Region of interest enabled learned image coding for machines

1st Jukka I. Ahonen
Nokia Technologies
Tampere University
Tampere, Finland
jukka.1.ahonen@nokia.com

2nd Nam Le
Nokia Technologies
Tampere University
Tampere, Finland
nam.le@nokia.com

3rd Honglei Zhang
Nokia Technologies
Tampere, Finland
honglei.1.zhang@nokia.com

4th Francesco Cricri
Nokia Technologies
Tampere, Finland
francesco.cricri@nokia.com

5th Esa Rahtu
Tampere University
Tampere, Finland
esa.rahtu@tuni.fi

***Abstract*—Image and video coding for machines has been recently gaining more and more interest from both the industry and the research community. One successful approach is based on end-to-end (E2E) learned compression and has shown significant gains over the state-of-the-art conventional image coding methods. However, one of the remaining challenges for such E2E-learned image codecs for machines is to adaptively allocate the bits over different regions of the image, while retaining the machine vision performance. In this paper, we propose a method that leverages Regions-Of-Interest (ROIs) for bitrate allocation within a Learned Image Codec (LIC) for machines. In particular, the proposed method reduces the bits allocated for the background regions of the image by reducing the variance of the elements corresponding to the background regions in the latent representation. This results in more heavily quantized background areas, while keeping the quality of the ROI areas suitable for machine tasks. The proposed method achieves significant gains, -15.80% and -22.43% Pareto BD-rate reduction, over the baseline LIC on object detection and instance segmentation tasks, respectively. To the best of our knowledge, this is the first research paper proposing an ROI-based inference-time technology for Learned Image Coding for machines.**

***Index Terms*—region of interest, learned image coding, video coding for machines, machine vision, neural networks**

I. Introduction

Learned image coding [1]–[5] has been a widely researched topic recently due to its superior performance in image coding compared to the state-of-the-art traditional codecs such as the High Efficiency Video Coding (HEVC) [6] and Versatile Video Coding (VVC) [7] in All-Intra configurations. Additionally, the need for codecs aimed at machines as the end-user has increased rapidly, in view of the increase of use cases, where machines are the main consumer, for example, in autonomous driving and surveillance systems. Related to this, machine-oriented learned image codecs (LIC) have also been explored in recent years. In [8], the authors propose an end-to-end learnable image coding system that uses a task performance-driven training loss. This work shows that a machine-oriented image codec can further minimize the bitstream size by compressing the less important regions more aggressively without penalizing the task performance, resulting in significantly better compression efficiency. Authors in [9] introduce an LIC capable of both human and machine-oriented compression. While these methods could outperform state-of-the-art traditional codecs in image coding for machines, they require, to some degree, the knowledge of the targeted task network in the training processes. In comparison, our method no longer relies on the task network information to gain coding efficiency.

In [10], [11], additional ideas to improve the task performance of existing image coding systems are presented. There has also been increasing interest in different techniques the LICs employ, one of the most prominent ones being Region Of Interest (ROI)-based methods, where the ROI information is utilized in the LIC in order to boost the coding performance further. In [12], the authors proposed an LIC where a binary mask, derived from a semantic segmentation network, is introduced to the input of the codec during the training. The trained network is then capable of taking the ROI indicating binary mask at the inference time to improve the compression efficiency. In [13] the authors propose a transformer-based LIC, which incorporates the usage of ROI information via prompt generation network using the input image, ROI mask and rate parameter, while the used loss function also takes into account the extra information. Most of the ROI techniques used in the LICs are designed for humans as the end user [14], [15]. For applications of Image Coding for Machines (ICM), the research field has not been thoroughly investigated. Moreover, all the existing methods require training the LIC in order to enable the ROI functionality. In this paper, we propose a simple, yet effective inference-time method that can enable a pretrained LIC to achieve effective spatial bitrate allocation, which in turn significantly increases the machine vision performance. This is achieved by leveraging the ROI information for manipulating the process of quantizing the latent representation output by a learned encoder.

