

# Task-driven Semantic Coding via Reinforcement Learning

Xin Li, Jun Shi and Zhibo Chen, *Senior Member, IEEE*,

**Abstract**—Task-driven semantic video/image coding has drawn considerable attention with the development of intelligent media applications, such as license plate detection, face detection, and medical diagnosis, which focuses on maintaining the semantic information of videos/images. Deep neural network (DNN)-based codecs have been studied for this purpose due to their inherent end-to-end optimization mechanism. However, the traditional hybrid coding framework cannot be optimized in an end-to-end manner, which makes task-driven semantic fidelity metric unable to be automatically integrated into the rate-distortion optimization process. Therefore, it is still attractive and challenging to implement task-driven semantic coding with the traditional hybrid coding framework, which should still be widely used in practical industry for a long time. To solve this challenge, we design semantic maps for different tasks to extract the pixelwise semantic fidelity for videos/images. Instead of directly integrating the semantic fidelity metric into traditional hybrid coding framework, we implement task-driven semantic coding by implementing semantic bit allocation based on reinforcement learning (RL). We formulate the semantic bit allocation problem as a Markov decision process (MDP) and utilize one RL agent to automatically determine the quantization parameters (QPs) for different coding units (CUs) according to the task-driven semantic fidelity metric. Extensive experiments on different tasks, such as classification, detection and segmentation, have demonstrated the superior performance of our approach by achieving an average bitrate saving of 34.39% to 52.62% over the High Efficiency Video Coding (H.265/HEVC) anchor under equivalent task-related semantic fidelity.

**Index Terms**—HEVC intra coding, task-driven semantic coding, bit allocation, reinforcement learning.

## I. INTRODUCTION

WITH the development of image/video analysis and understanding, many intelligent media applications, such as detection [1], [2], [3], [4], [5], [6], classification [7], [8], [9], [10], [11], retrieval [12], [13] and person reidentification [14], [15], [16], have been greatly promoted. These factors bring out the requirements for efficient compression of image/video signals, which can reduce the bitrate as much as possible when ensuring semantic fidelity for intelligent media applications. However, it is difficult to integrate the semantic distortion metrics directly into the traditional hybrid coding framework since the traditional hybrid coding framework cannot be optimized in an end-to-end manner. The current video coding standards are all based on the hybrid coding framework, such as Advanced Video Coding (H.264/AVC) [17], High Efficiency Video Coding (H.265/HEVC) [18] and

the recently released Versatile Video Coding (H.266/VVC) [19]. They are still widely used in the industry, and should not be replaced by learning based video coding schemes in a short time. Therefore, it is necessary to explore an efficient task-driven semantic coding method for the traditional hybrid coding framework.

In the past few years, traditional image/video coding technologies have been devoted to improving the rate-distortion performance, such as Joint Photographic Experts Group (JPEG) [20], Better Portable Graphics (BPG), H.264/AVC, H.265/HEVC, and H.266/VVC. The distortion is usually measured by Mean Square Error (MSE) [21], which can represent the pixel fidelity of an image/video. However, pixel fidelity cannot fully reflect the human perceptual viewing experience [21]. Therefore, many perceptual distortion metrics, such as structural similarity index (SSIM) [22] and multi scale structural similarity index (MS-SSIM) [23], have been proposed. To integrate perceptual distortions into traditional hybrid coding framework, the work [24] implemented perceptual coding by changing the rate-distortion optimization with SSIM. With the development of saliency detection [25], [26], many perceptual coding schemes [27], [28], [29], which implemented bit allocation based on saliency detection, have been proposed.

Unlike pixel fidelity and perceptual fidelity metrics, semantic fidelity metrics are difficult to integrate into traditional coding frameworks since the traditional hybrid coding framework cannot be optimized in an end-to-end manner. To solve this problem, a simple method is to build up an optimization schemes by heuristically adjusting image compression parameters (e.g., QP (quantization parameters)), such as [30]. However, this scheme cannot adaptively and automatically optimize coding configurations according to different semantic distortion metrics of different tasks. Recently, Bichon et al. [31] utilizes the psycho-vision guided function to weight the distortion in HEVC, which improves the subjective quality of encoded images. However, the weighting function for semantic coding is difficult to get through simple calculations. It's still a considerable challenge to implement task-driven semantic coding with the traditional hybrid coding framework.

In this paper, we attempt to achieve task-driven semantic coding within traditional codecs by solving two essential problems, which are “how to measure pixelwise semantic fidelity for different tasks” and “how to integrate the semantic fidelity metric into the in-loop optimization of semantic coding”. To solve the first problem, we utilize task-driven semantic maps, which are generated by Grad-CAM [32] and Mask R-CNN [33] to represent the pixelwise semantic importance of video/image. The semantic fidelity metric can be computed with the difference between the semantic maps before and

Xin Li, Jun Shi, and Zhibo Chen are with the Department of Electronic Engineer and Information Science, University of Science and Technology of China, Hefei, Anhui, 230026, China (e-mail: lixin666@mail.ustc.edu.cn; shi1995@mail.ustc.edu.cn; chen-zhibo@ustc.edu.cn). Xin Li and Jun Shi contribute equally to this paper. Corresponding author: Zhibo Chen.

after coding. The effectiveness of task-driven semantic maps can be seen in section IV-C. For the second problem, we implement semantic coding by formulating a semantic bit allocation scheme, i.e., deciding the quantization parameters of each coding unit(CU) as a Markovian decision process (MDP). Then, we introduce one RL agent to adaptively decide the quantization parameter (QP) for each CU by balancing the bit cost and semantic fidelity metric. According to RL-based semantic bit allocation, we can integrate the semantic fidelity metric into the in-loop optimization of semantic coding. When the training of the RL agent is completed, the whole process of the QP decision is off-policy, which can be processed in parallel with the encoder.

Since H.265/HEVC is the latest widely used video coding standard, we validate our task-driven semantic coding algorithm on H.265/HEVC in this paper, which is also easily generalized to other hybrid codecs such as H.264/AVC and H.266/VVC. To train the RL agent efficiently, we built a universal task-driven semantic coding dataset. In this dataset, the semantic importance maps are generated by Grad-CAM for classification and Mask R-CNN for segmentation and detection. The bit costs are extracted from the traditional coding framework H.265/HEVC [18]. With this dataset, the RL agent can be guided to make precise decision for task-driven semantic coding. Extensive experiments have demonstrated that our scheme can achieve an average bitrate reduction from 32.4% to 52.6% with comparable task-driven semantic fidelity.

The main contributions of our work presented in this paper can be summarized as follows:

- To the best of our knowledge, we are the first to implement task-driven semantic coding for the traditional hybrid coding framework by adaptive semantic bit allocation with reinforcement learning.
- We succeed in measuring the task-related pixelwise semantic fidelity with semantic importance map differences and integrate the semantic fidelity metric into the in-loop optimization of semantic coding.
- We create a dataset<sup>1</sup> that is suitable for the RL agent to learn to decide quantization parameters. Extensive experiments on various intelligent tasks validate that our algorithm can achieve the average bitrate reduction from 32.4% to 52.6% with comparable task-driven semantic fidelity.

The work described in this paper is related our previous work that was reported in [34]. In our previous paper, we presented the basic idea of RL-based semantic bit allocation and provide preliminary results. The difference of this paper and [34] are presented as follows. First, in this paper, we built a complete and general task-driven semantic coding framework by introducing the pixel-level semantic map, which can unify the different tasks and improve the scalability and generalization of our semantic coding scheme. Second, we validate the effectiveness of our pixelwise semantic map to represent the task-driven semantic fidelity. Third, we make a more reasonable evaluation for our task-driven semantic

coding scheme by conducting the performance evaluation in full bitrate (from low bitrate to high bitrate) space. Finally, more thoroughly experiment results and ablation studies are provided in this paper verify the effectiveness of the proposed framework.

The rest of the paper is organized as follows. In Section II, we briefly review related works of image/video compression and intelligent tasks. Section III describes our task-driven semantic coding scheme via reinforcement learning in detail. This algorithm is named RL-based semantic coding (RSC). Section IV introduces our dataset generation together with extensive experiments and ablation studies. Finally, we draw conclusions and provide directions for future work in Section V.

