A Perspective on Deep Vision Performance
with Standard Image and Video Codecs

Christoph Reich[1,2,3,5]   Oliver Hahn[1]   Daniel Cremers[2,5]   Stefan Roth[1,4]   Biplob Debnath[3]

[1]TU Darmstadt [2]TU Munich [3]NEC Laboratories America, Inc.
[4]Hessian Center for AI (hessian.AI) [5]Munich Center for Machine Learning (MCML)
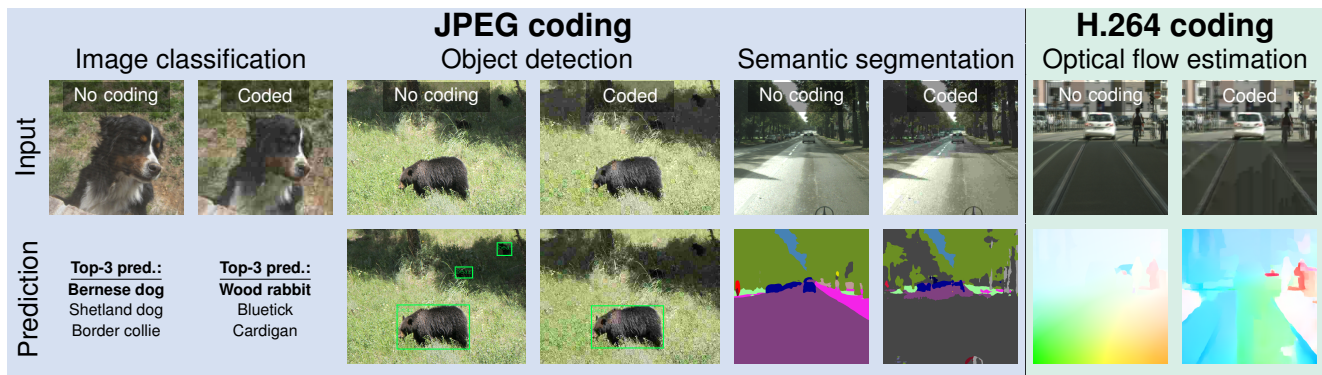
Figure 1. **Deep vision performance on standard coded images/videos.** We demonstrate that using standard image and video coding can vastly deteriorate the accuracy of current deep vision models. We visualize the original *(left)* and coded *(right)* image/video and the respective models' prediction. For optical flow estimation, we overlay the first and second frame. Best viewed in color; zoom in for details.

Abstract

Resource-constrained hardware, such as edge devices or cell phones, often rely on cloud servers to provide the required computational resources for inference in deep vision models. However, transferring image and video data from an edge or mobile device to a cloud server requires coding to deal with network constraints. The use of standardized codecs, such as JPEG or H.264, is prevalent and required to ensure interoperability. This paper aims to examine the implications of employing standardized codecs within deep vision pipelines. We find that using JPEG and H.264 coding significantly deteriorates the accuracy across a broad range of vision tasks and models. For instance, strong compression rates reduce semantic segmentation accuracy by more than 80% in mIoU. In contrast to previous findings, our analysis extends beyond image and action classification to localization and dense prediction tasks, thus providing a more comprehensive perspective.

1. Introduction

Low-resource compute environments pose a particular challenge when analyzing images or videos. While efficient deep vision models for limited resource settings have been proposed, extensive image and video analysis is often facilitated using powerful cloud servers [5, 29]. In particular, for dense prediction tasks or in the presence of high-resolution imagery, edge or mobile devices transfer the respective data to cloud servers for inference [12, 16, 19]. Data transfer is typically facilitated using wireless network connections.

Transferring image and video data over a network requires the use of coding to cope with network constraints, such as bandwidth limitations and fluctuations, to prevent data corruption [19]. To ensure low cost as well as interoperability, standardized codecs are the *de facto* standard in real-world image and video processing pipelines [17, 23]. However, computationally efficient standard codecs, such as JPEG [46] or H.264 [37, 47], utilize lossy compression, leading to image/video distortions (*cf*. Fig. 1).

While standard codecs had been developed prior to the deep learning era, targeting perceptual quality for humans [47], they were naïvely incorporated into deep vision pipelines [12]. In this paper, we analyze the effect of standard image and video codecs on the predictive accuracy of common downstream deep vision models. While recent work demonstrated the effect of standard-coded image and video data on the task of action recognition [32]

Table 1. **Overview of the 23 evaluated models.** We provide both the source and unique identifier of each model used in our experiments.

Model	Source	Identifier
Semantic segmentation		
DeepLabV3 (ResNet-18) [6]	MMSegmentation [31]	`deeplabv3_r18-d8_4xb2-80k_cityscapes-769x769`
DeepLabV3 (ResNet-50) [6]	MMSegmentation [31]	`deeplabv3_r50-d8_4xb2-40k_cityscapes-769x769`
DeepLabV3 (ResNet-101) [6]	MMSegmentation [31]	`deeplabv3_r101-d8_4xb2-40k_cityscapes-769x769`
UPerNet (ResNet-50) [49]	MMSegmentation [31]	`upernet_r50_4xb2-40k_cityscapes-769x769`
UPerNet (ResNet-101) [49]	MMSegmentation [31]	`upernet_r101_4xb2-40k_cityscapes-769x769`
Object detection		
DETR (ResNet-50) [3]	Official repository [3]	`detr_resnet50`
DETR (ResNet-101) [3]	Official repository [3]	`detr_resnet101`
DETR (ResNet-101-DC5) [3]	Official repository [3]	`detr_resnet101_dc5`
Faster R-CNN (ResNet-18) [36]	TorchVision [45]	`fasterrcnn_resnet50_fpn(weights=DEFAULT)`
Faster R-CNN V2 (ResNet-18) [24]	TorchVision [45]	`fasterrcnn_resnet50_fpn_v2(weights=DEFAULT)`
Faster R-CNN (MobileNetV3 L) [15, 36]	TorchVision [45]	`fasterrcnn_mobilenet_v3_large_fpn(weights=DEFAULT)`
Image classification		
ResNet-18 [13]	timm [48]	`resnet50(pretrained=True)`
ResNet-50 [13]	timm [48]	`resnet101(pretrained=True)`
ResNet-101 [13]	timm [48]	`resnet152(pretrained=True)`
ViT-T [11]	timm [48]	`vit_tiny_patch16_224(pretrained=True)`
ViT-S [11]	timm [48]	`vit_small_patch16_224(pretrained=True)`
ViT-B [11]	timm [48]	`vit_base_patch16_224(pretrained=True)`
Swin-T [26]	timm [48]	`swin_tiny_patch4_window7_224(pretrained=True)`
Swin-S [26]	timm [48]	`swin_small_patch4_window7_224(pretrained=True)`
Swin-B [26]	timm [48]	`swin_base_patch4_window7_224(pretrained=True)`
Optical flow estimation		
RAFT large [42]	TorchVision [45]	`raft_large(weights=Raft_Large_Weights.C_T_V2)`
RAFT small [42]	TorchVision [45]	`raft_small(weights=Raft_Small_Weights.C_T_V2)`
SMURF (RAFT large) [42]	Off. repo. [41] & TorchVision [45]	`raft_large(weights=Weights_From_Off_Repo)`