II. Proposed method

This section describes the proposed ROI-based quantization manipulation method for an LIC. We refer to this method as ROI-LIC. In its essence, the ROI-LIC aims to apply coarser quantization to the latent representation in the LIC for the non-ROI areas, i.e., background areas. Thus, fewer bits are spent on such background areas, which leaves a higher bitrate budget to be spent on areas that are more important for the machine tasks, i.e., the ROIs. The method is inference-time only, hence no retraining of the LIC is required. Thus, the

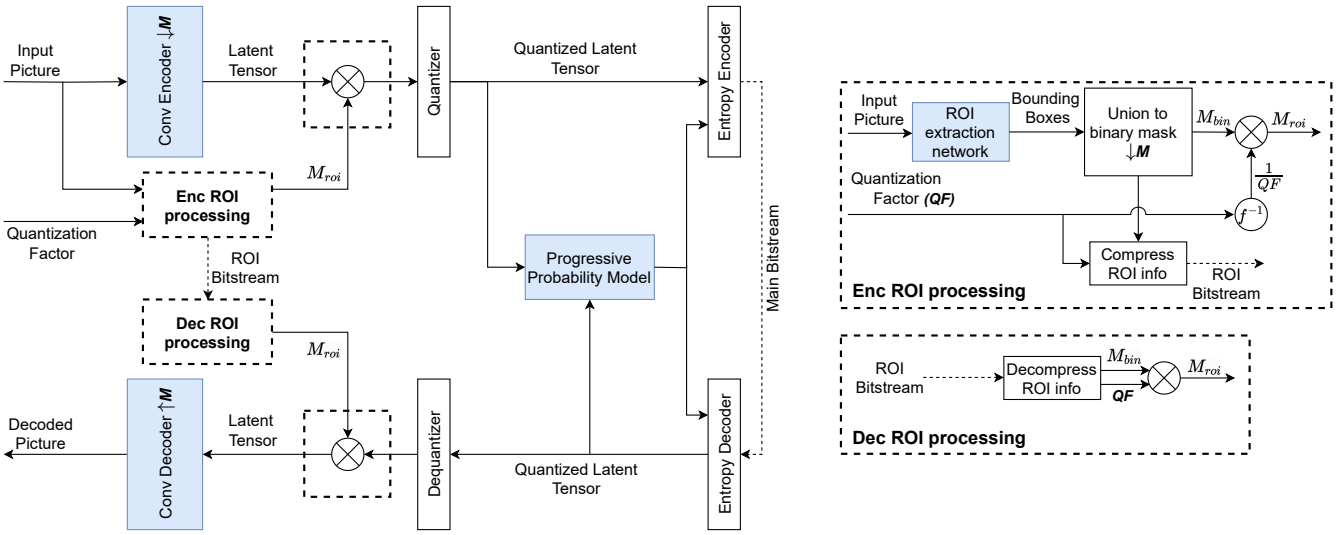


Fig. 1: Overview of the proposed ROI LIC method. Light blue boxes denote neural network-based components. Dashed boxes denote the components that are involved in ROI information processing.

method is applicable for any LIC, where a latent representation is used to represent the encoded input data.

Analysis and extra quantization Fig. 1 shows the overall structure of the ROI-LIC, where the dash-bordered boxes indicate the process involving ROI information manipulation. The process for learned image coding without ROI information can be described as follows: First, the input picture is fed to a convolutional neural network (CNN), which encodes the data into a downsampled latent representation. Next, the latent representation is quantized and losslessly encoded into a bitstream using an entropy encoder. On the decoder side, the bitstream is entropy-decoded and dequantized into the reconstructed latent representation. Finally, a decoded picture is generated from the reconstructed latent representation by running it through a decoder CNN. A progressive probability model such as [16] can be used to obtain the prior distributions that are used by the lossless entropy coding processes. In this example model, the distributions of the latent representation elements are inferred in a multi-scale, progressive manner that captures the context effectively while being parallelizable.

Unlike human-oriented pictures, the background regions of machine-targeted images have a much less impact on the performance of most task networks [8], [14], [15], [17]. Therefore, we propose using a simple quantization technique to further reduce the bitstream size without degrading the machine task performance. More specifically, as each element of the latent representation is approximated by a Gaussian-distributed continuous random variable, the variance of the element determines the number of bits required to encode the element.

