Influence of AVC and HEVC compression on detection of vehicles through Faster R-CNN

Pak Hung Chan, Anthony Huggett, Georgina Souvalioti, Paul Jennings, and Valentina Donzella

Abstract— **Situational awareness based on the data collected by the vehicle perception sensors (i.e. LiDAR, RADAR, camera and ultrasonic sensors) is key for achieving assisted and automated driving functions in a safe and reliable way. However, the data rates generated by the sensor suite are difficult to support over traditional wired data communication networks on the vehicle, hence there is an interest in techniques that reduce the amount of sensor data to be transmitted without losing key information or introducing unacceptable delays. These techniques must be analysed in combination with the consumer of the data, which will most likely be a machine learning algorithm based on deep neural networks (DNNs). In this paper we demonstrate that by compression tuning the DNNs (i.e. transfer learning by re-training with compressed data) the DNN average precision and recall can significantly improve when uncompressed and compressed data are transmitted. This improvement is achieved independently from the compression standard used for compression-training (i.e. AVC and HEVC), and also when training and transmitted data use the same compression standard or different compression standards. Furthermore, the performance of the DNNs is stable when transmitting data with increasing lossy compression rate, up to a compression ratio of approximately 160:1; above this value the performance starts to degrade. This work paves the way for the use of compressed sensor data in automated driving in combination with the optimisation of compression-tuned DNNs.**

Index Terms— **Compression, Perception Sensor, Camera, Deep Neural Network, Transfer learning, Intelligent Vehicles, ADAS**

II. Introduction

As passenger and commercial vehicles are swiftly transitioning to offer more assisted and automated driving functions, the role of perception is becoming increasingly significant. *Perception* in advanced driving assistance systems (ADASs) and automated vehicles (AVs) is the ability of the vehicle to 'understand' and localise itself and the other road stakeholders in the surrounding environment, and it is key for *planning* and *navigation*. In SAE J3016 this task is defined as 'monitoring the environment' and from L3 the vehicle is fully responsible for it [1]. In order to accomplish this task, multiple environmental perception sensors are deployed such as cameras, LiDAR, RADAR and ultrasonic sensors. Cameras are considered fundamental to achieve some ADAS and AV functions (e.g. object detection, classification, sign recognition, lane centring, etc.) and can leverage mature computer vision and machine learning algorithms to process their data [2-3].

Data are collected by automotive cameras in the form of a video stream, and then transmitted to one or more processing units for consumption via the wired data communication networks available on the vehicle. However, in achieving this data transmission, several challenges need to be confronted. First of all, cameras produce a considerable amount of data per second, e.g. a high-dynamic-range, HDR, 8 Mpixel 30 frames per second camera generates around 3.84 Gb/s (considering a pixel depth of at least 16 bits) of Bayer data. If we consider multiple cameras to achieve a 360^o coverage of the vehicle surrounding, and also that the sensor suite would include multiple LiDARs and RADARs, the amount of generated sensor data can be in excess of 40Gb/s, and therefore not supportable by current automotive wired communication networks [4–6]. Due to the placement of sensors around the vehicle and wire harnessing, some captured sensor data will have to be transmitted over wired networks with lengths in excess of 5 meters to reach a processing node. Only dedicated automotive networks fulfil the transmission requirements in terms of safety, latency, etc., but as mentioned above they cannot support the expected data rates of perception sensors' suites made up of up to 40 sensors [4]. Currently, camera video data used for ADAS functions (e.g. lane centring, sign recognition, park assist, etc.) are transmitted uncompressed by using dedicated copper based cables and connectors which are expensive and heavy [7]. Therefore, the need for considering and evaluating data reduction and compression techniques arises in automotive; however a careful analysis of potential loss of key information, artefacts, bit error propagation, combined with any delays introduced by the coding-decoding process, is needed [8-9].

Research is focusing on video compression for AV and ADAS applications, as many requirements need to be satisfied in order to ensure the safety of the passengers in the vehicle and of the nearby road stakeholders. Amongst these requirements, it is worth mentioning real-time data processing and low latency, that are key ones for any decision-making process deployed in the vehicle processing units [10-11]. Another requirement of compression on the sensor chip is to develop compression techniques able to comply with the restricted on-sensor resources. The final requirement is that the use of compression must not degrade the performance of the consumer of the video stream. Lossy video compression may produce

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 onsemi and High Value Manufacturing CATAPULT.

P H Chan, G Souvalioti, Prof. P Jennings and Dr V Donzella are with WMG, University of Warwick, Coventry, CV4 7AL, UK (e-mail: pak.chan.1@warwick.ac.uk, v.donzella@warwick.ac.uk)

Dr A Huggett is with onsemi, Greenwood House, Bracknell, RG12 2AA, UK (e-mail: anthony.huggett@onsemi.com).

differences between the input and decoded streams. These are referred to as artefacts and become more noticeable as the compression ratio increases. Other problems (not addressed here) are the effect of bit errors which may occur as the compressed data is transmitted across the communication data network. In an uncompressed scheme, the effect of such errors will be highly localised, but in a compressed scheme the effect of a single error might propagate, resulting in several frames with corruption over a large area, or even total loss of the picture stream for several frames. Error correction techniques such as Forward Error Correction or Automatic Retransmission can, if properly designed, reduce the probability of such an event to an acceptable level. Furthermore, measures can be taken to improve the design of video compression standards to make them intrinsically more resilient to errors, limiting error propagation [12]. In this paper, we focus on considering the effects of artefacts when machine learning algorithms are the users of the data. Currently, there is a trend in automotive to use machine learning (ML) techniques and deep neural networks (DNNs) for implementing some key data processing functions (e.g. object detection, tracking, prediction, classification, etc.), and we expect them to flourish in the near future [13-14].

A. *Contributions*

This work investigates the use of compression techniques to transmit automotive video camera data via traditional wired communication networks from the sensor to one of the vehicle processing units, as shown in Fig.1 a). Different ratios of video compression are implemented using well-established and mature compression techniques (i.e. AVC, HEVC, Sec. II. A). To support compressed data transmission and their consumption by DNN based perception, DNN training with lossy and lossless compressed data (from now on 'compressed' and 'uncompressed' data) is investigated and analysed.

This paper's main contributions are as follow:

1. it proposes a robust methodology to evaluate the effects of automotive sensor data compression on the perception step (i.e. object detection);
2. it demonstrates that high levels of lossy compression (up to ~130:1) can be applied to transmit real-time automotive video camera data to the processing unit(s) without degrading the performance of DNN-based vehicle detection;
3. it establishes that re-training Faster R-CNN with lossy compressed data is beneficial to the DNN performance, when evaluating compressed and uncompressed video camera data;
4. it shows that the benefits are independent from the compression standard used and also if different compression standards and compression rates are used for the training data and transmitted data over the wired vehicle networks;
5. it proposes a process to optimise the compression ratio of data used in training when the compression ratio of the transmitted data is known.

The presented results demonstrate that training with compressed data enhances DNN performance, in terms of Average Precision, AP, and Recall, R, of up to 15% and 20% respectively compared to baseline (i.e. training with uncompressed data). Moreover, results show an increase of up to 3% for both AP and R when transmitting uncompressed data and training with lossy data.