and image classification [14, 40], we provide results on a wide range of different computer vision tasks, namely image classification, object detection, optical flow estimation, and semantic segmentation for a variety of approaches (*cf*. Fig. 1 & Tab. 1). We observe that all models tested significantly suffer from coding at inference time. In particular, standard coding leads to a significant decrease in model performance, especially for localization and dense prediction tasks (*cf*. Fig. 1). When utilizing vast compression, required for limited bandwidth availability, the accuracy of deep vision methods completely breaks down (*cf*. Fig. 1). For instance, when applying strong JPEG compression, the (absolute) ImageNet-1k [38] classification accuracy of a Swin-B [26] Transformer drops from about 83% to below 20%.

We also discuss the limitations of recent approaches, such as optimizing standard codecs w.r.t. deep vision models [12, 27, 28]. Based on the presented results and recent work, we close with a discussion of the challenges and possible solutions to overcome the vast deterioration of the predictive performance of deep vision models when employing standardized coding. With this, we hope to facilitate the development of novel approaches towards optimized coding for deep vision models within the scope of standardization.

2. Experiments

We aim to analyze the impact of standard image and video coding on the predictive performance of current deep vision models. Using the original model weights of the respec-

tive methods, we experiment with coded data during inference. Despite the availability of more advanced coding approaches, we use JPEG and H.264 coding as these are still the most commonly used codecs in real-world image/video processing pipelines [17, 23].

Standard codecs. Real-world image and video processing pipelines predominantly utilize standard codecs for transferring or storing image and video data to ensure interoperability. JPEG [46] and H.264 [37, 47] are among the most widely used standard codecs [17, 23]. The popularity of both codecs in real-world applications is caused by their computational efficiency, acceptable rate-distortion trade-off, and widespread hardware and software support. Both codecs utilize quantization in the frequency domain to perform coding. By adjusting the quantization strength rate, the file size or required bandwidth can be traded off against the image distortion. While the rate-distortion trade-off of standard codecs has been studied extensively through the lens of Shannon's rate-distortion theory and by perceptual quality [2, 39], we analyze the implications of standard codecs on the predictive performance of current deep vision models across a variety of downstream tasks.

Evaluation approach. We aim to measure the effect of image and video coding on the accuracy of deep vision models, isolated from the models' respective baseline. To this end, we utilize the prediction on the original image as a reference, *i.e.*, pseudo-ground truth and not the ground-truth label. If not stated differently, we perform inference
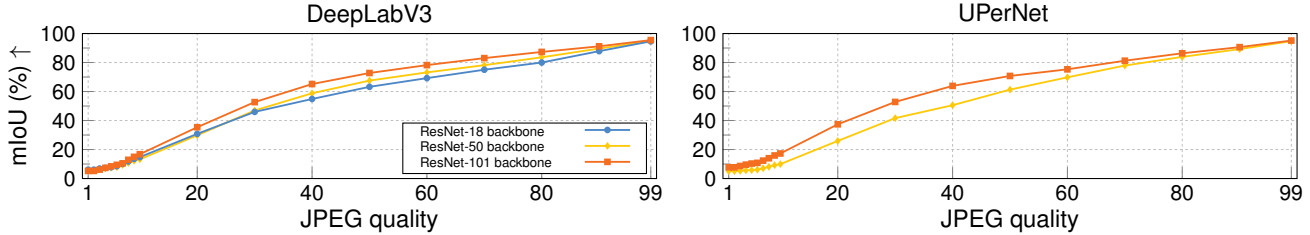
Figure 2. **Relative semantic segmentation accuracy on JPEG-coded Cityscapes (val) dataset.** The accuracy of all models vastly decreases as the compression rate increases (lower JPEG quality). ResNet-18 backbone in blue (■), ResNet-50 backbone in yellow (■), and ResNet-101 backbone in orange (■). Best viewed in color.

using the original (non-coded) image or video and use the models' prediction as the pseudo label. In the case of classification tasks, *e.g.*, semantic segmentation, we generate the pseudo labels by taking the $\arg\max$ of the models' output. We then measure the difference of the predictions, obtained on coded images or videos, w.r.t. the pseudo labels, resulting in a relative score on the respective downstream metric. This leads to a normalized and easy-to-compare measurement which reflects the amount of downstream performance in terms of accuracy lost due to coding. For all image classification methods, we also provide absolute scores w.r.t. the ground truth labels on ImageNet-1k [38], demonstrating the validity of our pseudo-label-based evaluation approach.

Depending on the vision task, we utilize task-specific metrics. For image classification, we report both the top-1 and top-5 accuracy (Acc). The mean average precision (mAP) is used to evaluate object detection. We also report mAP values using fixed intersection-over-union thresholds of 50% and 75%. To analyze semantic segmentation predictions, we use the mean intersection-over-union (mIoU) evaluation metric, computing the class-wise IoU before averaging over all classes. Finally, to evaluate optical flow estimation, we employ the average end-point error (EPE).

