A Two-stage H.264 based Video Compression Method for Automotive Cameras

Yiting Wang
WMG, University of Warwick
Coventry, UK
Yiting.Wang.1@warwick.ac.uk

Pak Hung Chan
WMG, University of Warwick
Coventry, UK
pak.chan.1@warwick.ac.uk

Valentina Donzella
WMG, University of Warwick
Coventry, UK
v.donzella@warwick.ac.uk

***Abstract*—With the development of automated vehicles and advanced driver assistance systems, the compression of the large amount of data generated by the vehicle camera sensors becomes a necessary processing step to improve the automated driving system efficiency. H.264 is a widely adopted video compression scheme, and it has been designed for human vision. Rate control in H.264 uses fixed quantisation parameter, however, this process can lead to fluctuation in different regions of the image quality of each frame. In this paper, we propose a two-stage H.264 based video compression framework, named "Two Stage Compression (TSC)", to compress the automotive camera videos with different values of compression rate in different regions of each frame. In the first stage, each frame will be divided into the region-of-interest and the region-out-of-interest. In the second stage, different compression ratios will be applied based on the importance of the region. The experimental results show that under the same overall compression ratio, our proposed TSC increments the semantic-aware PSNR by 3.213 dB compared to uniform H.264 compression. Our method is also compared to uniform H.264 compression using a segmentation algorithm, with an improvement of 1.77% in mIOU, the average Intersection over Union.**

***Index Terms*—Automotive camera data, video compression, semantic segmentation, machine learning, Intelligent Vehicles, Automated and Assisted driving.**

I. Introduction

AUTOMATED vehicles (AVs) and advanced driver assistant systems (ADASs) can bring vast benefits to the society including improved safety, more flexible mobility, reduced emissions and traffic jams [1]. The perception sensor suite is vital for aiding AV decision making process in all levels of SAE classification of automation [2]. The increased automation results in the remarkable increment of data amount generated by the sensors, including video data produced by the widely adopted camera sensors. However, current and near-future vehicles are limited in computational resources and bandwidth. Besides, a high volume of data transmission can lead to unacceptable latency when facing a bandwidth bottleneck, and this issue may lead to safety problems. To support the real-time transmission of the large amount of sensor data, video compression needs to be evaluated. While a higher compression ratio may reduce data volume and transmission latency, it will result in poor video quality, distortions and artefacts that might compromise the accuracy of the perception step and therefore safety in AV systems. Commonly used methods such as H.264 [3] can reduce the spatial and temporal redundancy within a single frame or across multiple frames by finding the balance between quality and compression ratio during video compression. However, this standard was designed for human vision, and the implications on machine learning (ML) based perception have been only recently started to be discussed and analysed (e.g. retraining deep neural networks with compressed data) [4], [5]. Therefore it is clear that more work needs to be done to exhaustively investigate the relationship between video compression and ML based perception for automated driving.

In this paper, we propose a novel two-stage based H.264 method, entitled TSC, to compress the automotive camera video frames to achieve an increased compression ratio without compromising the quality of the perception step. To achieve this semantic video compression, we propose to use higher compression ratios for regions of the video frames containing less important information, and lower compression ratios for key areas. We first use a semantic segmentation neural network to process the original video stream into regions of interest (ROI) and regions outside the ROI (non-ROI). This segmentation scheme can be low-quality (as long as the ROI is identified properly), but it needs to be lightweight so it can be implemented with no latency on sensors. More specifically, in the context of AVs, we consider the categories of object, human, vehicle, unlabeled, dynamic, ground, road, parking to be the ROI and the categories of construction, nature, sky to be non-ROI. The different compression ratios are achieved by changing the constant rate factor using a H.264 compliant codec. The experimental results show that our TSC technique outperforms uniform compression in terms of our proposed semantic-aware Structural Similarity Index (SA-SSIM) and Peak Signal to Noise Ratio (SA-PSNR).

