# nofu – A Lightweight No-Reference Pixel Based Video Quality Model for Gaming Content

Steve Göring, Rakesh Rao Ramachandra Rao, Alexander Raake
Audiovisual Technology Group; Technische Universitt Ilmenau, Germany
Email: [steve.goering, rakesh-rao.ramachandra-rao, alexander.raake]@tu-ilmenau.de

***Abstract*—Popularity of streaming services for gaming videos has increased tremendously over the last years, e.g. Twitch and Youtube Gaming. Compared to classical video streaming applications, gaming videos have additional requirements. For example, it is important that videos are streamed live with only a small delay. In addition, users expect low stalling, waiting time and in general high video quality during streaming, e.g. using http-based adaptive streaming. These requirements lead to different challenges for quality prediction in case of streamed gaming videos. We describe newly developed features and a no-reference video quality machine learning model, that uses only the recorded video to predict video quality scores. In different evaluation experiments we compare our proposed model *nofu* with state-of-the-art reduced or full reference models and metrics. In addition, we trained a no-reference baseline model using brisque+niqe features. We show that our model has a similar or better performance than other models. Furthermore, nofu outperforms VMAF for subjective gaming QoE prediction, even though nofu does not require any reference video.**

## I. Introduction

Besides classical video streaming applications like Netflix, YouTube, Amazon Prime Video, etc., there is also a community that streams gaming videos over the internet. Famous platforms to watch such streams of gaming videos are Twitch[1] and YouTube Gaming[2]. On both platforms two different types of streams can be distinguished – live and non-live transmissions. For live encoding different codec presets are applied in comparison to a non-live transmitted video encoding. Not all codecs are suitable for such a scenario, due to the requirement that encoding needs to be done in real time [5]. Most suitable codecs for live video encoding are H.264 and H.265, e.g., in combination with hardware encoding acceleration with a fast or ultrafast preset and a 1-pass encoding scheme. Our focus in this paper is the live scenario, because the non-live scenario is more similar to general video streaming platforms, where an advanced encoding process can be applied. Subjective evaluation methods for gaming QoE are currently developed and discussed [20, 19]. In general, similar to the classical video streaming case, there are several influencing factors for gaming QoE, e.g., human, system, context and more [20]. Beside classical video streaming factors such as content characteristics that influence encoding or the resulting quality, there exist factors specific for video streams of gaming sessions. One reason for this is that games have mostly similar content, similar patterns, specific camera movements, artificially generated textures and more [33]. Objective video quality metrics for gaming QoE have already successfully been analyzed [5, 2, 3]. For example, VMAF [22] shows good correlation with subjective ratings. However, full-reference models cannot be used in practice in live streaming applications, because mostly the calculation is slow and requires a high-quality reference version of the video, that is not available in typical gameplay recordings. For gaming videos, during a game session a player records or streams an encoded version of the video to external servers, and an uncompressed version of the video is not stored. Hence, no-reference metrics are more applicable for the gaming scenario. As shown in [3], they show promising results for gaming QoE, e.g., ***niqe*** [17] or ***brisque*** [18]. Also Zadtootaghaj et al. [34] analyzes different no-reference metrics for gaming content with promising results. We propose a method to build a no-reference (NR) video quality metric that uses state-of-the-art and newly developed features to predict video quality on a short-segment basis, e.g. 2-10 seconds, that are typically used in dynamic adaptive streaming [23]. We train a machine learning model, in our case a random forest model with feature selection, to predict per-segment subjective scores. An aggregation of the complete video session quality using several video segments scores can be done with other state-of-the-art temporal aggregation methods [27, 8], such as the integration module of ITU-T P.1203 [28, 11]. A more detailed long term quality analysis for gaming QoE is required, where also audio quality and delay could be included. Also the lack of availability of public long-term gaming QoE databases hinders the development of such models. In our evaluation experiments we use the GamingVideoSET [4]. We train and evaluate our model on the VMAF scores included for all videos and using the subjective scores. Our proposed model – **nofu** – outperforms VMAF in predicting accuracy. In addition, we train a model using brisque+niqe features as baseline NR model, which **nofu** also outperforms.