Experimental setup. All utilized models are trained using their respective *standard training protocols*. More precisely, we use publicly available pre-trained model weights of the respective methods (*cf*. Tab. 1). For semantic segmentation, we use the DeepLabV3 [6] and UPerNet [49] models from MMSegmentation [31]. We vary the backbone between a ResNet-18 (only for DeepLabV3), 50, and 101 [13]. Semantic segmentation experiments are performed on the Cityscapes [10] validation dataset. We follow the training resolution of the utilized models and evaluate using a resolution of 769×769 pixels. The Detection Transformer (DETR) [3] model (w/ ResNet-50, 101 & 101-DC5 backbones) from the official repository is used for object detection experiments. For the Faster R-CNN (V1 & V2) [24, 36], we utilize the checkpoints from TorchVision (w/ ResNet-50 and MobileNetV3 L [15] backbones). Object detection experiments are carried out on the Common Objects in Context (COCO) [25] dataset. We follow com-

mon evaluation and resize the images to a resolution of 480×640 before performing coding. For the task of image classification, we employ different ResNet [13], Vision Transformer [11] (ViT), and Swin Transformer [26] models from timm [48], trained on the ImageNet-1k [38] dataset. Following the timm evaluation protocol, we first resize the smaller image dimension to 256 pixels and keep the aspect ratio. Finally, we center-crop the images to a resolution of 224^2 before performing coding. Optical flow estimation experiments are carried out on the supervised RAFT (large & small) [42] from TorchVision [45] and the unsupervised SMURF [41] model (RAFT large architecture). We perform optical flow estimation on the Cityscapes validation sequences [10] using a clip length of eight frames with a temporal stride of 3. We center-crop each video frame to a resolution of 512×512 before performing video coding. In Tab. 1, we provide a full overview of the utilized models.

2.1. Predictive performance of deep vision models on JPEG-coded images

We analyze the effect of JPEG coding on the predictive performance of deep vision models using different compression rates. JPEG offers to control the rate using the JPEG quality parameter. More specifically, the JPEG quality controls the quantization strength of the discrete cosine transform (DCT) coefficients and ranges from 1 to 99. A low JPEG quality corresponds to vast compression, leading to a small JPEG file size at the cost of image distortion. *Vice versa*, a high JPEG quality leads to less compression.

Semantic segmentation. Fig. 2 demonstrates results of three different DeepLabV3 models [6] for JPEG-coded versions of Cityscapes [10] validation images. For weak compression rates (JPEG quality > 90), all models are able to largely maintain semantic segmentation accuracy. However, as the compression rate increases (JPEG quality reduces), the semantic segmentation accuracy constantly deteriorates. In the case of maximum compression (JPEG quality = 1), the semantic segmentation accuracy completely breaks down with an mIoU of below 10% relative to the baseline output on non-coded images. While all models struggle to maintain accuracy, a large backbone network
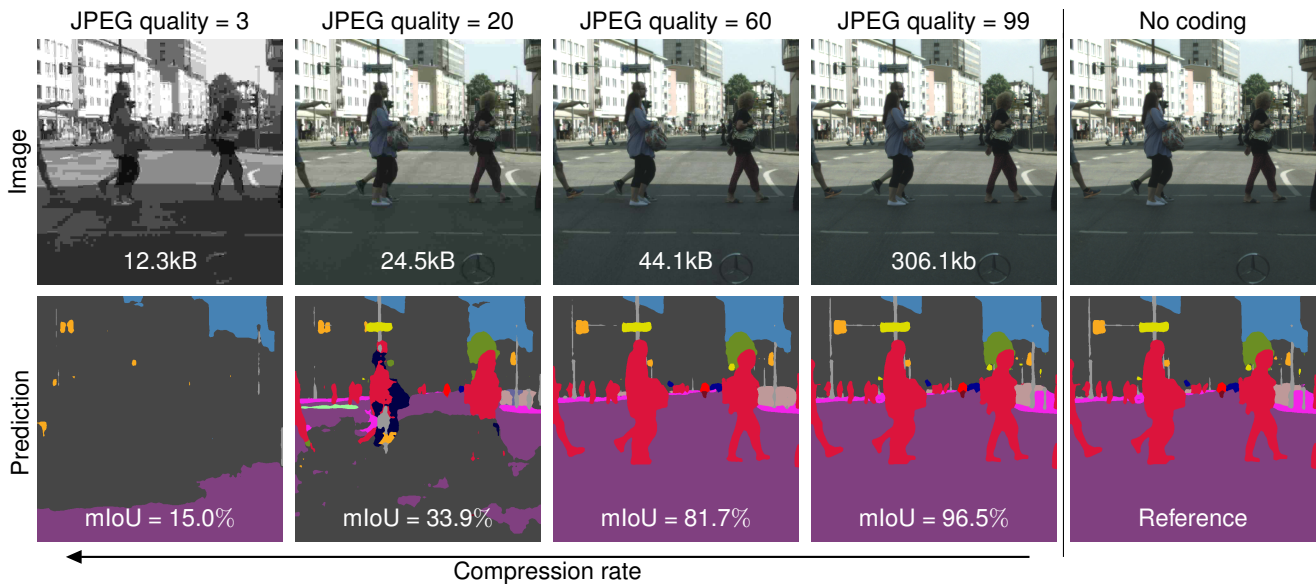
Figure 3. **Qualitative semantic segmentation example on a JPEG-coded Cityscapes (val) images.** As the compression rate increases (lower JPEG quality), the DeepLabV3 (w/ ResNet-18 backbone) model is not able to maintain its accuracy. For a JPEG quality of 3, the semantic segmentation accuracy completely breaks down. The mIoU relative to the uncoded baseline prediction is computed on a per-image basis. We report the JPEG file size for the coded images. Best viewed in color; zoom in for details.
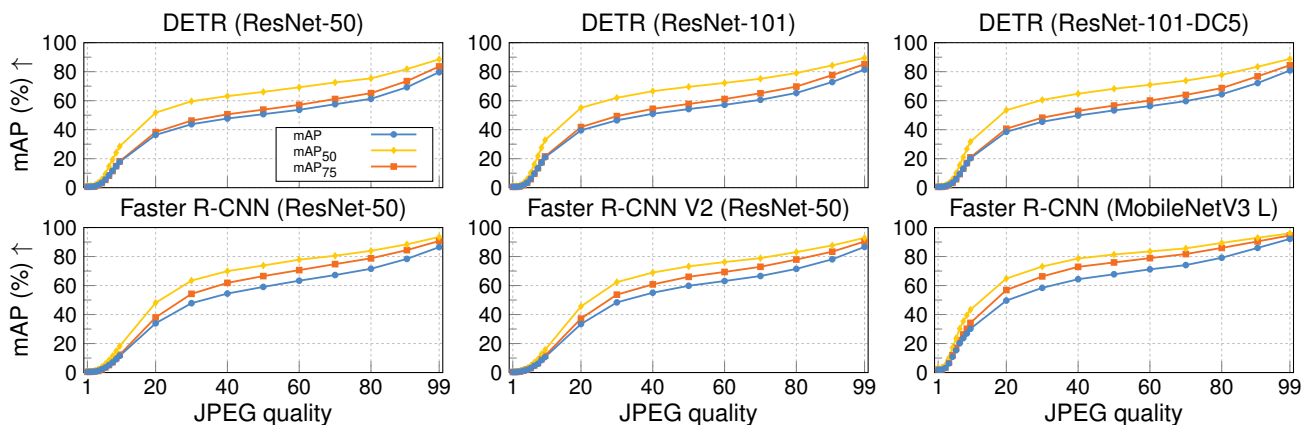

