IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 11, 2021, accepted April 8, 2021, date of publication April 20, 2021, date of current version April 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074199

Dataset and Network Structure: Towards Frames Selection for Fast Video Deblurring

ABDELWAHED NAHLI, SHAN CAO, (Member, IEEE), ZHIWEI JIA, RUNZE MA, AND SHUGONG XU, (Fellow, IEEE)

Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China

Corresponding author: Shan Cao (cshan@shu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62071284, Grant 61871262, Grant 61904101, and Grant 61901251; in part by the National Key Research and Development Program of China under Grant 2017YEF0121400; and in part by research funds from the Shanghai Institute for Advanced Communication and Data Science (SICS).

ABSTRACT Beyond the underlaying unrealistic presumptions in the existing video deblurring datasets and algorithms which presume that a naturally blurred video is fully blurred. In this work, we define a more realistic video frames averaging-based data degradation model by referring to a naturally blurred video as a partially blurred frames sequence, and use it to build REBVIDS, as a novel video deblurring dataset to close the gap between naturally blurred and synthetically blurred video training data, and to address most shortcomings of the existing datasets. We also present DeblurNet, a two phases training-based deep learning model for video deblurring, it consists of two main sub-modules; a Frame Selection Module and a Frame Deblurring Module. Compared to the recent learning-based approaches, its sub-modules have simpler network structures, with smaller number of training parameters, are easier to train and with faster inference. As naturally blurred videos are only partially blurred, the Frame Selection Module is in charge of selecting the blurred frames in a video sequence and forwarding them to the Frame Deblurring Module input, the Frame Deblurring Module in its turn will get them restored and recombine them according to the original order in a newly restored sequence beside their initially sharp neighbor frames. Extensive experimental results on several benchmarks demonstrate that DeblurNet performs favorably against the state-of-the-art, both quantitatively and qualitatively. DeblurNet proves its ability to trade between speed, computational cost and restoration quality. Besides its ability to restore video blurred frames with necessary edges and details, benefiting from its small size and its video frames selection integrated mechanism, it can speed up the inference phase by over ten times compared to existing approaches. This project dataset and code will be released soon and will be accessible through: https://github.com/nahliabdelwahed/Speed-up-video-deblurring-

INDEX TERMS Video deblurring, image deblurring, video frames classification, inference run-time, deep learning, two stages training, CNN, GANs.

I. INTRODUCTION

Video frames deblurring has long been an important problem in computer vision and image processing. Given a motion-blurred or focal-blurred input video, caused by camera shake, object motion or out-of-focus, the aim of deblurring is to recover sharp latent video frames with necessary edges and details.

Video frames deblurring task is highly challenging. Classical methods apply various constraints to model attributes of blur (e.g., non-uniform/uniform/depth-aware), and utilize several natural image priors [1]–[7] to regularize the solution domain. The majority of these methods involve intensive, sometimes heuristic, parameter-tuning and tremendous computations. Moreover, the simplified assumptions on the blur model often limit their performance on real-word examples, where real blur is far more complex than modeled and is entangled with in-camera image processing pipeline.

Learning-based methods have also been adopted for deblurring. Early methods [8]–[10] replace few modules or steps in traditional frameworks with learned parameters to make use of external data. More recent

The associate editor coordinating the review of this manuscript and approving it for publication was Khin Wee Lai.

approaches started to exploit end-to-end trainable networks for image [11] and video [12], [13] deblurring. Among them, Nah *et al.* [11] have achieved state-of-the-art results adopting a multi-scale Convolutional Neural Network. Their method begins from a very coarse scale of a blurry image, and progressively recovers a clear image at higher resolutions until the full resolution is reached. This strategy follows the multi-scale technique in traditional approaches, where the coarse-to-fine pipelines are commonly employed to handle large blur kernels [3].

Despite of the fact that some the early related learning-based video deblurring approaches have already a good enough performance in terms of restoration quality, but most of them still require long run time during the inference phase and a heavy computation cost. The thing that limits their overall performance and makes them unadoptable for time constrained industrial applications. In this paper, we explore DeblurNet, a more robust and fast deep learning-based video deblurring system. Our system consists of two main sub-modules, Frames Selection Module which is a CNN-based model, and Frame Deblurring Module which is a GAN-based architecture. Three major and general challenges in learning-based video deblurring systems are discussed and addressed, which are restoration quality, run time and computational cost.

This work comes with the following main contributions:

1) We introduce DeblurNet, as two stages training-based deep learning model for a fast and robust frame selective video deblurring. Our method can speed up the inference run time by over ten times and still can guarantees a high restoration quality. The found results prove its design and training strategy.
2) We explain the limitations of underlying presumptions in the existing deblurring datasets and methods. We defined a naturally blurred video as a partially blurred sequence of frames, and we introduce REBVIDS as novel video deblurring dataset to address the most shortcomings of the existing deblurring datasets and close the gap between naturally blurred and generated video data.
3) Beyond the commonly used Exponential Decay learning rate and L2 loss functions, we adopt the Cyclic learning rate [65] and Huber loss [54] functions for training our system sub-modules, jointly involving these two training functions prove their ability to prevent over-generalization and promoting the model convergence to a meaningful state within an optimal time.

II. RELATED WORKS

In this section, we briefly review video frame deblurring methods and recent CNN and GAN architectures for image processing.

A. IMAGE/VIDEO DEBLURRING

The non-uniform blur mathematic model is commonly formulated as:

$$I_b = K\,(m) * I_c + n \tag{1}$$

where I_b is a blurred image, $K\,(m)$ are unknown blur kernels function determined by the scene motion field m. I_c is a sharp latent image, $*$ denotes the convolution operator, n is an external additive noise.

Thanks to the pioneering works of Fergus *et al.* [14] and Shan *et al.* [15], many deblurring methods were introduced towards both restoration quality and adaptiveness to variant scenarios and situations. Natural image priors were designed to eliminate artifacts and enhance quality. They adopt total variation (TV) [2], sparse image priors [16], heavy-tailed gradient prior [15], hyper-Laplacian prior [17], l0-norm gradient prior [7], etc. The majority of these early approaches relay on the coarse-to-fine framework. The frequency-domain methods [18], [4] are also exceptional and remarkable, but unfortunately, they are only applicable to a limited range of situations.

Image deblurring task takes advantage of the recent advances in deep CNN. Sun *et al.* [9] utilized the network to forecast blur direction. Schuler *et al.* [8] stacked multiple CNNs in a coarse-to-fine way to perform an iterative optimization. Chakrabarti [19] predicted deconvolution kernel in the frequency domain. These methods follow the classical conventional framework with several components replaced and integrated to the CNN version. Su *et al.* [13] exploited an encoder-decoder network structure with skip-connections to learn video deblurring. Nah *et al.* [11] trained a multi-scale deep learning model to progressively retrieve sharp images. These end-to-end approaches make use of multi-scale information via different architectures.

B. CNNs FOR MOTION DEBLURRING

Unlike classification tasks, deep neural networks for image processing require particular design and layout. As one of the classical approaches, SRCNN [20] employed 3 flat convolution layers with the same feature map size for image super-resolution. Amelioration was achieved by U-net [21], also termed as encoder-decoder network structures [22], which tremendously boosts regression ability and is widely adopted in recent work of FlowNet [23], video deblurring [13], video super-resolution [24], frame synthesis [25], etc. Multi-scale CNN [11] and cascaded refinement network (CRN) [26] simplified training by gradually refining output commencing from a very low scale. They are successful in image deblurring and synthesis, respectively. Reference [27] make use of dilated convolution layers with increasing rates, which approximates increasing kernel sizes.