Let $x \sim \mathcal{N}\left(0, \sigma^2\right)$, where σ^2 is the variance of the Gaussian distributed random variable x. It can be derived that the expected number of bits to encode x is

$$H(x) = \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}. \tag{1}$$

Thus, reducing the variance of the element efficiently reduces the size of the bitstream. In the proposed method, we simply divide the values of the latent representation by a quantization factor (QF), e.g., a number greater than 1, reducing the range of the variable, i.e. the variance of the Gaussian distribution function.

The ROI-LIC processing starts at the analysis stage, which contains the ROI extraction network. In principle, any network or method could be used for the analysis as long as it provides suitable ROI information. This network predicts the bounding boxes from the uncompressed images, which correspond to the areas that are the most important for the machine task. The confidence scores of the predicted bounding boxes are used to determine whether the box is included for further processing or not. The number and size of included bounding boxes affect the size of the ROI area.

After the analysis, the union of bounding boxes is used to generate a binary mask with the same size as the latent representation, in which the background area is presented as ones and the ROI area as zeros. This process is denoted as "Union to binary mask" box in the Fig. 1. It is to be noted that the union of the bounding boxes needs to be downsampled M times to the same spatial size as the latent representation:

$$M_{bin} = \downarrow_M \left(\bigcup_{i=0}^{N} B_i\right) \tag{2}$$

where M_{bin} is the generated binary mask, $\bigcup_{i=0}^{N}$ denotes the union of the N bounding boxes, B_i is the bounding box i, and $\downarrow_M$ denotes down-sampling by M times. M_{bin} is then multiplied by the inverse of the quantization factor, which is a floating point number. The output of the multiplication is an ROI mask, which determines the amount of extra quantization

for the background area and can be formulated as

$$M_{roi} = \begin{cases} M_{bin}\frac{1}{QF}, & when\ at\ Encoder \\ M_{bin}QF, & when\ at\ Decoder \end{cases}, \quad (3)$$

.

Finally, the latent representation is multiplied with the ROI mask M_{roi} and the output of the multiplication is fed to the quantizer, similarly as with a normal LIC. The final element-wise product can be formulated as:

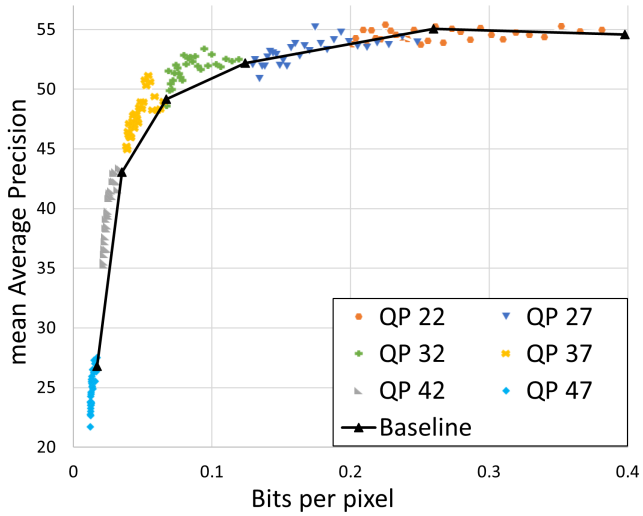
$$y_{roi} = y\delta(M_{roi}) + yM_{roi} \quad (4)$$

where y denotes the latent representation, M_{roi} denotes the ROI mask, and δ denotes Kronecker Delta function that indicates the elements of zeros in M_{roi}. Note that only the ROI mask values that are greater than 0, i.e. the background, are multiplied by the inverse of QF, in order to keep the ROI area of the latent representation unmodified. On the decoder-side, the corresponding latent representation is multiplied by decoder-side ROI mask, which is constructed from the ROI information signaled from the encoder side, and the value of QF instead of its inverse as at the encoder side, as shown in Eq. (3) and Fig. 1. In the next section, we will explain the mechanism of the signaling of the ROI information.