Figure 4. **Relative object detection accuracy on the JPEG-coded COCO (val) dataset.** The accuracy of all DETR and Faster R-CNN variants vastly deteriorates as the compression rate increases (lower JPEG quality). We report the mAP in blue (■), the mAP_{50} in yellow (■), and the mAP_{75} in orange (■). Best viewed in color.

(*e.g.*, a ResNet-152) seems to offer slightly more robustness to JPEG coding especially for lower compression strengths.

We also analyze semantic segmentation results qualitatively. Fig. 3 presents semantic segmentation predictions from a DeepLabV3 (w/ ResNet-18 backbone) on a non-coded image (right) and JPEG-coded images (left). While a JPEG quality of 99 (least compression) leads to minimal changes in the segmentation map, a JPEG quality of 60 already leads to some misclassification (*e.g.* "sidewalk" and "pole" classified as "fence"). A JPEG quality of 20 significantly deteriorates the semantic segmentation accuracy,

resulting in the misclassification of large image sections as well as major classes such as "road". Some objects are still segmented accurately but most parts of the segmentation prediction suffer severely from JPEG coding. In contrast, the visual appearance of the coded image is rather similar to the original. A JPEG quality of 3 completely deteriorates the predictive performance of the semantic segmentation where the vast majority of pixels are wrongly classified.

Object detection. We analyze the accuracy of three DETR [3] as well as three Faster R-CNN [24, 36] models on JPEG-coded COCO [25] validation images. The object
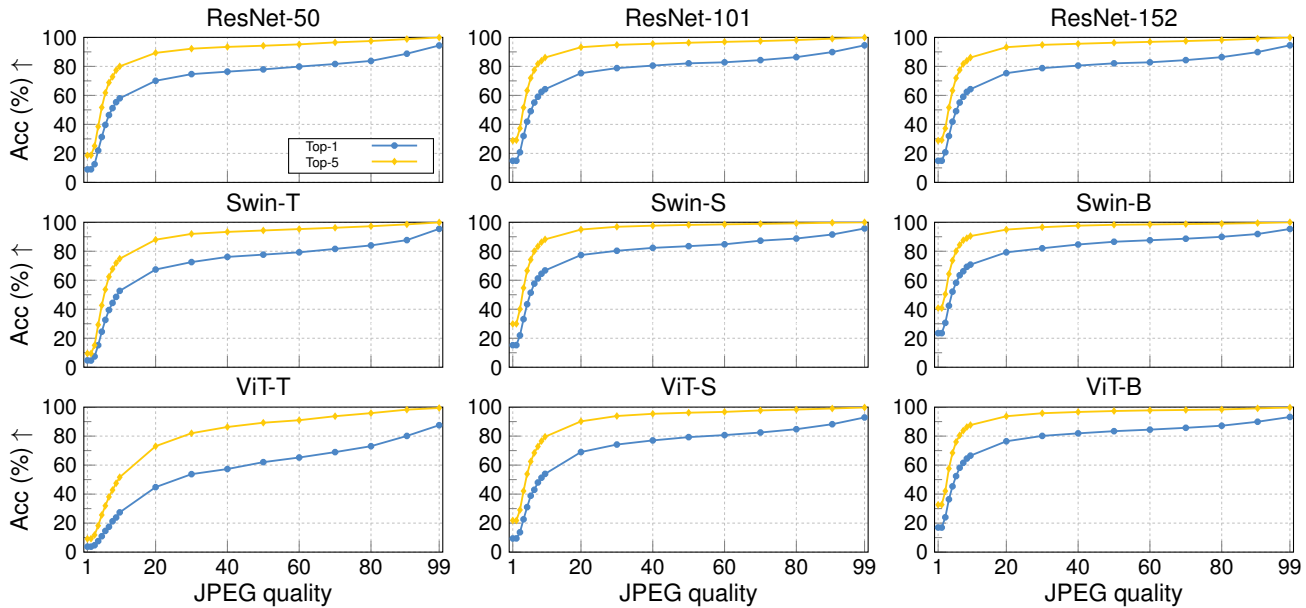
Figure 5. **Relative image classification accuracy on the JPEG-coded ImageNet-1k (val) dataset.** The relative accuracy (*i.e.*, w.r.t. the pseudo labels) of all models vastly deteriorates as the compression rate increases (lower JPEG quality). We report the top-1 accuracy in blue (■) and the top-5 accuracy in yellow (■). Best viewed in color.
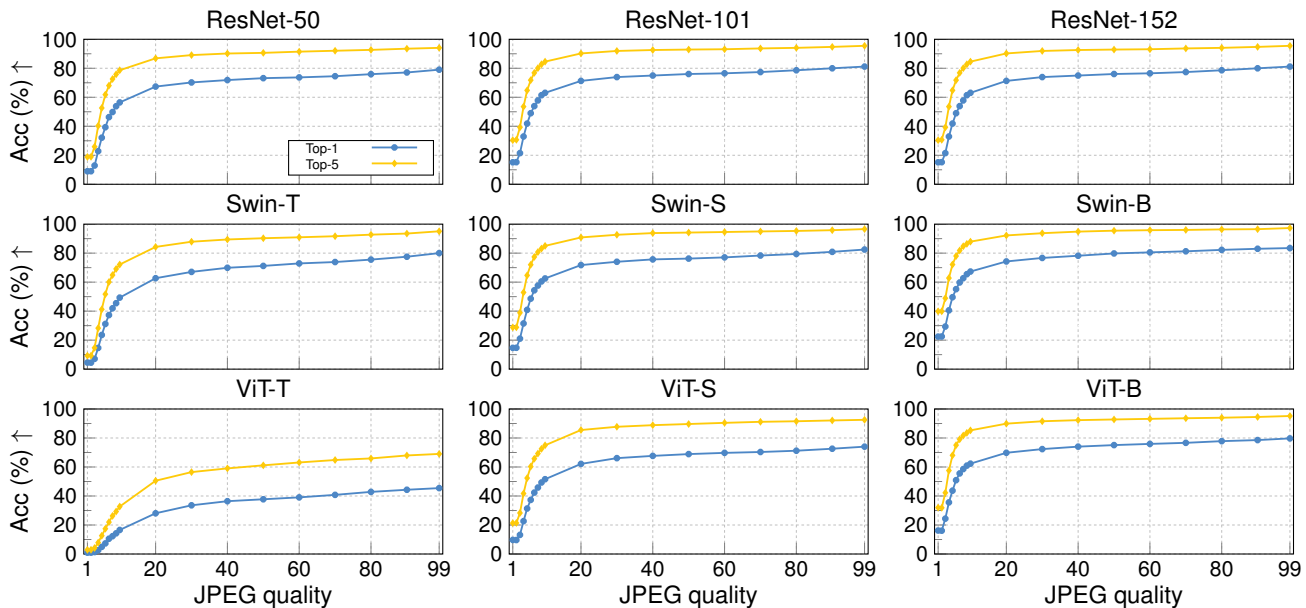

