Learned Enhancement Filters for Image Coding for Machines

Jukka I. Ahonen*, Ramin G. Youvalari†, Nam Le*†, Honglei Zhang†, Francesco Cricri†,
Hamed Rezazadegan Tavakoli†, Miska M. Hannuksela†, Esa Rahtu*
*Tampere University, Tampere, Finland
†Nokia Technologies, Tampere, Finland

***Abstract*—Machine-To-Machine (M2M) communication applications and use cases, such as object detection and instance segmentation, are becoming mainstream nowadays. As a consequence, majority of multimedia content is likely to be consumed by machines in the coming years. This opens up new challenges on efficient compression of this type of data. Two main directions are being explored in the literature, one being based on existing traditional codecs, such as the Versatile Video Coding (VVC) standard, that are optimized for human-targeted use cases, and another based on end-to-end trained neural networks. However, traditional codecs have significant benefits in terms of interoperability, real-time decoding, and availability of hardware implementations over end-to-end learned codecs. Therefore, in this paper, we propose learned post-processing filters that are targeted for enhancing the performance of machine vision tasks for images reconstructed by the VVC codec. The proposed enhancement filters provide significant improvements on the target tasks compared to VVC coded images. The conducted experiments show that the proposed post-processing filters provide about 45% and 49% Bjøntegaard Delta Rate gains over VVC in instance segmentation and object detection tasks, respectively.**

***Index Terms*—Image and video coding for machines, post-processing filter, image compression, perceptual loss, VVC**

I. Introduction

Images and video data consume the majority of the Internet bandwidth, globally. Additionally, because of the technological breakthroughs in artificial intelligence, in a few years most of such data is likely to be consumed by machines. According to an estimate by Cisco Annual Internet Report [1], by the year 2023, half of the internet traffic will be the Machine-To-Machine (M2M) communication data. As a consequence, there is a need to develop novel compression technologies that can cope with such demands in M2M communications in a more efficient way than the current codecs.

The existing state-of-the-art traditional video codecs, such as the High Efficiency Video Coding (HEVC) [2] and the brand-new Versatile Video Coding (VVC) [3] standards, are developed in such a way that they provide significant compression gains for content that is going to be consumed by humans. Hence, their performance may not be optimal for the data that is consumed by machines, for example, machine vision tasks such as object detection, object tracking and instance segmentation. Recently, JPEG-AI group of JPEG [4] as well as Video Coding for Machines (VCM) Ad-hoc group of MPEG [5] have initiated activities in order to standardize machine-oriented image and video compression technologies, respectively.

Many methods have been proposed to make image compression more efficient for machine consumption. In [6] and [7], the authors focus on fine-tuning the traditional codec to improve task performance on target machines. Since the codec is developed for human consumption, these methods may not be the most optimal for machine vision tasks. Neural networks-based codecs have been a popular topic recently on replacing traditional image codecs for both human and machine consumption [8]–[14]. For post-processing filters, there have been proposals before, such as [15] which considers a convolutional neural network (CNN) based post-processing filter as a method to improve the visual quality for human consumption. However, a post-processing filter for improving the performance of machine tasks has not been explored yet.

In this paper, we propose a system that makes use of a traditional codec, e.g., VVC, for compression stage followed by a learned enhancement filter that is applied to the decoded content. The aim of the learned filter is to enhance the quality of the decoded data which results in the improvement of the performance of machine tasks. For this purpose, we propose three neural network-based enhancement filters that have the same network architecture but are trained with different loss functions. The first filter, referred to as *Baseline Fidelity Enhancement (BFE)* filter, is trained with only mean squared error (MSE) loss and its main goal is to improve the visual quality of the decoded images. Secondly, *Task-Specific Enhancement (TSE)* filter is proposed to improve the quality of the content for the target machine vision task by incorporating the task loss of the target task network in the training phase. Finally, the *Task-Agnostic Enhancement (TAE)* filter is proposed. The *TAE* filter is trained by optimizing the perceptual loss [16] between the output image and the uncompressed image aiming at improving the quality of the output image for various machine tasks and different target task networks. Our experiments show that the proposed *TSE* enhancement filters achieve more than 45% Bjøntegaard Delta Rate (BD-Rate) [17] gains for object detection and instance segmentation tasks over the plain VVC method. Moreover, the proposed *TAE* enhancement filters can achieve significant gains over VVC on various machine tasks and task networks without incorporating the task network during the training.