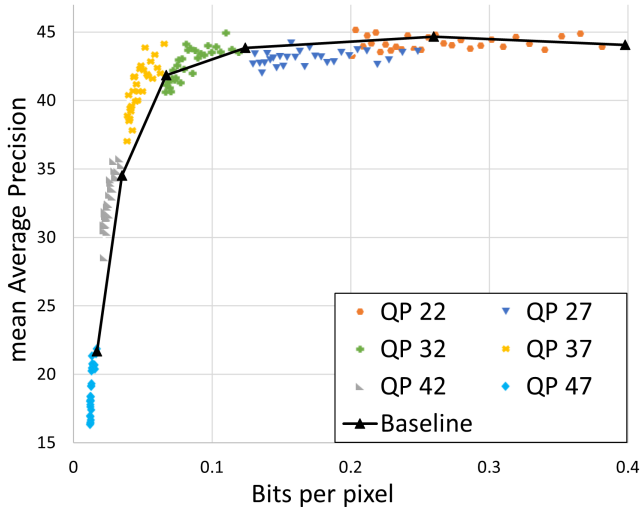
ROI information compression and signaling In Fig. 1 "Union to binary mask" component outputs the binary mask M_{bin}, which has the same spatial size as the latent representation as described in the previous section. In order for the decoder to use the required information, e.g., QF and M_{bin}, to generate the decoder side ROI mask, the binary mask M_{bin} is compressed and signaled to the decoder, which is represented by the "Compress ROI info" box in the same figure. First, the QF is linearly mapped into a Quantization Factor Index (QFI), which is an integer number with values in the range of 0 and 255. The QFI consumes one byte in the bitstream. The 2-dimensional binary mask M_{bin} is compressed by Run-Length-Encoding (RLE) method, which is efficient for data in binary format. Additionally, the length of the compressed ROI information is signaled to separate the image bitstream from the ROI bitstream at the decoder-side. Two bytes are used for the length information, resulting in a total signaling overhead of $3+N$ bytes, where N is the size of the RLE-encoded binary mask. At the decoder-side, the QFI is mapped back to QF and RLE-encoded binary mask is decoded to recover M_{bin}. The binary mask M_{bin} is then used to generate the decoder side ROI mask by following the process described in the earlier section.

III. Experiments

We take the machine-oriented LIC models from [18] as the baseline for our experiments. The baseline LIC contains 6 different pretrained models for 6 different rate points, corresponding to the rates of the VVC encoding using QPs [22, 27, 32, 37, 42, 47]. Similar to Fig. 1, the baseline includes a CNN-based encoder, a CNN-based decoder, a CNN-based progressive probability model and an entropy codec.



(a) Object detection



(b) Instance segmentation

Fig. 2: Rate-performance points using different QFI values and LIC models on the TVD image dataset on object detection and instance segmentation tasks, compared against the baseline RD curve. The used QFI values are in the range of [0, 30] for every LIC model.

To show the performance, the baseline and the proposed method are evaluated on the TVD image dataset [19] on object detection and instance segmentation machine tasks. The evaluation metrics are Bjontegaard Delta Bitrate (BD-rate) and BD-mAP scores [20]. Fasterrcnn_resnet50_fpn_v2 [21] from Torchvision [22] was used as the ROI-analysis network to predict the bounding boxes. A threshold value of 0.5 was chosen for the confidence scores to determine whether a bounding box is used to generate the ROI masks.

The range of the tested QFIs was empirically selected to be between [0, 30], which map linearly to QF values between [1, 3.8], as higher values caused the performance to drop drastically, especially for lower QP LIC models.

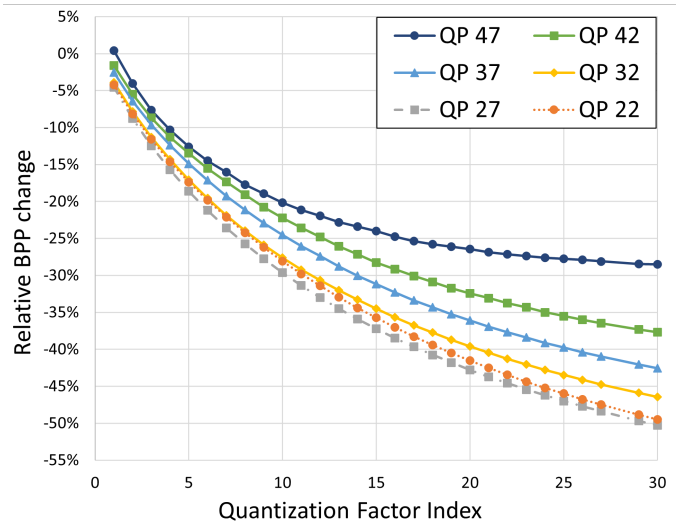


Fig. 3: Relative effect to the bitrate reduction with regard to quantization factor index.

