Netflix Technology Blog Follow
Learn more about how Netflix designs, builds, and operates our systems and engineering organizations
Oct 25 · 15 min read

VMAF: The Journey Continues

by Zhi Li, Christos Bampis, Julie Novak, Anne Aaron, Kyle Swanson, Anush Moorthy and Jan De Cock

How will Netflix members rate the quality of this video—poor, average or excellent?
Which video clip looks better—encoded with Codec A or Codec B?
For this episode, at 1000 kbps, is it better to encode with HD resolution, with some blockiness, or will SD look better?

These were example questions we asked ourselves as we worked towards delivering the best quality of experience for Netflix members. A few years ago, we realized that we were unable to answer these questions effectively by simply relying on "golden eyes." Expert viewing was not scalable across content, encoding recipes, and the overall output of our encoding pipeline. It was possible to deploy existing video quality metrics, such as PSNR and SSIM at scale, but they fell short of accurately capturing human perception. Thus, we embarked on a journey to develop an automated way to answer the question, "How will a Netflix member rate the quality of this encode?" This was the birth of VMAF.

Video Multi-method Assessment Fusion, or VMAF for short, is a video quality metric that combines human vision modeling with machine learning. The project started as research collaboration between our team and Prof. C.-C. Jay Kuo from University of Southern California. His research group had previously worked on perceptual metrics for images, and together, we worked on extending the ideas to video. Over time, we have collaborated with other research partners such as Prof. Alan Bovik from the University of Texas at Austin and Prof. Patrick Le Callet from Université de Nantes with the goal of improving VMAF accuracy related to human subjective perception, and broaden its scope to cover more use cases. In June 2016, we open-sourced VMAF on Github, and also published the first VMAF techblog. In this new techblog, we want to share our journey.

Industry Adoption

Outside of Netflix, the video community is finding VMAF a valuable tool for quality assessment. Because of industry adoption, the project is benefitting from broader contribution from researchers, video-related companies and the open-source community.

- VMAF has been integrated into 3rd-party video analysis tools (for example, FFmpeg, Elecard StreamEye, MSU Video Quality Measurement Tool and arewecompressedyet), putting it side-by-side with more established metrics such as PSNR and SSIM.
- In industry trade shows and meet-ups such as NAB, Video@Scale and Demuxed, demos and presentations are given using VMAF scores to compare quality and efficiency of various encoding techniques.
- The Video Quality Experts Group (VQEG) is an international consortium of video quality assessment experts. In the recent Los Gatos, Krakow and Madrid VQEG meetings, VMAF was evaluated in multiple discussions.

We are pleased to see that other research groups have cross-verified the perceptual accuracy of VMAF. Rassool (RealNetworks) reports high correlation between VMAF and DMOS scores for 4K content. Barman et al. (Kingston University) tested several quality assessment metrics on gaming content and concluded that VMAF was the best in predicting the subjective scores. Lee et al. (Yonsei University) applied quality metrics for multi-resolution adaptive streaming and showed that VMAF and EPSNR demonstrated the highest correlation with perceptual quality. VMAF and VQM were the best performing quality metrics in the study of Gutiérrez et al. (Université de Nantes) where MOS scores were generated for HD and UHD content. We have also read studies where it is claimed that VMAF does not perform as expected. We invite industry and researchers to evaluate the latest VMAF models and encourage them to share with us counterexamples and corner cases that can potentially improve the next VMAF version. We also give best practices of using VMAF at a later section to address some of the concerns.

VMAF can be used as an optimization criterion for better encoding decisions, and we have seen reports of other companies applying VMAF for this purpose.

VMAF at Netflix

Codec Comparisons

Traditionally, codec comparisons share the same methodology: PSNR values are calculated for a number of video sequences, each encoded at predefined resolutions and fixed quantization settings according to a set of test conditions. Subsequently, rate-quality curves are constructed, and average differences between those curves (BD-rate) are calculated. Such settings work well for small differences in codecs, or for evaluating tools within the same codec. For our use case—video streaming—the use of PSNR is ill-suited, since it correlates poorly with perceptual quality.

VMAF fills the gap, and can capture larger differences between codecs, as well as scaling artifacts, in a way that's better correlated with perceptual quality. It enables us to compare codecs in the regions which are truly relevant, i.e. on the convex hull of attainable rate-quality points. Comparing the convex hulls between different codecs and/or different configurations gives a comparison of the Pareto front of both options, in the rate-quality regions that practically matter. Some of our team's recent work on codec comparisons was published in a tech blog on shot-based encodes, and in academic papers at the Picture Coding Symposium 2018 and SPIE Applications of Digital Image Processing XLI.