## II. RELATED WORK

### A. Task related Coding

In recent years, the traditional hybrid coding framework has not been able to satisfy people's needs in some situations. Therefore, many task related coding schemes, such as perceptual coding and semantic coding, have been explored [35], [36], [37], [38], [39] with the development of image/video coding techniques.

To improve the perceptual quality of image/video, Huang et al. [36] first proposed developing a method of perceptual rate-distortion optimization by applying SSIM as a quality metric in H.264, which can improve the overall perceptual quality of encoded images/videos. However, it is unable to adapt to improve the perceptual quality of regions in which people are interested. With the advance of saliency detection, some researchers have succeeded in detecting the regions of interest in images/videos. Based on saliency detection, some works [40], [41], [42], [37] succeed in improving the subjective quality of coded video by implementing visual attention guided bit allocation in video compression. In recent years, learning-based perceptual coding [43], [44] has been explored to further improve the perceptual quality of encoded images/videos.

With the development of deep learning, some deep neural network (DNN)-based image/video coding frameworks that focus on semantic information of image/video have been proposed. Torfason et al. [35] let the compressed representation reserve the semantic information by jointly training compression networks with image understanding tasks on the compressed representations. Furthermore, Luo et al. [45] proposed a deep semantic image compression (DeepSIC) model that enables the compressed code stream to carry semantic information of the image during its storage and transmission by pre-semantic analysis or post-semantic analysis. Unlike transmitting semantic information through compressed representations without decoding, Akbari et al. [46] decomposed images into thumbnails and segmentation maps. Then they transmitted the semantic information by coding the segmentation maps and reconstructed the decoded images by utilizing the semantic information, which achieves more bit saving as well by transmitting the segmentation maps. To improve the compression quality of facial compression, Chen et al.

<sup>1</sup>The dataset and code will be released at <http://staff.ustc.edu.cn/~chenzhibo/resources.html>

[38] integrated the semantic fidelity into a facial compression framework, which employed a generative adversarial network (GAN) as a metric. However, the traditional hybrid coding framework cannot be optimized in an end-to-end manner, which has been used in the above works. It is still attractive and challenging to implement task-driven semantic coding with the traditional hybrid coding framework. In this paper, we achieve task-driven semantic coding for traditional hybrid coding framework by achieving semantic bit allocation for HEVC intra coding based on reinforcement learning (RL).

### B. Related computer vision tasks

Deep learning has revolutionized many computer vision fields, especially in image/video understanding and analysis. Many intelligent applications, such as object detection, segmentation, classification, and face recognition, have been greatly developed. Here, we briefly introduce the classification, segmentation and detection.

Since AlexNet [7] won the 2012 ImageNet LSVRC championship, many deep convolution neural networks have been proposed to improve the classification accuracy, including VGGNet [8], InceptionNet [47], ResNet [9], DenseNet [10] and NASNet [11]. These networks have been applied to various computer vision tasks as backbones, including pose estimation, person re-identification and image restoration, due to their powerful feature extraction capability. Additionally, with category-independent region proposals and supervised pretraining for auxiliary tasks, R-CNN [1] made a breakthrough in the object detection field. However, the training and object detection process of R-CNN is expensive in space and time. Thus, Fast R-CNN [2] applied SPPnet [48] to speed up R-CNN by sharing a feature map instead of performing a ConvNet forward pass for each object proposal. Faster R-CNN [3] introduced a novel region proposal network (RPN) that shares full-image convolutional features with the detection network where proposal computation is nearly cost-free. Instead of adapting the mechanism of region proposals, Redmon et al. proposed a new framework named YOLO [4], which re-defines object detection as a regression problem and separates bounding-boxes regression and associated class probabilities. However, it cannot locate the object accurately. Thus, SSD [5] solved the problem by combining the compression ideas of YOLO and the anchor mechanism of Faster R-CNN.

Unlike the object detection task, which returns the coordinates of the bounding-box, segmentation aims to label every pixel in an image with its class. Early works [6], [49], [50], [51] focused on noninstance semantic segmentation, which is mainly based on bottom-up segments. However, the upper layers cannot capture rich spatial information. To refine the coarse object segments, [51] proposed a novel bottom-up/top-down architecture that combines rich spatial information and object-level knowledge to obtain more accurate segmentation. Based on the development of semantic segmentation, many instance-aware semantic segmentations [52], [53], [54], [55], [56] have been proposed that can label pixels according to not only their class but also the object to which they belong. Recently, [33] is the state-of-the-art work in instance segmentation, bounding-box object detection with ROIAlign and predicting an object

mask in parallel with bounding-box recognition. Furthermore, Huang et al. achieved more improvement with a mask scoring strategy based on Mask-RCNN [57]. In this paper, we employ the Mask-RCNN to validate the effectiveness of our algorithm on the tasks of segmentation and detection.

## III. RL-BASED SEMANTIC CODING (RSC)

In this section, we introduce our task-driven semantic coding algorithm RSC. The essence of task-driven semantic coding is to balance the semantic fidelity and coding cost. As shown in Fig. 1, we achieve task-driven semantic coding by RL-based semantic bit allocation with H.265/HEVC. The overall architecture mainly contains two parts: task-driven semantic map generation and the RL agent for semantic bit allocation. Therefore, we first formulate the semantic bit allocation based on traditional MSE-based bit allocation. Then we detail the components of RL for semantic bit allocation, including the state, action, reward and agent architecture.

### A. Semantic Bit Allocation

Bit allocation can usually be implemented on three levels: GOP level, frame level and basic coding unit level. In this article, we mainly focus on coding unit (CU)-level semantic bit allocation.

In the process of bit allocation, we usually minimize the distortion  $D$  with a given number of bits  $R_c$ , which is formulated by:

$$\min \sum D_i, s.t. \sum r_i \leq R_c, \quad (1)$$

where  $R_c$  is the upper limit of the coding bits.  $D_i$  and  $R_i$  are the distortion and coding bits for the  $i$ th basic unit. This constrained optimization problem can be converted to an unconstrained optimization problem using the Lagrange optimization method as follows:

$$\min J, J = \sum D_i + \lambda \sum R_i. \quad (2)$$

Here,  $J$  represents rate-distortion performance which is one of fundamental considerations in bit allocation. Then, we can obtain the optimal solution to formula 2 by taking the derivative of formula 2 as:

$$\lambda = -\frac{\partial D}{\partial R}, s.t. D = \sum D_i, R = \sum R_i, \quad (3)$$

where  $D$  and  $R$  represent the total distortion and total bits, respectively, for encoding one coding tree Unit(CTU). To solve the formula 3, Dai et al. [58] modeled the relationship of  $R$  and  $D$  with the hyperbolic function when the distortion is MSE as:

$$D(R) = CR^{-K}, \quad (4)$$

where  $C$  and  $K$  are model parameters, which are determined by the characteristics of the source block. Then we can obtain the slop of the R-D curve  $\lambda$  with formula 4 as:

$$\lambda = CKR^{-K-1} \triangleq \alpha R^\beta, \quad (5)$$

and simultaneously we can get  $R$  by transforming the formula 5 as:

$$R = \left( \frac{\lambda}{\alpha} \right)^{\frac{1}{\beta}} = \alpha_1 \lambda^{\beta_1}. \quad (6)$$

