International Journal of Computer Vision (2021) 129:2889–2906
https://doi.org/10.1007/s11263-021-01505-4

Just Recognizable Distortion for Machine Vision Oriented Image and Video Coding

Qi Zhang[1] · Shanshe Wang[1] · Xinfeng Zhang[2] · Siwei Ma[1] · Wen Gao[1,3]

Received: 15 December 2020 / Accepted: 12 July 2021 / Published online: 13 August 2021

Abstract
Machine visual intelligence has exploded in recent years. Large-scale, high-quality image and video datasets significantly empower learning-based machine vision models, especially deep-learning models. However, images and videos are usually compressed before being analyzed in practical situations where transmission or storage is limited, leading to a noticeable performance loss of vision models. In this work, we broadly investigate the impact on the performance of machine vision from image and video coding. Based on the investigation, we propose Just Recognizable Distortion (JRD) to present the maximum distortion caused by data compression that will reduce the machine vision model performance to an unacceptable level. A large-scale JRD-annotated dataset containing over 340,000 images is built for various machine vision tasks, where the factors for different JRDs are studied. Furthermore, an ensemble-learning-based framework is established to predict the JRDs for diverse vision tasks under few- and non-reference conditions, which consists of multiple binary classifiers to improve the prediction accuracy. Experiments prove the effectiveness of the proposed JRD-guided image and video coding to significantly improve compression and machine vision performance. Applying predicted JRD is able to achieve remarkably better machine vision task accuracy and save a large number of bits.

Keywords Image and video coding · Machine vision · Deep learning · Just noticeable distortion

1 Introduction

Human beings enjoy high quality images and videos. In order to retrieve the best visual quality, the original images caught from camera lens or created by professional tools should be directly delivered to consumers without losing any signals. But in the real world, due to limitations on transmission or storage, the pristine images and videos usually need to be compressed before spread. Since there are enormous spatial, temporal and entropic redundancies in original signals, it is possible to compress images and videos without influencing visual quality significantly. Numerous methods have been proposed to compress images and videos effectively and efficiently. These techniques are further collected and unified to establish several coding standards, such as JPEG 2K (Skodras et al. 2001), AVC (Wiegand et al. 2003), HEVC (Sullivan et al. 2012), VVC (Bross et al. 2018), AV1 (Chen et al. 2018) and AVS3 (Zhang et al. 2019).

Despite only being viewed by humans, a trend that more images and videos are used for machine visual analysis has been witnessed in recent years. Unlike the human visual system (HVS), machines are usually more interested in the semantic information of an image rather than textures and details, because they are expected to understand the content rather than just viewing it. The information and knowledge of an image are able to be revealed by some high-level image

Communicated by Dong Xu.

✉ Qi Zhang
ywwynm@pku.edu.cn

Shanshe Wang
sswang@pku.edu.cn

Xinfeng Zhang
xfzhang@ucas.ac.cn

Siwei Ma
swma@pku.edu.cn

Wen Gao
wgao@pku.edu.cn

[1] National Engineering Laboratory for Video Technology, Peking University, Beijing, China

[2] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

[3] Peng Cheng Laboratory, Shenzhen, Guangdong Province, China

features, which can be extracted precisely using the modern deep-learning techniques, more specifically, convolutional neural networks (CNN). Therefore, CNN-based machine vision models have dominated many important visual tasks, such as image classification and object detection, even exceed human beings on the performance.

Because of the effectiveness to extract representative and distinguishable image features, several CNNs are applied as the backbone structure to support higher-level machine vision tasks, such as VGG (Simonyan and Zisserman 2014), ResNet (He et al. 2016), DenseNet (Huang et al. 2017a), ResNeXt (Xie et al. 2017) and EfficientNet (Tan and Le 2019). However, most of these models are trained on high-quality image datasets like ImageNet (Deng et al. 2009) and MS COCO (Lin et al. 2014). When the image is compressed, the content can be changed in texture and structure, leading to possible semantic information variation or even loss, from which the CNNs may suffer. More unfortunately, the majority of the leading image and video coding frameworks and standards are developed to minimize signal-level loss between pristine and distorted data, which contributes to improving visual quality for humans, yet lacks optimizations specifically for machines to better understand the contents.

In lower bandwidth or smaller storage environments where higher levels of image and video compression are required, the decrease of machine vision performance can be non-negligible or even unacceptable, while the subjective image quality can still maintain. The reason for the relatively stable visual quality is that the HVS cannot notice some small differences between clean images and their distorted variants. To take advantage of the perceptional redundancy, the Just Noticeable Distortion (JND) model for image and video coding has been proposed (Jayant et al. 1993; Chou and Li 1995; Yang et al. 2005), which suggests a theoretically perfect compression strategy that can reduce the coding cost as much as possible without bringing any visual loss.

In recent years, several JND-annotated datasets have been released, which provides a large quantity of labeled data that can be used to train machine-learning-based JND predictors. The predicted JNDs can then be adopted as the reference to select appropriate encoding parameters for sufficiently good image quality with the fewest possible bits cost. Furthermore, many models and methods have been proposed to enhance the power and scalability of JND, such as introducing multiple JND points and modeling the satisfied-user-ratio (SUR) curve.

Considering the similarity between HVS and deep neural networks (Yamins and DiCarlo 2016; Schrimpf et al. 2018), it is possible that a JND-liked model can be built for image and video coding for machines, which inspires our work in this paper. Our goal is to develop a new method to optimize existing image and video coding frameworks for machine vision tasks by utilizing the machine JND after verifying its existence. The main contributions of our work are summarized as follows:

- Comprehensive experiments are designed and performed for various machine vision tasks on several widely-used datasets to investigate the influence on machine vision performance caused by image and video coding, which also demonstrates the existence of JND for machine vision.
- A new concept called Just Recognizable Distortion (JRD) is proposed to present the maximum distortion caused by image and video coding that will not reduce the machine vision models' performance to an unacceptable level. A large-scale JRD-annotated dataset containing over 340,000 images is built for further researches and applications, and the factors of JRD are analyzed. Applying JRD for image and video coding is proven to be effective at achieving superior machine vision performance under the same level of bits.
- An ensemble-learning-based framework is established to predict the image JRD for various visual tasks under few- and non-reference conditions where only the pristine image and a few or even no distorted images are available. The predicted JRDs from the framework can be used to optimize image and video coding to save massive bits while still keep the same level of machine vision performance according to the experiments.

The rest of the paper is organized as follows. Section 2 will present some related works, such as image and video coding for machines and JND modeling. Section 3 will study the performance of various machine vision models for different tasks on distorted images. Section 4 will describe the concept of JRD and the proposed dataset. Section 5 will firstly introduce the proposed JRD prediction framework, then prove the effectiveness of predicted JRDs to optimize image and video coding for machine vision. Finally Sect. 6 concludes the paper.

2 Related Work

2.1 Machine Vision on Low Quality Images

Several works have proved that image and video quality can affect machine vision models significantly. Dodge and Karam add diverse distortions to ImageNet, including blur, noise, contrast and JPEG compression, to evaluate CNN-based image classification models on distorted images (Dodge and Karam 2016). They also design some experiments to compare machines with humans on identifying low-quality images (Dodge and Karam 2017). According to their report, machines suffer from image quality decrease

much more than humans, even if they were fine-tuned on distorted images. Geirhos et al. reveal the poor generalization ability of machine vision models among various distortions (Geirhos et al. 2018). Su et al. find that tiny distortion is also possible to fool the most powerful CNNs, like modifying only one pixel in the pristine image (Su et al. 2019). Recently, Aqqa et al. examine the performance of several object detection models on videos that are compressed into different quality levels (Aqqa et al. 2019). These studies demonstrate that machine vision models perform ineffectively on distorted data, but the impact from image and video compression has not been investigated thoroughly.

2.2 Coding for Machine Vision

There are two major coding schemes for machine vision, which are compress-then-analyze (CTA) and analyze-then-compress (ATC). In the CTA scheme, images and videos are compressed and delivered to end applications where machine vision models have to extract features from decompressed and distorted data. While in the ATC scheme, only the visual features extracted from the pristine images and videos are encoded and transferred, and after decoding these features can be used directly by task-specific models.

To optimize coding for machine vision under the CTA scheme, a few methods have been proposed. Liu et al. train a simple classifier to predict whether the distorted images can be analyzed correctly under a specific compression level to adjust encoding parameters (Liu et al. 2017). Shi et al. use a deep reinforcement learning model to select more appropriate encoding parameters for image areas that might be important for machine vision models (Shi and Chen 2020). Nevertheless, the performance increase from these methods is not impressive enough and the computations are more complicated than expected.

In most situations, ATC outperforms CTA for machine vision tasks under the same coding cost (Redondi et al. 2016). Therefore, several coding standards have been established for the ATC scheme. Targeting visual analysis on still images, MPEG CDVS standardizes features extraction, compression, representation and evaluation techniques to form a unified feature coding framework (Duan et al. 2015). Its successor, MPEG CDVA extends the framework for video analysis by utilizing the temporal redundancy in neighboring video frames and importing CNN features as a complement to handcrafted features for better task performance (Duan et al. 2018). A recent study further enhances the effectiveness and generalization capability for deep feature coding by compressing intermediate-layer CNN features rather than ultimate layers (Chen et al. 2019). And the application scenario of ATC has also been discussed (Lou et al. 2019).

ATC is usually adopted along with CTA to convey not only features to machines but also images to humans. Zhang et al. propose the joint compression framework for both features and visual contents, where the two coding performances can be improved by each other (Zhang et al. 2016). Li et al. study the joint rate-distortion optimization problem under this framework (Li et al. 2018). Wang et al. use the high-quality decoded features to help reconstruct face images, simultaneously increasing image visual quality and face recognition accuracy (Wang et al. 2020). An overview of joint feature and texture coding framework is provided by (Ma et al. 2018), where the advantages of ATC-CTA fusion are demonstrated.