III. Background

It is important to understand if data loss and artifacts generated by compression can degrade the performance of the machine learning (ML) algorithms used for perception in automotive. This section reviews some background knowledge needed to understand the relationship between video compression and ML.

A. *Video Compression*

Video compression has been studied and optimised over many years to reduce the bandwidth and storage requirements to keep up with the ever-increasing image quality and resolution that consumers are looking for. The main aim of this optimisation process has been to achieve video streams wherein the artefacts are not too detrimental to the human visual system perception when viewed at the intended range and speed. Most compression schemes use the following basic steps. Firstly, the image is divided into blocks, which may be as small as a pixel, or considerably larger, e.g. 16x16 or 32x32. Each block may be predicted in one of a number of different ways. *Intra-frame* prediction predicts blocks from previously encoded parts of the same frame. *Inter-frame* prediction predicts blocks from previously encoded parts of other frames (which may in fact be future frames in the presentation order, leading to increased latency in the encoding). In this way, the spatio-temporal redundant nature of video is exploited. A block may also be encoded with no prediction. In most cases the prediction will not be perfect, there will be some residual, i.e. a difference between the block we wish to encode and the chosen prediction block. This residual may be transformed. If the transform is reversible, the coding up to this point will produce a perfect reconstruction of the input stream when decoded. Lossy compression may result from a transform which is not quite reversible, and bigger losses are introduced by the quantisation of the transformed residuals. This quantisation is controlled by the quantisation parameter (QP) and this is the origin of compression artefacts. QP is the parameter by which the output quality of the encoder is controlled. Having chosen a method of prediction and optionally quantised the residual, the resulting bit stream is entropy coded to further reduce the bitrate by coding more common symbols with shorter words (strings of bits) than longer ones. Widely used video compression standards are Advanced Video Coding (AVC) and High Efficiency Video Coding (HEVC) [15]. HEVC provides more flexibility with respect to the compression techniques used in AVC, e.g. there are more macroblock sizes available [16-17]. However, the increase in complexity for HEVC results in a higher computational cost, not only in terms of implementing the tools themselves, but also in the algorithms for selecting which tool to use for every block.

Further to AVC and HEVC, there are newer standards (some of them listed below) released or under development aiming to satisfy the variety of requirements associated with novel applications for video compression techniques, including automotive applications.

1) *Versatile Video Coding (VVC)*

This standard, finalised in 2020, is aimed towards a variety of applications such as higher resolutions, High Dynamic

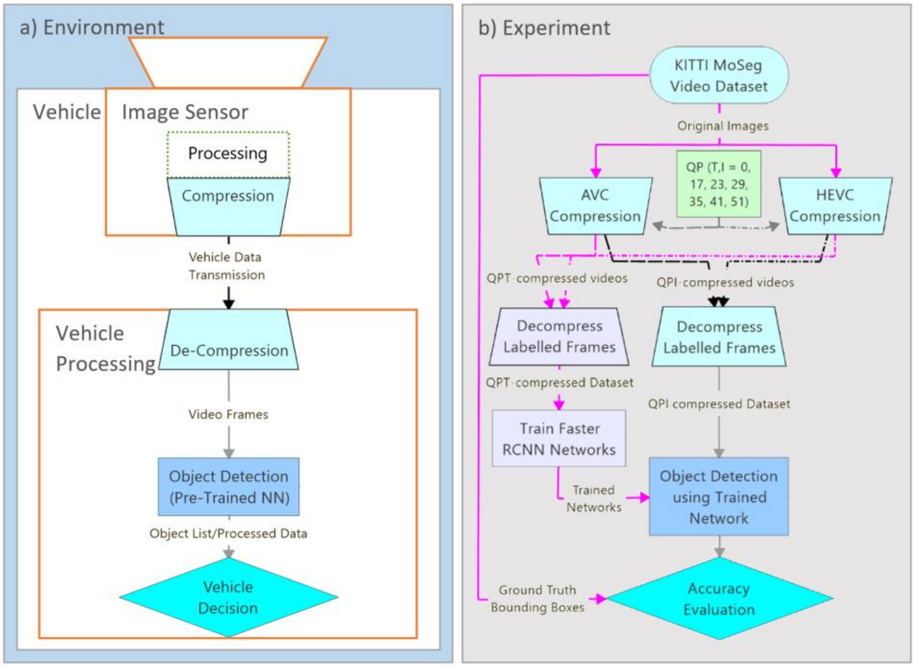


Fig. 1. Flow diagrams schematically representing: a) example of the camera data flow in a vehicle, b) the methodology in our experiment. The magenta lines represent the flow of data necessary for training/evaluation of the DNN. Within a vehicle, the transmitted data are the live videos captured by a camera that need to be transmitted from the sensor to one (or more) processing unit(s) using the vehicle communication data networks.

Range (HDR), etc., and it is expected to make broadcasting and streaming of 4K videos competitive on the market [18]. However, the encoding and decoding complexity is significantly higher than HEVC.

2) *VESA Display Stream Compression (DSC)*

Developed by the Video Electronics Standards Association (VESA), this scheme aims to provide a visually lossless compression for use in portable, embedded systems with a display [10]. DSC has the target of decreasing battery consumption, and the display frame buffer size. It has a maximum compression ratio of approximately 3:1.

3) *VESA Display Compression-M (VDC-M)*

This standard has been created for embedded mobile applications. It has a compression ratio of up to 5:1 but at higher complexity than DSC.

Currently, automotive applications use DSC or VCD-M standards to transmit sensor data to displays and also due to the reluctance to work with lossy data. Moreover, recently JPEG-XS has been proposed to support automotive sensor compression requirements. A review of lossless and lossy video codec for use in ADASs and AVs is presented in [19] and is outside the aim of this paper.

B. Machine Learning (ML)

Due to the recent dramatic improvement and availability of computational platforms, there has been an increase of deep neural network implementations and applications. In automotive systems, there is a significant trend in using Deep Neural Networks (DNNs) to implement functions like object detection and identification, tracking, localisation, planning, free space detection, etc., and these are trained using deep learning (DL) methods [13-14]. DNNs can offer a certain degree of flexibility in achieving their functions in spite of the variations of the cases (inputs) presented to them, and therefore the automotive research is focusing on the safe application of DL methods in ADASs and AVs [20]. However, DNNs can increase the computational costs, causing unacceptable latency, they may not be fully accurate, and their explicability is still debatable [21].

An established application of DL methods in combination with camera data is object detection and classification. This task mainly corresponds to drawing bounding boxes around the objects in the frame and attributing them to a class (category) with a specified confidence level. This task consists of identifying the position and dimension of objects within the scene, and assigning them a class with a certain probability. Many object detection DNNs have been proposed, with a trade-off between their speed, accuracy, and network size [22]. DL object detectors can be broadly categorised in two classes, as described below [23].

1) *Single-Staged Network*

Single-staged networks use an end-to-end process that includes the region proposal (object bounding box prediction) and identification in the same step. This approach generates a fixed number of bounding boxes for each frame. The generated bounding boxes are filtered out to produce as an output only one bounding box for each real object in the scene; this selected bounding box should have the correct size and position for a specific ground truth object. Examples of single-staged networks are: single short detectors (SSD), YOLO networks, etc.

