SCTE CABLE-TEC EXPO 22
SEPTEMBER 19-22 • PHILADELPHIA, PA

2022 Fall Technical Forum
SCTE • CABLELABS • NCTA

Artificial Intelligence in Real-Time Video Encoding

from Theoretical Promises to Operational Gains

A Technical Paper prepared for SCTE by

Jan De Cock
Director Codec Development
Synamedia
Luipaardstraat 12, 8500 Kortrijk, Belgium
+32 467 093721
jdecock@synamedia.com

Table of Contents

List of Figures

1. Introduction

Many books and articles have been written about artificial intelligence (AI) and machine learning (ML), in a variety of applications. ML is far from new, has an established theoretical foundation, and lots of different types of ML techniques have been introduced over the past decades. These techniques can be classified in different ways, but a full taxonomy is outside of the scope of this article. In this paper, we focus mostly on ML algorithms, as a subset of AI. Excellent introductions and overviews have been provided in e.g. [Goodfellow16, Bishop95].

Lots of successes have been claimed based on ML, and reports of AI intelligence are already the subject of ethical discussions [Google22]. Still, the powers of machine learning are not always a solution, and in many applications, even though they make for an interesting marketing statement, they do not lead to net gains or operational savings.

Machine learning has powerful applications in computer vision, image and video processing, and approaches using deep neural networks have become the center of academic and industry research. For example, residual neural networks have shown impressive results for image classification and recognition [Simonyan14, He16]. Still in most of these cases, very complex algorithms are needed, requiring e.g. deep neural networks containing dozens or hundreds of layers. While it's acceptable to have a very complex *training* stage (which needs to be executed once), it's primarily the complexity of the *inference* network (which needs to be repeated many times) that determines the feasibility of ML approaches[1]. An important unit of expressing the complexity of ML inference networks is the number of *multiply-accumulate operations* (MACs). Some of the best-performing image recognition networks use millions of MACs per image.

Often, new approaches are deemed feasible when they can be run on state-of-the-art GPUs inside a server. In certain cases, this is acceptable, and the cost of a dedicated CPU or GPU is warranted. For real-time, cost-sensitive applications, however, this is not an option. In typical video encoding/transcoding set-ups, dozens or even hundreds of channels need to be processed on a single server, and the cost per channel is a crucial criterion. Furthermore, the latency of offloading decisions to accelerators (if they would be cost effective, which is not the case), would be prohibitive.

In this paper, we discuss the applicability of machine learning approaches in different areas of *real-time* video compression. We successively cover encoder complexity reduction, rate control, video quality improvements and video quality measurement. In each of these areas, we have studied ways to reduce the complexity of ML inference, to end up with algorithms that are applicable in real-time, cost-sensitive applications.

[1] While in this paper we're mostly concerned with the complexity of the inference stage (which needs to be run every time a decision is made), the cost of *training* these networks can still become prohibitive in some cases. Not only the impact on computation, but also on emissions needs to be considered. In the field of natural language processing, Transformer Networks such as GPT-3 have been developed, with 175 billion ML parameters, requiring tons of CO2e just for training.

2. Applying ML to video compression: from ML to TinyML

In contrast to what our marketing departments would like us to believe, applying ML to any problem is not a trivial task, and it doesn't work out-of-the-box. For any problem solved using ML, *domain knowledge* is required. In the case of video encoding, a lot of specialized knowledge about video compression and its components is essential.

When attempting to combine the difficulties of domain knowledge, ML knowledge and complexity aspects, it is easy to get stuck in the "trough of disillusionment", where little or no net benefits of ML are reaped. In the end, product-grade encoders are usually mature, with smart human-designed algorithms. And indeed, in several areas, the gains are not immediately spectacular. Still, by pushing through, it is possible to obtain productivity and operational gains.

It is also tempting to assume that we can apply deep CNNs or similar computer-vision inspired techniques to video compression: detecting objects, humans, optical flow, or even apply a semantic meaning to each of these objects – after which we use that semantic information to help compression. While this is possible in theory, and a typical human reflex, such approaches are typically error-prone, inconsistent over time, and require a tremendous amount of computational resources.

In this paper, we focus on ML approaches that are feasible in real-time, with minimal impact on "channel density", i.e., the number of video channels that can be processed on a single server. As a result, these are realistic techniques that lead to operational savings and efficiency increases.

As an analogy, the challenges encountered in ML-based video compression are similar to those in the research field of "Tiny ML" [Warden20] – even though we're working on servers with multi-core CPUs. But instead of running single tasks on a very low-power platform (in the mW range), we process hundreds of video streams on a single CPU, making millions of decisions per second. To continue the analogy with TinyML, every individual building block inside an encoder has only milliwatts of power available. This results in an exciting new combination of research on *low-complexity real-time ML inference for video compression*.

3. Video encoding complexity (reduction)

3.1. A high-level view on encoding

In this paper, we continue to focus on video encoding as application. Compressing and encoding video is an extremely complex process, comprising different steps including pre-analysis, mode decision, motion estimation, interpolation, intra/inter prediction, transform, quantization, entropy coding, in-loop deblocking etc. Executing each block is a time-consuming operation, but it's mostly search space exploration (i.e. evaluating the different encoding options such as partitions and motion vectors) that is expensive in encoders. A simplified version of a hybrid block-based encoder is shown in Figure 1.

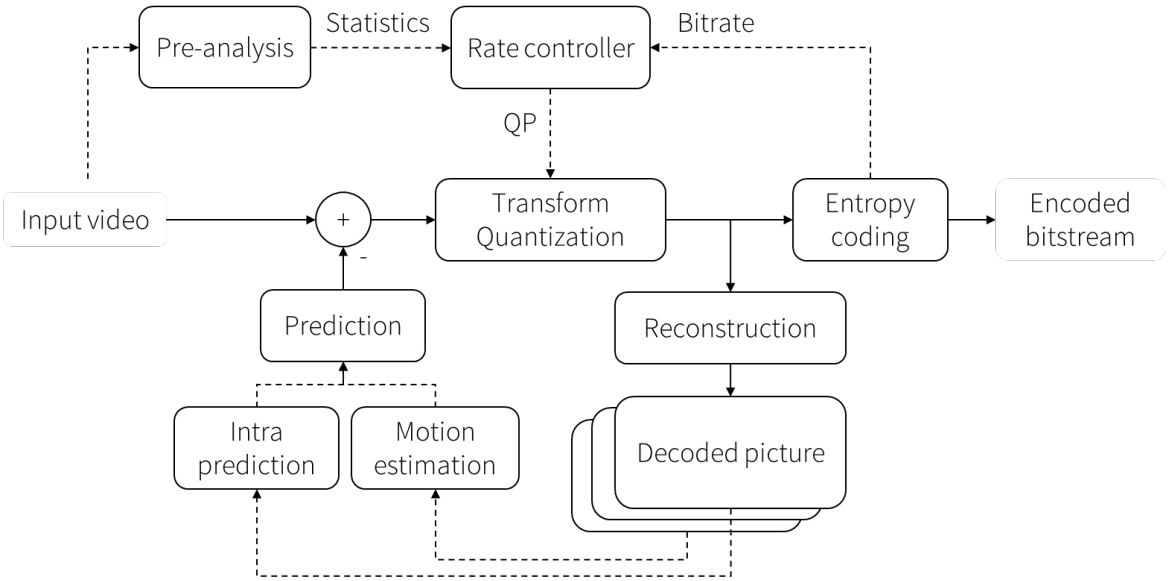

Figure 1. High-level overview of a hybrid block-based video encoder

Video encoding not only demands a lot of CPU power, but also memory (bandwidth). With higher resolutions (e.g. 8K) and higher frame rates (e.g. 120 fps), billions of pixels need to be processed every second. For lower-resolution streams, typically dozens to even hundreds of streams can be processed in parallel on a single server.

