BRIDGING THE GAP BETWEEN IMAGE CODING FOR MACHINES AND HUMANS

Nam Le[*][†]   *Honglei Zhang*[†]   *Francesco Cricri*[†]   *Ramin G. Youvalari*[†]
Hamed Rezazadegan Tavakoli[†]   *Emre Aksu*[†]   *Miska M. Hannuksela*[†]   *Esa Rahtu*[*]

[†]Nokia Technologies, [*]Tampere University

ABSTRACT

Image coding for machines (ICM) aims at reducing the bitrate required to represent an image while minimizing the drop in machine vision analysis accuracy. In many use cases, such as surveillance, it is also important that the visual quality is not drastically deteriorated by the compression process. Recent works on using neural network (NN) based ICM codecs have shown significant coding gains against traditional methods; however, the decompressed images, especially at low bitrates, often contain checkerboard artifacts. We propose an effective decoder finetuning scheme based on adversarial training to significantly enhance the visual quality of ICM codecs, while preserving the machine analysis accuracy, without adding extra bitcost or parameters at the inference phase. The results show complete removal of the checkerboard artifacts at the negligible cost of -1.6% relative change in task performance score. In the cases where some amount of artifacts is tolerable, such as when machine consumption is the primary target, this technique can enhance both pixel-fidelity and feature-fidelity scores without losing task performance.

Index Terms— Image coding for machines, GANs, finetuning, VCM

1. INTRODUCTION

Traditional image coding systems such as JPEG [1] and VVC [2] compress images to reduce the amount of the data required for storing and transferring while maintaining satisfactory visual quality for human viewers. On the other hand, Image Coding for Machines (ICM) systems aim to provide better compression efficiency when the primary consumers are machines that perform certain computer vision tasks [3, 4]. Recent works on ICM have shown convincing evidences that machine-oriented image codecs significantly improve the compression efficiency over traditional image codecs [3,5,6]. In many applications, e.g. surveillance, although machines are the main consumers, human involvement is occasionally required or even mandatory. Thus, support for human consumption is desired for ICM codecs.

However, the outputs of convolutional neural network (CNN) based ICM codecs, in particular at low bit-rate range, usually contain repetitive artifact patterns, also referred to as "checkerboard artifacts" [5, 6], which could be easily perceived by humans as "distortion" and significantly degrade the experience for human viewers. To support human consumption, ICM systems often include separate branches for machine consumption and human consumption [3,4,7], which unavoidably incurs bitrate overheads due to the need for encoding two bitstreams. In addition, such architecture design increases the complexity of the codec significantly.

In this paper, we propose a novel training technique for end-to-end learned ICM codecs to improve the visual quality without additional processing components or bitstreams for human consumption. In the proposed technique, selected layers from the decoder of a pretrained ICM codec are finetuned using a PatchGAN [8] adversarial training mechanism. Our experiments show that after being finetuned, the quality of the decoded images is significantly improved without compromising the performance of the machine tasks.

2. RELATED WORK

End-to-end learned codecs are a promising direction over conventional codecs for machine consumption since the codec may be optimized directly via a learning signal obtained through the task networks performing computer vision tasks. The authors of [5, 9] proposed an ICM codec with encoder, decoder, and probability model implemented by neural networks. An input image is first converted to a latent representation. Then, an arithmetic encoder encodes the quantized latent representation using the prior distribution estimated by the probability model. At the decoder side, an arithmetic decoder decodes the bitstream into the latent representation with the prior from the same probability model as the one on the encoder side and the ICM decoder reconstructs the image from the latent representation.

Although machine consumption is the main target of an ICM codec, human consumption is required for many applications, e.g., surveillance systems. Various system architectures have been proposed to support this requirement. Seppala *et al.* [7] proposed a multi-layer coding system comprising two branches: one implemented by a conventional codec for human consumption and one implemented by a learned codec that transfers additional information to enhance the per-

formance of machine consumption. A similar architecture is proposed in [10] where multiple learned codecs are used for human vision and various machine tasks separately. These designs significantly complicate the system and make optimization difficult. Also, an additional bitstream is required for targeting human consumption of images. As the task networks of these ICM systems are trained using images which are either uncompressed or compressed with high fidelity for human vision, the outputs of the ICM codecs are visually acceptable for humans. However, clearly noticeable checkerboard artifacts are present in the output, especially at lower bitrates, which significantly degrade the quality perceived by human viewers. Thus, it is important to improve the quality of the outputs of ICM codecs for human consumption without significantly degrading the machine task. Zhang *et al.* [11] propose to blur the features in CNN to mitigate the aliasing problem, but we found it is ineffective in deeper CNNs.