Although ATC can achieve excellent machine vision task performance with very few bits usage, it requires both the encoder and decoder side to upgrade for the feature coding scheme. Moreover, when the visual content is still essential, the feature processing of ATC will cost extra computing and network resources.

2.3 JND and Its Prediction

Picture-Wise Just Noticeable Distortion (PW-JND) describes the maximum distortion of an image that cannot be perceived by humans. A perfect PW-JND model is able to guide image and video coding to achieve the optimal balance between visual quality and coding cost. The existence of PW-JND on compressed images is firstly confirmed by (Lin et al. 2015). After that, several JND datasets have been released. For example, Jin et al. build the MCL-JCI dataset to annotate JNDs for images (Jin et al. 2016), and Wang et al. build the MCL-JCV dataset for videos (Wang et al. 2016). Both MCL-JCI and MCL-JCV are small-scale datasets that only contain 50 images and 30 videos respectively, but they can still be used as benchmark datasets for JND-related studies. Subsequently, a large-scale JND dataset, VideoSet, is established by academic and industrial organizations together after extensive subjective experiments, which contains 880 videos and their corresponding JNDs as well as SUR models (Wang et al. 2017). The VideoSet significantly empowers researchers to study the modeling, prediction and application for JND as well as SUR.

In order to use PW-JND as a reference to optimize image and video coding, the JND of a specific image should be predicted in advance. Huang et al. extract temporal and spatial features from pristine images to train a support vector regression (SVR) model to predict JNDs (Huang et al. 2017b). Wang et al. use quality degradation and masking features to increase the prediction accuracy for the first JND (Wang et al. 2018a), and they further extend this model to predict the second and third JNDs in VideoSet (Wang et al. 2018b). Zhang et al. propose to import saliency distribution and bit-rate change features to predict SURs, and they also investigate how to accelerate the JND and SUR prediction for practical applications (Zhang et al. 2020).

Deep-learning-based JND prediction methods have also been explored. Fan et al. predict SUR by training a CNN regression model with both pristine and distorted images as the inputs (Fan et al. 2019). Liu et al. propose a shared-weights neural network structure to estimate whether humans can notice the difference between two images, and search the correct JND with a sliding-window algorithm (Liu et al. 2019). Lin et al. fuse CNN features from different network layers to improve features representation capability, leading to more accurate JND prediction results (Lin et al. 2020).

Currently, JND models are only for human visual systems. And the effectiveness as well as efficiency of JND prediction models should be further improved.

3 Machine Vision on Compressed Images

In this section, we will study how image and video compression influence the performance of machine vision. A list of tasks, datasets and models are selected to make a comprehensive survey.

3.1 Image Classification

Image classification is the most fundamental machine vision task since class information plays an important role for the machine to understand an image or video. A classification model usually consists of two parts, a feature extractor and a classifier. The former is responsible to extract high-level, representative and distinguishable features from images. The latter can then use these features to categorize corresponding input images. The performance variation of a classification model under different levels of input image compression can reflect the robustness of the feature extractor when facing distorted images.

The experiments in this sub-section will be performed on two famous datasets, PASCAL VOC (Everingham et al. 2010) and MS COCO. These datasets are built not only for image classification but several other machine vision tasks such as object detection and semantic segmentation. Hence, one image may contain multiple objects in different categories. Besides, images in these datasets are retrieved and collected from the Internet so they have already been compressed. As a result, they may have diverse contents, resolutions and quality levels. Even though, both VOC and COCO provide a large number of object images and their category annotations, which are sufficient for training and evaluating classification models. Considering the fact that they are widely used in common machine vision researches, it is reasonable to make a hypothesis that the major part of images in these datasets only have negligible distortions introduced by image compression. This hypothesis will also be proved by experiments in this section later.

Table 1 Image classification performance in top-1 accuracy of VGG-19 and ResNet-101 on VOC and COCO datasets

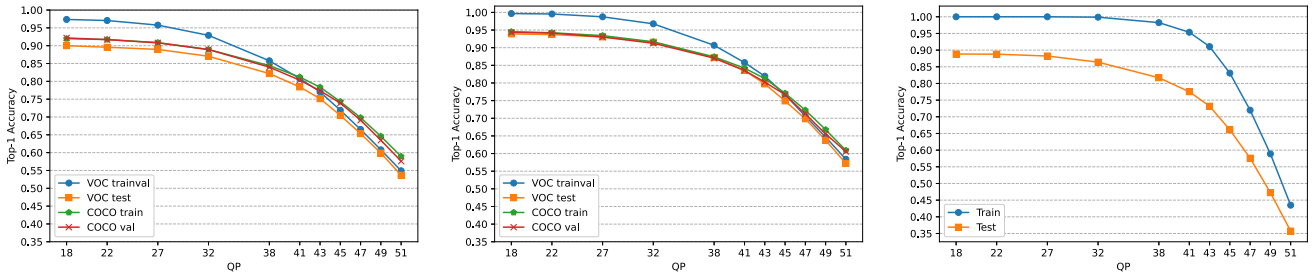
Dataset	Model	
	VGG-19	ResNet-101
VOC 07+12 trainval	0.9857	0.9998
VOC 07 test	0.9029	0.9418
COCO train2017	0.9217	0.9464
COCO val2017	0.9219	0.9452

Similar to the selection of datasets, we choose two well-known models, VGG-19 and ResNet-101, to test their performance on compressed images. In our experiments, we use the weights pre-trained on ImageNet to initialize these two models and fine-tune them on VOC 07+12 trainval dataset to output 20 object categories. The performances of the trained models measured by top-1 classification accuracy are presented in Table 1, which indicates that both VGG and ResNet are very powerful to categorize these high-quality images. We provide training, valid and test results here for a more comprehensive survey.

In this paper, without loss of generality, we choose HEVC as the reference coding standard, which consists of many advanced compression techniques. It is worth mentioning that although HEVC is usually used for videos, it can also be used for images. The reference software of HEVC, HM 16.16, is used as the image encoder. The experiment procedure is described as follows.

Firstly, we convert every image in the source datasets into YUV format. Secondly, each of these YUV images will be regarded as a video frame and encoded by HM under the All-Intra configuration. 11 quantization parameters (QP) are selected to compress each image into different quality levels, which are 18, 22, 27, 32, 38, 41, 43, 45, 47, 49 and 51. A larger QP leads to more severe signal loss and thus worse image quality. Thirdly, we collect the compressed YUV data and convert them back into RGB format to get the distorted images. After that, if a whole image contains multiple objects, we will crop each object using the annotated object bounding boxes to obtain the single-object images. Finally, the single-object images that have different quality levels are sent to image classification models as inputs, and the performance can then be acquired, which are shown in Fig. 1a, b. To keep the figures clean, we ignore the year information of the datasets in the legends.

According to the experimental results, the performance change is similar for 2 models on 4 sub-datasets. Obviously, as the QP increases and the image quality decreases, the classification accuracy drops. However, the dropping speeds are different for various QP ranges. When the QP enlarges from 0 (not compressed) to 32, the accuracy only falls at 3.26%, which is tolerable. This result also proves the hypothesis we



(a) VGG-19 **(b)** ResNet-101 **(c)** WS-DAN

Fig. 1 Coarse- and fine-grained Image classification performance under different QPs

make before that the images in selected datasets can be treated as pristine images for machine vision tasks, though they have already been compressed. However, when the QP is greater than 38, the accuracy decreases at 3% for every 2-step QP growth. Therefore, in extremely low image quality situations when the QP rises to 51, the performance loss of classification models becomes unacceptable.

There is an important observation from the experiment. Although the VGG-19 and ResNet-101 are trained on the VOC trainval dataset and behave very powerfully according to Table 1, their performance still decreases on compressed versions of the same dataset. Especially after QP 38, the accuracy drops significantly. This phenomenon suggests that the feature representation of compressed images may be obviously different from their corresponding pristine versions.

The differences in size, shape, structure, texture or other aspects between the object categories appeared in ImageNet, VOC or COCO are usually huge, like person and dog, or apple and mobile phone. Therefore, it is easier to identify these classes, compared with some quite similar categories such as wheat and weed, or cars that have different brands. Recently there are many techniques proposed specifically for these fine-grained image classification problems, but they may suffer from the decrease of image quality even more terribly, considering the minor distinguishable features might be erased more easily by image compression. We also test on a transcoded CUB-200-2011 birds dataset (Wah et al. 2011) to make an investigation. A powerful WS-DAN model (Hu et al. 2019) is trained and its performance under different levels of input image quality is recorded, which is drawn in Fig. 1c.

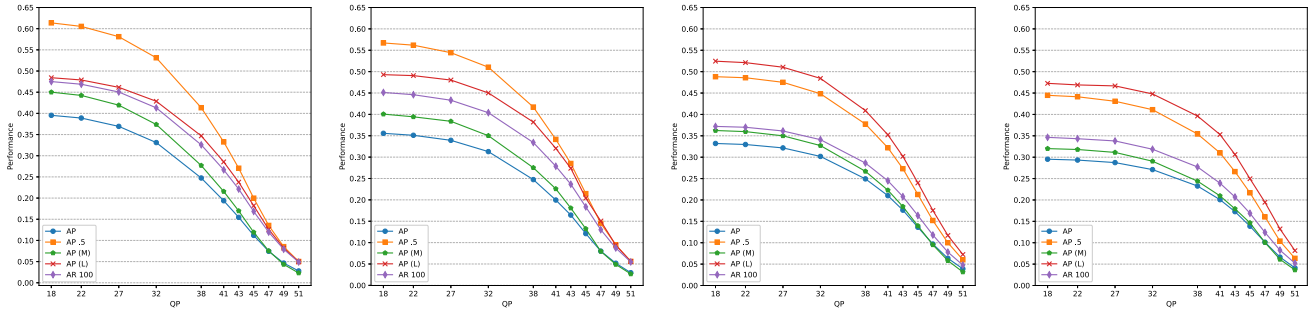
Comparing Fig. 1c with 1a, b, there are several noticeable differences. Firstly, the accuracy drops in different speeds on the training and testing dataset. The fine-grained classification model maintains remarkable performance until the QP exceeds 41 on the training dataset, while an 11.5% accuracy decrease is recorded at QP 41 on the testing dataset. A possible explanation for this phenomenon is that the attention module and the bilinear features module adopted in WS-DAN depend more on feature robustness. Secondly, for severely distorted images, the model performance is reduced to a very unfavorable level.