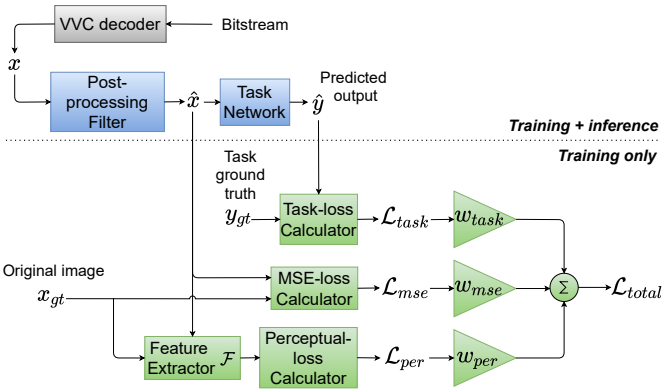


Fig. 1: Overview of the system. Triangles denote multiplication by the weights inside of them and sigma denotes summation.

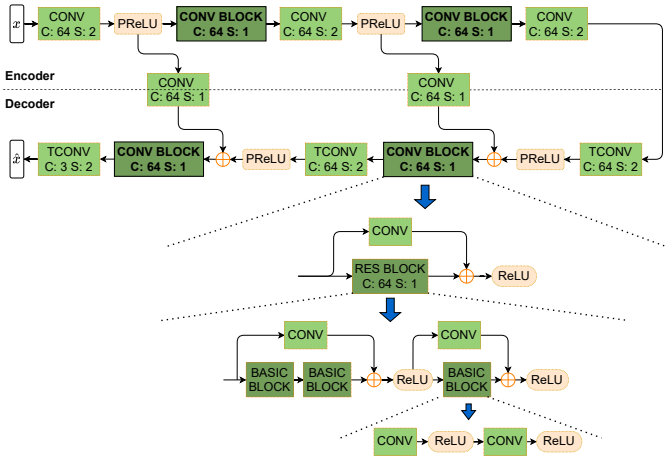


Fig. 2: Structure of the proposed enhancement filter. x is the VVC decoded image and $\hat{x}$ is the output of the enhancement filter. TCONV denotes a transposed convolutional layer. For every convolutional block, C denotes the number of output channels and S the stride. All the children blocks inherit the parameters from their parent blocks.

II. Proposed Methods

This section describes the proposed system and the enhancement filter's architecture for image coding for machines. Fig. 1 illustrates the overview of the proposed system in this paper. As shown in the figure, the compression and decompression phases of the data are performed with a traditional video codec, such as VVC. As previously described, VVC is optimized for human-targeted use cases, thus, it may not always be an optimal choice for machine-oriented use cases such as machine vision tasks. Particularly, in low bitrate cases, traditional video codecs result in poor task performances due to the heavy quantization processes. In order to improve the efficiency of traditional codecs for machine vision tasks, we propose to apply a post-processing enhancement filter to the decoded content to make them suitable for the target tasks.

The proposed post-processing enhancement filter is a convolutional neural network (CNN) based autoencoder with lateral and residual connections as shown in Fig. 2. The structure of the post-filter is similar to the autoencoder architecture in [13], [14]. The differences consist of having only 2 lateral connections, 3 basic block components (instead of 5), and the channel number of the last layer of the encoder is set to 64. This structure was selected empirically based on experiments with different structures. The filter is trained and kept unmodified throughout the tests, having $782,663$ trainable parameters.