2) *Two-Stage Network*

In two-staged networks, the first stage extracts the object regions. These regions are passed to the second stage to classify the objects and refine the region itself. Examples of this type of network are: region proposal CNN (R-CNN), spatial pyramid pooling (SPP) Network, Faster R-CNN, Featurised image pyramids networks, etc.

Generally, two-staged networks can have higher accuracy than single-stage networks but slower performance, even though Faster R-CNN are considered networks with performance closer to real-time detection [22].

A key aspect of any DNN is how it has been trained; training strategies can be divided in three categories [13, 24].

3) *Supervised Training*

In supervised training, the network uses labelled and classified data as inputs for the learning. The learning process is based on measuring the network prediction accuracy using the labelled data and iteratively improving this accuracy during the training process.

4) *Unsupervised Training*

In unsupervised training the network uses unlabelled data and does not have any additional information as input for training. During training, the network seeks to identify its own patterns and output classifications.

5) *Reinforcement Training*

In reinforcement training, similarly to unsupervised strategies, there is no need for labelled data. The network trains by trial and error to maximise the received "rewards", based on a "reward function" (defined by the network user). Based on this process, an optimal or nearly-optimal policy will be developed by the network.

Training DNNs from the base up can require a large amount of data and time. Transfer learning uses previously trained layers as a backbone and tunes the network by re-training it with a new dataset. This process can entail maintaining the same output classes or also defining new classes [25].

Once the DNNs are trained, they can be deployed to implement their own task (e.g. detection and classification) on data which the networks have not seen before. Common performance metrics for DNN performance are AP and R [26], see Sec. IV. D. Some recent publications have also focused on establishing new criteria to evaluate image quality in the context of machine learning applications; for example, Spatial Recall Index (SRI) aims to understand which pixels have a higher impact on the performance of used DNNs [27].

IV. Related Work

With the increasing number of sensors constituting the sensor suite of ADAS and automated functions, and the amount of their generated data, recent studies have been focusing on techniques to reduce the sensor data size without introducing unreasonable losses/artefacts.

One of the possible ways to reduce the amount of camera data to be transmitted over the wired data networks is to convert the HDR frames (in automotive a range in excess of 140dB is required to cope with luminosity variations) into Low Dynamic Range (LDR) using tone mapping and backward compatible approaches. Mantiuk et al. introduced a technique to encode a HDR stream into LDR and residual streams combined to reconstruct the original HDR frames. The proposed technique is compatible with MPEG decoders [28]. Debattista has proposed to use genetic programming to produce tone mapped LDR frames optimised for compression using JPEG with different quality levels. Presented results have improved performance with respect to previous works [29-30]. Recently, Dabrowski's group has been investigating tone-mapping specifically for ADAS. For example, they have been looking into tone-mapping specifically for night vision and rejection of light reflections. Presented results show improved performance in terms of road signs readability and image quality with respect to traditional tone mapping techniques, while offering higher throughput [31]. In a follow up work, the group also analysed tone mapping for a specific automotive colour filter array used in the camera sensor, red-clear-clear-clear, demonstrating the possibility to achieve competitive compression ratios and real-time performance [32].

However, more radical solutions to the video compression issue are under investigation in automotive. For example, end-to-end DNN techniques can be used, as shown in [33]. In this work, the authors have investigated the use of generative adversarial networks (GAN) to implement compression coding and decoding and compared the results with JPEG2000. The GAN output gives better results in terms of image segmentation compared to segmentation implemented on JPEG2000 compressed images, but the data compressed with JPEG2000 have a better PSNR. Interestingly, also Chamain et al. have investigated end-to-end compression techniques and discuss specifically different ways of optimising the pipeline coding-decoding-DNN based perception (e.g. object detection) [34]. Another area of research is compressive imaging combined with decoding and DNN detection, proposed in [35]. The paper presents a very interesting perspective on the topic, discussing how much pre-processing of images is needed, but also the actual need of implementing decoding, which the Authors remove, achieving comparable performance with respect to reconstructed videos.

In this context, our previous work has investigated compression tuning of DNN based object detection and the use Motion JPEG compressed datasets. The work shown an improvement of AP and R when the DNNs are trained with compressed data [2]. However, M-JPEG does not consider temporal redundancy between frames, and better compression standards can be considered for automotive. The authors have also investigated compression based on regions of interest [36]. In this previous work, the region not of interest, such as the sky and buildings, are compressed to a higher degree compared to region of interest. The result of the two level compression was evaluated with a segmentation perception task.

V. Methodology

This work expands on [2] and evaluates the effect on object detection via compression tuned Faster R-CNN (i.e. a two-stage detector, see Sec. II. B); video compression has been implemented using AVC and HEVC. The results have been further validated using an off the shelf one-stage detector, i.e. a YOLOv5 network, Sec. VI.D, similarly to what shown in [37]. Additionally, in this work, the use of images compressed using different compression standards for the training and transmitted datasets was investigated for the first time. The complete experimental process is shown in Fig.1b). The KITTI Moseg dataset was chosen for this study, hereafter referred to as the 'original dataset' [38]. The series of datasets used for testing and training the selected DNN are generated by compressing the original dataset into videos with different Quantisation Parameter, QP. We then implemented transfer learning on the

TABLE I
TOTAL SIZE OF THE COMPRESSED DATASET WHEN VARYING QP

QP	Total Compressed Dataset File Size (Bytes)		Compression Ratio (based on the raw dataset size of 2,024,616,250 Bytes)	
	AVC	HEVC	AVC	HEVC
0	780,118,312	879,129,975	1:3	1:2
17	146,862,722	206,085,465	1:14	1:10
23	64,705,207	101,412,155	1:31	1:20
29	28,299,339	43,227,152	1:72	1:47
35	12,616,765	15,586,079	1:160	1:130
41	5,740,979	5,152,473	1:353	1:393
51	2,015,010	1,585,235	1:1005	1:1277

selected Faster R-CNN network with the differently compressed training datasets, generating 14 compression-tuned networks. These networks correspond to the DNNs that can be deployed in processing units of the vehicles and used to implement object detection real-time, Fig.1a). The transmitted sensor data were again simulated by using one of the compressed datasets, and the DNNs were used to infer across all these different datasets.

A. KITTI MoSeg Dataset

There are many datasets available for developing automated vehicle capabilities, collected by various research groups [39]. These datasets are generally captured through an array of perception sensors mounted on a vehicle driving in the real world in a variety of environments and scenarios. These datasets can contain time synchronised sensor data as well as a variety of additional data such as object labels, information from the ego-vehicle, environmental conditions, etc.

The KITTI MoSeg dataset is a subset of the KITTI dataset with extended annotations [40-41]. This subset of the database contains 6 training videos and 2 testing videos of 1449 time correlated frames with moving and static vehicle labels captured at 10 frames per second [40-41].

We chose to detect vehicles (static and dynamic), since the vehicle class is of foremost importance for ADASs and AVs, as other vehicles are more commonly encountered by the ego vehicle in different environments and scenarios, posing a higher threat of accidents. During 2020 in the UK, there were 91,199 accidents recorded on the road with 72% of these incidents involving another vehicle [42].