II. Related Work

A. H.264 Standards

The international organization ISO/IEC groups named Moving Picture Experts Group (MPEG) and International Telecom Union (ITU) have developed several standards to regulate video coding techniques [6]. For example, the H.26X family has been developed by ITU since 1991 [7]. At present, some of the most widely used video coding techniques are based on H.264 standard [7]. This standard considers both spatial and temporal redundancy to reduce the bit rate. A group of pictures

(GOP) in H.264 contains three different types of frames: the intra-coded frames (I-frames), the predicted frames (P-frames), and the bidirectionally predicted frames (B-frames). The I-frames indicate that all of the macroblocks in the frames are coded by intra-prediction while the others mainly use inter-frame prediction. Accordingly, the P-frames are generated referencing historical frames by motion estimation and motion compensation, B-frames are generated using bidirectional referencing to I and P frames [8] . The predictive coding architecture used by H.264 consists of motion estimation, block-based motion compensation, transform, quantisation, inverse transform, entropy coding and frame reconstruction. These conventional video compression method has the advantages of maturity and wide adoption in numerous fields. However, without *ad hoc* strategies when using this standard for different applications, its efficiency is not optimised to be used in AV and ADAS video streaming. In our paper, we combine the well established compression techniques based on H.264 with a pre-segmentation to overcome the limitation of current compression techniques.

B. Rate control in H.264

Video encoding can produce a variable compressed bit stream due to entropy coding properties and variability in the information content of each frame [9]. However, that would cause a variable bit rate under constrained bandwidth which might cause problems in the systems [10]. On the one hand, if the encoded video bit rate is larger than the bandwidth, the channel will become congested, potentially causing data loss and therefore problems in the quality of the reconstructed video. One solution to this problem is to use rate control compression, which can fix the video bit rate while producing the best quality within the limited bandwidth. Various rate-control algorithms have been proposed through the history of video encoding to maintain the data size within the bandwidth limit. The constant rate factor (CRF) is the default quality setting for the X264 codec [11], the value of CRF can vary from 0 to 51. It has been demonstrated that the higher is CRF, the higher is the compression rate and therefore the worst is the decoded video quality [12]. Rate control might be a solution for AV and ADAS applications to have the best data quality under a unified constrained data flow.

C. Region-based video compression

Content-aware based compression means to compress the original video frames based on their content; our proposed method belongs to this category of compression techniques. Several content-aware compression techniques use video segmentation to separate the foreground objects from the background. One recent work has proposed a content-adaptive video compression for AVs remote control application [13]. It uses a simulated dataset for the ROI and non-ROI compression by varying the quantisation parameter. However, the realism and reliability of the synthetic dataset have not been discussed. Moreover, this work only emphasises the object detection results after compression and focuses on only one particular class (the traffic lights) as the target object. However, it is important to quantify the quality of the compressed video data and how it will influence the perception algorithms. Different from the existing methods, we use a new machine learning-based segmentation network to assess the quality of our proposed method. Semantic segmentation is an important task in computer vision that can be used in many applications, and it is based on assigning to each pixel in a frame a specific class [14]. Most of the existing works evaluate the influence on quality by considering the implications on object detection, without considering semantic segmentation [4], [5]. Therefore, we also evaluate the performance of the semantic segmentation on the compressed data with more than 20 classes and 8 categories. Building on previous works, in this paper we use attention-based semantic segmentation for ROI extraction on the Cityscape dataset; furthermore, the reconstructed videos are re-segmented and compared to the original data.

III. Methodology

Our methodology aims to study the performance of the two-stage H.264 compression scheme on the selected dataset and study the effect on semantic segmentation. The original video sequence can be expressed as I = $\{x_1, x_2, ..., x_t\}$, where x_t means the frame at time t. After the compression, the reconstructed video $\hat{I}$ can be obtained. A schematic view of the applied methodology is presented in Fig.1, and its detailed description is provided in the following subsections. This structure aims to provide a solution for AV data compression and to mimic the AV architecture where the sensor data may be transmitted over wired vehicle communication networks and fed to a semantic segmentation algorithm in a central processing unit.

