CAR-CNN: A Deep Residual Convolutional Neural Network for Compression Artifact Removal in Video Surveillance Systems

Miloud Aqqa and Shishir K. Shah
Quantitative Imaging Laboratory, Department of Computer Science, University of Houston, U.S.A.

Keywords: Compression Artifacts, Video Quality Enhancement, Deep Learning, Visual Surveillance.

Abstract: Video compression algorithms are pervasively applied at the camera level prior to video transmission due to bandwidth constraints, thereby reducing the quality of video available for video analytics. These artifacts may lead to decreased performance of some core applications in video surveillance systems such as object detection. To remove such distortions during video decoding, it is required to recover original video frames from distorted ones. To this end, we present a fully convolutional residual network for compression artifact removal *(CAR-CNN)* without prior knowledge on the noise distribution trained using a novel, differentiable loss function. To provide a baseline, we also trained our model by optimizing the *Structural Similarity (SSIM)* and *Mean Squared Error (MSE)*. We test *CAR-CNN* on self-collected data, and we show that it can be applied as a pre-processing step for the object detection task in practical, non-idealized applications where quality distortions may be present.

1 INTRODUCTION

In real deployments of public safety video systems, cameras are often backhauled via wireless media, where packet loss and jitter impact video quality available for video analytics. Furthermore, due to bandwidth constraints, lossy compression algorithms are often used to encode videos before transmission to central storage and processing sites in order to avoid network congestion and lower communication latency at the expense of introducing undesired complex artifacts (e.g., blocking, blurring, floating), which remove textures and details in video frames as shown in Figure 1. These artifacts are not only unpleasant to the human eye but also adversely impact the performance of various high-level vision algorithms such as object detectors (Aqqa et al., 2019).

Typically, video compression algorithms come with a parameter to tune the trade-off between file size and quality of the video. The larger this parameter, the stronger are quality distortions stemming from compression artifacts. However, merely opting for low compression rates is not always a practical solution. In video surveillance systems, video transportation from the transmitting node to the video analytics compute engine is typically performed over an IP network infrastructure, where transmission channels have limited bandwidth and are allowed a certain quota per camera. This constraint is often ensured by strong compression.

Most surveillance cameras adopt the H.264/AVC standard for video encoding (Wiegand et al., 2003), which is a lossy compression technique. A video consists of images; an image is divided into slices and blocks. A block is a square part (16×16, 8×8, and 4×4) of the image. H.264 is a block-based coder/decoder, meaning that a series of mathematical functions are applied on individual blocks to achieve compression and decompression (Juurlink et al., 2012). Also, it exploits spatial redundancy within images and temporal redundancy in videos to achieve appealing compression ratios with low latency, making it a widely accepted standard for video transmission for a myriad of applications.

Much work has been done on removing different compression artifacts from JPEG images using different techniques, from optimizing discrete cosine transform (DCT) coefficients (Zhang et al., 2013) to recently leveraging the representational power of convolutional neural network (CNN) (Galteri et al., 2017; Svoboda et al., 2016; Yu et al., 2015). CNNs have been successfully used in many other image restoration tasks including super-resolution (Kim et al., 2016; Dong et al., 2014), denoising (Zhang et al., 2017), structured noise removal (Eigen et al., 2013), and blind deconvolution (Schuler et al., 2016) with

Aqqa, M. and Shah, S.
CAR-CNN: A Deep Residual Convolutional Neural Network for Compression Artifact Removal in Video Surveillance Systems.
DOI: 10.5220/0009184405690575
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 4: VISAPP*, pages 569-575
ISBN: 978-989-758-402-2; ISSN: 2184-4321