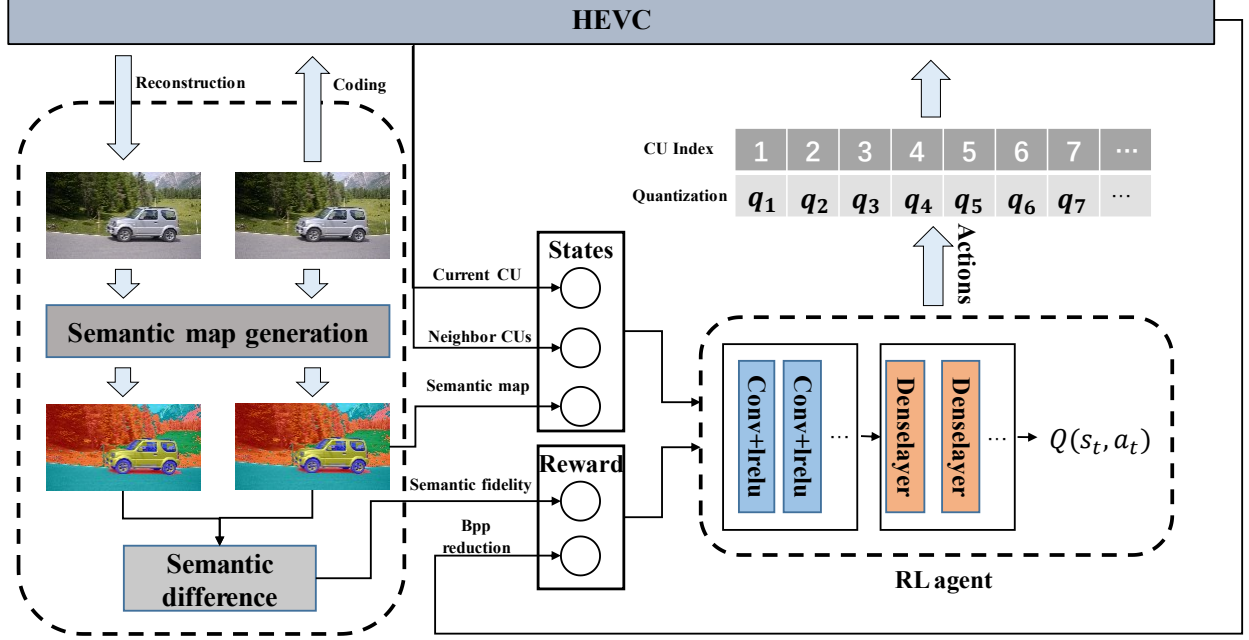


Fig. 1: Illustration of our task-driven semantic coding via reinforcement learning. It is achieved by semantic bit allocation with H.265/HEVC. Given an original image and the task, the image first passes the semantic map generation module to generate the task-driven semantic map. Then, the agent takes an action based on the observation, including the coding unit, its neighbor coding units and the semantic map. In other words, the agent selects the corresponding quantization parameters for the coding unit of the original image. Next, according to the semantic difference between the original image with the reconstructed image and bit reduction, we calculate the score as the reward to update the agent, which balances semantic fidelity and bitrate. After training, the agent can implement task-driven semantic coding with H.265/HEVC.

From formula 6, we can observe that the bits  $R$  are determined by the parameter  $\lambda$  for a certain block. Therefore, in the process of bit allocation which is based on the distortion MSE or SSIM,  $\alpha_1$  and  $\beta_1$  are usually computed by precoding the block. Then, we can perform bit allocation by computing  $\lambda$  with a given  $R$  for a coding block according to formula 5. As shown in [59] and [60], the best quantization parameter  $estQP$  corresponding to coding parameter  $\lambda$  can be computed by:

$$estQP = 4.2005 \ln \lambda + 13.7122. \quad (7)$$

However, the above bit allocation algorithm is designed for the distortion metric MSE, which cannot directly be applicable to semantic coding since the semantic distortion metric is more complex without an empirical formula. Therefore, we propose a new semantic bit allocation algorithm for semantic coding based on the above bit allocation algorithm in this section.

Based on formula 2, the semantic bit allocation can be modeled as:

$$\min J_s, J_s = \sum D_{si} + \lambda_s \sum R_i, \quad (8)$$

where  $J_s$  is semantic rate distortion for one frame.  $D_{si}$  and  $R_i$  represent the semantic distortion and encoding bits, respectively, of the  $i$ th block in one frame. And the definition of  $D_{si}$  can be seen in equation 11 of section III. B.  $\lambda_s$  is an adjustable semantic coding parameter that is responsible

for balancing  $D_{si}$  and  $R_i$ . Unfortunately, we cannot obtain the optimal solution to formula 8 directly because there is no formula that can characterize the relationship between  $D_{si}$  and  $R_i$  such as MSE-based bit allocation. Considering that the semantic rate-distortion optimization is a Markov process, which decides to increase or decrease  $R_i$  by observing the change in the results of the former state, we employ reinforcement learning (RL) to derive the optimal solution  $estR_i$ , and we express this process with:

$$estR_i = RL(\min J_s). \quad (9)$$

Actually, in the traditional hybrid coding framework, the encoding bits  $R_i$  is adjusted by the quantization parameter (QP) directly. Therefore, in this paper, we utilize RL agent to determine the best quantization parameters as equation 10.

$$estQP = RL(\min J_s). \quad (10)$$

### B. Semantic importance map generation

In the learning based coding framework, the formula 8 can be optimized easily with end-to-end training, because the semantic distortion can be substituted with the feature difference. However, it is not possible to apply this strategy to the traditional hybrid coding framework since the traditional coding framework cannot be optimized in an end-to-end manner. Thus, to optimize the formula 8, the semantic distortion

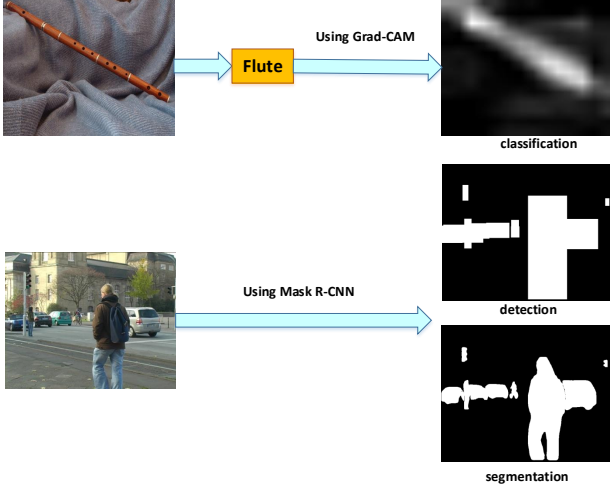


Fig. 2: Semantic importance map generation

$D_{si}$  must be expressed explicitly. Thus, we have to convert abstract semantic information to a measurable form first.

For some intelligent tasks such as classification, pose estimation, and person-reID, the outputs can accurately represent the semantic information that the tasks are concerned about. Thus, we can obtain the picture-level semantic importance map  $M_s$  by extracting semantic information from the corresponding outputs. The process of task-driven semantic importance map generation is shown in Fig. 2.

As seen in Fig. 2, the outputs of the task segmentation and detection can be mapped to the picture level easily. Thus, we employ Mask-RCNN to obtain outputs of detection and segmentation and directly use the outputs as task-driven semantic importance maps  $M_s$ . However, the results of the classification task cannot be directly mapped to  $M_s$  because the output is only a label. Inspired by [32], we implement Grad-Cam, which uses the gradient backpropagation to flow into the final convolution layer of the CNN model to produce a localization map that highlights the semantic importance regions for predicting the concept. The specific CNN model we adopt is VGG-19. Then we can obtain the semantic importance maps of frames for the classification task by Grad-Cam.

With the task-driven semantic map, we succeed in converting abstract semantic information to a measurable form  $M_s$ . The amount of semantic information is indicated by the pixel value of  $M_s$ . Then we can measure the task-driven semantic distortion by:

$$D_{si} = \Delta M_s. \quad (11)$$

Here,  $\Delta M_s$  represents the difference in the semantic importance map before and after coding. And the difference is calculated with L1 normalization. Finally, the RL agents can obtain the optimal solution to formula 8 with the measurable semantic distortion  $D_{si}$ .

### C. Reinforcement Learning for Semantic Bit Allocation

After obtaining the  $M_s$  of one frame, a simple bit allocation method is to heuristically increase the bitrate for the highly

weighted area, such as the threshold scheme. However, these heuristic methods can hardly obtain the optimal results of formula 8 and may introduce limitations because the results rely heavily on the handcrafted designs. Recently, RL has achieved outstanding performance in many tasks, especially in unsupervised or semisupervised scenarios. It has also been used in many approaches [61], [62], [63], [64] to optimize the traditional hybrid coding framework. In this paper, we adopt the reinforcement learning algorithm, Deep Q-Learning (DQN) [65], to solve the semantic bit allocation problem.

The details are as follows:

1) *Problem formulation*: to obtain the optimal  $est\lambda_i$  and corresponding  $estQP_{si}$  for each block, we have to obtain the optimal solution  $R_i$  to the formula 8. We formulate this process as an MDP process in this paper, which includes five elements: state, action, reward, state transition probability and policy(agent). Then, an agent is designed to observe the states from the environment, and executes a series of actions to optimize the formula 8. In this process, the state transition probability  $P_{sa}$  is 1 because the environment is deterministic. The remaining elements are detailed below:

2) *State*: To determine the action for the next step, the agent must observe the states of the coding block and global information of the frame. Therefore, the coding block states are sent to the agent according to the encode order, *i.e.* from left to right and from top to bottom. Here, the state of the coding block consists of the luminance, semantic importance map of the current block and a 15-d feature vector that can reflect the global information of the frame. The details of the feature vector are shown in Table. I .

