Video transcoding optimization based on input perceptual quality

Yilin Wang, Hossein Talebi, Feng Yang, Joong Gon Yim, Neil Birkbeck, Balu Adsumilli, and Peyman Milanfar

Google Inc., 1600 Amphitheatre Pkwy., Mountain View, CA, USA 94043

ABSTRACT

Today's video transcoding pipelines choose transcoding parameters based on rate-distortion curves, which mainly focus on the relative quality difference between original and transcoded videos. By investigating the recently released YouTube UGC dataset, we found that human subjects were more tolerant to changes in low quality videos than in high quality ones, which suggests that current transcoding frameworks can be further optimized by considering perceptual quality of the input. In this paper, an efficient machine learning metric is proposed to detect low quality inputs, whose bitrate can be further reduced without sacrificing perceptual quality. To evaluate the impact of our method on perceptual quality, we conducted a crowd-sourcing subjective experiment, and provided a methodology to evaluate statistical significance among different treatments. The results show that the proposed quality guided transcoding framework is able to reduce the average bitrate up to 5% with insignificant perceptual quality degradation.

Keywords: video transcoding, perceptual video quality, user generated contents, machine learning

1. INTRODUCTION

Balancing the trade-off between bitrate and visual quality is a core problem of video compression/transcoding. For large-scale video transcoding systems, it is common to use fixed settings (e.g., bitrates or CRFs) to transcode all videos, which ignores the intrinsic variance across the content. Some recent works[1,2] achieve better performance than fixed settings by optimizing transcoding parameters over the Rate-Distortion (R-D) curves of individual videos. However, existing R-D curves mainly focus on the relative difference (using PSNR or SSIM[3]) between the original and the transcoded versions, whose underlying assumption is that original videos are in pristine quality. For video sharing platforms like YouTube, the majority of uploaded videos are User Generated Content (UGC), which are usually non-pristine. Besides relative quality changes, the original quality becomes another important factor of the perceptual quality for UGC, opening new opportunities for UGC transcoding.

For UGC transcoding, distortions caused by compression do not always negatively impact perceptual quality, especially when the original quality is bad. Fig. 1 compares two 720P UGC videos with different original quality. Here the range of Mean Opinion Scores (MOS) is [1, 5]. We can see for the high quality original, there is no significant change in MOS when transcoding with recommended VP9 settings[5] (CRF=32), and MOS decreases by 0.2 if increasing CRF by 10. However, when the original is of low quality, MOS for all three versions are very close, which implies viewers are not sensitive to compression distortions when the original quality is already below a certain threshold. Inspired by this phenomenon, we proposed a framework called Quality Guided Transcoding (QGT), where videos can be transcoded with more aggressive settings (e.g., increased CRFs) if they are in low quality.

Exploring original quality brings a new viewpoint to UGC transcoding. Many practical problems will be addressed in this paper, like how to define low quality, how to evaluate model accuracy, and how to design criteria for insignificant quality changes. The major contributions of this work are as follows:

- An efficient machine learning based metric is proposed, which achieves good performance on detecting low quality UGC videos. We also discuss how to treat quality prediction as a classification problem, and calculate reasonable precision and recall (Section 3).

E-mail: {yilin,htalebi,fengyang,joonggonyim,birkbeck,badsumilli,milanfar}@google.com