C. GANS FOR MOTION DEBLURRING

A GAN [28] is an adversarial interaction between two deep learning models: a generator G and a discriminator D, that set up a two-player minimax game. The generator learns to produce artificial fake samples and is trained to mislead the discriminator, with an aim to capture the real data distribution. In particular, as a commonly used GAN variant, conditional GANs [29] have been widely applied to domain transfer tasks, with image enhancement and restoration as

special cases. The minimax game with the value function $V(D, G)$ can be mathematically formulated as the following [9] (real-fake labels set to 1−0):

$$\max_D V(D, G) = \mathrm{E}_{x\sim P_{\mathrm{data}}(x)}\left[\log D(x)\right] + \mathrm{E}_{z\sim P_z(z)}\left[\log\left(1 - D(G(z))\right)\right] \quad (2)$$

Typically, such an objective function is hard to optimize, and one should deal with several challenges, e.g., gradient explosion, gradient vanishing and mode collapse, during the training stage. To address the vanishing gradients and make the training more stable, Least Squares GANs discriminator [30] proposed a loss function that provides smoother and non-saturating gradient. The authors' observations state that the log-type loss in [9] saturates so fast as it neglects the distance between x to the decision boundaries. In contrast to this, an L2 loss delivers gradients proportional to that distance, so that fake samples closer to the boundaries receive less penalties, whereas the far away ones receive larger penalties. The introduced loss function also minimizes the Pearson χ^2 divergence which leads to better training stability.

The LSGAN objective function can be expressed as:

$$\begin{aligned}\max_D V(D) &= \frac{1}{2}\mathrm{E}_{x\sim P_{\mathrm{data}}(x)}\left[(D(x) - 1)^2\right] \\ &\quad + \frac{1}{2}\mathrm{E}_{z\sim P_z(z)}\left[D(G(z))^2\right] \\ \max_D V(G) &= \frac{1}{2}\mathrm{E}_{z\sim P_z(z)}[(D(G(z) - 1)^2] \end{aligned} \quad (3)$$

A further relevant contribution to GANs achieved by the Relativistic GAN [31]. It utilized a relativistic discriminator to estimate the probability that a given real data is more realistic than randomly generated fake data. In comparison to other GAN variants, including WGAN-GP [32] that was used in DeblurGAN-v1 [33], the relativistic discriminator demonstrates more stable training and computationally efficient inference.

III. PROPOSED METHOD

In this section we describe the proposed DeblurNet and its sub-modules, which we refer to as Frames Selection Module and Frame Deblurring Module, we further explain how these two modules are jointly involved to perform a selective fast and robust video deblurring task.

A. FRAME DEBLURRING MODULE

Frame Deblurring Module is a GAN-based network structure, it takes a blurred video frame as input and outputs its restored counterpart. Fig 1 and Fig 2 respectively illustrate the frameworks of their training and inference phases. The Generator layers configuration is detailed in Table 1, its architecture is inspired by ResNet18 and ResNet152 [52] and it is similar to the CNN architecture proposed by Johnson *et al.* [34] for domain transfer task. It contains two strided convolution blocks with stride 1/2, fifteen residual blocks [35] (ResBlocks) and two transposed convolution blocks. Each

TABLE 1. Frame deblurring module: The configuration of its Generator part, it is composed of two convolutional layers (L1 and l2), 15 residual blocks, two convolutional layers (l33 and 34) without skip connection, and three additional convolutional layers (l35, l36 and L37). Each residual block contains two convolutional layers, which are indicated by L(x) and L(x+1) in the table, where "x" equals 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27,29 and 31 respectively for these residual blocks.

layers	Kernel size	Output channels	operations	skip connection
L1	7 x 7	16	ReLU	-
L2	5 x 5	64	ReLU	L4, L36
L3	3 x 3	64	BN + ReLU	-
L4	3 x 3	64	BN	L6
L5	3 x 3	64	BN + ReLU	-
L6	3 x 3	64	BN	L8
L7	3 x 3	64	BN + ReLU	-
L8	3 x 3	64	BN	L10
L9	3 x 3	64	BN + ReLU	-
L10	3 x 3	64	BN	L12
L11	3 x 3	64	BN + ReLU	-
L12	3 x 3	64	BN	L14
L13	3 x 3	64	BN + ReLU	-
L14	3 x 3	64	BN	L16
L15	3 x 3	64	BN + ReLU	-
L16	3 x 3	64	BN	L18
L17	3 x 3	64	BN + ReLU	-
L18	3 x 3	64	BN	L20
L19	3 x 3	64	BN + ReLU	-
L20	3 x 3	64	BN	L22
L21	3 x 3	64	BN + ReLU	-
L22	3 x 3	64	BN	L24
L23	3 x 3	64	BN + ReLU	-
L24	3 x 3	64	BN	L26
L25	3 x 3	64	BN + ReLU	-
L26	3 x 3	64	BN	L28
L27	3 x 3	64	BN + ReLU	-
L28	3 x 3	64	BN	L30
L29	3 x 3	64	BN + ReLU	-
L30	3 x 3	64	BN	L32
L31	3 x 3	64	BN + ReLU	-
L32	3 x 3	64	BN	L36
L33	3 x 3	64	BN + ReLU	-
L34	3 x 3	64	BN	-
L35	1 x 1	64	ReLU	-
L36	1 x 1	64	ReLU	-
L37	1 x 1	64	-	-

TABLE 2. Frame deblurring module: The configuration of its discriminator part. Note, FC means fully connected.

Layers	1-2	3-5	6-9	10-14	15-16	17
kernel	3 x 3	3 x 3	3 x 3	3 x 3	FC	FC
channels	64	128	256	512	4096	2
BN	BN	BN	BN	BN	BN	BN
ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU

ResBlock consists of a convolution layer, instance normalization layer [36], and ReLU [37] activation. Dropout [38] regularization with a probability of 0.5 is added after the first convolution layer in each ResBlock. Moreover, we present an input to output skip connection which we refer to as ResInOut, which guides the model to learn a residual correction $\boldsymbol{I_r}$ to the blurred image $\boldsymbol{I_b}$, so $\boldsymbol{I_c} = \boldsymbol{I_b} + \boldsymbol{I_r}$. This formulation makes training faster and leads the model to generalize better. In order to train this model in end-to-end adversarial manner, we define a discriminator network, which is Wasserstein GAN [39] with gradient penalty [40], its layers configuration is described in Table 2. The discriminator network structure is identical to PatchGAN [41], [42]. All the convolutional layers except the last are followed by Instance Normalization layer and Leaky ReLU [43] with $\alpha = 0.2$.

B. FRAMES SELECTION MODULE

Frames Selection Module is a CNN-based network structure, its framework is illustrated in Fig 3 and its detailed layers configuration is shown in table 3. We design this model to

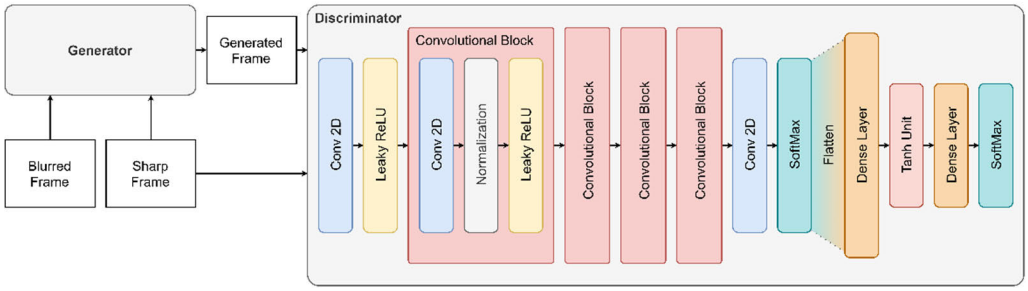


FIGURE 1. Frames deblurring module: A GAN-based deep learning model for frames deblurring, its structure consists of two main parts a discriminator and a generator. The generator architecture is shown Fig. 2.

