CAR-DCGAN: A Deep Convolutional Generative Adversarial 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 result in a degradation of frame quality due to their lossy approach to decrease the required bandwidth, thereby reducing the quality of video available for automatic video analysis. These artifacts may introduce undesired noise and complex structures, which remove textures and high-frequency details in video frames. Moreover, they may lead to decreased performance of some core applications in video surveillance systems such as object detectors. To remedy these quality distortions, it is required to restore high-quality videos from their low-quality counterparts without any changes to the existing compression pipelines through a complicated nonlinear 2D transformation. To this end, we devise a fully convolutional residual network for compression artifact removal (CAR-DCGAN) optimized in a patch-based generative adversarial approach (GAN). We show that our model is capable of restoring frames corrupted with complex and unknown distortions with more realistic details than existing methods. Furthermore, we show that CAR-DCGAN can be applied as a pre-processing step for the object detection task in video surveillance systems.

1 INTRODUCTION

In automated video surveillance systems, two key aspects impact video analytics algorithms: the compression parameters that facilitate the acquisition of video stream and the network characteristics that facilitate data transmission. In the current deployment of these systems, cameras are often backhauled via wireless links, where signal jitter and packet loss affect video quality. Oftentimes, these transmission channels have limited bandwidth and are allowed a certain quota per camera. Therefore, it is necessary to use lossy compression algorithms to encode videos before transmission to central storage and processing sites in order to reduce as much as possible the required bandwidth and lower communication latency. Unfortunately, whenever a lossy algorithm is used, undesired complex distortions will manifest. These distortions stemming from both spatial artifacts (i.e., blocking, blurring, color bleeding, and ringing) and temporal artifacts (i.e., edge floating, texture floating, and mosquito noise) remove textures and high-frequency details in video frames, as shown in Figure 1. There are two drawbacks to these artifacts. First, they make video frames appear unpleasant to the human eye. Second, they adversely impact the performance of various vision algorithms, such as object detectors (Aqqa et al., 2019).

Typically, lossy compression algorithms come with a factor to control the trade-off between the video's file size and quality. The larger this factor, the stronger is the video degradation stemming from these artifacts. However, opting for low compression rates is not always a practical solution due to bandwidth constraints in video surveillance systems. This constraint is often guaranteed by strong compression.

Most surveillance cameras use the H.264/AVC standard (Wiegand et al., 2003) for video compression, which is a lossy compression algorithm. H.264 exploits spatial redundancy within video frames and temporal redundancy in videos to achieve appealing compression ratios, making it the most widely accepted standard for video encoding. A video is a sequence of frames; a frame is divided into blocks of square sizes (16×16, 8×8 and 4×4). H.264 is a block-based coder/decoder that applies a series of mathematical functions to achieve compression and decompression (Juurlink et al., 2012).

Compression artifact removal aims to recover high-quality videos from their low-quality compressed counterparts. In the past, it has been addressed mainly without learning the denoising function from a large dataset. These techniques range

Aqqa, M. and Shah, S.
CAR-DCGAN: A Deep Convolutional Generative Adversarial Network for Compression Artifact Removal in Video Surveillance Systems.
DOI: 10.5220/0010312304550464
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 455-464
ISBN: 978-989-758-488-6