The paper is organized as follows. In Section II, a brief overview of NR video quality models is provided. We discuss models for gaming QoE prediction and outline the differences to our proposed model. Further, in Section III, we describe our model and features in detail. To verify and evaluate our model and features we conducted several experiments, as described in Section IV. Finally, we conclude with a discussion of the model, a short outlook on future ideas and work in Section V.

[1] https://twitch.tv
[2] https://gaming.youtube.com/

## II. Related Work

In general, video quality models can be categorized into three classes: no-reference (NR), reduced-reference (RR) and full-reference (FR) [30]. Furthermore, NR models can be sub-categorized into pixel based, bitstream based and hybrid (using pixel and bitstream data).

In this paper, we will focus only on pixel-based models. There are open-source implementations available also for bitstream-based model[3], however such models like ITU-T P.1203 [28, 11] are trained for classical video streaming applications with completely different encoding settings, and hence are not directly applicable to gaming QoE.

Specifically for gaming QoE, a few modern video quality models were already analyzed in [5, 2, 3]. Barman et al. [3] analyzed FR algorithms: Peak Signal to Noise Ratio (PSNR), Structural Similarity Index Metric [32] (SSIM) and VMAF [22, 14]. In addition, they analyzed the RR methods STRRED [31] and SpeedQA [1].

In addition, NR models were analyzed [3], e.g., brisque [18], niqe [17] and biqi [21]. The best performing model for the GamingVideoSET [4] was VMAF [3], with a Pearson correlation (PCC) of around 0.9 compared to subjective scores. The performance of all analyzed NR and RR approaches was lower than 0.7 PCC. Especially for classical video quality, other NR models are available that have a similar performance to FR models such as VMAF. Further, Zadtootaghaj et al. [34] developed a NR metric NR-GVQM, where the used feature sets are analyzed, this metric shows promising results, e.g. a PCC of 0.89. Göring et al. [10] trained two no-reference models for classical video quality up to 4K resolution. The first one is a brisque+niqe baseline model trained on per-frame VMAF scores. It shows a good PCC ($\approx 0.85$) with VMAF and subjective scores. This model is similar to the one that we use as baseline model. However, instead of per-frame scores, we use a temporal pooling step before training a machine learning algorithm, see Section III-C. The second metric, called deviq [10], uses a pre-trained classification deep neural network (DNN) to extract several features for each frame using a sub-image approach. The extracted features are used to predict per-frame VMAF scores. Similar to the used brisque+niqe model, deviq shows good performance for VMAF and subjective score prediction with a PCC of $\approx 0.84$. In case of gaming QoE prediction, a classification DNN as used as component in deviq [10] or Venice [7] is not suitable for several reasons. First, it is trained mostly on natural content, e.g. flowers, sunsets, and more, in contrast to the artificial content that is used in games. Even if a retraining or transfer learning of the used classification network is applied, most layers of the DNNs will correspond to natural content. Second, the used sub-image approach and prediction requires a lot of processing time, not available in the context of live monitoring or fast predictions.

To sum up, current state-of-the-art video quality models are not completely suitable for gaming QoE prediction, especially if live monitoring or fast calculations are required. Best performing approaches are FR models. However, there is no actual reference video available in gaming video streaming, because recording is usually done with a lossy setup.

[3]E.g. https://github.com/itu-p1203/itu-p1203

## III. Our No-Reference Approach

To tackle the problem of a fast prediction without reference video, we developed a no-reference video quality model – **nofu**, that we will describe and motivate in the following Section.

In this paper we focus on features that are fast to calculate. Furthermore, we try to use as few features as possible, to be able to have a model that can deliver live predictions. Another method to reduce the processing time is that we use a 360p center crop of the rescaled video sequence. In case of a rescaled input resolution of 1920x1080, a center crop of 640x360 pixels is used. All feature values are calculated based on this center part of the image.

### A. General Video Quality Features

We use some state-of-the-art features that have successfully been applied to video quality classification/prediction [13, 16, 9]. First, we use the spatial information measure ***SI*** based on ITU-T Recommendation P.910 [12]. We use our python implementation of this feature[4]. Spatial information is the standard derivation of the frame after applying a sobel filter.

