RT-VENet: A Convolutional Network for Real-time Video Enhancement

Mohan Zhang*
Zhejiang University
zhangmohan@zju.edu.cn

Qiqi Gao
Microsoft Research Asia
gaoqiqi0925@gmail.com

Jinglu Wang
Microsoft Research Asia
jinglwa@microsoft.com

Henrik Turbell
Microsoft
henrik.turbell@skype.net

David Zhao
Microsoft
david.zhao@skype.net

Jinhui Yu
Zhejiang Unviersity
jhyu@cad.zju.edu.cn

Yan Lu†
Microsoft Research Asia
yanlu@microsoft.com

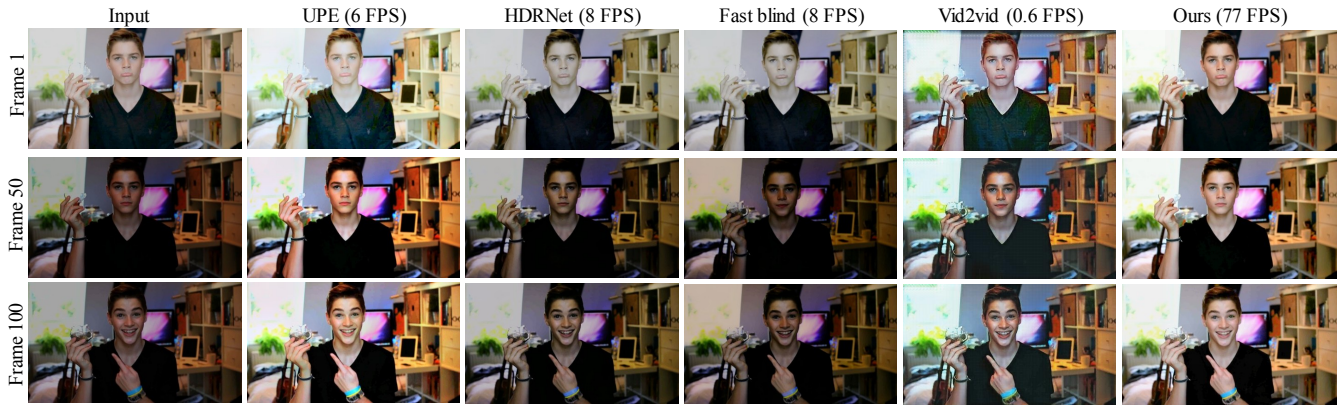


Figure 1: The proposed RT-VENet can perform high-quality and temporal-consistent video enhancement in real-time (77 FPS with an NVidia Tesla P40 GPU or 45 FPS with a CPU on 1080p videos), significantly outperforming the state-of-the-art methods, UPE [52], HDRNet [17], Fast blind [29] and Vid2vid [53] in terms of quality and speed. FPS of other methods on the same GPU.

ABSTRACT

Real-time video enhancement is in great demand due to the extensive usage of live video applications, but existing approaches are far from satisfying the strict requirements of speed and stability. We present a novel convolutional network that can perform high-quality enhancement on 1080p videos at 45 FPS with a single CPU, which has high potential for real-world deployment. The proposed network is designed based on a light-weight image network and further consolidated for temporal consistency with a temporal feature aggregation (TFA) module. Unlike most image translation networks [24, 35] that use decoders to generate target images, our network discards decoders and employs only an encoder and a small head. The network predicts color mapping functions instead of pixel values in a grid-like container which fits the CNN structure well and also advances the enhancement to be scalable to any video resolution. Furthermore, the temporal consistency of the output will be enforced by the TFA module which utilizes the learned temporal coherence of semantics across frames. We also demonstrate that the mapping representation is general to various enhancement tasks, such as relighting, retouching and dehazing, on benchmark datasets. Our approach achieves the state-of-the-art performance and performs about 10 times faster than the current real-time method [17, 52] on high-resolution videos.

*The work was done when Mohan Zhang was an intern at MSR.
†Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413951

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Computational photography**.

KEYWORDS

Image/video enhancement; Retouching; Relighting; Deep learning

ACM Reference Format:
Mohan Zhang, Qiqi Gao, Jinglu Wang, Henrik Turbell, David Zhao, Jinhui Yu, and Yan Lu. 2020. RT-VENet: A Convolutional Network for Real-time Video Enhancement. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413951

1 INTRODUCTION

Real-time video enhancement, targeting at improving visual qualities of live videos, can be applied practically into video communication, augmented reality and robotics. While image enhancement has already been extensively studied and even deployed in the industry, there is still a considerable gap between existing video approaches and the real-world deployment due to two key challenges, i.e., 1) how to reduce the computational complexity and 2) how to enforce the temporal stability.

Recent deep image-to-image translation methods [5, 7, 14, 24, 33, 34, 56–58] can generate high-quality enhanced images. All of them employ generative convolutional neural networks (CNN) with widely used encoder-decoder structures. However, the approaches of generating a high-resolution image with a decoder, which consists of a deep stack of upsampling and conv layers, are quite computationally expensive. Some methods [17, 52] are designed to accelerate the process by dealing with a downsampled image and then recovering the full-resolution output with the edge-aware upsampling [13], but the computation time (about 8 FPS for 1080p videos) is still not satisfactory. Intuitively, enhancement is an easier task than style transfer, because it is supposed to preserve structures of input images, only some of color-related attributes need to be adjusted, such as lighting, tone, and white-balance. Some traditional enhancement methods [25, 36, 44, 44, 59] imitate the means of human retouching, which simply adjust color curves. They are of linear time complexity in image processing but require sophisticated and heuristic algorithms to estimate the curve parameters. Fortunately, deep networks excelling in image analysis could fulfill this mission.

Another main challenge is to preserve temporal consistency in videos, which attracts a lot attention in research areas such as video segmentation [49], style transfer [12], and colorization [61]. Most methods [12, 61] enforce temporal consistency by recovering dense motion [39, 41], i.e., optical flow, between frames, which are of high complexity. Other methods exploit recurrent network structures, such as long short-term memory (LSTM) networks [20, 27, 55]. Although there are some dense prediction works [8, 23, 47, 48] for videos, they are limited to offline scenarios, and their network structures are tailored for specific tasks.