B. Compression of the Dataset

The QP used to compress the datasets using AVC and HEVC can be varied from a minimum of 0 up to a maximum of 51, where a QP of 0 represents lossless compression, QP of 17 is considered visually lossless (based on the human visual system), QP of 23 is the default in AVC. Every increase in 6 for the QP roughly results in the compressed file size being smaller by a factor of 2. In this work, the datasets compressed into videos using constant QPs of 0, 17, 23, 29, 35, 41 and 51 in both AVC and HEVC were created from the original KITTI Moseg dataset (so a total of 14 datasets were generated). This conversion from frames to compressed video was achieved through ffmpeg, using the libx264 and libx265 codecs for AVC and HEVC compression respectively, with varying QP through the dedicated flag. Then, each frame in the compressed videos was saved as a lossless image. The compressed video sizes at each QP and compression standard are shown in Table I.

C. Neural Network Compression Training

The network used for object detection is a Faster R-CNN with a Residual Network, ResNet-101, as the backbone. ResNet-101 offers a good trade-off between almost real-time operation and accuracy, has a size of 167 MB, 44.6M parameters, and 101 layer-depth with 3-layer residual blocks [43]. In 3-layer residual blocks, a shortcut is employed where the output of a layer is passed through and affects the output of the third layer after the original layer [43]. As mentioned above, a partial comparison of the achieved results has been carried out with YOLOv5 network, representative of a widely used one stage detector [44]. The experiments were carried out on an i9-10885H CPU with an Nvidia Quadro RTX 5000 with Max-Q design and 16GB GDDR6 VRAM using Matlab and pytorch platforms for Faster-RCNN and YOLOv5 respectively.

Each one of the 14 datasets described in par. III. B was split with an 8:2 ratio to form the training and validation sets respectively for the DNNs. The Faster R-CNN hyperparameters were optimised based on the original MoSeg dataset. A series of 14 networks were then re-trained (without changing the hyperparameters) using each of the generated training datasets. Of these 14 networks, two were trained with uncompressed data (QP_T=0 for AVC and HEVC compression, where QP_T is the QP used for the DNN training dataset) and 12 with lossy compressed data, Fig.1b) magenta lines; thereafter they will be referred to as *uncompressed tuned networks* and *compression tuned networks* respectively. Each trained network was used to evaluate every one of the 14 datasets compressed in both AVC and HEVC with the selected QP_I. We will call the datasets for testing the *transmitted data*, as they represent the data that will be transmitted on the vehicle communication networks, black arrows in Fig.1a-b).

D. Neural Network Evaluation

Precision is a measure of the number of achieved accurate predictions (True Positive predictions, TP) compared to all the predictions that the network outputs, (1).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

On the other hand, Recall measures the number of accurate predictions compared to the total number of objects in the ground truth data, as shown (2). Some of the ground truth labelled objects cannot be detected by the network and are therefore considered False Negative objects.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

A bounding box determined by the DNN is a True Positive if its Intersection over Union (IoU) with a bounding box in the ground truth is larger than the defined threshold (in this work the threshold is 0.5). The IoU is the area of overlap between the detected and ground truth bounding boxes divided by the total area covered by the two bounding boxes. If the detected object bounding box IoU is below the threshold, we have a False Positive. Finally, the False Negative value is the number of ground truth bounding boxes which do not have an associated

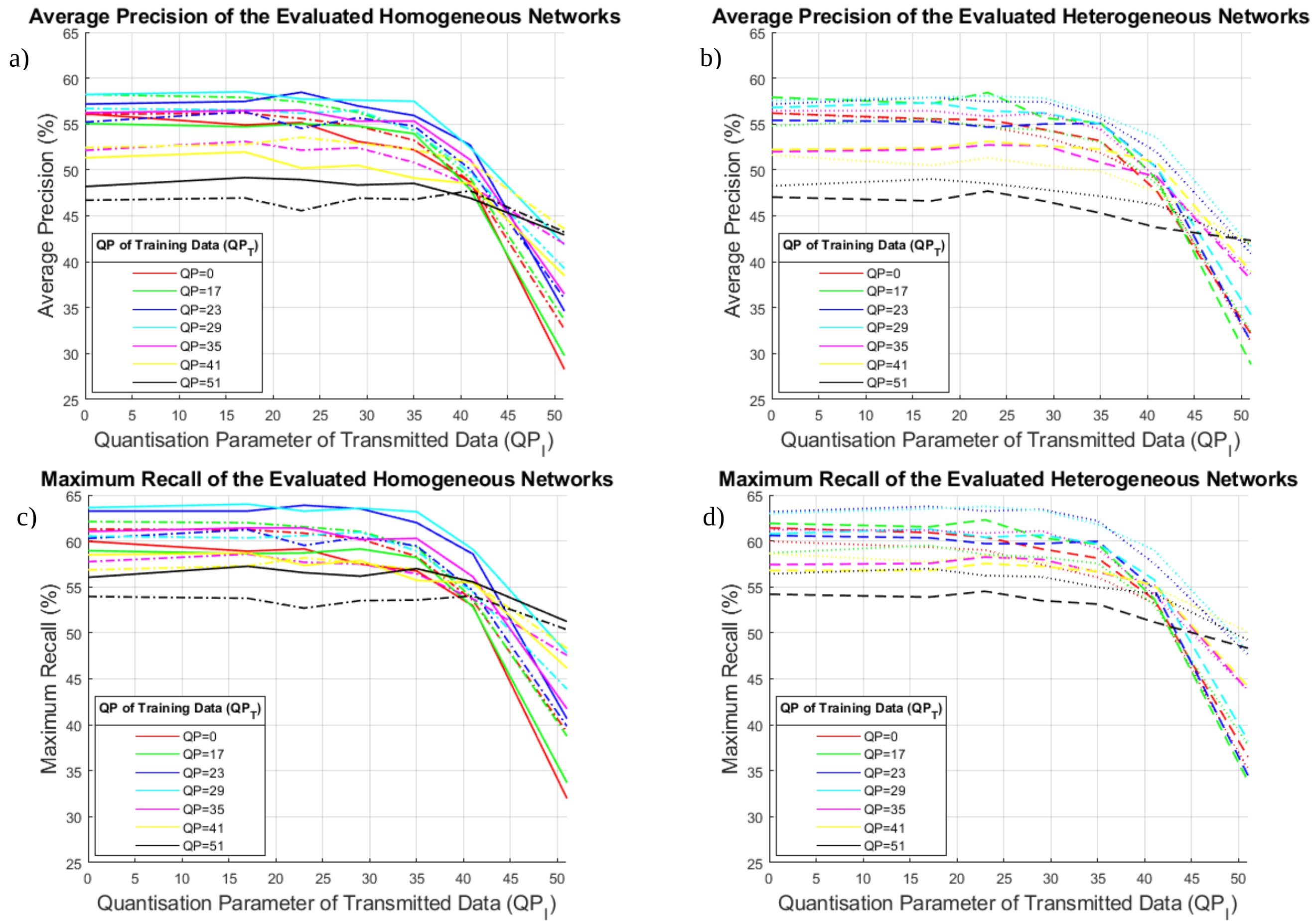


Fig. 2. Graphs showing the calculated AP and R for all the 14 DNNs evaluating the 14 created datasets. a) and c) Average Precision and Maximum Recall for homogeneous transmission, b) and d) Average Precision and Maximum Recall for heterogeneous transmission. Each line colour represents a different QP_T and different line styles represent the different combinations of compression standards used for training and transmitted datasets (respectively with quantization parameters QP_T and QP_I). Solid line represents using AVC -AVC compression for training and transmitted datasets; dash-dotted line represents using HEVC-HEVC; dashed line represents HEVC – AVC; dotted line represents AVC – HEVC.

