An extensible framework for video ASIC development and validation at Facebook scale

Shahid, Zafar, Chaudhari, Gaurang, Reddy, Vimal, Yang, Jinghan

Zafar Shahid, Gaurang Chaudhari, Vimal Reddy, Jinghan Yang, "An extensible framework for video ASIC development and validation at Facebook scale," Proc. SPIE 11842, Applications of Digital Image Processing XLIV, 118421M (1 August 2021); doi: 10.1117/12.2594309

SPIE.

Event: SPIE Optical Engineering + Applications, 2021, San Diego, California, United States

An extensible framework for video ASIC development and validation at Facebook scale

Zafar Shahid, Gaurang Chaudhari, Vimal Reddy, Jinghan Yang
{zshahid, gaurangc, vreddy2, jinghanyang}@fb.com
Facebook Inc, 1 Hacker Way, Menlo Park, CA, USA 94025

ABSTRACT

Video consumption across social platforms has increased at a rapid pace. Video processing is a compute-heavy workload, and domain-specific accelerators (ASICs) allow more efficient scaling than general purpose CPUs. One of the challenges for video ASIC adoption is that videos ingested in datacenters are user-generated content and have a long-tail distribution of uncommon features. Software stack can handle the outliers gracefully, but these uncommon features may pose a challenge for the ASIC with undesirable effects for the unsupported/unhandled end cases. To avoid undesirable effects in the production, it is critical to proof our system against the long-tail conditions early in the product cycle of the ASIC development. Similarly, critical signals like BD-rate quality and outlier detection are needed from production traffic early in the product cycle. To address these needs, we propose an extensible framework that allows a continuous development strategy using production traffic, through progressive evaluation in various product phases of the video ASIC development cycle. A similar framework would benefit other ASIC accelerator programs in reducing time to deploy on large-scale platforms.

Keywords: Video ASIC, video accelerator, user-generated content (UGC), extensible framework, system design, production testing

1. INTRODUCTION

Video consumption across social platforms like Facebook, YouTube etc. has increased at a rapid pace with billions of views per day on these platforms [1] [2]. In past two decades, we have consistently seen new video standards e.g., AVC [3] [4], VP9 [5], AV1 [6]. Video transcoding is a compute-heavy workload and to scale the infrastructure to meet this growing demand, it is imperative to invest more resources into system design and accelerate the video transcoding workload. For ABR delivery, multiple codecs families are used, which provides opportunities for cross-codec optimizations [7] . Increase in video traffic has led to the adoption of ASIC-based accelerators in datacenters [8]. Video ASICs offer significant performance efficiency over host CPU for the targeted workloads and provide a scalable path forward to meeting this growing demand.

One of the challenges for video ASIC adoption at Facebook is that videos ingested are mostly user-generated content (UGC), which has a long-tail distribution of uncommon features — unusual aspect ratios, frame rates, diverse container formats, timestamp issues, corrupted streams, etc. Our matured software (SW) stack can handle these non-conformant end conditions gracefully. But these pose a challenge for the ASIC and unsupported/unhandled end cases may have undesirable effects:

- Produce unintended effects in the outputs, affecting user experience.
- Cause the ASIC to enter a bad state, resulting in capacity loss.
- Cause hard to detect silent errors.

Hence, it's critical to test against these long-tail conditions early in the product cycle of the ASIC. Further, critical signals like BD-rate [9] quality are needed from production traffic early in the product cycle. To address these needs, we developed an extensible framework for video ASICs that allows a continuous development strategy using production traffic, through progressive evaluation in various product phases: bare-metal c-models, transaction level models that support firmware and software stack, emulator, and early silicon.

Applications of Digital Image Processing XLIV, edited by Andrew G. Tescher, Touradj Ebrahimi, Proc. of SPIE
Vol. 11842, 118421M · © 2021 SPIE · CCC code: 0277-786X/21/$21 · doi: 10.1117/12.2594309

This paper is organized as follows. In Section 2, we provide an overview of the previous works and the challenges one may face while deploying ASIC at scale in production. In Section 3, we present the proposed framework, followed by few applications of the proposed framework in Section 4.

