A Deep Learning post-processor with a perceptual loss function for video compression artifact removal

D. Ramsook[1], A. Kokaram[2]
SigMedia Group,
Trinity College Dublin
Dublin, Ireland
[1]*ramsookd@tcd.ie,* [2]*anil.kokaram@tcd.ie*

N. Birkbeck,[3] Y. Su,[4], B. Adsumilli,[5]
YouTube/Google,
California, USA
[4]*birkbeck@google.com,* [5]*yeping@google.com,*
[6]*badsumilli@google.com*

2022 Picture Coding Symposium (PCS) | 978-1-6654-9257-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/PCS56426.2022.10018047

***Abstract*—While video compression is necessary for large scale video streaming services, compression at low bitrate can degrade the original video and negatively affect the end user's quality of experience. Deep Neural Networks (DNNs) are actively researched with respect to artifact removal, however the loss functions that are typically employed follows a derivation of a pixel-wise L^p norm. In this paper we consider a DNN as a post-processor for video compression artifact removal. The DNN is trained using a composite perceptual loss that combines a traditional L^p norm loss and a VMAF proxy network based on the Video Multimethod Assessment Function (VMAF). Results show an improvement in VMAF score over both the training and testing sets.**

***Index Terms*—Perceptually Motivated Restoration, Artifact Removal, Video Restoration**

I. Introduction

Video streaming is projected to responsible for 82% of all web traffic in 2022 [1]. While video compression algorithms reduces the bandwidth required to stream video and makes video compression feasible on larger scales, they are responsible for distorting the original video by introducing compression artifacts. These artifacts can be attributed to spatial based (blurring, blocking, ringing and colour bleeding) and temporal based artifacts (flickering and floating) [2]. The introduction of these artifacts result in perceivable degradation in the quality of experience (QoE) of end users.

Deep neural networks (DNNs) have been increasingly successful in video processing tasks such as super-resolution [3], restoration [4]–[6] and compression [7]. This success can be attributed to the research into different network architectures and the their ability to create latent representations that are similar to perceptual stages of the human visual system [8].

While there are major strides in the research of network architecture, there is not much research being done in the loss function used when training. In traditional image/video based neural networks, the loss function is a variant of the L^p norm, where $p \geq 1$ (Eqn. 1), such as the mean squared error (MSE) and mean absolute error (MAE).

This work was conducted with the financial support of 1) The Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224, 2) Faculty award from YouTube/Google and 3) Disruptive Technology Innovation Fund, Enterprise Ireland, Grant No DT-2019-0068. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

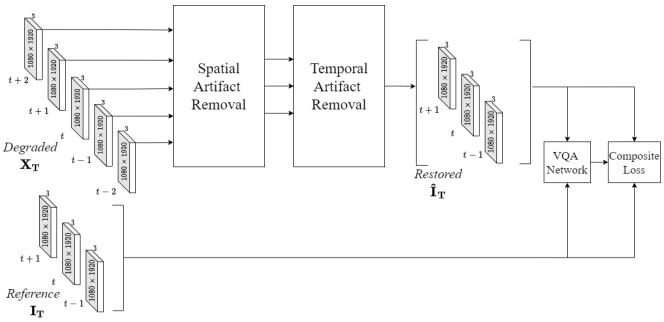


Fig. 1. *Architecture for compression artifact removal post-processor. Inputs during training are 5 1080p 3-channel (YUV) frame sequences. A spatial network is firstly used to process each frame independently. The outputs are windowed and concatenated over three frame sequences and then fed to the temporal network. The restored output from the temporal network is then passed to the VQA Network to produce a perceptual quality score. This score is combined with an L^p loss to form a composite loss. The composite loss is then used to update the weights of the spatial and temporal artifact removal networks.*

$$L^p(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_i |a_i - b_i|^p \right)^{1/p} \tag{1}$$

It is well known that metrics derived from the L^2 norm distances are not well correlated with the perceptual quality from the human visual system (HVS) [9], however it is still widely used in DNN loss functions for its ease of implementation, computability and convergence properties.

