Deep-BVQM: A Deep-learning Bitstream-based Video Quality Model

Nasim Jamshidi Avanaki
Quality and Usability Lab,
Technische Universität Berlin
Berlin, Germany
n.jamshidiavanaki@tu-berlin.de

Steven Schmidt
Sony Interactive Entertainment, Sony
Berlin, Germany
steven.schmidt@sony.com

Thilo Michael
Quality and Usability Lab,
Technische Universität Berlin
Berlin, Germany
thilo.michael@tu-berlin.de

Saman Zadtootaghaj
Advance Technology Group,
Dolby Laboratories
Berlin, Germany
saman.zadtootaghaj@dolby.com

Sebastian Möller
Quality and Usability Lab,
Technische Universität Berlin
Speech and Language Tech., DFKI
Berlin, Germany
sebastian.moeller@tu-berlin.de

ABSTRACT

With the rapid increase of video streaming content, high-quality video quality metrics, mainly signal-based video quality metrics, are emerging, notably VMAF, SSIMPLUS, and AVQM. Besides signal-based video quality metrics, within the standardization body, ITU-T Study Group 12, two well-known bitstream-based video quality metrics are developed named P.1203 and P.1204.3. Due to the low complexity and low level of access to the bitstream data, these models gained attention from network providers and service providers. In this paper, we proposed a new bitstream-based model named Deep-BVQM, which outperforms the standard models on the tested datasets. While the model comes with slightly higher computational complexity, it offers a frame-level quality prediction which is essential diagnostic information for some video streaming services such as cloud gaming. Deep-BVQM is developed in two layers; first, the frame quality was predicted using a lightweight CNN model. Next, the latent features of the CNN were used to train an LSTM network to predict the video quality in a short-term duration.

The implementation of the model for execution as well as training can be found below:
https://github.com/nasimjamshidi/Deep-BVQM.git

CCS CONCEPTS

• **Information systems** → **Multimedia streaming**;

KEYWORDS

Video Quality, Quality of Experience, Bitstream-based Quality Model

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548374

ACM Reference Format:
Nasim Jamshidi Avanaki, Steven Schmidt, Thilo Michael, Saman Zadtootaghaj, and Sebastian Möller. 2022. Deep-BVQM: A Deep-learning Bitstream-based Video Quality Model. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548374

1 INTRODUCTION

With the rapid increase of video streaming content, the gaming video content viewers worldwide surged rapidly, reaching 1.2 billion viewers in 2020. The gaming video content is streamed mainly in two paradigms of interactive cloud gaming services and passive streaming of gameplays such as Twitch.tv, and Facebook gaming. Gaming content has special characteristics, such as extreme motion in some games, and dense texture, which make gaming content bitrate demanding for video streaming services. Therefore, due to limited bandwidth, the video quality assessment (VQA) is a necessary step for any gaming video streaming provider. VQA can be done subjectively by collecting subjective ratings through experiments. Subjective video quality (VQ) experiments are expensive and time-consuming; therefore, VQ models are designed to estimate the VQ automatically.

As a result, several video quality metrics (VQM) have been developed with relatively high performance in the past few years, mainly signal-based VQMs (notably VMAF, SSIMPLUS, and AVQM). Among these models, only a few metrics have been designed for gaming, e.g., NDNetGaming [1]. This becomes more important when considering the bitstream-based models, which are believed to be more suitable metrics to monitor the real-time gaming video streaming services and avoid the computational cost of signal-based models.

Within the standardization body, ITU-T Study Group 12, two well-known bitstream-based VQMs are developed named P.1203 and P.1204.3. Due to the low complexity and low level of access to the bitstream data, these models gained attention from network providers and service providers. These models can only provide the VQ prediction in a minimum of one-second duration. However, due to the high complexity of video games, a much smaller time window might be needed to monitor the gaming VQ precisely. In

addition, both models are developed after conducting a long series of subjective tests, and for any updates of the model, e.g., considering a new codec, a new series of subjective tests is required.

In this paper, a new bitstream-based model is proposed which outperforms the standard models on the tested datasets. In addition, the proposed model is flexible in extension to new codecs and a higher range of encoding parameters. While the model comes with slightly higher computational complexity, it offers a frame-level quality prediction which is important diagnostic information for some video streaming services such as cloud gaming service providers. The proposed model uses two layers modeling approach, in which first, the frame-level quality will be predicted using a lightweight Convolutional Neural Network (CNN) model. Next, the latent features of the CNN will be used to train a Long Short-Term Memory (LSTM) network to predict the VQ in a 1-second.

The latent features extracted from the CNN network can predict the well-known VQ metric VMAF at the frame-level with Pearson Linear Correlation Coefficient (PLCC) over 0.89 and Root Mean Score Error (RMSE) below 12.89 (at VMAF's 100-point scale). The LSTM model predicts the VQ with high accuracy of approximately 0.95 in PLCC across the three validation datasets.