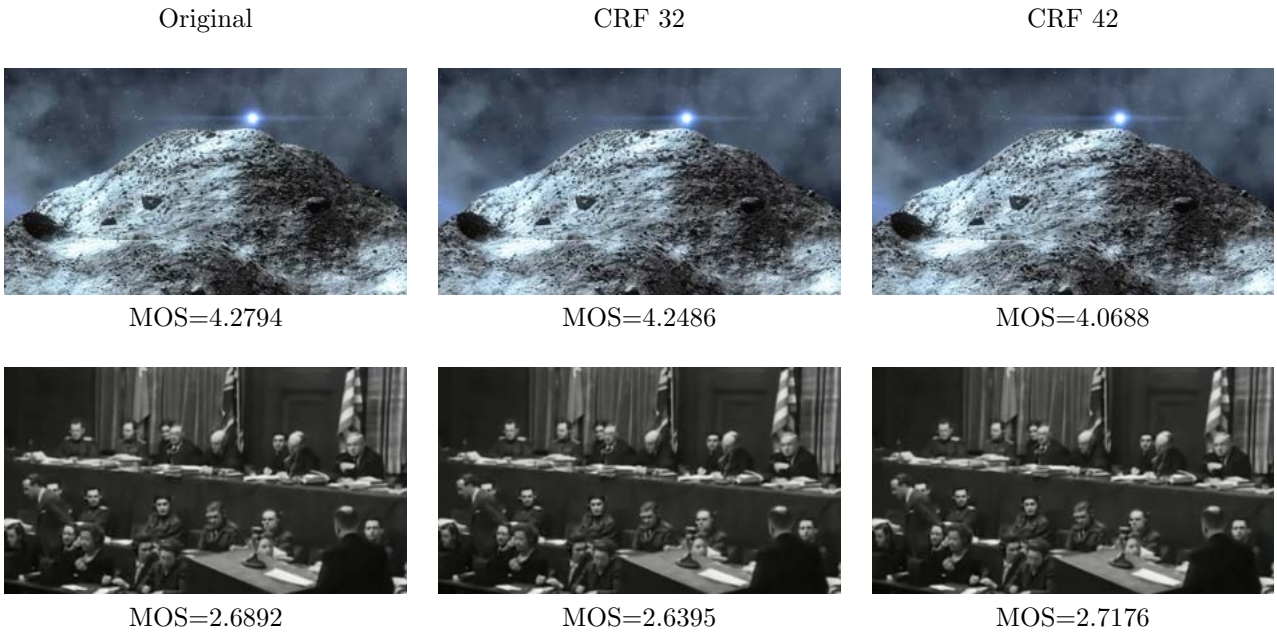

Figure 1. Viewers are less sensitive to quality changes in low quality videos than in high quality videos. Left: original videos with high and low perceptual quality. Middle: videos transcoded by VP9 with recommended settings (CRF 32). Right: videos transcoded with increased CRF 42. Sample videos are selected from YouTube UGC dataset[4](vid: Animation_720P-06a6 and NewsClip_720P-35d9). MOS are collected by our subjective experiment.

- We designed a framework to guide transcoding based on input quality, and proposed new transcoding parameters for low quality videos (Section 4).
- We conducted a crowd-sourced subjective experiment to validate our QGT approach, and provide a general methodology to evaluate significant difference between variants (Section 5).

2. RELATED WORK

Perceptual quality for UGC videos is a broad concept. Besides compression artifacts, distortions introduced during the process of video production (like lens blur and camera shake) could also influence viewers' watching experience. Traditional public video quality datasets (e.g., LIVE datasets[6–8]) mainly focus on compression distortions introduced to pristine originals, which contain very limited UGC features. Some public UGC datasets, like YouTube-8M[9] and AVA,[10] only provide extracted features instead of raw pixel data, making them less useful for compression research. Large-scale UGC quality datasets[4,11,12] were released in the past two years, which provide both raw videos and MOS collected from crowd-sourcing platforms. Within these datasets, YouTube's UGC dataset (YT-UGC)[4] is one of the most representative ones. The YT-UGC dataset is sampled from 1.5 million YouTube videos, contains 1500 20-second video clips, covering 15 categories (e.g., Gaming, Sports, and Music Video) and various resolutions (from 360P to 4K), making it a good basis for research on the practical applications of video compression and video quality assessment.

Video quality assessment has been studied for decades and is still a challenging research topic. Reference quality metrics (e.g., PSNR, SSIM, and VMAF[13]) are designed for measuring relative quality changes from the reference (pristine original), and are not suitable for estimating original quality. Since traditional no-reference metrics[14–18] mainly rely on manually designed features that are summarized from limited samples, they don't perform well on various UGC conditions. Recent machine learning based metrics[19,20] achieved significant improvements, benefiting from models pretrained on large scale datasets (e.g., ImageNet). However, most existing metrics focus on the accuracy across the full quality range, which is slightly different from our use case (i.e.,

only focusing on low quality boundary). Also, machine learning based metrics are usually computationally expensive (based on Deep Neural Networks, DNN), making them inefficient to be used as a video quality metric that performs on every frame. For our use case, high precision on low quality detection is more important than high correlation in the full quality range, which allows us to build a lightweight model that is fast enough to investigate every video frame.

3. LOW QUALITY DETECTOR

3.1 Low Quality Detector Training

The hypothesis of quality guided transcoding is that viewers have different sensitivities to compression distortions in videos with different input quality. The first open question is how to classify videos into two groups: low quality and non-low quality, which requires a no-reference quality metric. As discussed in Section 2, traditional no-reference metrics do not perform well on UGC videos, and the existing, computationally expensive DNN-based metrics are not sufficient for this low quality detection task (which is basically a binary classification problem). Thus we propose a lightweight model consisting of the first three layers of Inception-v2 model[21] (a 7×7 convolution layers with stride=2, a max pooling layer with stride=2, and then a 1×1 layer as shown in Fig. 2), followed by a spatial pyramid pooling (SPP)[22] layer, and a fully-connected (FC) layer. The predicted score (low quality probability) is normalized to $[0, 1]$ by a sigmoid function. Note that because of the SPP layer, the model can take RGB images with arbitrary input resolutions. In our model we implemented 4 scales of the SPP with average pooling. At training and testing we resize images to 256×192 with proper rotations to resize the largest dimension to 256. The proposed Tensorflow implementation of our model contains nearly 12 thousand parameters and takes 90 million flops to run on an input image. When compiled for 64-bit intel architectures with AVX, the resultant model binary is only 180KB, and can process 100 fps for 1080p on a single skylake core, which is efficient enough for large scale deployment.

Due to limited UGC datasets with associated mean opinion scores, our model was developed through transfer learning. The model weights were initialized by an image classification Inception-v2 model (using the first three layers) trained on JFT Dataset (300M images),[23] which was used to extract basic image features. The initial model was then trained on the Visual Quality Dataset[24] (250K image, pairwise ranks collected for 3.2M pairs), which helped the model to learn quality related features. The obtained model is then fine-tuned on our own pairwise UGC dataset (17K image pairs) to optimize its sensitivity to UGC compression artifacts.