Furthermore, blurriness is an additional important influencing factor, because of downscaling and subsequent upscaling of the video. To measure blurriness we use an own implementation of a ***fft*** feature, based on [13]. This feature applies FFT on a given image, and counts high-frequency parts in the transformed image.

Motion is another quality aspect, especially for gaming videos, using fast encoding presets. As a first motion feature we use the temporal perceptual information ***TI*** based on ITU-T Recommendation P.910 [12][5]. ***TI*** is the standard deviation of differences at fixed pixel positions between two consecutive frames.

### B. New Features for Gaming Videos

We observed a number of differences between classical video streams and gaming videos. For example, most video games consist of more or less static or constant game elements, e.g. to provide information to the player. To measure the staticness of a video we introduce a feature called ***staticness***.

For a frame $f$ in the video, we calculate the sum of all previous frames and normalize the resulting frame by the number of already shown frames. The summed frame $s$ calculated this way can be seen as a mean value of all previously shown frames. As next step, we calculate the ***SI*** value of the summed frame $s$ and use it as our feature. This feature is based on the assumption that the ***SI*** measure reflects the remaining image information in the summed frame $s$. E.g. in case of a completely static image our summed frame $s$ has a

[4]see https://bit.ly/2oXxQIN

[5]for the implementation see https://bit.ly/2oXxQIN

lot of spatial information. Whereas in case of a video without static content, where $s$ will be mostly blurred, ***SI*** will be low.

Furthermore, due to the fast encoding preset, the video encoder needs to predict movements and compresses I-Frames faster. We observed that such fast encoding introduces a lot of block artifacts. There exist some blockiness measurements, e.g. for images [25, 26]. However, these features were developed for JPEG compression with a fixed block size, and all tested implementations were quite slow. For this reason, we implemented our own blockiness measure that uses some ideas of the papers [25, 26]. For a given frame $f$ of a video, we apply a canny edge detector [6] and get the edges as $e$, where $e$ is a two-dimensional array with $n$ rows and $m$ columns. Here, $e[i,j]$ refers to the $i$th row in the $j$th column. As a next step, for each column $j$ we calculate a value $cs[j] = \frac{1}{n}\sum e[i,j] \forall i$, and for each row $i$ respectively the value $rs[i] = \frac{1}{m}\sum e[i,j] \forall j$. The estimated values $cs$ and $rs$ are column and row summations normalized by the number of rows/columns. Then, for a given blocksize $b$ we estimate for each shift $s \in [0..b]$ the mean value of a subset of $cs$/$rs$. For example, for a shift $s$ we select every $b$th value in $cs$/$rs$ starting from $s$. For such a selection we calculate the mean value. As a result, we obtain mean values for all possible shifts, and we assume that a maximum value of the shifts indicates where possible block artifacts can be found. We measure the difference of this mean value to the selected values in $cs$/$rs$. Using this approach we get values $mD_c$, $s_c$ (where $mD_c$ is the mean difference value for blocksize $b$ using a shift of $s_c$) and $mD_r$, $s_r$. Finally, for a given blocksize we calculate the following value as further measure $\sqrt{|mD_c - mD_r|}/2^{|s_c - s_r|/b}$.

This measure has a larger value if there are block artefacts in the frame. Usually blocks have a square shape, resulting in a measurable difference $mD_c - mD_r$ in both directions x and y. We further normalize the estimated value based on the assumed blocksize and shifts. We repeat the calculation for commonly used blocksizes $b \in [8, 16, 32, 64, 128]$, and the final measure ***blockiness*** is the maximum of all estimated values. Our implemented feature is faster compared to other state-of-the-art methods, however it relies on a fixed block alignment, which is not always the case in videos. We checked the feature with different real world videos, consisting of block artifacts and found out that it represents a good approximation.