2 MOTIVATION

Bitstream-based VQ models have shown a high accuracy, comparable to signal-based models, at the same time capable of measuring the VQ in real-time [2]. The current standard models, such as P.1203 and P.1204.3, predict the audiovisual quality in one-second duration as the shortest possible time window. While such a short duration might be enough for some applications and content, gaming content with its special characteristics might require a smaller window of prediction. Recently, Tencent [3] made a contribution and raised the importance of frame-level quality prediction, which has gained lots of attention at ITU-T Study Group 12. Gaming content is a special type of content that could consist of an extreme motion due to camera rotation (e.g., in shooting games) or extreme illumination changes (such as explosions in shooting games), which are not common type of videos characteristics in currently available video encoders, and they failed to take them into account. Thus, it results in severe quality changes even for a short time window as the rate controller increases the number of Intra blocks for such a high motion content. Figure 1 compares the quality changes based on VMAF [4] and P.1203 [5] for the compressed video sequence of Dauntless game, from Cloud Gaming Video DataSet (CGVDS) [6], created inspired by Tencent's contribution [3]. As it can be observed, the VMAF, similar to most signal-based models, is capable of predicting the frame-level quality, while P.1204.3 and P.1203 predict the quality in a one-second duration as the shortest time window. Such a short time window would allow the cloud gaming providers to gain diagnostic information that can be used to adapt their system accordingly and avoid a significant decrease in the gaming experience. However, the standardized bitstream-based audiovisual quality models cannot provide such diagnostic information, which might be crucial.

Another concern over the development of bitstream models is their sensitivity within the range of parameters that are used in training the model. Therefore, any change to the codec or encoding parameters may result in inaccurate quality prediction. Consequently, multiple models are required to be developed following massive conduction of subjective tests to cover new codecs and a range of parameters, as can be seen within the standardization body's efforts, e.g., the multiple phases of ITU-T work item P.NATS series. Therefore, the method followed in this paper is to make the model development less tied to the subjective data to allow covering a higher range of parameters without conducting massive subjective tests.

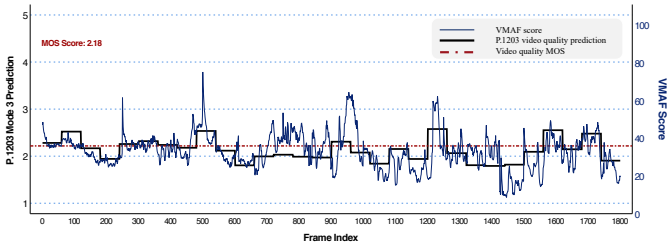


Figure 1: The quality changes for 1800 frames of the video sequence Dauntless, based on VMAF [4], MOS, and ITU-T Rec. P.1203 Mode 3 [5].

3 DATASETS

In this section, three gaming VQ datasets are presented that are partially used for training and validating the proposed model. These three datasets, called GVSET, KUGVD, and CGVDS, are created, targeting two different video streaming services. The first two were created following Twitch.tv encoding recommendation, while the last one was created based on encoding recommendation from Nvidia GeForce Now cloud gaming service. In addition, a new dataset without conducting any subjective experiment is created using 20 recorded video raw sequences of gaming content to increase the size of the dataset for training. This dataset is called the "Training dataset" for simplicity of reference.

3.1 GamingVideoSET

The GamingVideoSET, also known as the GVSET, is comprised of 12 distinct video games. Two 30-second video sequences are taken from each video game [7]. A total of 24 raw video games are recorded and encoded to create the dataset. The raw sequences in GVSET are encoded under multiple bitrates and resolutions to generate different spatial distortions in videos. The values assigned to the encoding parameters are 30 frames per second, 24 different bitrate-resolution pairs, and H.264/MPEG-AVC for the codec type. The dataset consists of a total of 576 degraded videos made from 24 source sequences. Aside from that, the GVSET contains subjective ratings for six video games played under 15 distinct resolution-bitrate pairs (three resolutions, five bitrates each), and with the H.264/MPEG-AVC codec type, which resulted in 90 stimuli. Thus, the subjective evaluations for 90 videos out of 576 are created.

3.2 KUGVD dataset

Barman and his colleagues have developed the Kingston University Gaming Video Dataset (KUGVD) [8]. In total, six sequences are

Table 1: Encoding parameters that is used to develop the first part (Part-1) dataset for training the frame-level module

Parameters	Quantity	Values
Source videos	20	10 from CGDVS
Encoding modes (rate controller)	1	CBR
Codec	2	H.264
Preset	3	Slow,llhq, Fast
Bitrate	5	1 Mbps to 10 Mbps
Resolution	1	1080p
Frame rate	1	60 fps
20 x 1 x 2 x 3 x 5 x 1 x 1 = 600 files		
600 x 360 (5th frame) = 216 000 (selected frames)		

selected from different computer games in the KUGVD dataset. The encoding parameters are kept the same as those of the GVSET dataset. In total, there are 144 distorted gaming sequences, of which 90 stimuli are used for the subjective test based on 15 resolution bitrate pairs from the reference videos, similar to GVSET.

3.3 CGVDS dataset

The Cloud Gaming Video DataSet (CGVDS) is created using the recorded gameplays of 15 games. Similar to GVSET and KUGVD, the videos are recorded at a 30-second duration [6]. The dataset is originally created to develop the ITU-T Rec. G.1072 model. In total, five subjective tests were conducted to create this large dataset with 380 stimuli, using four different bitrates, three framerate levels (20, 30, and 60 fps), and three resolutions (480p, 720p. and 1080p). As a choice of quality rating, a seven-point extended continuous ACR scale (EC scale) is used as recommended by ITU-T Rec. P.809. The encoding followed the Nvidia GeForce Now cloud gaming service, where Nvidia Encoder (NVENC) is selected that uses the hardware GPU accelerator engines. For encoding, low latency high quality preset is used with keyframes every 4 seconds.