A. Masks Generation

In Fig.1 the blue module is responsible for generating on the sensor chip the semantic segmentation masks for the ROI and non-ROI. This step is implemented through a designed segmentation network ϕ with the Residual Net (ResNet) as backbone for feature extraction and the attention-based criss-cross attention modules (RCCA) to obtain better segmentation efficiency [15], [16]. The output is $\phi(I)$. The categories of the segmentation pixels will firstly be represented with four shades of grey, which represent nature, construction, sky and ROI. After that, the frames will go through the thresholding operation; this step will decrease the number of the segmentation categories to two: non-ROI (i.e. nature, construction and sky) and ROI (i.e. vehicles, traffic signs, pedestrians and road). Since the following steps will involve using X264 codec, a 16×16 macroblock filter is used to simulate the macroblocks used in the standard for motion compensation and prediction. During filtering, any 16×16 pixel block that contains important area pixels will be regarded as an ROI block. This can reduce the risk of damaging the integrity of important targets. Once we got the masks for

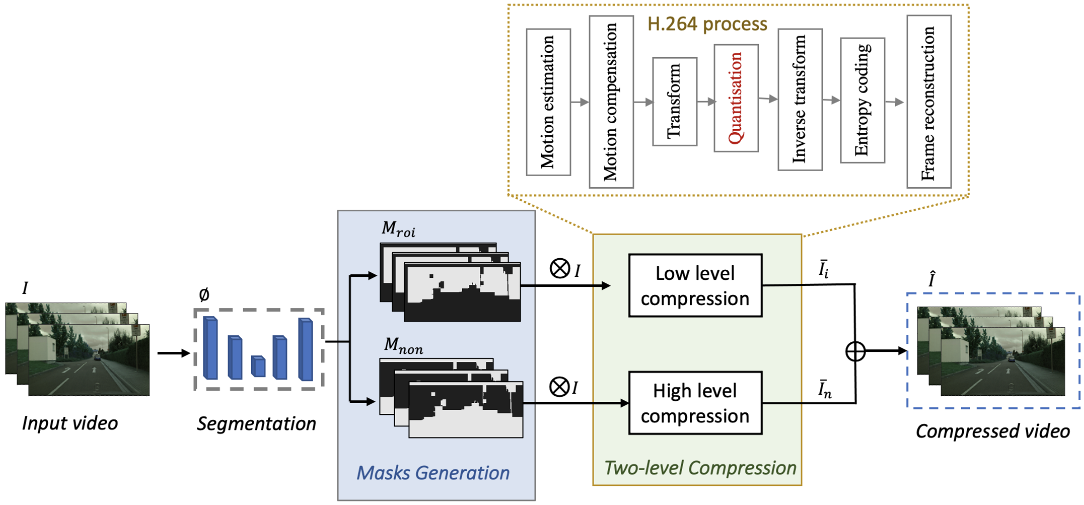


Fig. 1. Illustration of our video compression architecture. The input video firstly goes through the segmentation block, then the masks generation (blue block) and the two-level compression with H.264 (green block) is processed.

ROI (M_{roi}) and non-ROI (M_{non}), the two streams (S_{roi}) and (S_{non}) can be generated with the following functions:

$$S_{roi} = M_{roi} \otimes I \tag{1}$$

$$S_{non} = M_{non} \otimes I \tag{2}$$

The " $\otimes$ " denotes a pixel by pixel multiplication.

B. Two-level compression-decompression

After the last step above, two streams of ROI (S_{roi}) and non-ROI (S_{non}) frames will be generated. We set n_t and i_t to be the frames at the time t from the video sequence S_{non} and S_{roi} separately. They will need to meet the following criteria:

$$n_t \subset S_{non} \ , \ i_t \subset S_{roi} \tag{3}$$

$$x_t = n_t \oplus i_t \tag{4}$$

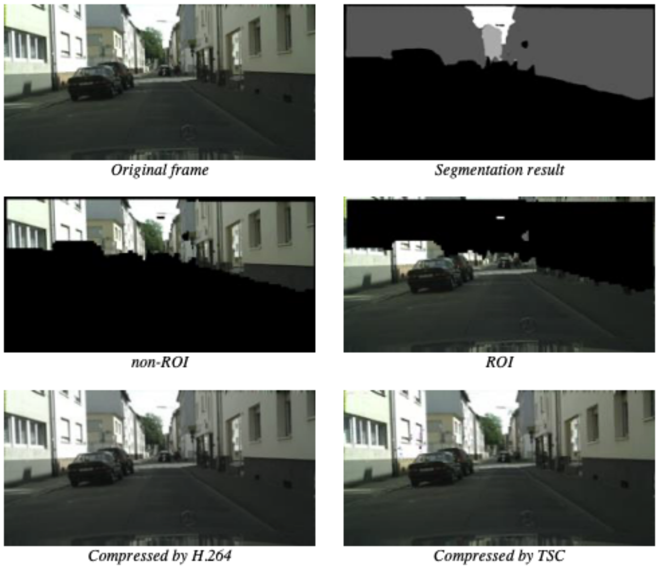
The process of the compression techniques used in our two level scheme can be divided into seven phases (P1 - P7), as can be seen in the green box, Fig. 1. These phases are: P1, motion estimation; P2, motion compensation; P3, transform; P4, quantisation; P5, inverse transform; P6, entropy coding; P7, frame reconstruction; for a more detailed description refer to [8]. These phases are applied to the ROI and non-ROI streams separately. The ROI area frame $\bar{i}_t$ can be reconstructed through the predicted frame $\hat{i}_t$ and the reconstructed residual $\bar{r}_t$ by the following equation:

$$\bar{i}_t = \hat{i}_t + \bar{r}_t \tag{5}$$

The other non-ROI area frames n_t can be reconstructed in a similar way, i.e. we set the reconstructed ROI area frame in time t as $\bar{n}_t$. Finally, the reconstructed whole frame $\bar{x}_t$ with two-level compression can be generated through the following equation:

$$\bar{x}_t = \bar{i}_t + \bar{n}_t \tag{6}$$

In the following time step $(t+1)$, the reconstructed residual $\bar{r}_t$ will be stored in the decoded frame buffer and used in the following iteration of the process. Although the above pipeline is similar to all encoder-decoder video compression methods, the main difference resides in the quantisation process (P4). In the process of quantisation, the constant rate factor parameter can be adjusted to achieve different quality levels. The higher the CRF value, the worse the video quality. The CRF parameter here is adjustable by different settings where the stream S_i CRF is smaller than the stream S_n CRF. The CRF rate control parameter is the key in our proposed TSC method.



Original frame *Segmentation result*

non-ROI *ROI*

Compressed by H.264 *Compressed by TSC*

Fig. 2. Video Compression Visual Result. From top to down, left to right: The original frames, the semantic segmentation results, the non-ROI, the ROI, the reconstructed frames by H.264 and TSC.

IV. Implementation

Our experiment used an Ubuntu 20 virtual machine with 100 GB space, Quadro P5000 GPU and a Conda environment with Python 3.8 for the libraries. The open-source PyTorch library was used for network training and testing.

1) Dataset: the Cityscape dataset was used for the experiment [17]. It is an AVs open benchmark dataset that provides semantic annotations for 30 classes with 8 categories (e.g.: humans, vehicles, constructions, nature, sky, etc). Data are captured in 50 cities during several months (spring, summer, fall), daytime, and good weather conditions. The dataset consists of 50 sequences of videos, with 5000 fine annotated images. The proportion of frames used for training, validation and testing is approximately 60%, 10% and 30%, as suggested on the dataset official website. The frames are 2048 × 1024 × 3 RGB images, but we downsized them into 1024 × 512 × 3 for ground truth and 1024 × 512 × 1 for labels to reduce computational costs.

2) Loss function. The accuracy of the deep learning segmentation not only relies on the network architecture but also on the choice of the loss function. When there are less common classes in the segmentation task, the imbalance of the classes may result in sub-optimal performance. The dice loss is useful for highlighting unbalanced segmentation [18]. However, the dice loss function is not useful when there are small-sized targets, therefore, adding the binary cross-entropy (BCE) loss function can address this aspect [19]. Here we designed the loss function to be the combination of the BCE [19] and dice loss [18], thereafter named the "BCE-Dice loss".

This novel loss function used in our experiments can be defined as below.

$$L_{BCE-Dice} = L_{BCE}\left(s, \hat{s}\right) + \ L_{Dice}\left(s, \hat{p}\right) \qquad (7)$$

3) Evaluation of segmentation. The evaluation metric for ROI-segmentation will be the accuracy of the "Intersection-over-Union" (IOU). If the ground truth area is A and the predicted area is B, the intersection region in the equation is $A \cap B$, the union is $A \cup B$. The IOU can be then expressed as:

$$IOU = \frac{A \cap B}{A \cup B} \qquad (8)$$

Assuming to have m categories of labels, the average IOU (mIOU) in the whole frame can be calculated as:

$$mIOU = \frac{\sum_{1}^{m} IOU_m}{m} \qquad (9)$$

Since in our task we are only interested in differentiating the non-ROI from the ROI, the concept of "Region-of-interets IOU" (iIOU) is proposed as described below.

$$iIOU = \frac{A_{roi} \cap B_{roi}}{A_{roi} \cup B_{roi}} \qquad (10)$$

4) Evaluation for Compression. FFmpeg was used for compression since it is a free and open-source environment consisting of a suite of libraries for compression. We used X264 codecs for our experiments. We used I_{crf} as the CRF value for stream I and N_{crf} as the CRF value for stream N. The used CRF values are summarised in table I. In this paper, the widely used peak signal-to-noise ratio (PSNR) [20] and SSIM [21] are also used to measure compression distortion. Besides, the compression ratio is also considered.