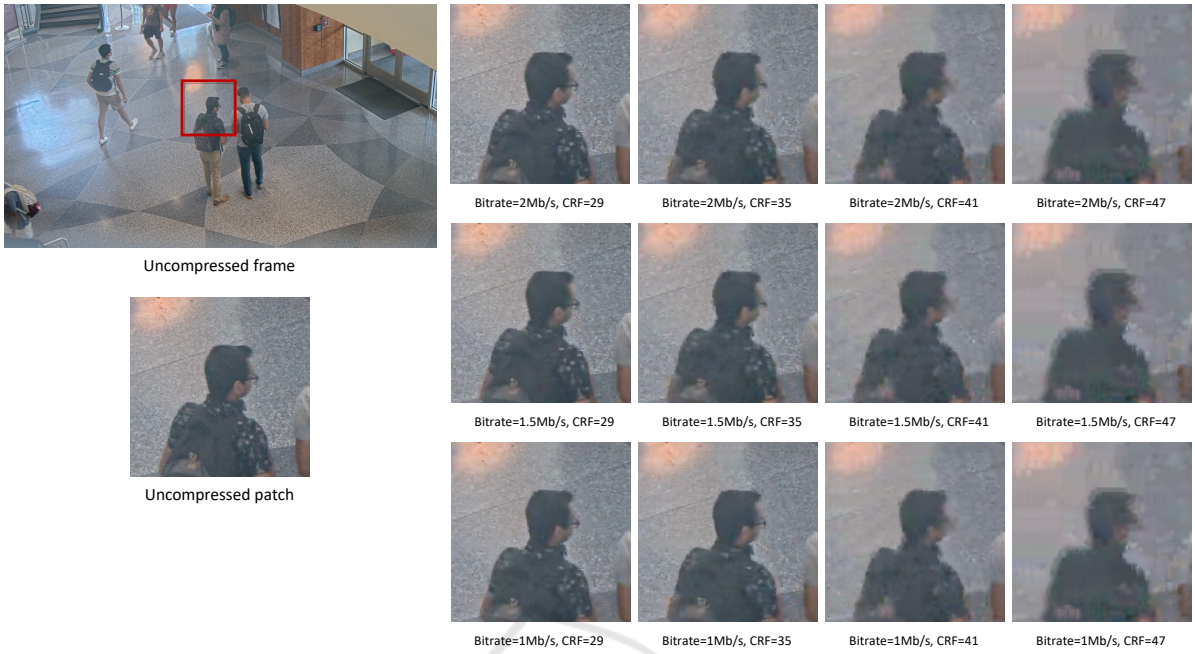

Figure 1: An uncompressed patch taken from a video frame and its 12 compressed versions. The compression artifacts can be visually perceived as CRF value increases and bitrate decreases. The combination of CRF=29 and maximum bitrate of 2Mb/s results in the lowest compressed version, thus better image quality. The combination of CRF=47 and maximum bitrate of 1Mb/s results in highest compressed version, thus worst image quality.

state-of-the-art performance.

In this paper, we target the problem of compression artifact removal (CAR) in H.264/AVC encoded videos. Compared to JPEG compression method (Wallace, 1992), the degradation caused by H.264 is not only originated from spatial artifacts (e.g. blocking, blurring, ringing) but also from temporal artifacts (e.g., floating, flickering). The aim is to devise an approach that can be applied as a post-processing step on decompressed frames, and therefore it can be used on many lossy video compression algorithms. In addition, it should be able to handle videos encoded at different bit rates, and thus at different qualities. This allows avoiding any changes to the existing compression pipelines, which are usually optimized (e.g., using dedicated hardware).

The contributions of our work are three-fold: (i) We present a feed-forward fully convolutional residual generative network by optimizing a novel loss function, and we show its superior performance compared to *de facto* error metrics in image reconstruction. (ii) We evaluate the performance of our approach using self-collected data, which consists of thirty uncompressed videos recorded in different surveillance scenarios (indoor and outdoor). The video frames from these videos are considered to be of high quality. We augment this dataset by introducing complex artifacts under different levels of video compression using H.264/AVC standard. (iii) We show that our method can be used as a pre-processing step for object detectors in video surveillance systems.

In section 2, we review some of the related work. In section 3, we detail the proposed method and different loss functions used in this work. We describe in section 4 the dataset, performance metrics, and implementation details. Section 5 reports the results obtained from our experiments. In section 6, we conclude our work.

2 RELATED WORK

There is a vast literature of image restoration, addressing image compression artifacts. These methods are ranging from hand-designed filters relying on information in the DCT domain to recently using deep convolutional neural networks (DCNN) following their success in other machine vision tasks.