Our own UGC dataset consists of two parts. The first one is 1500 frames extracted from YT-UGC[4] (one random frame from each video), where each frame is randomly paired with 7 frames from other videos. We showed these side-by-side frames to human subjects and asked for their preference (either $A > B$ or $A < B$). Each pair was rated by three domain experts, and their average rating was used as the final pairwise rank value. The second set is an automatically generated pairwise set, where another 1000 frames extracted from YT-UGC dataset are compressed into three different quality levels (Q1, Q2, Q3). The default order of quality is: Original >= Q1 >= Q2 >= Q3. Each original frame provides 6 pairs of frames, so in total we have 6000 automatically generated pairs. There is more than one way to generate such compression variants, and in this paper we compress the original with the H264 codec using CRF values of 30, 35, and 40. The purpose of the first dataset is to help the model learn UGC style features, and the second dataset forces the model to be more sensitive to compression distortions. The final training set contains 17K frame pairs.

The model was trained by a Siamese network (Fig. 2), where a pair of images (x_i, x_j) are fed into networks that shared the same weights to get corresponding predicted low quality probabilities y_i and y_j. Besides their pairwise rank $p_{i,j}$, we also estimated global rank scores (s_i and s_j) for the two images based on iterative rank aggregation.[25] The overall loss is the sum of pairwise loss and rank loss:

$$loss(x_i, x_j) = \underbrace{-p_{i,j} \log(y_i - y_j) - (1 - p_{i,j}) \log(1 - (y_i - y_j))}_{\text{pairwise loss}} + \underbrace{(s_i - y_i)^2 + (s_j - y_j)^2}_{\text{rank loss}}.$$

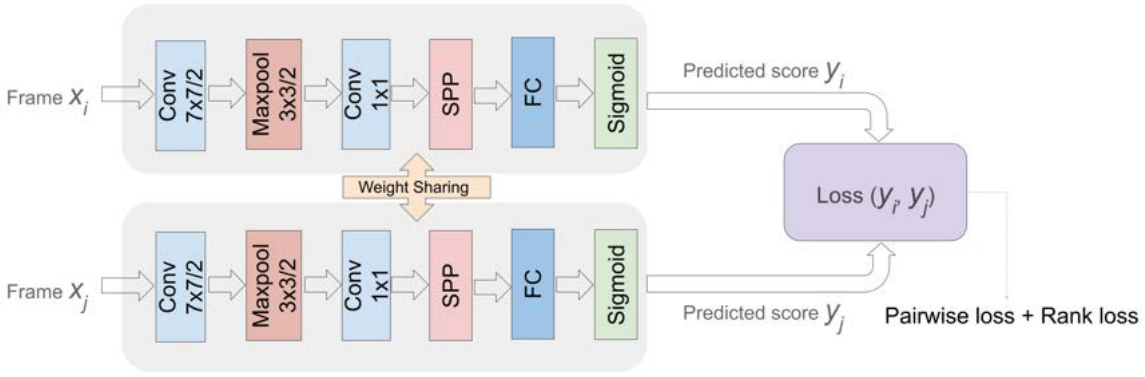

Figure 2. The training model based on a Siamese network. Our lightweight model consists of the first three layers of Inception-v2 model.

3.2 Evaluation on UGC Low Quality Detection

We split the created pairwise UGC dataset into training and test sets by 80/20, and the final model achieved 84% on predicting the pairwise rank on the test set. Fig. 3 shows sample images with various low quality probabilities, where we can see predicted scores match the decreasing trend of the perceptual quality. We also found that the image quality became very bad when the score was greater than 0.8.

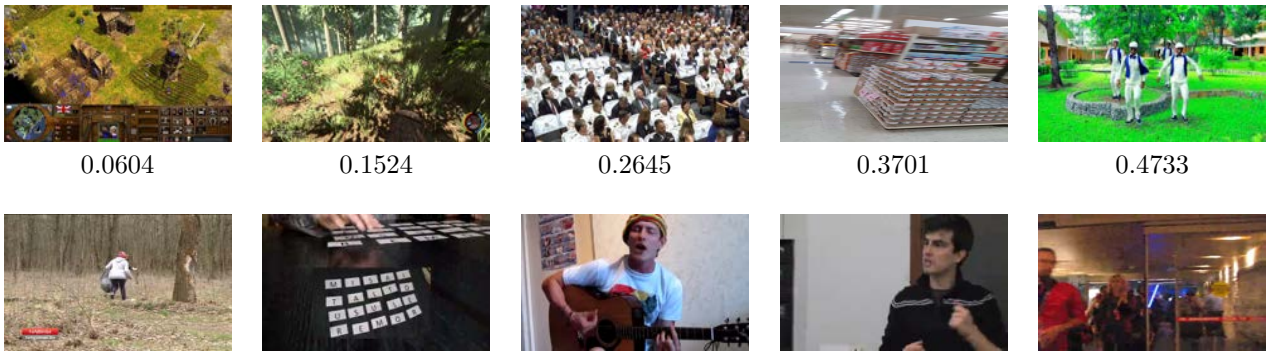

Figure 3. Samples with different predicted low quality probabilities.