TABLE I
CRF values used in the experiments

Method	CRF default settings
H.264	$\mathrm{I}_{crf} = N_{crf} = a(23)$
TSC	$\mathrm{I}_{crf} = a - 5(18), N_{crf} = a + 5(28)$

Note that TSC is the proposed two-stage compression based on the X264 codec

V. Results

A. Compression and artefacts

After 47 epochs with 2974 iterations in each epoch, the average ROI-segmentation result for various categories reaches 87.97% in training and 80.89%, 79.08% for validation and testing respectively. However, the iIOU remains high, all iIOU results are above 90% for the binary masks. The average time of segmentation each frame is 0.081 s.

The experimental results of one selected frame (at t =207s) are shown in Fig. 2. The second row represents the used masks; we can observe that the mask boundaries are not smooth since the macro-block filter has been applied. In the last row, the reconstructed frames are shown with the parameter settings shown in Table I. The two-stage compression shows comparable visual performance as the H.264 under the default settings. However, by visual inspection of the reconstructed frames, we noticed that some artefacts can appear at the boundaries. As an example, one generated artefact has been highlighted in the red rectangle in Fig. 3, with purple and dark blue coloured pixels appearing in the output frames.

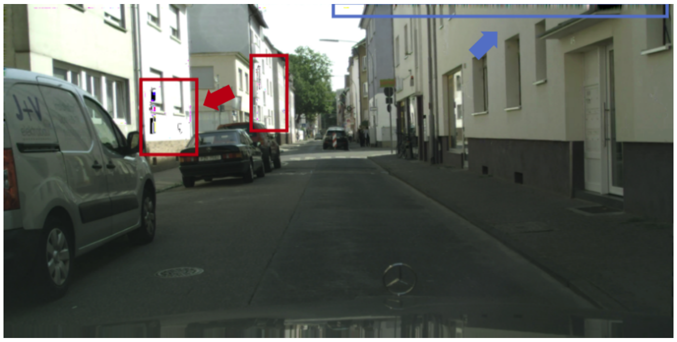


Fig. 3. The flash artefact phenomenon at t=207.

B. Image quality evaluation

In Fig.4 we compare PSNR and SSIM after compressing with X264 the original frames (X264-O), the ROI stream (X264-I) and non-ROI stream (X264-N). The evaluated video quality of the original frames has lower SSIM and PSNR than the separated two streams I and N under all compression methods and all CRF settings. This might be due to accurate

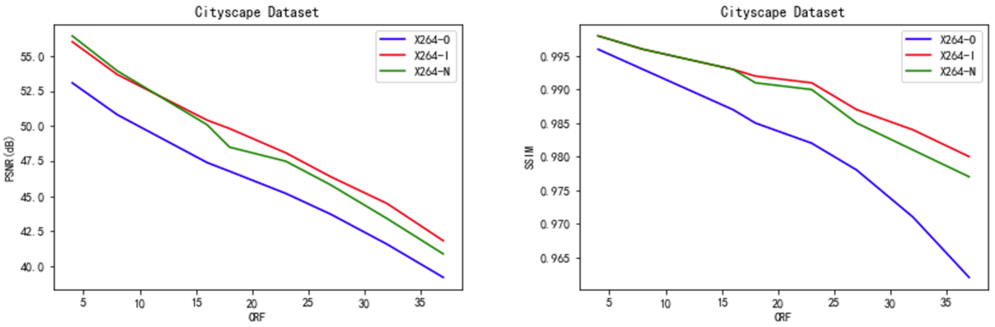


Fig. 4. The comparison results with different CRF settings on different frame areas. From left to right, the PSNR results and the SSIM results.

segmentation mask generation and the simplicity of the separated frames compared with the original whole frame. Higher PSNR or SSIM values mean better compression quality. There is a clear decreasing trend as the value of CRF increases for all the compared methods. Specifically, for smaller CRF values (CRF $<$12), the region of non-ROI has higher PSNR, while for higher CRF value (CRF $>$12), the ROI has higher PSNR. The result show a similar trend for the PSNR values. These results can be related to the different proportions of the ROI and non-ROI area in the dataset frames. Although the average proportion of the ROI is higher (55%) than non-ROI (45%), the proportion varies from frame to frame.