DNN detected bounding box with IoU above the threshold. Average precision and recall are calculated accordingly to the PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit [45]. Namely, Precision is calculated for each frame and averaged for each detection class. Recalls are assumed to be initialised to zero, and then the number of true positive objects is cumulated until all the frames are analysed and the value of recall per class is calculated.

AP and R were computed for every one of the 14 compression-tuned DNN and for every one of the 14 compressed datasets, each one of them characterised by the selected compression standard and QP_I. We have computed these two values for the four possible combinations of standards used to compress the training and the transmitted dataset. We call *homogeneous transmission* when we use either AVC-AVC for compression of training and transmission datasets, or when we use HEVC-HEVC. However, it is also possible to assume that transmitted data can be compressed with a standard (e.g. AVC) different from the standard used for the DNN training dataset (e.g. HEVC). We have therefore evaluated *heterogeneous transmission*, where AVC-HEVC or HEVC-AVC combinations have been used to compress the training and transmission datasets respectively.

VI. Results

A. *DNN object detection performance*

Fig. 2 a)-b) and c)-d) show respectively the evaluated DNN average precision and maximum recall plotted against QP_I, namely plotted for increasing values of compression of the transmitted data. Each line colour represents a different QP_T, i.e. the compression rate of the training dataset used for compression-tuning the Faster R-CNN. It is possible to observe that the trend of all the lines is similar, AP and R are stable or with small variations until a QP_I equal to 29 or 35, and then they decrease significantly, up to around 30% decrease (only for QP_T equal to 41 and 51 the decrease is smaller). Overall, re-training with $QP_T = 29$ using AVC compressed data and inferencing on AVC compressed transmitted data yielded better average precision at most QP_I tested. Furthermore, DNNs retrained with a QP_T of 23 and 29, AVC compression, are mostly the best performing for $QP_I < 41$.

B. *DNN's performance when training with compressed data*

Fig. 3 a)-d) show, for the four combinations of homogeneous and heterogeneous transmission, the comparison of the DNNs re-trained with uncompressed dataset (blue lines, QP_T=0) with compression-tuned DNNs (yellow and orange lines). AP (left axis) and optimised QP_T (right axis) are plotted versus QP_I. The blue lines are used as baseline, as it is what current implemented in vehicle object detectors, i.e. the DNN training is

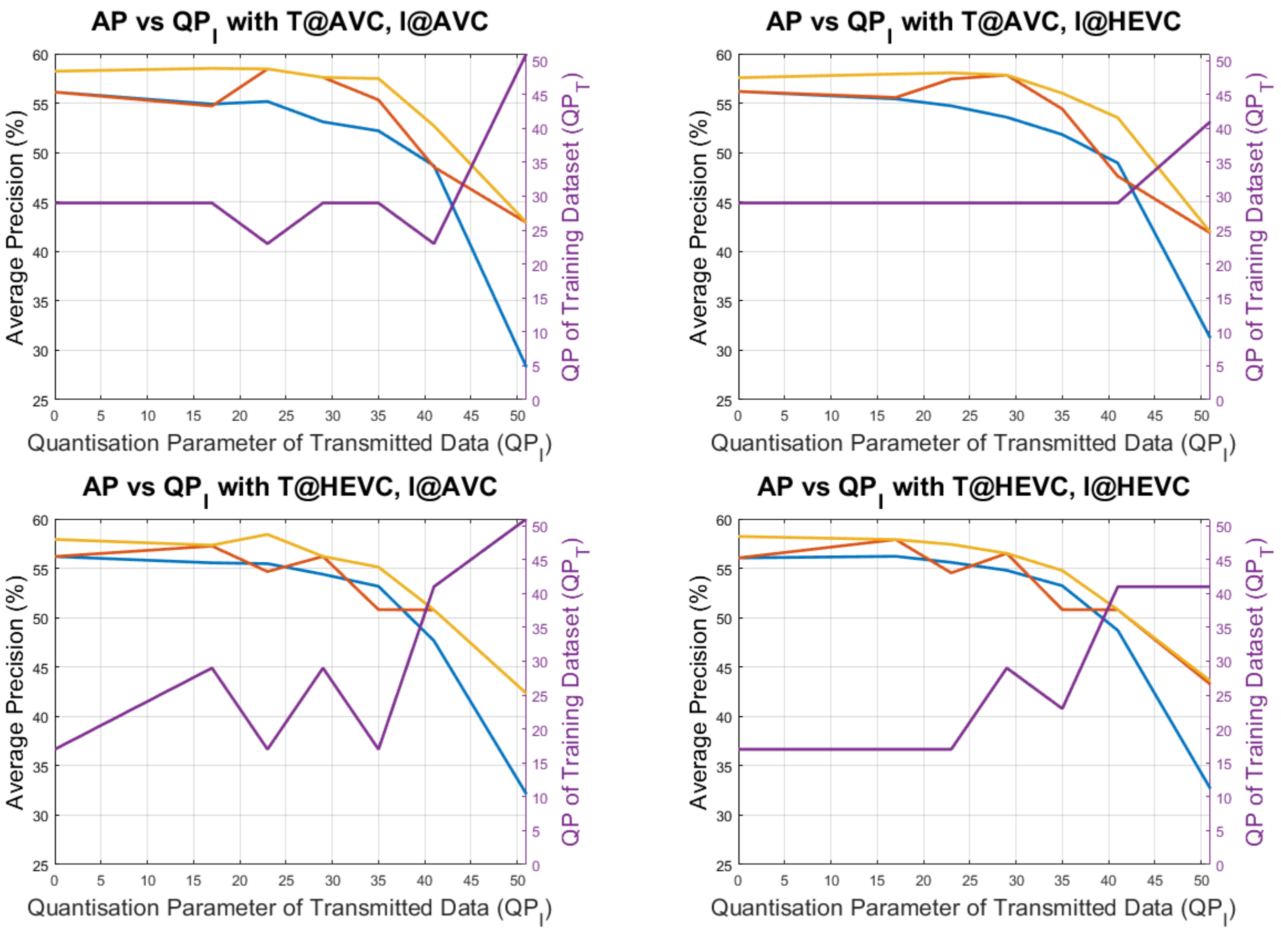


Fig. 3. Average precision (left axis) and training dataset QP_T (right axis) versus transmitted data QP_I for the following compression combinations: (a) AVC for training and transmitting; (b) AVC for training and HEVC for transmitting; (c) HEVC for training and AVC for transmitting; (d) HEVC for training and transmitting. In each plot, the blue lines are the values achieved for QP_T=0 (training with lossless data), the orange are for QP_T= QP_I (training with the same level of compression of the transmitted data), yellow are for the maximum achievable AP as a function of QP_I, and the purple plots (right axis) show the related QP_T for each point in the yellow plot.

implemented using uncompressed data. The orange lines show the achievable AP if the training and transmitted datasets have the same quantisation parameter (QP_T = QP_I), and outperforms the baseline in most of the points. The yellow lines show, for each QP_I, the maximum AP achieved as a function of the compression ratio of the dataset used for the re-training, and the purple lines (right axes) visualise the value of QP_T used for the training data that yields to the maximum AP. The yellow line always outperforms the blue line across a) to d), demonstrating that re-training with compressed data provides the best performance at all the compression ratios of the transmitted data, even for transmitted uncompressed data (QP_I =0).