Figure 6. **Absolute image classification accuracy on the JPEG-coded ImageNet-1k (val) dataset.** The absolute accuracy (*i.e.*, w.r.t. the ground-truth labels) of all models vastly deteriorates as the compression rate increases (lower JPEG quality). We report the top-1 accuracy in blue (■) and the top-5 accuracy in yellow (■). The general trend of the absolute scores aligns well with the relative scores reported in Fig. 5. Best viewed in color.

detection performance of both models is presented in Fig. 4. Similar to the task of semantic segmentation, we observe a strong drop in object detection performance of both models when the compression rate is increased. Even for weak compression rates (high JPEG quality values), the object

detection performance is negatively impacted by JPEG coding. For strong compression rates (JPEG quality < 5), the performance drops below an mAP score of 5% relative to the prediction on the uncoded images. Compared to semantic segmentation, the performance of the models remains

slightly more stable for medium compression rates, which we attribute to the more spatially precise nature of the task. While only leading to a minor increase in robustness, a large backbone leads to a slight improvement on coded images. Notably, Faster R-CNN seems to suffer less in terms of relative object detection accuracy from medium compression strengths (JPEG quality within 30 to 80) than DETR when using a ResNet-50 backbone (*cf*. Fig. 4 (*left column*)).

Image classification. In Fig. 5, we present relative image classification results on JPEG-coded ImageNet-1k [38] images. We tested three ResNet models [13], three Swin models [26] as well as three ViT models [11]. The performance of all models decreases slightly for compression rates up to around 30. However, for medium JPEG qualities, the deterioration is not as severe as for semantic segmentation (*cf*. Fig. 2) and object detection (*cf*. Fig. 4). Comparing the model behaviors with those of semantic segmentation or object detection models, image classification performance remains largely stable for a wide range of compression rates, while the opposite occurs for the more granular tasks. The high sensitivity of dense prediction and localization tasks compared to image classification indicates that the global information is largely preserved in coded images, whereas coding leads to erroneous local representations.

We also report the absolute image classification performance in Fig. 6. While a weak compression (JPEG quality > 95) maintains most of the models' accuracy, a vast compression (JPEG quality < 20) leads to a severe drop in classification accuracy. A moderate JPEG quality of 80 leads to a 3.1% drop in image classification accuracy when using a ResNet-50, relative to no coding (standard validation accuracy 79.0%) [13, 48]. A very weak compression rate (JPEG quality of 90) reduces the image classification accuracy still by 1.9%. Considering the more incremental recent improvements in ImageNet classification accuracy [48], this is a significant deterioration. More importantly, we show that the models perform analogously in terms of absolute numbers when compared to our evaluation approach using the prediction of the non-coded images as pseudo labels. Yet, our evaluation approach leads to normalized results that are easier to compare and interpret between models.

2.2. Predictive performance of deep vision models on H.264-coded videos

We analyze the impact of H.264 video coding [47] on the predictive performance of deep vision models. To control the compression rate, H.264 offers support for different quantization parameters (QP). QP ranges from 0 to 51, where 0 corresponds to the least compression. In contrast, a QP of 51 leads to strong compression, resulting in significant video distortion at the benefit of a small rate. As we utilize validation clips of Cityscapes [10] of eight frames, the group of images for H.264 coding is also set to
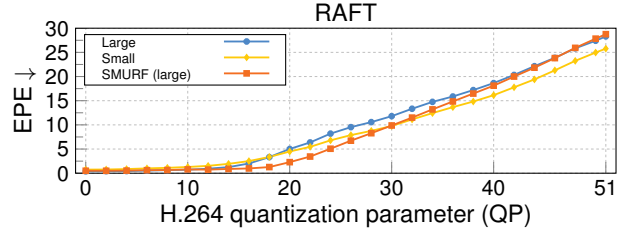

Figure 7. **Relative accuracy of optical flow on H.264-coded Cityscapes (val) clips.** Optical flow estimation of all three models vastly deteriorates as the compression rate increases (higher H.264 QP). We report the average EPE of RAFT large in blue (■), of RAFT small in yellow (■), and of SMURF (RAFT large unsupervised) in orange (■), relative to the respective baseline output. Best viewed in color.

eight. All other H.264 parameters are set to their respective default value [44]. In particular, we use the FFmpeg command `ffmpeg -y -i frame_%10d.png -preset medium -g 8 -qp QP -flags +cgop+mv4 -mbtree 0 -aq-mode 0 -b_strategy 0 -psy 0 video.mp4` and utilize different QP values.