3.4 Training dataset

The model development is a two-stage process, where in the first stage, a model is trained based on a pseudo-objective metric (i.e., VMAF). Since there is no need to access the subjective ratings for this stage of the model training, a new dataset is created that was solely used for the first stage of model development. To create the "Training dataset Part-1", 20 source sequences are selected and ten sequences are taken from CGVDS. Ten others are newly generated gameplay materials, all recorded losslessly at 1080p, 60 fps, and 30-second duration. The videos are then encoded using H.264 with the CBR mode in three different presets that are common in gaming video streaming, at five different bitrate levels. This results in a total of 600 video sequence files (see Table 1).

In addition to the "Training dataset Part-1", the second part of the dataset is built using all video sequences of GVSET that do not have subjective ratings. Therefore, a total of 144 distorted sequences from 18 video sequences are used in the second part of training dataset. Thus, the training dataset consists of 744 sequences in total.

4 MODEL ARCHITECTURE

In order to develop the model, a two-step approach is followed. First, a CNN architecture is designed and trained to predict the frame-level quality. Second, the latent features of the CNN model, together with some high-level metadata, are used to train a recurrent neural network (RNN) for VQ prediction. In this section, the frame quality model architecture is introduced first. Next, the VQ modeling approach is described.

4.1 Frame Quality Model

In order to develop the frame quality prediction model, different choices of CNN architecture are taken into consideration. While simple models did not perform well, a deep model similar to ResNet architect would not show a significant improvement in results. In fact, due to the low variance of bitstream data, a very deep network would not help but rather result in overfitting. In addition, it is not of interest to service providers to use a bitstream-based model with a high computational cost. Therefore, a medium-size CNN inspired by [9] in speech quality assessment is developed as the architect is shown in Table 2.

In order to train the CNN model, QP values of each frame, for macroblocks of 16×16, are used as input to the model, and VMAF values are used as the target values. Since the model is not supposed to predict the VMAF but rather to learn the quality behavior within the QP values, the VMAF values are divided into five groups with 20 VMAF value intervals. In addition, the 6 VMAF score difference is decided between each group to take into account the potential inaccuracy of VMAF in classification for frame quality. The six-score difference confidence interval is decided based on the subjective rating collected by authors in CGVDS dataset [6]. For all conditions in that at least six VMAF score differences can be seen, no statistically significant difference was observed between the subjective MOS scores.

4.1.1 Data Preparation. In order to prepare the data for the training, the bitstream information is extracted using "FFmpeg-qp-parser" tool[10]. This tool uses the FFmpeg inbuilt function and provides the list of QP values for each frame used in this work.

Table 2: CNN design (each convolutional layer is followed by a batch normalization and ReLu layer. The kernel size is 3x3.)

Layer	Output Size
Input	1x89x89
Pool	1x44x44
Conv1	16x44x44
Pool	16x22x22
Conv2	32x22x22
Conv3	64x22x22
Pool/Dropout (20%)	64x11x11
Conv4	64x11x11
Fc1	3000
Fc2	1000
Fc3	5

For training purposes, the QP values at a 16 x 16 block size are reordered into a 89 x 89 patch size as an input of the model.

For training, VMAF values are grouped into five different classes, class 1 to class 5; higher refers to the higher VMAF value class. Considering the the six-point distance between each class, the classes are defined as follows: Class-1 [0-17], Class-2 [23-37], Class-3: [43-57], Class-4 [63-77], and Class-5 [83-100]. While it might be argued that a higher number of classes could improve the model training, it must be noted that considering the six-point VMAF distance that is taken into account in this work, the number of data points could be reduced significantly by raising the number of classes. Also, we did not observe a significant improvement by a slight increase in classes. In addition, due to similarity between frames, only every 5th frame are used for training purpose. Therefore, in total, over 200k data points are kept for the training of the model.

In the first set of training, it was observed that the model prediction is biased towards the high-quality class. Therefore, the histogram of VMAF values in the training dataset is plotted to get insight into the VMAF distribution. It was observed that the data is strongly skewed towards high-quality classes (Class-5 [83-100]). Therefore, to balance out the training data, random sampling is applied based on a probability function that keeps the ratio of data points in Class-3 to Class-5 similar to data points in Class-2. The second class is selected as a target to balance out the data, as it has a smaller number of data points but not as significantly fewer data points that is available in Class-1, which may lead to underfitting. Due to the sampling process, the number of data points is significantly reduced, but it helps in the training process to obtain a high-performance trained model. It must be noted that the number of data points in Class-1 was less than half of the other classes.

4.1.2 Model Training. For training each CNN model, the QP values, together with bitstream metadata such as frame size, frame type, and preset, are used. The bitstream metadata, e.g., the frame size and frame type, are only added to the fully connected layer to allow better training through the back-propagation.

To evaluate if the model is overfitted or not, the model performance is not only reported based on the training dataset but also based on two validation datasets. These two validation datasets are KUGVD and GVSET videos that are used in the subjective tests. The validation is used for frame-level performance assessment and selection of the CNN model. Before training the model, all input data is normalized for better training.

Figure 2 illustrates the performance of the training model using a confusion matrix on training and validation datasets. As a choice of validation set, the GVSET dataset is used. As it can be seen, the model is capable of classifying the VMAF values on the training dataset. However, the prediction accuracy drops for Class-3 and Class-4.