Gaming videos are also different in their type of motion. For example, in a strategy game a player can move around the complete area of the game world. To measure motion as a video feature, we started with ideas from Men et al. [16]. Men et al. [16] calculate temporal features based on cuboid slices of the video. Considering that e.g. for full-hd or 4K videos such a feature would require to store all frames to access the cuboid view, we decided to extend and simplify the ideas. First of all, we only consider the first and last rows, referred to as ***cubrow-first***, ***cubrow-last***, and first and last columns, ***cubcol-first***, ***cubcol-last*** as model features. Considering that we use a 360p center crop, the used columns and rows are representing a middle cuboid view of a video. Furthermore, for a given sliding window with $w = 60$ frames we collect all column/row values. For each window $w$ we calculate spatial information of the column/row view of the video over time as final measure for the window. This idea follows the observation that e.g. in case of a static content, all video rows/columns are static, and the cuboid would lead to horizontal lines in case of the temporal view. Those horizontal lines have less spatial information, compared to a more chaotic motion, where the calculated ***SI*** value would be larger. In experiments during development we also selected more rows and columns, however we found two rows and columns to be sufficient to efficiently measure motion.

We further observed that blockmotion artifacts are observed more often in gaming videos. We use a simple blockmotion measure ***blockmotion***. Our feature is based on the blockmotion estimation implementation of scikit-video [6] with the SE3SS [15] method. We use 10% of the video height – in our case of the 360p center crop – as blocksize for blockmotion estimation. After extraction of the moved blocks, we count how often a block is moved left, right, top, down, or not. For each block value we get horizontal and vertical motion in the range [-1,0,1]. We ignore the differentiation between horizontal and vertical and just count how often [-1,0,1] are occurring in our estimation. As feature value we use the normalized -1, 0, 1 counts.

TABLE I
FEATURES THAT ARE USED FOR PREDICTION WITH SOURCES, *img* ARE IMAGE FEATURES, *mov* ARE MOTION-BASED FEATURES

| feature name | *img*/*mov* | source | # values |
|---|---|---|---|
| fft | img | [13] | 1 |
| ti | mov | [12] | 1 |
| si | img | [12] | 1 |
| blockiness | img | own | 1 |
| blockmotion | mov | own | 3 |
| staticness | mov | own | 1 |
| cubrow-first | mov | own | 1 |
| cubrow-last | mov | own | 1 |
| cubcol-first | mov | own | 1 |
| cubcol-last | mov | own | 1 |
| additional | | | |
| niqe | img | [17] | 1 |
| brisque | img | [18] | 36 |

In Table I all features are summarized. In total, we use 10 base features that in sum calculate 12 values per frame. We also added two further features in the table ***niqe*** [17] and ***brisque*** [18]. We will use ***brisque+niqe*** features in our evaluation to train a baseline model for comparison. The feature ***brisque*** consists of luminance-based scene statistics values that quantify possible losses of naturalness [18]. ***niqe*** measures the distance from naturalness using statistical features based on the spatial-domain NSS model [17]. Both features were already successfully used in the gaming video quality context [2].

Furthermore, we evaluated our developed features in several small experiments, where we, e.g., introduced blockiness or static content to videos, to validate the functionality of

[6]http://www.scikit-video.org/stable/

the features. Note that our newly collected features are not restricted to gaming videos only, they can also be used for general video-quality prediction.

### C. Temporal Pooling of Feature Values

For a given video, we are now able to estimate different feature values for each frame. Another step before the calculated values can be used in a final machine learning model is to remove the time dependency, e.g. using temporal pooling methods [29]. We decided to use a simple temporal pooling based on mean and standard derivation.

Assuming we have a given feature array $v$, where one type of feature values of each frame are stored. First, we calculate the mean $mean$ and standard derivation $std$ of all values. In addition, we store the first value $first$ as pooled value. As next step, we divide the per-frame features into 3 equidistant groups. For each group $g = [1, 2, 3]$, we calculate $mean_g$ and standard derivation $std_g$. In total, we calculate 9 values per feature, independent of the duration of the used video sequence. Furthermore, we repeat the pooling for all different feature types that we included in the final model. Considering that we extracted 12 different feature values per frame, we get 108 pooled feature values per video sequence. For comparison, our brisque+niqe baseline model uses 37 feature values per frame, resulting in 333 pooled features per video sequence.