We propose the novel RT-VENet, which can enhance high-resolution videos in real-time. Inspired by the efficient representation of traditional tone-mapping methods [44, 59], we predict similar input-output mapping using a light-weight network without a decoder. The mapping prediction is performed using low-resolution images since only scale-invariant features, such as color distributions and high-level semantics, are considered. Different from the methods [21, 37] adopting a single global mapping with limited capacity of fitting to the whole image, our representation contains multiple tiles of mapping functions, capturing both local variances and global contexts. Each tile contains a piece-wise linear function for mapping the corresponding input patch to the target patch. The mapping functions belonging to such grid-like tiles are constrained in terms of the smoothness of the target image. The task of the network is to predict all the parameters of mapping functions. Eventually, the full-resolution target is reconstructed by applying the predicted mapping functions to the input image. Temporal consistency is enforced by introducing a feature aggregation scheme, by which the extracted features from neighboring frames are fused with respect to spacial feature coherence.

The proposed method can not only produce high-quality results compared with the state-of-the-art methods on the image benchmarks [7, 32], but also outperform existing methods with a large margin on the constructed video dataset. The contributions of this paper are three-fold.

- A novel framework for real-time video/image enhancement benefits from a light-weight CNN model. The proposed method performs about 10 times faster than the existing real-time image methods [17, 52] on 1080p videos, as illustrated in Fig. 1.
- A general representation applicable for various image enhancement tasks, which effectively incorporates global context and local variance into the pixel value mapping. The experiments demonstrate the superior performance of our method on under-/over-exposure correction, retouching, and dehazing datasets.
- A deep feature aggregation scheme for enforcing video temporal consistency and even improving stability in training.

2 RELATED WORK

Automatic image adjustment. Image enhancement has been explored for a long time. Photographers adjusted images by many operations and filters such as highlight, contrast, saturation and hue. Hence, based on supervised learning from paired images, which are obtained before and after editing by an expert photographer, many works aimed at approximating photographers' adjustment skills. They extracted handcrafted features from input images and learned to determine editing parameters with different types of post-processing operations, including global tone adjustment [7], propagating editing[18], color adjustment [57], tone style [51] and tone mapping [2]. More recently, reinforcement learning was also employed to this area [21, 37], which proposed the step-wise interpretable action sequences for the image enhancement.

Image-to-image transformations. Recently, the generative adversarial network [35] showed impressive performance on many challenging tasks. By using the network, Pix2Pix[24] translated an image to another image based on paired images and has shown vigorous potentials to the color enhancement application. For CycleGAN [62] with unpaired training data, the translations were encouraged to be cycle consistent, which could also be applied in the same way to different image-to-image translation tasks. The Adversarial Inverse Graphics Network [50], by utilizing a problem specific renderer, can make use of unpaired data for image-to-image translation as well. Apart from that, the works of [56] [33] and [58] proposed architectures to learn recursive filters for edge-aware smoothing, denoising, and color interpolation. Nevertheless, most of these works incur a heavy computational cost that scales linearly

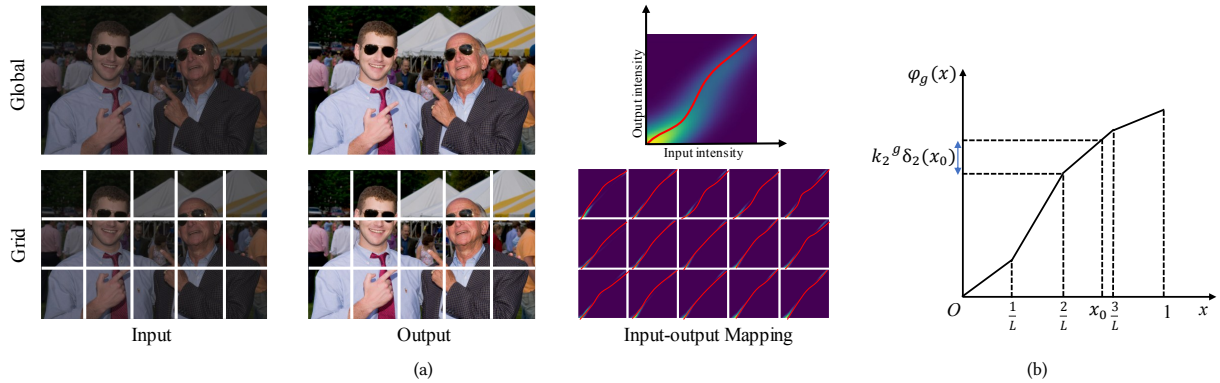


Figure 2: Problem formulation. (a) An example pair of input and user retouched output images. The input-output intensity mappings are plotted on the right, with red curves trying to fit the mappings. We can observe that the global curve (top) is difficult to describe the mapping of all pixels, while curves in the grid tiles (bottom) are more likely to fit to the local mapping in patches. (b) The piece-wise linear function is employed to fit the regional input-output mapping.

with the size of the input image, usually while the large number of stacked convolutions and non-linearities that must be evaluated at full resolution.

Transformation acceleration. Many works were also aimed at accelerating the transformation by applying low resolution and upsampling the result. By using a bilateral filter on a high-resolution guidance map, Kopf et al. [28] produced a piecewise-smooth edge-aware upsampling. Furthermore, Gharbi et al. [17] introduced a neural network architecture that can perform image enhancement on full-resolution images inspired by bilateral grid processing. On the basis of [17], Wang et al. [52] presented intermediate illumination in the network to associate the input with expected enhancement result, instead of directly learning an image-to-image mapping as previous work.