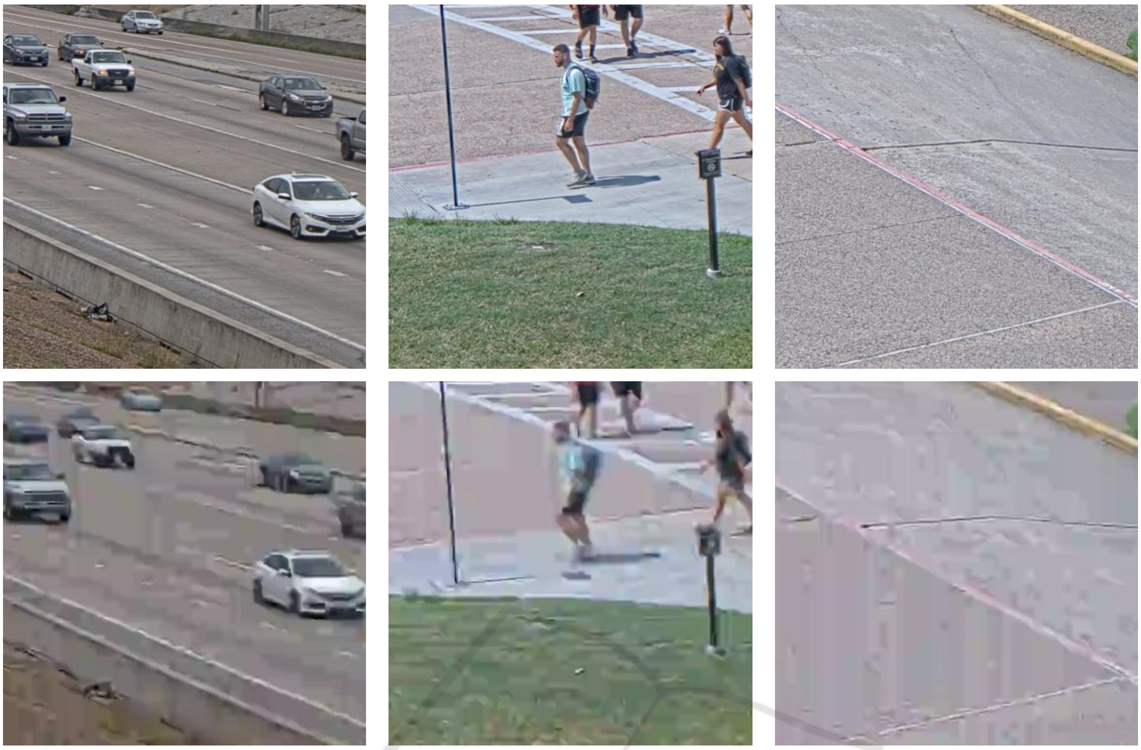


Figure 1: Examples of compression artifacts encountered in video surveillance systems. **Top row**: patches cropped from original video frames. **Bottom row**: patches cropped from compressed video frames. Different types of artifacts can appear in the same region. Best viewed in color on a computer screen.

from optimizing discrete cosine transform (DCT) coefficients (Zhang et al., 2013) to adding additional knowledge about images or patches based on adaptive distribution modeling (Liu et al., 2015). Following the success of deep convolutional neural networks (CNNs), few approaches have been proposed recently to address the artifact removal problem (Aqqa and Shah, 2020; Galteri et al., 2017; Svoboda et al., 2016; Yu et al., 2015). These techniques leverage the representational power of CNNs to accurately estimate the image manifold by learning a function that performs an image transformation from a compressed input image to a restored output.

In this work, we address the problem of compression artifact removal in H.264/AVC encoded videos. We propose a solution based on convolutional neural networks trained on large sets of video frame patches encoded at different bitrates, thus at different qualities. In contrast to (Aqqa and Shah, 2020), our generator network is optimized in an adversarial framework where there is no need to specify a loss function modeling the quality of frame patches. We show that our GAN can learn the conditional distribution of compressed and uncompressed video frames at any compression level, resulting in a better restoration. Furthermore, our experiments show that it can be applied as a reliable post-processing step for the object detection task in video surveillance systems.

In section 2, we review some of the related work. In section 3, we detail the architecture of *CAR-DCGAN* and the training approach. 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

In the past, compression artifact removal has been addressed mainly by designing hand-crafted filters relying on information in the DCT domain. Recently, few approaches have been proposed to learn the denoising function using deep convolutional neural networks (CNNs) following their success in other machine vision tasks. In the following, we will review both kinds of methods.

Many software for handling images, videos, and other multimedia files come with simple artifact removal filters. For example, the FFmpeg framework includes the simple post-processing (ssp) filter (Nosratinia, 1999), which applies JPEG compression to the shifted versions of the already-compressed images, and averages the results. Foi *et al.* proposed the Pointwise Shape-Adaptive DCT (SA-DCT) method (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.* have proposed to remove the artifacts introduced by quantization through a different approach (Yang et al., 2000), 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) developed a method to remove blocking artifacts from JPEG compression images by finding a sparse representation over a learned dictionary from a training set of images. While these algorithms have shown promising results, 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 and shapes of objects that machine vision algorithms such as object detectors may be looking for to classify an object.

