symmetry

MDPI

Article

Contiguous Loss for Motion-Based, Non-Aligned Image Deblurring

Wenjia Niu, Kewen Xia * and Yongke Pan

School of Electronic and Information Engineering, Hebei University of Technology, Tianjin 300401, China; niuwenjia9064@hotmail.com (W.N.); panyongke@hotmail.com (Y.P.)
* Correspondence: kwxia@hebut.edu.cn; Tel.: +86-1307-679-7365

Abstract: In general dynamic scenes, blurring is the result of the motion of multiple objects, camera shaking or scene depth variations. As an inverse process, deblurring extracts a sharp video sequence from the information contained in one single blurry image—it is itself an ill-posed computer vision problem. To reconstruct these sharp frames, traditional methods aim to build several convolutional neural networks (CNN) to generate different frames, resulting in expensive computation. To vanquish this problem, an innovative framework which can generate several sharp frames based on one CNN model is proposed. The motion-based image is put into our framework and the spatio-temporal information is encoded via several convolutional and pooling layers, and the output of our model is several sharp frames. Moreover, a blurry image does not have one-to-one correspondence with any sharp video sequence, since different video sequences can create similar blurry images, so neither the traditional pixel2pixel nor perceptual loss is suitable for focusing on non-aligned data. To alleviate this problem and model the blurring process, a novel contiguous blurry loss function is proposed which focuses on measuring the loss of non-aligned data. Experimental results show that the proposed model combined with the contiguous blurry loss can generate sharp video sequences efficiently and perform better than state-of-the-art methods.

Keywords: motion based image; image deblurring; conventional neural networks; contiguous blurry loss; spatio-temporal framework

Citation: Niu, W.; Xia, K.; Pan, Y. Contiguous Loss for Motion-Based, Non-Aligned Image Deblurring. *Symmetry* **2021**, *13*, 630. https://doi.org/10.3390/sym13040630

Academic Editor: Dumitru Baleanu

Received: 24 February 2021
Accepted: 4 April 2021
Published: 9 April 2021

1. Introduction

Blurry images can be caused for many reasons, including rapidly changing scenes or motion blur, camera shaking and depth variations [1]. Unpleasant blurring can destroy the reminiscence between people in photos and in real-life. In fact, a photo is not only an instant in time as it is commonly referred to. A photo needs exposure over a certain period of time to gather light from the scene; hence, there can be some motion occurring in the scene while it is being captured as a single photo.

Image deblurring is a classical problem in the field of computer vision. Given a single blurry image, deblurring is the process of estimating its corresponding sharp images. Past methods focused on removing blur kernels arising from one translation [2]. Recently, far more studies focused on recovering sharp images caused by depth variation and camera shaking in dynamic environments [3]. The majority of these methods were blur model-based:

$$I_B = K \cdot I_S + N \tag{1}$$

in which I_B represents the blurry image and K denotes a mixed unknown blur kernel. I_S is the corresponding sharp image, $\cdot$ represents the convolutional process and N is the often concomitant additive noise.

In practice, it is hard to model a blur kernel for each pixel, which is regarded as an ill-posed problem [4]. Some research works ascribe the blur merely to 3D convolution [5,6], while ignoring camera shake and the possibility of having multiple fast moving objects

in a dynamic scene captured as one photo. A blurry image captured as the result of such scene dynamically is called a motion-blurred image and the estimation of a blur kernel for the purpose of deconvolution is very problematic. Kim et al. [7] proposed a method that estimates the latent images and locally linear motion. However, the estimated blur kernels using their method are not accurate for abrupt motion [8]. Recently, convolutional neural networks (CNN) have achieved remarkable success in the fields of computer vision, including the deblurring problem. Some CNN-based methods utilize the unified blur kernel (a procedure directly applicable to any image with any blur) to synthesize blurry images for training [9–11], whereas other default methods estimate the blur kernel as a local linear kernel [12].

In addition, the method of [13] is trained based on the pixel2pixel loss and perceptual loss functions, which are proposed for aligning data. This is essentially different from our approach, which is designed for non-aligned data, as one motion-blurred image does not have one-to-one correspondence with a sharp video sequence. As Figure 1 shows, one motion-blurred image can be created based on different sharp video sequences. To deal with this problem, the contiguous blurry loss is proposed, which targets single blurred images and their corresponding sharp video sequences. The key to contiguous blurry loss is to treat both sharp and blurry images as collections of features. Our loss function contains two parts, of which one ignores the spatial positions of the features and focuses on measuring the similarity of sharp and blurry images on the basis of their similar features. The other part complements the loss caused by spatial positions at the pixel2pixel level.

Overall, our contributions are summarized as follows.

- Firstly, we developed a new structure to train a single generative model to recover sharp video frames from one motion-blurred image.
- Secondly, we introduced a contiguous blurry loss to constrain the estimation process, addressing the nonalignment problem between blurry images and sharp video sequences.
- Thirdly, the experiment results show that our framework can generate sharp image sequences and achieve state-of-the-art performance.

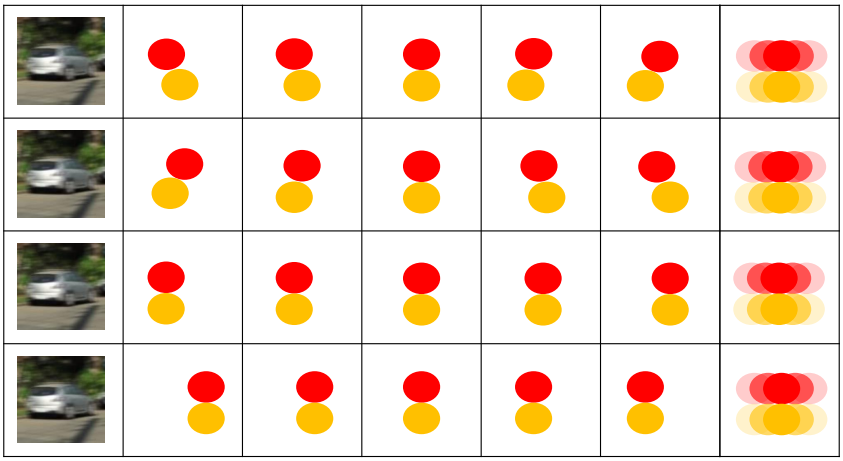

Figure 1. The process of generating blurry images. The second to sixth columns demonstrate modeling the sharp video sequence under four different conditions, but they can generate identical motion-blurred images, as shown in the seventh column. The blurry images in the first column are the portrayal of the seventh column in reality.

2. Related Work

Our proposed method is related to image deblurring and video deblurring. In this section, we will briefly review that content and common approaches.