We also evaluated the proposed metric with YT-UGC MOS data,[26] where the MOS range is [1, 5] where 1 means the worst quality and 5 means excellent quality. Since our metric focused on low quality detection instead of predicting quality in the full range, we did some pre-processing of the data. Videos were grouped into three quality levels: low, medium, and high, where MOS 3.0 and 3.8 were set as bars for low and high quality respectively. Based on this classification, there are 20% low quality, 41% medium quality, and 39% high quality videos in the YT-UGC dataset. In practice, the boundary between low and medium quality is usually ambiguous, and the potential risk of mislabeling "medium" as "low" is lower than mislabeling "high" as "low", thus we proposed the concept of "weighted precision" and "weighted false positive" for this low quality detection task, where we assigned a weight m ($\in [0, 1]$) for the count of true "medium" predicted as "low", where 0 means only considering true "high" in false positive computation, and 1 means treating "medium" equally as "high" in false positive computation. The calculation of recall is not affected by this weight. The following metrics are

used to evaluate the low quality detector:

$$\begin{aligned} \text{Precision} &= \frac{\text{pred}_{l,l}}{\text{pred}_{l,l} + m \cdot \text{pred}_{l,m} + \text{pred}_{l,h}} \\ \text{Recall} &= \frac{\text{pred}_{l,l}}{\text{true}_l} \\ \text{FalsePositive} &= \frac{m \cdot \text{pred}_{l,m} + \text{pred}_{l,h}}{\text{pred}_{nl,nl} + m \cdot \text{pred}_{l,m} + \text{pred}_{l,h}}, \end{aligned}$$

where l, m, h, and nl denote "low", "medium", "high", and "not low" (="medium"+"high"), $\text{pred}_{x,y}$ is the count of true y predicted as x. true_x is the count videos with true x.

Fig. 4 showes weighted precision, recall, and weighted ROC for the proposed low quality metric. In our application, controlling the impact of mislabeling is more important than bitrate saving (i.e., precision is more important than recall), so we set the low quality threshold $T_q = 0.8$, whose corresponding weighted precision is high (0.86 for $m = 0$ and 0.79 for $m = 0.2$), while recall is relatively low (0.27) but still acceptable. We also compares the proposed metric with three popular no-reference metrics: BRISQUE,[14] NIQE,[15] and VIIDEO.[16] Since they are also frame-based metrics, the average frame score is used as the final score for the entire video. Fig. 5 compared their weighted precision, recall, and ROC with $m = 0$. We can clearly see that existing no-reference metrics don't perform well for such low quality detection tasks, and the proposed metric outperforms the other three metrics in all aspects.

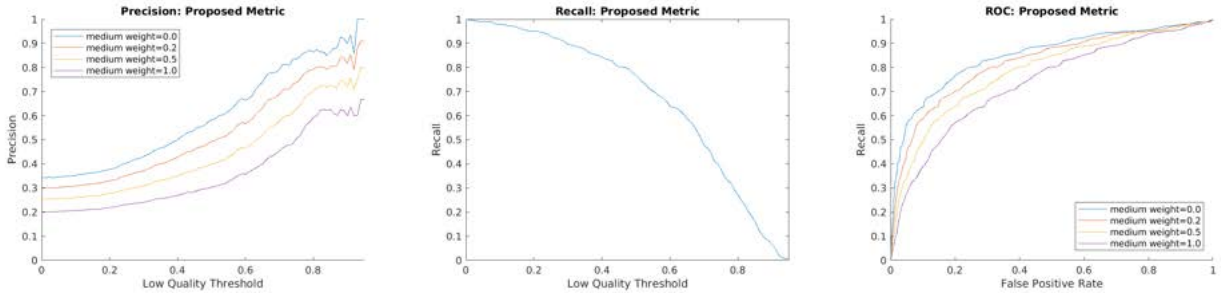

Figure 4. Weighted precision, recall, and ROC for the proposed metric.

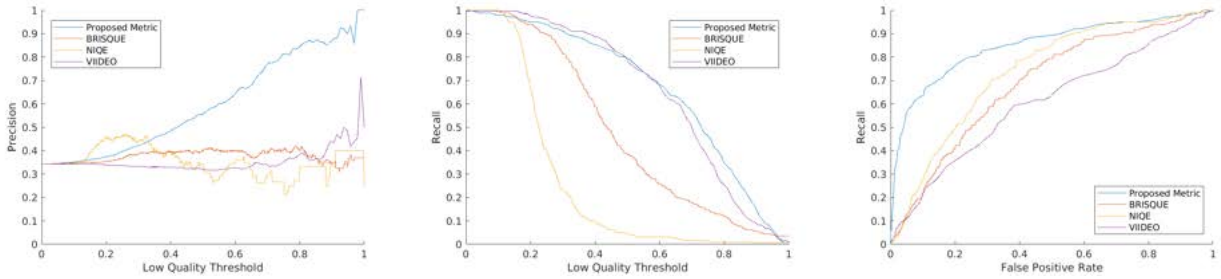

Figure 5. Comparison of weighted precision, recall, and ROC for BRISQUE, NIQE, VIIDEO, and the proposed metric with weight $m = 0$. All metric scores are normalized to [0, 1]. Similar results were observed for other weights.