The Video Multimethod Assessment Fusion (VMAF) [10] function is a complex non-differentiable perceptual visual quality metric (PVQM) that is better correlated with the nuances of the HVS when compared to traditional metrics. VMAF was developed for compression artifacts and so well suited to the domain of enhancement for video streaming. As VMAF is non-differentiable, it is difficuilt to use as a loss function. Previous work deployed a DNN as an estimator for VMAF, hence rendering an approximate differential [11].

A. Related Work

Zhao et al. [15] presented extensions of traditional mean squared error losses for DNN image restoration. They proposed a composite loss function that mixes traditional mean

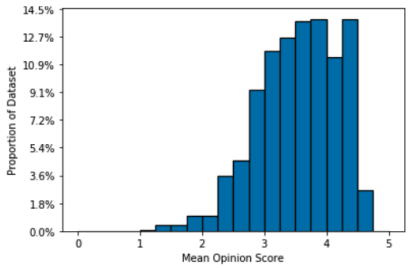


Fig. 2. *Relative frequency histogram of mean opinion scores over the entire dataset. The score has a range between 0 and 5. The bins of the histograms have a width of 0.25.*

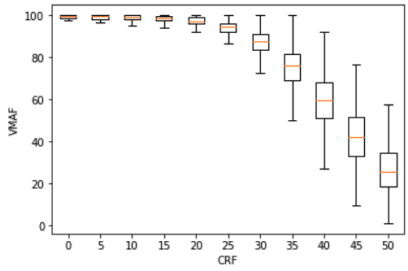


Fig. 3. *Boxplot of VMAF with respect to CRF parameter from H.265 compression (with CRF = 0 : 5 : 50) over all videos from the dataset. Each vertical boxplot shows the distribution of VMAF corresponding to a particular CRF parameter. Even though increasing CRF does increase degradation overall, for a particular clip that might not be the case.*

squared error with readily differentiable perceptual losses for images. Using $\hat{X}, X$ as the restored and reference image respectively their loss function $\mathcal{L}$ is defined as follows, where α is a weighting factor between traditional and perceptual losses.

$$\mathcal{L}^c(\hat{X}, X) = \alpha\mathcal{L}^\lambda(\hat{X}, X) + (1-\alpha)\mathcal{L}^{L^p}(\hat{X}, X) \qquad (2)$$

The composite loss, with $\mathcal{L}^\lambda(\hat{X}, Y)$ being the readily differentiable function MS-SSIM, showed increased performance. We use this approach as the formulation of for our composite loss.

Chadha et al. [16] proposes a deep perceptual preprocessing network for video coding. This network is trained using a mixture of a fidelity loss (L^1, MS-SIM), rate loss and a perceptual model that uses fine-tuned weights of VGG-Net. This model shows promising results as a pre-processor, while our work aims to be a post-processing step. Zhang et al. [17] proposes video compression using a CNN-based post-processor that is trained using an L^1, $SSIM$ and GAN based loss. Results are promising in this approach, however we aim to use a temporally relevant metric such as VMAF.

Chen et al. [18] proposed a two stage compression artifact reduction network (VCAR-CNN). The input to the DCNN is both the frame to be post-process and an auxiliary input frame. That auxiliary input is meant to capture some reasonable texture approximation of the current frame by assembling texture patches from previous motion compensated frames. The DCNN is then used as the second stage of the post-processor. In our case we wish to develop a less hand-crafted process which does not require auxiliary information. In addition VCAR-CNN uses the standard MSE loss while we explore here the use of perceptual losses. We compare our results with VCAR-CNN later and we see that we improve in terms of detail and hence perceptual quality.

TABLE I
Number of 5 frame compressed/uncompressed video clip pairs

Source	Count
YouTube-UGC [12]	14688
derf HD [13]	528
Dolby Atmos [14]	467
Gaming [13]	1440
Foundry	972
Total	18095

B. Contributions

We present a deep neural network compression artifact removal post-processor that is trained using a composite perceptual loss function. The loss combines traditional L^p norm losses and perceptual losses based on a differentiable proxy of VMAF. In addition, we design the network to match the use case of VMAF i.e. that it models perception over the entire image and not small patches. The dataset used in training was created from publicly available video sequences and also cinema quality data collected in collaboration with Foundry [19]. Realistic degradation was generated by compressing the clips with varying constant rate factor (CRF) using the H.265 compressor. The key points are as follows.