Encoding Decisions

VMAF is used throughout our production pipeline, not only to measure the outcome of our encoding process, but also to guide our encodes towards the best possible quality. An important example of how VMAF is used within encoding is in our Dynamic Optimizer, where encoding decisions for each individual shot are guided by bitrate and quality measurements for each encoder option. VMAF scores are essential in this optimization process to get accurate quality measurements, and to select the final resolution/bitrate points on the convex hull.

A/B Experimentation

Researchers in different business areas—TV UI teams and streaming client teams, for example—are constantly innovating to improve streaming quality. With VMAF, we have a tool that allows us to run system-wide A/B tests and quantify the impact on members' video quality. For example, a researcher changes the adaptive streaming algorithm or deploys new encodes, runs an experiment, and compares VMAF between the old and new algorithms or encodes. This metric is well-suited for assessing quality in experiments because of its consistency across content and accuracy in reflecting human perception of quality. For example, a VMAF score of 85 will mean "good" quality for all titles, as opposed to using bitrate, where the same bitrate may indicate different quality between titles.

What We've Been Working On

Speed Optimization

When we first released VMAF on Github back in June 2016, it had its core feature extraction library written in C and the control code in Python, with the main goal of supporting algorithm experimentation and fast prototyping. Upon user's request, we soon added a stand-alone C++ executable, which can be deployed more easily in the production environment. In December 2016, we added AVX optimization to VMAF's convolution function, which is the most computationally heavy operation in VMAF. This led to around 3x speedup of VMAF's execution

time. Most recently in June 2018, we added frame-level multithreading, a long-due feature (special thanks goes to DonTequila). We also introduced the feature of frame skipping, allowing VMAF to be computed on every one of N frames. This is the first time that VMAF can be computed in real time, even in 4K, albeit with a slight accuracy loss.

libvmaf

With help from the FFmpeg community, we packaged VMAF into a C library called libvmaf. The library offers an interface to incorporate VMAF measurements into your C/C++ code. VMAF is now included as a filter in FFmpeg. The FFmpeg libvmaf filter is now a convenient paved path for running VMAF on compressed video bitstreams as input.

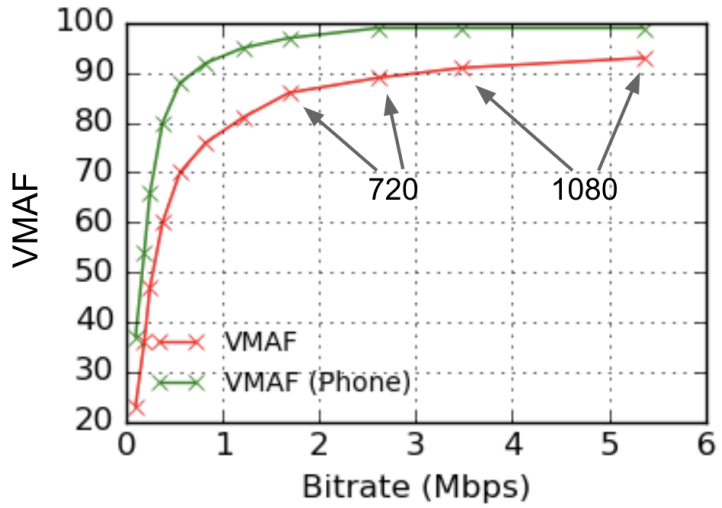
Accuracy Improvement

Since we open-sourced VMAF, we have been continuously improving its prediction accuracy. Over time, we have fixed a number of undesirable cases found in the elementary metrics and the machine learning model, yielding more accurate prediction overall. For example, the elementary metrics are modified to yield improved consistency with luminance masking; motion scores at the scene boundaries are updated to avoid overshoot due to scene changes; the QP-VMAF monotonicity is now maintained when extrapolating into high QP regions. Clearly, a VMAF model's accuracy also heavily depends on the coverage and accuracy of the subjective scores that it is trained on. We have collected a subjective dataset with a broadened scope compared to our previous dataset, including more diverse content and source artifacts such as film grain and camera noise, and more comprehensive coverage of encoding resolutions and compression parameters. We have also developed a new data cleaning technique to remove human bias and inconsistency from the raw data, and open-sourced it on Github. The new approach uses maximum likelihood estimation to jointly optimize its parameters based on the available information and eliminates the need for explicit subject rejection.