4. QUALITY GUIDED TRANSCODING FRAMEWORK

With the efficient low quality detector, we propose a framework called Quality Guided Transcoding (QGT, as shown in Fig. 6): the input video is first split into multiple disjoint 5 second chunks, then each chunk is evaluated by the low quality detector independently. If the predicted low quality probability is greater than the pre-defined

threshold ($T_q = 0.8$ in this paper), the CRF is increased by X and other parameters (like max and min bitrates) are kept the same; otherwise, the default CRF is used to transcode the chunk. All transcoded chunks are then merged together to get the final transcoded video.

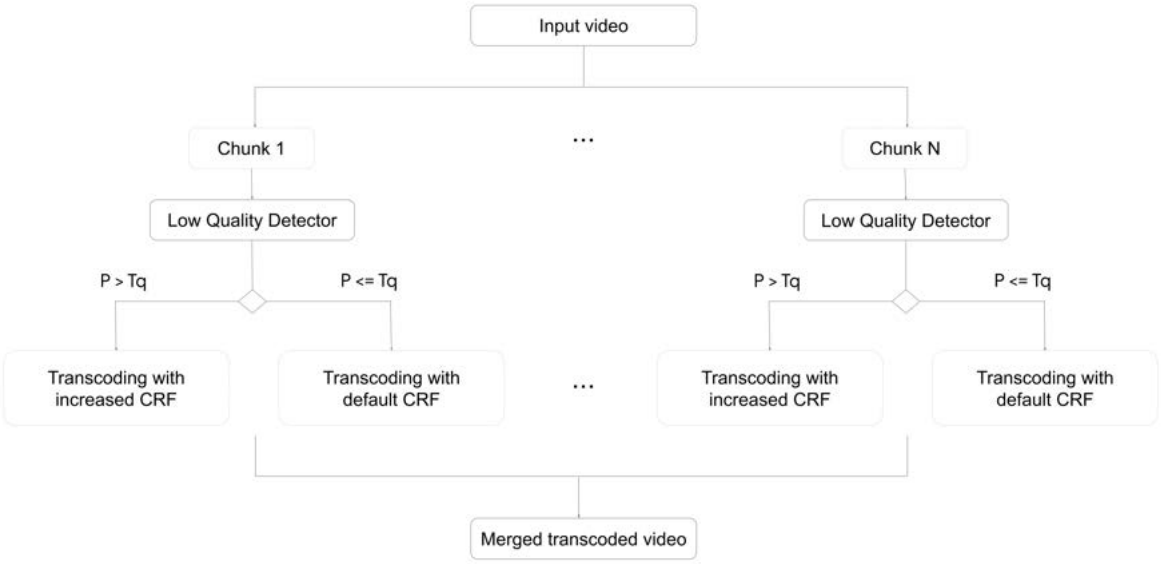

Figure 6. Framework of Quality Guided Transcoding.

To evaluate the performance of QGT framework, we selected all videos (excluding 4k) from the YT-UGC dataset, divided them into 5 second chunks, and compressed these videos in their native resolutions by VP9 with recommended 2 pass settings,[5] where CRF(CQ)=36, 34, 32, and 31 for 360P, 480P, 720P, and 1080P respectively. Three QGT treatments were evaluated, whose CRFs were increased by 10, 15, and 20 from default values. Table 1 showed detected low quality ratios and file size savings for different resolutions. We can see low resolution originals (360p and 480p) have a larger ratio of low quality chunks than higher resolution originals (720p and 1080p), which matches the common sense of video resolutions (i.e., higher resolution usually has better quality). In general, applying QGT can reduce file size (or bitrate) by 4 to 8% for 360P and 1 to 2% for 1080P videos, which are remarkable savings for large scale video sharing platforms like YouTube. However, we cannot arbitrarily increase CRF because viewers would be annoyed by noticeable quality degradation, which should be avoided. We need a well defined methodology to identify optimal settings for QGT.

Table 1. Performance of QGT on YouTube UGC dataset

Input resolution	360P	480P	720P	1080P
Total 5s chunks	1030	1234	1253	1632
Detected low quality chunk ratio	24%	18%	10%	9%
VP9 default CRF total size (in MB)	179.8	374.9	926.1	2322.8
QGT (CRF+10) total size (saving ratio)	172.7 (-3.9%)	362.1 (-3.4%)	905.3 (-2.2%)	2294.6 (-1.2%)
QGT (CRF+15) total size (saving ratio)	169.7 (-5.5%)	355.5 (-5.1%)	894.3 (-3.4%)	2278.5 (-1.9%)
QGT (CRF+20) total size (saving ratio)	165.4 (-7.9%)	348.0 (-7.1%)	886.3 (-4.2%)	2266.4 (-2.4%)

5. OPTIMAL SETTINGS FOR QGT

Identifying quality criteria for video transcoding is tricky. It is difficult to directly map quality changes to bitrate savings, or vice versa. In this paper, we treated the version transcoded with default VP9 two-pass settings as the base, and a valid treatment should be statistically insignificant from the base.

5.1 Subjective Data Collection

We grouped YT-UGC videos (all in 20s, i.e, 4×5s chunks) into three quality conditions: low (4 low quality chunks), medium (2 low quality chunks), and high (no low quality chunks), based on the predicted low quality probabilities. Then for each resolution (360P, 480P, 720P, and 1080P) we selected 10 videos for each quality condition. In total there are 4 resolutions $\times 3$ quality conditions $\times 10$ videos $= 120$ test videos. Within each subset, we also tried to include as many content categories as possible, although some categories (e.g., 1080P Gaming) had no low quality samples.