Video enhancement. While image-based CNNs focus on 2D image content, they can not directly adapt to video sequences, suffering from flickering artifacts. A lot methods [12, 30] enforced temporal consistency by employing high-cost estimated dense correspondences between frames. Eilertsen et al.[16] introduced a temporal-aware regularization loss function, but only considering the loss function without network structure is not sufficient to model the consistency in the time dimension. Leveraging recurrent network structure for video data has recently been demonstrated to be powerful in video classification [60] and parsing [49]. There were also a few methods introduced for image-to-image translation, such as video deblurring [47], super-resolution [8, 23]. Recently, some works about video temporal consistency [6, 29, 53] showed impressive performance on many challenging tasks. Methods of [6, 29] could produce the temporally consistent video by taking the original and per-frame processed videos as inputs, while [53] by learning a mapping function from an input source video to a target video. Wang et al. [54] proposed a low light video enhancing method by exploring the high sensitivity camera noise in low light imaging. Chen et al. [10] trained a siamese network on static raw videos, but it does not support other camera data or images after camera IPS i.e., the JPG or PNG data, while our task is training a universal network for any kind of data format. Anyway, all these methods posed the video enhancement problem as dense prediction with the constraints in the spatial and temporal dimension, while inherent lighting mechanism is neglected.

3 APPROACH

We propose the RT-VENet, an end-to-end light-weight CNN, for real-time video enhancement. In this section, we first introduce the formulation of our enhancement problem in Section 3.1, and then present the video network in Section 3.2. Finally, the loss functions for global and local constraints are elaborated in Section 3.3.

3.1 Problem Formulation

Since methods of dense pixel-wise prediction [24, 33, 34, 56] cannot discard the high-cost decoders, estimating color mapping functions [7, 21, 57] is an attractive direction for real-time applications. Nevertheless, employing color mapping functions introduces two challenges. First, a single function applied to the whole image scope is difficult to describe complex image processing operations with local variances. Second, methods of adopting mapping functions are usually based on user assessment [21] or reference images [22], while automatic correction without reference can be ambiguous.

Inspired by classic scale-aware image filters [13, 19], we construct the mapping functions on grid-like tiles, which not only capture finer details in the local region but also fit the structures of stacked convolutional and pooling layers in an encoder. After feeding an image I into the CNN encoder, we obtain a feature map of size $S_x \times S_y$. Each pixel in the feature map corresponds to a tile $g \in \mathcal{G}$ in the grid, and predicts a color mapping function $\phi_g(x)$ for all the input pixels $x \in g$, as illustrated in Fig. 2(b). The mapping function is defined as a piece-wise linear function:

$$\phi_g(x) = \sum_{l=0}^{L-1} k_l^g \delta_l(x), \tag{1}$$

$$\delta_l(x) = \begin{cases} 0, & x \in [0, \frac{l}{L}) \\ x - \frac{l}{L}, & x \in [\frac{l}{L}, \frac{l+1}{L}) \\ \frac{1}{L}, & x \in [\frac{l+1}{L}, 1] \end{cases}$$

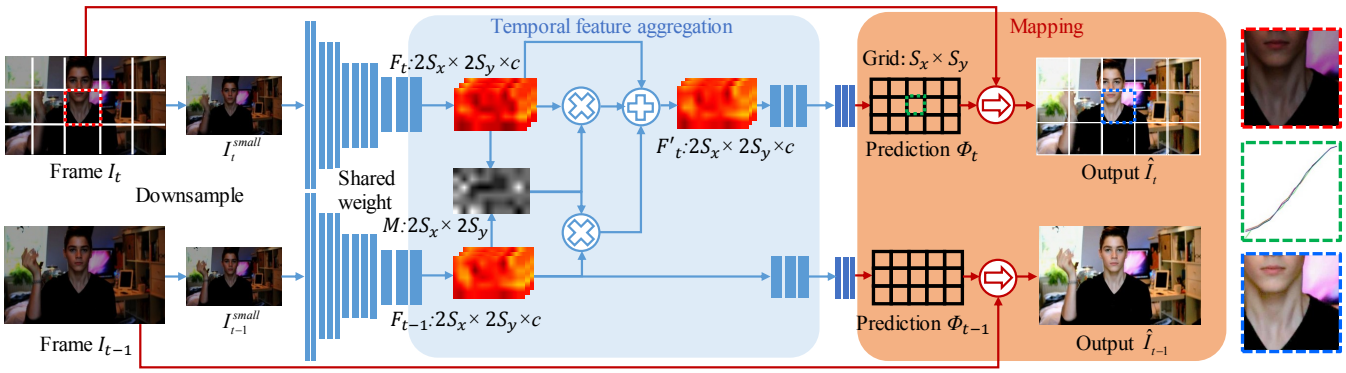


Figure 3: Overview. The proposed RT-VENet is built on a baseline image network (bottom row) and then consolidated with a temporal feature aggregation module. The light-weighted network contains only an encoder and a small head to reduce high cost of conventional decoders. Taking a downsampled image as input, the network predicts input-output mapping coefficients located in grid-like tiles. Each tile outputs a piece-wise linear function (green box on the right). The final full-resolution output is reconstructed by mapping the input using the predicted functions in tiles to preserve high-resolution details.

where L is the number of pieces in the function, $l = 0, ..., L-1$ is the index of each piece, and k_l^g is the slope of each linear function. Thus, the target of the network is to regress a set of coefficients of the mapping functions $\Phi = \{\phi_g | g \in \mathcal{G}\}, \phi_g = [k_0^g, ..., k_l^g, l_{L-1}^g]$.

3.2 Video Enhancement Network

In the following, we will detail the video enhancement network, which is built on a baseline image network and consolidated with a temporal feature aggregation module, as is illustrated in Fig. 3.

Baseline image network. The bottom row of Fig. 3 shows the architecture of the baseline image network. The suffix t or $t-1$ denoting the time stamp in video sequences is omitted in the following description for clarity. The input image I is first downsampled into a small resolution one I^{small} and then fed into the baseline image network to regress the mapping coefficient set Φ. The image network architecture is light-weighted, containing only an encoder similar to MobileNetV2 [42] and a small head. The encoder learns to map the image I^{small} to an embedding space to obtain a latent feature F of shape $S_x \times S_y \times c$ in terms of image context and color variance. The encoder is followed by a head containing three 1×1 conv layers, which aim to transfer the latent feature to color mapping coefficients Φ. The output is of shape $S_x \times S_y \times (c_{in} \times L)$, where c_{in} is the number of input image channels. Finally, the target image $\hat{I}$ is reconstructed by mapping I using the functions Φ.