Optical flow estimation. In Fig. 7, we quantitatively evaluate the optical flow estimation performance of (supervised) RAFT (large & small) [42] and (unsupervised) SMURF [41]. Note that the task of optical flow estimation relies on two consecutive frames, and thus also depends on the temporal quality of the coded clips. Optical flow estimation. In general, optical flow estimation performance for low QP values (below 15) remains fairly stable. As QP increases (QP > 15) optical flow estimation deteriorates linearly. For QP = 51 (strongest compression), the EPE of all models exceeds 25 w.r.t. the baseline output. While RAFT small offers an overall weaker optical flow estimation performance than RAFT large [42] (w.r.t. ground truth optical flow), the small RAFT variant tends to be more robust against H.264 coding than RAFT large, relative to the baseline output (pseudo label). Interestingly, while sharing the same architecture (RAFT large), unsupervised SMURF offers more robustness against H.264 than supervised RAFT large for QP values between 15 to 40. We attribute this slight improvement in robustness to the usage of strong data augmentation within the SMURF training approach [41].

We also qualitatively analyze the optical flow estimation performance of (supervised) RAFT large [42] on an H.264 coded video clip in Fig. 8. A QP of 0 as well as 20 only leads to a negligible change in the predicted optical flow map. Increasing QP to 35 leads to a notable deterioration of the predicted optical flow, especially in low-texture regions, such as the road. The apparent motion of object instances (*e.g*., car and cyclist) is still accurately estimated. For a strong compression rate (QP = 50), also the estimated optical flow of moving object instances vastly deteriorates.
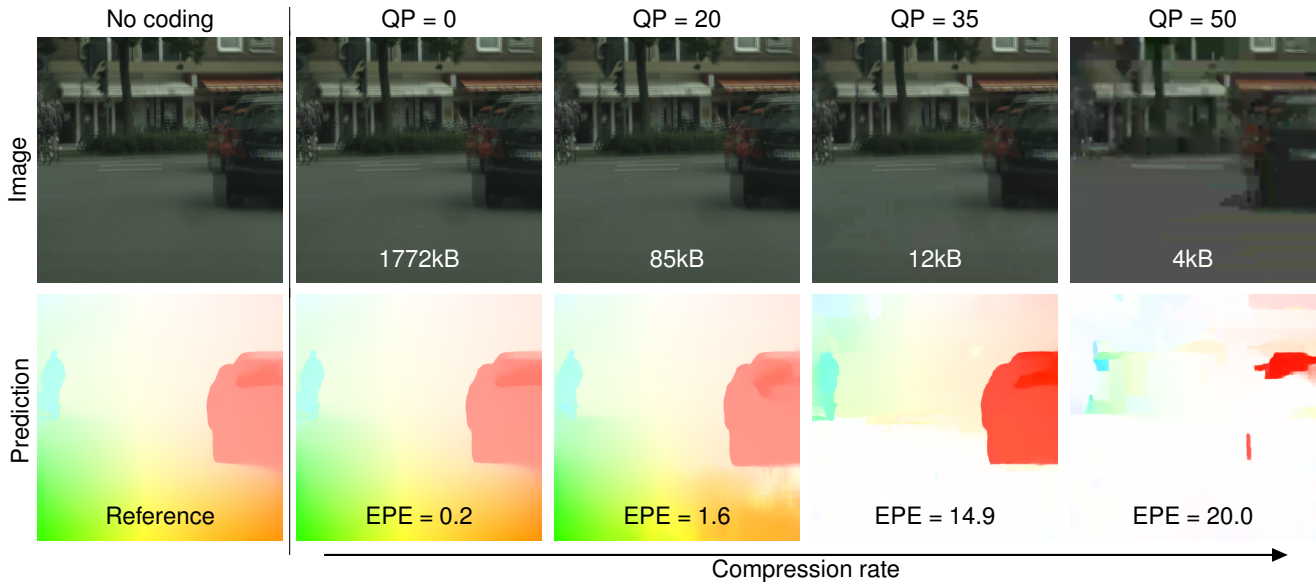
Figure 8. **Qualitative optical flow estimation example on H.264-coded Cityscapes (val) clips.** As the compression rate increases (higher QP), the optical flow estimation of RAFT large deteriorates. For a QP of 35, the predicted optical flow significantly deteriorates before completely breaking down for a QP of 50. We report the file size for the full 8-frame clip. We visualize the overlay of the first and second frame. Classical optical flow color encoding used as proposed by Baker *et al.* [1]. Best viewed in color; zoom in for details.
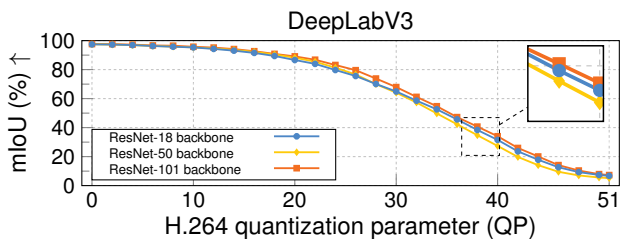

Figure 9. **Relative accuracy of semantic segmentation on H.264-coded Cityscapes (val) clips.** The accuracy of all DeepLabV3 models vastly deteriorates as the compression rate increases (higher QP). We report the mIoU for different backbones (ResNet-18 in blue ■, 50 in yellow ■ & 101 in orange ■), relative to the baseline output. Best viewed in color.