Viewing Condition Adaptation

The VMAF framework allows training of prediction models tailored to specific viewing conditions, no matter whether it is on a mobile phone or on a UHD TV. The original model released when we open-sourced VMAF was based on the assumption that the viewers sit in front of a 1080p display in a living room-like environment with the viewing distance of 3x the screen height (3H). This is a setup that is generally useful for many scenarios. In applying this model to the mobile phone viewing, however, we found that it did not accurately reflect how a viewer perceives quality on a phone. In particular, due to smaller screen size and longer viewing distance relative to the screen height (>3H), viewers perceive high-quality videos with smaller noticeable

differences. For example, on a mobile phone, there is less distinction between 720p and 1080p videos compared to other devices. With this in mind, we trained and released a VMAF phone model.



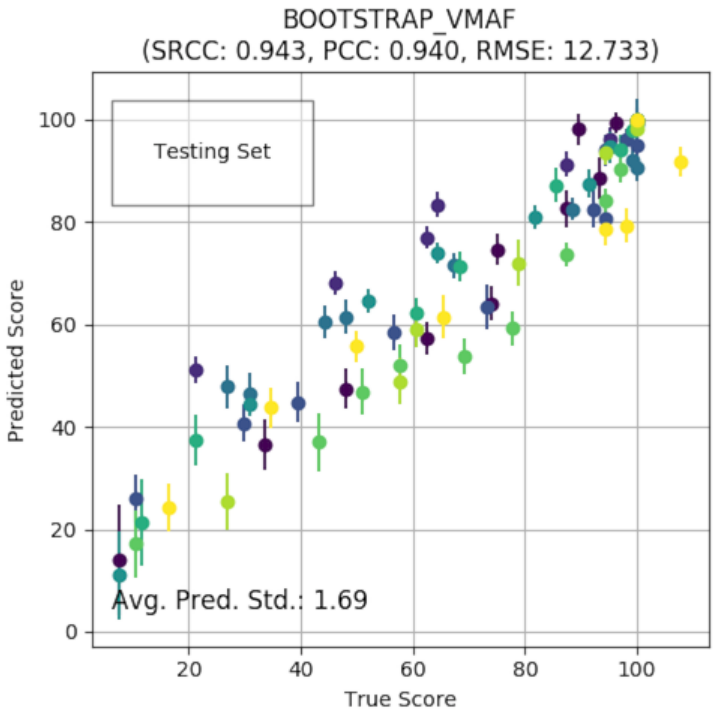
An example VMAF-bitrate relationship for the default model and the phone model is shown here. It can be interpreted that the same distorted video would be perceived as having a higher quality when viewed on a phone screen than on a HD TV, and that the VMAF score differences between the 720p and 1080p videos are smaller using the phone model.

Most recently, we added a new 4K VMAF model which predicts the subjective quality of video displayed on a 4K TV and viewed from a distance of 1.5H. A viewing distance of 1.5H is the maximum distance for the average viewer to appreciate the sharpness of 4K content. The 4K model is similar to the default model in the sense that both models capture quality at the critical angular frequency of 1/60 degree/pixel. However, the 4K model assumes a wider viewing angle, which affects the foveal vs peripheral vision that the subject uses.

Quantifying Prediction Uncertainty

VMAF is trained on a set of representative video genres and distortions. Due to limitations in the size of lab-based subjective experiments, the selection of video sequences does not cover the entire space of perceptual video qualities. Therefore, VMAF predictions need to be associated with a confidence interval (CI) that expresses the inherent uncertainty of the training process. Towards this end, we recently introduced a method to accompany VMAF prediction scores with a 95% CI, which quantifies the level of confidence that the prediction lies within the interval. The CI is established through bootstrapping on the prediction residuals using the full training data. Essentially, it trains

multiple models, using “resampling with replacement”, on the residuals of prediction. Each of the models will introduce a slightly different prediction. The variability of these predictions quantifies the level of confidence—the closer these predictions are, the more reliable the prediction using the full data will be.