A few recent approaches tackle the problem from a different angle by learning the denoising function using deep convolutional neural networks (DCNNs). These methods learn a 2D transformation function that can produce a restored version of the given degraded input image. Dong *et al.* (Dong et al., 2015) have proposed artifact reduction CNN (AR-CNN), which extends their super-resolution CNN (SRCNN) architecture with feature enhancement layers following sparse coding pipelines. 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 latter from scratch. Differently from AR-CNN, Svoboda *et al.* (Svoboda et al., 2016) developed a method with better results by training a feed-forward CNN that combines residual learning and skip architecture to get a sharper reconstruction. The authors in (Aqqa and Shah, 2020) have proposed a 34-layer fully convolutional residual neural network (CAR-CNN) to remove compression artifacts from H.264/AVC encoded videos. They trained their model by optimizing a loss function that combines MSE loss and SSIM based loss to capture both losses' characteristics, thus recovering high-frequency details without over-smoothing the restored frame. These methods have shown their ability to accurately estimate the image manifold with more image details and semantics thanks to not relying on local properties or DCT coefficient statistics.

Convolutional neural 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) have presented 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 photorealistic 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, the only method restoring compressed video frames is proposed by Aqqa *et al.* (Aqqa and Shah, 2020). Differently from their work, we propose an improved generator trained in a generative adversarial setup. We refer to the proposed method as *CAR-DCGAN*. We summarize our contributions as follows:
1) We present a new attempt to address compression artifact removal in H.264/AVC encoded videos. Unlike existing methods that directly optimize deep CNNs using hand-crafted loss functions, we train our generator in a generative adversarial setup using input patches encoded at different compression levels. Thus, learning a model that is quality-agnostic and can handle videos encoded at different bitrates.
2) Motivated by the fact that H.264/AVC is a block-based encoder, we propose a novel strategy to learn both the generator and the discriminator over patches of a single frame in a conditional setting to better estimate the frame manifold, leading to sharper reconstruction and more realistic images.
3) We demonstrate that our conditional GAN can produce better quality than other deep learning-based methods and can be used as a reliable pre-processing step for object detectors in video surveillance systems.

3 CAR-DCGAN

In the H.264/AVC compression artifact removal task, the aim is to restore a video frame F^R from a compressed frame F^L distorted by a lossy compression algorithm. In $\mathbb{R}^{W\times H\times L}$, we define F^H, F^C, and F^R as real valued tensors with width W, height H and number of image channels C. During H.264/AVC video encoding process, an uncompressed image F^H is encoded by:

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

using H.264 encoder E with some quantization parameter QP. We would like to learn an inverse function $\Phi \approx E_{QP}^{-1}$ to remove compression artifacts introduced by E, thus restoring F^H from F^L:

$$F^H \approx F^R = \Phi(F^L) \tag{2}$$

To this end, we define $\Phi(.)$ as a fully convolutional residual network $\Phi(F^L;\theta)$ with parameters θ that are learned using a Generative Adversarial Framework. We follow the assumption ”deeper is better,” and we propose a 34-layer fully convolutional residual neural network (FCN) and is, therefore, able to restore images of any resolution. Furthermore, FCN architectures are suitable for performing local nonlinear image transformations, which allows us to train the network over smaller frame patches. Indeed, H.264/AVC encoder/decoder operates over smaller blocks of square sizes (32×32, 16×16, 8×8 and 4×4); thus, the artifacts we are interested in removing appear at scales close to the patch size.

The GAN framework establishes two distinct players, a *generator* and a *discriminator*, and poses the two in an adversarial game. The generator (G) is fed some noisy input and tasked to create “fake” images that lay on the manifold of the real data with maximally confusing the discriminator; simultaneously, the discriminator (D) is tasked with distinguishing between samples from the generator and samples from the training data. In this work, we are not aiming to generate new unseen frames sampled from a distribution, but our task regards the output of an improved version of a degraded frame, thus learning a $\Phi(.)$ function able to process compressed frames and remove artifacts. This task can be achieved with GANs by conditioning the training. To condition the generative network, we feed as positive samples $F^H|F^L$ and as negative samples $F^R|F^L$, where .|. indicates channel-wise concatenation. Details of the proposed networks are presented in the following.

3.1 Generative Network

Inspired by (Aqqa and Shah, 2020), our generator (G) contains only blocks of convolutional layers and LeakyReLU non-linearities. We use layers with 32 feature maps with a 3×3 support. All convolutional layers are followed by a LeakyReLU activation with a slope of 0.2 for negative inputs. After the first two convolutional layers, we apply a chain of 16 residual blocks using a 1 pixel padding to keep the same frame size across all convolution layers after every convolution. Finally, to generate the enhanced frame, we use a single kernel convolutional layer with a *tanh* activation to keep the output values in the $[-1,1]$ range. An overview of the network is depicted in Figure 2.