Generative Adversarial Networks (GANs) were introduced as a system for generating realistic natural images and later used in many other applications, including for visual quality enhancement. Isola *et al.* [8] introduced the Patch-GAN technique for image-to-image translation, which restricts the attention of the networks in local image patches, thus enforces high-frequency correctness. Further, in [12], both the global and local patches are used for image deblurring. GANs have been also used in video coding systems . In [13], the authors used conditional GANs architecture to improve the perceptual visual quality of learned image compression targeting human consumption. The analysis in [14] suggests that the said adversarial system improves task performance of the codec.

Next, we will describe the proposed PatchGAN-based method that significantly improves the visual quality of the compressed images of an ICM codec for human consumption.

3. PROPOSED METHOD

3.1. Baseline image codec model

Similar to [5, 6, 9], we first train a base codec using metrics such as mean squared error (MSE) for human consumption, then finetune it to achieve better task performance. We follow the overall model training strategy as described in [5].

Our codec comprises 3 types of building blocks: **B**asic, **D**ownsampling, and **U**psampling. The **B** blocks are type A basic blocks defined in ResNet [15] with 128 output channels and PReLU activation function [16]. The **D** blocks are **B** blocks that have stride-2 *conv* (*convS2*) for their first *conv* and shortcut projection. Similarly, the **U** blocks use a sequence of a 512-channel *conv* followed by a PixelShuffle [17] layer (*PSConv*) for those parts. The layer orders in the encoder and decoder of the codec can be described as **D-B-D-B**-*convS2* and **B-U-B-U-B**-*PSConv*, respectively.

The training loss, which imposes high task performance, follows the same formulation in [5]:

$$\mathcal{L}_{total} = w_{rate}\mathcal{L}_{rate} + w_{mse}\mathcal{L}_{mse} + w_{task}\mathcal{L}_{task}, \quad (1)$$

where w_{rate}, w_{mse}, w_{task} are scalar weights, $\mathcal{L}_{rate}$, $\mathcal{L}_{mse}$, $\mathcal{L}_{task}$ are the loss terms for bitcost, distortion and task performance, respectively. These terms are defined in the same way as in [5], except for $\mathcal{L}_{task}$ which is replaced by a proxy loss $\mathcal{L}_{proxy}$ in our setup. Due to the correlations in the intermediate level features of different vision tasks [9], we can use an intermediate feature distortion metric as a proxy for $\mathcal{L}_{task}$, thus making the codec task-agnostic. Additionally, using a feature-based loss as such enables the training of the model with cropped images which is much more efficient. Similar to [9, 18], we define $\mathcal{L}_{proxy}$ as follows:

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

where $\mathbf{MSE}$ denotes the mean square error and $\mathcal{F}_k(\boldsymbol{t})$ denotes a feature tensor extracted from the P_k layer of the FPN [19] feature extractor network given the input $\boldsymbol{t}$.

3.2. Codec finetuning with PatchGAN

In order to improve the visual quality of a learned image codec, one can resort to re-training the model for human vision distortion metrics such as PSNR or SSIM [20]. However as demonstrated in recent works [6, 14], high scores in these metrics usually come with the cost of significant degradation of task performance. Furthermore, [9] provides us with evidences that even finetuning a single component of the system can achieve good performance gains. We propose a model finetuning scheme where a PatchGAN [8] discriminator is applied to guide the codec to effectively suppress the artifacts in its outputs. In this scheme, illustrated by Fig. 1, the image codec acts as the generator $\mathcal{G}$ that takes uncompressed image $\boldsymbol{x}$ as input and generates the compressed image $\hat{\boldsymbol{x}} = \mathcal{G}(\boldsymbol{x})$, and the discriminator $\mathcal{D}$, implemented as a CNN following [8], tries to distinguish the image patches from $\boldsymbol{x}$ from the ones from $\hat{\boldsymbol{x}}$ by detecting the noise pattern in $\hat{\boldsymbol{x}}$ produced by the codec. The reasons for choosing PatchGAN over normal GANs are: *i)* a small patch from $\hat{\boldsymbol{x}}$ can contain an adequate amount of artifacts; *ii)* the training runs much faster with small patches; *iii)* we can limit the random hallucinations that are typically introduced in GAN-based training, which may damage the task performance. Our experiments show that the compression artifacts are sufficiently removed by finetuning only the decoder. With the encoder frozen, the bitrate is unchanged.