To cope with this complexity, lots of effort has been spent on developing hardware accelerators, FPGAs, ASICs and so forth. Still, encoding in software has many benefits, and brings maximum flexibility in deployment (on-prem, cloud-based etc), upgrades, and video quality improvements. For operational efficiency and flexibility, software encoding is often preferred, and will be the focus in this paper.

3.2. Encoder complexity reduction

To gain a competitive advantage, a low cost per channel is important. This is becoming increasingly challenging, as the complexity keeps increasing for newer compression formats. Each generation of video compression standards brings an increase in computational complexity (e.g. from MPEG-2 to AVC to HEVC), and newer standards are on the horizon (such as VVC). Typically, *decoder* complexity doubles with every generation, while jumps in *encoder* complexity can be even larger.

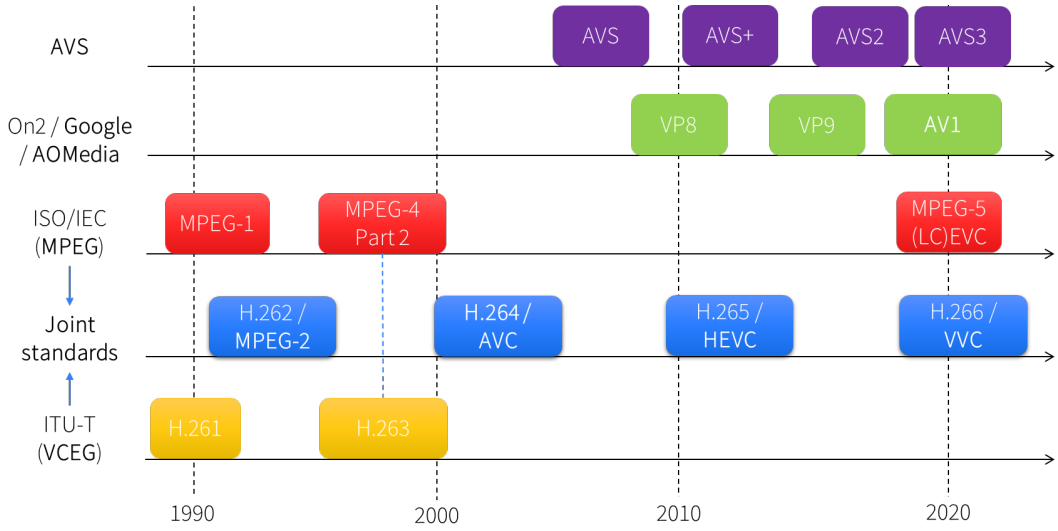

Figure 2. Evolution of video coding standards

Not only the resolutions are increasing, also the number of different ways to encode each individual frame or block is going up. While HEVC had about 80K different ways to partition a 64x64 block, the biggest coding units in VVC (128x128) can be partitioned in more ways than there are atoms in the universe[2].

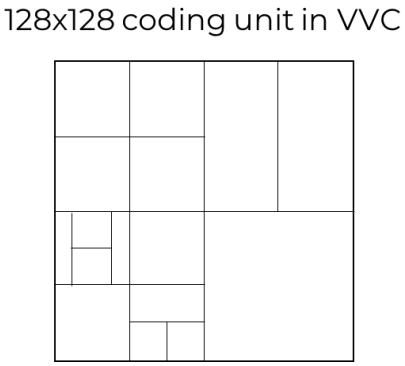

Figure 3. Example partitioning structure using VVC.

Given the potential for optimizations in this huge search space, finding efficient ways to make decisions inside the encoder is a popular research topic, and recent attention has shifted towards machine learning approaches. Plenty of references in this direction can be found in academic literature. Typically, however, the reference point in these papers is (extremely slow) reference software, and most of these gains cannot be transferred as such when applied to professional, real-time encoders.

Also, many publications on encoder complexity reduction focus on non-real-time (VOD-type) encoders. While this is useful as a starting point, those findings cannot be directly translated to *real-time* scenarios. Smart 'production-grade' encoders already make intelligent decisions, without exploring all options. In practical encoders, the promised complexity reductions are typically (way) lower.

[2] Every 128x128 coding unit can be split down to 4x4 coding units, with a recursive combination of binary, ternary or quaternary splits (or no split). I will leave the calculations up to the interested reader.

Still, interesting work in different directions has been performed resulting in ML networks with reasonable complexity, always while trying to limit the loss in compression efficiency:

- *Partitioning for HEVC and VVC*. [Liu16] presented a CNN-based CU partition size decision for HEVC with a reasonable complexity of 3,000 MACs, along with a hardware implementation. Among others, [Bhat21], [Wu21] and [Liu22] introduced CNN-based or SVM-based strategies for acceleration of VVC encoding, with a focus on the partitioning decisions, with encoding time reduction of roughly 30-80%, while limiting the impact on compression efficiency.
- *Intra prediction*. The work by [Santamaria20] presents NN-based intra prediction modes along with simplifications that lead to multiplications in the order of 100s up to 10,000s for 16x16 blocks. This builds on the work of [Pfaff18], and an interpretation analysis is run to come to simpler, explainable predictors that are easy to implement. The result is NN-based modes that are much closer to real-life usage.
- *Inter Prediction.* Although much of the recent work has focused on NNs, other ML techniques such as Decision Trees prove to be efficient ways to optimize encoder decisions, as in [Kim19], where inter prediction is accelerated for AV1.
- *Transform selection*. The transform search for AV1 is accelerated in [Su19], based on a neural network with one hidden layer. For transform kernel prediction, two shallow networks are used which are combined into a score for the 2D transform.

The message from these papers, along with our findings, are that fairly simple and shallow neural networks can produce accurate results, and at acceptable computational complexity.