TABLE I: Components of global vector

Index	Vector components
1	Number of overall CUs
2	Index of current CU
3	Mask ratio of current CU
4-7	Mask ratio of neighboring CUs
8	Mask ratio of overall frame
9	Instance number of current CU
10-13	Instance number of neighboring CUs
4-15	QPs of left and above CUs

3) *Action*: To obtain optimal coding bits  $estQP_i$  in formula 8, we have to take the action to change the coding bits  $R_i$ . In traditional coding framework such as H.265/HEVC, the coding bits and coding quality usually change by directly adjusting the quantization Parameter (QP). Lower QP leads to a higher bitrate and less distortion. From formula 7, we can find that the corresponding coding parameter  $\lambda$  can be computed by:

$$\lambda = \exp \left\{ \frac{QP - 13.7122}{4.2005} \right\}. \quad (12)$$

Thus, to meet the above coding method and reduce action space complexity, we take the action by optimally selecting the QP value. Specifically, the action space contains the QP values from 22 to 51. For one CTU, the agent can choose the best strategy to determine the optimal Quantization Parameter (QP) to encode this CTU according to the observation.

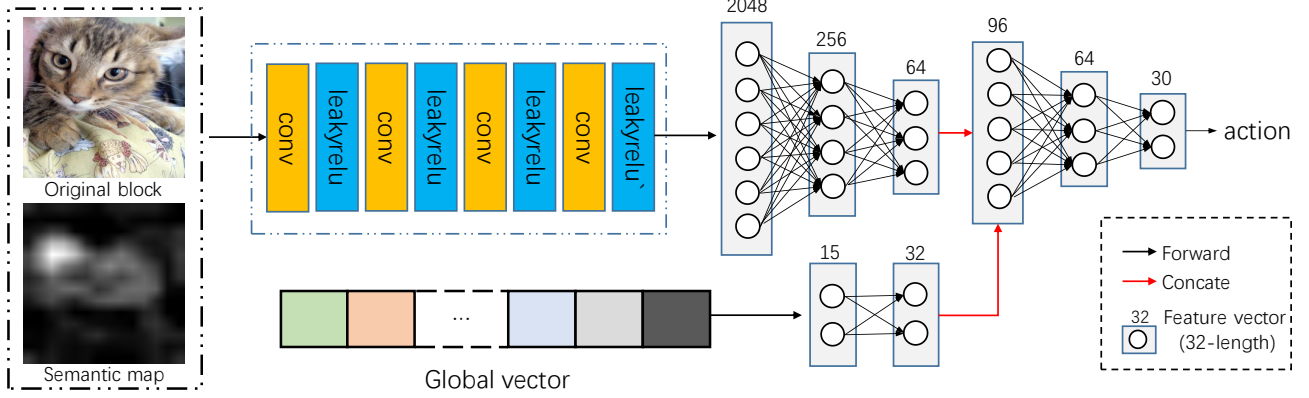


Fig. 3: Structure of the proposed Q-network. There are two input branches: the current block part and the global information part. The luminance and importance map of current blocks are concatenated in depth to represent local information

4) *Reward*: As a method for evaluating the action, the cumulative reward is the optimization goal of MDP. The agent is guided to learn the optimal action by observing the reward. In this paper, the formula 8 is our optimization goal in the semantic bit allocation process. When we set the base action as 22, which can be adjusted according to different tasks, the corresponding rate distortion  $J_s^{22}$  can be expressed by:

$$J_s^{22} = \sum D_{si}^{22} + \lambda_s \sum R_i^{22}, \quad (13)$$

where  $D_{si}^{22}$  and  $R_i^{22}$  are the semantic distortion and coding bits, respectively, when coding the frame with QP 22. Then the convex optimization problem formula 8 can be transformed to:

$$\begin{aligned} & \max \Delta J_s, \\ & \Delta J_s = J_s^{22} - J_s \\ & = \sum (D_{si}^{22} - D_{si}) + \lambda_s \sum (R_i^{22} - R_i), \quad (14) \\ & = - \sum \Delta M_s^a + \lambda_s \sum \Delta R_i \\ & = - \sum \Delta M_s^a + \lambda_s N \sum \Delta Bpp_i \end{aligned}$$

where  $\Delta M_s^a$ ,  $\Delta R_i$  and  $\Delta Bpp_i$  represent the difference in semantic information, coding bits and coding bits per pixel (bpp), respectively, between the coding block with QP 22 and the coding block after taking the action.  $N$  is the number of pixels in one coding block. This formula 14 is equivalent to:

$$\max \text{Reward} = \sum \Delta Bpp_i - \alpha_s \sum \Delta M_s^a, \alpha_s = \frac{1}{\lambda_s N}. \quad (15)$$

In the above formula, *Reward* is the reward of the MDP, which instructs the agent to take the action. The  $\alpha$  is an adjustable parameter that is related to the semantic coding parameters  $\lambda_s$  and  $N$ . Therefore, we can optimize different semantic coding models by adjusting the parameter  $\alpha$ .

5) *Agent*: In this paper, the agent is used to predict the optimal action for every block with a Q-network. Taking the state  $s_t$  as input, the Q-network outputs the decayed

cumulative reward (Q-value) of each action  $a$  as  $Q(s_t, a)$ . Then, we can obtain the optimal action  $a_t^*$  with:

$$a_t^* = \arg \max_a Q(s_t, a). \quad (16)$$

For the Q-network structure, we have two input branches: the current block part and the global feature vector. The block information flows through four convolution layers and two concatenate the features with the global feature vector after ascending the dimension together. The combination of these features can help the agent to better understand the environment. Next, the overall features flow through two fully connected layers, including one hidden layer and one output layer. All convolution layers and hidden fully connected layers are activated with a leaky rectified linear unit with  $\alpha=0.25$ , while the output layer is not activated. Fig. 3 shows the details of the proposed Q-network.

## IV. EXPERIMENTS

### A. Experimental setting

1) *Datasets*: As we employ the RL agent as the bit allocation predictor, we need a large quantity of training data. Therefore, we built a universe dataset for task-driven semantic coding, namely, the TSC dataset, which include three tasks: classification, detection, and segmentation. For classification, we selected 2,300 high-resolution images from the ImageNet [66] test set. Then, we resized them to 576x576. For detection and segmentation, we collect images from the TUD-Brussels pedestrian dataset [67], including 1,000 images.

Then, all images are encoded with the H.265/HEVC reference software HM16.19<sup>2</sup>, while the QPs from 22 to 51 are applied for encoding. The interval of QPs for coding can be adjusted according to the semantic tasks. During encoding, we collected the bit cost of each CU from HM16.19. After encoding, we obtain the reconstructions of all images, which

<sup>2</sup>Available: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-16.19/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.19/)



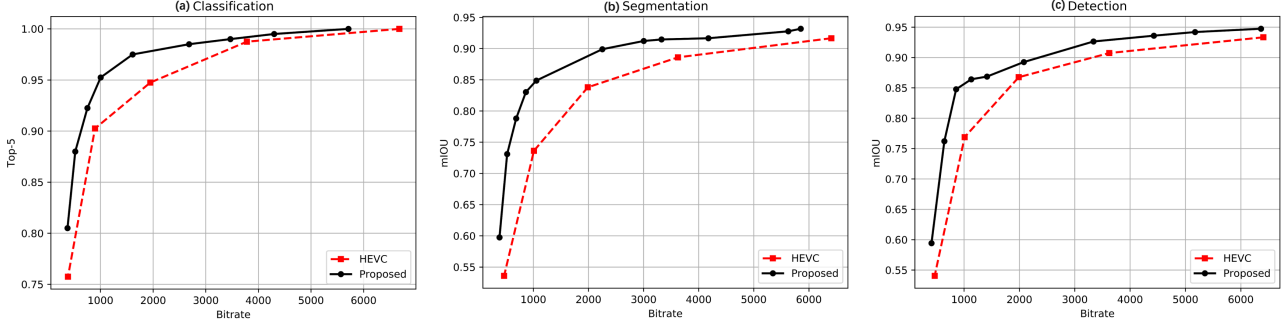


Fig. 4: Comparisons of classification, segmentation and detection between the proposed RSC algorithm and H.265/HEVC.