### D. Machine Learning Pipeline

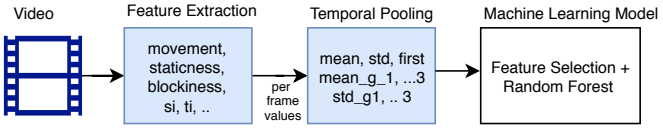


Fig. 1. General Model Structure: Feature extraction for 360p center crop of the rescaled input video, temporal pooling and training of machine learning model (with feature selection).

Starting from the final pooled 108 features per video sequence we train a machine learning model, shown in Figure 1. We considered several machine learning methods during development, e.g. random forest regression, support vector regression (SVR), .... We finally use a random forest regression algorithm, however also SVRs or gradient boosting trees showed similar results. Our general idea of **nofu** is not restricted to the applied machine learning component or specific random forest implementation.

Before the random forest regression step, feature selection using the ExtraTreesRegressor method is applied. For feature selection, we use the $0.5 \cdot mean$ as threshold value. As parameters for the machine learning model we use 10 trees in our random forest, all other parameters are default values. Our implementation is based on python 3, scikit-learn [24], scikit-image[7] and scikit-video[8].

As a comparison model for our evaluation, we use the same model structure and model parameters with ***brisque+niqe*** as features. For both features we use the implementation of scikit-video.

## IV. Evaluation of Proposed Prediction Model

To evaluate our selected features and machine learning pipeline we use the publicly available GamingVideoSET [4][9], that consists of 24 full-hd source videos [4] from different gaming genres. For each experiment we trained two models: *nofu* and *brisque+niqe*. In addition, we also compare our models to other metrics that are included in the GamingVideoSET.

### A. VMAF Predictions as Ground Truth

At first, we consider VMAF as ground truth for our model. Here, we evaluate whether the introduced features and machine learning pipeline are able to predict the FR scores of VMAF in our NR approach. VMAF per-frame scores were already successfully used for video quality prediction [10]. As a pre-processing step, we transform the [0,100]-VMAF scores linearly to a MOS scale of [1,5]. This ensures that we are able to compare it with the MOS values later.

TABLE II
Model Performance Values, VMAF Predictions; 576 videos

| model | pearson | kendall | spearman | rmse |
|---|---|---|---|---|
| nofu | 0.96 | 0.82 | 0.95 | 0.22 |
| brisqueNiqe | 0.94 | 0.80 | 0.94 | 0.24 |
| PSNR | 0.87 | 0.68 | 0.87 | 28.58 |
| SSIM | 0.71 | 0.55 | 0.74 | 2.31 |
| STRRED | -0.53 | -0.42 | -0.61 | 151.44 |
| SpeedQA | -0.55 | -0.45 | -0.63 | 446.75 |

The general performance values for VMAF prediction are summarized in Table II. It can be seen that both models (*brisque+niqe* and **nofu**) generally perform well. However, the **nofu** predictions are still better than the **brisque+niqe** baseline model. We conducted several (64) 10-fold cross validation runs. They all showed similar results, reflected by the statistics of the performance metrics. Our developed model shows a Pearson correlation of $\approx 0.96$, Kendall of $\approx 0.82$ and Spearman with $\approx 0.95$. Furthermore, it has an overall root mean square error (RMSE) of around $0.2$. In comparison, the *brisque+niqe* baseline model yields a slightly lower Pearson correlation of $\approx 0.94$, lower Kendall of $\approx 0.80$ and lower Spearman with $\approx 0.94$. The RMSE of $\approx 0.24$ for the *brisque+niqe* model is higher than with our **nofu** model. Comparing also with other models available from [4], **nofu** and the **brisque+niqe** baseline generally are the best performing models. Surprisingly, PSNR as a non-perceptual metric shows a rather good performance, too. However, a reference video is needed for its calculation.