Temporal feature aggregation. The video network is constructed by inserting a temporal feature aggregation (TFA) module into the baseline image network. The TFA module aims to enforce the temporal stability of the output across frames. To alleviate the flickering artifacts, we embrace the observation that given two continuous frames, the still regions should be paid attention from both frames and output similarly, while the moving regions should be determined by the current frame. The regional stillness is measured by the coherence between frames.

Thus, the temporal aggregation is a form of feature modulation in terms of feature similarities. Recalling that the $\frac{1}{16}$ feature map F of the size $2S_x \times 2S_y$, we use g to indicate the pixel index of the feature map. Let M_g be the matching score of two features, and $f = F_g$ be the $1 \times 1 \times c$ feature of feature map F at position g. The current feature map F_t can be modulated at pixel-level by the previous one F_{t-1} using:

$$f_t^* = w_{co}(f_t + f_{t-1}) \odot M_g + (1 - 2w_{co}) f_t \tag{2}$$

where f_t^* is the modulated feature, w_{co} the weight controlling the impact from the previous frame, $\odot$ the pixel-wise multiplication. As for the coherence map M that measures the confidence of pixel-level feature matching, we adopt the cosine similarity, taking the form:

$$M_g = \cos < f_{t-1}, f_t >= \frac{f_{t-1} \cdot f_t}{||f_{t-1}|| \cdot ||f_t||} \tag{3}$$

where cos is the cosine function, $\cdot$ the inner product operation.

Fig. 3 shows an example of the coherence map M (gray map in the center), where the pixel intensity reveals the confidence of feature matching. The central dark regions represent the moving person and the around light regions represent the still background.

3.3 Loss Function

Given the predicted mapping coefficients Φ and the full-resolution input frame I_t, we can obtain the result $\hat{I}_t$ by applying the transformation. Let $\bar{I}_t$ denote the ground truth. We propose a loss function considering image distance metrics and constraints tailored for the mapping functions. The loss function is made up of four components:

$$\mathcal{L}_t = \lambda^r \mathcal{L}_t^r + \lambda^p \mathcal{L}_t^p + \lambda^s \mathcal{L}_t^s + \lambda^{temp} \mathcal{L}_t^{temp} \tag{4}$$

Reconstruction loss. We employ the L1 loss to measure the reconstruction error.

$$\mathcal{L}_t^r = ||\hat{I}_t - \bar{I}_t||_1 \tag{5}$$

Perceptual loss. We adopt perceptual loss [26] to measure the semantic similarity between the output and ground truth.

$$\mathcal{L}_t^p = ||\Omega(\hat{I}_t) - \Omega(\bar{I}_t)|| \tag{6}$$

where Ω represent the feature extractor of VGG-19 [46] at layer $conv1\_2$. We employ shallow features since the enhancement task focuses more on low-level appearance.

Smooth loss. Since each grid cell predicts mapping coefficients individually, it may produce grid artifacts which are not expected.

Figure 4: Qualitative comparison to state-of-the-art methods on MIT-Adobe 5K dataset. The three rows show retouching (1st row), under-exposed (2nd row) and over-exposed samples (3rd row) respectively.

We therefore introduce a loss to encourage spatial smoothness. For adjacent pixels that are similar in the input image, they should also be similar in the target image. Boundary regions between the tiles should bring more attention. We introduce an edge-aware mask E as a weight map, and combine the smoothness prior [3] to calculate the smooth loss as:

$$\mathcal{L}_t^s = \sum_p \frac{\nabla \hat{I}_t^2}{|\nabla \log I_t|^\theta} \odot E \tag{7}$$

where $\sum_p$ denotes number of pixels in the mask with value of 1, $\odot$ pixel-wise multiplication, and θ the parameter controlling the sensitivity to frame gradients (set $\theta = 1.2$ in experiment).

Temporal-consistency loss. We have already presented the TFA module to preserve the temporal consistency in terms of the network architecture, explicit constraints defined in loss can directly guide the learning of the network. We define the loss to penalize the drastic change in highly matched regions across frames as follows:

$$\mathcal{L}_t^{temp} = \sum_{g \in \mathcal{G}} ||(\phi_g^t - \phi_g^{t-1}) \odot M_g'||_1 \tag{8}$$

where ϕ_g^t and ϕ_g^{t-1} are regressed coefficients in grid cell g from the current frame t and the previous frame $t-1$, M' is the resized coherence map.

4 EXPERIMENT

Implementation. The RT-VENet is built on Tensorflow [1]. The input image is first downsampled to the size of 160×96. The number of linear segments L is set as 16 in our implementation, and thus the output channel number is 48 for 3 (RGB) channels. As described in Fig. 3, the backbone contains five stages, where the first four stages are of the same structure as MobilenetV2 [42] and the last stage contains 3 layers with kernel size $\{3, 1, 1\}$ and strides $\{2, 1, 1\}$. Specific channel numbers will be detailed in supplemental material. The head network is made up of three 1×1 layers with channel numbers $\{192, 96, 48\}$. ReLU activation is applied after all conv layers except the last one which is activated by tanh. The last layer outputs the logarithmic value of the mapping coefficients. Let η denote the output of each channel. Each mapping coefficient is $e^{\beta \times \eta - \alpha}$, where α is set as 1.4 and β is set as 2.5 to control the regression range of mapping functions. With the above representation, our curve range is limited to the range $(e^{-3.9}, e^{1.1})$.

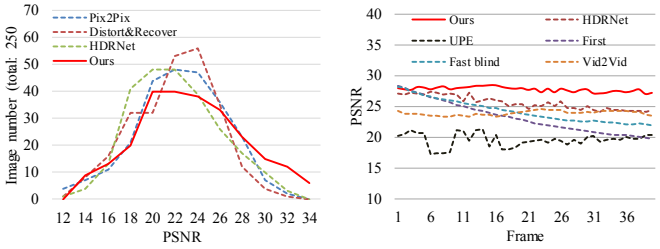
For the hyper-parameters, we set $w_{co} = 0.1$, $\lambda^r = 100$, $\lambda^p = 1$, $\lambda^s = 0.1$, and $\lambda^{temp} = 100$. We use Adam optimizer [15] to train the model with L2 regularization on a single GPU of NVidia Tesla P40. The batch size is set as 32 and the learning rate is 0.005 and decays to 0.001 after 40k iterations.