3.3. Reducing the complexity of ML inference networks

Optimization techniques can help push the boundaries of what ML can achieve inside an encoder, and can help limit the cost of deeper networks. For example, *pruning* can be used to reduce the complexity of the networks, and to eliminate redundant MACs in the inference networks. Also, *quantization* allows to reduce the bit depth of operations, at the possible cost of some accuracy in calculations.

Specialized hardware *accelerators* can be tolerated in some applications, but might lead to a large overhead in latency, limiting their feasibility. Dedicated *instruction sets* (such as VNNI) provide a more convenient way to parallelize operations, and specialized matrix multiplication instructions are made available on the most recent CPU generations. Unfortunately, offloading ML inference to external accelerators or GPUs is usually not preferred, and would lead to unacceptable latency in the real-time applications we're discussing in this paper.

In all cases, a trade-off between accuracy and complexity needs to be found – again, domain knowledge is needed to find a good balance. That domain knowledge is also essential for the most important part, which is intelligent network design: shallow network and intelligent feature design.

4. Rate control

Rate control is another area where ML has proven to provide improvements. Video encoding can be considered as a resource allocation problem. Given a certain bit rate (=bit budget), the available bits need to be distributed in the best possible way across different frames and coding units, to reach the highest possible video quality. This is done by choosing the right quantizer (quantization parameter) for each block.

While this seems a fairly trivial task, it is actually an extremely difficult problem, given the multitude of options that every individual block can be encoded with. Furthermore, due to prediction, blocks are dependent on previously encoded blocks, further exploding the complexity of the problem. The power of its rate control algorithm is actually one of the biggest differentiators in the quality of an encoder.

In practical encoders, estimations are made to allocate quantizers to every block based on pre-analysis of the video content. Each block will be encoded with an estimated quantization parameter (QP), and the total bit rate for the frame needs to approximate the given bit budget as closely as possible. This pre-analysis and prediction stage is essential, and the prediction error needs to be as small as possible. The rate control process is depicted in Figure 4.

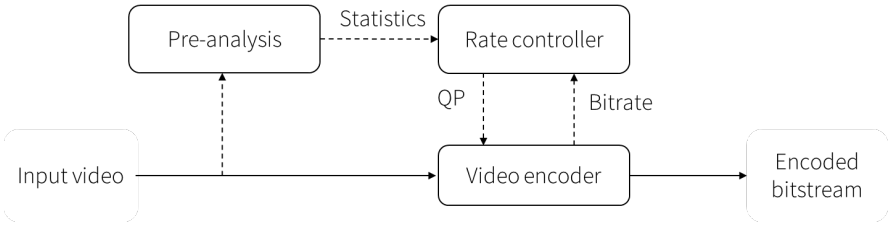

Figure 4. Simplified view on traditional rate-controlled encoding

Traditionally, prediction algorithms are based on human-designed heuristics, tweaking and testing. While this works well in general, there are cases where misprediction leads to fairly large bit estimation errors. This can lead to the rate controller over- or under-allocating bits for a number of frames. As a result, bits might be wasted, or quality might suddenly drop.

With machine learning, smart features can be calculated during the pre-analysis stage. With the resulting ML-based rate controller (Figure 5), we've noticed better resilience to a variety of content types, including sudden content or scene changes. As a result, we achieve better correlation between estimated and encoded bits, as illustrated in Figure 6.

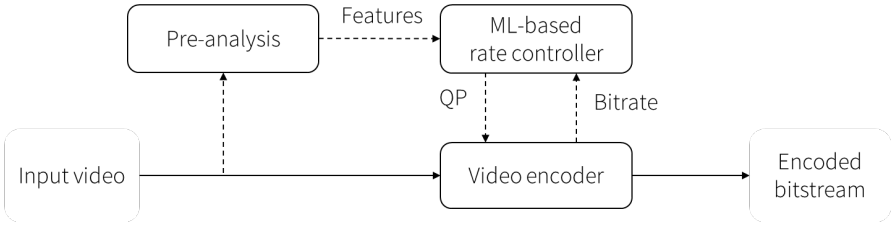

Figure 5. ML-based rate control

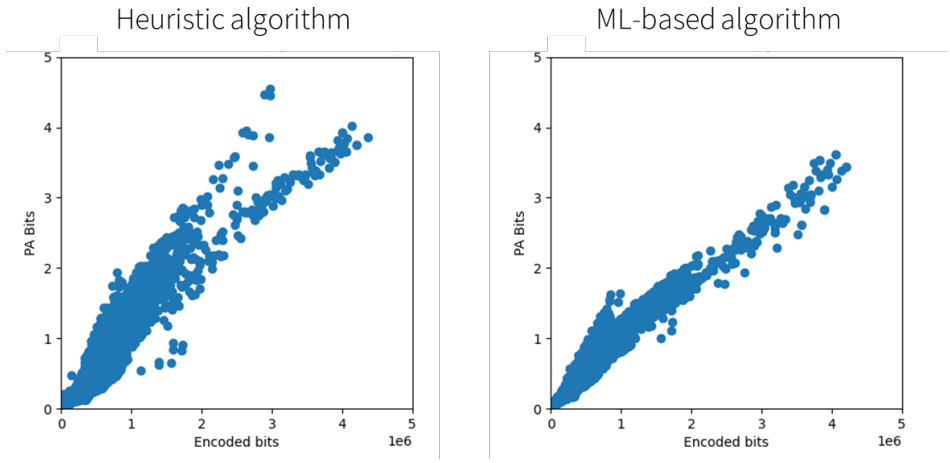

Figure 6. Improvement in rate control prediction accuracy between traditional (left) and ML-based (right) rate control.

5. Subjective improvements

Reaching a high level of video quality is extremely important when offering premium video content. Over the past decades, major steps forward have been made, going from analog TV to HD digital TV, and now it is common to watch premium sports in UHD, with the first 8K channels becoming available. In any case, we've come a long way since analog or DVD-level quality, and viewers are getting used to high-quality and ultra-high-definition video, by watching popular VOD services.

Pushing the limits in video quality is important to offer a premium to viewers, and as a differentiator when selling video encoding services. When comparing encoders (in so-called shoot-outs), offering the best quality is one of the most important criteria. In the next section, we will discuss VQ measurement based on *objective* metrics (as calculated by algorithms). But in this section, the focus is on *subjective* quality, as perceived by the viewer.

Several elements are important to optimize the visual quality as perceived by viewer. In early videos circulating on the Internet, digital video was suffering heavily from blocking artifacts, blurring, mosquito noise, ringing etc. These are quality artifacts that should be avoided at all costs, and that are no longer acceptable (and fortunately, less common) in modern video distribution.

In recent years, per-title, content-aware (CA), or even shot-based encoding have become mainstream. Introduced by Netflix [Aaron15], CA encoding finds the best bitrate (or settings) for every segment or shot. Content-adaptive encoding can help boost the quality of every segment of a video sequence. In a "black-box" version, a wide range of settings can be determined and fed into an ML framework, e.g. as demonstrated by Facebook in [Coward16].