Fig. 4 compares the best achieved AP values versus increasing QP_I, i.e. increasing compression ratio of transmitted data. The four combinations of homogeneous and heterogeneous compression are indicated by the line styles. Although using AVC for re-training and transmission (solid blue plot) performed the best, there were minor differences in performance and all the combinations exhibited a similar trend. All the combinations showed a stable performance between 0 and 29 QP_I. With an appropriate selection of QP_T for DNN tuning, the performance improved even when lossy data were transmitted (QP_I > 0). Additionally, at QP_I of 29, 35 and 41, there is a performance gap between training with AVC and HEVC compressed data.

VII. Discussion

As previously described, we have been focusing on compression tuning a Faster R-CNN by re-training with compressed video datasets using different compression standards (AVC and HEVC) and compression ratios (i.e. $0 \leq QP_T \leq 51$). The performance of these compression tuned DNNs (we have re-trained 14 DNNs) was evaluated when inferencing on compressed video camera data. The achieved results can ultimately inform the decision of transmitting compressed sensor data over traditional vehicle communication networks without impairment in the perception step (based on vehicle detection).

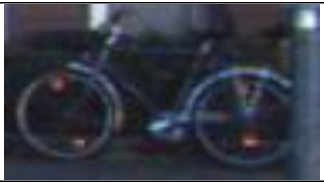
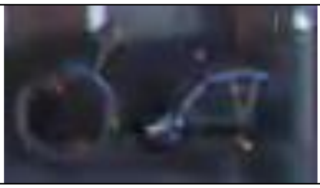
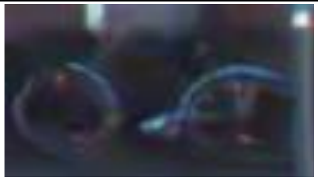
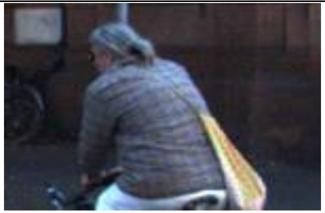
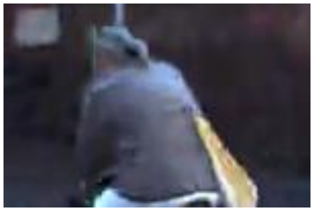
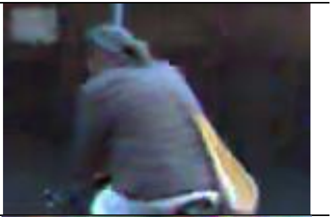
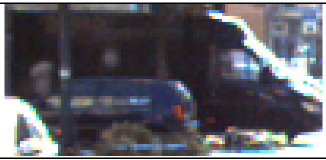
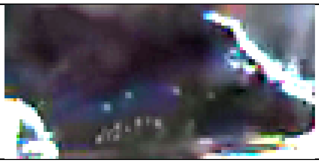
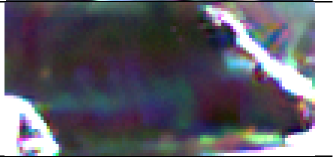
A. DNN's performance enhancement and optimisation when transmitting compressed data

Fig. 2 demonstrates an improvement of the AP for both compressed and uncompressed transmitted data when using compression tuned DNNs. This beneficial effect is observable for all the compression ratios (including QP_I=0), with an increase of the average precision up to around the 15% with respect to the performance of the DNN tuned with uncompressed data. Notably, for QP_I=0, i.e. equivalent to current transmission via wired vehicle networks, performance

d)

TABLE II

COMPARISON OF ARTEFACTS RESULTING FROM COMPRESSION, IMAGES HAVE BEEN STRETCHED FROM THEIR ORIGINAL SIZE

Artefact (Identified errors)	Original	Compressed at 41 QP AVC	Compressed at 41 QP HEVC
Loss of edge definition (deformation of rear bicycle wheel, blurring of bicycle)			
Attenuation of high frequencies (coat pattern and ground) Block artefacts (face of person)			
Posterisation and loss of edge definition (vehicle blurring and blending into van)			

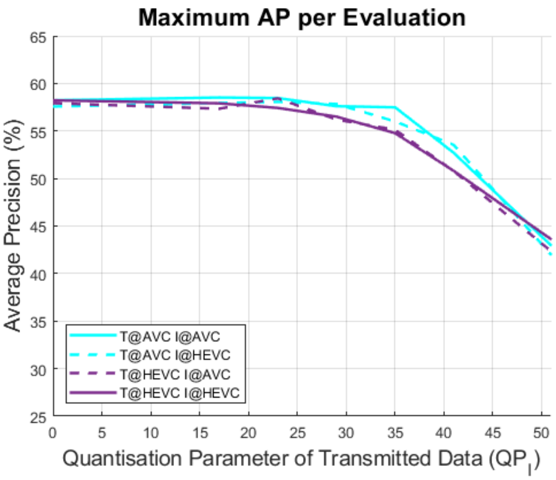


Fig. 4. Average precision against QP_I of the evaluation dataset. Each line represents the maximum average precision achieved based of a different combination of the standard used to compress the training, its QP_T, and evaluation dataset. T stands for the standard used for compressing the training dataset and I stands for the standard used for transmitted data

increases by ~2% when using compression tuned DNNs.

Fig. 3 shows that networks re-trained with uncompressed data do not yield the best performance with any of the transmitted datasets, from uncompressed to different degrees of compression. Compression of the dataset for DNN re-training provides higher AP and R, regardless of the compression rate and compression standard used for the transmitted data (even for uncompressed data), when the compression ratio of the re-training dataset is *optimised* (i.e. QP_T is optimised according to the purple lines in Fig. 3). If the compression ratio QP_I of transmitted data is known, QP_T optimisation is key to achieve the best DNN performance, as demonstrated by comparing the orange lines with the yellow ones in Fig 3. In fact, re-training and transmitting data using QP_T=QP_I provided the best results only for a few QP_I values (orange and yellow lines intersecting/overlapping in figure). The importance of QP_T optimisation is further demonstrated by the plots in Fig. 2. The red lines correspond to the uncompressed re-trained DNNs, and it is evident that re-training with compression ratios above a certain threshold (i.e. QP_T > 35, yellow and black lines) is not beneficial anymore.

Fig. 3, purple lines, shows that AVC re-trained DNN favours a slightly higher QP_T for training compared to HEVC. Fig. 3 a) and b) indicate a QP_T of 29 for AVC compression as the suitable value in most scenarios other than heavy compression of transmitted data (higher QP_I). For HEVC compressed training, the optimal value can be a QP_T of either 17 or 29, when transmitted data are not heavily compressed.