---

**Algorithm 1** Training process

---

**Input:** Original images  $I_0$  and their semantic maps  $M_s^0$   
**Output:** Well-trained RL agent

- 1: Initialize the network parameters  $\theta$
- 2: **for** Frame  $f = 0$  to  $M-1$  **do**
- 3:   **for** CU  $t = 0$  to  $N-1$  **do**
- 4:     Set state  $s_t$
- 5:     Choose action  $a_t = \arg \max_a Q(s_t, a)$
- 6:     Encode the CU with quantization parameter  $QP = a_t + 22$
- 7:     Obtain  $bpp$  and corresponding  $M_s$
- 8:     Compute reward  $r_{t+1}$  as Equ. 15
- 9:     Observe reward  $r_{t+1}$  and next state  $s_{t+1}$
- 10:    Store transition  $(s_t, a_t, r_{t+1}, s_{t+1})$
- 11:    Sample a mini-batch of transitions from buffer  $(s_{t'}, a_{t'}, r_{t'+1}, s_{t'+1})$
- 12:    Compute  $y_{t'} = r_{t'+1} + \gamma \max_{a_{t'+1}} Q(s_{t'+1}, a_{t'+1}; \theta)$
- 13:    Update  $\theta$  with the loss  $\sum_{t'} [y_{t'} - Q(s_{t'}, a_{t'}; \theta)]^2$
- 14:   **end for**
- 15: **end for**

---



---

**Algorithm 2** Process of RSC

---

**Input:** Original image  $I_0$   
**Output:** Bit stream and reconstructed image  $I_c$

- 1: Generate the semantic map  $M_s^0$  for  $I_0$
- 2: Obtain the QPs for each CU from the RL agent
- 3: Input the QPs for each CU to the HM16.19 and compress the original image  $I_0$
- 4: Obtain the bit stream and reconstructed image  $I_c$

---

are sent to the vision task CNNs to obtain the corresponding semantic importance map. We use Grad-Cam for classification and Mask R-CNN for detection and segmentation. Finally, the TSC database is obtained, which is randomly divided into training (80%) and test (20%) sets.

2) *Training settings:* In this paper, we apply H.265/HEVC reference software HM16.19 as our codec. We compress our data with an all-intra main configuration. For DQN, the parameters are randomly initialized. We set the learning rate

as 0.0001 and the discount factor as 0.9. The memory size is set to 50,000 and the mini-batch is set as 64. The target network parameters are updated every 300 steps. We utilize the TensorFlow framework to implement the whole model. It only takes 24 hours to train the DQN and obtain the best performance. The training process is as algorithm 1.

3) *Evaluation Standard:* To evaluate the effectiveness of our RSC, the average semantic fidelity and average bitrate are measured. To measure the task-driven semantic fidelity, we measure the top-5 accuracy for classification and the mean intersection of union (mIoU) for segmentation and detection.

### B. Compared with traditional codecs

We compare our RSC with traditional codec H.265/HEVC. For H.265/HEVC, a larger quantization parameter (QP) means a higher compression ratio. According to experiments, we find that the top-5 accuracy for classification is 100% when QP is lower than 27. Here, we select the QP as 27, 32, 37, 42, 47 for H.265/HEVC as our baseline. To validate the effectiveness of our task-driven semantic coding, we apply our algorithm in three tasks: classification, segmentation and detection. The process of RSC is shown in algorithm 2.

1) *Classification:* For classification, we set  $\alpha_s$  in equation 15 as 0.005, 0.01, 0.025, 0.05, 0.25, 1, 2, 4 and 8, respectively. The RL agent focuses more on semantic fidelity when  $\alpha_s$  becomes larger. As shown in Fig. 4(a), when  $\alpha_s$  increases, the compression ratio decreases and semantic fidelity increases. Moreover, the proposed semantic coding method can obtain better classification accuracy compared with H.265/HEVC under the same bit cost. To better learn about how task-driven semantic coding works, we visualize the semantic map for classification as shown in Fig. 5(b). The bright region represents the semantically important part of the image. According to the semantic map, the heads of rabbits, eyes of cats and cabin are essential for classification to make decisions. As shown in Fig. 5(c), our methods can better preserve the semantic fidelity at the above semantically important region. However, traditional codec wastes approximately 50% bits on semantically unimportant regions.

2) *Segmentation and Detection:* For different tasks, the measurements of semantic fidelity are different. Thus,  $\alpha_s$  is

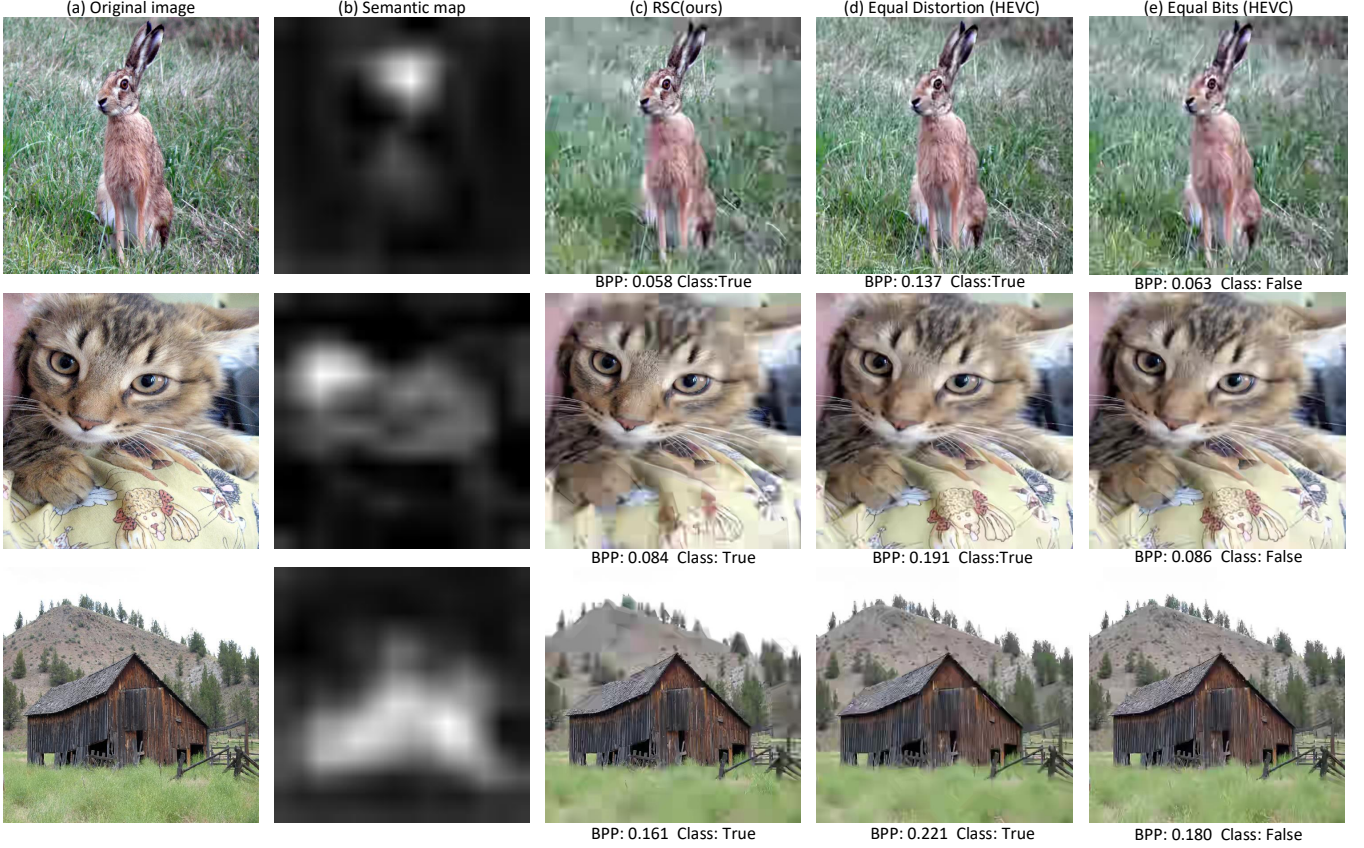


Fig. 5: Examples of task-driven semantic coding on classification. From left to right, the images are the original image, semantic map, image coded with our RSC algorithm, images coded with equivalent semantic distortion and equivalent bits.

also different. We empirically set the  $\alpha_s$  as 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 2, 4, 8 and 16 for segmentation and set  $\alpha_s$  as 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 2, 4 and 8 for segmentation. As shown in Fig. 4(b) and (c), for segmentation, when QP in traditional coding is set as 27, the coding bitrate is approximately 6,400 kpbs/s. With the same mIOU, we can save approximately 50% of the bits. For detection, when QP in traditional coding is set as 27, we can also save approximately 40% of the bits, which is significant for data transmission and storage. The coded images are visualized as Fig. 6.