Fig. 2 illustrates the coding performance data points with all the selected QFI values in the range of [0, 30] for the 6 LIC models, compared against the Rate-distortion (RD) curve of the baseline LIC, where the distortion is measured by the performance of the machine task. All the reported data points for the proposed method include the rate of ROI signaling. As the figure shows, the performance of the baseline LIC is preserved even with higher QFs, in particular, when the lower QP LIC models are used. Significant gains are observed over the LIC models, especially with the mid to high range of QP values. The same behavior is observed for both object detection and instance segmentation tasks. Fig. 3 further illustrates the relative BPP change with regard to the quantization factor. The figure shows that the relative bitrate savings are more significant with the lower QP LIC models, which can be explained by the signaling overhead described in Section II being proportionally larger with lower bitrate coding.

The results suggest that the used QFs should be higher with the lower QP LIC models, and vice versa. Based on these observations, a set of QF values were chosen for each LIC model, which provides good performance on both object detection and instance segmentation tasks. Table I describes the chosen quantization factor indices as well as the corresponding quantization factor values for each of the LIC models. Note that since the ROI signaling overhead for the baseline LIC model with the highest QP value is significant, we set the QFI to be 0, which turns off the ROI manipulation, generating the same results as the baseline model.

Using the selected QF values for the proposed method, we encoded the TVD image dataset with 6 bitrate points and calculated the BD-mAP score and the Pareto BD-rate gains against the baseline and the VVC. Table II shows the performance of the proposed method against the baseline and the VVC for the object detection and instance segmentation tasks. The Pareto BD-rate, instead of BD-rate, is used due to the non-monotonic RD-curve produced by the baseline, within which the BD-rate score is undefined. The proposed method achieves -15.80% and -22.43% Pareto BD-rate gains over the baseline on object detection and instance segmentation tasks, respectively. The corresponding BD-mAP gains are 1.85 and 2.71, respectively. Against the VVC, the proposed method achieves -41.52% and -50.38% Pareto BD-rate gains on the object detection and instance segmentation tasks, respectively, while the BD-mAP gains are 6.35 and 7.10, respectively. Fig. 4 further illustrates the rate-performance curves of the proposed method against the baseline and VVC. As shown in the figure, the effect on the bitrate reduction with the proposed method is significant while the machine task performance, measured by the mAP scores, is well preserved or improved. This suggests that in some cases, more distinct differences between the quality of the ROI-area and background area can be helpful to the machine task performance. The performance of VVC is also illustrated for reference, which is significantly lower than both the baseline and the proposed method.

Fig. 5 shows two examples of reconstructed images using QP 42 LIC model with a Quantization Factor of 1.63, as was selected according to Table I. The examples show that the background areas of the encoded pictures have lower visual qualities due to the information reduction caused by applying the quantization factor to those areas, while the foreground areas, i.e., detected ROI areas, such as persons in the images, have a higher visual quality.

The relative run-time of the proposed method is 40 % higher than the baseline, of which most of the extra processing time is spent in the ROI-analysis network.