While classification results show the model is well capable of classifying the VMAF values, it is more important to understand if the selected model provides an excellent latent feature to be used as perceptual features to predict the VQ in the later stage. Therefore, the suitability of the representation features (latent features) of the trained CNN model is evaluated by using it for regression tasks of VMAF quality prediction at the frame level. One can argue that a more straightforward solution could have been training the CNN architect following a regression task to predict the VMAF. But it has to be noted that we aimed to avoid developing a model that is biased towards another metric (i.e., VMAF) and this step is solely to observe the suitability of the features. Therefore, Random Forest (RF) is selected for the regression task. For training the RF model, the same training and validation sets are used, which were created for the classification task. However, instead of the QP values as a choice of input to the model, the latent representations are used, taken from the output of the last fully connected layer. In addition, the frame size, frame type, and preset are also used again to assist the training phase.

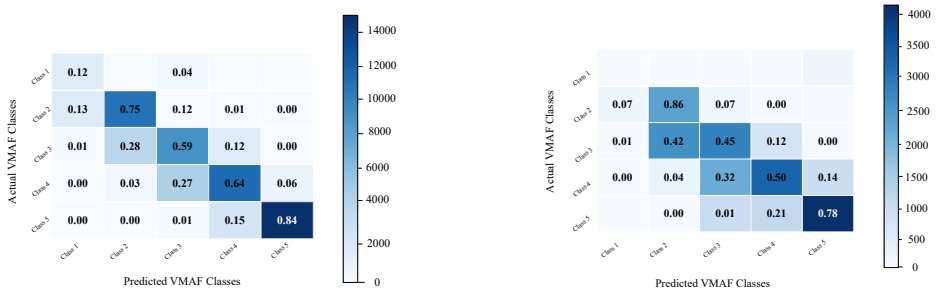
Table 3: Performance of trained Random Forest model to predict VMAF on the validation datasets, on frame level values and averaged video level.

	Averaged Video Level			Frame Level		
Dataset	**PLCC**	**SROCC**	**RMSE**	**PLCC**	**SROCC**	**RMSE**
GVSET	0.97	0.97	7.51	0.89	0.90	11.62
KUGVD	0.96	0.96	9.43	0.90	0.90	12.89
CGVDS	0.98	0.97	8.41	0.93	0.94	10.35

Table 3 presents the result of the validation sets of KUGVD, GVSET, and CGVDS. It has to be reminded that in the validation sets of KUGVD and GVSET, all videos with subjective ratings are kept out from all training processes and only used for validation purposes. For CGVDS, only five games out of 15 video games are kept for validations set. The results on the validation sets show a high correlation and low RMSE values on the validation set, which is promising and confirm the effectiveness of the latent features for frame quality tasks. It can be observed that the performance result for CGVDS is slightly higher compared to the two other datasets. The reason could be a higher number of data points in the training dataset that are similar to the encoding setting of CGVDS compared to GVSET/KUGVD. Next, the scatter plot of these results will be presented. The scatter plot could provide important information, such as a skewed plot, suggesting bias in the data.

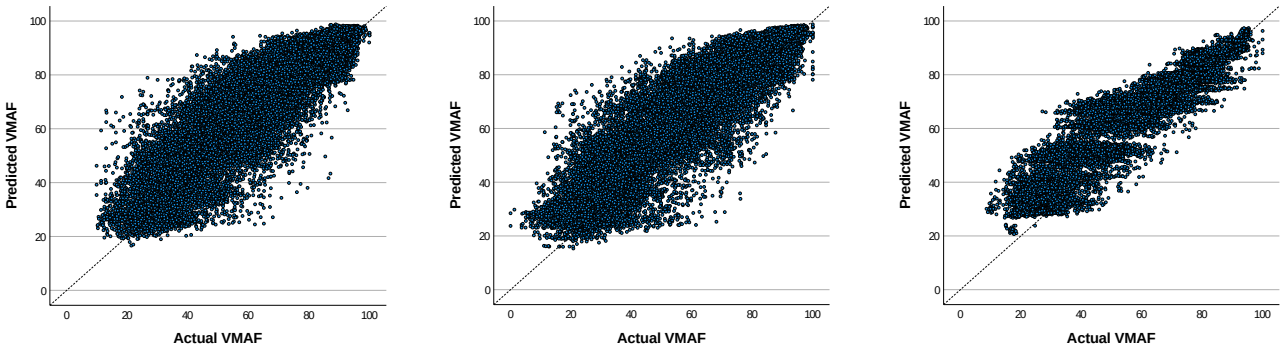
Figures 3, 4 show the scatter plots of the predicted VMAF and actual VMAF values, at frame-level and video-level for all three validation sets. The video-level VMAF is measured by averaging the frame-level VMAF over all frames in each video sequence. The scatter plots show promising results considering the fact that the model only has access to QP values and some basic bitstream metadata. It must be noted that in the CGVDS dataset, the 50 Mbps condition is removed from the scatter plot due to expected high-quality prediction, misleading the result by showing unreasonably high-performance results. Such data points would push a large portion of the prediction data points to the end of the scale and improve the correlation and RMSE without helping to show the trend of data.

The RF model could potentially be used to provide diagnostic information to the service and network providers, allowing them to track the sudden quality drops. Therefore, it is required to investigate whether the RF model can accurately keep the quality changes over a time window or not. To further investigate the suitability of the RF model, the VMAF predictions and actual VMAF values are compared for a video sequence, taken from validation sets using line plots.