3.2 Discriminative Network

The architecture of the discriminator network D is based mainly on a series of convolutional layers followed by LeakyReLU activation. We double the number of feature maps every two layers except the last one. The feature map’s size is decreased solely because of the effect of convolutions reaching the unitary dimension in the last layer, in which we use a sigmoid as the activation function. The discriminator is fed with frame patches rather than the whole frame, as indicated in Figure 2, this is motivated by the fact that the H.264/AVC encoder operates at the block level, and those artifacts we aim to remove are typically generated inside them. The weights φ of the discriminator D are learned by minimizing:

$$l_d = -\log(D_\varphi(F^H|F^L)) - \log(1 - D_\varphi(F^R|F^L)) \tag{3}$$

where $D(z)$ is taken from the sigmoid activation of the discriminator network, with z indicating channel-wise concatenation between the compressed input F^L and the correspondent uncompressed version F^H or restored one F^R.

3.3 Adversarial Training

Differently from *CAR-CNN* (Aqqa et al., 2019) that is trained using a direct supervision approach, we exploit to train our *CAR-DCGAN* in an adversarial framework given the ability of GANs to model complex multi-modal distributions, thus accurately estimate the image manifold. We train conditional GANs (Mirza and Osindero, 2014) to engage the generator to better capture the image transformation task. The training process for our model is as follows:

1. The generator processes the given degraded frame patch and produces an enhanced version of it.

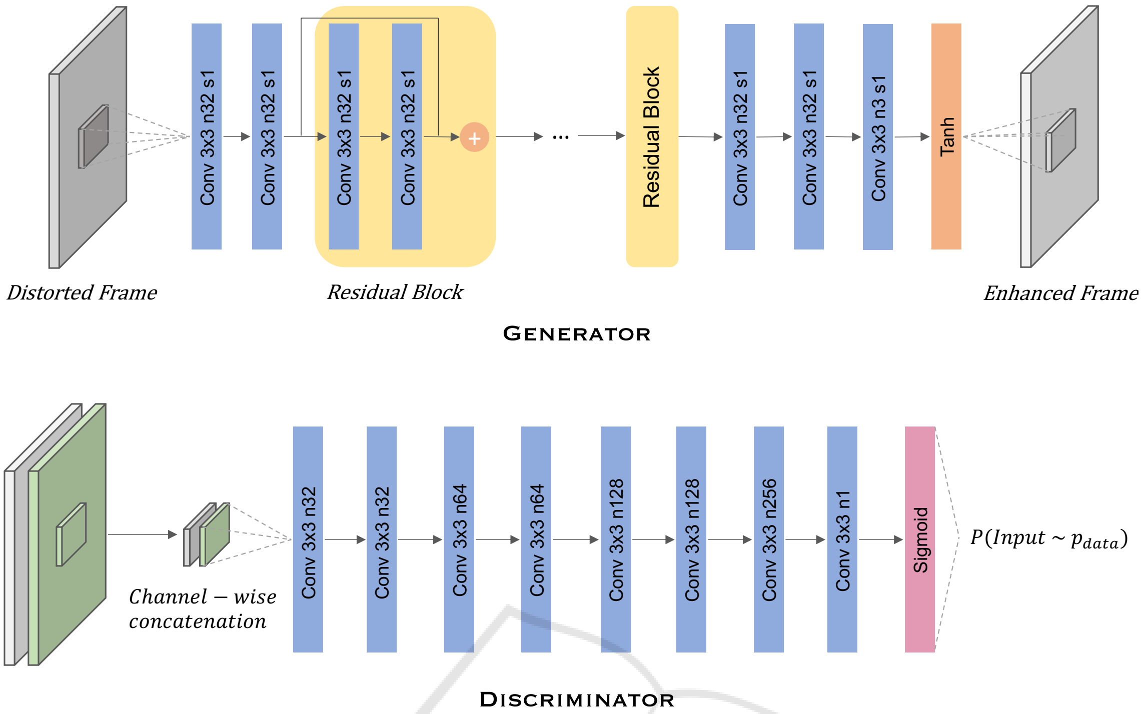


Figure 2: An overview of the proposed method. Top: Architecture of the generative network. It contains 16 residual blocks, and in each convolutional layer, we indicate by n the number of filters and s the stride. Bottom: Architecture of the discriminitive network. The input is created by performing the channel-wise concatenation between patches of the compressed frame F^L and the correspondent uncompressed version F^H or restored one F^R.