Semantic segmentation. In Fig. 9, we report quantitative semantic segmentation results of three different DeepLabV3 [6] models on H.264-coded video clips. H.264 coding leads to a similar deterioration of segmentation accuracy as JPEG coding (*cf*. Fig. 2). For low H.264 quantization parameters of up to approximately 15, the segmentation performance in mIoU is only affected slightly, whereas relative semantic segmentation accuracy decreases considerably in the subsequent medium and high compression range. For a QP above 40, the mIoU relative to the baseline output drops even below 40%. For semantic segmentation on JPEG-coded images, we observed an increase in robustness against coding for larger backbones (*cf*. Fig. 2). In the case of H.264 coded videos, all backbones (ResNet-

18, 50 & 101) suffer similarly from coding. In particular, DeepLabV3 with a ResNet-101 backbone offers better robustness against H.264 coding than the DeepLabV3 variant with a ResNet-18 backbone. The ResNet-50 DeepLabV3 variant suffers the most from H.264 coding.

3. Discussion

Relying on cloud servers for extensive processing of image and video data can, in principle, overcome bottlenecks from limited compute resources in mobile or edge devices. However, this approach requires the efficient and standardized transmission of image and video data. As we showed, using standardized codecs to facilitate efficient transmission can lead to a significantly reduced downstream deep vision performance. In our experiments, all methods suffered significantly in performance across a variety of vision tasks. Next, we will discuss different avenues to overcome the negative impact of standardized coding on deep vision models.

3.1. Optimizing standard codecs

Recent approaches have aimed to optimize the codec's parameters for deep vision models without breaking standardization. Luo *et al.* [27] proposed optimized JPEG quantization tables for image classification using end-to-end learning. This approach, however, requires the typically non-differentiable image codec to be differentiable. While it is feasible to find differentiable surrogates of simple codecs, such as JPEG [35, 40], an increasing complexity of the im-

age or video codec makes developing or learning a differentiable codec surrogate more challenging [18, 43]. Du *et al*. [12] proposed an approach to optimize the allocation of the H.264 quantization parameters for vision models, which does not require a differentiable codec. Learning is performed on a saliency-based proxy task. However, learning on a proxy task can introduce unique challenges. The approach is also tailored to the task of object detection. A promising avenue for future research could be to build more powerful codec surrogate models to circumvent proxy task training. Alternatively, the use of reinforcement learning could overcome the need for a differentiable codec surrogate [28]. Additionally, offering support for rate control and multiple computer vision tasks in these approaches could facilitate a potentially widespread application of methods augmenting the parameters of standard codecs.

3.2. Deep codecs for deep vision models

Deep codecs aim to learn image and video coding using deep neural networks [30, 50]. While reaching a better rate-distortion trade-off than standard codecs and offering support for custom objectives [4, 7, 8, 21] (*e.g*., maintaining the accuracy of a downstream deep vision model), these approaches entail three major limitations so far. First, no standardization for deep codecs has emerged yet, severely limiting their application in real-world applications. Second, deep codecs only offer limited support for rate control. Finally, deep codecs are significantly less computationally efficient than standard codecs. This poses challenges, especially in low-resource environments. While offering great support for custom downstream objectives, optimizing a deep codec for multiple downstream vision tasks and models is non-trivial [7]. Recently, Chen *et al*. [7] proposed a promising prompting-based approach to facilitate the support for multiple vision tasks and models. Overcoming these existing limitations and offering support for multiple vision tasks and models could potentially lead to the adoption of deep codecs in real-world deep learning pipelines.

3.3. Data augmentation

Using coding as a data augmentation strategy could enable models to be more robust to coding artifacts. As shown by Otani *et al*. [32] and Reich *et al*. [34], coding-based data augmentation can introduce some degree of robustness, however, does not fully mitigate the deterioration in downstream accuracy as relevant information is still lost during coding. Additionally, training each downstream deep vision model from scratch using a custom augmentation pipeline based on the employed codec might be not feasible.

3.4. Adapting deep vision models for coded data

An alternative approach to overcome the deterioration in accuracy on coded images and videos is to augment the architecture of deep vision models. As demonstrated by Park *et al*. [33] ViTs [11] can consume JPEG DCT features instead of RGB images. This can introduce some degree of robustness to JPEG coding. However, similar to data augmentation, adapting all current deep architectures for custom representations of standard image and video codecs (*e.g*., JPEG DCT features or H.264 byte code) is non-trivial, requires considerable effort, and might be impractical.

3.5. Dataset design and data pre-processing

While beyond the scope of the paper and not considered in our experiments, some datasets already entail some degree of compression (*e.g*., ImageNet [38] and Kinetics-400 [20]). For instance, Kinetics-400 contains videos scraped from the internet compressed with various compression rates [20]. Additionally, pre-processing large video and image datasets often utilizes standard coding [9, 22]. For example, MMAction2 [9] typically extracts individual video frames from standard-coded videos using JPEG coding, resulting in frames that are coded multiple times. As we demonstrate standard coding can significantly impact downstream deep vision performance, thus, the careful consideration of coding during the design stage of datasets as well as during pre-processing is crucial.

4. Conclusion

In this paper, we analyzed the effect of standard image and video coding on four common vision tasks: image classification, object detection, semantic segmentation, as well as optical flow estimation. We tested a wide variety of current deep vision models (23 vision models in total) and observed that the predictive accuracy severely deteriorated with standard coding. Even a complete breakdown of downstream performance can arise for strong compression rates. Our analysis shows that image and video coding is particularly problematic for localization and dense prediction tasks. We discussed both challenges as well as possible avenues to mitigate the negative effects of (standardized) coding. We hope that our findings contribute to facilitating the development of improved standardized image and video codecs for deep vision models and shed light on the implications of deploying standardized codecs with deep vision models.