Finetuning objective: We empirically found the proposed system is sufficiently robust to different adversarial loss functions. Therefore, we choose the vanilla GAN objective function for the sake of simplicity:

$$\mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}) = \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{x}}}[\log \mathcal{D}(\boldsymbol{x})] + [\log(1 - \mathcal{D}(\mathcal{G}(\boldsymbol{x})))] \quad (3)$$

Unlike [8], we only sample a small fixed number of square patches to feed the discriminator since the artifacts appear uniformly in the decoded images. Besides, we do not condition the discriminator on the input image x. In our system, the generator is a pretrained codec that outputs a fair reconstruction $\hat{x}$ of x. Therefore, we use ℓ_2-distance to impose a weak pixel-fidelity constraint. Our funetuning objective is formulated as:

$$G^* = \arg \min_{\mathcal{G}} \max_{\mathcal{D}} \underbrace{\mathbb{E}_x \|x - \mathcal{G}(x)\|_2^2}_{\mathcal{L}_{L2}} + \lambda_a \cdot \mathcal{L}_{GAN}(\mathcal{G}, \mathcal{D}) \tag{4}$$

where the weight of the adversarial loss is set as $\lambda_a = 10^{-3}$ in our tests if not stated otherwise.
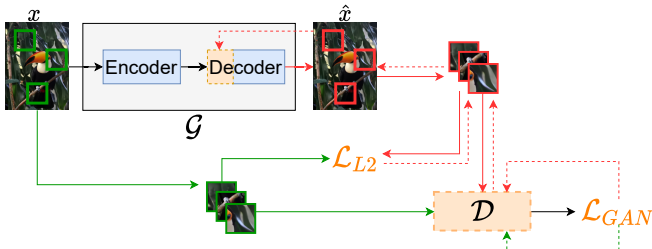

Fig. 1. The overview of the finetuning scheme using Patch-GAN discriminator. The dashed lines denote gradient back-propagation flows and dotted boxes denote the parameters getting updated by the optimizer. The green lines indicate the data from the input of the ICM codec and the red lines indicate the data from the output of the ICM codec.

4. EXPERIMENTS

4.1. Experimental setup

Dataset and task evaluation: We used the Open Images V6 [21] dataset of 9.2M high quality JPEG images for training and evaluation. The training and finetuning processes of the base model saved the model weights in subsequent checkpoints. Every checkpoint was trained with random 6000 images from the training split of the dataset. Each image was randomly cropped to a 512×512 patch. The task performance measured in mAP[1] was obtained by following the Instance Segmentation testing conditions in [22], which consists of tailored benchmarks for the practical use cases of MPEG Video Coding for Machines (VCM). In addition, we also carried out evaluations on other common visual metrics, namely PSNR, SSIM [20] and VGG19 [23] perceptual loss[2][3].

---

[1]https://storage.googleapis.com/openimages/web/evaluation.html
[2]https://pytorch.org/docs/stable/torchvision/models.html
[3]Same as proxy loss in [9] but apply on the 2nd and 5th layers.

Training: *The base codec* was trained following the strategy proposed in [5] for 200 checkpoints. The proxy loss $\mathcal{L}_{proxy}$ used the FPN backbone of the MaskRCNN [24] network[2] to extract the feature maps at the top-down pathway. *The PatchGAN-based finetuning* experiments were performed on different settings of patch sizes (64×64, 128×128), number of patches (1, 3, 5) per image and learning rates (2×10^{-5}, 2×10^{-9}). We empirically found that finetuning only the first 2 layers of the decoder sufficiently removed the artifacts effect whilst keeping the number of updated parameters to minimum, which led to reducing the time and resources for the finetuning. We finetuned the codec with Adam [25] optimizer for 50 checkpoints in each experiments, evaluated the task performance every 10 checkpoints, and evaluated all other metrics on the checkpoint with the highest task score in Table 1. Using a batch size of 4, the finetuning process took $\approx$ 7 hours per settings on an NVIDIA A100 GPU.