B. Homogeneous vs Heterogeneous compression tuning

Another aspect highlighted by the presented results is that compression tuned DNN have enhanced performance for transmitted data compressed with the same standard (i.e. AVC-AVC and HEVC-HEVC compression pairs for re-training and transmission), but also when transmitted data are compressed with another standard, as shown in Fig. 4. Furthermore, when QP_T is between 29 and 41, there is an increased difference in performance between training AVC and HEVC, with AVC performing better. As QP increases, artefacts in the decoded videos are more prominent. The artefacts could be recognised as a feature of the detected class during training or evaluation causing false positives. Artefacts arising from lossy compression will differ between AVC and HEVC standards and may cause the difference in performance; Table II highlights some of the artefacts arising from compression.

C. Colour space and qualitative analysis

It is worth noting, when training with lossless compressed data and inferring with lossless data (QP_T=QP_I=0), the average precision is not the same for the four homogeneous and heterogeneous combinations (average precision ranges between 56.05 to 56.19 and max recall ranges between 59.96 and 61.42). In theory lossless compression should yield the same image even if different compression algorithms are used. Upon closer inspection of the two lossless datasets (QP=0, AVC and HEVC

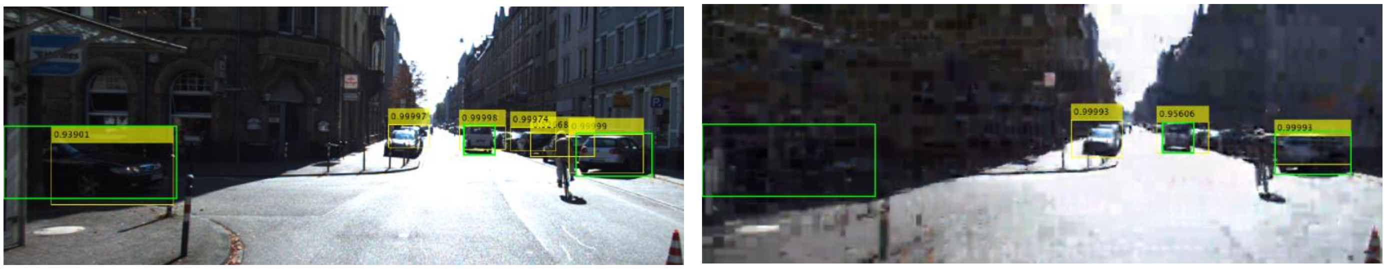


Fig. 5. Sample of the testing dataset with bounding boxes overlayed: green represents the ground truth labels and yellow represents neural network vehicle labels with confidence level above. a) Image compressed at 0 QP with AVC, b) Image compressed at 51 QP with AVC.

compression) compared to the original dataset, there are minor differences in many pixels. In the image shown by Fig. 5 a), QP=0 AVC compression, the 18.4% of the RGB pixels differed by a value of one with respect to the original dataset. Similarly, the 9.8% of the QP=0 HEVC compressed pixels differed by 1 from the original data. There were also around 0.1 % of pixels differed by 2 in both compression standards and a minor amount in HEVC which differed by 3. These minor differences are the result of the colour space conversion. In AVC and HEVC, the RGB data is converted into YCbCr colour space during compression, and the reversed back during decompression [19]. This conversion will involve rounding, resulting in the produced AVC and HEVC lossless images being slightly different between each other and with respect to the original frames. The DNN AP and R for $QP_T=QP_I=0$ show that minor changes to pixel values can have an effect on the performance of the object detection algorithm, in our case up to ~ 2% difference between the HEVC and AVC trained DNNs.

Finally, to further understand the cause of false positives and negatives identified by the DNNs, we visually inspected some of the frames. We plotted the frames of the transmitted datasets, e.g. Fig. 5, overlaying the ground truth bounding boxes (green rectangles) and the bounding boxes identified by the DNN (yellow rectangles). In the selected frame, there are 6 vehicles identified by the neural network in Fig. 5a, and 3 vehicles in Fig 5b. In both these images the DNN generated bounding boxes are all correctly identified vehicles. However, the detections not overlapping with the ground truth are recognised incorrectly as false positive objects. Although they all are true positives, the lack of accurate ground truth is detrimental to the evaluation of the DNN performance. When reviewing the KITTI dataset, this kind of mismatched ground truth was present in a significant number of evaluated images. There were also mislabels in which vehicles fully obstructed by another object were still labelled in the ground truth, as further discussed in [46].

D. Comparison with one-stage detectors

To validate our results, a reduced set of the experiments have been carried out with a one-stage detector, i.e. YOLOv5. The network has been compression-tuned with uncompressed and compressed data (i.e. $QP_T=0$, 29, respectively). In terms of testing sets, $QP_I=0$, 29, 35, 41 have been evaluated. The AP results show similar trends to that of the ones identified in our original experiments. The $QP_T=29$ trained YOLOv5 yields similar performance than the2 $QP_T=0$ trained YOLOv5 when evaluating uncompressed data. Moreover, when evaluating compressed data, the $QP_T=29$ trained YOLOv5 outperforms the YOLOv5 trained with uncompressed data (up to ~3%). These results demonstrate that different DNNs will have different behaviours when consuming compress data, and a part of the future work in the design of the AAD *sense-perceive-control* pipeline will be to select and optimise the DNN architecture best suited to consume compressed data.

VIII. Conclusion

This work investigated the effect of lossy compression on the transmission via traditional communication data networks of video camera data for ADAS and AV applications. Currently, automotive manufacturers prefer to obtain uncompressed data from camera sensors, which puts a significant burden on the vehicle communication networks. Assuming that these data will be consumed by neural networks, we have analysed how to optimise the DNN compression tuning based on the compression level of transmitted data. We have shown that lossy compression of transmitted data with AVC or HEVC, with a QP≤29, does not significantly affect the performance of Faster R-CNN. Moreover, the DNNs re-trained and optimised with a compressed dataset always outperform the DNNs trained with uncompressed datasets (QP=0), with an improvement of up to the 15% and 20% for average precision and recall respectively. This improvement is noticeable also when transmitting uncompressed data, potentially because compression-tuning the DNNs makes them more robust to real world variations in the dataset. This work paves the way to the use of compressed videos in combination with neural networks in assisted and automated driving functions.

Future work will entail a further inspection of the minor changes in pixels and their effect on the DNN performance, as well as expanding the study to include a larger number of object classes. Additionally, the use of optimised light-weight and low-latency lossy coding techniques needs to be investigated, in combination with the optimisation of the selected DNN for the selected compression method.

Acknowledgment

The Authors wish to acknowledge Yingning Sun and Graham Kirsch for early discussions on this topic. The Authors are deeply grateful to Gabriele Baris, vising PhD student at the University of Warwick, for the experiments with YOLOv5.

References

[1] SAE, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," J3016_201806, 2018.

[2] 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 2021: Imaging for Deep Learning*, 2021, pp. 78–82.

[3] F. Reway, W. Huber, and E. P. Ribeiro, “Test Methodology for Vision-Based ADAS Algorithms with an Automotive Camera-in-the-Loop,” in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2018, pp. 1–7.