2.1. Image Deblurring

Image deblurring is an ill-posed computer vision problem. Traditional methods usually optimize complex objective functions based on adding constraints to the blur kernel. Kim et al. [14] proposed a segmentation-based method to jointly estimate blur kernel and sharp images. However, the forward motions and depth variations are not considered in their methods. In [7], a segmentation-free approach was put forward using a locally linear optical flow field to model the blur kernel. However, in the real world, the motions are complex, and thus this assumption does not hold. Pan et al. [14] presented a soft segmentation approach to deal with the severe blur, but the initialization of their method is based on user inputs.

Recently, deep learning methods have won tremendous success in the realms of computer vision, such as object detection [15], image classification [16] and facial analysis [17]. For the low-level vision issues, many deep learning based models are also used, for example, image denoising [18], super-resolution technology [19], image dehazing [20] and image deblurring [8]. There are also many algorithms that try to estimate the blur kernel in a non-uniform way or use a non-blind deblurred method to generate sharp frames based on deep learning models. Sun et al. [12] calculated the motion blur of each block based on a proposed NN-based model, and then obtained dense motion based on a Markovian random field. Gong et al. [21] proposed a much deeper CNN model for the estimation of motion flow. Both of them [12,21] are not trained in the way end-to-end, and their deblurring processing is time-consuming. Beyond that, many end-to-end approaches are also being studied to solve the image deblurring problem [8,22]. For the sake of using a large receptive field, many have proposed multi-scale models. Nah et al. [8] proposed complicated three-scale CNN models, and each of them contained 40 convolutional layers. In addition, Zhang et al. [23] presented a theory of relativistic blur loss which combined the learning-to-Blur GAN and learning-to-DeBlur GAN.

2.2. Video Deblurring

The goal of video deblurring is generate a sharp video from a blurry video. Some approaches aim to combine multiple images directly in the spatial domain. In [24,25], the algorithm selects the best pixels obtained from aligning multiple low-quality frames in one sequence to reconstruct the final results. Cho et al. [26] utilized patch-based methods to improve the robustness for moving objects. However, the method cannot solve the large depth variations, and the procedure of patch matching is very time-consuming. Klose et al. [27] fused pixels via using 3D reconstruction to project them into the reference coordinate system. However, this method is fragile for highly dynamic videos. The aggregation methods rely on the assumption that there are some sharp frames in general. Thus, sharp pixels can be generated based on nearby frames. Delbracio et al. [28] demonstrated that aggregating multiple images is beneficial to computationally highly efficient video deblurring. In [29], adjacent frames were distorted according to optical flow. However, it did not achieve satisfactory performance under the conditions of occlusions because of the computational limit of optical flow. Recently, [30] built a correlation volume pyramid among all the pixel-pairs between neighboring frames to construct distant pixel correspondences for fast motions.

Usually, all above approaches need the internal information contained in the whole video to restore the sharp sequence frames, but what we have done is recover a sequence of sharp frames form just one single blurred image. The most related to our task is [13]. They proposed a deep learning network that reestablishes the time sequence via extracting frames and generates a clear video sequence. However, this method is computationally extensive because it generates seven continuing sharp frames via seven sub-models. Meanwhile, the loss functions utilized in their methods are based on aligning data while ignoring the characteristics of the motion-blur, whereas the recovery can be regraded as an non-aligning process. These next sections firstly demonstrate the process of generating blurry images; next we introduce our proposed framework and contiguous blurry loss functions.

3. Approach

In this section, we first analyze how the blurry images are generated based on imaging principles, and thus introduce the contiguous blurry loss, and then explain the composition and proportion of the whole loss function. Finally, we proposed the architecture of our learning network.

3.1. The Generation of Blurry Images

In most previous studies, blurry images were generated as (1) by convolving the modeled blur kernel on one sharp image. However, the fountainhead of the blurry image generated is the synthesis of information over time. Specifically, because a camera sensor receives light all the time during exposure, sharp images stimulated at each moment are superimposed, which results in blurred images. Then, the camera response function transforms the integrated signal into a pixel value. Thus, the process of generating blurry image can be approximated as an integration of signals from high-speed video frames rather than a manually defined kernel.

The accumulation of motion blur can be approximately modeled as follows:

$$B = g(\frac{1}{T}\int_0^{T-1} S(t)\mathrm{d}t) \simeq g(\frac{1}{M}\sum_{i=0}^{M-1} S[i]) \tag{2}$$

Here, T is time of exposure and $S(t)$ denotes the sensor signal of one sharp image at time t. $S[i]$ and M are the i-th captured sharp frame and the total number of frames, respectively. g denotes the operator which can transfer the sharp signal $S(t)$ into an blurry frame through the camera response function. Obviously, our target is to reestablish and acquire sharp frames $S[i]$ from the input blurry image B, which is a totally inverse process.

3.2. Loss Functions

Three types of loss function are mainly considered in our GAN-based model: adversarial loss, content loss and contiguous loss.

Adversarial loss. To prompt G to generate sharp frames as close as possible to the actual sharp images, the adversarial loss should be introduced to perfect models iteratively. Correspondingly, in the training phase, Resnet-based model parameters will be updated repeatedly, trying to muddle through in discriminator D. The following is the function expression of adversarial loss:

$$\mathcal{L}_{adversarial} = \log(1 - D(G(I^{blurry}))), \tag{3}$$

where $D(G(I^{blurry})$ represents the probability that the restored frame is a sharp enough image.

Content loss. Many previous studies on video deblurring will have considered mean square error (MSE) loss in solving objective optimization. On the grounds of the measurement index MSE, we define the content loss function as follows:

$$\mathcal{L}_{content} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}(I_{x,y}^{sharp} - G(I^{blur})_{x,y})^2, \tag{4}$$

where H and W denote the height and weight of the frame. $I_{x,y}^{sharp}$ and $G(I^{blur})_{x,y}$ correspond to the values of real sharp frame and deblurred frame generated from the ResNet-based model, respectively, at locations (x, y).

Contiguous loss. As Figure 1 shows, a single motion-blurred image does not have one-to-one correspondence with a sharp video sequence. Extracting a video sequence from a blurry image is a non-aligned process, inspired by the work in [31], so an improved contiguous blurry loss is introduced in this work.

In order to measure the similarity between images without aligning them, a novel measurement is defined to settle this issue. The core idea of this measurement is to map images into points (features) in higher dimensional space. Additionally, if two images are

similar in the corresponding point sets in higher dimensions, they are considered to be semblable. When a large proportion of features of an image have similar features to another one, we can assume that the images are similar. On the contrary, when the features of one image nearly have no resemblance to those of another image, we consider the images to be different from each other. Following this observation, the measurement of contiguous similarity between images can be formulated.