3.2 Object Detection

Object detection is one of the hottest machine vision tasks among current researches, which is usually more complicated and challenging than image classification because the machine detector should not only predict the category of an object but also output its location in the image as well as the size. Furthermore, an image might contain multiple objects that differ a lot in size and category, and these objects can overlap with each other. After compression, the outlines, structures or textures of objects in the image can be altered significantly, which makes it even harder for detectors to identify, locate and recognize targets.

Common object detection models can be divided into two categories, Two-Stage detectors and One-Stage detectors. Two-Stage detectors usually perform better than One-Stage detectors, but their computation is also more time-consuming. On the contrary, One-Stage detectors are usually fast while not able to achieve the best accuracy. For our experiment, Faster R-CNN (Ren et al. 2016) as a representative for Two-Stage detectors, and CenterNet (Zhou et al. 2019) for One-Stage detectors, are selected to test machine detector robustness on a variety of image quality after compression. To make a fair comparison between these two models, their feature extractors are unified to be ResNet-101, and both of them are trained on the COCO train2017 dataset. The experiments are performed on the transcoded COCO dataset, which has already been introduced in Sect. 3.1. The results are shown in Fig. 2 in different kinds of measurements that will be explained later. To keep the figures clean, we use "val set" as an abbreviation for "validation set" in the sub-captions.

Similar to or even worse than fine-grained image classification, both Two-Stage and One-Stage models totally lose their efficiency for detecting heavily-distorted objects, according to Fig. 2. Faster R-CNN outperforms CenterNet in relatively high image quality situations, while the advantage



(a) Faster R-CNN on train set **(b)** Faster R-CNN on val set **(c)** CenterNet on train set **(d)** CenterNet on val set

Fig. 2 Object detection performance of Faster R-CNN and CenterNet on COCO dataset under different QPs

is reversed after QP 41 if we adopt mean average precision (mAP or AP) from the COCO evaluation framework as the performance measurement. A possible reason is that a Two-Stage model uses the extracted features twice, once for locating and another for classifying, leading to larger error accumulation during processing. However, if the mean average recall (AR) is considered, the Two-Stage Faster R-CNN is always achieving better scores than the One-Stage CenterNet, which suggests the benefit of a separated locating stage.

Figure 2 also presents whether the object size can influence the object detection performance along with image quality by drawing the curves for AP (M) and AP (L). The AP (M) stands for the detection AP on those middle-sized object images that are larger than 32×32 pixels but smaller than 96×96 pixels, and the AP (L) is for those large-sized objects that have at least 96×96 pixels of area. From the figure, it is clear that larger objects are easier to detect no matter how distorted the images are. Nevertheless, the decrease of detection accuracy is in a similar trend for both middle- and large-sized objects when the compression level enlarges.

4 Just Recognizable Distortion

In the last section, we investigate the influence of image and video compression on machine vision performance. According to the comprehensive experiments, worse input image quality and larger distortion lead to lower model accuracy in all kinds of machine vision tasks. It is verified that the machine vision models are not reliable even for images that may have only slightly different compression levels. As for extremely poor image quality conditions, the models can be totally useless. Therefore, it is meaningful to improve the machine vision task performance when higher compression levels are forced to be adopted due to limitations for transmission or storage.

From the experimental results in Sect. 3, there are two interesting observations. Firstly, the performance of machine vision models is usually very similar between pristine and lightly-distorted input images. Secondly, for some images, the recognition never fails even under the maximum level of compression. These phenomenons indicate the possible existence of machine visual redundancy. If the most appropriate quality levels for images that will ensure the correct prediction for machine vision models can be found, we can save massive bits by giving up with better but more redundant image quality.

Table 2 Training, validation and test image counts of the JRD dataset for VGG-19 and ResNet-101

	N_{train}	N_{val}	N_{test}	N_{total}
VGG-19	299680	28087	27774	355541
ResNet-101	290097	27686	27131	344914

In this section, we propose the concept of JRD, which denotes the maximum distortion that can be introduced by image and video coding without dropping the machine vision performance to an unacceptable level. An optimal JRD model can be used to optimize image and video coding for reliable machine visual analysis more economically by guiding the selection of compression parameters.

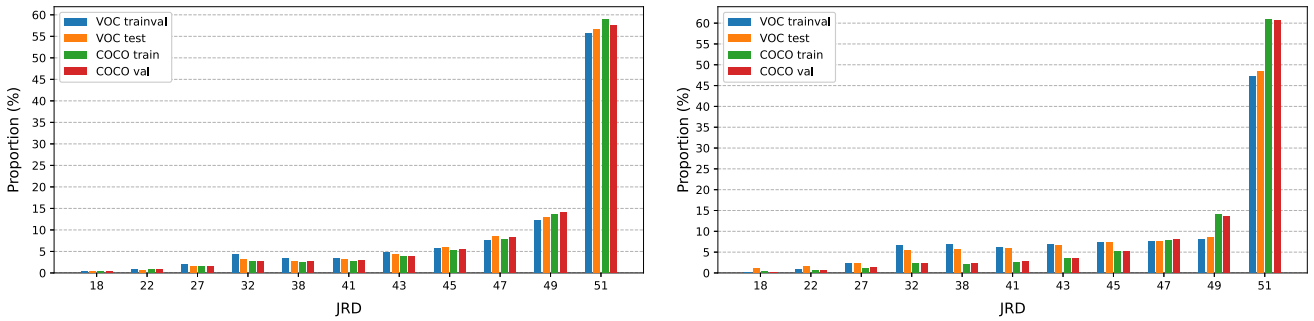
Now we will give the definition of the JRD. Suppose there is a pristine image I_0 that can be compressed into several distorted images $I_1, I_2, \ldots, I_n$, in which larger indices represent worse image quality. For a machine vision model M that can give a correct or acceptable prediction on I_0, the JRD of I_0 is defined as

$$\text{JRD}(I_0; M) = j \tag{1}$$

if and only if the following constraints are satisfied:

$$\begin{aligned} M(I_j) &= M(I_0) \\ M(I_{j+\epsilon}) &\neq M(I_0) \end{aligned} \tag{2}$$

where j is the coding parameter at the JRD point, and ϵ represents any positive integers.



(a) VGG-19 **(b)** ResNet-101

Fig. 3 The JRD categories distributions image classification in proportions of total images

Table 3 The distribution of JRD for image classification in exact counts of images

	18	22	27	32	38	41	43	45	47	49	51
VGG-19	1179	2683	5367	10500	9140	10126	14354	19285	27914	47732	207261
ResNet-101	994	2223	4663	10393	10072	11018	14449	19270	26649	44298	200885
N^*	23	107	501	1702	1182	1220	2028	3094	5239	12120	157585

It should be mentioned that there is a hypothesis behind the definition of JRD, which is smaller coding distortion leads to closer or more acceptable machine visual analysis result with the ground truth as well as the output from pristine images. This hypothesis is proven statistically by extensive experiments in Sect. 3. Therefore, those images that are easier to be analyzed in a more distorted situation are ignored for more consistent JRD definition and annotation.

In this paper, we will study the JRD for the two most fundamental machine vision tasks, image classification and object detection. The QP in HEVC is used to measure the JRD. Similar to the establishment of datasets in Sect. 3, 11 QP values are selected to determine the JRDs of images for the two tasks separately.

4.1 JRD for Image Classification

For a typical image classification model, the outputs are usually the probabilities of different categories that the object may belong to, which can be denotes as $p_{c_1}, p_{c_2}, \ldots, p_{c_n}$, where c_k stands for the k-th category. In order to define and annotate the JRD for image classification, the most strict top-1 accuracy is adopted. For image I_0, we have:

$$M(I_{\mathrm{JRD}}) = l$$
$$p_{c_l} = \max(p_{c_1}, p_{c_2}, \ldots, p_{c_n}) \tag{3}$$

We annotate JRDs for image classification on VOC and COCO datasets using VGG-19 and ResNet-101. Both of the models are pre-trained on pristine ImageNet and fine-tuned on pristine VOC 07+12 trainval dataset to avoid fitting with distorted data. Considering there are many small objects that are too hard to identify, we only target objects that have at least 32×32 pixels. Similar to the data preparation in Sect. 3.1, we crop the objects from the whole images using the bounding box annotations of source datasets to obtain more usable single-object images. After annotation, a JRD dataset containing over 340,000 images is built.

The training, validation and test split of the JRD dataset naturally inherits from the source datasets, except we use images in COCO val2017 as test images. The data distributions of the JRD dataset for the two models are presented in Table 2, where N_{train}, N_{val}, N_{test} and N_{total} denote the image counts of training, validation, test dataset and the complete dataset, respectively. The JRD categories distributions for the two models are shown in Fig. 3 in proportions of total images and Table 3 in exact image counts. The legends in Fig. 3 show the source datasets of annotated images, and the N^* row in Table 3 stands for the counts of images that have the same JRDs. Again, we ignore the year information of the source datasets in the legends of Fig. 3 to make them clean.