[4] S. Heinrich, “Flash memory in the emerging age of autonomy,” in *Proceedings of the Flash Memory Summit*, 2017, pp. 7–10.

[5] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, “Intra-Vehicle Networks: A Review,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 534–545, 2015.

[6] L. van Dijk and G. Sporer, “Functional Safety for Automotive Ethernet Networks,” *J. Traffic Transp. Eng.*, vol. 6, no. 4, pp. 176–182, 2018.

[7] L. Lavagno, C. Oppedisano, and S. Cicciarell, “Design of a data aggregation circuit for Autonomous Driving LiDAR sensors,” Politecnico di Torino, 2021.

[8] P. Krömer, M. Prauzek, M. Stankuš, and J. Konečný, “Fuzzy Video Compression Control for Advanced Driver Assistance Systems,” in *2018 26th International Conference on Systems Engineering (ICSEng)*, 2018, pp. 1–9.

[9] H. Hsiang, K. C. Chen, P. Y. Li, and Y. Y. Chen, “Analysis of the Effect of Automotive Ethernet Camera Image Quality on Object Detection Models,” in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 21–26.

[10] F. G. Walls and A. S. Macinnis, “VESA Display Stream Compression for Television and Cinema Applications,” *IEEE J. Emerg. Sel. Top. Circuits Syst.*, vol. 6, no. 4, pp. 460–470, 2016.

[11] A. Descampe *et al.*, “JPEG XS-A New Standard for Visually Lossless Low-Latency Lightweight Image Coding,” in *Proceedings of the IEEE*, 2021, vol. 109, no. 9, pp. 1559–1577.

[12] R. Swann and N. Kingsbury, "Error resilient transmission of MPEG-II over noisy wireless ATM networks," in *Proceedings of International Conference on Image Processing*, 1997, pp. 85-88.

[13] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, “A Survey of Deep Learning Applications to Autonomous Vehicle Control,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 712–733, 2021.

[14] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A survey of deep learning techniques for autonomous driving,” *J. F. Robot.*, vol. 37, no. 3, pp. 362–386, 2020.

[15] Y. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms and Standards, Third Edition*, Third. Boca Raton, Florida: CRC Press, 2019.

[16] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[17] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[18] B. Bross *et al.*, “Overview of the Versatile Video Coding (VVC) Standard and its Applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021.

[19] P. Pawlowski, K. Piniarski, and A. Dabrowski, “Selection and tests of lossless and lossy video codecs for advanced driver-assistance systems,” in *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2018, pp. 344–349.

[20] Z. Wang, B. Jiao, and L. Xu, “Visual Object Detection: A Review,” in *2021 40th Chinese Control Conference (CCC)*, 2021, pp. 7106–7112.

[21] S. Strauß, “Deep automation bias: How to tackle a wicked problem of ai?,” *Big Data Cogn. Comput.*, vol. 5, no. 2, pp. 1–14, 2021.

[22] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A Survey of Modern Deep Learning based Object Detection Models,” Preprint submitted to IET Computer Vision, 2021.

[23] J. Huang *et al.*, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3296–3305.

[24] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, “A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science,” in *Supervised and Unsupervised Learning for Data Science*, Springe, Cham, 2020, pp. 3–21.

[25] L. Shao, F. Zhu, and X. Li, “Transfer learning for visual categorization: A survey,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, 2015.

[26] I. Park and S. Kim, “Performance indicator survey for object detection,” in *2020 20th International Conference on Control, Automation and Systems (ICCAS*, 2020, pp. 284–288.

[27] P. Müller, M. Brummel, and A. Braun, “Spatial Recall Index for Machine Learning Algorithms,” in *London Imaging Meeting 2021: Imaging for Deep Learning*, 2021, pp. 58–62.

[28] R. Mantiuk, A. Efremov, K. Myszkowski, and H. P. Seidel, “Backward compatible high dynamic range MPEG video compression,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 713–723, 2006.

[29] R. K. Mantiuk, “Practicalities of predicting quality of high dynamic range images and video,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, vol. 2016-Augus, pp. 904–908.

[30] K. Debattista, “Application-specific tone mapping via genetic programming,” *Comput. Graph. Forum*, vol. 37, no. 1, pp. 439–450, 2018.

[31] K. Piniarski, P. Pawlowski, and A. Dabrowski, “Efficient HDR tone-mapping for ADAS applications,” in *Efficient HDR tone-mapping for ADAS applications*, 2019, pp. 325–330.

[32] P. Pawłowski, K. Piniarski, and A. Dąbrowski, “Highly efficient lossless coding for high dynamic range red, clear, clear, clear image sensors,” *Sensors*, vol. 21, no. 2, pp. 1–17, 2021.

[33] J. Löhdefink, A. Bär, N. M. Schmidt, F. Hüger, P. Schlicht, and T. Fingscheidt, “GAN-vs. JPEG2000 image compression for distributed automotive perception: Higher peak SNR does not mean better semantic segmentation,” *arXiv Prepr.*, vol. arXiv:1902, 2019.

[34] L. D. Chamain, F. Racape, J. Begaint, A. Pushparaja, and S. Feltman, “End-to-End optimized image compression for machines, a study,” in *Data Compression Conference Proceedings*, 2021, vol. 2021-March, pp. 163–172.

[35] S. Lu, X. Yuan, and W. Shi, “Edge Compression: An Integrated Framework for Compressive Imaging Processing on CAVs,” in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020, pp. 125–138.

[36] Y. Wang, P. H. Chan, and V.Donzella, “Semantic-aware Video Compression for Automotive Cameras,” in *IEEE Traactions on Intelligent Vehicles*, Accepted, March 2023.

[37] B. Li, P. H. Chan, G. Baris, M. D. Higgins and V. Donzella, "Analysis of Automotive Camera Sensor Noise Factors and Impact on Object Detection," in *IEEE Sensors Journal*, vol. 22, no. 22, pp. 22210-22219, 2022,

[38] K. Gauen *et al.*, “Comparison of visual datasets for machine learning,” *2017 IEEE Int. Conf. Inf. Reuse Integr.*, pp. 346–355, 2017.

[39] C.-É. N. Laflamme, F. Pomerleau, and P. Giguère, “Driving Datasets Literature Review,” Québec City, 2019.

[40] M. Siam, H. Mahgoub, M. Zahran, and A. El-Sallab, “MODNet: Moving Object Detection Network with Motion and Appearance for Autonomous Driving.,” *arXiv Prepr. arXiv1709.04821*, 2017.

[41] Geiger Andreas, Lenz Philip, Stiller Christoph, and Urtasun Raquel, “Vision meets robotics: The KITTI dataset,” *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[42] Department for Transport, “Road Safety Data - Accidents 2020,” Department for Transport, 2021, Accessed on 08 December, 2021. Avaliable: https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data.

[43] 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.

[44] *YOLOv5 by Ultralytics*. Version 7.0, Github. Accessed: 4th April 2023. Software. Available: https://doi.org/10.5281/zenodo.3908559

[45] M. Everingham and J. Winn, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit,” in *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[46] B. Li, G. Baris, P. H. Chan, A. Rahman and V. Donzella, "Testing ground-truth errors in an automotive dataset for a DNN-based object detector," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, 2022, pp. 1-6.