Task network The task network used in the pipeline is based on Mask R-CNN for instance segmentation [18]. In our tests, we use the network with pretrained weights, which are always kept frozen. The task loss that we use for training one of our enhancement filters is defined similarly as the training loss that was originally used for training the task network:

$$\mathcal{L}_{task} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{preg} + \mathcal{L}_{obj} + \mathcal{L}_{mask}, \quad (1)$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{reg}$, $\mathcal{L}_{preg}$, $\mathcal{L}_{obj}$ and $\mathcal{L}_{mask}$ denote classification loss, regression loss, region proposal regression loss, objectness loss and mask prediction loss, respectively. The task loss is calculated between the predicted output $\hat{y}$ and the task ground truth y_{gt} as seen in Fig. 1.

Perceptual loss The perceptual loss calculation in Fig. 1 is introduced for generalization purposes and to make the system more task-agnostic. To extract the features for perceptual loss calculation, a VGG-16 model [19] pretrained on ImageNet dataset [20] is used as the feature extractor $\mathcal{F}$. The perceptual loss of the system is defined as:

$$\mathcal{L}_{per} = \mathbf{MSE}(\mathcal{F}_2(\hat{x}), \mathcal{F}_2(x_{gt})) + \mathbf{MSE}(\mathcal{F}_4(\hat{x}), \mathcal{F}_4(x_{gt})), \quad (2)$$

where $\mathbf{MSE}$ denotes mean squared error operator and the $\mathcal{F}_i(z)$ denotes the feature tensor extracted at the i^{th} Max-pooling layer of the VGG-16 for the input z.

Total loss The total loss of the system is given by:

$$\mathcal{L}_{total} = w_{mse} \cdot \mathcal{L}_{mse} + w_{task} \cdot \mathcal{L}_{task} + w_{per} \cdot \mathcal{L}_{per}, \quad (3)$$

where $\mathcal{L}_{task}$ and $\mathcal{L}_{per}$ are the loss terms described in (1) and (2), respectively; and $\mathcal{L}_{mse}$ is the mean squared error between the filtered image $\hat{x}$ and the ground-truth image x_{gt}. Furthermore, the corresponding weights w_{mse}, w_{task} and w_{per} are introduced in order to balance the losses. The weights' values are selected empirically based on our experiments.

Enhancement filters We propose three different enhancement filters. In the first proposed filter, we aimed at enhancing the fidelity of the VVC decoded images by using only the MSE-loss term in the training phase. For this, we used the weighting setting in Eq. (3) as following: $w_{mse} = 1$, $w_{task} = 0$ and $w_{per} = 0$. This filter is used as the baseline model for the other proposed filters, hence, it is referred to as *Baseline Fidelity Enhancement* (*BFE*) filter, hereafter.

In the second proposed filter, named as *Task-Specific Enhancement* (*TSE*) filter, the enhancement filter is trained in such a way that it improves the performance for a specific task. For this purpose, we included the task loss of the target task network into the system and the weights in Eq. (3) are set as

$w_{mse} = 1$, $w_{task} = 0.01$ and $w_{per} = 0$. To reduce the training effort, the *TSE* filter is initialized to the best-performing *BFE* model in the training phase.

Directly optimized for the target task network, the *TSE* model can provide significant gains for the target machine task. However, this method requires to run the target task network during the training in order to build an optimal enhancement filter. In real-world applications, a general codec may serve various machine tasks using different task networks. This creates issues in generalization of the enhancement filter and the generated models may not work efficiently when applied to different task networks than the ones they are trained on. In order to alleviate this issue, we propose the third enhancement filter, named as *Task-Agnostic Enhancement* (*TAE*) filter. To train the *TAE* filter, the perceptual loss w_{per} is used instead of the task loss. The corresponding weights in Eq. (3) are set as following: $w_{mse} = 1$, $w_{task} = 0$ and $w_{per} = 0.01$. Similar to *TSE* filter, the weights of the best-performing *BFE* model are used to initialize the *TAE* filter before training.