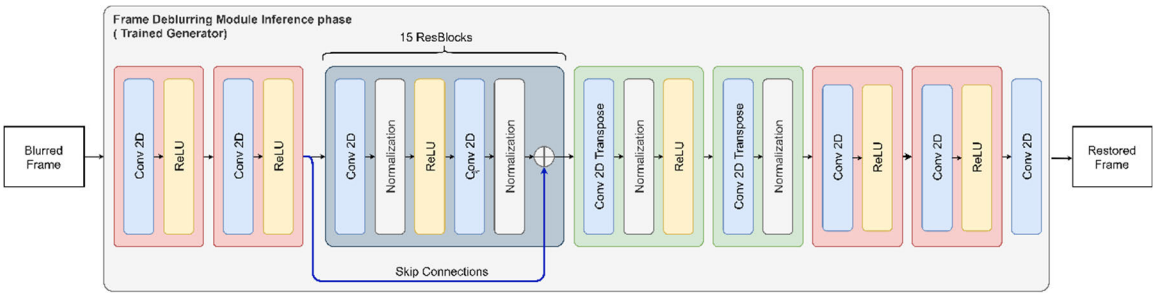


FIGURE 2. The generator part of the frame deblurring module: its architecture is inspired from ResNet18 CNN. It contains 15 ResBlocks and several skip connections.

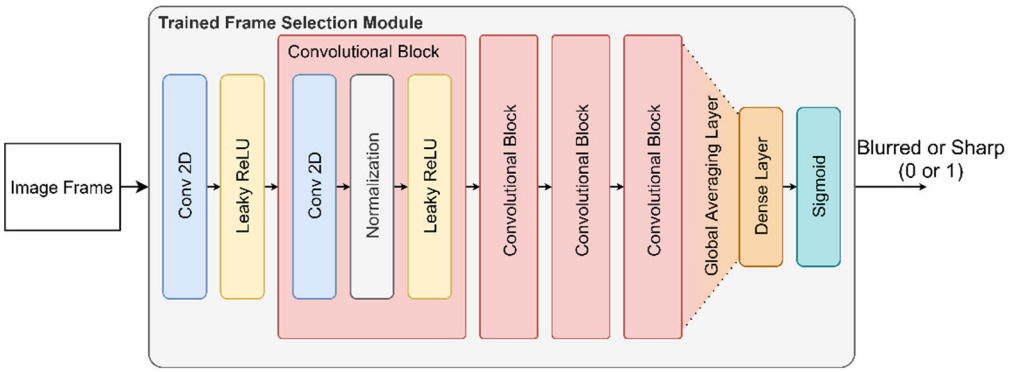


FIGURE 3. Frame selection module: A CNN-based deep learning model for clear/blurred frame classification, its small size, fast and accurate classification ability make it play an important role in our DeblurNet system Fig. 4. Involving this module can dramatically speed up video restoration.

TABLE 3. Frame selection module: The configuration of its discriminator part. Note, FC = fully connected, AP = average pooling.

Layers	1-2	3-6	7-9	10-11	12-13
kernel	5 x 5	3 x 3	3 x 3	AP	FC
channels	64	128	512	4096	2
BN	BN	BN	BN	BN	BN
Activation	ReLU	ReLU	ReLU	ReLU	Sigmoid

perform binary classification on blurred video frames, it splits video frames to two classes, blurred and sharp frames, which gives it the ability to select only the blurred frames to pass a following through the deblurring stage. It contains four strided convolution blocks with stride 1/2. Each convolution block consists of a convolution layer, instance normalization layer [36], and Leaky ReLU [43] activation. The network input layer is a 2D convolution followed by a Leaky ReLU activation function. A global averaging pooling layer and a fully connected net are respectively the last two network layers before a sigmoid function layer at the network output. The architecture of Frames Selection Module is identical to the one introduced in [44]. All the convolutional layers except the last and the first are followed by InstanceNorm layer and LeakyReLU [43] with $\alpha = 0.3$.

C. DEBLURNET

An overall framework of DeblurNet inference phase is shown in Fig 4.

A naturally blurred video can be defined as a sequence consists of blurred and sharp image frames, where the most of them are blurred, which means that every naturally blurred

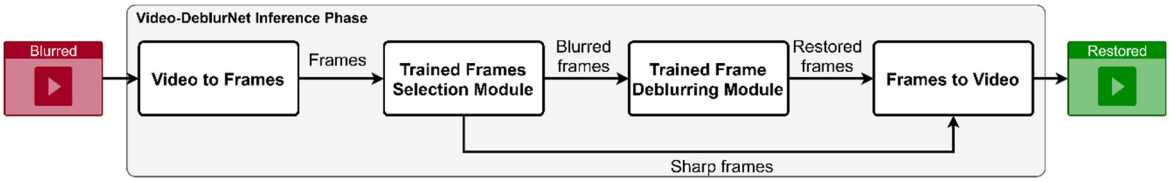


FIGURE 4. DeblurNet: A fast, robust and frame selective video deblurring system, it consists of two essential modules; Frame deblurring module Fig. 1 and frame selection module Fig. 3. This framework illustrates the DeblurNet system inference phase pipelines.

video is not fully blurred but partially blurred, as it always contains an important percentage of clear frames. In order to deblur such a video, it is rational to first classify its frames and separate blurred and clear ones each in a side. Therefore, the Frame Selection Module is designed to perform a binary classification on blurred video frames, whereas the Frame Deblurring Module is designed to be in charge of blurred frames restoration. A trained Frame Selection Module is able to accurately classify each frame in a video sequence to two classes, sharp or blurred. The sharp frames will be saved directly, whereas the blurred ones will be forwarded to a trained Frame Deblurring Module input to get restored. The restored frames in their turn will be saved with the original order in a newly restored sequence beside the initially clear neighbor frames. Compared to Frame Deblurring Module size, Frames Selection Module is a smaller model with fewer training parameters which makes its run-time faster. The ability of the Frames Selection Module to select only the blurred frames for restoration leads our system to save a considerable amount of time and resources. As a result, we get a high quality deblurred video within a minimal possible run time and computations cost. On the contrary to the former advantages of our proposed video deblurring approach, existing methods [45], [46], [45], [46], [74], [75] do not take in consideration the fact that naturally blurred videos are partially blurred, but they blindly pass every and each frame through a video restoration model including sharp frames, which leads to a long run time and heavy computational cost, in addition to this unreliability, passing an already sharp frame through such restoration model mostly damages its quality and cause artifacts or even get it noisy and blurred.

D. LOSS FUNCTIONS

We formulate the loss function as a superposition of content and adversarial loss:

$$l = l_{\text{GAN}} + \lambda l_X \tag{4}$$

where the λ equals 100 in all experiments, l_{GAN} is the adversarial loss and l_X is Huber loss. Unlike Isola *et al.* [41] we do not condition the discriminator as we do not have to penalize the mismatch between the input and output.

1) ADVERSARIAL LOSS

Most of the relevant papers to conditional GANs, adopt vanilla GAN objective as the loss [47], [48] function. Recently [49] suggests an alternative way of using least square GAN [50] which is more stable and generates higher quality results. We use WGAN [51] as the discriminator structure, which is shown to be adequate to the choice of generator architecture [39].