The example plot here based on the NFLX Public Dataset showcases a 95% CI associated with each data point. It is interesting to note that points on the higher-score end tend to have a tighter CI than points on the lower-score end. This can be explained by the fact that in the dataset to train the VMAF model, there are more dense data points on the higher end than the lower. Notably, the bootstrapping technique will not necessarily improve the accuracy of the trained model, but will lend a statistical meaning to its predictions.

Best Practices

Oftentimes we have been asked whether a particular way of calculating VMAF score is appropriate, or how to interpret the scores obtained. This section highlights some of the best practices of using VMAF.

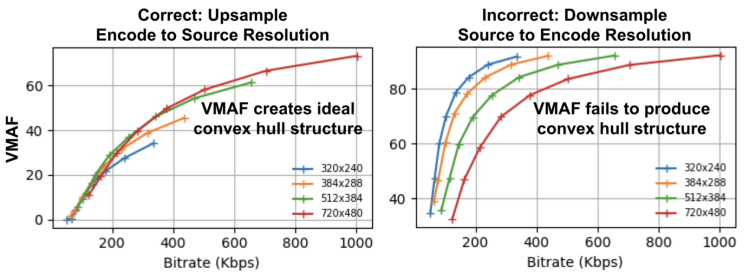
Interpreting VMAF Scores

VMAF scores range from 0 to 100, with 0 indicating the lowest quality, and 100 the highest. A good way to think about a VMAF score is to linearly map it to the human opinion scale under which condition the

training scores are obtained. As an example, the default model v0.6.1 is trained using scores collected by the Absolute Category Rating (ACR) methodology using a 1080p display with viewing distance of 3H. Viewers voted the video quality on the scale of "bad", "poor", "fair", "good" and "excellent", and roughly speaking, "bad" is mapped to the VMAF scale 20 and "excellent" to 100. Thus, a VMAF score of 70 can be interpreted as a vote between "good" and "fair" by an average viewer under the 1080p and 3H condition. Another factor to consider is that the best and the worst votes a viewer can give is calibrated by the highest- and lowest-quality videos in the entire set (during training, and before the actual test starts, subjects are typically accustomed to the experiment's scale). In the case of the default model v0.6.1, the best and the worst videos are the ones compressed at 1080p with a low quantization parameter (QP) and the ones at 240p with a high QP, respectively.

Computing VMAF at the Right Resolution

A typical encoding pipeline for adaptive streaming introduces two types of artifacts—compression artifacts (due to lossy compression) and scaling artifacts (for low bitrates, source video is downsampled before compression, and later upsampled on the display device). When using VMAF to evaluate perceptual quality, both types of artifacts must be taken into account. For example, when a source is 1080p but the encode is 480p, the correct way of calculating VMAF on the pair is to upsample the encode to 1080p to match the source's resolution. If, instead, the source is downsampled to 480p to match the encode, the obtained VMAF score will not capture the scaling artifacts. This is especially important when using VMAF as the quality criterion for per-title encoding, where the construction of the convex hull is crucial for selecting the optimal encoding parameters.



The above example illustrates the convex hull forming when VMAF is calculated correctly (left) and incorrectly (right). When VMAF is calculated with encode upsampled to match the source resolution, one can easily identify intersections among curves from different

resolutions. On the other hand, if the source is downsampled to match the encode resolution, the low-resolution encodes will yield undeservingly high scores, and no intersection pattern among the curves can be spotted.

Picking An Upsampling Algorithm

When upsampling an encode to match the source resolution, many upsampling algorithms are available, including bilinear, bicubic, lanczos, or even the more advanced neural net-based methods. It is untrue that the higher quality the upsampling algorithm is, the better. In principle, one should pick an algorithm that can best match the display device's. In many cases, the precise circuit for upsampling the video in a display is unknown. In this case, we recommend using bicubic upsampling as the approximation in general.

Interpreting VMAF Score When Resolution Is Not 1080p

A frequently asked question is: if both the source and the encoded video have a resolution lower than 1080p, can a 1080p model (e.g. the default model v0.6.1) still apply? Note that the default model measures quality at the critical angular frequency of 1/60 degree/pixel. One way to think about the default model is that it is a "1/60 degree/pixel" model, which means that it assumes that 60 pixels are packed into one degree of viewing angle. If applying the geometry, one can find that it equally applies to 1080p video with 3H, or 720p video with 4.5H, or 480p video with 6.75H. In other words, if applying the 1080p model to 720p / 480p videos, the resulting scores can be interpreted as the ones obtained with viewing distance of 4.5H / 6.75H, respectively. At such long viewing distances, a lot of artifacts are hidden from the human eye, yielding much higher scores.