III. Experiments

A. Experimental Setup

The proposed enhancement filters specified in Section II were trained using a subset of the training set of the Open Images dataset [21]. For that we randomly selected $30,000$ images that contain at least 1 instance of these 5 classes: *human*, *car*, *cat*, *dog* and *bird*. For the evaluation, we used two distinct datasets: $\mathcal{X}_1$) $2,352$ randomly selected images from the corresponding 5 classes of validation set in the Open Images dataset, and $\mathcal{X}_2$) validation set of the COCO dataset which contains $5,000$ images and 80 classes [22].

For the compression and color conversion stages, we followed the MPEG VCM's guidelines which are specified in evaluation framework for Video Coding for Machines [23]. Accordingly, the training and validation sets were compressed using the VVC's reference software VTM-8.2 [1] with All Intra configuration. Datasets were encoded according to JVET common test conditions [24] with quantization parameters (QPs) in $\{22, 27, 32, 37, 42, 47, 52\}$. All the original data were converted to YUV420 color space prior to encoding, and the decoded images were converted back to the original RGB format before applying enhancement filters or inputting them to the task network.

For each QP point, we trained a separate model which results in a total of 7 models for each filter that operate based on the quality of the VVC encoded content. The *BFE* models were trained for 150 epochs (until convergence), which took around 45 minutes per epoch. To train the models for *TSE* and *TAE* filters, the best-performing *BFE* models of the corresponding QPs were used as the initializations. The *TSE* and *TAE* filters were trained for 70 epochs, which took around 3h5min and 1h20min per epoch, respectively. All the filters were trained on Nvidia DGX1 system on a Tesla V100-SXM2 16GB GPU.

[1]https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM

TABLE I: Average BD-Rate (%) results of the proposed methods over plain VVC

Task network	BD-Rate (with mAP)		
	BFE	***TSE***	***TAE***
Instance Segmentation (Mask R-CNN)	–30.72%	–45.79%	–40.30%
Object Detection (Faster R-CNN)	–13.91%	–49.39%	–40.57%
Object Detection with 5 classes (YOLOv5)	–7.43%	–13.51%	–30.02%
Object Detection with 80 classes (YOLOv5)	–6.65%	–5.14%	–24.85%

For the $\mathcal{X}_1$ dataset, all of the proposed filters were evaluated on both instance segmentation and object detection tasks. The pretrained models for these tasks are `mask_rcnn_X_101_32x8d_FPN_3x` and `faster_rcnn_X_101_32x8d_FPN_3x`, respectively, provided by Detectron2 [25]. We used Mean Average Precision (mAP@0.5) [26] as metric for task performance and Bits Per Pixels (BPP) as metric for bitrate. The different bitrates were achieved by encoding the validation dataset with different QPs using VTM. The performance of the proposed methods were compared to plain VVC results (i.e., none of the proposed enhancement filters are applied to the decoded images) using Bjøntegaard Delta Rate (BD-Rate) metric [17]. For the $\mathcal{X}_2$ COCO validation dataset, the proposed methods were evaluated on object detection task. `YOLOv5s`, which is the smallest version of the YOLOv5 [2] network, was used for the evaluation. For the performance evaluation of YOLOv5 we used Mean Average Precision (mAP@$[0.5 : 0.05 : 0.95]$) metric[3].