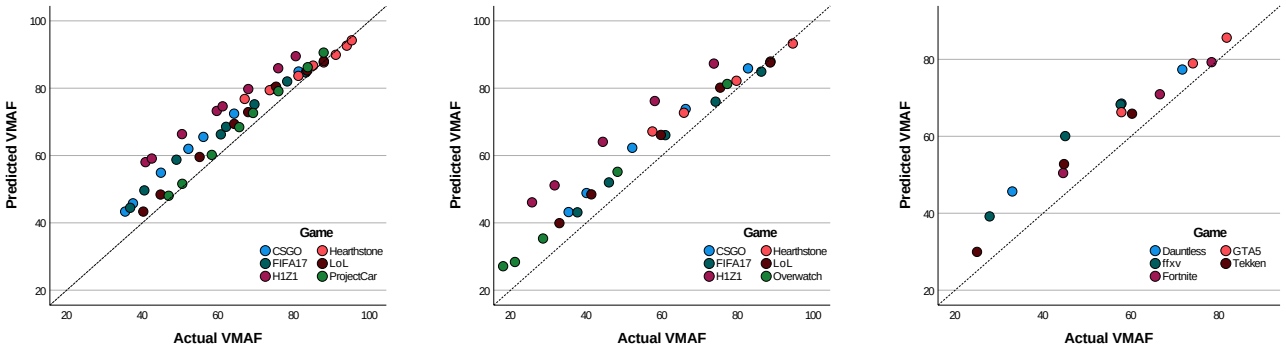
(a) Result on the Training Dataset. **(b) Result on the Evaluation Dataset.**

Figure 2: Confusion matrices of the CNN model performance on the training and validation data.



(a) GVSET. **(b) KUGVD.** **(c) CGVDS.**

Figure 3: Scatter plots of predicted VMAF and actual VMAF at frame-level.



(a) GVSET. **(b) KUGVD.** **(c) CGVDS.**

Figure 4: Scatter plots of predicted VMAF and actual VMAF at video-level.

The line plot allows understanding if the VMAF prediction values take into account the sudden changes over a period of time which is the main motivation of this work. Figure 5 shows the line plots from the game GTA. As it can be seen from the line plot, the model is well capable of keeping the trend of the VMAF values with some noisy variation. However, in most severe quality drops or gains, the model follows the trend quite accurately. One of the issues that can be seen from Figure 5, from frame number 550 to 600, is the higher range of variation in prediction compared to actual VMAF values. A similar trend is observed for other videos in the validation dataset, which will be released in the repository where the source code will be shared.

In the next section, the latent features will be used to predict the VQ at a 1-second duration.

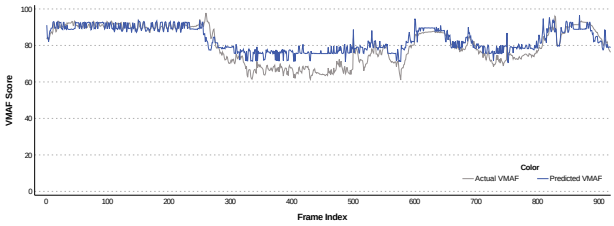


Figure 5: Comparison of predicted VMAF, blue line, vs actual VMAF scores, grey line, over 900 frame, for GTA-5 sequence at 4000 kbps, 1080p resolution from CGVDS.

4.2 Video Quality Model

A video sequence consists of sequences of frames over a period of time that has been seen in some papers as a time series problem [11, 12]. In this work, the VQ prediction is also seen as a time series problem, and a time series model is taken into consideration. Therefore, an LSTM network is selected as it has shown a reliable model for time series problems.

The frame-level quality module is responsible for providing perceptual representation features that can be used for VQA. To predict the VQ, the representation features from the previous step will be used as input to the LSTM model. In addition, some other metadata will be used to ensure adequate rich features are available to the model to predict the VQ accurately.

The LSTM is developed with an input time series of 15 data points, each data point including the latent features, frame type, frame size, and framerate. The LSTM model architect is shown in Table 4. In order to allow the model to predict the MOS at a 1-second duration, similar prediction level to ITU-T Rec. P.120X series models [5, 13], a sampling for each one-second duration is needed. An example of such sampling is taking every second frame of a 30 fps video sequence to reduce the input size to predefined size of 15. In addition, the MOS values are not available at a one-second time window. Therefore, the MOS values are estimated based on the ratio of average VMAF in one-second duration over the total VMAF of the video sequence, as shown in Equation 1. It must be noted that VMAF is only used to define the distribution of quality changes within a video sequence and not the actual video quality value. Therefore, the model would not be biased toward VMAF but the variation of VMAF within a video sequence.

$$\widehat{MOS_{1-second}} = MOS_{video} \times \frac{VMAF_{1-second}}{VMAF_{video}} \quad (1)$$

In the training phase, the MOS scores are used for training. The MOS scores are taken from CGVDS, GVSET, and KUGVD for training and validation purposes as a choice of the dataset. Due to the limited number of MOS scores available, the selection of training and validation is made carefully. The training has been done three times as follows:

- **GVSET as Training set**: GVSET is used for training the model, and the KUGVD dataset is selected as a validation set. Due to the similarity of the codec and encoding parameters, it is expected to gain high-quality results.
- **KUGVD as Training set**: KUGVD is used as training, and the model performance is validated on the GVSET dataset.
- **CGVDS as Training set**: For training the CGVDS, ten games are used for training, and five others are used to validate the model. The five video games that are left out for validation are Tekken, Dauntless, GTA5, Final Fantasy XV (ffvx), and Fortnite sequences.