3) *BD-BR and BD-metric*: In this section, we utilize Bjontegaard metric [68] as our evaluation method for semantic coding scheme. Specifically, we compute the BD-BR and BD-metric [69] for our RSC algorithm and utilize HM16.19 as our baseline. The BD-BR represents the average bit-rate reduction under the equivalent task-related accuracy, and the BD-metric represents the average task-related accuracy improvement under the equivalent bit-rate. The metrics for classification, segmentation and detection are Top-5 accuracy and mIOU, respectively. As shown in Table. II, with the same semantic fidelity, our RSC algorithm can achieve 52.62%, 51.01% and 34.39% bitrate savings on classification, segmentation and detection tasks, respectively, compared with HM16.19. Under the same bit cost, our RSC algorithm can improve the accuracy by 2.38% on the classification task and the mIOU

by 5.02% and 3.11%, respectively, on the segmentation task and detection task.

TABLE II: BD-BR and BD-metric relative to the baseline in HM16.19.

Tasks	Classification	Segmentation	Detection
BD-BR	-52.62%	-51.01%	-34.39%
BD-metric	2.38%	5.02%	3.11%

### C. Analysis of the semantic map

In this section, we analyze the relationship between semantic map differences and semantic fidelity. To implement the semantic rate-distortion optimization, we represent the task-driven semantic fidelity with semantic map difference  $\Delta M_s$ , which leads the RL agent to balance the semantic fidelity and bit-rate. To validate its effectiveness, we allocate bits by selecting four QP values: 22, 27, 32, and 37. Then, we set these QPs as centers and randomly adapt the QP in the interval of plus or minus 5 for each CTU in one frame. The semantic importance is computed for the whole frame. We compute the classification accuracy and mIOU as our semantic fidelity for classification, segmentation and detection. The relationship of semantic map differences and semantic fidelity is shown in Fig. 7. According to Fig. 7, the semantic fidelity and





Fig. 6: Examples of task-driven semantic coding on (1) detection and (2) segmentation. The first column is the original image before coding. The second column is coded with our proposed RSC algorithm. The third and fourth columns are coded with H.265/HEVC. The third column aims to compare the bits cost under the similar distortion, and the fourth column aims to compare the distortion under similar bits cost. Higher IOU means the better performance.

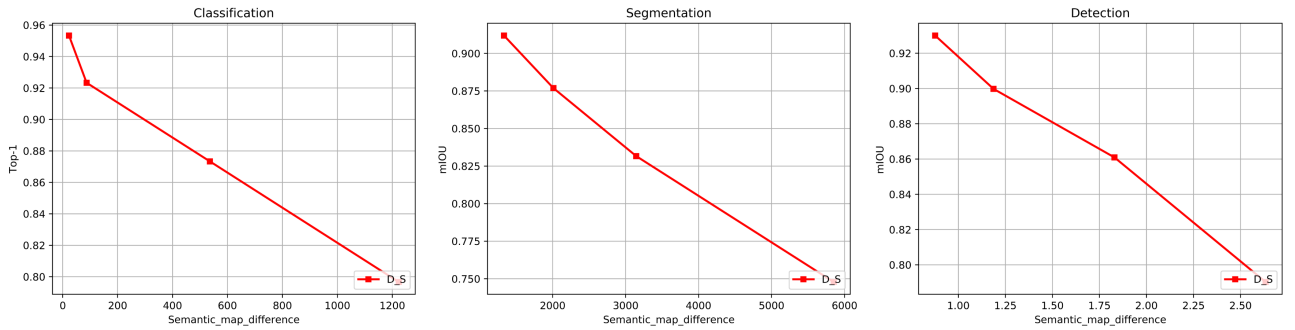


Fig. 7: The relationship between semantic fidelity and the semantic map difference for classification, segmentation and detection. Semantic fidelity is negatively correlated with semantic map differences.

semantic map difference are almost linear, which validates the effectiveness of our semantic map. Since the QP for each CTU is randomly set, the total semantic map differences of all CTUs are consistent with the semantic fidelity of the whole frame.

#### D. Ablation Study

1) *Comparison with handcrafted scheme:* To compare our method with the handcrafted scheme, after extracting the semantic map, we directly set the QPs for CUs of coding images instead of utilizing the RL agent to make a decision for classification. Specifically, we set larger QPs for semantically important regions and lower QPs for semantically unimportant regions. We set the QP from 22 to 51, which is the same as our action space. For a CTU of size 64x64, we first calculate the semantic importance by summing the corresponding semantic map as Eq. 17 and then normalize the S value.

$$S = \frac{\sum_{x \in X} \sum_{y \in Y} M_s}{XY}, S = \frac{S}{S_{MAX}}. \quad (17)$$

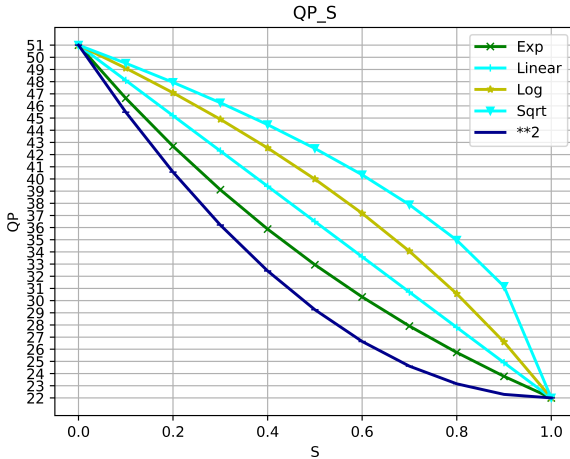


Fig. 8: The method for calculating the QPs according to the semantic maps.

Then, we set the QP values with linear mapping or non-linear mapping as shown in Fig. 8 according to the semantic importance  $S$  since the relationship  $S$  value and best QP value might be nonlinear. As shown in Fig. 9, our algorithm is better than handcrafted schemes. There are two key reasons. First, handcrafted methods cannot adaptively adjust the QP value according to semantic importance because they cannot capture the best relationship between the QP value and semantic map. Second, it cannot balance the bitrate and semantic fidelity because the bitrate cannot be computed before coding. However, the RL agent can capture the above information with a learnable mechanism. With little training data, our method can adapt to the task and achieve state-of-the-art semantic coding performance by utilizing semantic bit allocation.

2) *Effects of global information for Q-network:* Since the quantization parameter (QP) determination of each block is not only associated with current block, but also associated

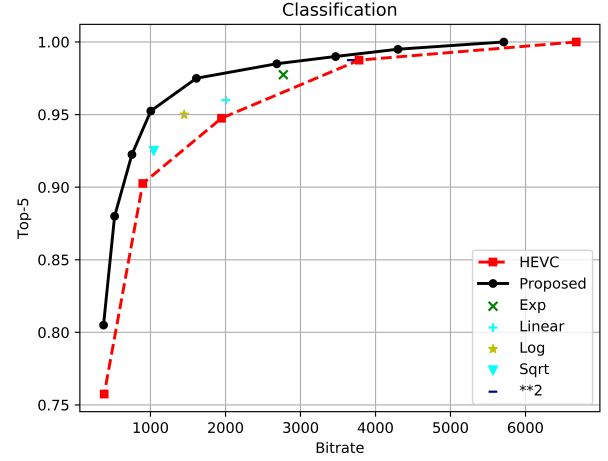


Fig. 9: Comparison between our algorithm and handcrafted schemes.

with the state of whole coding image, our designed Q-network contains two branches, capturing global information and local information, respectively. The local information contains the current block and its semantic map, which is necessary to provide task-related information to the network. To validate the effectiveness of global information, we remove the branch used to capture the global information from our Q-network. And we select classification task to conduct experiments. The experimental results are as shown in Fig. 10. The removing of global information causes obvious performance drop for our Q-network, which validate the effectiveness of our two-branches for Q-network.