- We create a dataset of compressed/uncompressed video pairs that matches a pre-defined real world distribution for visual quality.
- We extend the framework for perceptually motivated image tasks [20] to include a temporal component.
- We present a new architecture for perceptually motivated video restoration with VMAF, based on our previous experience [21], that uses full 1080p resolution frames as input rather than patches.

II. Dataset Curation

Having a large robust dataset that is representative of the actual distribution of perceptual metrics has always been a key problem in supervised video restoration tasks. In traditional approaches, the amount of degradation used is randomly sampled between a set range and then applied to the video. Any desired attributes are then removed or carved from this distribution. This can be problematic as it does not accurately represent the distribution of a real-world scenario and content can be wasted.

In this paper we present a method of generating a compressed/uncompressed dataset that follows a pre-defined distribution. This was done to utilize all clips available. We utilized a combination of publicly available sources and also introduce a collection of cinematic quality clips that was sourced in collaboration with Foundry. Degraded versions of the clips were generated such that the VMAF of the degradation follows

a pre-defined distribution. The distribution used was based on the distribution of the mean opinion scores (MOS) of the YouTube-UGC dataset [12]. Table I shows the clip count of the resultant 5 frame clips curated across the different sources. The dataset can be found here [22].

A. Defining the target distribution

The MOS scores associated with the Youtube-UGC collection [12] was used as the target distribution of our dataset. This was done under the assumption that the MOS scores reported here are a good indicator of the ranges of the perceptual quality in real-life streaming applications. Figure 2 shows the relative frequency histogram of the MOS scores. These scores were then mapped to a range of 0-100 and served as the target distribution.

B. Generating the VMAF target

Only 1080p clips from the YouTube-UGC dataset with a MOS of ≥ 3 were used in collaboration with the other datasources listen in Table I in the remainder of this process.

Recall that we wish to degrade these clips in such as way that the resulting dataset follows our required VMAF distribution. The original clips were then compressed using CRF $0:5:50$ with the H.265 codec to create a large number of realisations of degraded material. Degraded clips were then randomly sampled without replacement from the collection such that they satisfy a certain VMAF score given a CRF value. However, not all clips can be compressed to achieve a specific VMAF value by tuning the CRF parameter because artifacts are heavily content dependent. We ensured that the degradation is possible by modeling the relationship between the CRF and VMAF for any given clip.

Figure 3 shows the distribution of VMAF with respect to CRF over our degraded dataset. We use the following sigmoid to model the observed relationship between CRF x_{crf} and VMAF y_{vmaf}.

$$y_{vmaf} = \frac{A}{1 + e^{-k(x_{crf} - m)}} + b \tag{3}$$

with parameters A, k, m and b. These parameters were estimated for each VMAF Vs CRF relationship per clip using Powell's dog-leg method [23] provided by NumPy.

1) Selecting CRF based on VMAF criteria: We can rearrange equation 3 to estimate the CRF x_{crf} parameter needed for a specific VMAF. If the resulting CRF is valid and within the operating range of $0 - 51$, then the clip was added to a bank of possible clips that can be used to generate data in that VMAF range. These banks were then randomly sampled without replacement to assign clips to different VMAF bins such that the resultant distribution matches the one described in Figure 2.