According to Fig. 3 as well as Table 3, more object images have larger JRDs. Surprisingly, over 58% of objects have the JRD of 51, which indicates remarkably robust machine recognition under extremely bad image quality. The two tested models are sharing approximate JRD distribution, but for a specific image, its JRD may vary with the model. According to statistics, about 55.86% of all annotated object images have identical JRDs on both VGG-19 and ResNet-101, 20.31% of objects have larger JRDs on VGG-19, and 22.83% of objects have larger JRDs on ResNet-101. The difference between the average JRD value on VGG-19 and

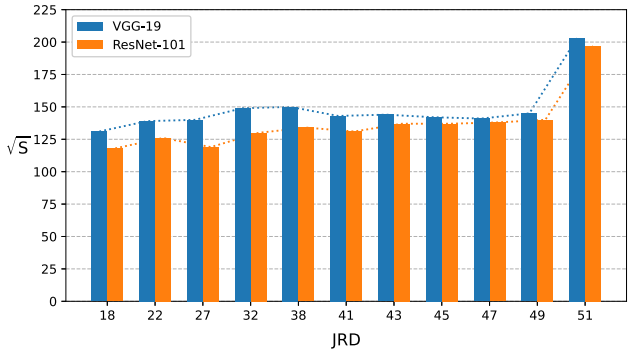


Fig. 4 Average sizes in the square root of pixels count of objects having different JRDs for image classification

ResNet-101 is -0.21, which suggests better robustness for the image quality variation on the latter model.

In Sect. 3.2, we showed and analyzed the influence of the image size to object detection performance. Although small objects have already been ignored during the establishment of our JRD dataset, such impact may still exist. Therefore, we record the average object image sizes for different JRDs, which are shown in Fig. 4. The size is presented in the form of the square root of pixels count. From the figure, it is obvious that the average sizes of images that have different JRDs are close, except for QP 51, thus the image size will not influence the JRD significantly. As for JRD 51, objects that are very easy to recognize are gathered, which includes many large objects that increase the average size.

We also explore the distribution of JRD for different categories. As shown in Table 4, 6 categories of images are selected as representatives for ResNet-101 to determine JRDs. For each category, the proportions of total images belonging to the specific JRD values are presented on the first row, and their average sizes in the form of the square root of pixels count are presented on the second. The total annotated objects counts are also given after the category names.

There are a few observations from Table 4. Firstly, for categories having more objects like person, car and chair, their JRDs are more likely to be larger. The proportions of objects that have JRD of QP 51 for these 3 categories are 72.83%, 50.92% and 42.21%, respectively, which suggests better capability for the model to identify these categories even when the input images are in a very low-quality level. One possible reason is that the model can learn more representative features that can be maintained during image compression after training on more data. On the contrary, it is more difficult for a machine classifier to learn well-generalized features for those categories that have less or even insufficient amounts of images, such as cat and dog, leading to more balanced JRD distributions. However, for several other categories that do not own too many objects, like the aeroplane, their JRDs might also be gathered in QP 51 due to the existence of some extremely large-sized objects that are relatively simple to classify.

Once the JRD of an image is determined, it is possible to save a large number of bits when a certain level of machine vision performance is required. In order to prove the effectiveness and power of the proposed JRD concept, we test on COCO train2017 dataset for image classification using ResNet-101 as the representative of machine classifiers. Before being classified, images are compressed with different QPs and JRD. For the JRD encoding pattern, the actual QP used to compress a specific image is set to the annotated JRD. The bit-rate is adopted as the measurement of transmission and storing cost for encoded images, for which a larger bit-rate indicates more bits. We use the top-1 accuracy to compare the classification performance between different QPs and JRD, which is shown in Fig. 5, and more detailed data can be checked in Table 5 where QP 0 denotes images in the original dataset.

According to Fig. 5 and Table 5, over 96% of bit-rate can be saved to achieve similar or even better [1] classification performance as QP 18 or even pristine images if the JRD is used as the actual QP for compression. From another perspective, the corresponding bit-rate of JRD is between QP 43 and 45, closer to the latter, while the classification accuracy increases for more than 16% in absolute value. Therefore, in low bit-rate situations, the JRD can be used to improve the image classification performance. On the other hand, given demand for a specific level of classification performance, applying JRD can also save a large number of bits.

4.2 JRD for Object Detection

Comparing with image classification, it is more complicated to determine the JRD of an image for object detection. There are two reasons. Firstly, the JRD is annotated for a whole image for image classification, while for object detection the annotation should be made for every possible object in an image, which can conflict with each other. Secondly, for image classification, the correctness of a prediction can be verified more intuitively by only considering the predicted category that has the largest probability, thus the JRD of an image can be found by Eqs. 1 and 2. However, for object detection, the evaluation depends on several threshold values for permitted detection count, Intersection over Union (IOU) between ground truth and detected bounding box, class confidence, and so on. Therefore, in order to annotate the JRD of

[1] Which means for some images, lower image quality can lead to more accurate category prediction from machine classifier. Nevertheless, these images are still ignored as explained in Sect. 4, and they will not influence the conclusion of this section or the paper.

Table 4 JRD distributions for different categories for ResNet-101

Category	JRD										
	18	22	27	32	38	41	43	45	47	49	51
Person (187707)	0.08	0.19	0.36	0.90	1.03	1.35	2.06	3.37	5.87	11.96	72.83
	122	137	116	105	105	96	100	98	101	112	195
Cat (5735)	0.99	2.14	4.62	11.23	9.15	8.65	10.01	11.35	11.21	11.12	19.51
	182	190	194	214	233	248	263	280	292	318	355
Dog (6569)	1.28	2.25	4.90	12.07	9.50	8.98	10.81	11.27	11.39	10.87	16.68
	154	148	139	171	196	207	233	251	271	285	321
Car (25768)	0.26	0.62	1.35	2.73	3.16	3.59	5.20	7.07	9.74	15.38	50.92
	98	100	70	83	90	88	94	95	101	108	149
Aeroplane (5498)	0.24	0.49	0.96	2.29	2.20	3.51	3.38	4.82	6.11	8.44	67.55
	98	146	83	95	86	108	115	109	140	166	270
Chair (32838)	0.58	1.18	2.33	4.40	4.39	4.49	5.97	7.73	10.35	16.36	42.21
	124	121	124	131	131	125	131	121	125	120	131

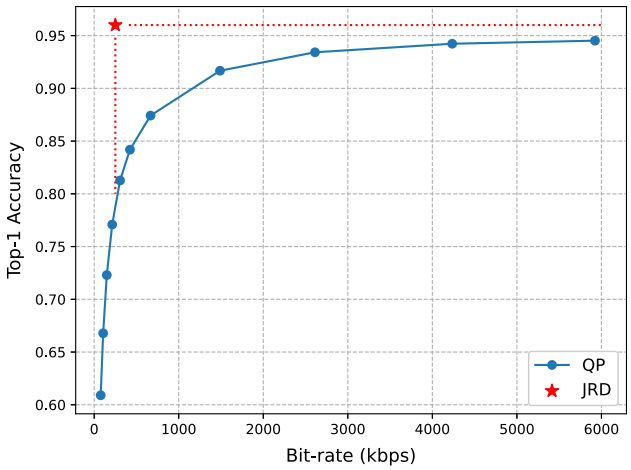


Fig. 5 Bit-rate and image classification performance comparison among JRD and different QPs

an image for object detection, a group of evaluation thresholds should be fixed at first.

The JRD annotating method for object detection will be described as follows. Given an object o in the pristine image I_0 whose JRD is denoted as j, the machine object detector M may give n possible detection results for the distorted image I_j as (4):

$$M(I_j) = \{B_j^1, B_j^2, \ldots, B_j^n\} \tag{4}$$

Each prediction is presented by a quintuple:

$$B_j^k = (x_k, y_k, w_k, h_k; p_{j,k}^{c^*}) \tag{5}$$

where (x_k, y_k) is the coordinate of the left-top corner of predicted object bounding box, and w_k and h_k are the width and height of the box, respectively. As for $p_{j,k}^{c^*}$, it presents the maximum class confidence across all possible categories predicted by the object detection model. In other words, the output class of B_j^k from the model will be c^*. The object o can also be presented in the similar form as:

$$B_l^o = (x_o, y_o, w_o, h_o; c_o) \tag{6}$$

where l represents any QP values, and c_o is the actual category that the object o belongs to.

The n detection results will be sorted according to their class confidences in descending order, and only the first T_n ones are preserved. Finally, the JRD of object image o equals to j if and only if the following constraints are satisfied:

$$\begin{aligned}
\exists B_j^k \in \{B_j^1, B_j^2, \ldots, B_j^{T_n}\} \rightarrow\ & IOU(B_j^k, B_j^o) \geq T_{\text{IOU}} \\
& \wedge\ p_{j,k}^{c^*} \geq T_p \\
& \wedge\ c^* = c_o \\
\forall B_r^k \in \{B_r^1, B_r^2, \ldots, B_r^{T_n}\} \rightarrow\ & IOU(B_r^k, B_r^o) < T_{\text{IOU}} \\
& \vee\ p_{r,k}^{c^*} < T_p \\
& \vee\ c^* \neq c_o
\end{aligned} \tag{7}$$

where T_{IOU} and T_p are thresholds for IOU and class confidence, respectively, and r represents any QP values that are larger than j.