Simple artifact removal filters are included in many software for handling images, videos, and other multimedia files. For example, the FFmpeg framework includes the simple postprocessing (ssp) filter (Nosratinia, 1999), which applies JPEG compression to the shifted versions of the already-compressed image, and averages the results. Foi *et al.* devel-

oped Pointwise Shape-Adaptive DCT (SA-DCT) (Foi et al., 2006), in which the thresholded transform coefficients are used to reconstruct a local estimate of the image signal within the adaptive-shape support. Yang *et al.* presented in (Yang et al., 2000) a different approach, which consists of applying DCT-based lapped transform on the signal already in the DCT domain. The authors in (Li et al., 2014) decompose images into texture and structure components, then eliminate artifacts that are part of the texture component due to contrast enhancement. Chang *et al.* (Chang et al., 2014) propose to remove blocking artifacts of JPEG compression images by finding a sparse representation over a learned dictionary from a training set of images. The main disadvantage of these algorithms is that they explicitly attempt to reverse the effect of DCT-domain quantization optimally, and thus they are very specific to the applied compressor. Furthermore, they tend to overly smooth texture regions without reproducing sharp edges.

The work presented in this paper was inspired by DCNN based approaches. The main idea is to learn an image transformation function that can produce a restored version of the given input image. Dong *et al.* (Dong et al., 2015) propose artifact reduction CNN (AR-CNN), which extends their super-resolution CNN (SRCNN) architecture with feature enhancement layers. They trained AR-CNN in two stages - a shallow network is trained first, then it is used as an initialization for a final 4 layer CNN due to training difficulties encountered when training the later from scratch. Differently from AR-CNN, Svoboda *et al.* (Svoboda et al., 2016) report better results by training a feed-forward CNN that combines residual learning and skip architecture to get a better reconstruction quality.

Convolutional networks have successfully shown their ability in different image transformation problems, such as image denoising (Zhang et al., 2017), super-resolution (Kim et al., 2016; Dong et al., 2014), and style-transfer (Gatys et al., 2016). Zhang *et al.* (Zhang et al., 2017) propose a denoising convolutional neural network (DnCNN) to eliminate Gaussian noise, showing that residual learning and batch normalization are beneficial for this task. Kim *et al.* (Kim et al., 2016) addressed the problem of image super-resolution, using a deep architecture trained on residual images. Ledig *et al.* (Ledig et al., 2017) propose a deep residual convolutional network, trained in an adversarial fashion by optimizing a perceptual loss that combines an adversarial loss and a content loss. The authors state that their model can recover photo-realistic textures from heavily downsampled images. A style-transfer method of Gatys *et al.* (Gatys et al., 2016) uses image representations from convolutional neural network optimized for object recognition while optimizing a loss that accounts for both image content and style to keep the content of an arbitrary photograph with the appearance of numerous well-known artworks.

To the best of our knowledge, we are the first to adapt the idea of residual learning (Kim et al., 2016) for H.264/AVC compression artifact removal based on CNN in video surveillance systems. We follow the assumption "deeper is better", and we propose 34-layer fully convolutional residual neural network and is, therefore, able to restore images of any resolution. Moreover, we investigate the use of two loss functions (MSE and SSIM), and define a new loss function that combines the advantages of MSE and SSIM.

3 METHODOLOGY

In H.264/AVC compression artifact removal task, the goal is to restore a video frame I^R from a compressed one I^L. Most video compression algorithms (e.g., H.264/AVC, H.265/HEVC) encode images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance. Therefore, we transform the color images to YCbCr space, and only the luminance channel Y is used further.

In $[0,255]^{W\times H\times C}$, we define I^H, I^L, and I^R as image tensors with width W, height H and number of image channels C. During H.264/AVC video encoding process, an uncompressed image I^H is encoded by:

$$I^L = E(I^H, QP) \tag{1}$$

using H.264 encoder E with some quantization parameter QP. The goal is to learn an inverse function $\phi \approx E_{QP}^{-1}$ to remove compression artifacts introduced by E, thus restoring I^H from I^L:

$$I^H \approx I^R = \phi(I^L) \tag{2}$$

To this end, we define $\phi(.)$ as a fully convolutional residual network $\phi(I^L;\theta)$ with parameters θ that are learned by optimizing a loss function l_{CAR}. Given N training video frames, we solve:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} l_{CAR}(I^H, \phi(I^L;\theta)) \tag{3}$$

In the following, we describe the architecture of *CAR-CNN* and different loss functions used to remove H.264/AVC compression artifacts.