2. STATE OF THE ART

There is always a need to compare different encoders and evaluate which has better quality, compression efficiency, performance tradeoffs [10]. Are We Compressed Yet? or commonly known as AWCY [11] is a widely used workflow. It helps to run exhaustive sets of video encoder evaluations at scale and compare the results. If we want to tune for convex-hull, we may use per-shot dynamic optimizer [12]. This solution works well for evaluating software models and development of software transcoders with static video dataset. However, such static frameworks do not provide sufficient data points to evaluate ASICs in a large-scale requirement of the social platforms like Facebook, YouTube, etc.

2.1 Quality vs Compute Tradeoffs.

For video quality, golden standard is subjective quality by human eye, but it is not scalable. This problem is solved by deploying objective quality metrics in production e.g., SSIM [13] and VMAF [14]. Facebook has deployed an efficient quality measurement [15], which is based on SSIM. On the other hand, test video clips are not necessarily representative of the UGC content at scale. Moreover, ASIC development cycle is quite different, and we cannot directly use the production media pipeline. A production media pipeline typically focuses on flexible rule driven workflow, scalability, resilience, availability, and operability. It is not very friendly for experimentation or as a platform for regression and quality debug.

2.2 Reliability for long-tail content

Over the last few years, cross functional teams in Facebook collaborated with industry ASIC partners to enable ASIC solutions to support video transcode workload. We have deployed these hardware systems in our datacenters enabling production traffic. At the container/bitstream level, the codec and container options can have a very long tail. Moreover, UGC content may be quite varied and different than the ones used by common test clips.

Enabling video ASICs in production had various challenges that had to be met. First, full stack reliability had to be established to serve production traffic, including reliability at hardware, application, and service level software, and all the way to product deliverability.

Second, the massive volume of video streams on Facebook platform meant high diversity of video content. Video sources may have non-compliant or unsupported features e.g., interlacing, mid-stream resolution changes, unsupported resolutions, corrupted bitstreams at codec or container level, timestamp errors etc. Some examples of such issues are shown in Figure 1 (out of order frames for ASIC transcodes), Figure 2 (ASIC transcoded video is darker) and Figure 3 (ASIC transcoded video is shifted towards right and bottom).

Third, debugging production video streams may be challenging due to strict security/privacy policies on access to video sources, which increased time to debug and triage failures. Especially working with our HW partners with only the broad characteristics of failed video streams. This meant debug process had long lead-times and highly reactive. A typical firmware/SW user-space regression could take weeks considering provisioning, production shadowing, issue identifying and debugging, and finally package deployment.

Knowing the production failure scenarios and the challenges of ASIC debug, a solution to support long-term production debug and future generation ASIC development, would be not just necessary but critical. In general, we wanted to establish a framework which could:

- Front-load production test in early ASIC development phases.
- Have the mechanism to fetch production video streams and called by offline shadow framework to conduct continuous regression test.
- Have the capability to analyze results and flag issues to quickly identify the source (silicon, firmware, driver, or user-space).
- What features in video streams caused the issues.