Note that this interpretation is without the calibration of other factors such as the eye focal length, foveal vision, and viewer's expectation on SD vs. HD videos.

Temporal Pooling

VMAF produces one score per video frame. It is often useful to generate a summary score per longer time duration, for example, for a video segment of a few seconds, or even for a two-hour movie. Although sophisticated techniques exist to temporally pool the per-frame scores, our empirical results suggest that simple arithmetic mean (AM) is the best way of averaging, in that it yields highest correlation with subjective scores.

Another useful averaging technique is harmonic mean (HM). Oftentimes it produces a summary score very similar to AM, except that in the presence of outliers, HM emphasizes the impact of small values.

$$AM(85, 90, 87, 91) = 88.25$$
$$HM(85, 90, 87, 91) = 88.19$$
$$AM(90, 87, \mathbf{1}, 91) = 67.25$$
$$HM(90, 87, \mathbf{1}, 91) = 3.87$$

The examples above demonstrate the differences between AM / HM in the absence / presence of small-value outliers. Temporal pooling based on HM is useful when one wants to optimize quality based on VMAF while avoiding the occasional low-quality video encodes.

Metrics for A/B Experimentation

To understand the effects of different treatments in A/B tests, we need metrics to summarize per-frame VMAF scores into per session metrics (i.e. one number per session). To add to the challenge, since in adaptive streaming the best bitrates to stream are selected based on a variety of factors (such as throughput), each session will have a different "combination" of per-frame VMAF values based on how long each stream was played and during which time period of the session. To reach a single VMAF summary number per session, we must 1) determine an appropriate level of aggregation and 2) build upon these aggregates with metrics that reflect different aspects of perceptual quality.

If we aggregate the per-frame VMAF scores to one average number per stream, we will miss drops in quality that happen during the session. If we do not aggregate at all and use per-frame values, the computational complexity for creating final summary metrics based on per-frame values for every session will be too high. To this end, we recommend going with intervals in the granularity of seconds to strike a balance between analytic accuracy and ease of implementation.

In order to understand the effects of different treatments in A/B tests, we recommend metrics which capture three key aspects of quality: aggregate quality, startplay quality, and variability. These could be the average VMAF over the entire session, average VMAF over the first N seconds, and the number of times the VMAF value drops below a certain threshold relative to the previous values.

The Journey Continues

From an internal project to help deliver the best video quality to Netflix's members, to an open-source project that starts attracting a community of users and contributors, VMAF has been constantly evolving and continuously finding new areas of applications. We are pleased to see that inside and outside Netflix, VMAF has been applied to codec comparison, encoding optimization, A/B experimentation, video analysis, post-processing optimization, and many more areas.

Independent researchers have helped cross-verify the perceptual accuracy of VMAF. The open-source community has helped speed up the tool, create easy-to-use interfaces, and moderate the Github repo.

But we are just getting started.

Through conversations with researchers and VMAF adopters, we have realized that there are many areas that the current VMAF can improve upon. To give a few examples:

- VMAF uses a simple temporal feature, which is the average low-pass filtered differences between adjacent frames. Potentially, VMAF could benefit from more sophisticated models that can better measure the temporal perceptual effects.
- VMAF does not fully capture the benefits of perceptual optimization options found in many codecs, although it is moving in the right direction compared to PSNR.
- Currently, VMAF does not use chroma features, and does not fully express the perceptual advantage of HDR / WCG videos.
- The VMAF model works the best with videos of a few seconds. It does not capture long-term effects such as recency and primacy, as well as rebuffering events.

In the coming years, we strive to continue improving VMAF, and we invite industry, academia and the open-source community to collaborate with us.

Acknowledgments

We would like to acknowledge the following individuals for their help with the VMAF project: Prof. C.-C Jay Kuo (University of Southern California), Joe Yuchieh Lin, Eddy Chi-Hao Wu, Haiqiang Wang, Prof. Patrick Le Callet (Université de Nantes), Jing Li, Yoann Baveye, Lukas Krasula, Prof. Alan Bovik (University of Texas at Austin), Todd Goodall, Ioannis Katsavounidis, the members of the Video Algorithms team at Netflix, and the open-source contributors to the VMAF project.