Original videos were transcoded at their native resolution with default (recommended) settings for VP9, as well as CRF increased by 10, 15, and 20. There are $5 \times 120 = 600$ video clips in total (original, default, CRF+10, CRF+15, and CRF+20), and corresponding subjective scores (5 quality levels: bad, poor, fair, good, excellent) were collected by our crowd-sourcing platform.[26] Each subject received about 80 randomly selected videos for rating, and finally each clip was rated by 30 to 60+ subjects.

Fig. 7 compares MOS between variants. We can see that the difference between original and default versions (top left) are relatively small, since most nodes (in all three quality conditions) are roughly equally distributed along the diagonal line. In the other three cases, the default versions in high quality conditions are clearly better than versions transcoded with increased CRFs (below the diagonal line), which implies the current default CRFs are already optimal for high quality videos. However, when looking at low and medium quality conditions, we can see MOS between default and CRF+10 versions (top right) are also very close, which suggests that it is promising to increase CRFs for those videos without causing noticeable quality degradation. Such results match the hypothesis of QGT. Since most CRF+20 versions have lower MOS than corresponding default versions (bottom right), which doesn't meet QGT quality criterion, it is not a valid candidate. Now there are two potential treatments (CRF+10 and CRF+15), and we need a more precise approach to validate them.

5.2 Statistical Significance

In this section, we use the Bootstrap Hypothesis Test to evaluate the three proposed CRF values for QGT. Bootstrap[27] is a classical statistical technique for small datasets that provides unbiased estimate that some attribute of them differ.

Suppose the entire subjective test included M individual videos, each of them has multiple variants (e.g., original, default, crf+10, etc). Test videos are grouped into three classes (low, medium, high) based on their upload quality. N individual subjects finished the test, and each of them only rated a subset (e.g., 80) of the entire set. The same subject can see multiple variants from the same content.

Let q_{c_i,v_j,s_k} be the quality score for the jth variant (treatment) of the ith content rated by subject k. To estimate the difference between variant v_x and v_y on a video set, we perform the following sequence of steps:

- Step 1: build a set $\mathbb{P}$ that includes all valid $< c_i,\ s_k >$ pairs where subject s_k has rated both variant v_x and v_y for video c_i. The size of $\mathbb{P}$ is L.
- Step 2: compute paired differences for all pairs in $\mathbb{P}$:

$$\mathbb{D}_{\text{raw}} = \{(q_{c_i,v_x,s_k} - q_{c_i,v_y,s_k}) \mid \forall < c_i,\ s_k > \in \mathbb{P}\},$$

- Step 3: compute mean and t-score for $\mathbb{D}$:

$$\begin{aligned} \bar{d}_{\text{raw}} &= \sum_{d \in \mathbb{D}_{\text{raw}}} d/L, \\ t_{\text{raw}} &= \frac{\bar{d}_{\text{raw}}}{\sigma(\mathbb{D}_{\text{raw}})/L}, \end{aligned}$$

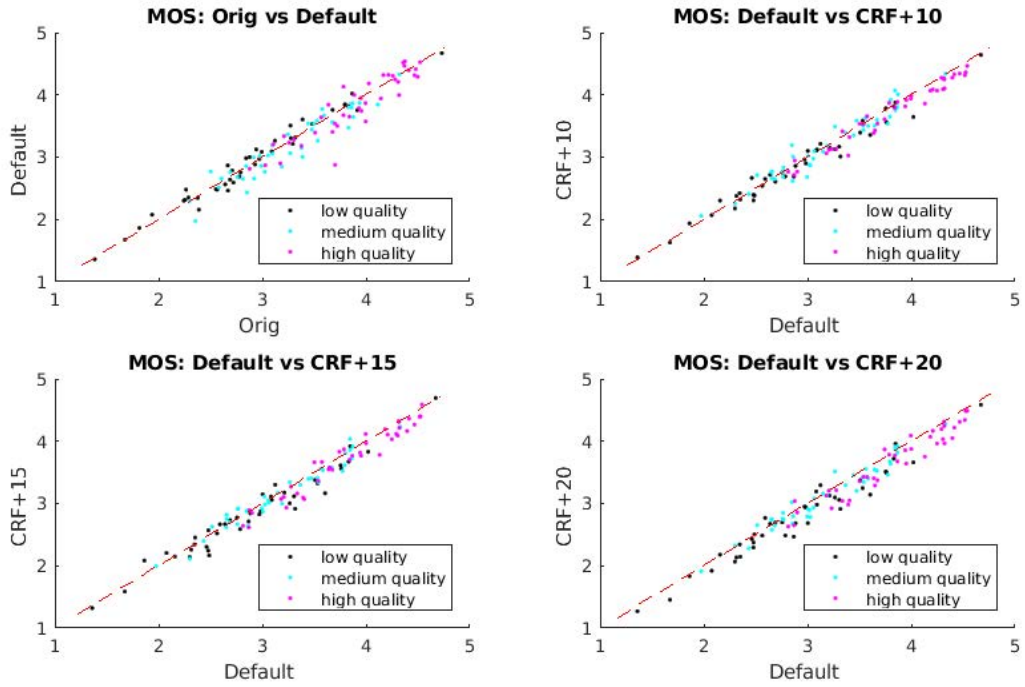

Figure 7. Pairwise MOS comparisons for different variants. Nodes denote individual videos in three different quality conditions (low, medium, and high). The dash line (in red) means $y = x$.