Given a pair of images for analysis, the original image x and its corresponding target image y, they can be represented as sets of points, respectively: $X = \{x_i\}$ and $Y = \{y_j\}$. Set d_{ij} as the cosine distances between x_i and y_j. As X and Y are the most matching features, the value of d_{ij} must be smaller than the cosine distance of points from other features $Y^{'}$, and we assume those distances as d_{ik}. In the work of reference [31], they defaulted to $d_{ij} << d_{ik}$, but in practice, their values may be very close. For example, there are many worker bees on the hive in one image, and the features of worker bees may be very similar. If we apply the learned features to deblur directly, the network may confuse two worker bees who are close to each other. Hence, we add two restrictions to avoid the occurrence of error recognition and to ensure that $d_{ij} << d_{ik}, \forall k \neq j$.

Hence, the normalizing distance between the similar features x_i and y_j can be defined as the following formula:

$$\tilde{d}_{ij} = \frac{d_{ij}}{d_{ik} + \epsilon} \tag{5}$$

for the preset constant $\epsilon = 1 \times 10^{-5}$. Then we transform distance measurements into similarities of features with the exponentiation operator:

$$w_{ij} = \exp\left(\frac{1 - \tilde{d}_{ij}}{h}\right) \tag{6}$$

where $h > 0$ denotes the parameter of bandwidth. Finally, CX_{ij} is introduced to represent the contiguous similarity between features, which can be expressed in scaled invariant versions of the standardized similarities as follows:

$$CX_{ij} = \frac{w_{ij}}{\sum\limits_{k} w_{ik}} \tag{7}$$

To compute the similarity index between images is to seek out the most similar feature x_i for each feature y_j, and then sum the similarity index over all y_j. Specifically, feature x_i is considered to be similar to feature y_j contextually if no other feature is found in Y that is closer to it. In another case, if x_i is not near enough to any particular y_j, no matter how far away x_i is, its contiguous similarity to any y_j is supposed to be comparatively low. Therefore, this method has strong robustness to the shift of features caused by motion blur. The similarity between features, CX_{ij}, can be incorporated and integrated into the global image context. Formally, define the number of features $|Y| = |X| = N$; then the contiguous similarity between images can be deduced as follows:

$$CX(x, y) = CX(X, Y) = \frac{1}{N} \sum_{j} \max_{i} CX_{ij} \tag{8}$$

As mentioned above, due to the misalignment in the training data, the generated blurred images mainly contain two losses, $\mathcal{L}_{content}$ and $\mathcal{L}_{adversarial}$. On the other hand, it is unsatisfactory to consider only the CX loss for unaligned data, as it is only associated with mapping features but not their spatial locations in the image. For training our model more appropriately, we integrate the spatial pixel coordinates and pixel-level RGB information into the image features. In conclusion, the proposed contiguous blurry loss (CBL) can be defined as:

$$CX_f(X,Y) = \frac{1}{N}\sum_j \max_i CX'_{ij}(CX_{ij}+1) \tag{9}$$

where $CX'_{ij} = ||(x_i,y_i)-(x_j,y_j)||_2$. (x_i,y_i) and (x_j,y_j) represent different spatial coordinates of features.

In order to be consistent with the form of other losses, the function of contiguous blurry loss can be converted to logarithmic form:

$$\mathcal{L}_{contiguous} = \mathcal{L}_{CX_f}(x,y) = -\log(CX'(x,y)(CX(x,y)+1)) \tag{10}$$

Weight Balance between Loss Functions.As G and D network are jointly trained in an alternative way, the different loss functions should be merged in the form of weight fusion. Integrating losses in our GAN-based model, the ultimate loss function can be defined by following:

$$\mathcal{L} = \mathcal{L}_{content} + \alpha \cdot \mathcal{L}_{adversarial} + \beta \cdot \mathcal{L}_{contiguous}\,. \tag{11}$$

α and β are hyper-parameters introduced in the whole loss $\mathcal{L}$ for balancing the content, adversarial and contiguous blurry losses. When $\alpha = 0$ and $\beta = 0$, $\mathcal{L}$ simplifies to the content loss. In this specific case, the GAN-based model will degrade to a ResNet-based model. With the increasing of the two parameters, the two corresponding loss functions play more important roles during the training stage. Usually, α should take a relatively small value, because the experimental results indicate that the performance of proposed network will be degraded when α takes a larger value. In the next section, we will do experiments to choose the befitting values of them.

3.3. The Model Architecture

GAN is a popular framework in the area of image enhancing tasks, such as super-resolution [32], image deblurring and image denoising. To realize the inverse conversion of blurry images, we developed a deblur network, a GAN-based model, on the basis of the ResNet-based model. Hence, we first introduce our ResNet-based model, and then draw forth the proposed GAN-based model. Both the ResNet-based and GAN-based models are end-to-end systems.

3.3.1. ResNet-Based Model

In two-dimensional CNN operations, convolution is usually implemented on 2D patterns or feature maps only for feature learning in spatial dimensions. As for our deep residual networks, we also take this form to perform 2D convolutions in the convolution stages to learn feature representations for extracting a video sequence through one blurred image. The operation process of 2D convolution is to convolve 2D kernels/filters on the blurry images. By doing so, the dynamic variations can be captured easily by mapping the features in the convolution layers, which is conducive to the modeling of blur evolution and further restoration of sharp frames.

Typically, 2D convolution operations can be formally expressed as

$$V_{ij}^{xy} = \sigma\Big(\sum_m \sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} V_{(i-1)m}^{(x+p)(y+q)} \cdot g_{ijm}^{pq} + b_{ij}\Big)\,, \tag{12}$$

in which V_{ij}^{xy} represents the convolution value of j-th feature mapping at point (x,y) on the i-th layer; (P_i, Q_i) means the 2D convolution kernel size. g_{ijm}^{pq} is the inter-layer correlation coefficient, which denotes from the $(i-1)$-th layer, the m-th feature mapping connection to the kernel (p,q)-th value. We selected the non-linearity activation function ReLU as $\sigma(\cdot)$, whose performance is superior to other activation functions in various computer vision tasks, e.g., tanh and sigmoid.

In this work, a ResNet-based model is constructed to define and implement 2D convolution. This model consists of several residual blocks [16], wherein each contains

two-layer convolutions, in addition to five other convolution layers. The design of the architecture was derived from fully convolutional neural network (FCNN), which was originally applied to semantic segmentation. However, in our ResNet-based model, unlike FCNN, the spatial size of feature mapping remains constant. That is, the model has neither up-sampling processes nor down-sampling processes. The specific configuration of our model is in Table 1.