The loss is calculated as the following:

$$l_{\text{GAN}} = \sum_{n=1}^{N} -D_{\varphi_D}(G_{\varphi_G}(I_b)) \tag{5}$$

The Frame Deblurring Module still can converge even if it is trained without adversarial loss, but produces smooth and blurry images, which makes adopting this loss function a must to get sharp restored images.

2) CONTENT LOSS

functions commonly used for regression are $L_1(x) = |x|$ and $L_2(x) = 0.5x^2$. Both of these functions have advantages and disadvantages; L1 is less sensitive to outliers in the data, but it is not differentiable at zero. Whereas, the L_2 is differentiable everywhere, but it is highly sensitive to outliers. Huber proposed the following loss as a compromise between the L_1 and L_2 losses [53]:

$$H_\alpha(x) = \begin{cases} \frac{1}{2}x^2, & |x| \le \alpha \\ \alpha(|x| - \frac{1}{2}\alpha), & x > \alpha \end{cases} \tag{6}$$

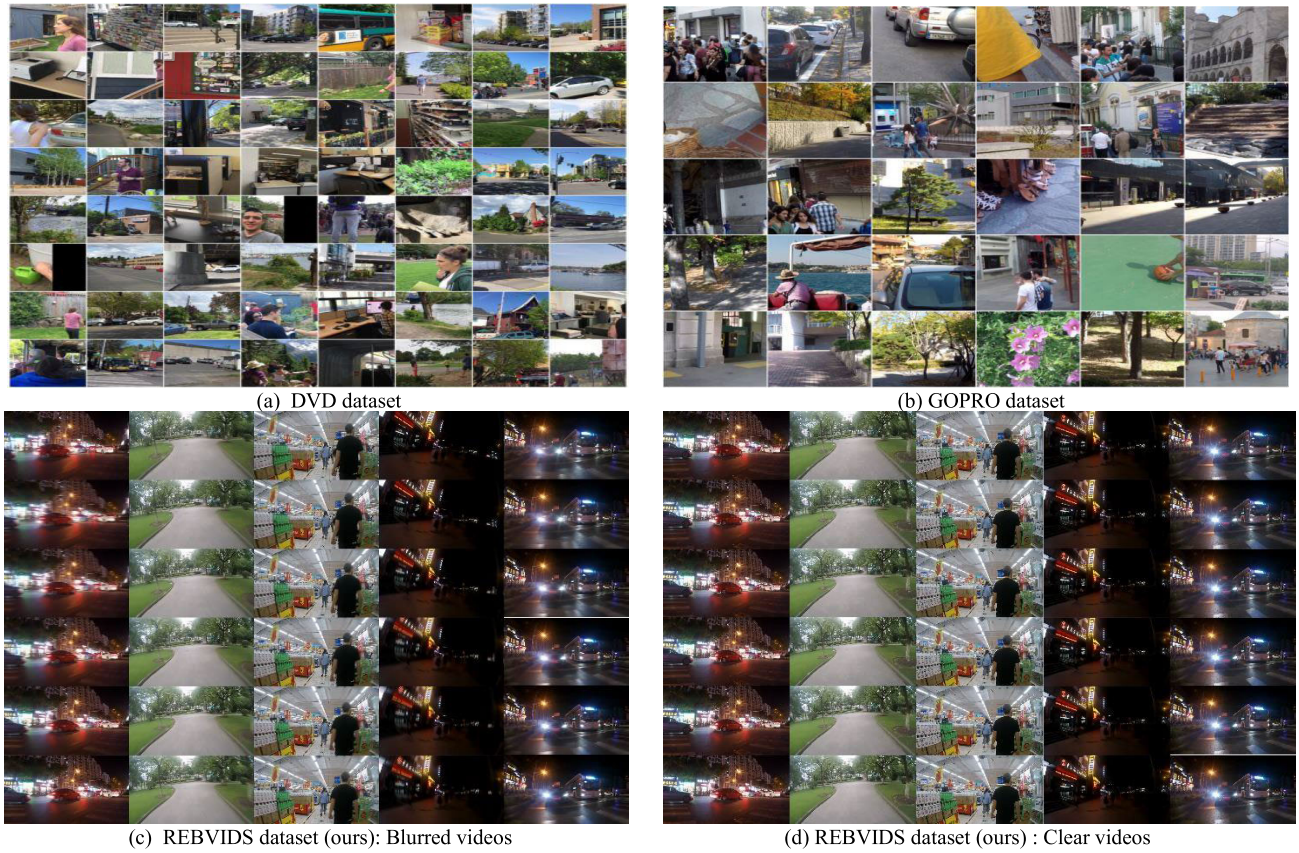
where $\alpha \in R^+$ is a positive real number that controls the transition from L1 to L2. The Huber loss is both differentiable everywhere and robust to outliers. Huber loss function focuses on restoring general content [54], whereas adversarial loss [48] focuses on restoring texture details. Frame Deblurring Module trained without Huber loss or with simple MSE on pixels instead mostly doesn't converge to meaningful state.

IV. EXPERIMENETS AND FOUND RESULTS

In this section, we describe DeblurNet conducted experiments, share their quantitative and qualitative results and compare them with other deep learning-based deblurring methods, also we introduce our novel REBVIDS video deblurring dataset, as well as describe the other existing datasets that are studied in video deblurring literature.

A. VIDEO DEBLURRING DATASETS

In the early studies of video deblurring, only blurry videos have been used for their experiments [55], [56]. As the ground-truth sharp videos were not available, the perceptual



(a) DVD dataset (b) GOPRO dataset

(c) REBVIDS dataset (ours): Blurred videos (d) REBVIDS dataset (ours) : Clear videos

FIGURE 5. (a) and (b) are respectively frames samples from DVD and GoPro video deblurring datasets, whereas (c) and (d) are from REBVIDS (ours).

quality was the primary way to compare different methods. Wulff and Black [57] presented a double-layered blur model that can have different blur statuses in the front and back layer segments. Kohler *et al.* [58] recorded and played back the 6D motion of the camera to capture blurry and reference sharp images. To emulate such a blurring process in more diverse dynamic environments, Kim *et al.* [59] used a high-speed event camera to take an average of sharp frames to synthesize blurs in high resolution (720×1280). Based on the same idea Nah *et al.* [60] extended their data and presented GOPRO dataset consisting of 2103 training and 1111 testing image pairs, presuming gamma function as CRF. In a second attempt to build novel video deblurring dataset based on more realistic data degradation model, Nah *et al.* [61] introduced REDs dataset, which consists of 330 video pairs, each video with 100 frames, he claimed that the used data degradation model can produce more realistic motion blur. Su *et al.* [13] used multi-cameras to present a dataset containing 5708 training and 1000 testing frame pairs. Wieschollek *et al.* [62] collected high-resolved videos from the web, interpolated their frames by means of linear optical flow and down-sampled them to generate smoother blurs for training.

Despite the fact that the above video deblurring datasets share the same aim, which is to build as much as realistic training data based on precisely optimized data degradation models. But also, they share a common shortcoming, they did not pay attention to the fact that a naturally blurred video is not fully blurred, as it always contains a considerable amount of sharp frames. Such data degradation models are only optimized for video spatial dimensions and do not cover its temporal dimension, the thing that limits their performance and prevent them from generating realistic training data. In order to tackle this problem, we built REBVIDS dataset via a well optimized data degradation model that takes in consideration all the above facts.