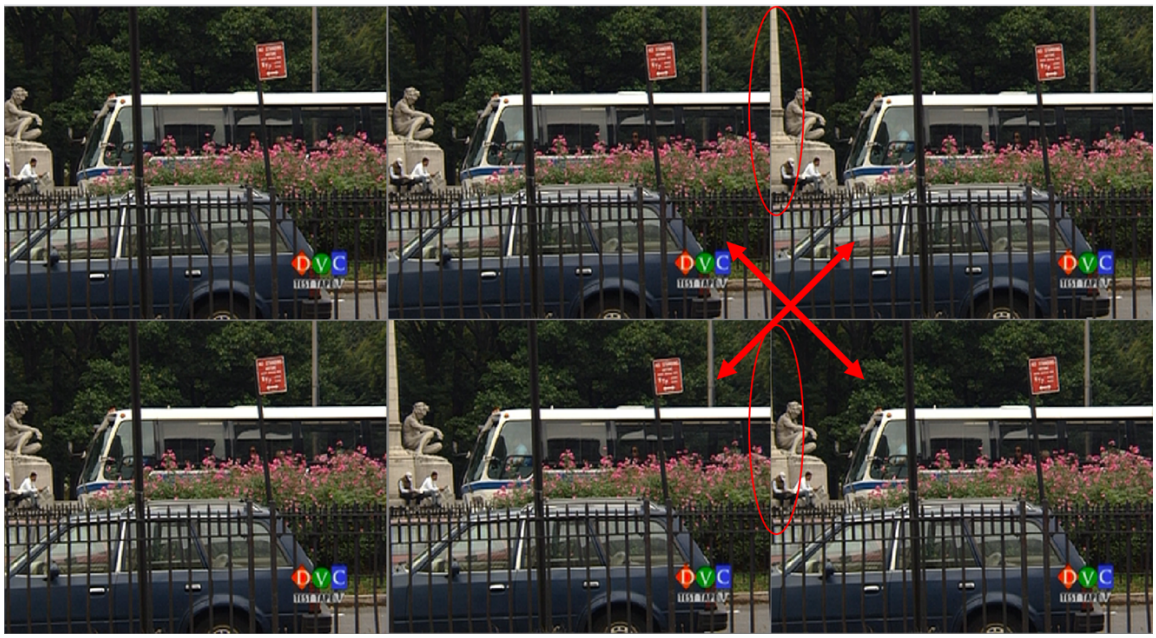

Figure 1 Top video sequence is transcoded correctly from SW pipeline, while bottom one is output of ASIC pipeline, wherein frames may be out of order.

Figure 2 Left video frame is output of SW pipeline, while right video frame is output of ASIC pipeline, wherein color may be shifted if color metadata is not handled properly by ASIC.

(a) input frame (b) ASIC transcoded frame (c) both frames super-imposed

Figure 3 Output frames may be shifted if input or output resolution are not standard ones, because of ASIC scaler constraint.

3. PROPOSED FRAMEWORK

The proposed framework for video ASIC development may be applicable for a wide variety of ASIC products, especially for those wherein input to the ASIC is not from a controlled environment and can be diverse. In this paper, we use video transcode ASIC as an example for illustration of this extensible framework.

Due to the long-tail conditions arising from the production scale at Facebook, it is difficult to create a test dataset for validating the variety of end cases on the ASIC before deployment. Moreover, there is a cost benefit to finding issues early in the ASIC development cycle. The proposed solution front-loads the ASIC-development with validation on production traffic on different ASIC IP components progressively swappable during the development cycle – c-models, RTL, emulation, early silicon. The proposed extensible framework for ASIC development consists of an online and offline framework.

3.1 Online Shadow Framework

In online shadow framework, actual production traffic is run on reference SW modules and ASIC modules. Metrics may be computed on both pipelines and compared to make sure that ASIC quality is within the acceptable limit to the reference software modules, as shown in Figure 4. Figure 5 shows that actual production traffic is run on reference SW modules and ASIC modules in parallel in both SW and ASIC pipelines.

We extract the following key signals from the online shadow framework:

1. Identify transcode failures and classify the failure type for further analysis.
2. ASIC state machine in case of failures: ASIC should exit cleanly in case of errors and should be in a stable state.
3. Compute BD-rate for ASIC encodes with respect to SW.

Outliers from this step are anonymized to protect privacy and further analyzed in the offline shadow framework.