Metrics. To quantitatively measure the methods, we use PSNR (Peak Signal to Noise Ratio), SSIM (structural similarity index) and MSE (Mean Squared Error) in sRGB space as objective metrics and conduct a user study for subjective measures. Note that we use the SSIM implementation in Tensorflow [1]. Usually, the algorithms with higher PSNR/SSIM and lower MSE have better performance.

4.1 Image Enhancement

Image dataset. To evaluate our method on images, we adopt the benchmark MIT-Adobe 5K dataset [7], which contains 5000 source images, of which each is retouched by five different exports (A/B/C/D/E). The retouched images from expert C are treated as ground truth. We use the same train/test split as [37], with 4750 images for training and 250 for evaluation. The test split is named RANDOM250 following [37].

Quantitative comparison. We compare our method to three representative state-of-the-art image enhancement methods, i.e., transformation acceleration method HDRNet [17], image adjustment method Distort&Recover [37], and image-to-image transformation method Pix2Pix [24]. To provide a fair comparison, all the results are run in their publicly available code and checkpoints with default parameter settings. We use RANDOM250 for image evaluation, and resize the images to maximum side 500px as set in [37]. As is shown in Table 1, our method outperforms others in terms of PSNR and MSE, and is comparable to the best one in terms of SSIM. Fig. 5 (a) shows the PSNR histogram of RANDOM250. Compared with other methods, our method generates more results of higher PSNR, especially for PSNR higher than 30.



(a) PSNR histogram on MIT-Adobe 5K dataset.

(b) PSNR of each frame on Arrow of Time dataset.

Figure 5: Quantitative comparison. (a) Compared with image-based methods, our method obtains more output images with high PSNR on the image dataset. (b) Compared with image-based and video-based methods, our method achieves more stable and higher-quality results on the video dataset.

Model complexity and running time. Our network is of a light-weighted structure and contains about 62M floating point operations (FLOPs), while other networks are of billions of FLOPs. The model size of Tensorflow checkpoint is only 2MB while others are hundreds of MB. As for the running time on a Tesla P40 GPU, our method takes about 7.7 ms per image (the maximum side is 500px), while others are much slower. Note that our network inference time takes only 3.4 ms, which takes the same time for a much higher resolution image, while all the other methods will boost the computation.

Table 1: Running time analysis and quantitative comparison to the state-of-the-art methods on MIT-Adobe 5K dataset.

Method	Time (ms)	PSNR	SSIM	MSE×10^3
Input	-	17.88	0.764	-
HDRNet [17]	33.2	21.99	0.921	8.79
Distort&Recover [37]	1797.1	21.86	0.892	9.08
Pix2Pix [24]	139.8	22.39	0.881	8.72
Ours	7.7	23.21	0.916	8.08

Qualitative comparison on images. We compare our method with the state-of-the-art works of image enhancement [17, 24, 37, 52], as shown in Fig. 4 and Fig. 6. Compared with HDRNet [17], our method presents results with better contrast and color distribution in both foreground and background. UPE [52] could also recover plausible colors and contrast, but it can not deal with the overexposed images as shown in the 3rd row of Fig. 4. Distort&Recover [37] is a global adjustment method, which applies a single function to the whole image, thus limiting the ability of handling more sophisticated adjustment. For example, in the 2nd row of Fig. 4 the foreground and the background should be adjusted with different brightness. For Pix2Pix [24], it can generate good tone and colors for low-resolution input, but it leads to edge distortions and quality degradation when generating high-resolution results as shown in the close-ups of Fig. 6. In contrast, our approach is generally appropriate to various tasks, including relighting and retouching in high quality, and it also has no limitation to image resolutions.

A user study is conducted to evaluate our enhanced images. We randomly select 40 images from Mit-Adobe 5k test set, and compare

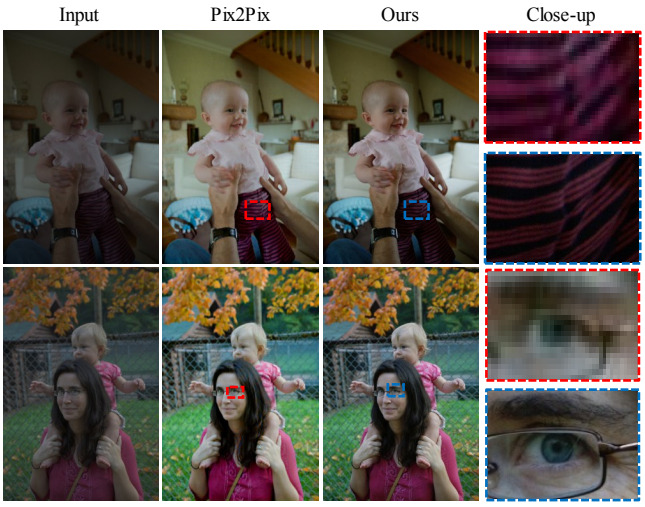
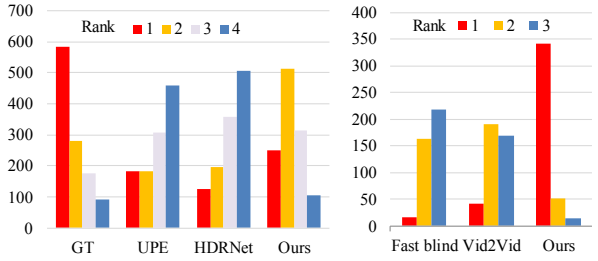


Figure 6: Comparison to Pix2Pix [24]. It can be seen that Pix2Pix could generate vivid color but lead to distorted edges and blurred details at the full resolution due to limitation of regressing output from a decoder, while our method has no distortion and no limitation to image resolutions.