While these techniques were introduced for VOD-type encoders, and operating at a very high complexity (e.g. by evaluating multiple options before deciding on the final encoder settings), they can also be applied to real-time encoders. In this case, decisions need to be taken much faster, with low latency and limited lookahead, and before the entire shot is available.

Smart bit distribution inside encoders can make a substantial difference in video encoders, and is necessary to preserve detail in the right places (e.g. players on a football field, or the football itself), to reduce artifacts and preserve textures. ML is well-suited to make decisions in real-time, and with a high degree of content adaptivity. Coding tools such as adaptive quantization, sample adaptive offset (SAO) filtering in HEVC or VVC, ALF in VVC, are excellent candidates for smart decision making based on ML. Some examples of improvements that were obtained by moving from handcrafted algorithms to ML-based subjective decisions are shown in Figure 7.

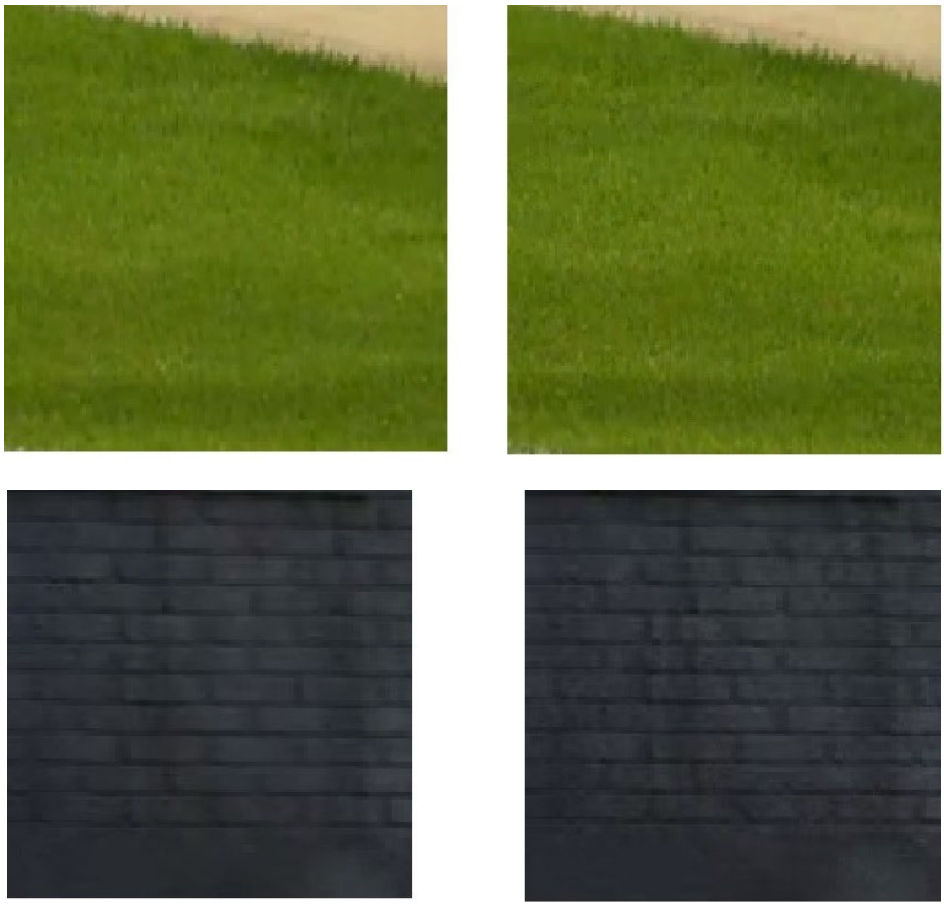

Figure 7. Example quality improvement between traditional (left) and ML-based (right) texture preservation.

For detection of areas of importance, ML algorithms can be used for higher accuracy. For example logos, faces, football players etc can be better detected and protected in sports games.

Figure 8. ML-based logo detection

Also filtering and post-processing are excellent candidates for smart ML-based techniques. Several articles have been published in this direction, such as in [Yang17] and [Kuanar18]. Also deinterlacing can benefit from CNNs, as described in [Bernasconi20], by combining residual and dense neural networks.

6. Video quality measurement

To verify the effectiveness of VQ optimizations, a great deal of time needs to be spent on subjective quality assessment. This is a process in which viewers (experts and/or non-experts) provide feedback on the video quality. Subjective test methodologies have been designed and standardized to handle this process. And every codec development team will have a set of 'golden eyes' in house to guide this process.

In practice, it's not feasible to verify the subjective quality of each and every video stream or channel, and subjective quality assessment is typically used only in specific occasions, e.g. during encoder comparisons, during the set-up or configuration of an encoder – or whenever an issue occurs. To reduce the high cost of human intervention, *objective* quality measurements have been introduced, to assist VQ measurement in an automated way, and as accurately as possible.

Objective VQ measurement is useful in applications such as encoder comparison and configuration, bitrate selection, ABR ladder (resolution, bitrate) optimization, real-time VQ measurement and monitoring, and in-loop quality control.

Different types of objective VQ measurement exist. In cases where the source video is available, a *Full-Reference* (FR) VQ comparison is possible (Figure 9). Examples of such metrics are MSE (mean squared error) or PSNR (peak signal-to-noise ratio), which calculate the difference between original and distorted pixels.

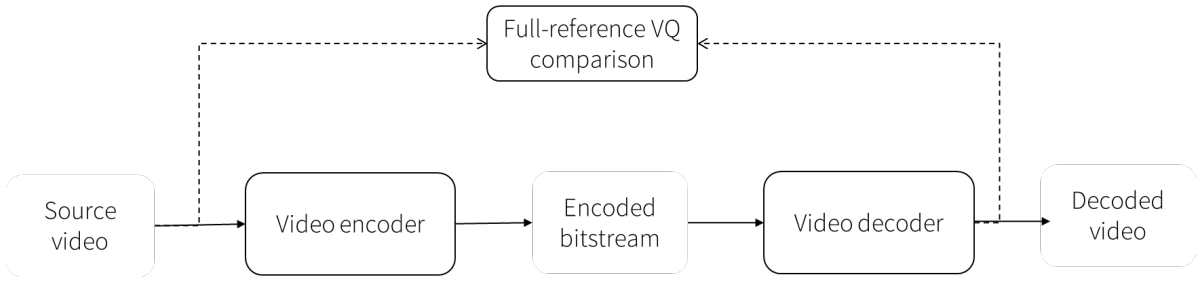

Figure 9. Full-reference video quality assessment

In other cases, a more difficult task is to evaluate the VQ without reference to the original (NR, no-reference). Here, indicators of e.g. the *naturalness* of images need to be calculated to get an impression of the overall quality. While NR metrics are extremely useful, the absence of a reference point makes NR scores less accurate. Inside the encoder, the encoded video can be compared to its input, and FR metrics can be used. We revisit NR metrics later on in the context of quality *monitoring*.