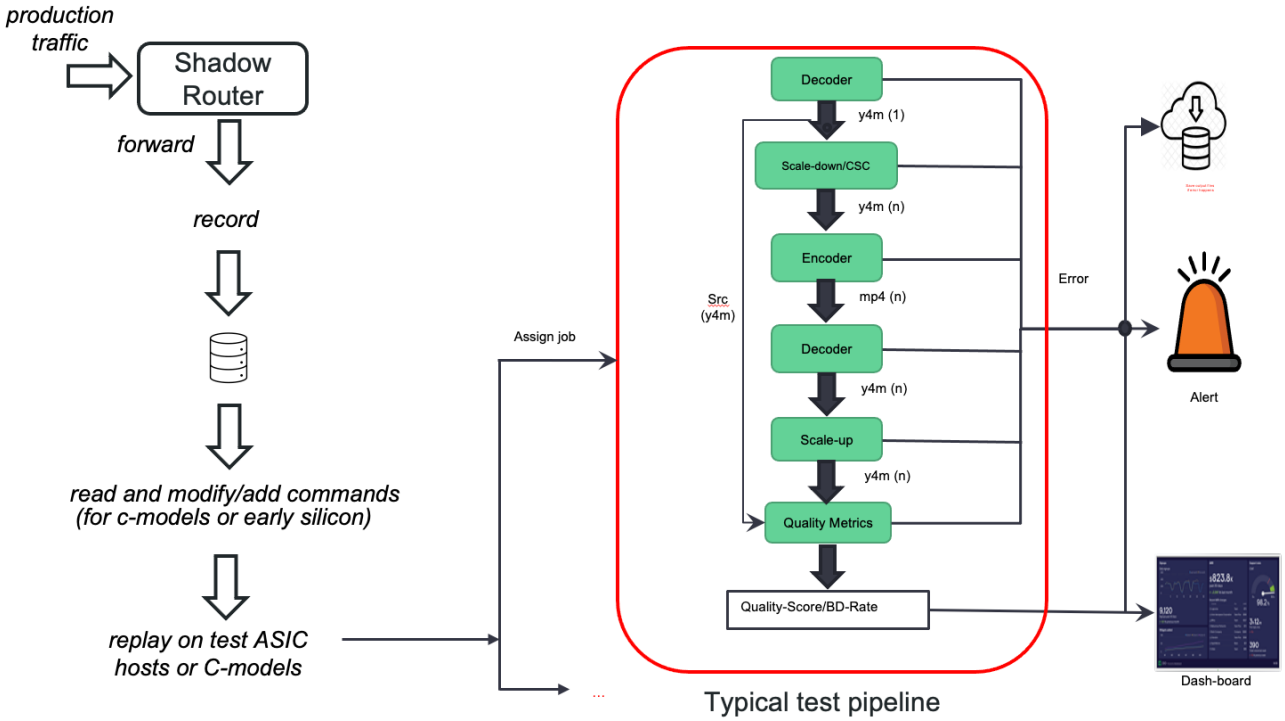

Figure 4 Proposed online shadow framework to replay production traffic on ASIC modules.

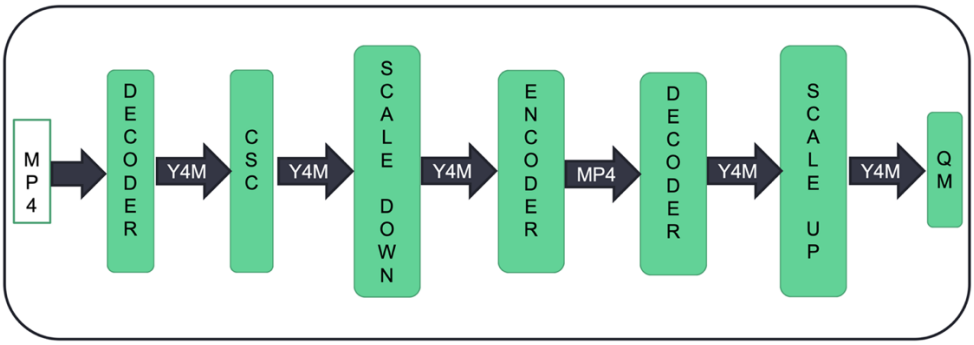

Figure 5 End-to-end transcode pipeline, where every component may be ASIC or SW.