C. Semantic-aware Evaluation

Since our proposed method is based on two streams with different compression ratios, inspired by [22], we thereby propose a novel fairer way of evaluating it. The new evaluation metrics respectively called Semantic-aware SSIM (SA-SSIM) and Semantic-aware PSNR (SA-PSNR) are used to evaluate our results. These new metrics take both compression ratio and performance into account as the weights of the two regions. We define the (I_{crf}, N_{crf}), (S_i, S_n) and (P_i, P_n) to be the CRF, SSIM and PSNR values for stream I and N separately. The compression ratio index which indicates the relationships between the streams I and N compression ratios (r_{roi}, r_{non}) can be calculated as the following equations:

$$r_{roi} = N_{crf} \div (N_{crf} + I_{crf}) \tag{11}$$

$$r_{non} = I_{crf} \div (N_{crf} + I_{crf}) \tag{12}$$

The SA-PSNR and SA-SSIM can be calculated as the following equations.

$$\text{SA-SSIM} = r_{roi}S_i + r_{non}S_n \tag{13}$$

$$\text{SA-PSNR} = r_{roi}P_i + r_{non}P_n \tag{14}$$

The performance of our proposed method are compared with the compressed video quality under a uniform compression ratio. The experiment is to use different CRFs to keep the ROI with higher quality and the non-ROI with lower quality, which is different from the settings of the conventional uniform method where the same CRF is used through the whole frame. We started setting the CRF value of the H.264 to be a =23 (i.e. this value is often used as the default value), and selected our TSC CRFs as shown in Table I to achieve the same overall compression ratio with the two methods. The quality results using the newly described indicators are summarised in TableII.

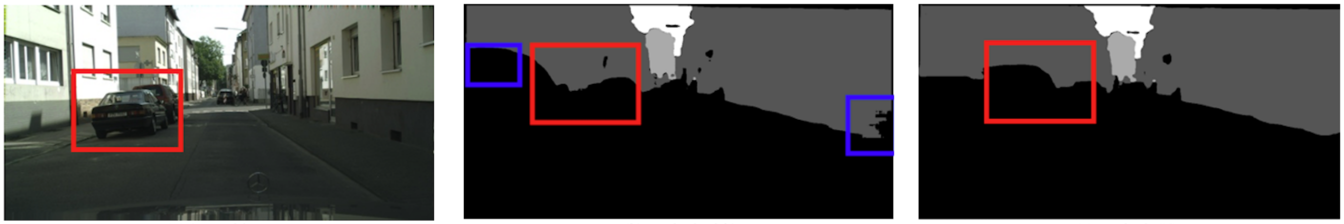
TABLE II
COMPRESSION AND POST-SEGMENTATION RESULTS USING TRADITIONAL UNIFORM VS OUR TWO STAGE COMPRESSION MAINTAINING OVERALL SAME COMPRESSION RATIO.

Method	SA-PSNR	SA-SSIM	mIOU	iIOU
H.264	45.181	0.982	85.71%	91.43%
TSC	47.762	0.991	87.48%	92.04%

From Table II, we can see that our TSC outperforms H.264 in terms of both the SA-SSIM and SA-PSNR. The calculated value is more than 2 dB higher than H.264 for SA-PSNR. This is similar to the trend in Fig. 4. After compression, we are also interested in evaluating its implications on the perception tasks. Here, we use our proposed semantic segmentation model to test the outcome of the videos compressed by the two investigated methods. The performance of segmentation is evaluated using mIOU and iIOU, see section IV, and the results are presented in Table II. The results show that techniques with a higher SA-PSNR and SA-SSIM achieve better segmentation performance in terms of both mIOU and iIOU. This quantitative analysis is in accordance with the visual results after segmentation (See Fig. 5). Our proposed TSC techniques result in better segmentation mIOU with respect to traditional compression based on uniform compression of each frame. This might be due to a better compression quality of the ROI. As can be seen from the figure, traditional H.264 based segmentation presents both false negatives (shown in red box) and false positives (shown in the blue box).