B. PROPOSED REBVIDS DATASET

By relying on an unrealistic data degradation model, which presume that a naturally blurred video is fully blurred, which means that every and each of their frames is blurred, existing video deblurring datasets synthesized such data via video frames averaging. Unlike this unrealistic assumption we define a naturally blurred video as a partially frames sequence consists of clear and blurred frames, where the most of frames are blurred but not all. We introduce REBVIDS dataset, a novel REalistically Blurred VIdeos from Dynamic Scenes dataset of 720×1280 resolution for training and benchmarking. REBVIDS is meant to complement the existing video deblurring datasets, and to increase the content diversity and provide more realism in the video spatiotemporal degradation, as we focus on making smooth and natural blurs and high-quality reference frames. Paying attention to the fact that a naturally blurred video is partially blurred but not fully blurred, we generate blurred video data by averaging high

TABLE 4. Proposed dataset properties and comparisons.

	Data Partitions			Data Proprieties				Scenes Proprieties			
Datasets	Train	Test	Validate	Video pairs	Frames per video	Frames sequence	FPS of the origin video	Night	Day	Static	Dynamic
GoPro	22	11	0	33	100	Entirely blurred	250	0%	100%	25%	75%
REDs	277	30	3	300	100	Entirely blurred	120	0%	100%	37%	63%
REBVIDS (ours)	319	37	4	360	100	Partially blurred	240 ,120, 90 and 60	36%	64%	27%	73%

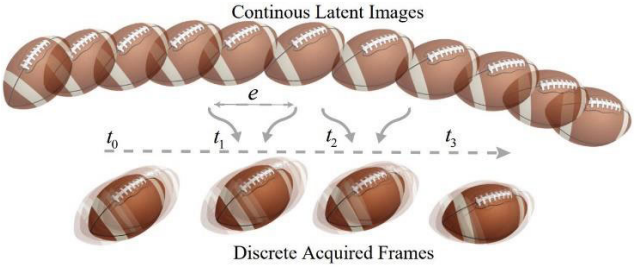


FIGURE 6. Example of frame capturing: Camera sensors acquire the discrete frames at time step t_0, t_1, t_2 and t_3, each of which requires continuous latent images within an exposure time interval e.

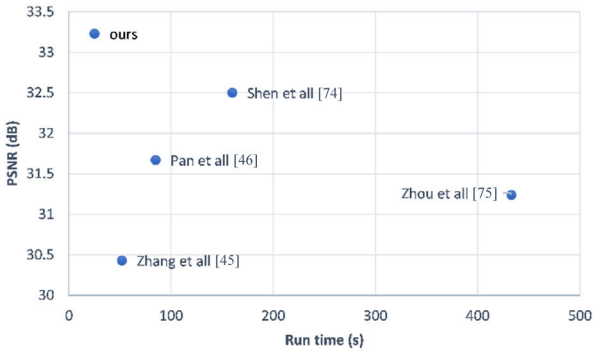


FIGURE 7. Runtime vs. PSNR results on a partially blurred video with 58 blurred and 42 sharp frames from REBVIDS test data.

FPS video frames on a regular interval length, we include sharp frames besides the generated blurred ones throughout the video temporal dimension, by randomly skipping some frame intervals. Table 4 summarizes and compare our dataset properties to other existing datasets.

1) RECORDING

We used XiaoMI MIJIA high speed events camera we manually record 360 RGB video clips, paying attention to the quality of each video clip, diversity contents, lighting and scene dynamics. The early deblurring datasets videos were recorded with a constant high frame rate (240 fps) [23], [28], unlike that we choose to adjust the frame rate according to scene lighting and dynamics. For example, in the day time we record videos within high frame rates range (240 or 100 fps), whereas within a lower frame rates range (60 or 30 fps) for static or slow-motion night scenes, as lower frame rate allows enough light rays to access the camera sensor in such a lighting condition.

2) BLUR SYNTHESIS AND DATA DEGRADATION MODEL

Typically, a camera record video frames by frequently turning on and off their shutter [76]. While the shutter is open, the sensors are exposed to the luminous reflected by the scene objects, their function is to integrate the luminous intensity and acquire the brightness of objects' pixels. Therefore, the pixel brightness depends to exposure time, and the shutter on-off velocity determines video frame rate.

Let is assume that there exists a latent image $L(\tau)$ at each instant time τ, as shown in Fig 6. We average the latent sharp frames from time t_1 over the exposure time interval e to obtain one captured frame. The acquisition of a single frame can be mathematically formulated as:

$$B_{t_1} = \frac{1}{e}\int_{t_1}^{t_1+e} L(\tau)\, d\tau \tag{7}$$

Then at the next shutter opening time t_2, the camera produces another frame denoted by B_{t_2} during a new exposure time interval. The frame rate of the captured video can be defined as:

$$f = \frac{1}{t_2 - t_1} \tag{8}$$

Basically, high speed object motion or camera shake during the exposure time would leads to deteriorating the pixel brightness, these deteriorations are often appearing in form of visual blur.

Relying on (7), which is supposed to be the most realistic data degradation model, we average each video frame in the collected high FPS video dataset to produce its partially blurred counterpart with a frame rate f. In order to ensure that the generated video is partially blurred, and that no more than 70% of its frames are blurred, we did not perform the averaging on the entire video sequence, but we average frame intervals of random length (between 7 to 12 frames) and arbitrarily skip some and exclude them out of the averaging process. We use OpenCV image processing library to imitate a camera imaging pipeline and perform frames averaging process in the signal space.

We down-sample reference videos temporal dimension to fit the synthesized partially blurred ones, then couple them to build a dataset with 360 video pairs in total, each video sequence contains 100 frames. We split the built dataset into three partitions, for training, validating and testing our model.

3) DIVERSITY

We visited various countries, cities and towns, institutes and facilities, theme parks, festivals, palaces and castles, tourist

attractions, historical places, zoos, stores, water parks, etc. to record diverse environments and objects. The contents include static and dynamic scenes, people from various nationalities, crowds, handmade objects, buildings, structures, artworks, furniture, vehicles, colorful textured clothes, and many other objects of different categories with various lighting conditions, day time, night time, morning time, afternoon time and evening time.

4) PARTITIONS

After collecting and processing the 360 video sequences, we split them into training, validation, testing sets. We randomly generated sets of 300 pair of sequences for training, 30 for validation, and 30 for testing until we achieved a good balance in quality. Fig. 5 illustrates some samples from the 30 sequences for validation and testing of the REBVIDS dataset and other communally used deblurring datasets.

C. IMAGE QUALITY METRICS

A considerable attention has been giving to automatic assessment of image quality, and several methods have been proposed. When a carefully generated dataset is available with ground truth frames, we typically focus on the full reference measures. If we have a reference image $\boldsymbol{G}$ with $\boldsymbol{C}$ color channels and $\boldsymbol{H} \times \boldsymbol{W}$ pixels, the quality of a corresponding (degraded or restored) image $\boldsymbol{I}$ can be referred to as the pixel-level fidelity to its ground truth. One of the most communally used image quality metrics is Mean Square Error (MSE) defined as in (9). Another popular measure which is directly relies on MSE is Peak Signal-to-Noise Ratio (PSNR) defined as in (10). Even though, since minimizing MSE is equivalent to forecasting a mean of the solution space, MSE-based restoration models reconstruct blurry images, and they are also vulnerable to even a simple translation.

$$\text{MSE} = \frac{\mathbf{1}}{\boldsymbol{CHW}} \sum_{\boldsymbol{c,h,w}}^{\boldsymbol{C,H,W}} (\boldsymbol{G_{chw}} - \boldsymbol{I_{chw}})^{\mathbf{2}} \tag{9}$$

$$\text{PSNR} = \mathbf{10} log_{10} \frac{255^2}{\text{MSE}} \tag{10}$$

Another kind of referenced metrics measures images similarity in terms of their structure rather than the raw value. While ***MSE*** and ***PSNR*** evaluate the amount of error, the Structural Similarity [77] is a perceptual quality-based model that considers image degradation as changes in the perceived structural information.