A multitude of FR metrics have been introduced over the past decades, including PSNR (peak signal-to-noise ratio), SSIM (structural similarity), MS-SSIM (multiscale SSIM), VMAF (video multimethod

assessment fusion) and others. The more complex metrics (such as MS-SSIM and VMAF) have shown to provide higher accuracy, while simple metrics like PSNR are not reliable enough for most purposes.

Machine learning has started to play a big role in VQ assessment and the creation of new metrics. An excellent example of an ML-trained metric is VMAF, which takes into account compression and scaling artifacts, and reaches a high accuracy on a variety of test sets. It is being used by many companies and is on its way to become a de facto industry standard. VMAF was trained on subjective data collected from human viewers, and uses a combination of underlying metrics as features. These features are weighted using SVM-based regression, resulting in a score between 0-100 to output the overall quality of the frames.

6.1. From offline to real-time VQ measurement: tracking video quality

A first obvious step to reach operational efficiency is to select the most efficient encoder and encoding configuration. Comparing encoders can be tedious, in particular when many different bitrates and parameters can be configured. Objective metrics can help identify the strengths and weaknesses of encoders, and can point to difference in encoder behavior over time or for different types of video content. Still, the question remains how the encoder will continue to perform, when an upgrade is applied, or when changes in configurations are made. Repeating extensive testing every time is an expensive and time-consuming task. Real-time VQ measurement is the preferred approach to track video quality over time.

As discussed above, ML metrics such as VMAF have been developed that work well for off-line measurement. For *real-time* measurement, however, we need metrics that are both accurate and affordable (meaning fast enough). VMAF's main downside is its computational complexity. Although efforts are ongoing to reduce the complexity of VMAF, its computational requirements remain high, especially when looking at the operational cost.

For operationally feasible VQ measurement, the cost of calculating VQ metrics needs to be reduced by several orders of magnitude compared to VMAF, and should be a fraction of the cost of the encoding itself. While simpler metrics like PSNR are often embedded inside encoders (as in Figure 10), they are not reliable enough for accurate VQ tracking.

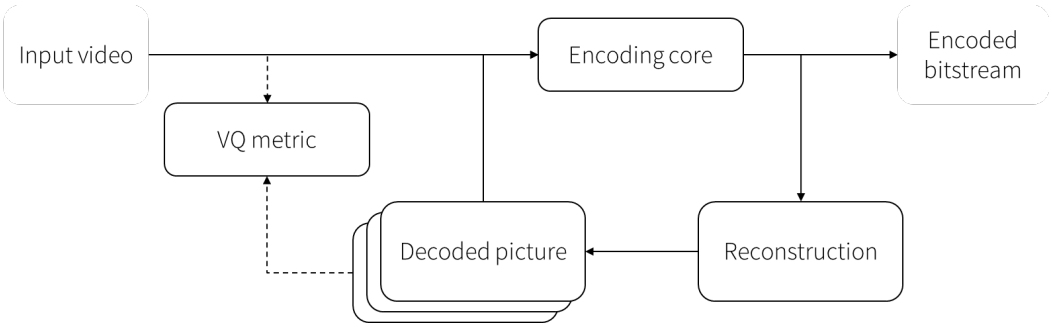

Figure 10. Calculating VQ inside the encoder

Machine learning provides ways to calculate VQ metrics in a smarter way. Deep video quality metrics such as DeepVQA have been proposed [Kim18], or using dynamic receptive fields and CNNs [Kim20], which provide state-of-the-art correlation with subjective scores. Still, they require multiple convolutional layers to reach the final score, and as a result, a high computational cost. As an alternative, smart features can be

calculated inside the encoder (or even reused from the pre-analysis stage inside the encoder), as a more powerful input to ML networks. This process is illustrated in Figure 11.

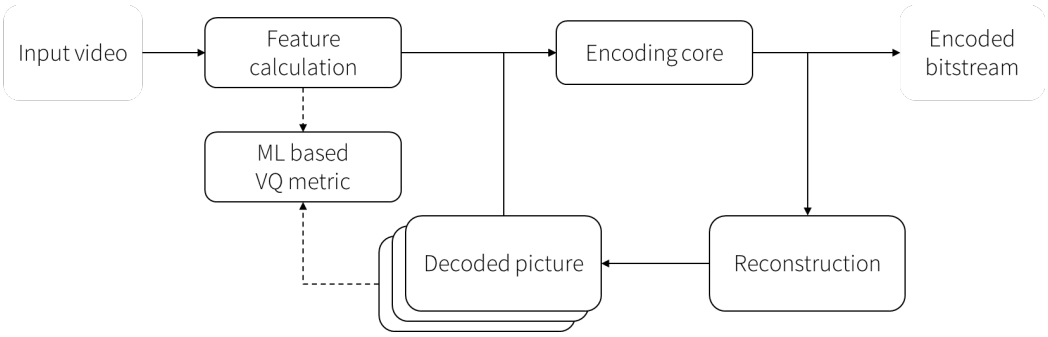

Figure 11. ML-based VQ measurement inside the encoder, based on pre-analysis features.

Finding a balance between input features, network structure, network complexity and accuracy is a complicated process, which as mentioned before, requires a lot of domain knowledge. We have identified ML networks that have more than 90% correlation with subjective scores, and that provide a fast and reliable alternative to expensive VMAF calculation. Example ML networks are shown in Figure 12, where we started from the network on the left, but were able to reduce the number of MACs by 70% to reach the same accuracy with network on the right.

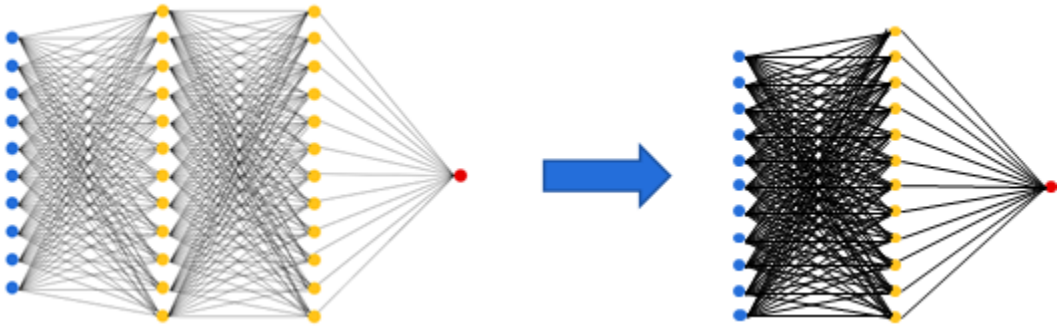

Figure 12. Example networks used for ML-based VQ prediction.