### B. Subjective Scores as Ground Truth

In our second evaluation experiment, we focus on subjective scores that are available for a subset of the GamingVideoSET [4]. Table III summarizes the performance metrics of all models. In general, it should be mentioned that

[7]https://scikit-image.org/download.html
[8]http://www.scikit-video.org
[9]download https://kingston.box.com/v/GamingVideoSET

TABLE III
MODEL PERFORMANCE VALUES: SUBSET OF GAMINGVIDEOSET; 90 VIDEOS

| model | pearson | kendall | spearman | rmse |
|---|---|---|---|---|
| nofu | 0.91 | 0.75 | 0.91 | 0.42 |
| brisqueNiqe | 0.89 | 0.73 | 0.90 | 0.44 |
| VMAF | 0.86 | 0.69 | 0.86 | 0.64 |
| SSIM | 0.79 | 0.61 | 0.80 | 2.03 |
| PSNR | 0.74 | 0.57 | 0.74 | 29.37 |
| SpeedQA | -0.71 | -0.56 | -0.74 | 488.83 |
| STRRED | -0.72 | -0.55 | -0.74 | 160.48 |

Netflix's VMAF metric is quite good in predicting subjective scores, shown for example in [10] for general video streaming quality. Here, PSNR, SpeedQA and STRRED show the worst results. Their scores are not scaled on a [1,5]-scale, therefore the RMSE values can be ignored.

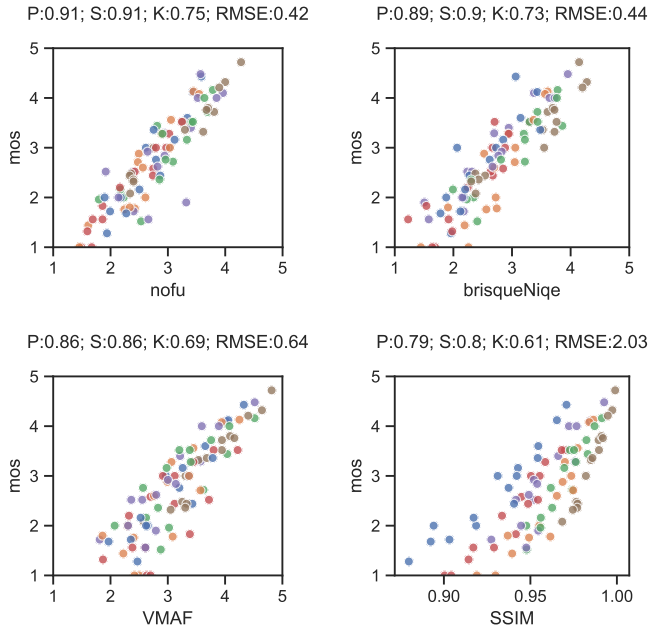


Fig. 2. Scatter plots for top-4 models, colors corresponds to different source videos, SSIM x-axis is not [1,5]-scaled

For a more detailed analysis, we checked scatter correlation plots. The top-4 performing models are shown in Figure 2. Our introduced model **nofu** has a better overall performance than VMAF, further it is also better than our *brisque+niqe* baseline model, as indicated e.g. by the lower RMSE in comparison to VMAF and *brisque+niqe*. Furthermore, Figure 2 also shows that the *brisque+niqe* model has some strengths, e.g. in case of low-quality videos. For further refinement of *nofu* it may be combined with *brisque+niqe*, which we will explore in future experiments. In general, only two no-reference models show a good performance, our new model **nofu** and our introduced baseline *brisque+niqe* model.

### C. Discussion and Performance Analysis

We started with a model to predict VMAF scores on the full set (576) of videos included in the GamingVideoSET [4]. Our introduced features and model **nofu** was shown to provide good performance. Using 10-fold cross validation we found that our model shows a better performance than other metrics. We further used a 360p crop of the recorded videos, to speed up metric calculations. In additional experiments we analyzed other center crops, and found out that 360p is the best trade-off between speed and model accuracy.

In addition, we trained our model to predict the subjective scores included in the employed database and found that also in this case the overall performance of **nofu** is better than for other metrics. Moreover, **nofu** outperforms VMAF, as the best included metric in the GamingVideoSET, in case of prediction subjective ratings, and in addition it does not require any reference video.

**nofu** is also able to outperform the baseline model *brisque+niqe* in both scenarios. However, the two models have a similar performance, indicating that the temporal pooling method used for both delivers good results. Our reduced feature set and minimal pooling strategy ensures that the overall computational requirements of the model are as low as possible. In future experiments, we will evaluate the computation time.