Table 1. The configuration of the ResNet-based model. It consists of 2 convolution layers (L1, L2), 13 residual blocks, another 2 convolution layers (L27, L28) with no jumper connections and 3 extra (L29, L30 and L31) convolution layers. The residual blocks, represented by L(X) and L(X + 1), each contain 2 convolution layers, where "X" can take 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23 and 25 of these residual blocks.

Layers	Kernel Size	Output Channels	Operations	Skip Connection
L1	$3 \times 3 \times 1$	16	ReLU	-
L2	$3 \times 3 \times 1$	64	ReLU	L4, L28
L3	$3 \times 3 \times 1$	64	ReLU	-
L4	$3 \times 3 \times 1$	64	-	L6
L5	$3 \times 3 \times 1$	64	ReLU	-
L6	$3 \times 3 \times 1$	64	-	L8
L7	$3 \times 3 \times 1$	64	ReLU	-
L8	$3 \times 3 \times 1$	64	-	L10
L9	$3 \times 3 \times 1$	64	ReLU	-
L10	$3 \times 3 \times 1$	64	-	L12
L11	$3 \times 3 \times 1$	64	ReLU	-
L12	$3 \times 3 \times 1$	64	-	L14
L13	$3 \times 3 \times 1$	64	ReLU	-
L14	$3 \times 3 \times 1$	64	-	L16
L15	$3 \times 3 \times 1$	64	ReLU	-
L16	$3 \times 3 \times 1$	64	-	L18
L17	$3 \times 3 \times 1$	64	ReLU	-
L18	$3 \times 3 \times 1$	64		L20
L19	$3 \times 3 \times 1$	64	ReLU	-
L20	$3 \times 3 \times 1$	64	-	L22
L21	$3 \times 3 \times 1$	64	ReLU	-
L22	$3 \times 3 \times 1$	64	-	L24
L23	$3 \times 3 \times 1$	64	ReLU	-
L24	$3 \times 3 \times 1$	64	-	L26
L25	$3 \times 3 \times 1$	64	ReLU	-
L26	$3 \times 3 \times 1$	64	-	L28
L27	$3 \times 3 \times 1$	64	ReLU	-
L28	$3 \times 3 \times 1$	64	-	-
L29	$3 \times 3 \times 1$	256	ReLU	-
L20	$3 \times 3 \times 1$	256	ReLU	-
L31	$3 \times 3 \times 1$	21	-	-

As shown in the Figure 2, the input to the ResNet-based model is a single motion-blurred image generated from the previous operation. However, we should be aware that we do not directly deblur in the original RGB space, but perform the actual deblurring process in the grayscale space. In detail, RGB images are first converted into YCbCr space, and the reason for choosing channel Y is that the illumination is the most salient feature of an image. In reference to the architecture, the first and second layers begin with the 2D convolution operation using the kernel size of $3 \times 3 \times 1$. For further elaboration, in the first layer, 2D kernels are used to perform convolution operations on three sets of consecutive frames to form a group of feature mappings. Then in the second layer, the feature mappings are convoluted by 2D filters to get higher-level feature mappings. In the later residual and convolution layers, the size of convolution kernels is still set to $3 \times 3 \times 1$ to keep the temporal dimension in a computationally efficient dimension. The output of ResNet-based

model is seven consecutive frames after deblurring. Finally, through the original Cb and Cr channels, the output is colorful images converted from the gray-scale images.

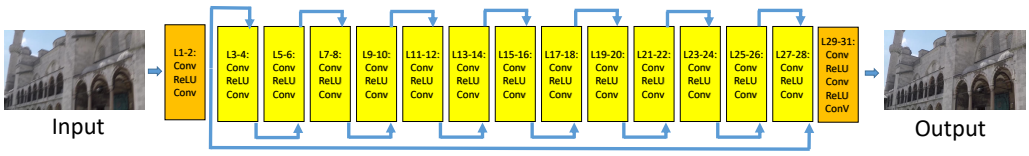

Figure 2. The ResNet-based framework for extracting a sharp video sequence. The input is a single motion-blurred image. The output is seven sharp continuing frames. It consists of 15 blocks. Each block has two/three convolutional layers and one/two ReLU activation functions. The details of the architecture can be seen in Table 1.

3.3.2. GAN-Based Model

The GAN is a model training generative method proposed by [33]. The GAN network comprises a pair of adversarial networks, one generator (G) and one discriminator (D). The G is trained to synthesize very similar samples that can muddle up D, whereas D aims to distinguish the synthesized samples and the real samples. They improve their networks by confronting each other. Inspired by the strategy of adversarial training, a GAN-based model is proposed in this work, in which G is used for deblurring images, and D to distinguish the deblurry images from the subsistent sharp images. After alternate training, if the output images of the generator are sharp enough, the deblurred image can trick the discriminator.

By performing the deblurring procedure through the generative adversarial framework, introducing the min-max optimization problem shown below is inevitable, which is similar to the formulation in [33]:

$$\min_G \max_D V(G,D) = \mathrm{E}_{h\sim p_{train(h)}}[\log(D(h))] + \mathrm{E}_{\hat{h}\sim p_{G(\hat{h})}}[\log(1-D(G(\hat{h})))] \tag{13}$$

where h denotes a sharp frame image taken from real life, and $\hat{h}$ indicates a blurred image. When training G and D models alternately, G aims to deceive D to misclassify synthetic frames, whereas D is aimed at discriminating deblurry frames from subsistent sharp images. The ultimate aim of model training is to perfect the G network, which can recover sharp frames from the input blurry images.

As illustrated in the flow diagram Figure 3, the front part is the G model. Its framework is shown in Figure 2 and its configuration is shown in Table 1. The latter part of Figure 3 is our D network, a CNN-based model, which was built according to the guidance for CNN architecture construction presented by Radford et al. [34], and it resembles the VGG network presented in [35]. Our D model consists of 16 convolution layers and one top layer with a bidirectional soft-max classifier. From the bottom up, the number of channels in convolution kernels increases from 64 up to 512; the top layer has 4096 channels. The entire network is trained to recognize deblurred frames among all the sharp images. For more information about the network configuration, please refer to Table 2.

Table 2. Configurations of our D model in the GAN-based model. ReLU denotes the activation function and BN represents batch normalization.