III. Network Architectures

VQA Network: The VQA network was used as a differentiable estimate to a traditionally non-differentiable complex PVQM. In the context of this paper, we used VMAF as the complex PVQM. The goal of the VQA network is to produce a PVQM score $\hat{y}_v$ given a three frame sequence of restored ($\hat{\mathbf{I}}_\mathbf{T}$) and reference frames ($\mathbf{I}_\mathbf{T}$). This is described in equation 4, where D_v represented the VQA network which is characterised by weights Θ_v. Figure 4(a) shows the architecture of this network.

$$\hat{y}_v = D_v\left(\Theta_v | \hat{\mathbf{I}}_\mathbf{T}, \mathbf{I}_\mathbf{T}\right) \tag{4}$$

The VQA network is trained using the MSE between the actual PVQM score of the metric ($M(\hat{\mathbf{I}}_\mathbf{T}, \mathbf{I}_\mathbf{T})$) and the predicted score: $L_v = MSE\left(\hat{y}_v, M(\hat{\mathbf{I}}_\mathbf{T}, \mathbf{I}_\mathbf{T})\right)$

Artifact Removal Network: The artifact removal network was trained to restore a three frame signal ($\hat{\mathbf{I}}_\mathbf{T}$), from a degraded signal ($\mathbf{X}_\mathbf{T}$). This is described in equation 5, where D_a is the artifact removal network parameterized by weights Θ_a.

$$\hat{\mathbf{I}}_\mathbf{T} = D_a\left(\Theta_a | \mathbf{X}_\mathbf{T}\right) \tag{5}$$

The artifact removal network consists of two sub-models: a spatial and temporal artifact removal network. The spatial artifact removal network, described in Figure 4(b), is used to restore the images with reference to only themselves. The same network is applied in a parallel manner across all frame inputs. The temporal artifact removal network, described in Figure 4(c), restores a single frame given a three frame input. This similar network architecture has been used successfully in previous video restoration tasks [6], [24] The artifact removal network is trained with a composite loss function (L_c) as follows.

$$L_c = \alpha(\beta(L_\lambda(\hat{\mathbf{I}}_\mathbf{T}, \mathbf{I}_\mathbf{T}))) + (1 - \alpha)(L_{L^p}(\hat{\mathbf{I}}_\mathbf{T}, \mathbf{I}_\mathbf{T})) \tag{6}$$

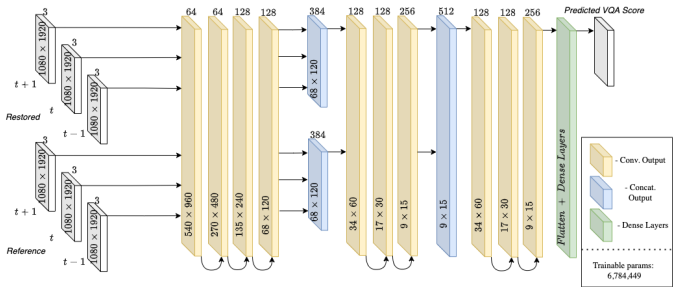
In this loss, L_λ and L_{L^p} are the perceptual and L^p norm losses and α and β are weighting and scaling factors respectively. β scales the perceptual loss so that it is in the same numerical range as the L^p loss, while α controls the weight attributed to the different losses. L_λ is defined as follows, where $M_u = 100$ is the upper limit of VMAF.

$$L_\lambda = MSE(M_u, \hat{y}_v) \tag{7}$$

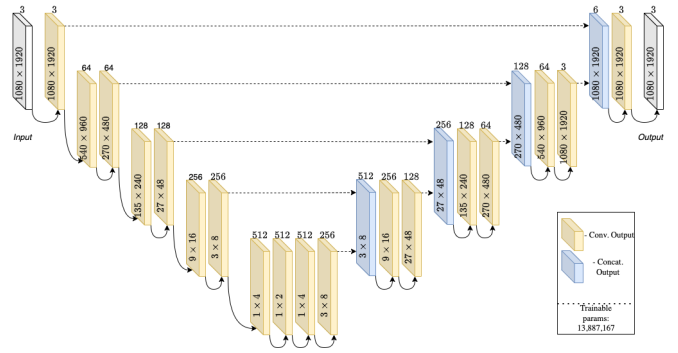
Training Details: The spatial, temporal and VQA networks were trained individually first to initialize the weights and then trained jointly.

The spatial and temporal artifact removal networks were trained for 50 epochs with 10000 training pairs each using a L^1 loss function. Adam optimizers with an initial learning rate of $1e-4$ was used for the first 35 epochs, while a learning rate of $1e-5$ was used for the remainder 15 epochs. Both the spatial and temporal artifact removal networks used a batch size of 10.