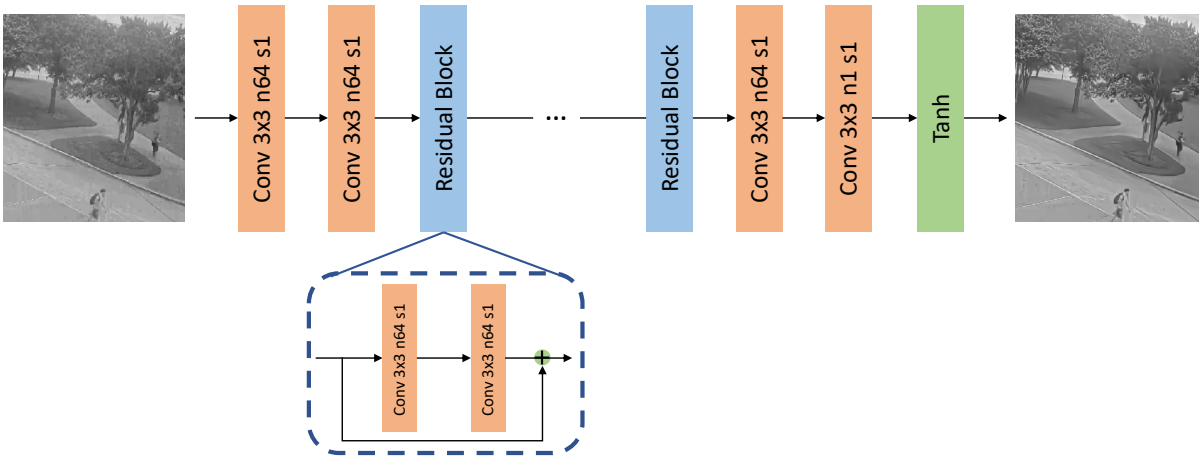

Figure 2: An Overview of CAR-CNN architecture. The network contains 15 residual blocks indicating with n the number of filters and s the stride of each convolutional layer.

3.1 CAR-CNN Architecture

An overview of our proposed network is shown in Figure 2. Inspired by (Kim et al., 2016), we propose a 34-layers residual generative network that contains only blocks of convolutional layers and LeakyReLU non-linearities. In particular, we use convolutional layers with 3×3 kernels and 64 feature maps. All convolutional layers are followed by a LeakyReLU activation with a slope of 0.2 for negative inputs. After every convolution, we apply a padding of 1 pixel to keep the same image size across all convolution layers. Since the input images were transformed to gray-scale considering only the luminance channel Y, the last layer is a simple convolutional layer with one feature map followed by a *tanh* activation function to keep the output values between the $[-1, 1]$ range.

3.2 Loss Functions

In this section, we describe different loss functions used to train *CAR-CNN*.

3.2.1 Mean Squared Error

As first attempt to remove compression artifact, we use Mean Squared Error loss (MSE):

$$l_{MSE} = \frac{1}{WH} \sum_{i=1}^{H} \sum_{j=1}^{W} (I_{i,j}^{H} - I_{i,j}^{R})^2 \tag{4}$$

Although MSE has shown improved performance in JPEG artifact removal task (Svoboda et al., 2016), its results are still sub-optimal as it doesn't recover most of the high-frequency details from a distorted video frame.

3.2.2 Stuctural Similarity

A popular index in the image restoration task is the structural similarity index (SSIM) (Wang et al., 2004). It evaluates images accounting for the fact that the human visual system is sensitive to changes in local structure. If the goal is to produce visually pleasing images, the loss function should be perceptually motivated, as is the case of SSIM.

SSIM for pixel p is defined as:

$$SSIM(p) = \frac{2\mu_i\mu_j + C_1}{\mu_i^2 + \mu_j^2 + C_1} \cdot \frac{2\sigma_{ij} + C_2}{\sigma_i^2 + \sigma_j^2 + C_2} \tag{5}$$

The loss function for SSIM can be then written as:

$$l_{SSIM} = 1 - SSIM(p) \tag{6}$$

The network is trained to optimize the structural similarity between the reference images and the restored ones.