The results from training datasets are presented in Table 5. It must be noted that the result for each dataset is solely reported based on training and validating on the same dataset. In addition, the performance of training is reported for the one-second duration as well as averaged prediction over the full 30-second duration time. The correlation and RMSE results revealed the high performance of the developed model. However, the performance must be investigated on the validation dataset, which will be presented in the next section.

5 PERFORMANCE ANALYSIS

This section presents the performance results of the proposed and state-of-the-art bitstream-based models in terms of RMSE, PLCC, and SROCC. Since the proposed model is designed for 1080p resolution, to have a fair comparison, the performance of all models is compared only at 1080p resolution. Furthermore, for CGVDS, all videos sequences encoded at a framerate of 20 fps are removed. These videos are removed since the proposed model does not take into account the jerkiness. This is also the case for other models, except for the BQGV model [6].

Table 6 presents the summary of performance evaluation, in terms of PLCC, SROCC, and RMSE, of the investigated bitstream-based VQ models in this paper for all three gaming datasets. Based on Table 6, the result shows a very high performance of the two ITU-T models of P.1203 Mode 3 and P.1204.3. However, their performance drops for CGVDS. The main reason for the performance drop might be the different encoding settings that CGVDS uses compared to the training dataset of the ITU-T recommended models. This raises the concern over the usage of these models out of the range of parameters that these models are developed for. The BQGV is trained based on the CGVDS dataset encoding setting. Therefore, the model provides very high performance for the CGVDS dataset. However, the performance significantly drops for GVSET and KUGVD. It has to be mentioned that BQGV does not take B-frames as input to the model. Therefore, all B-frames in GVSET are considered P-frames to allow the model to be executed. For P.1203 Mode 1,

Table 4: LSTM design using ReLU activation function.)

Layer	Output Shape	Parameters
Input	1 x 15 x 8	0
LSTM_1	(None, 15, 64)	18688
LSTM_2	(None, 15, 32)	12416
LSTM_3	(None, 15, 32)	8320
LSTM_4	(None, 16)	3136
Dropout	(None, 16)	0
Dense	(None, 17)	17
Total trainable params: 42,577		

Table 5: Result of training the LSTM network on different datasets, at one-second time window and 30-second time window. The result is presented based on the training and validating on the same dataset that is referred.

	One-Second Video Level			Full-Length Video Level		
Dataset	**PLCC**	**SROCC**	**RMSE**	**PLCC**	**SROCC**	**RMSE**
GVSET	0.95	0.95	0.40	0.97	0.98	0.29
KUGVD	0.95	0.94	0.42	0.97	0.96	0.33
CGVDS	0.97	0.96	0.38	0.98	0.97	0.28

the result shows a medium performance for GVSET and KUGVD. Surprisingly, the performance increases for CGVDS, which has a different encoding setting compared to the P.1203 training dataset.

Generally, from the performance results of bitstream-based models, it can be concluded that the performance of the models depends strongly on the encoding setting used in the evaluation dataset. It was already one of our main criteria for model development to allow the model to be trained on multiple different encoding settings without conducting a massive subjective test.

The video quality measurement of Deep-BVQM is calculated based on averaging the per-second predictions over 30 seconds duration. It must be noted that for the CGVDS dataset, the result is presented only for five video sequences that are kept out from the training set. The games that the evaluation is reported based on are Tekken, Dauntless, GTA-5, Final Fantasy XV (ffxv), and Fortnite. In addition, the results are based on three different models trained on a different dataset than the evaluating dataset. The reader can refer to Section 4.2 for more details on the training process. The result suggests the high performance of the proposed model and consistency of the model performance across all validation datasets. To get more insight into the model performance, the predicted VQ compared to video quality MOS using scatter plots, shown in Figure 6 (a,b). The scatter plots show slight shifts in the prediction compared to the ground truth. This shift is due to the subjective bias between the two datasets, which is reflected in the result by training on one dataset and testing on the other. It must be reminded that the performance results for GVSET are reported based on the training of the model using the KUGVD and vice versa. Such a mix match training is done due to the limited number of data points and avoids any training biases.

Another interesting observation is when the model is trained based on only the GVSET dataset and tested on the CGVDS dataset. Such evaluation might not be valid since GVSET has different bitstream characteristics due to differences in encoding settings. However, the model performs well, as can be seen in Table 7. The reason behind such a performance is due to the rich perceptual features of the CNN model. However, some metadata that is used in the training of LSTM would negatively impact the result, such as frame size. Frame size in CGVDS is expected to be much larger than GVSET because of using NVENC encoder with llhq preset. This result could be significantly improved by training the LSTM model by merging all three datasets, which were not evaluated due to the unavailability of the validation dataset for evaluation, but the model will be released for public usage.

Table 6: Performance of bitstream-based models on all three gaming dataset, only at 1080p resolution.

Dataset	Metrics	P.1203 m1	P.1203 m3	P.1204.3	BQGV	DBVQM
GVSET	PLCC	0.67	0.91	0.92	0.68	0.95
	SROCC	0.72	0.91	0.91	0.65	0.95
	RMSE	1.53	0.86	0.53	0.98	0.60
KUGVD	PLCC	0.65	0.92	0.96	0.74	0.96
	SROCC	0.73	0.97	0.97	0.70	0.95
	RMSE	1.45	0.59	0.41	0.87	0.57
CGVDS	PLCC	0.80	0.86	0.83	0.88	0.96
	SROCC	0.81	0.85	0.83	0.87	0.96
	RMSE	0.70	0.49	0.52	0.40	0.49

Table 7: Performance results of proposed model, trained on GVSET, evaluated on CGVDS dataset.