- Step 4: build a set $\mathbb{D}_i$ by independently randomly selecting L elements from $\mathbb{D}_{raw}$ (allowing duplicates of the same element), and compute the sample mean shifted by $\bar{d}_{\text{raw}}$ and corresponding t-score:

$$\begin{aligned} \bar{d}_i &= \sum_{d \in \mathbb{D}_i} d/L, \\ t_i &= \frac{\bar{d}_i - \bar{d}_{\text{raw}}}{\sigma(\mathbb{D}_i)/L}, \end{aligned}$$

- Step 5: repeat Step 4 K (e.g., 10000) times to get bootstrap mean and a set of t-scores ($\mathbb{T}_{\text{boot}}$):

$$\begin{aligned} \text{mean}_{\text{boot}}(v_x, v_y) &= \frac{(\bar{d}_1 + \bar{d}_2 + \cdots + \bar{d}_K)}{K}, \\ \mathbb{T}_{\text{boot}} &= \{t_1, t_2, \ldots, t_K\}, \end{aligned}$$

- Step 6: compute Achieved Significance Level (ASL) for t statistics:

$$\begin{aligned} \text{ASL}_{\text{boot}}(v_x, v_y) &= \sum_i (t_i \geq t_{\text{raw}})/K, \text{ if } t_{\text{raw}} \geq 0, \\ &\quad \sum_i (t_i \leq t_{\text{raw}})/K, \text{ if } t_{\text{raw}} < 0. \end{aligned}$$

Two statistics $\text{mean}_{\text{boot}}$ and ASL_{boot} are used as our quality criteria. $\text{mean}_{\text{boot}}(v_x, v_y) > 0$ means variant/treatment v_x tends to generate videos with better quality than v_y does, and vice versa. $\text{ASL}_{\text{boot}} < 0.05$ means v_x and v_y are significantly different (roughly corresponding to 95% probability that v_x and v_y are different).

Summary 1: For low quality videos, it is safe to increase CRF by 10. As the histogram for the difference between the original version and variants (default, CRF+10, CRF+15, CRF+20) for low quality videos

Table 2. Difference between the original version and variants for videos in low, medium, and high quality conditions, where default and CRF+10 versions could have slightly better quality than the original version for low quality videos.

Original	v.s. Default	v.s. CRF+10	v.s. CRF+15	v.s. CRF+20
low quality	mean=-0.024 ASL=0.054	mean=-0.005 ASL=0.367	mean=0.009 ASL=0.300	mean=0.031 ASL=0.019
medium quality	mean=0.049 ASL=0.001	mean=0.046 ASL=0.003	mean=0.059 ASL=0.000	mean=0.069 ASL=0.000
high quality	mean=0.064 ASL=0.000	mean=0.097 ASL=0.000	mean=0.109 ASL=0.000	mean=0.131 ASL=0.000

shown in Table 2 (first row), mean values $\text{mean}_{\text{boot}}(orig, default)$ and $\text{mean}_{\text{boot}}(orig, CRF+10)$ are negative, which means using default setting or CRF+10 actually gives better quality than the original version (as the example shown in Fig. 8). For other cases, the original version always has better quality than the variants.

Original Default CRF+10

Figure 8. For low quality videos, the transcoded version can have better perceptual quality than the original version. In this example, the original frame has noticeable noise, while the transcoded versions look smoother. Corresponding MOS for original, default, and CRF+10 are 1.93, 2.07, and 2.06 respectively.

Summary 2: It is promising to increase CRF by 15 for QGT affected videos (low and medium quality). Table 3 shows the difference between the default version and three variants for videos in low, medium, and high quality conditions. For the high quality case, all variants have significant difference from the default version ($\text{ASL}_{\text{boot}} < 0.05$), which validated that default setting is already optimal for high quality videos. For low and medium quality videos, the differences between default and CRF+10 are insignificant, where $\text{ASL}_{\text{boot}}(default, CRF+10)$ are 0.098, and 0.420 respectively. CRF+15 and CRF+20 are also indistinguishable from default version in medium quality case, but significantly different in low quality case.

Although CRF+15 is significantly different from default in low quality case, it is still promising to apply it. The reason is that according to Table 2, the default version actually has slightly better quality than the original version for low quality videos, while CRF+15 is indistinguishable from the original version ($\text{ASL}_{\text{boot}}(orig, CRF+15)$=0.375).

To conclude, CRF+10 is a safe choice for QGT framework. CRF+15 is also a reasonable choice that may cause slightly quality degradation but still in the acceptable range.

Discussion of methodology. A 0.1 difference in individual scores may not be distinguishable due to measurement accuracy. However, the differences we presented in this paper are the delta between average of means of 10000 Bootstrap samples, where each sample has about 40 videos, or over 1000 scores (in total 10M scores). A difference of 0.02 presented above means that the majority of 10M scores in one variant is 0.02 higher than in the other variant, which is sufficient to make reliable conclusions.

Table 3. Difference between default and variants (CRF+10, CRF+15, and CRF+20) for videos in low, medium, and high quality conditions.