In addition, we performed a source video based train and validation fold approach for subjective score prediction. For the 6 different video sources, we use 5 sources for training and 1 for validation, we calculated correlation values and mean values of all folds. We got Pearson (P) 0.77, Kendall (K) 0.59, and Spearman (S) of 0.75 for our model **nofu**, and P 0.42, K 0.41 and S 0.50 for the **brisque+niqe**. It should be mentioned that such an evaluation scenario is hard, because of the fact that each garming video is from a different gaming genre. Also here **nofu** outperforms **brisque+niqe**.

## V. CONCLUSION

In this paper, we have first discussed the particular charactristics of video streaming quality for gaming sessions. Beside classical video streaming providers, there are platforms for live gaming-video streaming, such as Twitch or YoutubeGaming. In general, gaming videos have different requirements for encoders and are related with different expectations from end users. To measure and predict gaming video quality, we introduced features that are based on already used state-of-the-art features for general video quality, and complementary own video features specifically developed for gaming-video live-streaming evaluation. Our new features capture different aspects of typical gaming videos, such as blockiness, staticness and motion. We further described a quality prediction system using machine learning algorithms. The final model uses a random forest approach with a feature selection pipeline. The developed model – **nofu** – uses a lightweight set of features and a simple-to-calculate temporal pooling to predict per-segment video quality scores, combining fast computation with high accuracy. Our temporal pooling approach is

based on statistical analysis of feature values that change over time. As an additional optimization for computational speed, a 360p center crop of the streamed video is used instead of the full frame. We evaluated our model in two different settings. As a first step, the VMAF scores provided in the GamingVideoSET [4] were used for model training and validation. In a second step, the subjective ratings included in the dataset were used as prediction target. In both settings, our model showed highly accurate predictions in terms of RMSE, Pearson, Kendall and Spearman correlation. When evaluated on subjective ratings, the model outperforms both the FR metric VMAF and a baseline NR metric, *brisque+nique*, trained for comparison. In future experiments, we will evaluate our introduced features in different scenarios, e.g. for classical video streaming, or streaming of 360° video content. Our prototype implementation is written in python 3, representing an easily portable code for a computationally fast algorithm. In future work, a more detailed analysis of computation time will be carried out. Due to its modular design, new features can easily be added, including meta-data or bitstream information as complementary features for an even more accurate gaming-video quality prediction.

## ACKNOWLEDGMENT

This research work was partially funded by Deutsche Telekom AG, Germany.

## VI. REFERENCES

[1] C. G. Bampis et al. "SpEED-QA: Spatial efficient entropic differencing for image and video quality". In: *IEEE Signal Processing Letters* 24.9 (2017), pp. 1333–1337.

[2] N. Barman et al. "A Comparative Quality Assessment Study for Gaming and Non-Gaming Videos". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[3] N. Barman et al. "An evaluation of video quality assessment metrics for passive gaming video streaming". In: *Proceedings of the 23rd Packet Video Workshop*. ACM. 2018, pp. 7–12.

[4] N. Barman et al. "GamingVideoSET: a dataset for gaming video streaming applications". In: *2018 16th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE. 2018, pp. 1–6.

[5] N. Barman and M. G. Martini. "H. 264/MPEG-AVC, H. 265/MPEG-HEVC and VP9 codec comparison for live gaming video streaming". In: *Quality of Multimedia Experience (QoMEX), 2017 Ninth International Conference on*. IEEE. IEEE, 2017, pp. 1–6.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] P. P. Dash et al. "VeNICE: A very deep neural network approach to no-reference image assessment". In: *Industrial Technology (ICIT), 2017 IEEE International Conference on*. IEEE. 2017, pp. 1091–1096.

[8] M. N. Garcia et al. "On the accuracy of short-term quality models for long-term quality prediction". In: *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*. IEEE. 2015, pp. 1–6.

[9] S. Göring et al. "Analyze And Predict the Perceptibility of UHD Video Contents". In: *Electronic Imaging, Human Vision Electronic Imaging* (2019).

[10] S. Göring et al. "DeViQ – A deep no reference video quality model". In: *Electronic Imaging, Human Vision Electronic Imaging* (2018).