TABLE I: Selected quantization factors for each LIC model

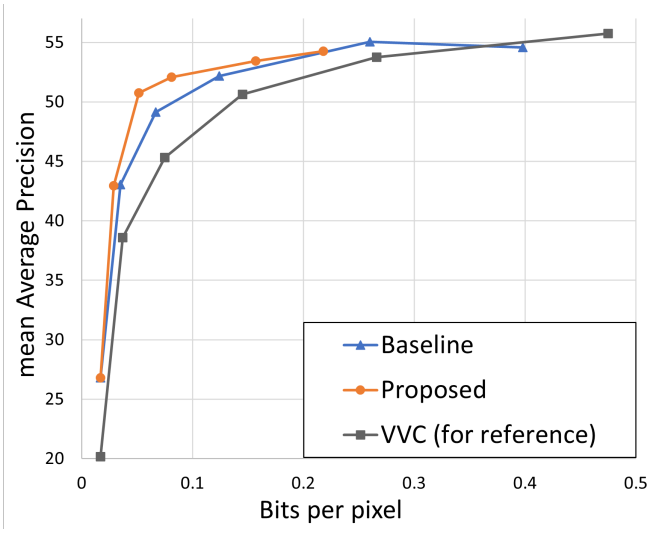
LIC model QP	QF Index	Corresponding QF
22	24	3.16
27	17	2.53
32	15	2.35
37	9	1.81
42	7	1.63
47	0	1.00

TABLE II: Average pareto BD-Rate (%) and BD-mAP gains of the proposed method over the baseline LIC and VVC on the TVD image dataset for object detection and instance segmentation tasks.

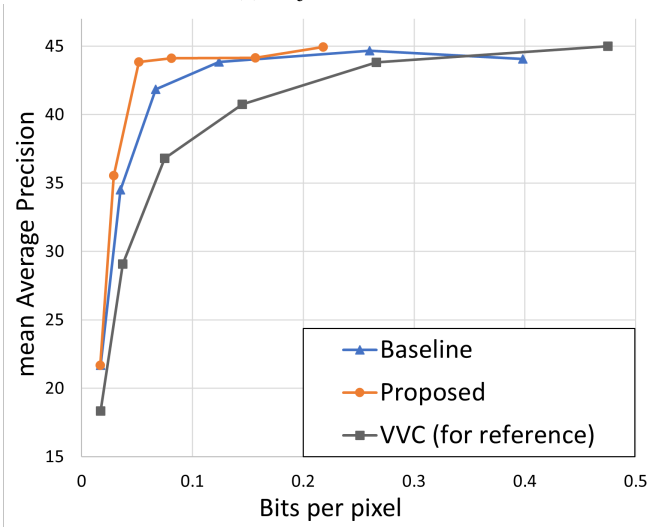
Task	**baseline LIC**		**VVC**	
	P-BD-Rate	BD-mAP	P-BD-rate	BD-mAP
Object detection	-15.80 %	1.85	-41.52 %	6.35
Instance segmentation	-22.43 %	2.71	-50.38 %	7.10

IV. Conclusion

In this paper, we proposed an efficient inference-time method to improve the performance of E2E-learned image codecs for machine consumption. The proposed method utilizes ROI information derived from an analysis network and applies more quantization to the background regions of the



(a) Object detection



(b) Instance segmentation

Fig. 4: Rate-performance curves of the proposed ROI LIC method with selected QP values, compared against the baseline LIC model and VVC on the TVD image dataset on object detection and instance segmentation tasks.

image to reduce the size of the bitstream. On the encoder side, a quantization factor is applied to the elements in the background regions in the latent representation before the quantization and entropy encoding. The quantization factor efficiently reduces the variance of the elements in the latent representation of the background regions, resulting in fewer bits used for the background regions. On the decoder side, the inverse of the quantization factor is applied to the corresponding elements to restore the elements in the latent representation to their original scale. We also proposed methods to effectively pack and compress the ROI information to be transferred to the decoder. The efficiency of the proposed method is demonstrated by the significant BD-rate gains over the baseline method, an E2E learned image codec for machines, when tested on the TVD image dataset on object detection and instance segmentation tasks.

References

[1] 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.

[2] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 10 771–10 780.

[3] 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.

[4] D. Wang, W. Yang, Y. Hu, and J. Liu, "Neural data-dependent transform for learned image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 379–17 388.

[5] M. Li, S. Gao, Y. Feng, Y. Shi, and J. Wang, "Content-oriented learned image compression," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX*. Springer, 2022, pp. 632–647.

[6] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[7] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, Aug 2021.

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

[9] H. Choi and I. V. Baji'c, "Scalable Video Coding for Humans and Machines," *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2022.

[10] 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.

[11] 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.

[12] J. Löhdefink, A. Bär, N. M. Schmidt, F. Hüger, P. Schlicht, and T. Fingscheidt, "Focussing learned image compression to semantic classes for v2x applications," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1641–1648.