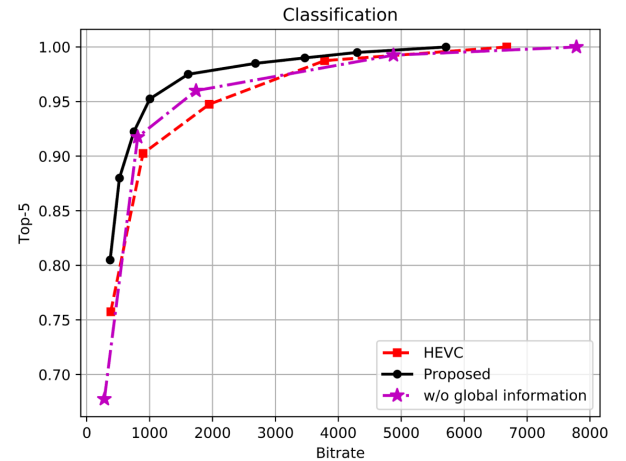


Fig. 10: Effects of global information for Q-network.

#### E. Complexity Analysis

In this section, we analyze the complexity of our algorithm. As shown in Table. III, our algorithm does not change the decoding time. Moreover, the encoding time is almost the

same as H.265/HEVC. The QP decision takes approximately 0.25s and semantic map generation only takes about 0.45s for one frame of size 576x576 when running with NVIDIA 2080Ti. It is efficient and effective to integrate our algorithm into intelligent media applications. The speed of QP decision and semantic map generation is not optimized in this work, which can be further optimized in future work.

TABLE III: Time complexity

Run time	Encoder		Decoder
Proposed	RL agent	0.25s	0.036s
	Semantic map generation	0.45s	
	coding	1.735s	
HEVC	1.735s		0.036 s

#### F. Comparisons of our RSC with its conference version

In this section, we clearly clarify the difference between this paper and its conference version [34], which can be summarized into following three parts.

1) *Scalability and generalization*: This paper builds a complete and general task-driven semantic coding framework by introducing a pixel-level semantic map. For the conference version [34], the different tasks have different semantic maps, which causes the inputs of RL network and reward definition are different. For example, the semantic map of detection task in the conference version [34] is the distribution of instance number and the semantic map of classification task is pixelwise importance map. Therefore, the conference version [34] lacks of scalability and generalization for different tasks. To overcome these issues, we introduce a pixel-wise semantic map to unify the different tasks in this journal paper. In this way, all tasks share the same RL architecture as shown in Fig. 3 and reward definition as Equ. 15, which promises the scalability and generalization of our RSC (i.e, when meeting new task, the RL architecture and reward definition do not need any modification). To prove that, we utilize the RL agent trained only on classification task to determine the QPs for segmentation and detection tasks. As shown in the Fig. 11, our RL for classification can also achieve the considerate performance in detection and segmentation tasks, which validates the generalization of our semantic coding scheme in this paper.

Moreover, the pixelwise semantic map can bring considerate improvement on detection task compared to the conference version [34]. The rate-distortion comparison on detection task of our RSC and its conference version [34] is shown in Fig. 12. With the same semantic distortion, our RSC can save 8.81% higher than its conference version [34], which is shown in TABLE IV. We also additionally add the subjective comparison of our RSC with its conference version [34] as shown in Fig. 13.

TABLE IV: BD-BR and BD-metric of our RSC and its conference version [34] with the baseline HM16.19.

Tasks	RSC	Conference version [34]
BD-BR	-34.39%	-25.58%
BD-metric	3.11%	2.53%

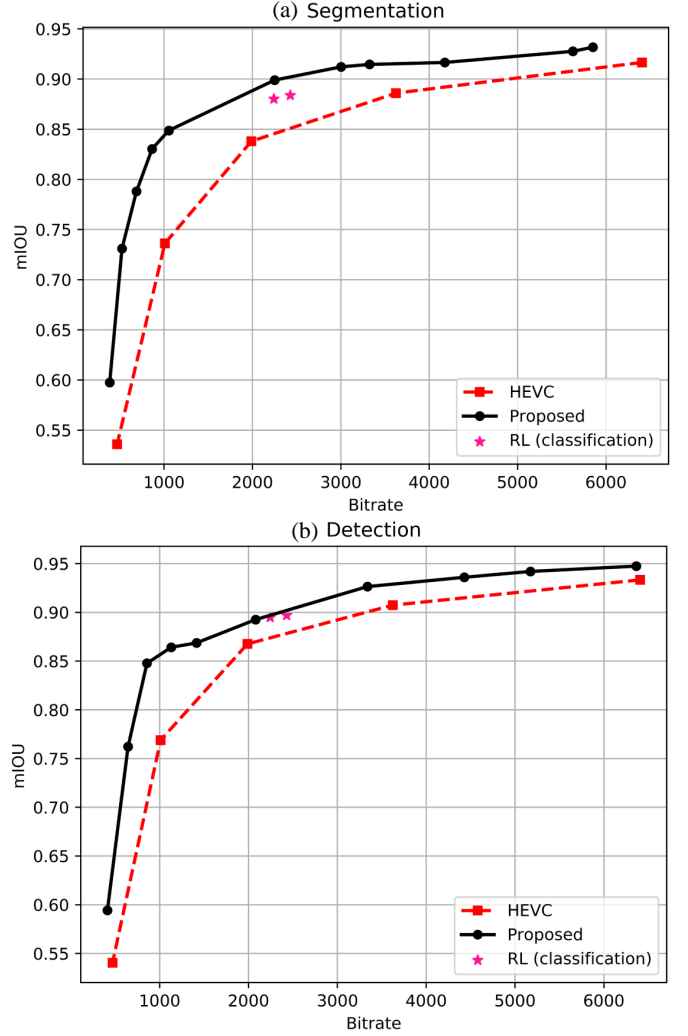


Fig. 11: Scalability and generalization from classification task to segmentation and detection tasks. RL (classification) means that we utilize the RL agent trained only on classification task to determine the QPs for segmentation and detection.

2) *Evaluation methods for task-driven semantic coding*: The conference version [34] of our RSC only works and is evaluated at one bitrate point. Moreover, we only compare our scheme with HEVC on QP 22 in the conference version, which is limited. To overcome these limitations, in this paper, we provide the formula 15 and change the bitrate of our RSC for coding by adjusting the parameter  $\alpha$ , that represents the importance degrees of semantic distortion. In this way, our RSC can be suitable for large range of bitrate. As shown in Fig. 4, our RSC can achieves considerate performance improvement from low to high bitrate. To comprehensively evaluate our RSC, we also compute the BD-BR and BD-metric as shown in table II to demonstrate the superiority of our RSC in this paper.

3) *Theory and experimental analysis*: In our conference version [34], the reward definition is not clear and lacks of reasonable explanation. Besides, the effectiveness of importance map is not validated. Therefore, in this paper, we design the



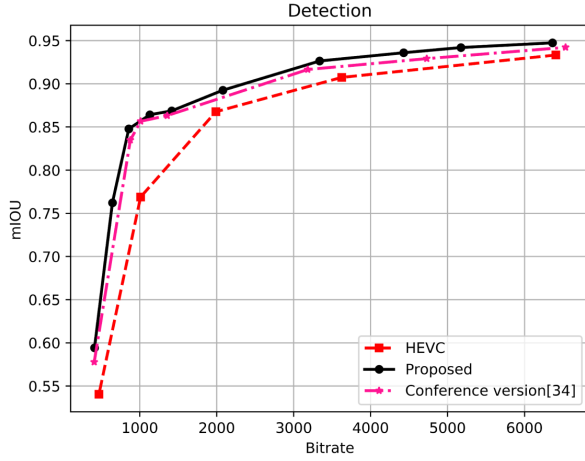


Fig. 12: Comparison between our RSC with its conference version [34].