[11] ITU-T. *Recommendation P.1203 - Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport*. Tech. rep. International Telecommunication Union, 2016.

[12] ITU-T. *Subjective video quality assessment methods for multimedia applications*. Serie P: Telephone Transmission Quality, Telephone Installations, Local Line Networks. Vol. Recommendation ITU-T P.910. International Telecommunication Union. Geneva, 2008.

[13] I. Katsavounidis et al. "Native resolution detection of video sequences". In: *Annual Technical Conference and Exhibition, SMPTE 2015*. SMPTE. 2015, pp. 1–20.

[14] J. Y. Lin et al. "A fusion-based video quality assessment (fvqa) index". In: *APSIPA, 2014 Asia-Pacific*. Dec. 2014, pp. 1–5.

[15] J. Lu and M. L. Liou. "A simple and efficient search algorithm for block-matching motion estimation". In: *IEEE Transactions on Circuits and Systems for Video Technology* 7.2 (1997), pp. 429–433.

[16] H. Men et al. "Spatiotemporal Feature Combination Model for No-Reference Video Quality Assessment". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–3.

[17] A. Mittal et al. "Making a "completely blind" image quality analyzer". In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212.

[18] A. Mittal et al. "No-reference image quality assessment in the spatial domain". In: *IEEE Trans. Image Process.* 21.12 (2012), pp. 4695–4708.

[19] S. Möller et al. "Towards a new ITU-T recommendation for subjective methods evaluating gaming QoE". In: *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*. 2015, pp. 1–6.

[20] S. Moller et al. "New ITU-T Standards for Gaming QoE Evaluation and Management". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[21] A. K. Moorthy and A. C. Bovik. "A two-step framework for constructing blind image quality indices". In: *IEEE Signal processing letters* 17.5 (2010), pp. 513–516.

[22] Netflix. *Netflix VMAF*. URL: https://github.com/Netflix/vmaf (visited on 07/08/2017).

[23] R. Pantos. *HTTP Live Streaming*. 2011. URL: https://tools.ietf.org/html/draft-pantos-http-live-streaming-13 (visited on 11/01/2018).

[24] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *JMLR* 12 (2011), pp. 2825–2830.

[25] C. Perra. "A low computational complexity blockiness estimation based on spatial analysis". In: *Telecommunications Forum Telfor (TELFOR), 2014 22nd*. IEEE. 2014, pp. 1130–1133.

[26] M. Qadri et al. "Frequency domain blockiness measurement for image quality assessment". In: *Computer Technology and Development (ICCTD), 2010 2nd International Conference on*. IEEE. 2010, pp. 282–285.

[27] W. Robitza et al. "At home in the lab: Assessing audiovisual quality of HTTP-based adaptive streaming with an immersive test paradigm". In: *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*. IEEE. 2015, pp. 1–6.

[28] W. Robitza et al. "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P.1203 Open Databases and Software". In: *9th ACM Multimedia Systems Conference*. Amsterdam, 2018.

[29] M. Seufert et al. ""To pool or not to pool": A comparison of temporal pooling methods for HTTP adaptive video streaming". In: *Quality of Multimedia Experience (QoMEX), 2013 Fifth International Workshop on*. IEEE. 2013, pp. 52–57.

[30] M. Shahid et al. "No-reference image and video quality assessment: a classification and review of recent approaches". In: *EURASIP Journal on Image and Video Processing* 2014.1 (2014), p. 40.

[31] R. Soundararajan and A. C. Bovik. "Video quality assessment by reduced reference spatio-temporal entropic differencing". In: *IEEE Trans. Circuits Syst. for Video Technology* 23.4 (2013), pp. 684–694.

[32] Z. Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Trans. Image Process.* 13.4 (2004), pp. 600–612.

[33] S. Zadtootaghaj et al. "A classification of video games based on game characteristics linked to video coding complexity". In: *2018 16th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE. 2018, pp. 1–6.

[34] S. Zadtootaghaj et al. "NR-GVQM: A No Reference Gaming Video Quality Metric". In: *2018 IEEE Int. Symp. on Multimedia (ISM)*. IEEE. 2018, pp. 131–134.