2. The discriminator learns basic convolutional filters in order to distinguish between “real” frame patches and “fake” ones.
3. The generator learns the correct bias and basic filters to remove artifacts induced by the compression process, thus confusing the discriminator.
4. The discriminator becomes more accustomed to “real” frame patches and is able to use signals in the conditional data to look for particular triggers in patches.

In the following, we describe the adversarial loss used to train the generator network.

3.3.1 Pixel-wise MSE Loss

Mean Squared Error loss (MSE) is defined as:

$$l_{MSE} = \frac{1}{WH} \sum_{i=1}^{H} \sum_{j=1}^{W} (F^H_{i,j} - F^R_{i,j})^2 \qquad (4)$$

l_{MSE} has shown improved performance in JPEG artifact removal task (Svoboda et al., 2016). However, it doesn’t recover most of the high-frequency details from a distorted input.

3.3.2 Perceptual Loss

Perceptual loss has been employed successfully in many image transformation tasks such as super-resolution and image restoration (Dosovitskiy and Brox, 2016; Gatys et al., 2016; Galteri et al., 2017). The main idea is to optimize the network in a feature space rather than the pixel space, encouraging uncompressed video frames and restored ones to have similar feature representations. The distance between two video frames is computed by projecting F^H and F^R on a pre-trained network feature space, hence extracting some meaningful latent representations. The perceptual loss is defined as:

$$l_P = \frac{1}{W_f H_f} \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} (\phi_k(F^H)_{i,j} - \phi_k(F^R)_{i,j})^2 \qquad (5)$$

where H_f and W_f are the height and the width of the feature maps respectively, and $\phi_k(F)$ represents the feature maps of some k-th layer of the pre-trained network for an input video frame F. In this work, we use the outputs of the *pool4* layer of the VGG-19 model (Simonyan and Zisserman, 2015) as the feature extractor.

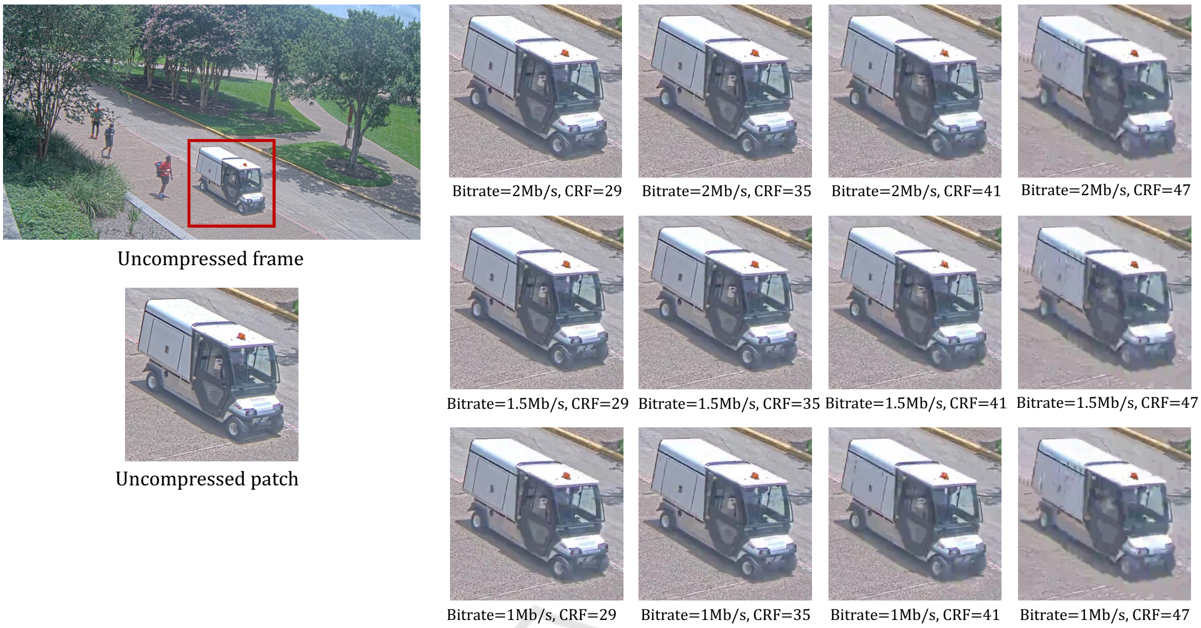


Figure 3: An uncompressed patch taken from a video frame and its 12 compressed versions. The compression artifacts can be visually perceived as CRF value increases and bitrate decreases. The combination of CRF=29 and maximum bitrate of 2Mb/s results in the lowest compressed version, thus better image quality. The combination of CRF=47 and maximum bitrate of 1Mb/s results in highest compressed version, thus worst image quality. Best viewed in color on a computer screen.