Layers	1–2	3–6	7–11	12–16	17–18	19
kernel	3×3	3×3	3×3	3×3	FC	FC
channels	64	128	256	512	4096	2
BN	BN	BN	BN	BN	-	-
ReLU	ReLU	ReLU	ReLU	ReLU	-	-

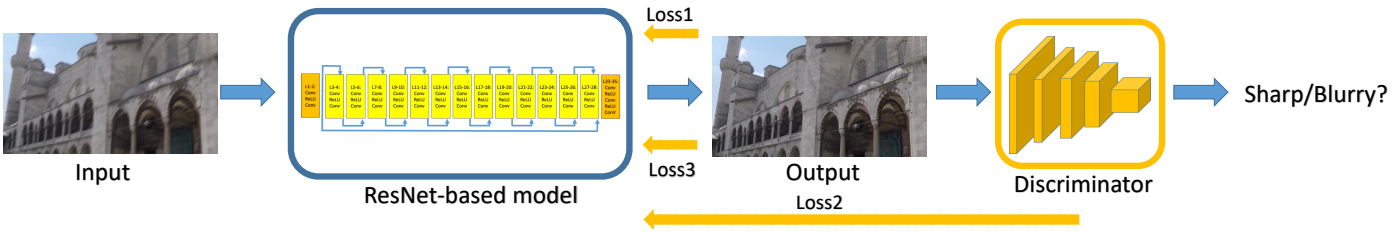

Figure 3. The GAN-based framework for extracting a sharp video sequence. The whole network architecture is composed of one discriminator and one generator. The generator is the ResNet-based model, whose architecture can refer to Figure 2. The discriminator is a VGG-like CNN, which consists of 16 convolutional layers and a two-class software layer. During the training stage, the pixel2pixel loss (loss1), adversatial loss (loss2) and contiguous blurry loss (loss3) work together to update the parameters of the ResNet-based mode.

4. Results

In this part, several experiments on the task of recovering sa harp frame sequence from a single motion-blurred image are shown to demonstrate how effective the proposed GAN-based model is and how close our contiguous blurry loss is to the real loss.

4.1. Datasets

Video Blurred Database. Su et al. [36] established a benchmark database which captures at 240 fps with multifarious equipment, such as the Canon 7D, GoPro Hero 4 Black, iPhone 6s and so on. It collected from 71 videos, including 6708 synthetic blurry frames with corresponding ground truth, and each video contains 100 frames of resolution size 1280 $\times$ 720. We took 61 of them as training videos and the remaining 10 as testing videos; the partitioning was consistent with previous studies by [36].

GoPro Database. In addition, we tested and evaluated the effectiveness of presented model on the GoPro Database, which was built to study image deblurring specifically by Nah et al. [8]. It is split into a training dataset and a test dataset, which contain 21 sharp videos and 11 different sequences, respectively.

We contrast the performance of our proposed model with those of state-of-the-art approaches in both qualitative and quantitative aspects.

4.2. Implementation Details and Parameter Settings

The video recording speed of our training databases was 240 frames per second, and the general standard real-time video rate is 30 frames per second, so it was more appropriate to split the blurred image into eight frames. However, the middle frame often corresponds to the centroid of local blur, and splitting into odd frames is more conducive to the rapid convergence of the model, so the blurred images were split into seven frames in our experiment. In the ResNet-based model training, we initialized the weights first by a Gaussian distribution with a zero mean and 0.01 standard deviation. Then, parameters were updated with the mini-batch size of four in each iteration. In order to augment the amount of sub-images for feature learning, we cut out a 128 $\times$ 128 patch spread all over an image (1280 $\times$ 720), and flipped the frame randomly during the training phase in the meanwhile. In this way, 712,193 patches can be generated per frame in the database [36] in the ResNet-based model training, such that no features will be omitted. Only content loss was used—the learning rate being 10^{-4}. After about 1.5×10^5 iterations, the training loss was no longer reduced; then we decreased the learning rate to 10^{-5} in pursuit of extra performance improvements. When training the GAN-based model, the optimal algorithm was adopted to test the distribution of coefficients among different losses, and the result indicate that setting the hyper-parameters $\alpha = 0.001$ and $\beta = 0.1$ can provide peak performance.

4.3. The Effectiveness of the GAN-Based Model

The proposed GAN-based model is adept at learning the representations of spatio-temporal features. For verifying the superiority of the proposed GAN-based model, we assessed our model against other state-of-the-art models on the Video Blurred and GoPro Database.

Table 3 records the PSNR values of different approaches using the Video Blurred Database, and Table 4 shows the PSNR and SSIM values of different models on the GoPro Database. It is evident from the two tables that compared to other models, ours (GAN-based model) can achieve approximately 1–5% improvements of PSNR and SSIM values. This just goes to show that the framework is more adept at learning spatio-temporal features. By conducting these experimental comparisons, the effectiveness of GAN-based model was verified.

Table 3. Performance contrast, in terms of PSNR values, to PSDEBLUR, WFA [28] and DBN [36] on the 10 testing videos of Video Blurred Database. The optimum result of each video is displayed in bold, and the sub-optimal result is marked with an underline. All results of GAN-based models were obtained without aligning.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
INPUT	24.14	30.52	28.38	27.31	22.60	29.31	27.74	23.86	30.59	26.98
Methods										
PSDEBLUR	24.42	28.77	25.15	27.77	22.02	25.74	26.11	19.71	26.48	24.62
WFA [28]	25.89	32.33	28.97	28.36	23.99	31.09	28.58	24.78	31.30	28.20
DBN [36]	25.75	31.15	29.30	28.38	23.63	30.70	29.23	25.62	31.92	28.06
Our method	27.73	32.56	31.38	30.54	24.59	31.11	30.39	26.16	33.32	29.89
Our method (with CBL)	**28.29**	**33.46**	**32.68**	**31.32**	**25.37**	**32.33**	**31.39**	**27.23**	**34.56**	**30.74**

Table 4. Performance comparison on the *GOPRO* Database and an ablation study of our model after different stages of training.

Method	**PSNR**	**SSIM**
Jin et al. [13]	26.98	0.8936
Nah et al. [8]	28.98	0.9135
Our method	29.12	0.9236
Our method (CBL)	**29.62**	**0.9294**

4.4. The Effectiveness of Contiguous Blurry Loss

In this subsection, the capacity of the proposed model when adding contiguous blurry loss (CBL) is investigated to see whether it is better than not adding CBL.

Quantitative results on the experimental datasets are presented in Tables 3 and 4. Obviously, our model containing CBL outperformed the model without CBL in quantitative terms, though slightly (about 1% improvement).

The GAN-based generator model was designed to generate frames whose pixel values are similar, whereas D, with the adversarial loss, together with the discriminator drive G, is used to restore photo-realistic frames. The purpose of introducing contiguous blurry loss is to alleviate the problem of training data being unaligned. The GAN-based model and the three losses complement each other and achieve better results.