In this paper, we will annotate the JRDs of objects images that belong to the person class for object detection on the COCO dataset. The Faster R-CNN with ResNet-101 as feature extractor is selected to be the representative of machine detectors, which is trained on the pristine COCO train2017 dataset. Small objects that have fewer pixels than 32×32 are still ignored, and the objects are also cropped from the whole image to enlarge data size. The evaluation thresholds are set to $T_n = 100$, $T_{\text{IOU}} = 0.8$, $T_p = 0.5$. After annotation, a JRD dataset containing over 130, 000 images is built. Similar to

Table 5 Bit-rate and image classification performance under JRD and different QPs

	JRD	0	18	22	27	32	38	41	43	45	47	49	51
Accuracy	0.9600	0.9475	0.9452	0.9423	0.9342	0.9167	0.8742	0.8419	0.8127	0.7709	0.7230	0.6678	0.6090
Bit-rate (kbps)	250	–	5924	4235	2612	1488	667	423	306	213	149	106	77

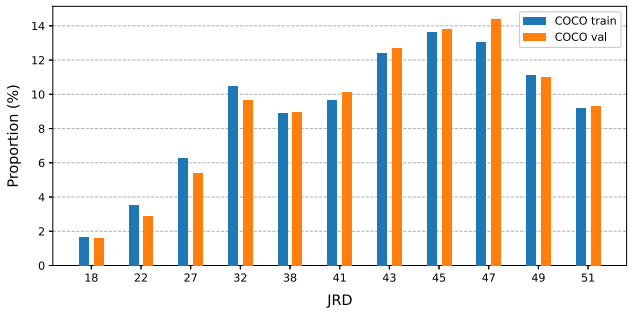


Fig. 6 The JRD categories distribution for object detection in proportions of total images

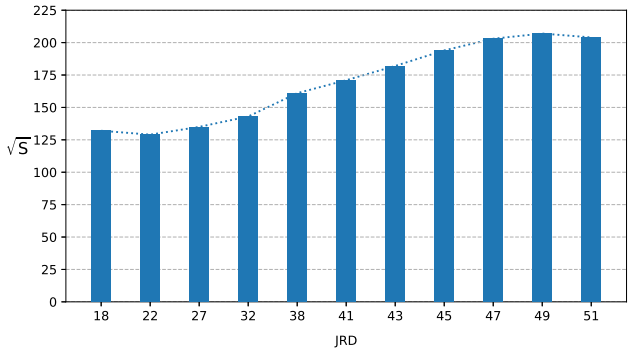


Fig. 7 Average sizes in the square root of pixels count of objects having different JRDs for object detection

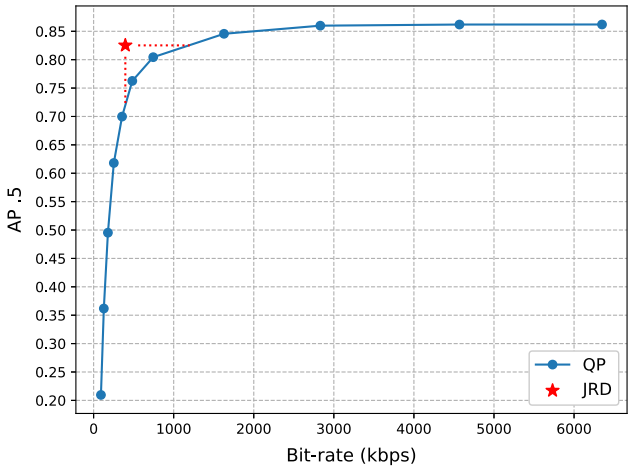


Fig. 8 Bit-rate and object detection performance comparison among JRD and different QPs

the JRD dataset for image classification, we use the images in COCO val2017 as the JRD test dataset. Besides, 5000 object images are randomly split from the COCO train2017 to form the JRD validation dataset, and the rest becomes the JRD training dataset. These three datasets have 122929, 5000 and 5847 images, respectively. The JRD categories distribution for object detection is shown in Fig. 6, where the legends show the source datasets of annotated images. It can be observed that the distribution of JRD for object detection is more balanced than for image classification.

The average sizes of objects having different JRDs for object detection are also presented in Fig. 7. It is clear that larger objects have larger JRDs, but do not cluster only under the largest QP. Therefore, the size can be a non-negligible factor for determining the JRD of an object image for object detection.

Similar to the application of JRD for image classification, we also test on COCO val2017 dataset to prove the effectiveness of JRD for object detection. Since an original image from the dataset can contain multiple person objects, we sort these objects according to their JRDs in descending order, and then set the LCU-level QP to JRD for each object areas one-by-one before compression. The LCU is the largest coding unit in HEVC, which is usually a square area that contains 64×64 pixels. An object area might be included in a combined area of several adjacent LCUs, thus can cost extra bits to compress. To maximize bit-rate saving, other areas that do not contain person objects, or contain person objects but not annotated for JRDs,[2] will be compressed with QP 51. Nevertheless, due to the possible overlapping among objects and the reversed JRD adoption strategy, the cross regions are very likely to be compressed with smaller QPs, which may increase not only the detection accuracy but also the bit-rate at the same time. Moreover, an LCU that covers an object bounding box area but does not belong to the object itself (since an object may not always occupy an entire rectangular area) will also be compressed with the corresponding JRD as QP. Under such configurations, the detection performance of ResNet-101-based Faster R-CNN under JRD and different QPs are shown in Fig. 8 as well as Table 6 where QP 0 denotes images in the original dataset.

According to Fig. 8 and Table 6, over 47% of bit-rate can be saved to achieve better detection performance than QP 38 if the JRD is used to determine the actual QP. From another

[2] Which means these person objects cannot be detected under any selected compression levels.

Table 6 Bit-rate and object detection performance under JRD and different QPs

	JRD	0	18	22	27	32	38	41	43	45	47	49	51
AP .5	0.8252	0.8622	0.8620	0.8619	0.8599	0.8456	0.8044	0.7626	0.6998	0.6181	0.4953	0.3618	0.2096
Bit-rate (kbps)	396	–	6349	4569	2831	1629	744	483	355	252	179	127	91

perspective, the corresponding bit-rate of JRD is between QP 41 and 43, closer to the former, while the detection accuracy increases for more than 10% in absolute value. Compared with the superior performance improvement from the adoption of JRD for image classification, the enhancement is smaller for object detection because of different task difficulties, model complexities and evaluation strategies.

Finally, we investigate the correlation between JRDs for image classification and object detection. According to the statistics, among the 133, 776 person objects that have been annotated JRDs for both image classification and object detection, 10.73% has identical JRDs for these two tasks, while 82.82% of this part of objects have JRD of QP 51. Furthermore, only 1.20% of all objects have larger object detection JRD than image classification JRD. The difference between the average JRD value of image classification and object detection is $+9.7423$, which indicates a higher requirement of input image quality for object detection than image classification.

5 Coding Optimization Based on JRD Prediction

In the last section, we propose the concept of JRD for image and video coding to improve the machine vision performance, especially in low bit-rate situations. According to the experiments, applying JRD for image and video compression can help save massive bits for acceptable model accuracy. Therefore, if the JRDs of images can be predicted accurately before image compression, the coding process can be optimized by selecting more appropriate parameters according to JRD to achieve the best possible balance between bit-rate and machine vision performance.

In this section, we will study the prediction for JRD on the proposed JRD dataset under few- and non-reference circumstances. In the few-reference situation, in addition to pristine images, several distorted images generated by pre-compression are available as references for the JRD prediction model to find how image features vary with compression levels change. On the contrary, in the non-reference situation, only the original images are required, which does not need any time-consuming pre-compression, yet is more challenging for JRD prediction.

Because we focus on how image quality influences the performance of deep-learning-based models, we design the JRD prediction framework with advanced CNN modules, which will be described in detail in the following sub-sections. Since we use QP to measure the JRD of an image, which is finite and discrete, the prediction for JRD can be regarded as a classification problem. Without loss of generality, we predict the JRDs for ResNet-based models, which are ResNet-101 for image classification and Faster R-CNN with ResNet-101 as feature extractor for object detection.

5.1 Non-reference JRD Prediction and Coding Optimization for Image Classification

In this section, we will study the JRD prediction problem under the non-reference circumstance for the image classification task. Firstly, to better understand the JRD prediction problem, we would like to test the difficulty of predicting JRDs with straightforward methods. Therefore, we modify a VGG-19 network by changing the dimension of its output layer to the count of possible JRD values to build a baseline JRD prediction model, which can be trained through end-to-end learning on the JRD dataset.

From the investigation on Sect. 4.1, the distribution of JRD for image classification is imbalanced, where the majority of images have large JRDs. It is hard for an ordinary, non-optimized learning-based model to perform well on such long-tailed data. According to the experiment, the average difference between actual and predicted JRD-pairs from the baseline model is -5.57 on the JRD test dataset, and the mean average error (MAE) is 5.57. The bias for different JRDs is so significant that nearly all input images are predicted to be having the JRD of QP 51, which also makes the MAE equals to the absolute value of the average difference. Therefore, the baseline model is not powerful enough to solve the problem individually. Nevertheless, the output of the baseline model is not random, which suggests that several JRD-related image features are possible to be extracted by the CNN module.

To obtain more accurate JRD predictions, an ensemble-learning-based framework is proposed, which consists of several binary classifiers. Suppose there are n possible QPs $q_1, q_2, \ldots, q_n$ that will be adopted to compress the pristine image I_0 to distorted images $I_{q_1}, I_{q_2}, \ldots, I_{q_n}$, n binary classifiers $C_{q_1}, C_{q_2}, \ldots, C_{q_n}$ will be trained, in which C_{q_k}

Table 7 Positive class weights settings and accuracy of binary classification models for image classification JRD prediction

q_k	38	41	43	45	47	49	51
W_{pos}	0.06	0.12	0.18	0.24	0.4	0.5	1.0
Acc_{pos}	0.8307	0.8361	0.8284	0.8306	0.8341	0.8211	0.7651
Acc_{neg}	0.7148	0.7427	0.7405	0.7318	0.7321	0.7391	0.7535
Acc_{all}	0.8258	0.8301	0.8203	0.8179	0.8159	0.7999	0.7593

is responsible to predict whether the image I_{q_k} can be recognized successfully. In our experiment, C_{q_k} is a modified VGG-19 network that only outputs the possibilities for positive and negative classes.