3.2.3 The Best of Both Worlds: SSIM + MSE

To capture the best characteristics of both SSIM and MSE, we propose to combine them:

$$l_{both} = \alpha \cdot l_{SSIM} + (1 - \alpha) \cdot l_{MSE} \tag{7}$$

where we empirically set $\alpha = 0.79$. This is motivated by the fact that MSE can excel in recovering low frequencies while SSIM can better preserve contrast in high-frequency regions.

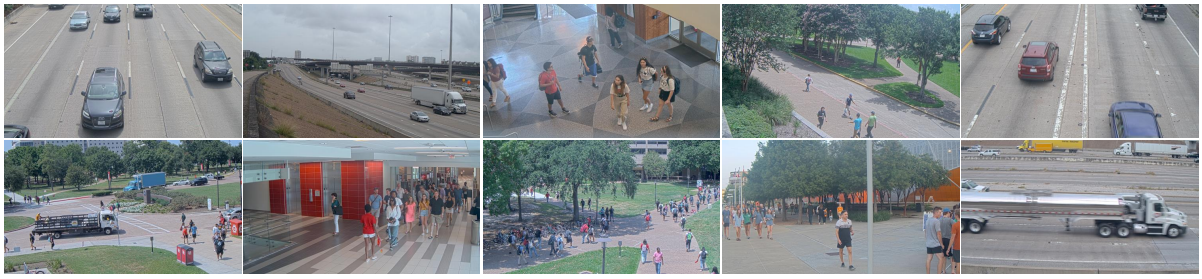

Figure 3: Samples of video frames from uncompressed videos recorded in indoor and outdoor surveillance scenarios.

4 EXPERIMENTAL SETUP

4.1 Dataset

Previous work for JPEG compression artifact removal were tested on BSDS500 (Martin et al., 2001) and LIVE1 (Sheikh et al., 2014) datasets. These datasets contain still images that have distinctly different characteristics as compared to video frames encountered in video surveillance systems. For this reason, we have collected thirty uncompressed videos that represent common scenarios where video surveillance cameras are deployed. The videos are 5 minutes long movie clips and were acquired using AXIS P3227-LVE network camera and recorded in 1080p high definition (1920×1080) at 30fps. Samples of video frames are shown in Figure 3.

H.264/AVC encoding uses Constant Rate Factor (CRF) as the default quality (and rate control) setting. CRF achieves constant quality by compressing different frames by different amounts, thus varying the Quantization Parameter (QP) as necessary to maintain a certain level of perceived quality. It does this by taking motion into account, similar to the encoder on a surveillance camera. CRF ranges between 0 and 51, where lower values would result in better quality, and higher values lead to more compression. To simulate the trade-of between quality and bitrate, we have used CRF in conjunction with Video Buffer Verifier (VBV) mode to ensure that the bitrate is constrained to a certain maximum as in real-world settings. An exhaustive combination of CRF values (29, 35, 41, and 47) and maximum bitrate values (2Mb/s, 1.5Mb/s, and 1Mb/s) are selected to create a total of 12 data variants.

4.2 Performance Metrics

The most wide-spread evaluation metrics for quality assessment in compression artifact removal task are MSE, and the peak signal-to-noise ratio (PSNR), which is the MSE normalized to the maximum possible signal values expressed in decibel (dB). Another alternative is to use the structural similarity index (SSIM), which is the mean of the product of three terms assessing similarity in luminance, contrast, and structure over multiple localized windows. We report the mean PSNR and SSIM across the 12 data variants created.

4.3 Implementation Details

For training our networks, we use 90k video frames as the training set and 12k for the validation set. Testing is performed on 17k video frames for each of the 12 data variants.

We use the PyTorch framework (Paszke et al., 2017) for our evaluations. The training process was distributed over two Nvidia Tesla v100 GPUs with a mini-batch of 128 images and have been carried on for 360 epochs. For each image, we first rescale it to (910×512), and then we randomly crop a 64×64 patch with horizontal flipping. We optimize the network's parameters with Adam (Kingma and Ba, 2015), starting with a learning rate of 10^{-4} and a momentum of 0.9.