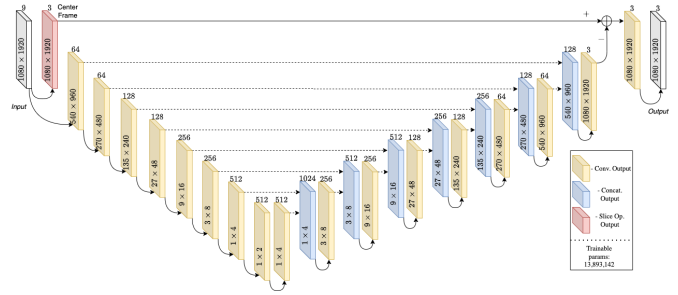
The VQA network was trained for 25 epochs using 5000 training pairs using the L^2 norm loss. An adam optimizer with an initial learning rate of $1e-4$ was used for the first 10 epochs, and a learning rate of $1e-5$ was used for the remainder. A batch size of 5 was used when training the VQA network.



(a) *VQA Network. Three restored and three reference frames are taken as input. Each frame is processed independently to extract latent spatial features by the same weight bank. The outputs across reference and restored branches are then concatenated and processed jointly to extract temporal features. these features are then concatenated across restored and reference branches and further processed to produce a final predicted score.*



(b) *Spatial Artifact Removal Network. Each frame is processed individually by this network to provide a spatially restored version. The architecture is based on the U-Net.*



(c) *Temporal Artifact Removal Network. Three sequential frames are concatenated along the last axis and put as input to this network. The center frame is sliced and used as a residual connection to the output of the U-net.*

Fig. 4. *VQA (a), Spatial Artifact Removal (b) and Temporal Artifact Removal (c) networks. The height and width of each tensor is labelled within each block and the number of channels is placed above each block.*

The entire network was then trained end to end using the composite loss shown in equation 6 for 30 epochs using 10000 training pairs. The value of $alpha$ was set to be $1.5e-3$ for the first 20 epochs and then $1.5e-2$ for the next 5 epochs and finally $1.5e-1$ for the final 5 epochs. The β value was set to be $1e-5$ for all epochs. A normalized clipping of 0.1 is applied to the gradients in this network. The procedure used for training is identical to the one deployed in [21].

TABLE II

Results of our proposed method. Our method has improvement in VMAF, while VCAR-CNN shows improvement in PSNR. This is due to the difference in optimization criteria when training both models.

		Degraded	Our Method	VCAR-CNN [18]
Training	**PSNR/dB**	37.39	36.23	*37.42*
	VMAF	69.22	*69.99*	68.05
Testing	**PSNR/dB**	37.45	36.16	*37.47*
	VMAF	69.36	*69.51*	68.01

IV. Results & Conclusion

After initializing the weights by training the networks separately and then training together using the composite loss function, our model produces an output with higher VMAF scores recorded across the training and test set, however lower PSNR scores that is measured across the original degraded dataset. Results are summarized in Table II. Figure 5 shows a restored example where compression artifacts are visibly removed and a higher VMAF score is achieved and PSNR has decreased. We compare our result against a post-processing compression artifact reduction network (VCAR-CNN [18]). This network used an MSE loss for training and is chosen as it gives an appropriate comparison of a model trained with an L2 norm over output pixel values alone. This is reflected in the results, whereby our method outperforms with respect to VMAF, but VCAR-CNN outperforms with respect to PSNR.

It is known that traditional scores do not correlate well with the HVS [9]. In this paper we look at creating a perceptually motivated artifact removal post-processor for video sequences using a DNN. The network is made of spatial and temporal layers to exploit information within and between frames. The network is trained using a composite loss function that is made of a traditional L^p norm loss and a perceptual loss. In future work this model can be improved by using metrics that exploit features learnt by DNNs such as the LPIPS metric [8].

References

[1] "Cisco visual networking index: Forecast and trends, 2017–2022," *Cisco Report*, p. 13–14, Nov 2018.