To train C_{q_k} using all available data, the positive training samples are set to be all annotated training images of which JRDs are greater than or equal to q_k, and the negative samples are set to be images of which JRDs are smaller than q_k. However, the counts of positive and negative samples are not comparable according to Sect. 4.1, and the gap enlarges as q_k decreases. Therefore, two approaches are adopted to improve the class-imbalanced training. Firstly, we will not train any binary classifiers for $q_k < 38$ considering the facts that less than 5% of images have smaller classification JRDs than QP 38, and the bit-rate is going to explode if the image is compressed with such small QPs while the classification accuracy increase is not significant. Secondly, we set different weights for the gradients calculated from positive and negative samples through back propagation so the model can learn the correct features more equally. In the experiment, the weights for negative classes are always 1.0, and the weights for positive classes are shown in the second row of Table 7.

During the training process, we only adopt horizontal flipping to augment the training procedure. Because other methods like random cropping, scaling or color jitter might alter the JRD of the image. Each binary classifier will be trained on the JRD training dataset for 12 epochs by the stochastic gradient descent (SGD) optimizer with batch size set to 32, learning rate to 0.001, weight decay to 0.001, and momentum to 0.9. The classification accuracy for different binary classifiers on the JRD test dataset is shown in Table 7, where Acc_{pos} is the accuracy for positive samples, Acc_{neg} for negative samples, and Acc_{all} for all samples.

Combining these well-trained binary classifiers, the JRD of a certain object image can be searched from back to front as shown in Fig. 9. The searching procedure is allowed to be interrupted if the JRD has already been determined using fewer binary classifiers. The binary classifiers and the searching strategy jointly compose the JRD prediction framework, which is flexible and extensible. For example, for real-world deployment, the count of binary classifiers can be adjusted based on the model accuracy or the demand to bit-rate and image classification performance.

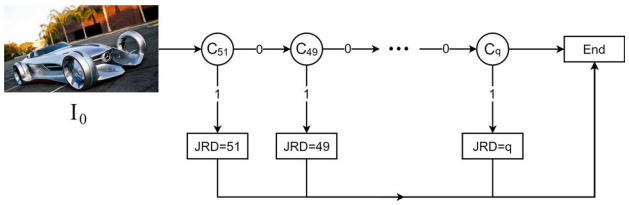


Fig. 9 JRD search strategy of the proposed ensemble-learning-based JRD prediction framework

According to the prediction results on the proposed JRD test dataset, the average difference between actual and predicted JRD-pairs from the proposed framework is -1.54 when the images having smaller JRD than QP 38 are ignored, and the MAE is 1.90.

To prove the effectiveness of the proposed JRD prediction framework on saving a large amount of bits while still achieving good image classification performance, an experiment is performed on the COCO val2017 dataset. The predicted JRD is used as the actual QP to compress the pristine images in the experiment. For images that are predicted to have smaller JRDs than QP 38, they will be compressed with QP 32. A ResNet-101 model pre-trained on pristine ImageNet and fine-tuned on pristine VOC 07+12 trainval dataset is the test target. The experimental results including the top-1 image classification accuracy and bit-rate under the predicted JRD, actual JRD and several other reference QPs are presented in Table 8.

From Table 8, it can be observed that the bit-rate is between QP 41 and 43, closer to the latter, while the image classification accuracy is higher than QP 41 after using predicted JRD as the QP for compression. If the changes of accuracy and bit-rate are regarded as approximately linear between QP 38 and 43, then about 40.47% of bit-rate can be saved under the same level of image classification performance, and 4.45% of performance improvement in absolute value can be achieved under the same level of bit-rate with predicted JRDs.

According to Table 8, the predicted JRDs from the proposed model can help save a large number of bits for good image classification performance. It is also possible to achieve higher image classification accuracy within equal or similar bit-rate by adjusting encoding parameters according to the predicted JRD. Although currently there is a huge gap between the effectiveness of the predicted and actual JRD,

Table 8 Bit-rate and image classification performance under predicted JRD, actual JRD and different QPs

QP	Predicted JRD	Actual JRD	18	32	38	41	43	45
Accuracy	0.8571	0.9518	0.9437	0.9126	0.8703	0.8354	0.8024	0.7669
Bit-rate (kbps)	338	242	5886	1472	659	418	302	211

further improvements to the prediction or the application of JRD for image classification can be expected.

5.2 Few-Reference JRD Prediction and Coding Optimization for Object Detection

In this sub-section, we will study the prediction of JRD for object detection under few-reference circumstances. For the pristine image I_0 and its distorted versions $I_1, I_2, \ldots, I_n$, a few-reference JRD prediction model should give the correct JRD of I_0 as (8):

$$M(I_0; I_{k_1}, I_{k_2}, \ldots, I_{k_r}) = j, \quad 1 \le k_1, k_2, \ldots, k_r \le n, \ r \ge 1 \tag{8}$$

Based on Eq. (8), the optimal JRD prediction accuracy can be achieved when all of the distorted images are available as input, namely $r = n$. However, in such case, we are able to directly obtain the correct predictions by sending these images to the machine object detector, which makes it meaningless to predict the JRDs. As r decreases, fewer times of compression are required, while it is more challenging to predict JRDs accurately.

We focus on the situation where $r = 1$ in this paper, which means only one distorted image as reference is used along with the pristine image to predict JRD for object detection. According to the distribution of JRD for object detection in Sect. 4.2, about 52.88% of all annotated images have JRDs that are smaller than or equal to QP 43, and the rest have larger JRDs than QP 43, which results in an approximately half-to-half distribution. With the assistance of images compressed with QP 43 as references, the binary classification models can make more accurate predictions in two steps. Firstly, they divide inputs into two balanced categories, i.e., images that can be detected after being compressed with QP 43 and images that cannot. Secondly, they predict whether the inputs can be detected after being compressed with the specific QP values that they are responsible for. Therefore, we choose the image of QP 43 as the only reference to help predict the JRDs of images for object detection.

The proposed ensemble learning framework for predicting JRDs for image classification is also adopted for object detection, in which the binary classifiers are extended for multiple inputs. For each binary classifier, the feature variance between original and compressed images can be captured by a re-designed CNN structure shown in Fig. 10. At first, the model will extract features from both the pristine and distorted image using the same weights, which are denoted by F_0 and F_q respectively. Then the difference between F_0 and F_q, namely F_Δ, is calculated by subtracting F_0 with F_q, which represents the features change. Finally, a fusion of F_0, F_p and F_Δ will be formed as high-dimensional distinguishable features and sent to the fully connected layer to output the binary class probabilities.

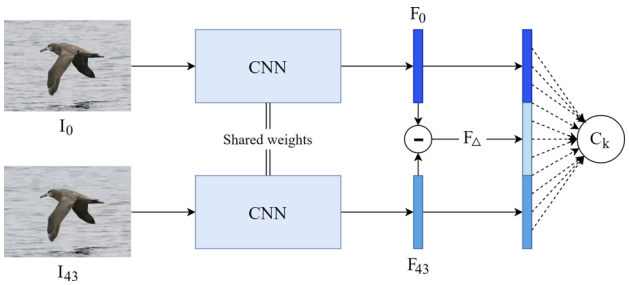


Fig. 10 The structure of binary classifier in proposed JRD prediction framework for object detection

Similar parameters are used to train the binary classifiers for object detection JRD prediction. However, due to the more balanced distribution of JRDs for object detection presented in Sect. 4.2, the weights for positive and negative classes are updated to values in the second and third row of Table 9, and the classification accuracy for different binary classifiers on JRD test dataset is shown in the last three rows of Table 9.

After all binary classifiers are trained, the JRD of an image for object detection can be searched. Without objects having smaller JRDs than QP 32 considered, the average difference between the actual and predicted JRD-pairs is -2.12 on the proposed JRD test dataset, and the MAE is 2.41.

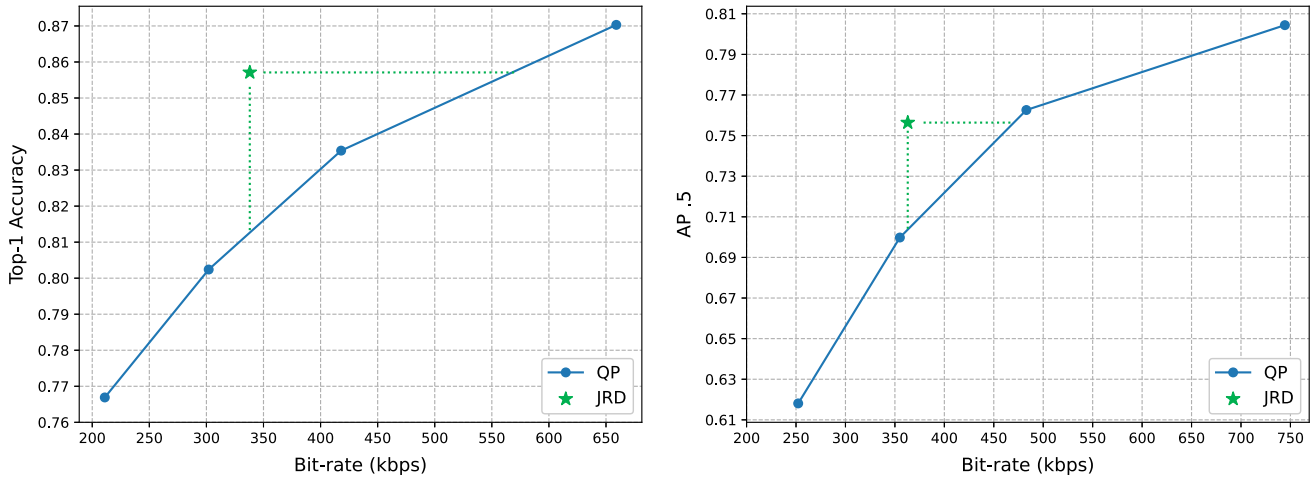
To prove the effectiveness of the predicted JRD, we choose the Faster R-CNN with ResNet-101 as the feature extractor to detect the objects on the COCO val2017 dataset where the images will be compressed with JRDs and different QPs. For images that are predicted to have smaller JRDs than QP 32, they will be compressed with QP 27. The detection model is trained on the pristine COCO train2017 dataset, hence will not fit with any distorted image data. We use the same strategy introduced in Sect. 4.2 to adopt the predicted JRD and compare the detection performance with actual JRD and several other QPs, which are presented in Table 10.