(a) User study on MIT-Adobe 5K dataset.

(b) User study on Arrow of Time dataset.

Figure 7: User study results on (a) image (MIT-Adobe 5K) and (b) video (Arrow of Time) datasets. Y-axis is the number of user selections. Ours are more preferred by human subjects.

our result images with UPE [52], HDRNet [17] and the ground truth. Thirty volunteers who have background knowledge in computer vision or multimedia are recruited as the subjects for this task. For each image, the volunteers are asked to give a rank for the presented results according to the exposure, colors, realistic and details, and the feedback result is illustrated in Fig. 7 (a). The result shows that our enhanced images are the closest to the ground truth compared with UPE and HDRNet.

4.2 Video Enhancement

Video dataset. To evaluate our method on videos, we construct the dataset by synthesizing (details are in the supplemental material). We collect 180 video sequences (about 200 frames in each sequence) from the Arrow of Time dataset [38], which mainly contains people, animals and landscapes. We generate different degrees of under- and over-exposed video clips as inputs and original video clips as ground truth. 31K paired clips (5 frames) from 160 sequences are used for training and the remaining 20 sequences are for testing.

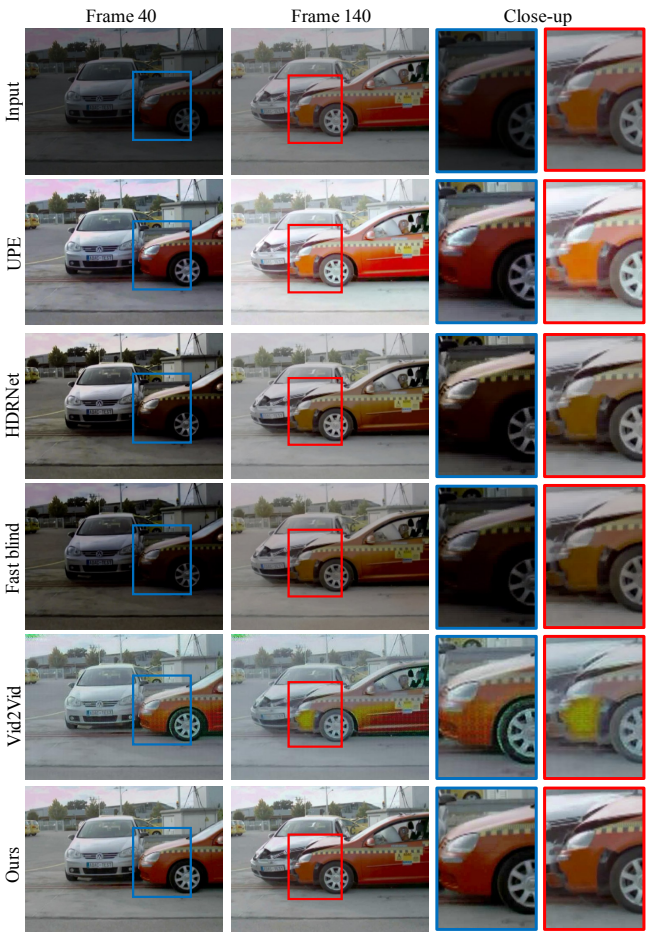


Figure 8: Qualitative comparison on video dataset with state-of-the-art works UPE [52], HDRNet [17], Fast blind [29] and Vid2Vid [53]. Our network can generate more temporal-consistent results.

Quantitative comparison on videos. Considering the real-time application, we compare our method with two recent image method, which could extend to video enhancement in terms of computational time, i.e., UPE [52] and HDRNet [17]. We also compare two video methods, Fast blind [11], Vid2Vid [53], and a naive extended method, which applies the estimated color mapping function of the first frame to all the later frames. Fig. 5 (b) shows the PSNR curves with frame propagation tested on a video sequence from the dataset [38]. The naive method provides high PSNR at first but deteriorates significantly thereafter. The curves obtained from UPE and HDRNet severely fluctuate, suffering from flickering in videos. Besides, compared with Fast blind and Vid2Vid, our method attains the most stable PSNR curve.

We also compare our video enhancement with two methods [29, 53] that show performance on video enhancement in the user study. We use 10 videos randomly selected from the Arrow of Time test set [38]. For each video, we ask the user to rank the results generated from [53] [29] and our method in terms of visual effect. Fig. 7 (b) shows the result based on the feedback from 41 volunteers. The distribution of the result shows that our results are more preferred by human subjects.

Table 2: Running time analysis and quantitative comparison to the state-of-the-art methods on one test video sequence.

Method	Time(ms)	PSNR	SSIM	MSE $\times 10^3$
Vid2Vid [53]	103,150	19.65	0.730	11.36
Fast blind [11]	3,150	18.30	0.912	15.14
Ours (step=1)	390	25.02	0.930	3.80
Ours (step=2)	195	24.39	0.928	4.44
Ours (step=5)	80	23.66	0.921	5.34

As for the running time on 1080p videos, our method runs at 77 FPS on an NVidia Tesla P40 GPU, while the state-of-the-art methods UPE [52] , HDRNet [17], Fast blind [11] and Vid2Vid [53] run at 6, 8, 4 and 0.5 FPS respectively. With a single Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, our network runs at 45 FPS, satisfying the speed requirement of real-time applications. Last but not least, one inferred mapping could be applied to multiple frames. We show the evaluation of applying the same mapping over 1/2/5 continuous frames in table 2. Our method outperforms the state-of-the-art methods in terms of running time, PSNR, SSIM and MSE. Furthermore, the performances do not change too much with frame step larger than 1, but the efficiency could be improved significantly.

Qualitative comparison on videos. Fig. 8 shows the qualititaive comparison with the state-of-the-art methods UPE [52], HDRNet [17], Fast blind [29] and Vid2Vid [53]. The image enhancement methods UPE and HDRNet could not inference frames with temporal consistency, thus suffering from flickering. For the video temporal consistency works, the method of Fast blind takes the original and per-frame processed videos as inputs to produce a temporally consistent video. However, the frame enhancement baseline of [29] is HDRNet [17] which is limited to process frames with different degrees of exposure, thus causing the results as shown in Fig. 8. To predict the current frame result, the method of Vid2Vid requires dense correspondences from the optical flow of the previous frame, thus only enforcing the temporal consistency between consecutive frames, but not long-term temporal consistency, better seen in the close-up part of Fig. 8. In contrast, our method could inference more stable result, even if the input frame interval is more than 100 frames, better seen in our video material.