Once a VQ metric with acceptable accuracy and complexity is available in the encoder, it becomes possible to track video quality, and to verify 24/7 that a certain quality level is reached during encoding.

6.2. From rate control to quality control

Measuring video quality is one part of reaching efficient compression. An even bigger challenge is to actively *control* the VQ in real-time. In video compression, it is essential to reach a high video quality. But at the same time, for operational efficiency, you want to avoid overspending bits where they don't matter. At a certain point, bits can be added, but visual quality will no longer improve. This leads to waste, higher-than-necessary-bitrate, and delivery networks that are overloaded with redundant bits. Accurate video

quality metrics can help determine the saturation point and improve the intelligence of encoders. Figure 13 shows an example *rate-distortion* plot where the saturation effect is visible.

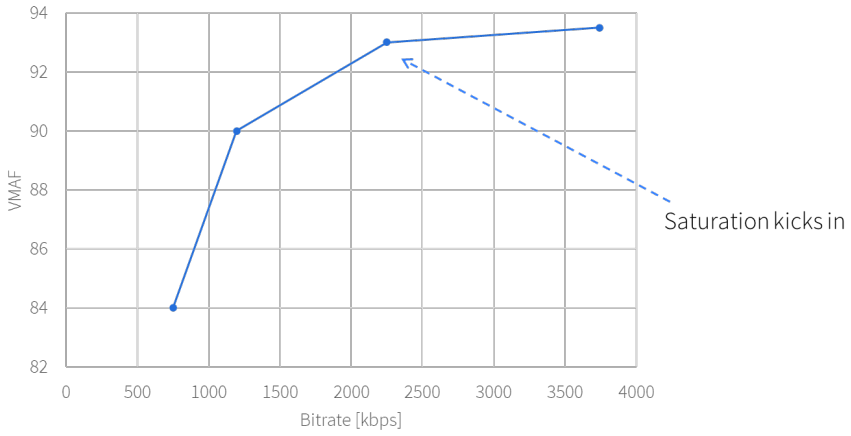

Figure 13. Above a certain bitrate, adding more bits will no longer (or hardly) improve quality

In the end, more important than constant bit rate, is to reach *consistent* quality. Note that *constant* quality is not feasible in practice. In all practical encoder systems, there is a maximum bitrate (cap rate) constraint, which will limit the allocation of bits. In case of very difficult scenes, quality might be limited because of that cap. Still, on average, in-loop quality steering will lead to less overspending (wasted bits) and less underspending (quality drops).

On top of VQ measurement, quality control poses an even more challenging problem, as VQ needs to be predicted before encoding. As a result, you end up with a chicken-and-egg problem. Fortunately, ML turns out to enable accurate prediction of encoded VQ, even before encoding. In this way, rate control can be turned into *quality* control. The result is a VBR stream which not only reaches a consistent quality, but also saves bits compared to traditional CBR rate control. Figure 14 shows how the different components fit together: ML-based rate control and VQ measurement, working alongside the ML-optimized video encoder.

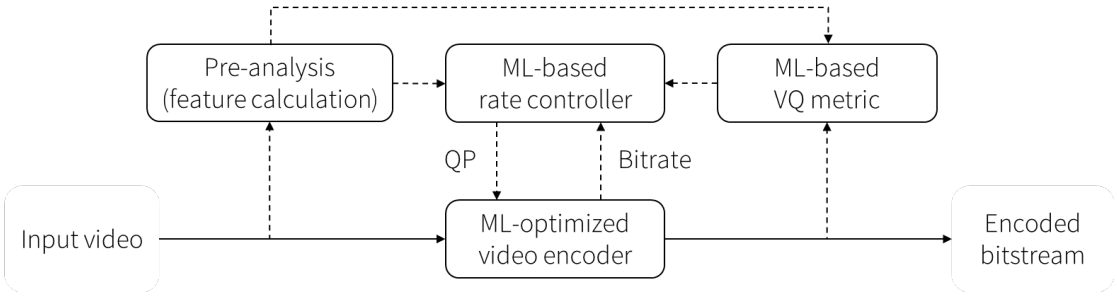

Figure 14. Quality-controlled compression

Figure 15 shows how quality-controlled compression can accurately reach a targeted quality value (in this case VMAF=90). Some variation is possible, and acceptable, when sudden changes in content occur, or when bitrate caps are reached.

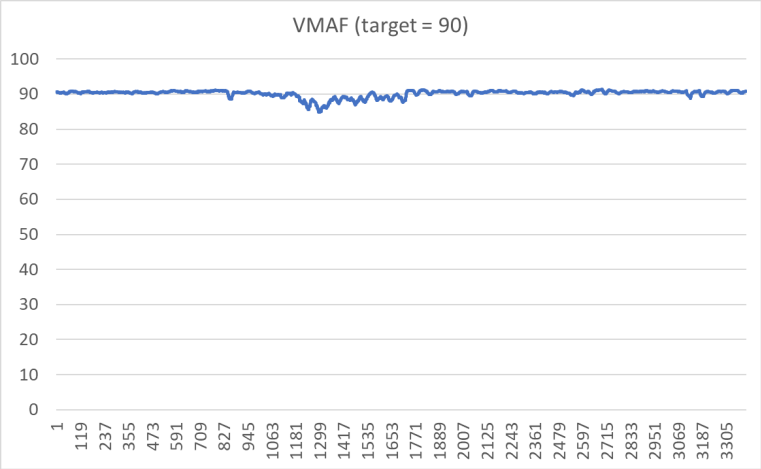

Figure 15. Result of quality-controlled compression (target VMAF=90).

Quality-controlled compression can be extended across multiple streams, e.g. in *statistical multiplexing* scenarios. Even though statmuxed streams are inherently VBR in nature, traditionally they have not been optimized with a particular VQ in mind. By using quality control, target quality levels can be specified for each individual stream in a statmux bundle, while still satisfying the total constant bit rate.

Also, *adaptive bitrate (ABR)* encoding can benefit – not only while optimizing individual streams, but also across the whole ladder. Instead of statically defining an ABR ladder as fixed resolution/bitrate pairs, it becomes possible to define quality targets for bitrate ladders, avoiding the need to specify fixed bitrates or even resolutions.

7. Video quality monitoring

One step beyond video quality tracking is VQ *monitoring*. Automated monitoring reduces the need for human inspection to keep track of video quality, with the objective of detecting issues with input sources, transcoding, delivery, or unexpected quality loss. This is another area where machine learning can provide clear benefits.