Default	v.s. CRF+10	v.s. CRF+15	v.s. CRF+20
low quality	mean=0.019 ASL=0.098	mean=0.033 ASL=0.017	mean=0.055 ASL=0.000
medium quality	mean=-0.003 ASL=0.420	mean=0.010 ASL=0.256	mean=0.020 ASL=0.099
high quality	mean=0.032 ASL=0.015	mean=0.044 ASL=0.001	mean=0.066 ASL=0.000

6. CONCLUSION

In this paper, we proposed quality guided transcoding as a technique to optimize transcoding by considering the original video quality. We demonstrated that the bitrate of low quality input videos can be reduced without sacrificing perceptual quality. We discussed how to build and evaluate a low quality detector on UGC data, and how to chose optimal settings. A novel subjective test methodology based on bootstrap was introduced to analyze differences between three transcoding treatments. We hope this work can bring new viewpoints and inspire additional optimizations for UGC transcoding.

REFERENCES

[1] De Cock, J., Li, Z., Manohara, M., and Aaron, A., "Complexity-based consistent-quality encoding in the cloud," in [*2016 IEEE International Conference on Image Processing (ICIP)*], (2016).

[2] Chen, C., Lin, Y., Benting, S., and Kokaram, A., "Optimized transcoding for large scale adaptive streaming using playback statistics," in [*2018 25th IEEE International Conference on Image Processing (ICIP)*], (2018).

[3] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**(4) (2004).

[4] Wang, Y., Inguva, S., and Adsumilli, B., "Youtube ugc dataset for video compression research," in [*2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*], (2019).

[5] "Recommended settings for vp9 vod." https://developers.google.com/media/vp9/settings/vod.

[6] Seshadrinathan, K., Soundararajan, R., Bovik, A. C., and Cormack, L. K., "Study of subjective and objective quality assessment of video," *IEEE Transactions on Image Processing* **19**(6) (2010).

[7] Bampis, C. G., Li, Z., Moorthy, A. K., Katsavounidis, I., Aaron, A., and Bovik, A. C., "Study of temporal effects on subjective video quality of experience," *IEEE Transactions on Image Processing* **26**(11) (2017).

[8] Ghadiyaram, D., Pan, J., and Bovik, A., "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Transactions on Circuits and Systems for Video Technology* **29**(1) (2017).

[9] Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S., "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675* (2016).

[10] Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., and Malik, J., "Ava: A video dataset of spatio-temporally localized atomic visual actions," *Proceedings of the Conference on Computer Vision and Pattern Recognition* (2018).

[11] Sinno, Z. and Bovik, A. C., "Large scale subjective video quality study," *IEEE International Conference on Image Processing* (2018).

[12] Hosu, V., Hahn, F., Jenadeleh, M., Lin, H., Men, H., Szirányi, T., Li, S., and Saupe, D., "The konstanz natural video database (konvid-1k)," in [*2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*], 1–6, IEEE (2017).

[13] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M., "Toward a practical perceptual video quality metric," *Blog, Netflix Technology* (2016).

[14] Mittal, A., Moorthy, A. K., and Bovik, A. C., "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing* (2012).

[15] Mittal, A., Soundararajan, R., and Bovik, A. C., "Making a completely blind image quality analyzer," *IEEE Signal Processing Letters* **22**(3) (2013).

[16] Mittal, A., Saad, M. A., and Bovik, A. C., "A completely blind video integrity oracle," *IEEE Transactions on Image Processing* (2016).

[17] Wang, Y., Kum, S.-U., Chen, C., and Kokaram, A., "A perceptual visibility metric for banding artifacts," *IEEE International Conference on Image Processing* (2016).

[18] Chen, C., Izadi, M., , and Kokaram, A., "A no-reference perceptual quality metric for videos distorted by spatially correlated noise," *ACM Multimedia* (2016).

[19] Lin, K. and Wang, G., "Hallucinated-iqa: No-reference image quality assessment via adversarial learning," in [*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*], (2018).

[20] Li, D., Jiang, T., and Jiang, M., "Quality assessment of in-the-wild videos," in [*Proceedings of the 27th ACM International Conference on Multimedia*], (2019).

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the inception architecture for computer vision," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (2016).

[22] He, K., Zhang, X., Ren, S., and Sun, J., "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence* **37**(9), 1904–1916 (2015).

[23] Sun, C., Shrivastava, A., Singh, S., and Gupta, A., "Revisiting unreasonable effectiveness of data in deep learning era," in [*Proceedings of the IEEE international conference on computer vision*], 843–852 (2017).

[24] Talebi, H., Amid, E., Milanfar, P., and Warmuth, M., "Rank-smoothed pairwise learning in perceptual quality assessment," in [*2020 IEEE International Conference on Image Processing (ICIP)*], (2020).

[25] Negahban, S., Oh, S., and Shah, D., "Iterative ranking from pair-wise comparisons," in [*Proceedings of the 25th International Conference on Neural Information Processing Systems*], (2012).

[26] Yim, J., Wang, Y., Birkbeck, N., and Adsumilli, B., "Subjective quality assessment for youtube ugc dataset," in [*2020 IEEE International Conference on Image Processing (ICIP)*], (2016).

[27] Efron, B. and Tibshirani, R., "An introduction to the bootstrap," (1993).