4.3 Ablation Study

Ablation study on loss functions. As shown in Table 3, ablation experiments are performed to analyze the contribution of different loss functions. The smooth loss forces the enhancement of quality around grid boundaries with marginal improvement, while the perceptual loss pays more attention to tailored features of the enhancement task.

Ablation study on mapping scope. We conduct the ablation experiment on mapping scope by using different grid sizes, i.e., 1×1, 5×3, and 10×6, which represent a global mapping, and different levels of local mappings, namely, the local patches being 1/32 and 1/16 of the input size (160×96). 1×1 grid is constructed with a global pooling layer after the last layer of current encoder, while

Table 3: Ablation study on loss functions on MIT-Adobe 5K dataset.

$\mathcal{L}^r$	$\mathcal{L}^s$	$\mathcal{L}^p$	PSNR	SSIM	MSE $\times 10^3$
✓			22.84	0.855	8.57
✓	✓		22.88	0.855	8.55
✓	✓	✓	23.21	0.857	8.08

Table 4: Ablation study on mapping scopes on MIT-Adobe 5K dataset.

Grid size	PSNR	SSIM	MSE $\times 10^3$
1x1	22.94	0.856	8.54
5x3 (ours)	23.21	0.857	8.08
10x6	22.99	0.855	8.32

10×6 grid is constructed by discarding stage 4 in the encoder. Table 4, where 5×3 grid performs best, shows that the network is not deep enough to capture global context in 10×6 grid and local variances are neglected in 1×1 grid.

Figure 9: Dehazing comparison on RESIDE dataset [32] with existing methods, MSCNN [40], NLD [4], and AOD-Net [31].

4.4 Extension for Dehazing

The RT-VENet is general for many enhancement tasks, and dehazing is an extended application. Following the experiment setting in [32], we train the RT-VENet on RESIDE [32] dataset, which contains 13990 synthetic hazy images from 1399 clear indoor images in NYU2 [45] and Middlebury stereo [43]. We evaluate our method on dehazing task on Synthetic Objective Testing Set (SOTS), which consists of 500 indoor hazy images with 10 different degrees. In Table 5, we compare PSNR and SSIM of RT-VENet with typical dehazing works, and the proposed method achieves promising results in SOTS. We show the visual comparison in Fig. 9 on SOTS, Hybrid Subjective Testing Set (HSTS) and real hazy images. Our method provides the most stable results in various scenarios.

Table 5: Quantitative comparison to the state-of-the-art dehazing methods.

Method	PSNR	SSIM
NLD[4]	17.29	0.782
MSCNN[40]	17.13	0.791
AOD-Net[31]	19.07	0.824
DehazeNet[9]	21.34	0.863
Ours	22.02	0.835

5 CONCLUSION

We have presented a novel convolutional network for real-time video enhancement. The key observation is that although a lot of generative CNNs with the basic encoder-decoder structure are demonstrated to produce nice image-to-image translation results, they are not suitable for the real-time enhancement task. This is because enhancement limits the output to be more faithful to the input and requires the inference speed to be super fast regardless of any resolutions. The proposed RT-VENet utilizes the grid-like CNN structure and traditional tone-mapping methods to introduce a grid-like representation to map the input to output. The RT-VENet discarding the decoder structure performs 10 times faster than the existing fastest method. We further consolidate the temporal stability of the network by enforcing the consistency between matched semantics across frames. The extensive experiments show that our network can produce promising results compared to state-of-the-art methods and save considerable computational cost.

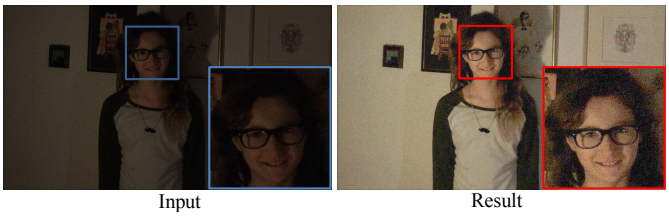


Figure 10: Limitation. Our method could not clear the noise in the enhancment result.

However, our method still has limitation. Our method predicts region-level color mapping functions, but not pixel-level colors, and thus our method could not clear pixel-level noise as shown in Fig. 10. We will target denoising in the future work.

6 ACKNOWLEDGMENTS

Mohan Zhang was partly supported by the Natural Science Foundation of China No. 61772463. We also thank the reviewers and all the people who offered help.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Mathieu Aubry, Sylvain Paris, Samuel W. Hasinoff, Jan Kautz, and Frédo Durand. 2014. Fast Local Laplacian Filters: Theory and Applications. *ACM Trans. Graph.* 33, 5 (Sept. 2014), 167:1–167:14.

[3] Jonathan T Barron and Jitendra Malik. 2014. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence* 37, 8 (2014), 1670–1687.

[4] D. Berman, T. Treibitz, and S. Avidan. 2016. Non-local Image Dehazing. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1674–1682.

[5] Simone Bianco, Claudio Cusano, Flavio Piccoli, and Raimondo Schettini. 2019. Content-Preserving Tone Adjustment for Image Enhancement. In *CVPR Workshops*.

[6] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. 2015. Blind Video Temporal Consistency. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2015)* 34, 6 (2015).

[7] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR 2011*. 97–104.

[8] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4778–4787.

[9] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. 2016. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing* 25, 11 (2016), 5187–5198.

[10] Chen Chen, Qifeng Chen, Minh N Do, and Vladlen Koltun. 2019. Seeing motion in the dark. In *Proceedings of the IEEE International Conference on Computer Vision*. 3185–3194.

[11] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. 2018. Learning to See in the Dark. *CoRR* abs/1805.01934 (2018). arXiv:1805.01934 http://arxiv.org/abs/1805.01934

[12] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*. 1105–1114.