3.3.3 Adversarial Loss

We train the generator using a weighted combination of the MSE loss, the perceptual loss and the standard adversarial loss:

$$l_{CAR} = l_{MSE} + \alpha l_P + \beta l_{adv} \tag{6}$$

where l_{adv} is defined as:

$$l_{adv} = -\log(D_\varphi(F^R|F^L)) \tag{7}$$

that rewards maximal confusion to the discriminator.

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 with distinctly different characteristics compared to video frames encountered in video surveillance systems. For this reason, Aqqa *et al.* (Aqqa et al., 2019) have presented a new dataset of 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. To the best of our knowledge, it's the only dataset available with original uncompressed video stream for video surveillance systems, and therefore, we conduct experiments on this dataset.

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-off between quality and bitrate, CRF is used 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 in this dataset. An uncompressed video frame and its 12 compressed versions available in this dataset are shown in Figure 3.

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

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
H.264	29.65	28.09	25.93	24.53	29.25	28.07	25.90	24.53	28.58	27.70	25.85	24.53
CAR-CNN (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**
CAR-CNN (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 (SSIM+MSE)	29.76	28.19	26.07	24.81	29.38	28.18	25.96	24.81	28.60	27.78	25.93	24.81
CAR-DCGAN	28.94	27.84	25.79	24.49	28.82	28.18	25.49	24.38	27.90	27.55	25.64	24.40

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

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
H.264	0.817	0.789	0.725	0.604	0.807	0.788	0.723	0.604	0.787	0.774	0.720	0.604
CAR-CNN (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
CAR-CNN (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 (SSIM+MSE)	**0.873**	**0.867**	**0.786**	**0.695**	**0.858**	**0.826**	0.762	**0.693**	**0.835**	**0.796**	**0.765**	**0.693**
CAR-DCGAN	0.791	0.784	0.729	0.603	0.782	0.771	0.707	0.591	0.770	0.767	0.713	0.588

4.2 Similarity Measures

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 structure similarity index (SSIM) (Wang et al., 2004), which is the mean of the product of three terms assessing similarity in luminance, contrast and structure over multiple localized windows. For a fair comparison with CAR-CNN (Aqqa et al., 2019), we report evaluation of PSNR and SSIM measures across all 12 data variants.