5 RESULTS

5.1 Artifact Removal

The evaluation results of the mean PSNR and SSIM across different data variants are shown in Table 1 and Table 2, respectively. We compare our results to the simple postprocessing filter (ssp) in the FFmpeg framework. As can be seen from the results, our approach outperforms ssp in all data variants. More specifically, in the case of the highest compression level (i.e., Bitrate=1Mb/s and CRF=47), we can see an improvement in PSNR of 0.30 dB over ssp while SSIM is improved from 0.604 to 0.693. At the lowest compression (i.e., Bitrate=2Mb/s and CRF=29),

Table 1: Restoration Quality Comparison. Results reported for average PSNR (dB) using luminance.

Method	Bitrate = 2 Mb/s				Bitrate = 1.5 Mb/s				Bitrate = 1 Mb/s			
	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47
ssp	29.65	28.09	25.93	24.53	29.25	28.07	25.90	24.53	28.58	27.70	25.85	24.53
Our MSE	**30.07**	**28.36**	**26.65**	**25.27**	**29.61**	**28.36**	**26.64**	**25.17**	**28.88**	**27.98**	**26.62**	**25.05**
Our SSIM	29.69	28.12	26.02	24.78	29.32	28.12	25.91	24.78	28.54	27.72	25.88	24.78
CAR-CNN	29.76	28.19	26.07	24.81	29.38	28.18	25.96	24.81	28.60	27.78	25.93	24.81

Table 2: Restoration Quality Comparison. Results reported for average SSIM using luminance.

Method	Bitrate = 2 Mb/s				Bitrate = 1.5 Mb/s				Bitrate = 1 Mb/s			
	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47
ssp	0.817	0.789	0.725	0.604	0.807	0.788	0.723	0.604	0.787	0.774	0.720	0.604
Our MSE	0.827	0.805	0.758	0.681	0.822	0.801	0.757	0.681	0.794	0.784	0.756	0.681
Our SSIM	0.845	0.829	0.777	0.689	0.837	0.819	**0.763**	0.689	0.817	0.794	0.761	0.687
CAR-CNN	**0.873**	**0.867**	**0.786**	**0.695**	**0.858**	**0.826**	0.762	**0.693**	**0.835**	**0.796**	**0.765**	**0.693**

Table 3: Detection performance of YOLO measured as mean average precision (mAP) at IoU=0.50 on the 12 data variants for different reconstruction methods.

Method	Bitrate = 2 Mb/s				Bitrate = 1.5 Mb/s				Bitrate = 1 Mb/s			
	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47	CRF-29	CRF-35	CRF-41	CRF-47
ssp	0.766	0.736	0.661	0.522	0.756	0.735	0.661	0.519	0.745	0.733	0.557	0.519
Our MSE	0.774	0.759	0.698	0.577	0.768	0.761	0.697	0.577	0.758	0.753	0.669	0.577
Our SSIM	0.783	0.772	0.703	0.587	0.779	0.773	0.706	0.587	0.764	0.759	0.675	0.587
CAR-CNN	**0.789**	**0.781**	**0.723**	**0.589**	**0.783**	**0.776**	**0.727**	**0.589**	**0.769**	**0.763**	**0.676**	**0.588**

we see a gain of 0.42 dB in PSNR over ssp, and we improve SSIM from 0.817 to 0.873. From a quality index point of view, MSE based model outperforms SSIM and (SSIM+MSE) based models in PSNR, which can be explained by the fact that MSE tends to evaluate better more blurry and smooth regions than realistic textures.

5.2 Object Detection

In video surveillance systems, we are more interested in understanding how video quality affects machine vision algorithms. During video compression, quality distortions stemmed from spatial and temporal artifacts are introduced to the video frames leading to decreased performance of object detectors, as shown in (Aqqa et al., 2019). This degradation in performance can be explained because compression artifacts remove textures and details in these video frames. These high-frequency features represent edges and shapes of objects that the detector may be looking for to classify an object.