Figures 4–6 display three sets of exemplar results on the GoPro Database. Images in the first column on the left are the input motion-blurred images: the first of the three images is the original blurred image, and the next two images show details of the first picture framed by color boxes. The following columns on the right side of the vertical line are the shape frames generated by Jin et al. [13] and ours. Please note that the upper row has the results of [13], and bottom row corresponds to the proposed model. It can be clearly seen in the details of Figure 4 that the edge contours of the car and pillar are clearer in the sharp video sequence generated by our method. In Figure 5, the sharp video

sequence generated by [13] has a pronounced sense of graininess. From the comparison of Figure 6, it can be seen that the details of the window and the edge of the yellow box are sharpened more in our sharp video sequence than that of [13]. In general, through the qualitative comparison in Figures 4–6, the results of our model's deblurring are obviously more photo-realistic than those of [13].

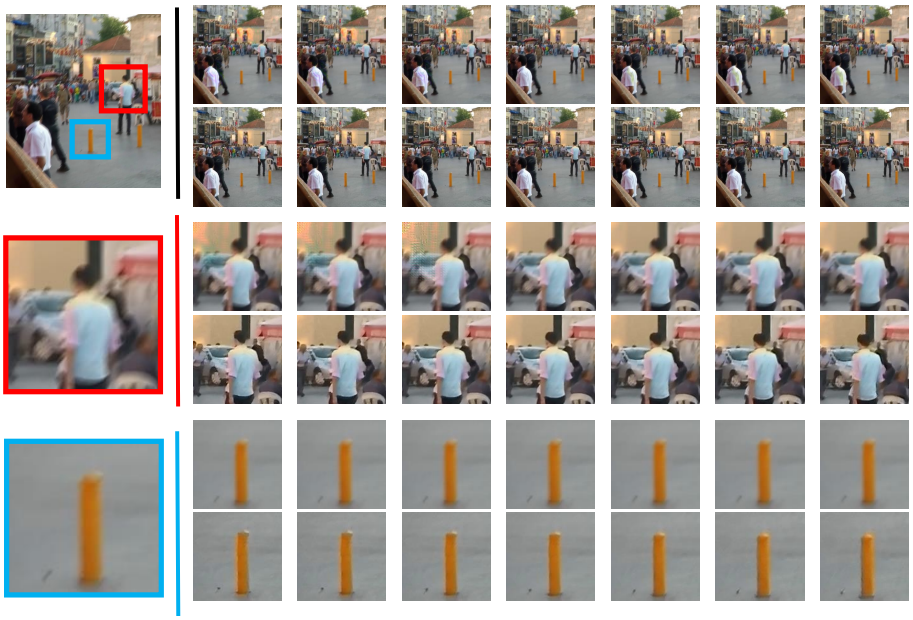

Figure 4. Photo of shopping street for qualitative comparison. The input images are shown in the first column. The following seven columns show the results of the method by Jin et al. [13] (**top row**) and ours (**second row**).

Figure 5. Flowers photo for qualitative comparison. The input images are shown in the first column. The following seven columns show the results of the method by Jin et al. [13] (**top row**) and ours (**second row**).

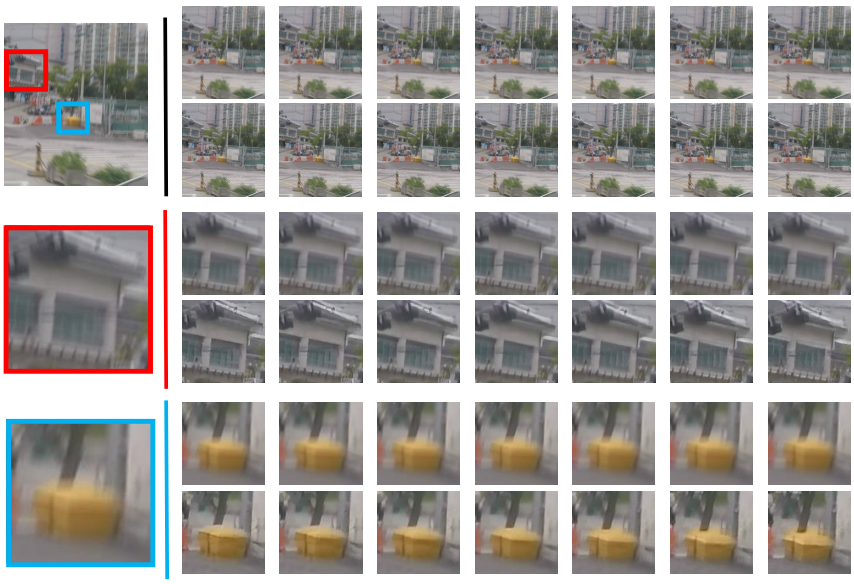

Figure 6. Photo of street view for qualitative comparison. The input images are shown in the first column. The 2nd/8th columns show the results of the method by Jin et al. [13] (**top**) and ours (**down**).

4.5. A Comparison with Other Approaches

In order to inspect the performance of the proposed model in depth, we conducted a series of experiments to contrast the effectiveness of the proposed model with other state-of-the-art methods using Video Blurred Database and GoPro Database.

In the experiment using Video Blurred Database, methods PSDEBLUR (deblurred results using PHOTOSHOP software), WFA (multiple frames as input) [28] and DBN [36] were compared with our proposed model.

In [36], they compared DBN with other state-of-the-art methods, and achieved optimum effectiveness with Video Blurred Database, so we compare the PSNR values of the proposed model with those of DBN and other state-of-the-art approaches on the 10 testing videos of Video Blurred Database. Table 3 indicates that the results of PSNR values for our proposed model are superior to those of the other listed models on the Video Blurred Database.

In the experiment on GoPro Database, the proposed model was compared with the state-of-the-art models designed in [8,13]. The method of Jin et al. is also a model based on CNN, and it achieved the most advanced results in [13], but it utilizes seven sub-models to extract the sharp video sequences from a single motion-blurred image, which means relatively high computational complexity. Table 4 indicates that the proposed model trained on GoPro Database can achieve preferable results to [13]. In addition, experimental results demonstrate that our model containing CBL is superior to other methods both on Video Blurred Database and GoPro Database.

4.6. Different Frames

4.6.1. Optimum Parameters

It is necessary to study how do hyper-parameters α and β affect the effectiveness of the proposed model. According to the empirical values of hyper-parameters in the GAN model's loss function, the range of α decreases in magnitude from 0 to 0.00001, whereas the value range of β is increased by 0.05 arithmetically from 0 to 0.25. The values of α and β were varied to compare the corresponding PSNR values on GoPro Database. Our model can achieve optimum performance when α and β are set to 0.001 and 0.1, respectively. Figure 7 shows the performance comparisons of our method in terms of PSNR by varying the α when β was fixed in 0.1. Similarly, Figure 8 shows the performance comparisons of our method in terms of PSNR by varying the β when α was fixed in 0.001.