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new video compression method that uses semantic segmentation to extract region-of-interest and region-out-of-interest for each video frame and then applies different compression ratios to the two regions. Besides, we proposed three new quality evaluation metrics (i.e. SA-SSIM, SA-PSNR, iIOU) that also consider the division of the frames into different areas. Experimental results show that our



(a) Original frame

(b) Result of H.264

(c) Result of TSC

Fig. 5. The visual segmentation results on the original frame and the compressed frame under same compression ratio.

method outperforms traditional H.264 compression in terms of SA-SSIM and SA-PSNR. In the future, better and lightweight segmentation algorithms, other compression parameters and more variations can be investigated to optimised our proposed technique in terms of performance and speed. More advanced algorithms such as H.265 [23] or other machine learning-based compression techniques can also be implemented into our framework. Our research shows that optimising compression techniques in combination with high-level perception tasks, such as semantic segmentation and object detection, may be a new and promising exploration direction in the future for addressing the sensor data conundrum in automotive. Further investigation is needed for the compression artefacts, e.g. the "flash phenomenon" in Fig. 3. However, from the overall results, these small artefacts do not to have a detrimental effect on the perception task. We believe that our proposed method can inspire other researchers to propose new algorithms for other applications achieving superior compression performance with respect to traditional techniques.

Acknowledgment

Dr Donzella acknowledges that this work was supported by the Royal Academy of Engineering under the Industrial Fellowships scheme. The Authors wish to acknowledge the support of HVM CATAPULT. The Authors thank Dr Anthony Huggett, ON Semiconductor, for the fruitful discussions.

References

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58443–58469, 2020.

[2] SAE, "J3016b: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." Accessed 24 Dec 2021. https://www.sae.org/standards/content/j3016$_2$01806/, 2018.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[4] W. Xu, N. Souly, and P. P. Brahma, "Reliability of gan generated data to train and validate perception systems for autonomous vehicles.," in *WACV (Workshops)*, pp. 171–180, 2021.

[5] P. H. Chan, G. Souvalioti, A. Huggett, G. Kirsch, and V. Donzella, "The data conundrum: compression of automotive imaging data and deep neural network based perception," in *London Imaging Meeting*, vol. 2021, pp. 78–82, Society for Imaging Science and Technology, 2021.

[6] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*. Springer Science & Business Media, 1996.

[7] F. S. Mahammad and V. M. Viswanatham, "A study on h. 26x family of video streaming compression techniques," *International Journal of Pure and Applied Mathematics*, vol. 117, no. 10, pp. 63–66, 2017.

[8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[9] M.-T. Sun, *Compressed video over networks*. CRC Press, 2000.

[10] Z. Chen and K. N. Ngan, "Recent advances in rate control for video coding," *Signal Processing: Image Communication*, vol. 22, no. 1, pp. 19–38, 2007.

[11] W. Robitza, "Crf guide (constant rate factor in x264, x265 and libvpx)," 2017.

[12] J. Bienik, M. Uhrina, and P. Kortis, "Impact of constant rate factor on objective video quality assessment," *Advances in Electrical and Electronic Engineering*, vol. 15, no. 4, pp. 673–682, 2017.

[13] I. Dror, R. Birman, and O. Hadar, "Content adaptive video compression for autonomous vehicle remote driving," in *Applications of Digital Image Processing XLIV*, vol. 11842, p. 118420Q, International Society for Optics and Photonics, 2021.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[16] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 603–612, 2019.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[18] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571, IEEE, 2016.

[19] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[20] T. K. Tan, R. Weerakkody, M. Mrak, N. Ramzan, V. Baroncini, J.-R. Ohm, and G. J. Sullivan, "Video quality evaluation methodology and verification testing of hevc compression performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 76–90, 2015.

[21] 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.

[22] L. Wu, K. Huang, H. Shen, and L. Gao, "Foreground-background parallel compression with residual encoding for surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[23] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of h. 265/mpeg-hevc, vp9, and h. 264/mpeg-avc encoders," in *2013 Picture Coding Symposium (PCS)*, pp. 394–397, IEEE, 2013.