Table 9 Class weights settings and accuracy of binary classification models for object detection JRD prediction

q_k	32	38	41	43	45	47	49	51
W_{pos}	0.14	0.32	0.46	0.69	1.0	1.0	1.0	1.0
W_{neg}	1.0	1.0	1.0	1.0	0.96	0.48	0.22	0.1
Acc_{pos}	0.8622	0.8510	0.8457	0.8725	0.8299	0.8412	0.8365	0.8352
Acc_{neg}	0.8066	0.8145	0.8279	0.8624	0.8403	0.8386	0.8411	0.8430
Acc_{all}	0.8567	0.8439	0.8406	0.8686	0.8352	0.8395	0.8402	0.8423

Table 10 Bit-rate and object detection performance under predicted JRD, actual JRD and different QPs

QP	Predicted JRD	Actual JRD	18	32	38	41	43	45
AP .5	0.7564	0.8252	0.8620	0.8456	0.8044	0.7626	0.6998	0.6181
Bit-rate (kbps)	363	396	6349	1629	744	483	355	252



(a) Image classification **(b)** Object detection

Fig. 11 Machine vision performance under the predicted JRD and several comparable QPs

From Table 10, it can be observed that the bit-rate is between QP 41 and 43, closer to the latter, while the object detection performance is closer to the former after using predicted JRD as the QP for compression. If the changes of precision and bit-rate are regarded as approximately linear between QP 41 and 43, then about 22.77% of bit-rate can be saved under the same level of object detection performance, and 5.27% of performance improvement in absolute value can be achieved under the same level of bit-rate with predicted JRDs used. According to the experimental results, the image and video coding can be optimized based on the JRDs predicted by the proposed JRD prediction framework for higher object detection accuracy with even fewer bits.

From not only the larger prediction error but also the smaller rate-accuracy improvements achieved, it can be concluded that the JRDs of images for object detection are more difficult to predict than for image classification, even if an extra distorted image is referenced for prediction. Nevertheless, the proposed ensemble learning JRD prediction framework behaves acceptably for both two tested vision tasks, and according to a more visualized comparison in Fig. 11, the coding optimization based on predicted JRDs lead to better classification and detection performance, which jointly prove the effectiveness and practicability of the proposed JRD model. Furthermore, the framework can be extended by updating the binary classifiers with more powerful models as well as improving the searching strategy, and the object-region-wised QP adjustment based on predicted JRDs is adaptive to real-world coding patterns and rate-control methods, making it possible to adopt the JRD for several other machine vision applications than image classification and object detection.

Table 11 Image classification JRD prediction and application performance from different methods

	AD	MAE	Accuracy	Bit-rate
VGG w/o CWA	− 5.57	5.57	0.6056	76
VGG w/ CWA	− 4.07	7.39	0.7761	714
EL-VGGs w/o CWA	− 3.78	3.88	0.6815	163
EL-VGGs w/ CWA	− 1.54	1.90	0.8571	338
EL-VGGs w/ CWA & OR	− 1.39	1.66	0.8610	307

5.3 Discussion

5.3.1 Ensemble Learning and Class Weights Adjustment

In the proposed JRD prediction model, two approaches are adopted to improve the prediction accuracy, which are (1) replacing the straightforward VGG network with an ensemble-learning-based framework and (2) adjusting class weights for more balanced training and learning. To better understand how they contribute to increasing the JRD prediction performance, we make additional experiments on JRD prediction for image classification.

Firstly, we train the VGG-19 network again with and without class weights adjustment (CWA) respectively. For a fair comparison with the ensemble-learning-based JRD prediction framework, all of the images that have smaller JRDs than QP 38 will be categorized into one class such that the VGG-19 network will only output 8 classes. With CWA, the 8 classes of samples will contribute equally to optimizing the weights of VGG. Secondly, to verify the effectiveness of CWA, we train the binary classifiers in the JRD prediction framework again like in Sect. 5.1, but keep the weights for the backward gradients calculated from positive and negative samples as the same. Other training and testing setups are all the same as presented in Sect. 5.1. Table 11 shows the experimental results, where the second and third rows are the results from straightforward VGG and the next two rows are the results from ensemble learning (EL) VGGs. In Table 11, the AD is the average difference between actual and predicted JRD values, the image classification accuracy is top-1-ranked, and the bit-rate is measured by kbps. Furthermore, we also provide a more visualized comparison in Fig. 12.

It can be observed from Table 11 that a straightforward multi-class VGG-19 cannot be trained successfully to predict JRD without EL. Firstly, if CWA is not adopted, the significantly-imbalanced data distribution of the JRD dataset makes the network quickly biased to the class that has the largest sample count, which is the JRD of QP 51 in our case, thus applying predicted JRDs achieves the exact accuracy and bit-rate as QP 51. Secondly, even with CWA, the difference between neighboring JRD classes (such as JRD of QP 45 and 47) is so small that makes the JRD prediction problem more like a fine-grained classification task, which requires more powerful network structures to solve it well. In this situation, the JRD predictions from the multi-class VGG-19 are even worse reliable, making the JRD-guided coding and image classification performance not acceptable.

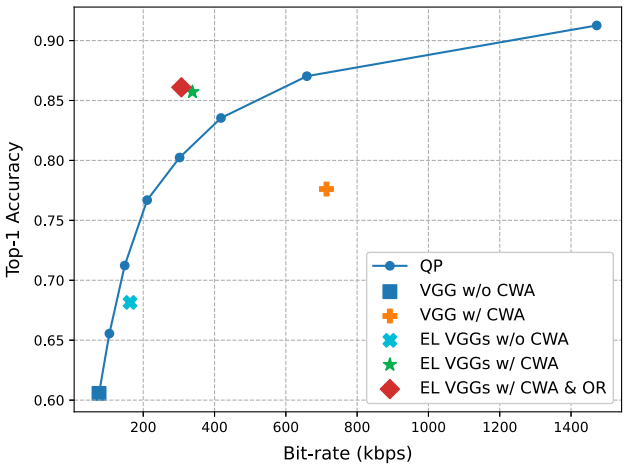


Fig. 12 Image classification JRD application performance comparison between different prediction methods

Similarly, without CWA, the EL VGGs cannot be trained successfully either because of the unequal sample counts between positive and negative classes. In our case, nearly every binary classifier prefers to predict the input into the positive category, especially before QP 47, which makes the predicted JRDs larger than actual ones. Although the image classification performance is much better than the straightforward VGG without CWA, it is worse than QP 47 while requires a larger bit-rate.

According to the experimental results and analysis above, the contribution to the JRD prediction and application performance improvement comes from both ensemble learning and class weights adjustment.

5.3.2 Non-Reference Versus One-Reference

In this paper, we focus on the JRD prediction for image classification under non-reference circumstances and for object detection under one-reference circumstances. To verify the effectiveness of the reference image, we perform another experiment on predicting JRD for the image classification

task. The same network structure as in Fig. 10 is adopted to replace the naive VGG-19 for the binary classifiers in the proposed JRD prediction framework. The selection of QP that is used to generate reference images follows the same principle that has been explained in Sect. 5.2. According to Sect. 4.1, nearly half of the images in the constructed JRD dataset have JRDs that are smaller than or equal to QP 45, and the other half have larger JRDs than QP 45, so the reference images will be compressed with QP 45. It should be mentioned that we exclude images whose JRDs equal to QP 51 for the JRD distribution analysis here. The reason is that these images are the majority in the JRD dataset, making it relatively easier for JRD prediction models to learn representative features from such a large number of samples. Other training and testing setups are all the same as presented in Sect. 5.1 for a fair comparison. The JRD prediction accuracy as well as the JRD-guided coding and image classification performance are shown on the last row of Table 11 where OR is short for one-reference. And a more visualized comparison with other methods is also shown in Fig. 12.

According to the experimental results, with one reference distorted image, the JRD prediction and application performances are both improved. More specifically, a higher image classification accuracy can be achieved with even fewer bits used for compression, which suggests the benefit of the reference. However, this method requires the encoder to encode the pristine image once before the actual compression, while the performance gain is not very significant compared with the non-reference method. Considering that the compression usually cost much time (which will be revealed soon in the next section), we prefer the non-reference JRD prediction to one-reference for more economical JRD-guided coding and machine vision applications.

5.3.3 Computation Complexity

The proposed ensemble-learning-based JRD prediction framework contains multiple binary classification networks. To estimate the JRD, the image should be sent to each partial classifier until one of them gives a positive inference as shown in Fig. 9. In order to test the computation complexity of the proposed JRD prediction method as well as the whole JRD-guided coding and machine vision application, we perform the experiments in Sects. 5.1 and 5.2 again and record the average time for processing each image input at every step. These steps include: (1) preparation (like generating reference distorted images, which is optional), (2) JRD prediction, (3) JRD-guided coding and (4) machine visual analysis. The experiments are performed on a PC that has an Intel i9-10900KF CPU and two NVIDIA RTX 2080Ti GPUs. The evaluation results for the two focused tasks, image classification (Cls) and Object detection (Det), are shown in Table 12 where the time cost is measured in milliseconds.