Table 1. Evaluation results of PatchGAN finetuning configurations on different metrics. $\uparrow$ denotes the higher the better and $\downarrow$ denotes the lower the better. The average bitrate of outputs is 0.057 bits per pixel. Configuration with low adversarial impact settings are marked with "*LI*"

Configuration	mAP$\uparrow$	PSNR$\uparrow$	SSIM$\uparrow$	VGG$\downarrow$
Base codec	**0.766**	27.695	0.709	0.495
3 patches (64x64)	0.752	28.526	0.764	0.497
3 patches (32x32)	0.750	28.316	0.740	0.537
5 patches (64x64)	0.749	**28.540**	**0.765**	0.499
1 patch (64x64)	0.748	28.518	0.763	0.502
3 patches (128x128)	0.754	28.498	0.763	0.502
3 patches (128x128), *LI*	**0.766**	28.129	0.726	**0.469**

4.2. Experimental results

Table 1 summarises the best task performing checkpoints of the configurations. Overall, all of the configurations can considerably increase PSNR and SSIM scores, which are the metrics that are closely correlated to human perception. The minor differences between the lower rows indicate the robustness of the method. Patch size of 32×32 shows less appealing results than other sizes in all metrics, possibly due to limited context exposure that hinders the finetuning process. Marginal task performance losses are observed when applying the proposed finetuning in most cases. Therefore, we carried out experiments where the parameter updates due to the adversarial training were greatly attenuated, specifically by reducing λ_a to 1×10^{-4} and the learning rate to 2×10^{-9}. The result shown on the last row of Table 1 proves that the visual metrics can be improved by the proposed technique without losing task performance.

Fig. 2 shows the close-ups of a few output examples of the compared codecs. We chose the outputs of 2 configura-
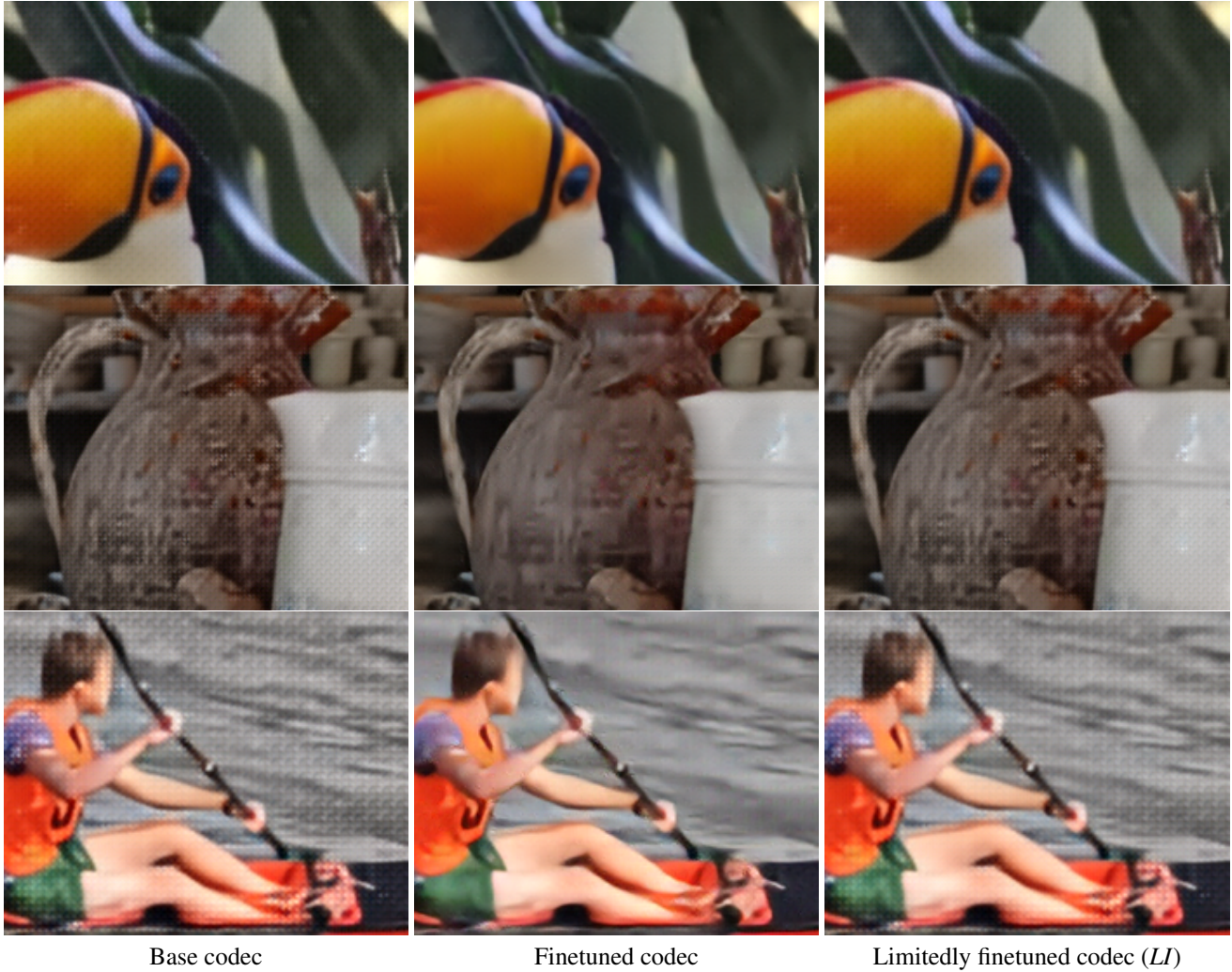
Base codec	Finetuned codec	Limitedly finetuned codec (*LI*)