The above metrics are not specially designed to evaluate the quality of the restored images or videos from blur. However, they tend to generalize well for a number of types of image distortions as well, so often used to measure the accuracy of many approaches that try to enhance the visual quality.

As deblurring aims to recover the lost detailed textures and the high frequency components from the latent blurred frame, a perfect image quality assessment measure would be the one that reflect fidelity to the ground-truth. However, the ground-truth reference is not available for real blurred data, and the space of possible solutions is very wide. Therefore, plausible and perceptually qualitative restoration results are a must as long as the information from the degraded data is preserved. There are several deblurring research studies which aim to improve perceptual quality with adversarial and perceptual losses [33], [34], [60], [78], [79].

Mostly image and video restoration approaches evaluate their performance on either the image luminance component or RGB channels. Luminance component ($\boldsymbol{Y}$ channel from the ***YCbCr*** color representation) is considered to be with more importance, since the human perception in fact recognizes the texture by the luminance while the variations in chroma components are less sensitive to the human eye. In this work, we measure the image quality using the RGB channels to put more weight on the color as well as the luminance.

D. MODEL TRAINING STRATEGY

We implemented all of our models using TensorFlow [63] deep learning library. The training was performed on a single Maxwell GTX Titan-X GPU using REBVIDS dataset. The models are fully convolutional and can be trained on image patches of arbitrary size. We follow the method of [39] and perform eight gradient descent steps on D_{φ_D}, then one step on G_{φ_G}, choosing Adam [64] as the optimization algorithm. The standard forward and back propagation functions could be formulated as:

$$Y_j = f\left(N_j\right), \text{where } N_j = \sum_j W_{ij} X_i \tag{11}$$

$$\frac{\partial E}{\partial X_i} = \sum_j V_{ij} f'\left(N_j\right) \frac{\partial E}{\partial Y_j} \tag{12}$$

where E is the objective function, W and V respectively denote the feedforward and feedback weight matrices. X denotes the inputs and Y the outputs. W_{ij} and V_{ij} are the feedforward and feedback connections between the j-th output Y_j and the i-th input X_i, respectively. $f(\cdot)$ and $f'(\cdot)$ are the transfer function and its derivative. Whereas $\frac{\partial E}{\partial X_i}$ is the i-th input derivative with respect to the objective function. Algorithm 1, is a pseudocode which simplify the entire training procedure of Frame Deblurring Module.

We adopt Cyclical Learning Rate function [65] during the training phase, as it varies between the minimum and maximum boundaries and each boundary value declines by an exponential factor of $gamma^{iterations}$. The minimum and maximum boundaries respectively are initially set to 0.001 and 0.006 with a cycle length of 2600 iterations. At inference phase, we pursue the idea of [41] and apply both dropout and instance normalization. All the models were trained with a batch size $= 4$, which showed empirically better results on validation. The training stage took 3 days for training DeblurNet system with both of its sub-modules.

Algorithm 1 Frame Deblurring Module (GAN) Training

```
// Assign all network inputs and output
Input: Batch-size,Patterns, iterat-max,learn-rate
output : Generator, Discriminator
Generator == Construct Network Layers()
Discriminator == Construct Network Layers()
for every Network in (Generator, Discriminator)
// Initialize all weights with small random numbers, typically between −1 and 1
Networkweighs == InitializeWeights (Network,Batchsize)
Repeat
    for every pattern in the training set
    Present the pattern to the network
// Propagated the input forward through the Net:(11)
  for each layer in the network
    for every node in the layer
       Calculate the weight sum of inputs to node
        Add the threshold to the sum
        Calculate the activation for the node
    end
  End
// Propagate the errors backward through the Net:(12)
for every node in the output layer
   calculate the error signal
end
for all hidden layer
     for every node in the layer
        Calculate the node's signal error
        Update each node's weight in the network
     end
End
// Calculate Huber Loss (6)
  Calculate the Huber Loss value
// Calculate Adversarial Loss (5)
  Calculate the Adversarial Loss value
// Calculate Global Loss (4)
   Calculate the JointLoss value
 end
  while (iterat-num < iterat-max && loss > specified
Return Generator, Discriminator
```

E. RESULTS ANALYSIS AND DISCUSSION

We evaluate our video deblurring system overall performance on different benchmarked video deblurring datasets. We separately investigate each of its sub-modules robustness and compare their performance with a number of deep learning-based state-of-the-arts.

TABLE 5. Performance and comparison on the REBVIDS test dataset.

Methods	Features	Average accuracy	Top-accuracy
Two-step way [69]	Handcrafted	86.31%	88.78%
Bayes [70]		54.16%	70.07%
SVM [71]		80.22%	82.73%
Single-layered NN [72]	Learned	93.75%	96.99%
DNN [73]		91.41%	94.46%
Frame Selection Module (ours)		95.10%	97.84%

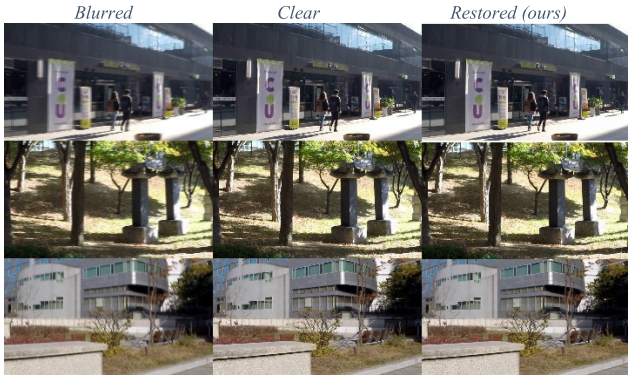


FIGURE 8. Qualitative results on GoPro testing dataset.

1) FRAME DEBLURRING MODULE

results are compared to [66] by Kupyn *et al.*, [67] by Yuan *et al.*, SRN [68] and DeblurGAN [33], we use both standard evaluation metrics (PSNR, SSIM), and also the average run-time on a single GPU to assess the model inference efficiency. The quantitative results are summarized in Table 6, Table 7 and Table 7. In terms of PSNR/SSIM, The Frame Deblurring Module ranked top-one. In particular, Frame Deblurring Module only costs an average of 0.43s per image frame of size 730 × 1280.

2) FRAME SELECTION MODULE

performance is compared to several blur/clear image classification approaches, [69] by Su et al., Bayes [70], SVM [71], Single-layered NN [72] and DNN [73]. The classification accuracy rate is employed metric to evaluate the model performance, which is defined as:

$$\text{Accuraccy} = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{13}$$

where the $N_{correct}$ denotes the number of correctly classified samples, N_{total} indicates the total number of samples to be classified. We measure this module performance using the classification Average-Accuracy (13) and the Top-Accuracy metrics. The comparison results are illustrated in Table 5. The Frame Selection Module obviously outperforms the other approaches on both metrics. Moreover, as this module is based on a small network structure with fewer training parameters compared to the Frame Deblurring Module, it costs only 0.078s per an image frame of size 730 × 1280, which even allows a near real-time video frames classification, for 24-fps

TABLE 6. Performance and inference efficiency comparison on the GoPro test dataset.

	Tao et all [68]	DeblurGAN [33]	Yuan et all [67]	Kupyn et al [66]	Frame Deblurring Model (ours)
PSNR	30.10	28.7	27.10	29.55	**33.58**
SSIM	0.932	0.958	0.900	0.934	**0.962**
Time	1.6s	0.85s	4.33s	0.35s	0.43s
FLOPS	1434.82G	678.29G	N/A	411.34G	527.64G

TABLE 7. Performance and comparison on the reds test dataset.