While VQ measurement as described in the previous section can give a good view of the quality loss introduced by encoding or transcoding itself (by comparing output and input), it does not provide a view of the *absolute* quality of the signal, or whether a potential problem might have occurred. If a disrupted signal enters the transcoder, the FR metric might still report high output quality. Basically, garbage in leads to garbage out…

To allow more flexibility in monitoring, *no-reference (NR) metrics* are essential since they give an impression of the overall video quality. In recent years, NR VQ assessment has been a hot research topic, with successful introduction of new metrics, often using deep neural networks, such as:

- [You19] proposed 3D-CNN and LSTM networks to extract local spatiotemporal features from small cubic video clips.
- [Bosse17] presented a deep neural network approaches with 10 convolutional layers and 5 pooling layers, and 2 fully connected layers for regression, totaling 5.2 million trainable parameters.
- [Bianco18] discusses different design choices for CNN-based blind image quality assessment, where the best design reaches a correlation of 0.91 with subjective scores.

For a good overall impression of quality, a combination of metrics (or *indicators*) is recommended. In the Video Quality Experts Group, studies have been made of different NR metrics. One of the higher-accuracy metrics is Sawatch (v3) [vqeg22], which combines multiple underlying quality indicators to detect e.g. blurriness, blockiness, saturation levels etc.

Still, even the more complex NR metrics reach relatively low accuracies when compared to FR metrics. Also, when quality issues occur, they are more prone to false positives or false negatives when compared to full-reference comparison. For more reliable VQ monitoring, measurements can be taken in different places during delivery, and their results correlated in a central point (e.g. in a cloud service). This provides a reliable comparison point for VQ degradation, transcoding artifacts, or simply transmission errors. Whenever a problem occurs, captures taken from the devices can be uploaded for deeper inspection. Only in those cases, human intervention is needed.

By combining FR and NR metrics, different monitoring scenarios become possible, such as:

- Observing quality fluctuations at a *single point* in the delivery chain. For example, quality measurements can be taken inside an encoder or transcoder and monitored over time. For this use case, either FR metrics (comparing encoded output to its input, Figure 16(a)) or NR metrics (without comparison to a source, Figure 16(b)) can be used.
- Comparing measurements taken in *multiple points* in the video delivery chain (Figure 16(c)). For example in video transport cases, or in multiple processing steps, metrics can be calculated to track the evolution of quality in the delivery chain.

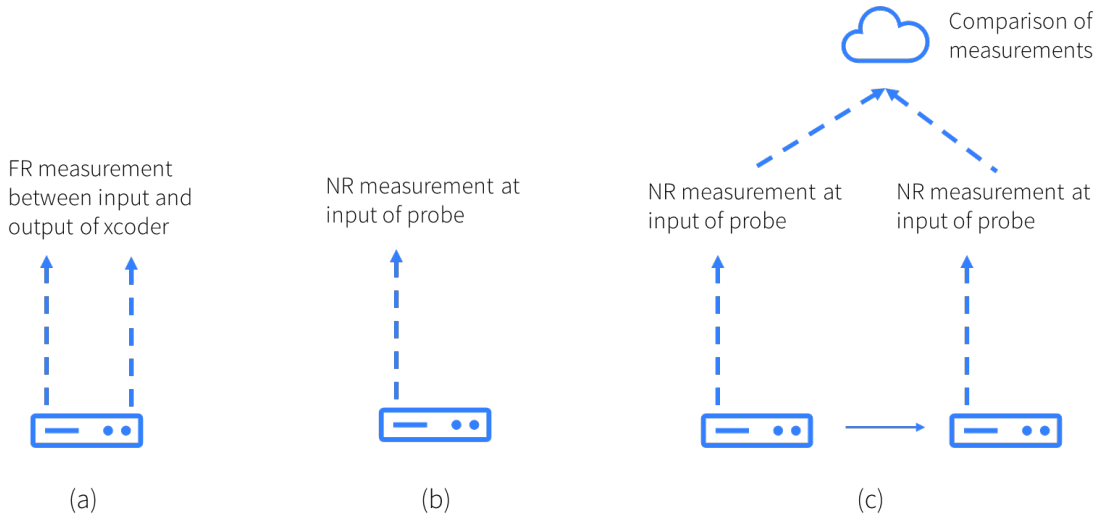

Figure 16. FR and NR quality measurement for VQ monitoring.

By combining these different measurement capabilities, monitoring across the whole delivery chain becomes possible, from contribution encoding to end delivery, as illustrated in Figure 17.

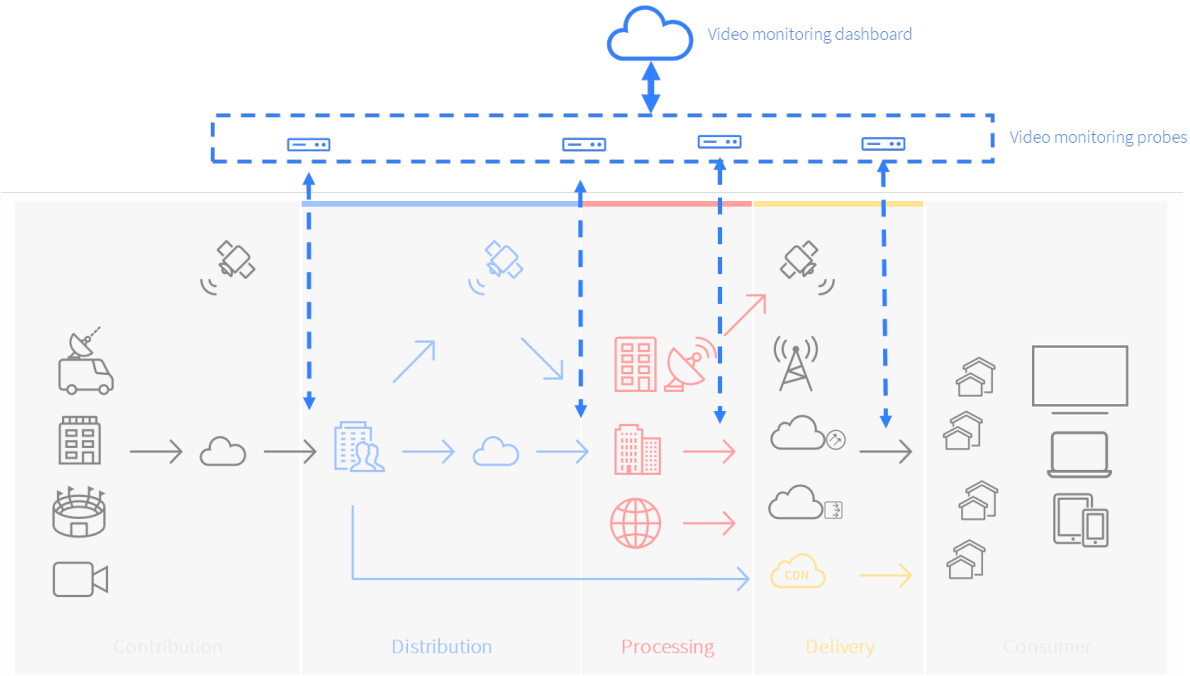

Figure 17. VQ monitoring at different points in the video delivery chain.