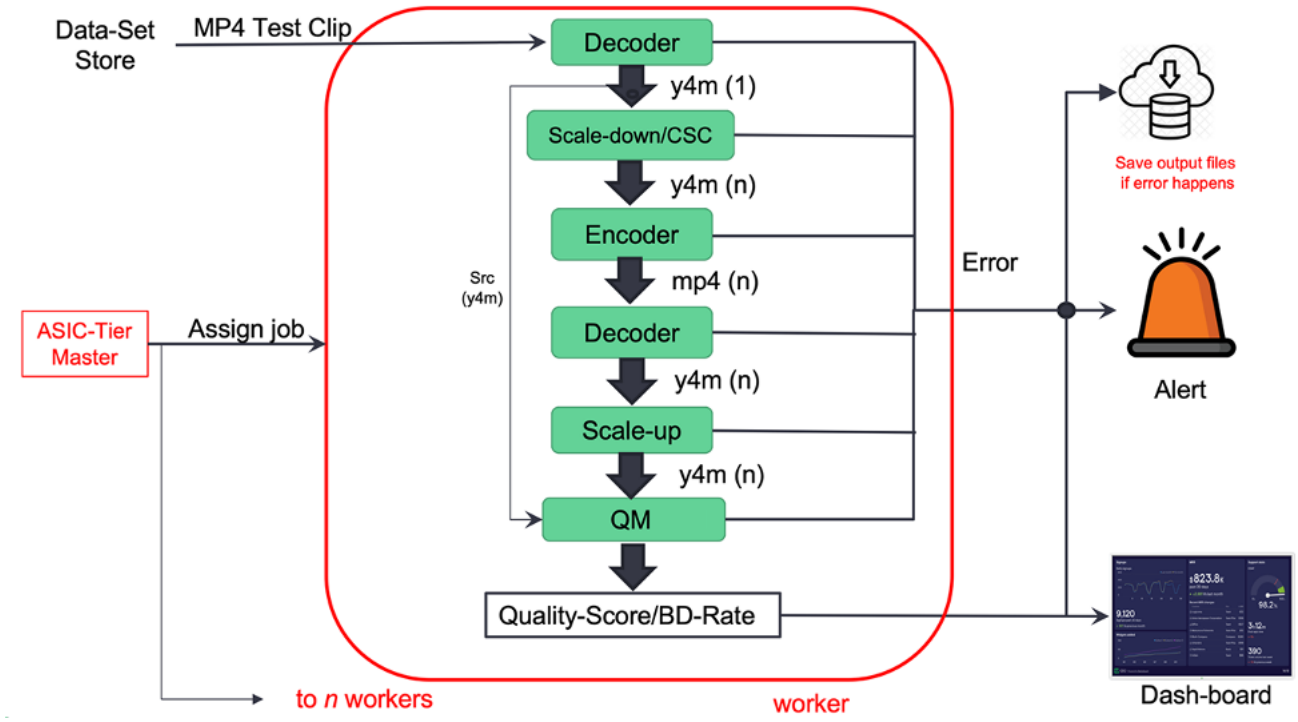

Figure 6 Offline shadow framework wherein both software and c-model based transcoding is performed on test dataset, instead of live production traffic.

3.2 Offline Shadow Framework

Once we have identified input clips which have either failed to transcode on ASIC, or their quality was below a threshold as compared to SW, these are further analyzed in the offline shadow framework as shown in Figure 6.

What we want to achieve is as following:

1. Root cause the failed bitstreams and localize the issue to concerned ASIC modules for further debugging. A hybrid model may be used for this purpose, wherein one model in the transcode pipe is swapped with the corresponding software model.
2. Transcode success/failure for as many clips as possible.

3. Root cause compliance issues for bitstream.
4. Flag quality issues for production videos on ASICs.

3.3 Logging

1. First, we need to dump the output to a file. The output may be the quality scores e.g., VMAF/SSIM/PSNR.
2. Output format of software and ASIC modules might be in different formats. We need to grep the output to extract the quality scores.
3. Log this data to database for comparison/tracking.

3.4 Dashboard and Alerts

Once logging is set up, dashboards and alerts are created to monitor the progress, quality, stability, and throughput of c-models/emulators in an automated way. It may help all partner teams in becoming more self-sufficient in triaging/investigating the issues. Similarly, alerts are set up so that we may not need to analyze dashboards manually.

4. APPLICATIONS OF THE FRAMEWORK

The proposed framework may be used for faster development and deployment of ASIC at large scale, where we do not have control over the input user generated content. It may help to improve ASIC development in various aspects including:

4.1.1 BD-rate quality analysis

As new innovations and standards are developed, there is always a need to compare different encoders and evaluate which has better compression efficiency, performance tradeoffs, etc. UGC content is challenging because it is neither pristine, nor it is from some limited set of devices. To evaluate ASIC encoders on UGC content, the proposed framework facilitates testing on a regular or even daily subset of incoming UGC content. Thus, employing the proposed framework can help us to find outliers, and tune rate-control algorithms for those outliers.

4.1.2 Reliability for diverse type of uploads

As explained in Section 2.2, uploads to Facebook are quite diverse in terms of device model (latest vs 20 years old), bandwidth (2G vs WIFI), codecs (AVC vs AV1), and equipment (studio vs common user). A static dataset may not cover this diverse type of content for ASIC development. The proposed framework with shadow traffic can play a vital role wherein we may transcode subset of new content every day.

