surveillance at the edge: a cross-layer approach. *Simul. Model. Pract. Theory* **105**, 102171 (2020). <https://doi.org/10.1016/j.simpat.2020.102171>
- Alsmirat, M.A., Sarhan, N.J.: Cross-layer optimization for many-to-one wireless video streaming systems. *Multimed. Tools Appl.* **77**(19), 24789–24811 (2018). <https://doi.org/10.1007/s11042-018-5698-x>
- Alsmirat, M., Sarhan, N.J.: Cross-layer optimization for automated video surveillance. In: 2016 IEEE International Symposium on Multimedia (ISM), pp. 243–246 (2016). <https://doi.org/10.1109/ISM.2016.0055>
- Alsmirat, M.A., Sarhan, N.J.: Cross-layer optimization and effective airtime estimation for wireless video streaming. In: 2012 21st International Conference on Computer Communications and Networks (ICCCN), pp. 1–7 (2012). <https://doi.org/10.1109/ICCCN.2012.6289275>
- Sharrab, Y.O., Sarhan, N.J.: Accuracy and power consumption tradeoffs in video rate adaptation for computer vision applications. In: 2012 IEEE International Conference on Multimedia and Expo. IEEE, pp. 410–415 (2012)
- Shafique, M., Molkenthin, B., Henkel, J.: An HVS-based adaptive computational complexity reduction scheme for H.264/AVC video encoder using prognostic early mode exclusion. In: Proceedings of Design, Automation and Test in Europe Conf. and Exhibition, pp. 1713–1718 (2010)
- Sharrab, Y.O., Sarhan, N.J.: Detailed comparative analysis of vp8 and h. 264. In: 2012 IEEE International Symposium on Multimedia. IEEE, pp. 133–140 (2012)
- Richardson, I.E.: The H.264 Advanced Video Compression Standard, 2nd edn. Wiley, New York (2010)



19. Alsmirat, M., Al-Mnayyis, N., Al-Ayyoub, M., Al-Mnayyis, A.: Deep learning-based disk herniation computer aided diagnosis system from mri axial scans. *IEEE Access* **10**, 32315–32323 (2022). <https://doi.org/10.1109/ACCESS.2022.3158682>
20. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT Press (2016). <http://www.deeplearningbook.org>
21. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



**Mohamad Alsmirat** received his PhD Degree in computer engineering from Wayne State University in 2013, his Master Degree in Computer Science from New York Institute of Technology in 2003, and his bachelor's degree in computer science from Jordan University of Science and Technology in 2002. He joined the department of computer science in Jordan University of Science and Technology as an assistant professor. In 2018, Dr. Alsmirat

was promoted to associate professor rank. In 2020, he joined the department of computer science at the University of Sharjah as an associate professor and he is currently working there. His research fields include Multimedia system and Networks, Image Processing and Computer Vision, Parallelization using GPU, Machine Learning.



**Yousef Sharrab** is a professor of artificial intelligence and Internet of Things Engineering at Isra University, Jordan. He has a Ph.D. in engineering (Artificial Intelligence and Communication Networks) from Wayne State University, Michigan, USA. His research experience and interest include Deep Learning, Computer Vision, and Cyber Security in the Domain of the Internet of Vehicles (IoV). Dr. Sharrab worked for General Motors in the R & D Global

Center for a USA national safety project and has over 20 years of

experience in technology-based systems for governments and private sectors.



**Monther Tarawneh** is a computer science professor at Isra University, Jordan. He has a Ph.D. from the University of Sydney. Prior to his Ph.D., he got a master's in Software Engineering and a bachelor's in Computer Science. Dr. Monther's research interests are in the area of algorithms, Parallel and distributed computing, computational biology, Bioinformatics, System Biology, Natural Language Processing, and Molecular Evolution.



**Sana'a Al-shboul**, Data Scientist and Computer vision developer. She has a bachelor's and master's of computer engineering from Jordan University of Science & technology. Sana'a interest is deep learning and computer vision in business and engineering.



**Nabil J. Sarhan** is an Associate Professor of Electrical and Computer Engineering at Wayne State University and the Director of Wayne State Computer Systems and Deep Learning Research Laboratory. He served as the Graduate Program Director of Electrical and Computer Engineering and the Chair of the College of Engineering Faculty Assembly. Dr. Sarhan received his Ph.D. and M.S. degrees in Computer Science and Engineering at the

Pennsylvania State University and a B.S. degree in Electrical Engineering at Jordan University of Science and Technology.