[2] Liqun Lin, Shiqi Yu, Liping Zhou, Weiling Chen, Tiesong Zhao, and Zhou Wang, "Pea265: Perceptual assessment of video compression artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3898–3910, 2020.

[3] Man M. Ho, Jinjia Zhou, and Gang He, "Rr-dncnn v2.0: Enhanced restoration-reconstruction deep neural network for down-sampling-based video coding," *IEEE Transactions on Image Processing*, vol. 30, pp. 1702–1715, 2021.

[4] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," in *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, Eds., Cham, 2020, pp. 191–207, Springer International Publishing.

[5] Roman Sizyakin, Viacheslav Voronin, Nikolay Gapon, Marina Pismenskova, and Alexey Nadykto, "A blotch detection method for archive video restoration using a neural network," in *Eleventh International Conference on Machine Vision (ICMV 2018)*, Antanas Verikas, Dmitry P. Nikolaev, Petia Radeva, and Jianhong Zhou, Eds. International Society for Optics and Photonics, 2019, vol. 11041, pp. 230 – 237, SPIE.

[6] Matias Tassano, Julie Delon, and Thomas Veit, "Dvdnet: A fast network for deep video denoising," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1805–1809.

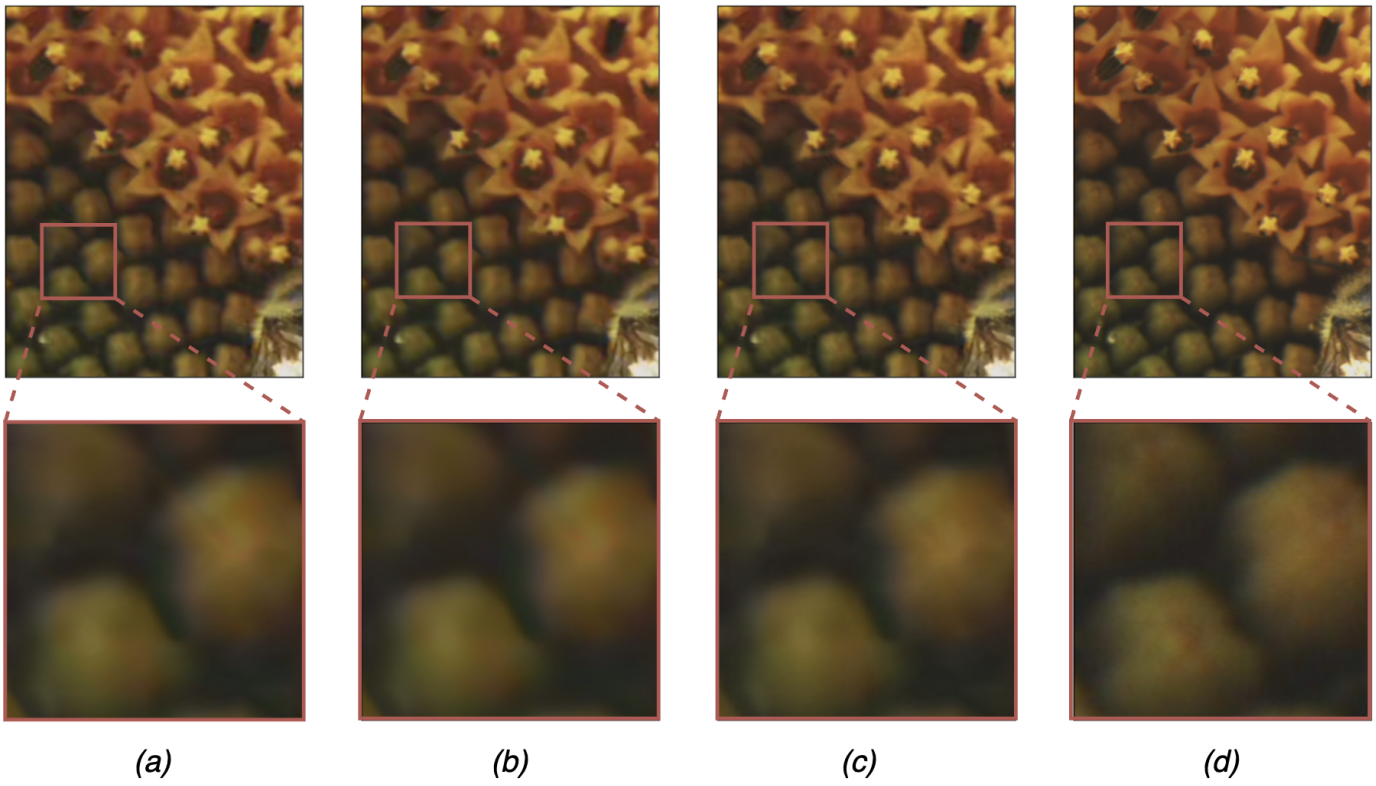


Fig. 5. *Patch of center frame of a restored sequence using our method (c) and VCAR-CNN (b) compared to the degraded (a) and original (d) sequences. (a),(b),(c) has VMAF/PSNR of 57.13/34.63, 55.43/34.92 and 58.21/34.74 respectively. Our restored version has less blocking artifacts which are visible in the zoomed patch region.*

[7] Dong Xu, Guo Lu, Ren Yang, and Radu Timofte, "Learned image and video compression with deep neural networks," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2020, pp. 1–3.

[8] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, jun 2018, pp. 586–595, IEEE Computer Society.

[9] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[10] Netflix, "Netflix/vmaf: Perceptual video quality assessment based on multi-method fusion.," *https://github.com/Netflix/vmaf*.

[11] Darren Ramsook, Anil Kokaram, Noel O'Connor, Neil Birkbeck, Yeping Su, and Balu Adsumilli, "A differentiable estimator of vmaf for video," in *2021 Picture Coding Symposium (PCS)*, 2021, pp. 1–5.