[13] C.-H. Kao, Y.-C. Weng, Y.-H. Chen, W.-C. Chiu, and W.-H. Peng, "Transformer-based variable-rate image compression with region-of-interest control," *ArXiv*, vol. abs/2305.10807, 05 2023.

[14] C. Cai, L. Chen, X. Zhang, and Z. Gao, "End-to-end optimized roi image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 3442–3457, 2020.

[15] H. Akutsu and T. Naruko, "End-to-end learned roi image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[16] H. Zhang, F. Cricri, H. R. Tavakoli, E. Aksu, and M. M. Hannuksela, "Leveraging progressive model and overfitting for efficient learned image compression," *ArXiv*, vol. abs/2210.04112, 2022.

[17] N. Le, H. Zhang, F. Cricri, R. G. Youvalari, H. R. Tavakoli, E. Aksu, M. M. Hannuksela, and E. Rahtu, "Bridging the gap between image coding for machines and humans," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3411–3415.

[18] "Common test conditions for video coding for machines," *ISO/IEC JTC 1/SC 29/WG 04*, Jan 2023.

[19] W. Gao, X. Xu, M. Qin, and S. Liu, "An Open Dataset for Video Coding for Machines Standardization," in *2022 IEEE International Conference on Image Processing (ICIP)*, Oct. 2022, pp. 4008–4012.

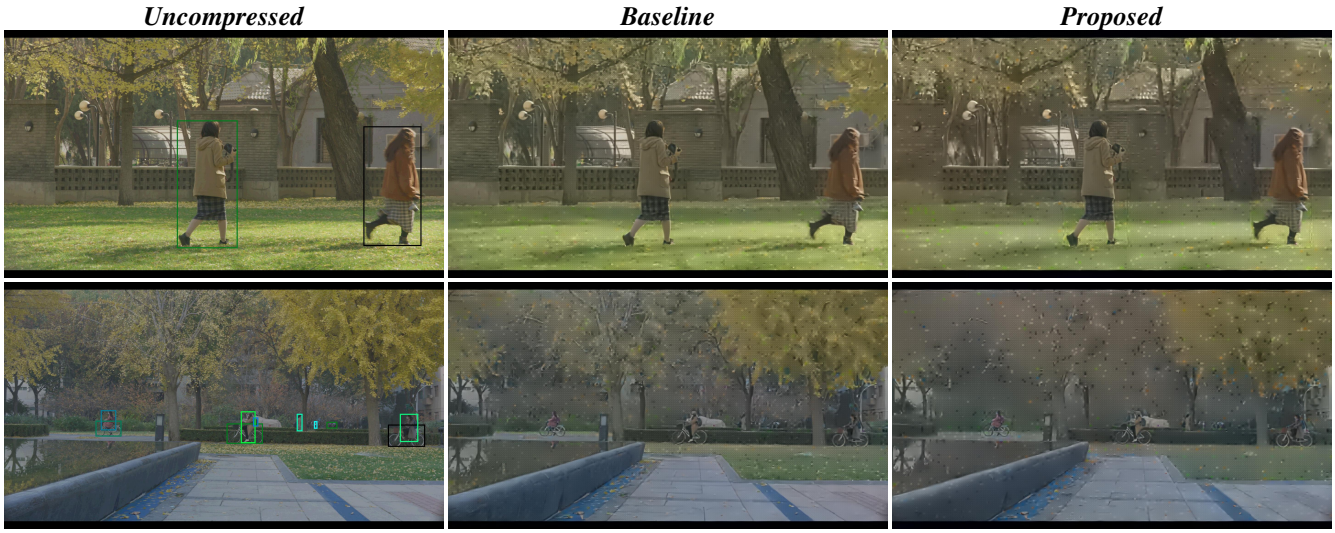


Fig. 5: Encoded images of the proposed ROI LIC method compared against the baseline LIC model on QP 42. Predicted bounding boxes used with the proposed method are shown on top of the uncompressed input images.

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

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2020, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[22] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM international conference on Multimedia*, ser. MM '10. Association for Computing Machinery, 2010, pp. 1485–1488.