	Tao et all [68]	DeblurGAN [33]	Yuan et all [67]	Kupyn et al [66]	Frame Deblurring Model (ours)
PSNR	29.93	27.94	25.83	29.73	**30.05**
SSIM	0.924	0.921	0.862	**0.932**	0.926

TABLE 8. Performance and comparison on the REBVIDS test dataset.

	Tao et all [68]	DeblurGAN [33]	Yuan et all [67]	Kupyn et al [66]	Frame Deblurring Model (ours)
PSNR	30.03	28.07	26.42	29.12	**32.25**
SSIM	**0.945**	0.928	0.868	0.918	0.938

TABLE 9. DeblurNet run-time performance on naturally blurred videos and comparison.

Videos			Other methods				DeblurNet(ours)					
			Without frames selection				With frames selection					
			Shen et all [74]	Pan et all [46]	Zhou et all [75]	Zhang et all [45]	Classify every and each frame			Classify only the middle frame of a batch of 15 frames and assume that rest have the same status.		
Video number	Duration	Total frames	Run time (s)				Blurred frames	Sharp frames	Run time (s)	Blurred batches	Sharp batches	Run time (s)
1	6 Seconds	180	288	153	779.4	93.6	60	110	39.84	4	7	26.65
2	10 Seconds	300	480	255	1299	156	120	180	75	8	12	53.16
3	21 seconds	630	1008	535.5	2727.9	327.6	340	290	195.34	22	19	149.39
4	15 Seconds	450	720	382,5	1948.5	234	205	245	123.25	13	14	90.25K

FIGURE 9. Qualitative results on REDs testing dataset.

FIGURE 10. Frame Deblurring Module qualitative results on REBVIDs (ours) testing dataset.

videos. The high classification accuracy, speed, the small model size and the low computational cost are main criteria to choose this model as one of the two principal parts of our video deblurring system.

3) DeblurNet

is a deep learning model based on a two-phase training strategy, it is built after training their main sub-modules and separately evaluating their performance. We set up our video deblurring system and we measured its run time performance on naturally blurred video and on three benchmarked video deblurring datasets with various video lengths and motion blur scenarios. Figure 8, Fig 9 and Fig 10 showcase our model qualitative performance on three different datasets. Whereas Fig 11, Fig 12 and Fig 13 illustrate a qualitative comparison of our deblurring system with four recent related deblurring methods.

We compared DeblurNet performance with the state-of-the art video deblurring methods, [74] by Shen et al., Pan *et al.* [46], Zhou *et al.* [75] and [45] by Zhang et al. The comparison results are shown in Table 9 and Fig 7, our system proves its outperformance on all levels, frame restoration quality, speed and computational cost. The amount of time required by our system to restore a blurred video can be

FIGURE 11. Frame Deblurring Module qualitative results on GoPro testing dataset and comparison with state-of-the-art deblurring methods.

FIGURE 12. Frame Deblurring Module qualitative results on GoPro testing dataset and comparison with state-of-the-art deblurring methods.

FIGURE 13. Frame Deblurring module qualitative results on REBVIDS (ours) testing dataset and comparison with state-of-the-art deblurring methods.

calculated as:

$$T_{\text{run}} = t_d * N_b + t_c * N_a \tag{14}$$

where T_{run} is the system run-time required to restore a video of N_a frames. N_b is the number of frames classified as blurred by the Frame Selection Module, whereas t_d and t_c are respectively the average run time to restore one single frame by the Frame Deblurring Module and the average run time to classify one single frame by the Frame Selection Module. DeblurNet is able to restore a 21s video consists of 630 frames with a size of 730 × 1280 only within 149.39 s. To the best of our knowledge, DeblurNet is the only deblurring method so far that can simultaneously achieve high performance and that high inference efficiency.

V. CONCLUSION

Within this work, beyond the underlying assumptions in the existing deblurring datasets and methods, we have defined a naturally blurred video as a partially blurred frames sequence and introduced REBVIDS as a novel video deblurring dataset to address the most of the shortcomings of the existing deblurring dataset, and to close the gap between naturally blurred and generated blurred video training data. In this paper, we have introduced DeblurNet and explained why it is a proper network structure for a robust, fast and frame-selective video deblurring task, we have also explored an efficient two-phase training strategy for its two main sub-modules. This network sub-models' structures have fewer parameters than previous related ones and are easier to train and faster during the inference phase.

We had discussed and addressed three major and general challenges in learning-based video deblurring systems, which are restoration quality, run time and computational cost. Our approach has achieved state-of-the-arts results, both qualitatively and quantitatively, as it is can accurately restore video blurred frames to their sharp looking with necessary edges and details, and performs favorably against other deblurring methods in terms of two standard image quality assessment metrics. Benefiting from the small model size and its video frames selection integrated mechanism, our approach has reduced the computation cost and speed up video deblurring task by over ten times compared to existing approaches. Unlike other slow deblurring existing methods our work is adoptable for many time constrained industrial applications, e.g., video surveillance and robotics. We argue that our approach is innovative and very inspiring, as it is paving the way to further technical and scientific contributions in computer vision field, we believe that this method can be applied to other computer vision tasks, and we will explore them in future work.

ACKNOWLEDGMENT

The authors would like to thank the entire research team for the constructive discussions, and the (SICS) for the computational resources.

REFERENCES

[1] Y. Bahat, N. Efrat, and M. Irani, "Non-uniform blind deblurring by reblurring," in *Proc. ICCV*, Oct. 2017, pp. 3286–3294.
[2] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, Mar. 1998.
[3] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, p. 145, 2009.
[4] A. Goldstein and R. Fattal, "Blur-kernel estimation from spectral irregularities," in *Proc. ECCV*. Springer, 2012, pp. 622–635.
[5] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via L0-regularized intensity and gradient prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2901–2908.
[6] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2010, pp. 157–170.
[7] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1107–1114.
[8] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, Jul. 2016.
[9] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 769–777.
[10] L. Xiao, J. Wang, W. Heidrich, and M. Hirsch, "Learning high-order filters for efficient blind deconvolution of document photographs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 734–749.
[11] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.
[12] T. H. Kim, K. M. Lee, B. Scholkopf, and M. Hirsch, "Online video deblurring via dynamic temporal blending network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4038–4047.
[13] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1279–1288.
[14] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006.
[15] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, p. 73, 2008.
[16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1964–1971.
[17] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. NIPS*, 2009, pp. 1033–1041.
[18] M. Delbracio and G. Sapiro, "Burst deblurring: Removing camera shake through Fourier burst accumulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2385–2393.
[19] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 221–235.
[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 184–199.
[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI)*. Springer, 2015, pp. 234–241.
[22] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. NIPS*, 2016, pp. 2802–2810.
[23] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[24] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4472–4480.
[25] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4463–4471.
[26] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*. Springer, 2017, pp. 1511–1520.
[27] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*. Springer, 2017, pp. 2497–2506.