4.1.3 Improving Encoding Recipes

Bandwidth of the end devices is dynamic, and hence ABR is used to deliver video content to these devices, wherein we have a family of resolutions/bitrates. To evaluate ASIC encoders over the whole convex hull for these families, it is pertinent to test encoders on shadow traffic for these ABR families.

4.1.4 Future research

To develop new codecs, we use static video datasets known as common test conditions (CTC) for evaluation of encoding tools [6]. In addition to the existing codecs, the proposed framework may help to develop new codecs, as researchers may test new tools on UGC content and may recommend refined configurations for such content.

5. CONCLUSION

In this paper, we presented an extensible framework for video ASIC development and validation, for production use-cases where a single fixed input dataset may not be a representative verification space for the ASIC getting deployed at a large scale. The proposed framework may help to reduce the overall ASIC development time by a big margin, by catching bugs early in either the bare-metal c-models, transaction level models that support firmware and software stack, emulator, or early silicon. Moreover, the same framework may help validate the compression efficiency (BD-rate quality evaluation) provided by ASIC transcoding, facilitate improving encoding recipes and future codec development on the production

traffic. While this framework is developed for video ASICs, a similar strategy would benefit other ASIC programs in reducing time to deploy on large-scale platforms.

REFERENCES

[1] D. Etherington, "People now watch 1 billion hours of YouTube per day," TechCrunch, 28 Feb. 2017. [Online]. Available: https://techcrunch.com/2017/02/28/people-now-watch-1-billion-hours-of-youtube-per-day/.

[2] "Facebook Video Statistics," 2021. [Online]. Available: https://99firms.com/blog/facebook-video-statistics/#gref.

[3] "Coding of Audiovisual Objects - Part 10: Advanced Video Coding," ISO/IEC 14496–10:2003.

[4] "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264.

[5] D. Mukherjee, J. Han, J. Bankoski, R. Bultje, A. Grange, J. Koleszar, P. Wilkins and Y. Xu, "A Technical Overview of VP9 – The Latest Open-Source Video Codec," in *SMPTE 2013 Annual Technical Conference & Exhibition*, 2013.

[6] Y. Chen, D. Murherjee, J. Han, A. Grange, Y. Xu, Z. Liu, S. Parker, C. Chen, H. Su, U. Joshi, C.-H. Chiang, Y. Wang, P. Wilkins, J. Bankoski and L. Trudeau, "An overview of core coding tools in the AV1 video codec," in *Picture Coding Symposium*, San Francisco, CA, USA, 2018.

[7] G. Chaudhari, H. Lalgudi and H. Reddy, "Cross-codec encoding optimizations for video transcoding," in *SPIE 11510, Applications of Digital Image Processing XLIII*, 2020.

[8] K. Lee, V. Rao and W. Arnold, "Accelerating Facebook's infrastructure with application-specific hardware," 14 March 2019. [Online]. Available: https://engineering.fb.com/2019/03/14/data-center-engineering/accelerating-infrastructure/.

[9] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves (VCEG-M33)," *VCEG Meeting (ITU-T SG16 Q. 6),* 2001.

[10] D. Grois, T. Nguyen and D. Marpe, "Performance Comparison of AV1, JEM, VP9, and HEVC Encoders," in *SPIE 10396, Applications of Digital Image Processing XL*, 2018.

[11] R. Zumer, "Scalable codec testing with Are We Compressed Yet?," Vimeo Engineering Blog, 2020. [Online]. Available: https://medium.com/vimeo-engineering-blog/scalable-codec-testing-with-are-we-compressed-yet-c3a64003f67b.

[12] I. Katsavounidis, "Dynamic optimizer - a perceptual video encoding optimization framework," 2018. [Online]. Available: https://netflixtechblog.com/dynamic-optimizer-a-perceptual-videoencoding-optimization-framework-e19f1e3a277f.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, no. 4, pp. 600-612, 2004.

[14] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy and M. Manohara, "Toward a practical perceptual video quality metric," [Online]. Available: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[15] S. L. Regunathan, H. Wang, Y. Zhang, Y. Liu, D. Wolstencroft, S. Reddy, C. Stejerean, S. Gandhi, M. Chen, P. Sethi, A. Puntambekar, M. Coward and I. Katsavounidis, "Efficient measurement of quality at scale in Facebook video ecosystem," in *SPIE 11510, Applications of Digital Image Processing XLIII*, 2020.