	30 FPS			60 FPS		
Dataset	**PLCC**	**SROCC**	**RMSE**	**PLCC**	**SROCC**	**RMSE**
CGVDS	0.84	0.84	0.83	0.86	0.85	0.84

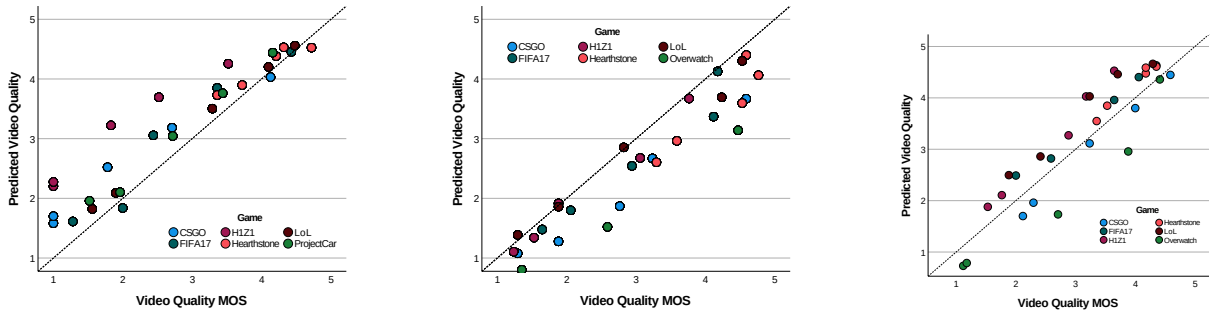
Finally, it is interesting to see how the model performs on a non-gaming dataset. We could not find a non-gaming dataset with a similar encoding setting that is used for the training of the model. Therefore, the AVT-VQDB-UHD-1 dataset [14] was selected. The AVT-VQDB-UHD-1 is created using 2160p source content and tested on 65" and 55" TVs. For a lower resolution than 2160p, the videos are upscaled to 2160p. In addition, for all videos, 4:2:2 chroma sub-sampling was used compared 4:2:0 chroma sub-sampling that was used in gaming datasets used in this work. Therefore, the model is not expected to perform well due to differences between the training and validating dataset. However, it is interesting to investigate how the model performs on a different dataset in terms of content, display test size, and encoding setting. For evaluation, only the 1080p videos at a framerate of 60 fps are selected. The results revealed that the model performs at a medium level with PLCC of 0.74, SROCC of 0.68, and RMSE of 0.67. Considering the significant differences between the datasets, it is a promising result.

6 DISCUSSION AND CONCLUSION

6.1 Computational Complexity

The proposed model's computational complexity relies on two steps, bitstream extraction, and model execution. The first step depends strongly on the bitstream parser, which is not a part of the model design. Thus, its computational complexity is not reported.

For analysis of the model execution complexity, two parts of the model must be considered. First, the CNN model is used to measure the latent representation features. This layer has in total of 7916996 trainable parameters. For testing the run time, the NVIDIA Tesla K80 GPU is used from Google Colaboratory Pro tool [15]. If all the bitstream files are available in the right format and the model is loaded, the model can be executed in 0.022 seconds for 1800 frames of a 30-second video sequence with 60 fps framerate and 1080p resolution. In addition, the test was conducted for a single core hyper-threaded Xeon Processors 2.3Ghz (1 core, two threads), which revealed that the model execution time could be as high as 0.29 seconds. It must be noted that the run time is measured without considering any required prepossessing. The second part of the model is an LSTM model that has a total of 42577 trainable



(a) GVSET dataset 1080p. (b) KUGVD dataset 1080p. (c) KUGVD dataset 720p.

Figure 6: (a) and (b) present scatter plots of Predicted VQ vs. VQ MOS on the validation datasets at 1080p. (c) presents scatter plots of Predicted VQ vs. VQ MOS on KUGVD datasets at 720p.

parameters, and it needs 0.133 seconds to be executed for a 30-second video sequence at 60 fps. The execution time goes up to 0.2 seconds for a CPU with the same specification mentioned above.

6.2 Future Extension

One of the main limitations of this work is the limited range of parameters selected for the training. This limitation is because of the time-consuming process of encoding and measuring VMAF values in the prepossessing step. Among them, two critical limitations are single codec usage and the selection of 1080p resolution. Extending the model to a newer codec requires access to a parser to extract the QP values. Therefore, the model has developed only for H.264, for which an open-source parser was accessible. However, newer codecs raise a new challenge, which is different structures of partitioning, known as Coding Tree Units (CTUs) or Coding Tree Blocks (CTBs) [16]. This new structure would make it impossible to keep the same input size for training in the CNN model. One solution is to select a constant block size, e.g., 16×16, and split the large CTUs into smaller blocks. For example, a CTU of 64×64 can be split into four 16×16 blocks with the same QP value. Therefore, the model can be easily trained for a newer codec, but prepossessing would be required. In this case, access to the CTU would be necessary and could potentially enhance the prediction by adding one more channel to the input of the CNN, denoting the block size.