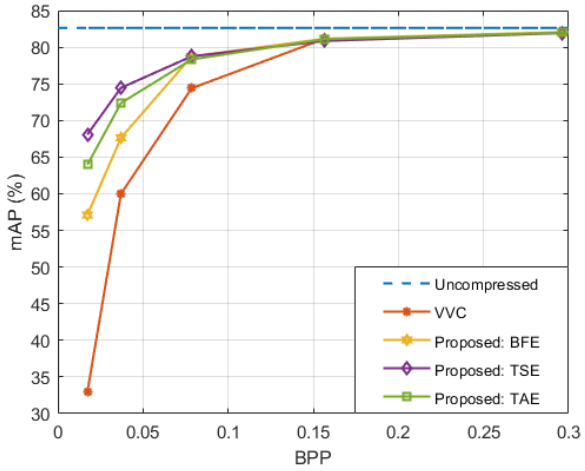
B. Experimental Results

The BD-Rate performance of the proposed methods, in terms of mAP and BPP, compared to plain VVC is shown in Table I. As shown in the results, all the proposed enhancement filters improve the performance in both object detection and instance segmentation tasks. For the instance segmentation and the object detection tasks with R-CNN task networks on $\mathcal{X}_1$ dataset, the highest gain is obtained with the *TSE* filter. The *TAE* filter achieves the best gain in object detection task with YOLOv5 network on $\mathcal{X}_2$ dataset, whether all the 80 classes are considered or just the 5 classes corresponding to the $\mathcal{X}_1$ dataset. The reason for such difference in performance is that the *TSE* method benefits from using the task loss of the Mask R-CNN network in the training phase. As a result, the trained model is not able to function as expected when a different task network is used. As can be seen from the object detection results with YOLOv5 network, the *TSE* filter provides significantly less gain than the *TAE*. Whereas the *TAE*, which does not use the task loss of any specific task network in the training phase, is task agnostic and performs very well on different tasks with different task networks.

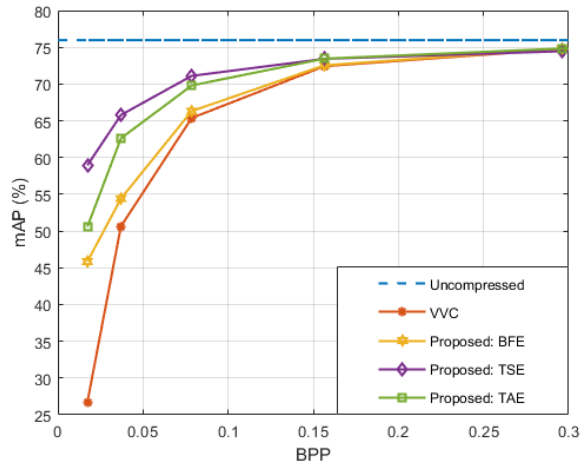
Fig. 3 illustrates the rate-performance curves of different methods on the target tasks. As shown in the figure, the proposed enhancement filters provide significant improvements

[2]https://pytorch.org/hub/ultralytics_yolov5/

[3]https://cocodataset.org/#detection-eval



(a) Instance segmentation task with Mask R-CNN



(b) Object detection task with Faster R-CNN

Fig. 3: Rate-performance curves of different methods on $\mathcal{X}_1$

over the plain VVC approach, particularly in lower bitrates. In the high bitrate range, for example, QP points 22 and 27, the proposed enhancement filters do not bring noticeable gains compared to the VVC anchor. This shows that at a high bitrate range the VVC decoded images are already in a high quality, thus the enhancement filters are not able to improve the content significantly for machine tasks. This phenomenon is visible in the performance curves where the task performance in high bitrates are very close to the mAP scores of uncompressed dataset. Similar behavior is also observed in [14].

As mentioned before, two color conversion operations (RGB $\rightleftarrows$ YUV) are applied in the plain VVC pipeline. This results in significant degradation of the quality of the final output images that are consumed by machine vision tasks as well as humans. The proposed enhancement filters improve the objective and subjective quality of the images significantly. Table II demonstrates the average improvements of the peak signal-to-noise-ratio (PSNR) values (in the experimented 7 QP points) in different methods compared to plain VVC on $\mathcal{X}_1$. As shown, PSNR gains of up to 5.55 [dB] is achieved. Moreover, the BD-Rate results, in terms of PSNR and BPP, illustrate that the proposed enhancement filters provide significant gains over the VVC decoded images after they are converted back to RGB color space. However, this aspect requires further study in order to avoid such quality degradation caused by color space conversion stages.

Fig. 4 illustrates examples from $\mathcal{X}_1$ on how the proposed filters improve the segmentation and detection performances over VVC at QPs 42 and 47. There can be clearly seen that for plain VVC encoded images, the Mask R-CNN task network is not able to predict high-quality segmentation masks, and it also produces extra false positive predictions in both QPs. These detection mistakes can be seen corrected by all of the proposed filters along with a noticeable improvement on quality of the segmentation masks.