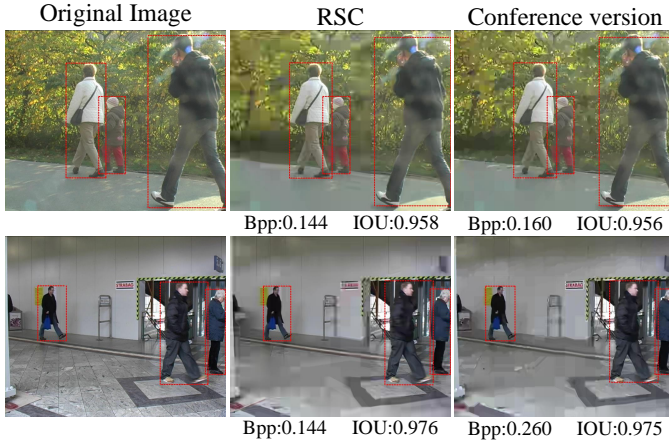


Fig. 13: Comparison between our RSC with its conference version [34].

reward with detailed theory derivation as formula 15. Moreover, we validate the effectiveness of our pixelwise semantic map to represent the task-driven semantic fidelity as shown in Fig. 7. In addition, we provide the thorough experimental analysis for ablation study, complexity and stability.

## V. CONCLUSION

In this paper, we first implement task-driven semantic coding for the traditional hybrid coding framework, which utilizes RL-based semantic bit allocation. Specifically, we design semantic maps for different tasks to extract the pixelwise semantic fidelity. Then, we utilize reinforcement learning (RL) to integrate the semantic fidelity metric into the in-loop optimization of semantic coding. Extensive experiments demonstrate the effectiveness of our algorithm. Our method can save 34.39% to 52.62% bits over the traditional coding framework with comparable semantic fidelity in different tasks such as classification, segmentation and detection. By designing the task-driven semantic map, our algorithm can be

extended to other intelligent media applications easily without modifying the network for specific tasks. In future work, we will consider to extend our scheme to support heterogeneous intelligent video tasks.

## REFERENCES

- [1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [2] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [3] 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*, 2015, pp. 91–99.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [9] 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*, 2016, pp. 770–778.
- [10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [11] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [12] R. R. Saritha, V. Paul, and P. G. Kumar, "Content based image retrieval using deep learning process," *Cluster Computing*, vol. 22, no. 2, pp. 4187–4200, 2019.
- [13] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk, "Solar: Second-order loss and attention for image retrieval," *arXiv preprint arXiv:2001.08972*, 2020.
- [14] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification," *arXiv preprint arXiv:2003.12224*, 2020.
- [15] X. Jin, C. Lan, W. Zeng, G. Wei, and Z. Chen, "Semantics-aligned representation learning for person re-identification," *arXiv preprint arXiv:1905.13143*, 2019.
- [16] X. Jin, C. Lan, W. Zeng, Z. Chen, and L. Zhang, "Style normalization and restitution for generalizable person re-identification," in *CVPR*, 2020.
- [17] 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.
- [18] 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.
- [19] Joint Video Experts Team (JVET), "VTM Software," available: [https://vcgit.hhi.fraunhofer.de/jvet/VVCSSoftware\\_VTM/](https://vcgit.hhi.fraunhofer.de/jvet/VVCSSoftware_VTM/), 2020, [Accessed 23-Feb.-2020].
- [20] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [21] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [22] 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.



- [23] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, vol. 2. Ieee, 2003, pp. 1398–1402.
- [24] T.-S. Ou, Y.-H. Huang, and H. H. Chen, "Ssim-based perceptual rate control for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 682–691, 2011.
- [25] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proceedings of the IEEE International Conference on computer vision*, 2017, pp. 212–221.
- [26] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *European conference on computer vision*. Springer, 2016, pp. 825–841.
- [27] W. Gao and S. Kwong, "Phase congruency based edge saliency detection and rate control for perceptual image and video coding," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2016, pp. 000 264–000 269.
- [28] S. Li, M. Xu, Y. Ren, and Z. Wang, "Closed-form optimization on saliency-guided image compression for hevc-msp," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 155–170, 2017.
- [29] C. Ku, G. Xiang, F. Qi, W. Yan, Y. Li, and X. Xie, "Bit allocation based on visual saliency in hevc," in *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2019, pp. 1–4.
- [30] D. Liu, D. Wang, and H. Li, "Recognizable or not: Towards image semantic quality assessment for compression," *Sensing and Imaging*, vol. 18, no. 1, p. 1, 2017.
- [31] M. Bichon, J. Le Tanou, M. Ropert, W. Hamidouche, and L. Morin, "Optimal adaptive quantization based on temporal distortion propagation model for hevc," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5419–5434, 2019.
- [32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [33] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [34] J. Shi and Z. Chen, "Reinforced bit allocation under task-driven semantic distortion metrics," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.
- [35] R. Torfason, F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Towards image understanding from deep compression without decoding," *arXiv preprint arXiv:1803.06131*, 2018.
- [36] Y.-H. Huang, T.-S. Ou, P.-Y. Su, and H. H. Chen, "Perceptual rate-distortion optimization using structural similarity index as quality metric," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1614–1624, 2010.
- [37] H. Hadizadeh and I. V. Bajić, "Saliency-preserving video compression," in *2011 IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–6.
- [38] Z. Chen and T. He, "Learning based facial image compression with semantic fidelity metric," *Neurocomputing*, vol. 338, pp. 16–25, 2019.
- [39] M. Zhou, X. Wei, S. Kwong, W. Jia, and B. Fang, "Just noticeable distortion-based perceptual rate control in hevc," *IEEE Transactions on Image Processing*, 2020.
- [40] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE transactions on image processing*, vol. 19, no. 1, pp. 185–198, 2009.
- [41] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image and Vision Computing*, vol. 29, no. 1, pp. 1–14, 2011.
- [42] M. T. Khanna, K. Rai, S. Chaudhury, and B. Lall, "Perceptual depth preserving saliency based image compression," in *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*, 2015, pp. 218–223.
- [43] S. Ki, S.-H. Bae, M. Kim, and H. Ko, "Learning-based just-noticeable-quantization-distortion modeling for perceptual video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3178–3193, 2018.
- [44] Z. Cheng, P. Akyazi, H. Sun, J. Katto, and T. Ebrahimi, "Perceptual quality study on deep learning based image compression," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 719–723.
- [45] S. Luo, Y. Yang, Y. Yin, C. Shen, Y. Zhao, and M. Song, "Deep semantic image compression," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 96–106.
- [46] M. Akbari, J. Liang, and J. Han, "Dsslic: deep semantic segmentation-based layered image compression," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2042–2046.
- [47] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [49] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.
- [50] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3992–4000.
- [51] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.
- [52] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.
- [53] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.
- [54] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5008–5017.
- [55] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5221–5229.
- [56] A. Arnab and P. H. Torr, "Pixelwise instance segmentation with a dynamically instantiated network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 441–450.
- [57] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418.
- [58] M. Dai, D. Loguinov, and H. Radha, "Rate-distortion modeling of scalable video coders," in *2004 International Conference on Image Processing, 2004. ICIP'04.*, vol. 2. IEEE, 2004, pp. 1093–1096.
- [59] S. Li, M. Xu, Z. Wang, and X. Sun, "Optimal bit allocation for ctu level rate control in hevc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2409–2424, 2016.
- [60] B. Li, H. Li, L. Li, and J. Zhang, "λ domain rate control algorithm for high efficiency video coding," *IEEE transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, 2014.
- [61] N. Li, Y. Zhang, L. Zhu, W. Luo, and S. Kwong, "Reinforcement learning based coding unit early termination algorithm for high efficiency video coding," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 276–286, 2019.
- [62] L.-C. Chen, J.-H. Hu, and W.-H. Peng, "Reinforcement learning for hevc/h. 265 frame-level bit allocation," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. IEEE, 2018, pp. 1–5.
- [63] J.-H. Hu, W.-H. Peng, and C.-H. Chung, "Reinforcement learning for hevc/h. 265 intra-frame rate control," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.
- [64] P. Helle, H. Schwarz, T. Wiegand, and K.-R. Müller, "Reinforcement learning for video encoder control in hevc," in *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2017, pp. 1–5.
- [65] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [67] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 794–801.
- [68] G. Bjøntegaard, "Calculation of average psnr differences between rd-curves, document rec," *ITU-T VCEG-M33, Austin, TX, USA*, 2001.
- [69] S. Pateux and J. Jung, "An excel add-in for computing bjontegaard metric and its evolution," *ITU-T SG16 Q*, vol. 6, p. 7, 2007.