[13] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W Hasinoff. 2016. Bilateral guided upsampling. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 203.

[14] Y. Chen, Y. Wang, M. Kao, and Y. Chuang. 2018. Deep Photo Enhancer: Unpaired Learning for Image Enhancement from Photographs with GANs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6306–6314. https://doi.org/10.1109/CVPR.2018.00660

[15] Jimmy Ba Diederik P. Kingma. 2015. Adam: A method for stochastic optimization. (2015).

[16] Gabriel Eilertsen, Rafal K Mantiuk, and Jonas Unger. 2019. Single-frame Regularization for Temporally Stable CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11176–11185.

[17] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. 2017. Deep Bilateral Learning for Real-time Image Enhancement. *ACM Trans. Graph.* 36, 4, Article 118 (July 2017), 12 pages. https://doi.org/10.1145/3072959.3073592

[18] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2013. Optimizing color consistency in photo collections. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–10.

[19] Kaiming He and Jian Sun. 2015. Fast guided filter. *arXiv preprint arXiv:1505.00996* (2015).

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. 2018. Exposure: A White-Box Photo Post-Processing Framework. *ACM Trans. Graph.* 37, 2, Article 26 (May 2018), 17 pages.

[22] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 172–189.

[23] Yan Huang, Wei Wang, and Liang Wang. 2015. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *Advances in Neural Information Processing Systems*. 235–243.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-Image Translation with Conditional Adversarial Networks. *arxiv* (2016).

[25] D. J. Jobson, Z. Rahman, and G. A. Woodell. 1997. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing* 6, 3 (March 1997), 451–462.

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

[27] Nal Kalchbrenner, Aäron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. 2017. Video pixel networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1771–1779.

[28] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. 2007. Joint Bilateral Upsampling. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) *(SIGGRAPH '07)*. ACM, Article 96.

[29] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *European Conference on Computer Vision*.

[30] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning blind video temporal consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 170–185.

[31] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. 2017. AOD-Net: All-in-One Dehazing Network. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 4780–4788.

[32] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. 2019. Benchmarking Single-Image Dehazing and Beyond. *IEEE Transactions on Image Processing* 28, 1 (Jan 2019), 492–505.

[33] Sifei Liu, Jinshan Pan, and Ming-Hsuan Yang. 2016. Learning Recursive Filters for Low-Level Vision via a Hybrid Neural Network. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 560–576.

[34] J. Long, E. Shelhamer, and T. Darrell. 2015. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

[35] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *CoRR* abs/1411.1784 (2014). arXiv:1411.1784 http://arxiv.org/abs/1411.1784

[36] Nayar and Branzoi. 2003. Adaptive dynamic range imaging: optical control of pixel exposures over space and time. In *Proceedings Ninth IEEE International Conference on Computer Vision*. 1168–1175 vol.2.

[37] Jongchan Park, Joon-Young Lee, Donggeun Yoo, and In So Kweon. 2018. Distort-and-Recover: Color Enhancement using Deep Reinforcement Learning. *CoRR* abs/1804.04450 (2018). arXiv:1804.04450 http://arxiv.org/abs/1804.04450

[38] Lyndsey C. Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Schölkopf, and William T. Freeman. 2014. Seeing the Arrow of Time. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[39] Anurag Ranjan and Michael J. Black. 2016. Optical Flow Estimation using a Spatial Pyramid Network. *CoRR* abs/1611.00850 (2016). arXiv:1611.00850 http://arxiv.org/abs/1611.00850

[40] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. 2016. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*. Springer, 154–169.

[41] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. 2017. Unsupervised Deep Learning for Optical Flow Estimation. In *AAAI*.

[42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4510–4520.

[43] D. Scharstein and R. Szeliski. 2003. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 1. I–I.

[44] S. Shimizu, T. Kondo, T. Kohashi, M. Tsurata, and T. Komuro. 1992. A new algorithm for exposure control based on fuzzy logic for video cameras. *IEEE Transactions on Consumer Electronics* 38, 3 (Aug 1992), 617–623.

[45] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*. Springer, 746–760.

[46] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[47] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. 2017. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1279–1288.

[48] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. 2017. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*. 4472–4480.

[49] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. 2017. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision*. 4481–4490.

[50] Hsiao-Yu Fish Tung, Adam W. Harley, William Seto, and Katerina Fragkiadaki. 2017. Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision. *CoRR* abs/1705.11166 (2017). arXiv:1705.11166 http://arxiv.org/abs/1705.11166

[51] Baoyuan Wang, Yizhou Yu, and Ying-Qing Xu. 2011. Example-based Image Color and Tone Style Enhancement. In *ACM SIGGRAPH 2011 Papers (SIGGRAPH '11)*.

Article 64, 12 pages.

[52] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. 2019. Underexposed Photo Enhancement Using Deep Illumination Estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[53] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[54] Wei Wang, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. 2019. Enhancing Low Light Videos by Exploring High Sensitivity Camera Noise. In *Proceedings of the IEEE International Conference on Computer Vision*. 4111–4119.

[55] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*. 802–810.

[56] Li Xu, Jimmy Ren, Qiong Yan, Renjie Liao, and Jiaya Jia. 2015. Deep Edge-Aware Filters. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1669–1678.

[57] J. Yan, S. Lin, S. B. Kang, and X. Tang. 2014. A Learning-to-Rank Approach for Image Color Enhancement. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2987–2994.

[58] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. 2016. Automatic Photo Adjustment Using Deep Neural Networks. *ACM Trans. Graph.* 35, 2, Article 11 (Feb. 2016), 15 pages.

[59] Lu Yuan and Jian Sun. 2012. Automatic exposure correction of consumer photographs. In *European Conference on Computer Vision*. Springer, 771–785.

[60] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4694–4702.

[61] Bo Zhang, Mingming He, Jing Liao, Pedro V Sander, Lu Yuan, Amine Bermak, and Dong Chen. 2019. Deep Exemplar-based Video Colorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8052–8061.

[62] J. Zhu, T. Park, P. Isola, and A. A. Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 2242–2251.