Fig. 2. The codec finetuned with PatchGAN *(middle)* effectively removes the checkerboard artifacts commonly found in the decoded images of the NN-based convolutional codec such as the base model *(left)*, while the codec finetuned with limited adversarial impact *(right)* only mildly suppresses the artifacts. More examples available at: `https://flysofast.github.io/human-finetuned-icm/`.

tions that resulted in the closest task performance scores to the base codec for the visual comparison. The outputs of the base codec show the typical repetitive patterns found in decoded images of ICM codecs. After being finetuned, the artifacts are almost completely removed, while the image content is well-preserved without any clear random hallucinations. The codec that was finetuned with an attenuated adversarial-training could not completely remove the checkerboard patterns, however managed to prevent the task performance loss while improving the scores of visual metrics.

5. CONCLUSIONS

In this paper, we propose a novel codec finetuning scheme that can effectively eliminate the well-known "checkerboard" pattern produced by deep CNN autoencoders for ICM codecs. Our experimental results demonstrate that the artifacts can be sufficiently removed from the outputs of an ICM codec, thus greatly enhance the visual quality and reconstruction fidelity without any additional components or bitcost. Although the machine task accuracy may be affected by this process, the relative performance loss of -1.6% is negligible in considered cases. Furthermore, our study shows that the task performance loss could be totally avoided by confining the impact of the adversarial dynamics. In this settings, all of the benchmarked metrics are improved, suggesting more desirable results could be achieved with granularity in training setup. We also found that only a small subset of the codec parameters is responsible for the artifacts, future study on this subject could lead to interesting findings in neural networks explainability.

6. REFERENCES

[1] G. Hudson, A. Léger, B. Niss, and I. Sebestyén, "Jpeg at 25: Still going strong," *IEEE MultiMedia*, vol. 24, no. 2, pp. 96–103, 2017.

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

[3] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video Coding for Machine: Compact Visual Representation Compression for Intelligent Collaborative Analytics," *arXiv:2110.09241 [cs]*, Oct. 2021.

[4] L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," *IEEE Transactions on Image Processing*, vol. 29, pp. 8680–8695, 2020.

[5] 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.

[6] K. Fischer, F. Brand, and A. Kaup, "Boosting neural image compression for machines using latent space masking," *arXiv preprint arXiv:2112.08168*, 2021.

[7] J. Seppälä, H. Zhang, N. Le, R. G. Youvalari, F. Cricri, H. R. Tavakoli, E. Aksu, M. M. Hannuksela, and E. Rahtu, "Enhancing image coding for machines with compressed feature residuals," in *2021 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2021, pp. 217–225.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 5967–5976, IEEE.

[9] 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.

[10] H. Choi and I. V. Bajic, "Scalable image coding for humans and machines," 2022.

[11] R. Zhang, "Making convolutional networks shift-invariant again," in *International conference on machine learning*. PMLR, 2019, pp. 7324–7334.

[12] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

[13] F. Mentzer, G. Toderici, M. Tschannen, and E. Agustsson, "High-Fidelity Generative Image Compression," *arXiv:2006.09965 [cs, eess]*, Oct. 2020.

[14] K. Fischer, C. Forsch, C. Herglotz, and A. Kaup, "Analysis Of Neural Image Compression Networks For Machine-To-Machine Communication," in *2021 IEEE International Conference on Image Processing (ICIP)*, Sept. 2021.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, ISSN: 1063-6919.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[17] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[18] J. I. Ahonen, R. G. Youvalari, N. Le, H. Zhang, F. Cricri, H. R. Tavakoli, M. M. Hannuksela, and E. Rahtu, "Learned enhancement filters for image coding for machines," in *2021 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2021, pp. 235–239.

[19] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[20] 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.

[21] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[22] M. Rafie, Y. Zhang, and S. Liu, "Evaluation framework for video coding for machines (m57658)," *ISO/IEC JTC 1/SC 29/WG 2, MPEG Technical requirements, Document: N104*, July 2021.

[23] 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.

[24] 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.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.