[28] J. Ian Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Jun. 2014, *arXiv:1406.2661*. [Online]. Available: https://arxiv.org/abs/1406.2661

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[30] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," 2016, *arXiv:1611.04076*. [Online]. Available: http://arxiv.org/abs/1611.04076

[31] A. Jolicoeur-Martineau, "The relativistic discriminator: A key element missing from standard GAN," 2018, *arXiv:1807.00734*. [Online]. Available: http://arxiv.org/abs/1807.00734

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[33] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192, doi: 10.1109/CVPR.2018.00854.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: http://arxiv.org/abs/1512.03385

[36] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*. [Online]. Available: https://arxiv.org/abs/1607.08022

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[39] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," Jan. 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," Mar. 2017, *arXiv:1704.00028*. [Online]. Available: https://arxiv.org/abs/1704.00028

[41] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, *arXiv:1611.07004*. [Online]. Available: https://arxiv.org/abs/1611.07004

[42] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," Apr. 2016, *arXiv:1604.04382*. [Online]. Available: https://arxiv.org/abs/1604.04382

[43] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1–9, doi: 10.1145/3065386.

[45] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 291–301, Jan. 2019, doi: 10.1109/TIP.2018.2867733.

[46] J. Pan, H. Bai, and J. Tang, "Cascaded deep video deblurring using temporal sharpness prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3043–3051.

[47] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2016, *arXiv:1609.04802*. [Online]. Available: http://arxiv.org/abs/1609.04802

[48] S. Nah, T. Hyun, K. Kyoung, and M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3883–3891.

[49] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag, 2010.

[50] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," 2016, *arXiv:1611.04076*. [Online]. Available: http://arxiv.org/abs/1611.04076

[51] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," Mar. 2017, *arXiv:1704.00028*. [Online]. Available: https://arxiv.org/abs/1704.00028

[52] S. Nah, T. Hyun, K. Kyoung, and M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2791–2799.

[53] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, Mar. 1964.

[54] G. P. Meyer, "An alternative probabilistic interpretation of the Huber loss," 2019, *arXiv:1911.02088*. [Online]. Available: https://arxiv.org/abs/1911.02088

[55] S. Cho, J. Wang, and S. Lee, "Video deblurring for hand-held cameras using patch-based synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, p. 64, Aug. 2012.

[56] T. H. Kim and K. M. Lee, "Generalized video deblurring for dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5426–5434.

[57] J. Wulff and M. J. Black, "Modeling blurred video with layers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 236–252.

[58] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 27–40.

[59] T. H. Kim, S. Nah, and K. M. Lee, "Dynamic video deblurring using a locally adaptive blur model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2374–2387, Oct. 2018.

[60] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.

[61] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. M. Lee, "NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1996–2005, doi:10.1109/CVPRW.2019.00251.

[62] P. Wieschollek, M. Hirsch, B. Scholkopf, and H. P. A. Lensch, "Learning blind motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 231–240.

[63] *TensorFlow*. Accessed: Nov. 9, 2015. [Online]. Available: https://tensorflow.org/

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[65] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 464–472, doi: 10.1109/WACV.2017.58.

[66] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-V2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8878–8887.

[67] Q. Yuan, J. Li, L. Zhang, Z. Wu, and G. Liu, "Blind motion deblurring with cycle generative adversarial networks," *Vis. Comput.*, vol. 36, pp. 1591–1601, Oct. 2019.

[68] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.

[69] B. Su, S. Lu, and C. L. Tan, "Blurred image region detection and classification," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, Scottsdale, AZ, USA, 2011, pp. 1397–1400.

[70] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[71] R. Wang, R. Li, Y. Lei, and Q. Zhu, "Tuning to optimize SVM approach for assisting ovarian cancer diagnosis with photoacoustic imaging," *Bio-Med. Mater. Eng.*, vol. 26, no. s1, pp. S975–S981, 2015.

[72] C. Butakoff and V. N. Karnaukhov, "Blurred image restoration using the type of blur and blur parameter identification on the neural network," *Proc. SPIE*, vol. 4667, pp. 460–471, May 2002.

[73] R. Yan and L. Shao, "Blind image blur estimation via deep learning," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1910–1921, Apr. 2016.

[74] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Blurry video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5114–5123.

[75] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, 2482–2491, doi: 10.1109/ICCV.2019.00257.

[76] J. Telleen, A. Sullivan, O. Wang, P. Gunawardane, I. Collins, and J. Davis, "Synthetic shutter speed imaging," *Comput. Graph. Forum*, vol. 26, pp. 591–598, 2007, doi: 10.1111/j.1467-8659.2007.01082.x.

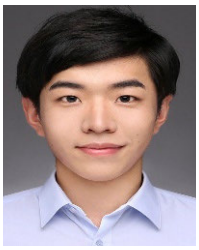
[77] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[78] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
[79] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 372–386.

ABDELWAHED NAHLI was born in Casablanca, Morocco, in 1992. He received the B.S. degree in computer engineering from the University of Hassan II, Casablanca, in 2015. He is currently pursuing the M.S. degree in information and communication engineering with Shanghai University, Shanghai. He is also a Research Fellow with the Shanghai Institute for Advanced Communication and Data Science (SICS). His research interests include computer vision, artificial intelligence, and data science.

SHAN CAO (Member, IEEE) received the B.S. and Ph.D. degrees in microelectronics from Tsinghua University, China, in 2009 and 2015, respectively. She was a Postdoctoral Researcher with the School of Information and Electronics, Beijing Institute of Technology, from 2015 to 2017. She is currently an Assistant Professor with Shanghai University. Her current research interests include wireless communication systems, channel encoding and decoding, and machine learning acceleration and its ASIC design.

ZHIWEI JIA received the B.E. degree from the Department of Communication Engineering, Shanghai University, Shanghai, China, in 2019. He is currently pursuing the master's degree in information and communication engineering with Shanghai University. His research interests include scene text recognition, and low quality text image recovery.



RUNZE MA received the bachelor's degree in communications engineering from Shanghai University, China, in 2018, where he is currently pursuing the master's degree in electronic engineering. His research interests include speech enhancement, speech separation, and so on.

SHUGONG XU (Fellow, IEEE) received the degree from Wuhan University, China, in 1990, and the master's degree in pattern recognition and intelligent control and the Ph.D. degree in electrical engineering from the Huazhong University of Science and Technology (HUST), China, in 1993 and 1996, respectively. He is currently a Professor with Shanghai University and the Head of the Shanghai Institute for Advanced Communication and Data Science (SICS). He was the Center Director and the Intel Principal Investigator of the Intel Collaborative Research Institute for Mobile Networking and Computing (ICRI-MNC), prior to December 2016 when he joined Shanghai University. Before joining Intel in September 2013, he was a Research Director and a Principal Scientist with the Communication Technologies Laboratory, Huawei Technologies. Among his responsibilities at Huawei, he founded and directed the Huawei's Green Radio Research Program, Green Radio Excellence in Architecture and Technologies (GREAT). He was also the Chief Scientist and a PI for the China National 863 Project on End-to-End Energy Efficient Networks. He was a one of the co-founders of the Green Touch Consortium together with Bell Labs and a Co-Chair of the Technical Committee for three terms in this international consortium. Prior to joining Huawei in 2008, he was with Sharp Laboratories of America, as a Senior Research Scientist. Before that, he conducted research as a Research Fellow with The City College of New York, Michigan State University, and Tsinghua University. He has published over 100 peer-reviewed research articles in top international conferences and journals. One of his most referenced articles has over 1400 Google Scholar citations, in which the findings were among the major triggers for the research and standardization of the IEEE 802.11S. He has over 20 U.S. patents granted. Some of these technologies have been adopted in international standards, including the IEEE 802.11, 3GPP LTE, and DLNA. His current research interests include wireless communication systems and machine learning. He was elevated to IEEE Fellow in 2015 for contributions to the improvement of wireless networks efficiency. He was awarded the National Innovation Leadership Talent by the China Government in 2013. He is also the Winner of the 2017 Award for Advances in Communication from the IEEE Communications Society.