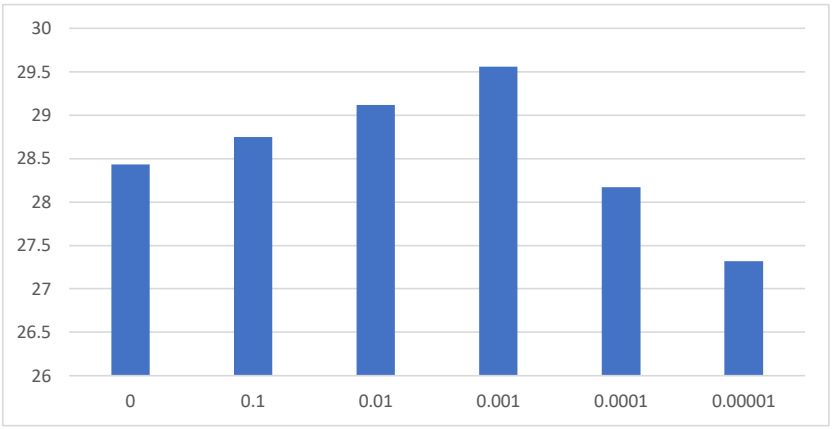

Figure 7. Performance comparisons of our method in terms of PSNR by varying the α on the first set of Video Blurred Database.

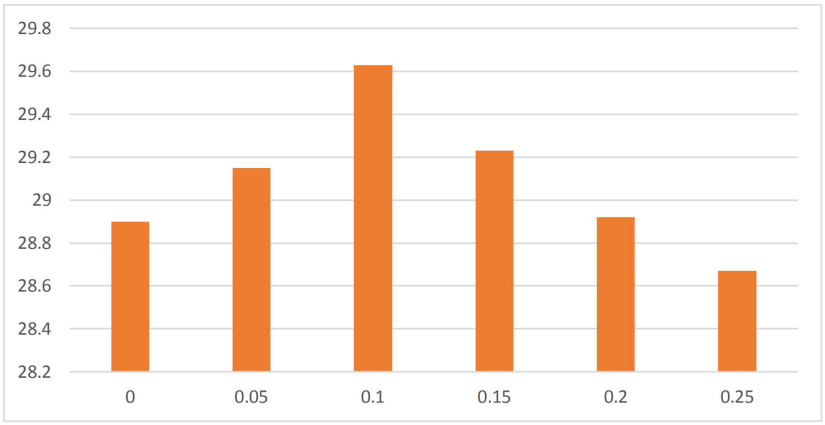

Figure 8. Performance comparisons of our method in terms of PSNR by varying the β on the first set of Video Blurred Database.

4.6.2. Motion Interpolation

Furthermore, the proposed model can recover number of frames by applying the G model to output frames. Every two adjacent frames of the seven output frames can form six groups, and each group can be averaged to produce a new blurry image, which can be fed into our generator to generate seven new frames again. For instance, we can recover $4 \times 7 = 28$ sharp frames, as shown in Figure 9. Back and forth, a blurred image can get the explosive number of frames, which means the model has the potential to interpolate subsequent frames with high accuracy. By exploiting the information embedded in motion blur [37], our model can be employed to disassemble a motion-blurred image into multiple frames, which can break through the limitations of the device to get an intelligent high-frame-rate video in the future. It also can be used in many applications, such as video editing and temporal super-resolution of videos.

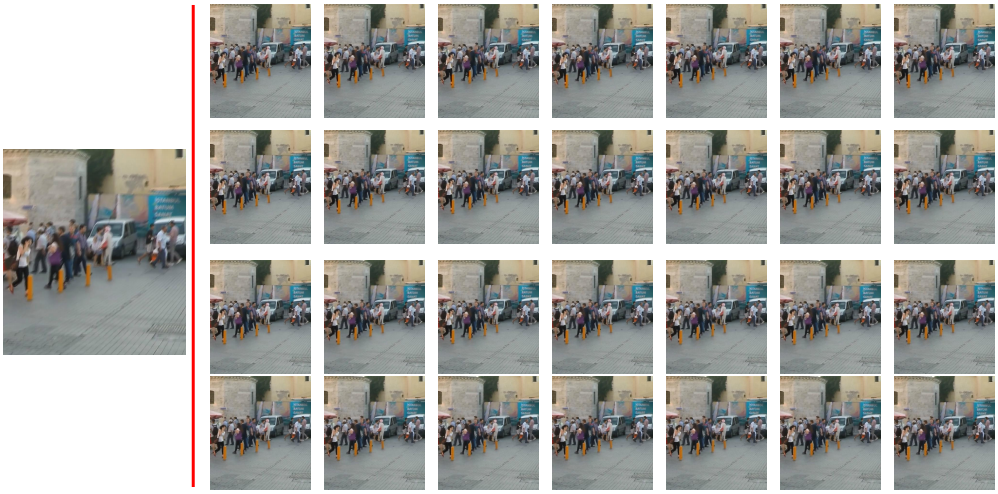

Figure 9. An example of the interpolation of subtle motion: 28 frames were extracted by the proposed method based on the input image.

5. Discussion

Blur is the result of the motion of multiple objects, camera shaking or scene depth variations. As a reversal process, restoring deblurred images from blurred images is a typical ill-posed computer vision problem. As the blurred image does not correspond to its sharp video sequence one-to-one, neither the traditional pixel2pixel nor perceptual loss is suitable for processing non-aligned data. Therefore, it is not ideal to simply superimpose the blur as a 3D kernel, or to train the neural network on a specific frame (which does not have the same uniqueness).

To solve this problem, this paper considers combining three loss training models to learn and integrate to generate sharp image sequences. The contiguous blurry loss ignores the spatial positions of the features and focuses on measuring the similarity of sharp and blurry images on the basis of their similar features. The adversarial loss and content loss complement the loss caused by spatial positions at the pixel2pixel level. The adversarial loss can enhance the accuracy of the output of each network. The content loss refers to the pixel2pixel loss of each layer of feature map, which can learn the manifold space where the image is located.

Due to the feature set learning for the non aligned data, we can obviously see from the experimental results that the proposed model can segment the edges of some objects more sharply, and the generated deblurred images have no sense of particles compared with other methods, and the features and details of objects can be better restored. The model can be used not only in the task of deblurring, but also in the task of rain removal through the relearning of a sharp image sequence [38]. In addition, the artificial high-frame-rate video can be obtained by the application of high-precision interpolation.

6. Conclusions

In this paper, the method of adversarial training was modified to extract the sharp frames of a video sequence from one motion-blurred image. Specifically, a novel model based on GAN was proposed, which can generate multiple sharp frames on the basis of one CNN model. In addition, in order to alleviate the problem that one blurred image does not corresponding one-to-one to a sharp video sequence, as different video sequences can create almost identical blurry images, a new contiguous blurring loss method was proposed, which mainly measures the loss of unaligned data. Experimental results demonstrated that the combination of the proposed network and the contiguous blurry loss can generate sharp video sequences and improve the shortcomings of existing methods.