In this experiment, we select YOLO (Redmon et al., 2016) as the object detector. We consider its detections on uncompressed videos as ground-truth bounding boxes, and we compare them against its detections on reconstructed versions of the 12 compressed variants. As a lower bound, we report performance on images restored using ssp; results are reported in Table 3. As can be seen, the performance drop of YOLO with respect to its performance on uncompressed video frames is significant. Even for moderate compression levels (i.e., CRF=29), the performance decreases by at least 21.1%. Our (SSIM+MSE) based network outperforms MSE and SSIM based networks across all data variants. In particular, at the lowest compression level (i.e., Bitrate=2Mb/s and CRF=29) the improvement in mAP using CAR-CNN is the following: ssp(+2.3%), MSE(+1.5%) and SSIM(+0.6%) highlighting the benefits of combining both MSE and SSIM for the compression artifacts removal task. As can be observed in Table 3, at the highest compression level (i.e., Bitrate=1Mb/s and CRF=47), our approach has an improvement in mAP of 6.9% over ssp, 1.1% over MSE, and 0.1% over SSIM.

6 CONCLUSION

We have presented a 34-layer fully convolutional residual neural network for H.264/AVC compression artifact removal in video surveillance systems. Our network was trained using a novel loss function and outperforms MSE and SSIM based networks. Moreover, we have shown that it is possible to improve video quality, both for human viewers and machines and thus avoiding changes to the compression pipelines through applying a pre-processing step that removes compression artifacts.

ACKNOWLEDGEMENT

This work was performed in part through the financial assistance award, Multi-tiered Video Analytics for Abnormality Detection and Alerting to Improve Response Time for First Responder Communications and Operations (Grant No. 60NANB17D178), from U.S. Department of Commerce, National Institute of Standards and Technology.

REFERENCES

Aqqa, M., Mantini, P., and Shah, S. K. (2019). Understanding how video quality affects object detection algorithms. In *14th International Conference on Computer Vision Theory and Application*.

Chang, H., Ng, M. K., and Zeng, T. (2014). Reducing artifacts in jpeg decompression via a learned dictionary. *IEEE Transactions on Image Processing*, 62:718–728.

Dong, C., Deng, Y., Loy, C. C., and Tang, X. (2015). Compression artifacts reduction by a deep convolutional network. In *IEEE International Conference on Computer Vision (ICCV)*.

Dong, C., Loy, C. C., He, K., and Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*.

Eigen, D., Krishnan, D., and Fergus, R. (2013). Restoring an image taken through a window covered with dirt or rain. In *IEEE International Conference on Computer Vision (ICCV)*.

Foi, A., Katkovnik, V., and Egiazarian, K. (2006). Pointwise shape-adaptive dct for high-quality deblocking of compressed color images. In *14th European Signal Processing Conference*.

Galteri, L., Seidenari, L., Bertini, M., and Bimbo, A. D. (2017). Deep generative adversarial compression artifact removal. In *IEEE International Conference on Computer Vision (ICCV)*.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Juurlink, B., Alvarez-Mesa, M., Chi, C. C., Azevedo, A., Meenderinck, C., and Ramirez, A. (2012). Understanding the application: An overview of the h.264 standard. *Scalable Parallel Programming Applied to H.264/AVC Decoding*, pages 5–15.

Kim, J., Lee, J. K., and Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *the 3rd International Conference for Learning Representations*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y., Guo, F., Tan, R. T., and Brown, M. S. (2014). A contrast enhancement framework with jpeg artifacts suppression. In *European Conference on Computer Vision (ECCV)*.

Martin, D. R., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision (ICCV)*.

Nosratinia, A. (1999). Embedded post-processing for enhancement of compressed images. In *Proceedings DCC'99 Data Compression Conference*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schuler, C. J., Hirsch, M., Harmeling, S., and Schölkopf, B. (2016). Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1439–1451.

Sheikh, H. R., Wang, Z., Cormack, L., , and Bovik, A. C. (2014). Live image quality assessment database release 2.

Svoboda, P., Hradis, M., Bařina, D., and Zemcík, P. (2016). Compression artifacts removal using convolutional neural networks. *Journal of WSCG*, 24:63–72.

Wallace, G. K. (1992). The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612.

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

Yang, S., Kittitornkun, S., Hu, Y.-H., Nguyen, T., and Tull, D. (2000). Blocking artifact free inverse discrete cosine transform. In *Proceedings 2000 International Conference on Image Processing*.

Yu, K., Dong, C., Deng, Y., Loy, C. C., and Tang, X. (2015). Compression artifacts reduction by a deep convolutional network. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26:3142–3155.

Zhang, X., Xiong, R., Fan, X., and Gao, W. (2013). Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. *IEEE Transactions on Image Processing*, 22:4613–4626.