8. Conclusions

In this paper, we discussed several areas in video coding where machine learning has shown to provide benefits, including computational complexity reduction, rate control, and subjective video quality improvement. In contrast to popular deep neural networks for image recognition and classification tasks, less complex networks are needed in real-time scenarios, resembling the field of "TinyML". Relatively shallow ML networks are both computationally acceptable and have been shown to lead to improvements.

The field of VQ measurement benefits from ML approaches, both for non-real-time (such as VMAF) and real-time metrics that can be embedded inside encoders. For VQ monitoring, the combination of FR and NR metrics can be used to analyze quality across the end-to-end delivery chain. By bringing together the ML optimizations for rate control, subjective improvements and VQ measurement, quality-controlled compression can be achieved.

Abbreviations

ABR	adaptive bit rate
AI	artificial intelligence
AVC	Advanced Video Coding
CBR	constant bit rate
CNN	convolutional neural network
DNN	deep neural network
DPB	decoded picture buffer
FR	full-reference
HD	high definition
HEVC	High Efficiency Video Coding
LSTM	long short-term memory
MAC	multiply-accumulate
ML	machine learning
MPEG	Moving Picture Experts Group
MSE	mean squared error
NN	neural network
NR	no-reference
PSNR	peak signal-to-noise ratio
QP	quantization parameter
SD	standard definition
SSIM	structural similarity
SVM	support vector machine
VBR	variable bit rate
VQ	video quality
VMAF	video multimethod assessment fusion
VNNI	vector neural network instructions
VOD	video on demand
VVC	Versatile Video Coding

Bibliography & References

[Aaron15] Anne Aaron, Zhi Li, Megha Manohara, Jan De Cock and David Ronca, “Per-Title Encode Optimization”, https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2.

[Andreopoulos22] Y. Andreopoulos, “Neural pre and post-processing for video encoding with AVC, VP9 and AV1”, AOMedia Research Symposium 2022

[Barman19] N. Barman, E. Jammeh, S. A. Ghorashi and M. G. Martini, “No-Reference Video Quality Estimation Based on Machine Learning for Passive Gaming Video Streaming Applications”, in IEEE Access, vol. 7, pp. 74511-74527, 2019.

[Bernasconi20] A. Bernasconi, A. Djelouah, S. Hattori, C. Schroers, “Deep Deinterlacing”, SMPTE 2020

[Bhat21] M. Bhat, J. -M. Thiesse and P. L. Callet, "VVC partitioning decision driven by machine learning for a comprehensive hardware encoder," 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), 2021.

[Bianco18] S. Bianco, L. Celona, P. Napoletano and R. Schettini, "On the use of deep learning for blind image quality assessment", Signal Image and Video Processing, vol. 12, no. 2, pp. 355-362, Feb. 2018.

[Bishop95] Christopher Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, 1995, ISBN 0-19-853864-2.

[Bosse17] S. Bosse, D. Maniry, K-R. Müller, T. Wiegand and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment", IEEE Trans. Image Proc., vol. 27, no. 1, pp. 206-219, Oct. 2017.

[Coward16] Mike Coward, “AI Encoding”, Video @ Scale 2016, https://www.facebook.com/atscaleevents/videos/ai-encoding-at-scale/1682906415315789/

[Goodfellow16] Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016.

[Google22] https://www.washingtonpost.com/technology/2022/06/11/google-ai-lamda-blake-lemoine/, June 2022.

[He16] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”, Computer Vision and Pattern Recognition (CVPR), 2016.

[Kim18] Kim, W., Kim, J., Ahn, S., Kim, J., Lee, S. (2018). Deep Video Quality Assessor: From Spatio-Temporal Visual Sensitivity to a Convolutional Neural Aggregation Network. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science(), vol 11205.

[Kim19] J. Kim, S. Blasi, A. S. Dias, M. Mrak and E. Izquierdo, “Fast Inter-prediction Based on Decision Trees for AV1 Encoding”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1627-1631.

[Kim20] W. Kim, A. -D. Nguyen, S. Lee and A. C. Bovik, "Dynamic Receptive Field Generation for Full-Reference Image Quality Assessment," in IEEE Transactions on Image Processing, vol. 29, pp. 4219-4231, 2020.

[Kuanar18] S. Kuanar, C. Conly and K. R. Rao, "Deep Learning Based HEVC In-Loop Filtering for Decoder Quality Enhancement," 2018 Picture Coding Symposium (PCS), 2018, pp. 164-168.

[Li17] Y. Li, B. Li, D. Liu and Z. Chen, "A convolutional neural network-based approach to rate control in HEVC intra coding," IEEE Visual Communications and Image Processing (VCIP), 2017.

[Liu16] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji and D. Wang, "CU Partition Mode Decision for HEVC Hardwired Intra Encoder Using Convolution Neural Network", IEEE Transactions on Image Processing, vol. 25 (11), pp. 5088-5103, Nov. 2016.

[Liu22] Y. Liu, M. Abdoli, T. Guionnet, C. Guillemot and A. Roumy, "Light-Weight CNN-Based VVC Inter Partitioning Acceleration," 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2022.

[Mao16] X.-J. Mao, C. Shen and Y.-B. Yang, Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections, Barcelona, SPAIN:NIPS, 2016.

[Pfaff18] J. Pfaff, P. Helle, D. Maniry, S. Kaltenstadler, W. Samek, H. Schwarz, D. Marpe, and T. Wiegand, "Neural network based intra prediction for video coding," SPIE Applications of Digital Image Processing XLI, vol. 10752, 2018.

[Santamaria20] M. Santamaria, S. Blasi, E. Izquierdo and M. Mrak, "Analytic Simplification of Neural Network Based Intra-Prediction Modes for Video Compression", IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2020.

[Simonyan14] Simonyan, Karen & Zisserman, Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition", 2014, arXiv 1409.1556.

[Su19] H. Su, M. Chen, A. Bokov, D. Mukherjee, Y. Wang and Y. Chen, "Machine Learning Accelerated Transform Search for AV1," Picture Coding Symposium (PCS), 2019.

[vqeg22] NRMetricFramework, https://github.com/NTIA/NRMetricFramework

[Warden20] Pete Warden and Daniel Situnayake, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers", 1st Edition, O'Reilly, 2020.

[Wu21] G. Wu, Y. Huang, C. Zhu, L. Song and W. Zhang, "SVM Based Fast CU Partitioning Algorithm for VVC Intra Coding," 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021.

[Yang17] R. Yang, M. Xu and Z. Wang, Decoder-side HEVC Quality Enhancement with Scalable Convolutional Neural Network, Hong Kong, China:IEEE, ICME, 2017.

[You19] J. You and J. Korhonen, "Deep Neural Networks for No-Reference Video Quality Assessment," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 2349-2353, doi: 10.1109/ICIP.2019.8803395.