References

[1] Simon Baker, Daniel Scharstein, James P Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92:1–31, 2011. 7

[2] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *ICML*, pages 675–685, 2019. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 2, 3, 4

[4] Lahiru D Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for machines, a study. In *DCC*, pages 163–172, 2021. 8

[5] Chunlei Chen, Peng Zhang, Huixiang Zhang, Jiangyan Dai, Yugen Yi, Huihui Zhang, and Yonghui Zhang. Deep learning on computational-resource-limited platforms: A survey. *Mob. Inf. Syst.*, 2020:1–19, 2020. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587 [cs.CV]*, 2017. 2, 3, 7

[7] Yi-Hsin Chen, Ying-Chieh Weng, Chia-Hao Kao, Cheng Chien, Wei-Chen Chiu, and Wen-Hsiao Peng. TransTIC: Transferring transformer-based image compression from human perception to machine perception. In *ICCV*, pages 23297–23307, 2023. 8

[8] Hyomin Choi and Ivan V Bajić. Scalable image coding for humans and machines. *IEEE Trans. Image Process.*, 31:2739–2754, 2022. 8

[9] MMAction2 Contributors. OpenMMLab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020. 8

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 3, 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16×16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 3, 6, 8

[12] Kuntai Du, Qizheng Zhang, Anton Arapin, Haodong Wang, Zhengxu Xia, and Junchen Jiang. AccMPEG: Optimizing video encoding for video analytics. In *MLSys*, 2022. 1, 2, 8

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 3, 6

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 2

[15] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for MobileNetV3. In *ICCV*, pages 1314–1324, 2019. 2, 3

[16] Miao Hu, Zhenxiao Luo, Amirmohammad Pasdar, Young Choon Lee, Yipeng Zhou, and Di Wu. Edge-based video analytics: A survey. *arXiv:2303.14329 [cs.DC]*, 2023. 1

[17] Graham Hudson, Alain Léger, Birger Niss, István Sebestyén, and Jørgen Vaaben. JPEG-1 standard 25 years: past, present, and future reasons for a success. *J. Electron. Imaging*, 27(4):040901, 2018. 1, 2

[18] Berivan Isik, Onur G. Guleryuz, Danhang Tang, Jonathan Taylor, and Philip A. Chou. Sandwiched video compression: Efficiently extending the reach of standard codecs with neural wrappers. In *ICIP*, pages 2055–2059, 2023. 8

[19] Hayato Itsumi, Florian Beye, Vitthal Charvi, and Koichi Nihei. Learning important regions via attention for video streaming on cloud robotics. In *IROS*, pages 6833–6839, 2022. 1

[20] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics human action video dataset. *arXiv:1705.06950 [cs.CV]*, 2017. 8

[21] Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, and Esa Rahtu. Image coding for machines: an end-to-end learned approach. In *ICASSP*, pages 1590–1594, 2021. 8

[22] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. FFCV: Accelerating training by removing data bottlenecks. In *CVPR*, pages 12011–12020, 2023. 8

[23] Stefan Lederer. 2019 video developer report – the future of video: AV1 codec, AI & machine learning, and low latency. https://bitmovin.com/bitmovin-2019-video-developer-report-av1-codec-ai-machine-learning-low-latency/, 2019. 1, 2

[24] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv:2111.11429 [cs.CV]*, 2021. 2, 3, 4

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 3, 4

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2, 3, 6

[27] Xiyang Luo, Hossein Talebi, Feng Yang, Michael Elad, and Peyman Milanfar. The rate-distortion-accuracy tradeoff: JPEG case study. In *DCC*, pages 354–354, 2021. 2, 7

[28] Amol Mandhane, Anton Zhernov, Maribeth Rauh, Chenjie Gu, Miaosen Wang, Flora Xue, Wendy Shang, Derek Pang, Rene Claus, Ching-Han Chiang, et al. MuZero with self-competition for rate control in VP9 video compression. *arXiv:2202.06626 [eess.IV]*, 2022. 2, 8

[29] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Comput. Surv.*, 55(12):1–37, 2023. 1

[30] Dipti Mishra, Satish Kumar Singh, and Rajat Kumar Singh. Deep architectures for image compression: A critical review. *Signal Process.*, 191:108346, 2022. 8

[31] MMSegmentation Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 2, 3

[32] Aoi Otani, Ryota Hashiguchi, Kazuki Omi, Norishige Fukushima, and Toru Tamaki. Performance evaluation of action recognition models on low quality videos. *IEEE Access*, 10:94898–94907, 2022. 1, 8

[33] Jeongsoo Park and Justin Johnson. RGB no more: Minimally-decoded jpeg vision transformers. In *CVPR*, pages 22334–22346, 2023. 8

[34] Christoph Reich, Biplob Debnath, Deep Patel, Tim Prangemeier, and Srimat Chakradhar. Deep video codec control. *arXiv:2308.16215 [eess.IV]*, 2023. 8

[35] Christoph Reich, Biplob Debnath, Deep Patel, and Srimat Chakradhar. Differentiable JPEG: The devil is in the details. In *WACV*, 2024. 7

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE PAMI*, 39(06):1137–1149, 2017. 2, 3, 4

[37] Iain E Richardson. *H.264 and MPEG-4 video compression: Video coding for next-generation multimedia*. John Wiley & Sons, 2004. 1, 2

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comp. Vis.*, 115:211–252, 2015. 2, 3, 6, 8

[39] Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959. 2

[40] Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *NIPS Workshop on Machine Learning and Computer Security*, 2017. 2, 7

[41] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. SMURF: Self-teaching multi-frame unsupervised RAFT with full-image warping. In *CVPR*, pages 3887–3896, 2021. 2, 3, 6

[42] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 2, 3, 6

[43] Yuan Tian, Guo Lu, Xiongkuo Min, Zhaohui Che, Guangtao Zhai, Guodong Guo, and Zhiyong Gao. Self-conditioned probabilistic learning of video rescaling. In *ICCV*, pages 4490–4499, 2021. 8

[44] Suramya Tomar. Converting video formats with FFmpeg. *Linux J.*, 146, 2006. 6

[45] TorchVision maintainers and contributors. TorchVision: PyTorch's computer vision library. `https://github.com/pytorch/vision`, 2016. 2, 3

[46] Gregory K Wallace. The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.*, 38(1):xviii–xxxiv, 1992. 1, 2

[47] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003. 1, 2, 6

[48] Ross Wightman. PyTorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 2, 3, 6

[49] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 2, 3

[50] Yun Zhang, Linwei Zhu, Gangyi Jiang, Sam Kwong, and C.-C. Jay Kuo. A survey on perceptually optimized video coding. *ACM Comput. Surv.*, 55(12):1–37, 2023. 8