4.3 Implementation Details

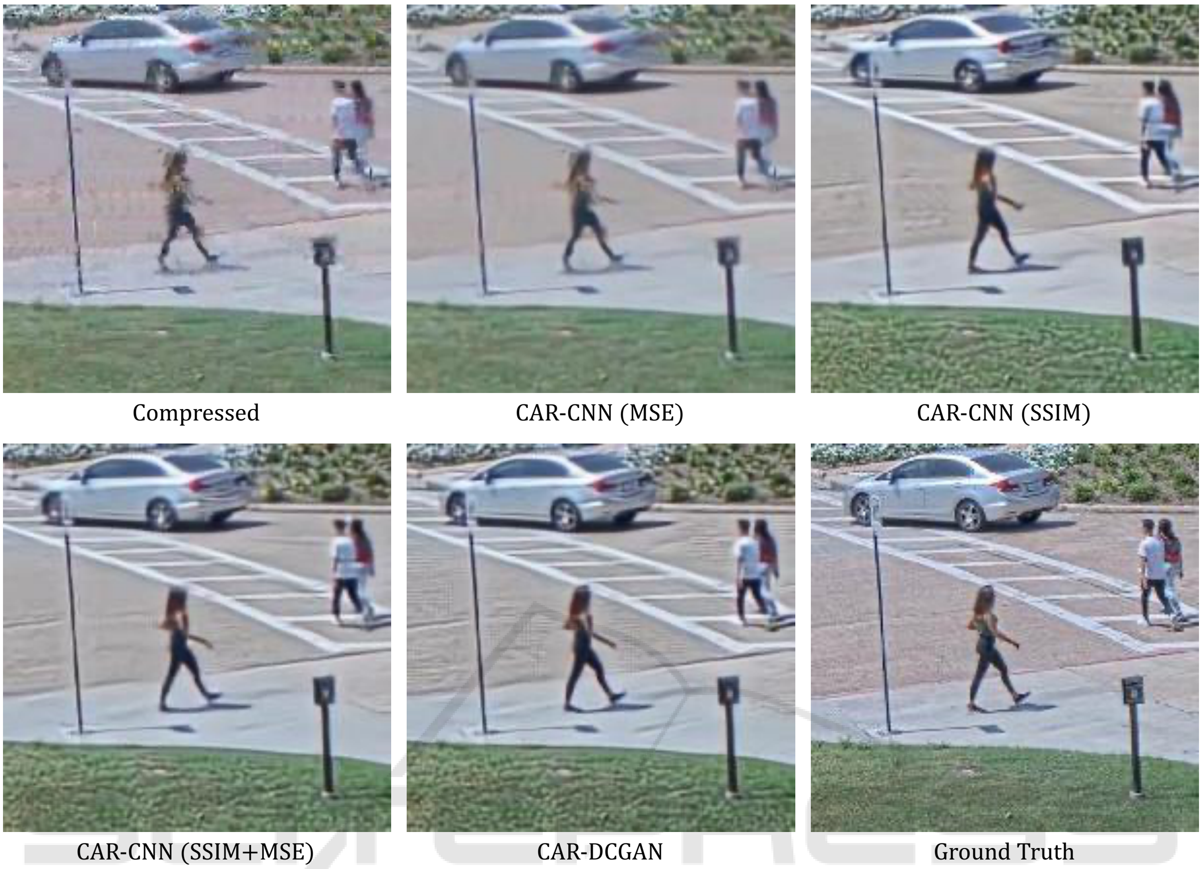
For training our networks, we use 90k video frames as the training set and 12k for the validation set. Testing is performed on 17k video frames for each of the 12 data variants. We have used the PyTorch framework (Paszke et al., 2017) for our evaluations. The training process was distributed over two Nvidia Tesla v100 GPUs with a mini-batch of 64 video frames and have been carried on for 360 epochs. For each image, we first rescale it to (910×512) and then we randomly crop a 32×32 patch with horizontal flipping. At the training stage, we have optimized the networks parameters with Adam (Kingma and Ba, 2015) starting with a learning rate of 10^{-4} and momentum of 0.9.

5 RESULTS

The evaluation results of the mean PSNR and SSIM across different data variants are shown in Table 1 and Table 2, respectively. The performance of our generator *CAR-DCGAN* is compared with the standard H.264/AVC compression and *CAR-CNN* (Aqqa and Shah, 2020), which was trained using three different loss functions: MSE loss, SSIM loss, and a weighted combination of MSE and SSIM. As can be seen in Table 1 and Table 2, the performance of our generator is much lower than all the three variants of CAR-CNN across all data variants from a quality index point of view. In fact, CAR-CNN was trained to optimize MSE and SSIM hence achieving better results in PSNR and SSIM measures. Moreover, these measures are known for better evaluating blurry regions over more realistic ones, as reported in other image transformation tasks such as super-resolution. However, the restored video frames by our generator are perceptually more realistic and have more finer consistent details, as shown in Figure 4. This can be explained by the fact that the combination of adversarial loss and perceptual loss tends to generate realistic textures rather than the smooth and poor detailed patches of the MSE/SSIM based generators.

5.1 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


Compressed

CAR-CNN (MSE)

CAR-CNN (SSIM)

CAR-CNN (SSIM+MSE)

CAR-DCGAN

Ground Truth

Figure 4: Example of a video frame compressed at the highest compression rate (i.e., CRF=47 and Bitrate=1Mb/s) and reconstruction results from different methods. Best viewed in color and zoomed in on a computer screen.

be looking for to classify an object.