TABLE II: Average BD-Rate (%) and PSNR [dB] gains of the proposed methods over the plain VVC in RGB color space

Filter	*BD-Rate (with PSNR)*	*PSNR gain [dB]*
BFE	–84.48%	+5.55
TSE	–78.65%	+4.26
TAE	–81.52%	+5.35

IV. Conclusion

In this paper, we proposed three post-processing filters in order to enhance the performance of the decoded data with traditional video codec VVC on machine visions tasks. The proposed enhancement filters improve the machine task accuracy of the VVC encoded images significantly. Average BD-Rate gains over 45% and 49% compared to plain VVC approach were obtained for instance segmentation and object detection tasks, respectively. Furthermore, the proposed *Task-Agnostic Enhancement* (*TAE*) filter attains significant improvements on machine tasks without using the target task network in the encoding and training phases.

References

[1] Cisco annual internet report (2018–2023) white paper. Accessed: Aug. 2021. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] Recommendation ITU-T H.265 | ISO/IEC 23008-2, "High efficiency video coding," 2013.

[3] Recommendation ITU-T H.266 | ISO/IEC 23090-3, "Versatile video coding," 2020.

[4] J. Ascenso, "JPEG AI use cases and requirements," in *ISO/IEC JTC1/SC29/WG1 M90014*, Jan 2021.

[5] "Call for evidence for video coding for machines," in *ISO/IEC JTC 1/SC29/WG 2, m55065*, Oct 2020.

[6] K. Fischer, F. Brand, C. Herglotz, and A. Kaup, "Video coding for machines with feature-based rate-distortion optimization," *IEEE 22nd International Workshop on Multimedia Signal Processing*, p. 6, September 2020.

[7] B. Brummer and C. de Vleeschouwer, "Adapting JPEG XS gains and priorities to tasks and contents," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 629–633.

[8] H. Zhang, F. Cricri, H. R. Tavakoli, N. Zou, E. Aksu, and M. M. Hannuksela, "Lossless image compression using a multi-scale progressive statistical model," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

Fig. 4: Examples of improvements on object detection and instance segmentation performances at QP 42 (top images) and QP 47 (bottom images)

[9] N. Zou, H. Zhang, F. Cricri, H. Tavakoli, J. Lainema, M. Hannuksela, E. Aksu, and E. Rahtu, "L^{2}C – learning to learn to compress," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, ser. IEEE International Workshop on Multimedia Signal Processing. IEEE, Sep. 2020, pp. 1–6.

[10] N. Zou, H. Zhang, F. Cricri, H. R. Tavakoli, J. Lainema, E. Aksu, M. Hannuksela, and E. Rahtu, "End-to-end learning for video frame compression with self-attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 142–143.

[11] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7939–7948.

[12] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," pp. 10 771–10 780, 2018.

[13] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: an end-to-end learned approach," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1590–1594.

[14] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, H. R. Tavakoli, and E. Rahtu, "Learned image coding for machines: A content-adaptive approach," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

[15] Z. Wang, R.-L. Liao, and Y. Ye, "Joint learned and traditional video compression for p frame," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 560–564.

[16] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711.

[17] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T Video Coding Experts Group (VCEG)*, 2001.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[19] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 730–734.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] "Open Images V6." [Online]. Available: https://storage.googleapis.com/openimages/web/index.html

[22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2015.

[23] M. Rafie, Y. Zhang, and S. Liu, "Evaluation framework for video coding for machines," *ISO/IEC JTC 1/SC 29/WG 2, MPEG Technical requirements, Document: N104*, July 2021.

[24] F. Brossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, "JVET common test conditions and software reference configurations for sdr video," *Joint Video Experts Team (JVET), Document: JVET-N1010*, March 2019.

[25] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[26] "OpenImages evaluation." [Online]. Available: https://storage.googleapis.com/openimages/web/evaluation.html