[12] Yilin Wang, Sasi Inguva, and Balu Adsumilli, "Youtube ugc dataset for video compression research," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, 2019.

[13] "Xiph.org video test media [derf's collection]," *https://media.xiph.org/video/derf/*.

[14] "Dolby trailers," *https://thedigitaltheater.com/dolby-trailers/*, Jul 2020.

[15] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[16] Aaron Chadha and Yiannis Andreopoulos, "Deep perceptual preprocessing for video coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14852–14861.

[17] Fan Zhang, Di Ma, Chen Feng, and David R Bull, "Video compression with cnn-based postprocessing," *IEEE MultiMedia*, vol. 28, no. 4, pp. 74–83, 2021.

[18] Wei-Gang Chen, Runyi Yu, and Xun Wang, "Neural network-based video compression artifact reduction using temporal correlation and sparsity prior predictions," *IEEE Access*, vol. 8, pp. 162479–162490, 2020.

[19] Foundry, "Foundry - imagination engineered," *https://www.foundry.com/*.

[20] Li-Heng Chen, Christos G. Bampis, Zhi Li, Andrey Norkin, and Alan C. Bovik, "Proxiqa: A proxy approach to perceptual optimization of learned image compression," *IEEE Transactions on Image Processing*, vol. 30, pp. 360–373, 2021.

[21] Darren Ramsook, Anil Kokaram, Noel O'Connor, Neil Birkbeck, Yeping Su, and Balu Adsumilli, "A differentiable vmaf proxy as a loss function for video noise reduction," in *Applications of Digital Image Processing XLIV*. International Society for Optics and Photonics, 2021, vol. 11842, p. 118420X.

[22] "Artifact-removal-perceptual: Perceptually optimized compression artifact removal for video," *https://github.com/DarrenR96/Artifact-Removal-Perceptual*.

[23] M.J.D. POWELL, "A new algorithm for unconstrained optimization," in *Nonlinear Programming*, J.B. Rosen, O.L. Mangasarian, and K. Ritter, Eds., pp. 31–65. Academic Press, 1970.

[24] Matias Tassano, Julie Delon, and Thomas Veit, "Fastdvdnet: Towards real-time deep video denoising without flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1354–1363.