Table 12 Average time cost in milliseconds at each step of JRD-guided coding and machine vision application

Task	Step			
	1	2	3	4
Cls	–	19	1164	3
Det	1229	44	1195	94

According to the evaluation, the HEVC coding of step 1 and 3 costs the most time. Since in this paper we use the slow HEVC reference software and only perform intra-coding, such phenomenon can be expected. In contrast, the time cost of JRD prediction is almost negligible because we use the simple VGG-19 network and its variant to establish the prediction framework. Comparing the two tasks, the JRD prediction time for Det is slightly longer than Cls because the latter needs to extract features not only from the pristine image but also the reference distorted image. As for the last step, the tested Two-Stage Faster R-CNN for Det is more complicated than a naive ResNet for Cls, which makes the Det costs much more time than Cls.

In real-world applications, the efficiency of the overall system can be improved significantly by using faster encoders as well as adopting other common coding configurations like Random-Access or Low-Delay.

6 Conclusion

In this paper, we focus on optimizing image and video coding for machine vision with a new JRD concept. Firstly, on several large-scale and widely-used datasets, we sufficiently investigate how image and video coding influence the machine vision model performance for various important tasks. Secondly, based on the investigation, we verify the existence of an almost perfect balance between the coding cost and machine vision performance, which is described by the proposed JRD concept. For further researches and applications, we build a large-scale JRD-annotated dataset and analyze the factors of JRDs. Thirdly, to apply the JRD in practical situations, we study the prediction of JRD and establish an ensemble-learning-based JRD prediction framework. The framework is effective, efficient and extensible to infer JRDs for different machine vision tasks, which requires only the pristine images and a few or even no distorted images as input. According to extensive experiments, the predicted JRDs can guide the selection of coding parameters before compression, achieving remarkable machine vision performance and saving a large amount of bits at the same time, which proves the effectiveness of the proposed JRD model.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (62072008, 62025101), PKU-

Baidu Fund (2019BD003) and High-performance Computing Platform of Peking University, which are gratefully acknowledged.

References

Aqqa, M., Mantini, P., & Shah, S. K. (2019). Understanding how video quality affects object detection algorithms. In *VISIGRAPP (5: VISAPP)* (pp. 96–104).

Bross, B., Chen, J., & Liu, S. (2018). Versatile video coding (draft 5). JVET-K1001.

Chen, Y., Murherjee, D., Han, J., Grange, A., Xu, Y., Liu, Z., Parker, S., Chen, C., Su, H., Joshi, U., & Chiang, C. H. (2018). An overview of core coding tools in the av1 video codec. In *2018 picture coding symposium (PCS)* (pp. 41–45). IEEE.

Chen, Z., Fan, K., Wang, S., Duan, L., Lin, W., & Kot, A. C. (2019). Toward intelligent sensing: Intermediate deep feature compression. *IEEE Transactions on Image Processing*, *29*, 2230–2243.

Chou, C. H., & Li, Y. C. (1995). A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on Circuits and Systems for Video Technology*, *5*(6), 467–476.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Dodge, S., & Karam, L. (2016). Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)* (pp. 1–6), IEEE.

Dodge, S., & Karam, L. (2017). A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)* (pp 1–7). IEEE.

Duan, L. Y., Chandrasekhar, V., Chen, J., Lin, J., Wang, Z., Huang, T., et al. (2015). Overview of the mpeg-cdvs standard. *IEEE Transactions on Image Processing*, *25*(1), 179–194.

Duan, L. Y., Lou, Y., Bai, Y., Huang, T., Gao, W., Chandrasekhar, V., et al. (2018). Compact descriptors for video analysis: The emerging mpeg standard. *IEEE MultiMedia*, *26*(2), 44–54.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Fan, C., Lin, H., Hosu, V., Zhang, Y., Jiang, Q., Hamzaoui, R., & Saupe, D. (2019). Sur-net: Predicting the satisfied user ratio curve for image compression with deep learning. In *2019 eleventh international conference on quality of multimedia experience (QoMEX)* (pp. 1–6), IEEE.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., & Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. *Advances in Neural Information Processing Systems*, *31*, 7538–7550.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hu, T., Qi, H., Huang, Q., & Lu, Y. (2019). See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. arXiv preprint arXiv:1901.09891.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017a). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Huang, Q., Wang, H., Lim, S. C., Kim, H. Y., Jeong, S. Y., & Kuo, C. C. J. (2017b). Measure and prediction of hevc perceptually lossy/lossless boundary qp values. In: *2017 data compression conference (DCC)* (pp. 42–51). IEEE.

Jayant, N., Johnston, J., & Safranek, R. (1993). Signal compression based on models of human perception. *Proceedings of the IEEE*, *81*(10), 1385–1422.

Jin, L., Lin, J. Y., Hu, S., Wang, H., Wang, P., Katsavounidis, I., et al. (2016). Statistical study on perceived jpeg image quality via mcl-jci dataset construction and analysis. *Electronic Imaging*, *2016*(13), 1–9.

Li, Y., Jia, C., Wang, S., Zhang, X., Wang, S., Ma, S., & Gao, W. (2018). Joint rate-distortion optimization for simultaneous texture and deep feature compression of facial images. In *2018 IEEE fourth international conference on multimedia big data (BigMM)* (pp 1–5). IEEE.

Lin, H., Hosu, V., Fan, C., Zhang, Y., Mu, Y., Hamzaoui, R., et al. (2020). Sur-featnet: Predicting the satisfied user ratio curve for image compression with deep feature learning. *Quality and User Experience*, *5*, 1–23.

Lin, J. Y., Jin, L., Hu, S., Katsavounidis, I., Li, Z., Aaron, A., & Kuo, C. C. J. (2015). Experimental design and analysis of jnd test on coded image/video. In *Applications of digital image processing XXXVIII, International Society for optics and photonics* (Vol. 9599, p. 95990Z).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755). Springer.

Liu, D., Wang, D., & Li, H. (2017). Recognizable or not: Towards image semantic quality assessment for compression. *Sensing and Imaging*, *18*(1), 1.

Liu, H., Zhang, Y., Zhang, H., Fan, C., Kwong, S., Kuo, C. C. J., et al. (2019). Deep learning-based picture-wise just noticeable distortion prediction model for image compression. *IEEE Transactions on Image Processing*, *29*, 641–656.

Lou, Y., Duan, L. Y., Wang, S., Chen, Z., Bai, Y., Chen, C., et al. (2019). Front-end smart visual sensing and back-end intelligent analysis: A unified infrastructure for economizing the visual system of city brain. *IEEE Journal on Selected Areas in Communications*, *37*(7), 1489–1503.

Ma, S., Zhang, X., Wang, S., Zhang, X., Jia, C., & Wang, S. (2018). Joint feature and texture coding: Toward smart video representation via front-end intelligence. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(10), 3095–3105.

Redondi, A., Baroffio, L., Bianchi, L., Cesana, M., & Tagliasacchi, M. (2016). Compress-then-analyze versus analyze-then-compress: What is best in visual sensor networks? *IEEE Transactions on Mobile Computing*, *15*(12), 3000–3013.

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.

Schrimpf, M., Kubilius, J., Hong, H., Majaj, N. J., Rajalingham, R., Issa, E. B., Kar, K., Bashivan, P., Prescott-Roy, J., Schmidt, K., et al. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like? BioRxiv p. 407007.

Shi, J., & Chen, Z. (2020). Reinforced bit allocation under task-driven semantic distortion metrics. In *2020 IEEE international symposium on circuits and systems (ISCAS)* (pp. 1–5). IEEE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Skodras, A., Christopoulos, C., & Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, *18*(5), 36–58.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, *23*(5), 828–841.

Sullivan, G. J., Ohm, J. R., Han, W. J., & Wiegand, T. (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(12), 1649–1668.

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946.

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The caltech-ucsd birds-200-2011 dataset.

Wang, H., Gan, W., Hu, S., Lin, J. Y., Jin, L., Song, L., Wang, P., Katsavounidis, I., Aaron, A., & Kuo, C. C. J. (2016). Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset. In *2016 IEEE international conference on image processing (ICIP)* (pp. 1509–1513). IEEE.

Wang, H., Katsavounidis, I., Zhou, J., Park, J., Lei, S., Zhou, X., et al. (2017). Videoset: A large-scale compressed video quality dataset based on jnd measurement. *Journal of Visual Communication and Image Representation*, *46*, 292–302.

Wang, H., Katsavounidis, I., Huang, Q., Zhou, X., & Kuo, C. C. J. (2018a). Prediction of satisfied user ratio for compressed video. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6747–6751). IEEE.

Wang, H., Zhang, X., Yang, C., & Kuo, C. C. J. (2018b). Analysis and prediction of jnd-based video quality model. In *2018 picture coding symposium (PCS)* (pp 278–282). IEEE.

Wang, S., Wang, S., Yang, W., Zhang, X., Wang, S., Ma, S., & Gao, W. (2020). Towards analysis-friendly face representation with scalable feature and texture compression. arXiv preprint arXiv:2004.10043.

Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the h. 264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 560–576.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500.

Yamins, D. L., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, *19*(3), 356–365.

Yang, X., Ling, W., Lu, Z., Ong, E. P., & Yao, S. (2005). Just noticeable distortion model and its applications in video coding. *Signal Processing: Image Communication*, *20*(7), 662–680.

Zhang, J., Jia, C., Lei, M., Wang, S., Ma, S., & Gao, W. (2019). Recent development of avs video coding standard: Avs3. In *2019 picture coding symposium (PCS)* (pp. 1–5). IEEE.

Zhang, X., Ma, S., Wang, S., Zhang, X., Sun, H., & Gao, W. (2016). A joint compression scheme of video feature descriptors and visual content. *IEEE Transactions on Image Processing*, *26*(2), 633–647.

Zhang, X., Yang, C., Wang, H., Xu, W., & Kuo, C. C. J. (2020). Satisfied-user-ratio modeling for compressed video. *IEEE Transactions on Image Processing*, *29*, 3777–3789.

Zhou, X., Wang, D., & Krähenbühl, P. (2019). Objects as points. arXiv preprint arXiv:1904.07850.