In this experiment and for a fair comparison with CAR-CNN, we use YOLO as the object detector (Redmon et al., 2016). We adopt the same evaluation procedure as in (Aqqa and Shah, 2020). More specifically, the detections of YOLO on uncompressed videos are considered ground-truth bounding boxes and compared to its detections on reconstructed versions of the 12 compressed variants. As a lower bound, we also report performance on video frames restored using H.264; results are reported in Table 3. As we can expect, the more a video frame is compressed by the H.264/AVC encoder, the highest the drop in the performance of YOLO, especially at higher CRFs, as shown in Figure 5. Even at moderate compression levels (i.e., CRF=29), the detection performance drops by at least 15.7% after restoration. CAR-CNN generators are able to recover part of the object characteristics, but the improvements compared to the lower bound are not impressive, as they gain around 0.08% (MSE-based), 1.7% (SSIM-based), and 2.3% (SSIM+MSE) at the lowest compression level (i.e., CRF=29 and Bitrate=2MB/s). In fact, the over-smoothed and blurry patches recovered by MSE/SSIM generators lack sharp textures and high-frequency details that represent edges and shapes of objects. Compared to CAR-CNN, our GAN approach is able to restore the distorted frames in a more effective manner yielding the best result, increasing the performance by 7.7% in mAP from 0.766 to 0.843. At higher compression levels (i.e., CRF=41 and CRF=47), the variation in mAP across different restoration methods is minimal as it becomes difficult to recover missing details due to heavy compression.

These results assess the benefits of our patch-based generative approach compared to traditional methods for the H.264/AVC compression artifact removal task. Although MSE/SSIM trained generators yield better restoration performance from a quality index point of view, they are still simplistic and insufficient to capture the complexity of the image manifold. On the other hand, our adversarial approach allows the generator to accurately estimate the image manifold, thus better model the artifact removal task. Moreover, our experiments show that machine vision algorithms can suffer heavily from artifacts in-

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
H.264	0.766	0.736	0.661	0.522	0.756	0.735	0.661	0.519	0.745	0.733	0.557	0.519
CAR-CNN (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
CAR-CNN (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 (SSIM+MSE)	0.789	0.781	0.723	0.589	0.783	0.776	0.727	0.589	0.769	0.763	0.676	0.588
CAR-DCGAN	**0.843**	**0.831**	**0.743**	**0.597**	**0.838**	**0.815**	**0.738**	**0.594**	**0.811**	**0.804**	**0.680**	**0.589**

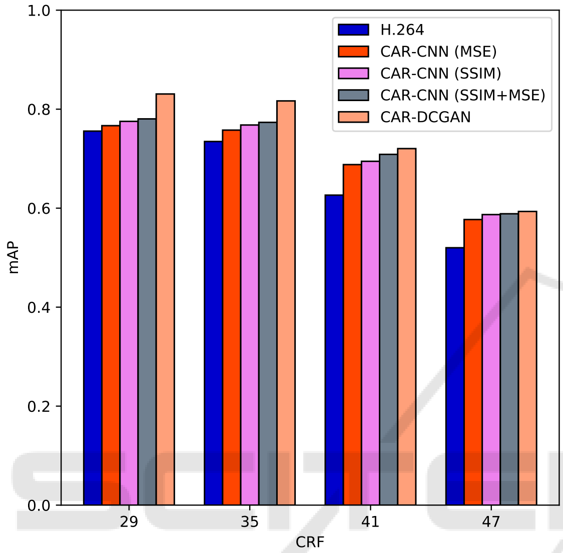


Figure 5: Mean Average Precision (mAP) of different reconstruction methods in regards to different CRFs. CAN-DCGAN outperforms other methods accross all CRFs. The variation is minimal for CRFs higher than 41.

troduced during the compression process due to an inability to generalize from their sharp training sets.

6 CONCLUSION

We have presented a fully convolutional residual neural network for the H.264/AVC compression artifact removal task in video surveillance systems. We trained our model in a patch-based generative adversarial approach to accurately learn the conditional distribution of compressed and uncompressed video frames, leading to better restoration than MSE/SSIM trained networks. Our experiments show that conditional GANs produce higher video frames with finer and sharper details relevant to both human viewers and machine vision algorithms. Moreover, we have shown that our approach can be used as a preprocessing step for computer vision tasks such as object detection in applications where quality distortions may be present.

ACKNOWLEDGMENT

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*.

Aqqa, M. and Shah, S. K. (2020). CAR-CNN: A deep residual convolutional neural network for compression artifact removal in video surveillance systems. In *15th 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)*.

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *International Conference on Neural Information Processing Systems (NIPS)*.

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)*.

Liu, H., Xiong, R., Zhang, J., and Gao, W. (2015). Image denoising via adaptive soft-thresholding based on nonlocal samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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)*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. In *arXiv preprint arXiv:1411.1784*.

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)*.

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

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

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

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.