Another limitation in this work is supporting different resolutions affecting the number of QP values as input to the model. Therefore, two straightforward solutions are discussed in this section. First is the development of multiple models for different resolutions. This would be possible since, typically, only a few common sets of standard resolutions are used in the video streaming pipeline, which is limited. Therefore, developing multiple models would still be a reasonable solution. However, it comes with the cost of reloading different models, which might not be a practical solution.

Another solution could be rescaling the input to 89×89 that is used for 1080p resolution. A solution for allowing the model to work for lower resolution is to scale up the QPs patch to 89×89 format, which can be done using a near-neighbor interpolation technique. In this work, an extension from 720p to 1080p is conducted without any retraining using 720p resolution sequences in the KUGVD dataset. The result show that without any retraining, the current model (trained on GVSET at 1080p) would reach a high PLCC of 0.92 and RMSE of 0.49. Figure 6 (c) illustrates the scatter plot of the predicted VQ and actual rated VQ for 720p resolution videos. In order to allow a higher resolution to be used for the model, a pooling layer can be added at the beginning of the network. For example, for 2160p resolution, an average pooling layer of 2×2 and stride of 2 can be used. To achieve an accurate model that works for multiple resolutions, complete retraining of the model is required.

Finally, the model can be extended by adding structures of partitioning as well as the intra/inter prediction blocks as two new channels to the input of the model. Such information could significantly improve the model performance.

6.3 Conclusion

This paper aims to develop a bitstream-based VQ model that is suitable for monitoring the quality of video streaming services. As an example of the usage of this model, it is trained on a large gaming dataset. While the number of encoding parameters is limited (e.g., limited resolution), this work illustrates a potential approach to training a bitstream model without conducting a massive subjective test, which is typically done in traditional bitstream-based model development. The core idea is to train perceptual features from bitstream metadata, that can be used for training and measuring VQ. If rich latent features are learned from this stage of the model, the dependency on bitstream metadata will be decreased; consequently, a smaller subjective dataset might be required. Therefore, in the presence of a large video dataset annotated by a pseudo-objective metric, an accurate model can be developed for desired application.

In addition, this model provides a frame-level prediction, i.e., predicting VMAF, which could be suitable diagnostic information even if that would not have a major impact on user perception. Providing a relative constant quality is desired by service providers compared to the variable quality, and such a frame-level prediction could provide much higher granularity for quality prediction.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871793.

REFERENCES

[1] Markus Utke, Saman Zadtootaghaj, Steven Schmidt, Sebastian Bosse, and Sebastian Möller. NDNetGaming-Development of a No-Reference Deep CNN for Gaming Video Quality Prediction. *Multimedia Tools and Applications*, pages 1–23, 2020.

[2] Rakesh Rao Ramachandra Rao, Steve Göring, Robert Steger, Saman Zadtootaghaj, Nabajeet Barman, Stephan Fremerey, Sebastian Möller, and Alexander Raake. A large-scale evaluation of the bitstream-based video-quality model itu-t p. 1204.3 on gaming content. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020.

[3] Joel Jung, Xiang Li, and Shan Liu. New indicator to reflect sudden quality variations of gaming content for P.BBQCG. Itu-t contribution c.588, ITU-T Study Group 12, Geneva, 2021.

[4] Netflix. VMAF - Video Multi-Method Assessment Fusion. https://github.com/Netflix/vmaf. [Online: Accessed 09-December-2021].

[5] ITU-T Recommendation P.1203. *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport.* International Telecommunication Union, Geneva, 2017.

[6] Saman Zadtootaghaj, Steven Schmidt, Saeed Shafiee Sabet, Sebastian Möller, and Carsten Griwodz. Quality Estimation Models for Gaming Video Streaming Services Using Perceptual Video Quality Dimensions. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 213–224, 2020.

[7] Nabajeet Barman, Saman Zadtootaghaj, Steven Schmidt, Maria G Martini, and Sebastian Möller. Gamingvideoset: a dataset for gaming video streaming applications. In *2018 16th Annual Workshop on Network and Systems Support for Games (NetGames)*, pages 1–6. IEEE, 2018.

[8] Nabajeet Barman, Emmanuel Jammeh, Seyed Ali Ghorashi, and Maria G Martini. No-reference video quality estimation based on machine learning for passive gaming video streaming applications. *IEEE Access*, 7:74511–74527, 2019.

[9] Gabriel Mittag and Sebastian Möller. Deep learning based assessment of synthetic speech naturalness. *arXiv preprint arXiv:2104.11673*, 2021.

[10] FFmpeg-qp-parser tool. https://github.com/slhck/ffmpeg-debug-qp.

[11] Domonkos Varga and Tamás Szirányi. No-reference video quality assessment via pretrained cnn and lstm networks. *Signal, Image and Video Processing*, 13(8):1569–1576, 2019.

[12] Qiuxia Bao, Ruochen Huang, and Xin Wei. Video quality assessment based on the improved lstm model. In *International Conference on Image and Graphics*, pages 313–324. Springer, 2017.

[13] ITU-T Recommendation P.1204.3. *Video Quality Assessment of Streaming Services over Reliable Transport for Resolutions Up to 4K with Access to Full Bitstream Information.* International Telecommunication Union, Geneva, 2020.

[14] Rakesh Rao Ramachandra Rao, Steve Göring, Werner Robitza, Bernhard Feiten, and Alexander Raake. AVT-VQDB-UHD-1: A large scale video quality database for UHD-1. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 17–177. IEEE, 2019.

[15] Google colab. https://colab.research.google.com/.

[16] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.