Author Contributions: Conceptualization, W.N. and K.X.; methodology, W.N.; software, W.N.; validation, K.X. and Y.P.; formal analysis, W.N. and Y.P.; resources, K.X.; writing—original draft preparation, W.N.; writing—review and editing, W.N., Y.P. and K.X.; visualization, W.N.; supervision, K.X.; project administration, K.X.; funding acquisition, K.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (number U1813222), the Tianjin Natural Science Foundation (number 18JCYBJC16500) and the Key Research and Development Project from Hebei Province (number 19210404D).

Institutional Review Board Statement: This study was approved by the academic and Ethics Committee of School of electronic and information engineering, Hebei University of technology.

Informed Consent Statement: All the authors were informed consent without objection.

Data Availability Statement: Video Blurred Database: http://www.cs.ubc.ca/labs/imager/tr/2017/DeepVideoDeblurring/, accessed on 10 August 2019; GoPro Database: https://github.com/SeungjunNah/DeepDeblurrelease, accessed on 23 November 2019.

Conflicts of Interest: We declare no conflict of interest.

References

1. Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Liu, W.; Li, H. Adversarial spatio-temporal learning for video deblurring. *IEEE Trans. Image Process.* **2018**, *28*, 291–301. [CrossRef]
2. Xu, L.; Lu, C.; Xu, Y.; Jia, J. Image smoothing via L 0 gradient minimization. In Proceedings of the 2011 SIGGRAPH Asia Conference, Hong Kong, China, 13–15 December 2011.
3. Xu, L.; Zheng, S.; Jia, J. Unnatural l0 sparse representation for natural image deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
4. Zhang, K.; Luo, W.; Stenger, B.; Ren, W.; Ma, L.; Li, H. Every Moment Matters: Detail-Aware Networks to Bring a Blurry Image Alive. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle WA USA, 12–16 October 2020; pp. 384–392.
5. Whyte, O.; Sivic, J.; Zisserman, A.; Ponce, J. Non-uniform deblurring for shaken images. *Int. J. Comput. Vis. (IJCV)* **2012**, *98*, 168–186. [CrossRef]
6. Gupta, A.; Joshi, N.; Zitnick, C.L.; Cohen, M.; Curless, B. Single image deblurring using motion density functions. In Proceedings of the European Conference on Computer Vision (ECCV), Crete, Greece, 5–11 September 2010.
7. Hyun Kim, T.; Mu Lee, K. Segmentation-free dynamic scene deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
8. Nah, S.; Kim, T.H.; Lee, K.M. Deep multi-scale convolutional neural network for dynamic scene deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
9. Chakrabarti, A. A neural approach to blind motion deblurring. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016.
10. Schuler, C.; Hirsch, M.; Harmeling, S.; Schölkopf, B. Learning to deblur. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2015**, *38*, 1439–1451. [CrossRef] [PubMed]
11. Xu, L.; Ren, J.S.; Liu, C.; Jia, J. Deep convolutional neural network for image deconvolution. *Adv. Neural Inf. Process. Syst. (NIPS)* **2014**, *27*, 1790–1798.
12. Sun, J.; Cao, W.; Xu, Z.; Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
13. Jin, M.; Meishvili, G.; Favaro, P. Learning to Extract a Video Sequence from a Single Motion-Blurred Image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
14. Hyun Kim, T.; Ahn, B.; Mu Lee, K. Dynamic scene deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
17. Zhang, K.; Huang, Y.; Du, Y.; Wang, L. Facial expression recognition based on deep evolutional spatial-temporal networks. *IEEE Trans. Image Process. (TIP)* **2017**, *26*, 4193–4203. [CrossRef] [PubMed]
18. Burger, H.C.; Schuler, C.J.; Harmeling, S. Image denoising: Can plain neural networks compete with BM3D? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
19. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014.

20. Ren, W.; Liu, S.; Zhang, H.; Pan, J.; Cao, X.; Yang, M.H. Single image dehazing via multi-scale convolutional neural networks. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
21. Gong, D.; Yang, J.; Liu, L.; Zhang, Y.; Reid, I.; Shen, C.; Van Den Hengel, A.; Shi, Q. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
22. Hradiš, M.; Kotera, J.; Zemcık, P.; Šroubek, F. Convolutional neural networks for direct text deblurring. In Proceedings of the British Machine Vision Conference (BMVC), Swansea, UK, 7–10 September 2015.
23. Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; Li, H. Deblurring by realistic blurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–18 June 2020; pp. 2737–2746.
24. Law, N.M.; Mackay, C.D.; Baldwin, J.E. Lucky imaging: high angular resolution imaging in the visible from the ground. *Astron. Astrophys.* **2006**, *446*, 739–745. [CrossRef]
25. Joshi, N.; Cohen, M. Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal. In Proceedings of the IEEE International Conference on Computational Photography (ICCP), Cambridge, MA, USA, 29–30 March 2010.
26. Cho, S.; Wang, J.; Lee, S. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–9. [CrossRef]
27. Klose, F.; Wang, O.; Bazin, J.C.; Magnor, M.; Sorkine-Hornung, A. Sampling based scene-space video processing. *ACM Trans. Graph. (TOG)* **2016**, *35*, 1–11. [CrossRef]
28. Delbracio, M.; Sapiro, G. Burst deblurring: Removing camera shake through fourier burst accumulation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
29. Delbracio, M.; Sapiro, G. Hand-held video deblurring via efficient fourier aggregation. *IEEE Trans. Comput. Imaging* **2015**, *1*, 270–283. [CrossRef]
30. Li, D.; Xu, C.; Zhang, K.; Yu, X.; Zhong, Y.; Ren, W.; Suominen, H.; Li, H. ARVo: Learning All-Range Volumetric Correspondence for Video Deblurring. *arXiv* **2021**, arXiv:2103.04260.
31. Mechrez, R.; Talmi, I.; Zelnik-Manor, L. The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany 8–14 September 2018.
32. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
33. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Adv. Neural Inf. Process. Syst. (NIPS)* **2014**, *3*, 2672–2680. [CrossRef]
34. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; Wang, O. Deep video deblurring for hand-held cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017.
37. Niu, B.; Wen, W.; Ren, W.; Zhang, X.; Yang, L.; Wang, S.; Zhang, K.; Cao, X.; Shen, H. Single image super-resolution via a holistic attention network. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 191–207.
38. Zhang, K.; Luo, W.; Ren, W.; Wang, J.; Zhao, F.; Ma, L.; Li, H. Beyond Monocular Deraining: Stereo Image Deraining via Semantic Understanding. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.