# High-Resolution Dual-Stage Multi-Level Feature Aggregation for Single Image and Video Deblurring

Stephan Brehm* Sebastian Scherer* Rainer Lienhart
University of Augsburg
Augsburg, Germany
www.multimedia-computing.de

## Abstract

*In this paper we address the problem of dynamic scene motion deblurring. We present a model that combines high-resolution processing with a multi-resolution feature aggregation method for single frame and video deblurring. Our proposed model consists of 2 stages. In the first stage, single image deblurring is performed at a very high-resolution. For this purpose, we propose a novel network building block that employs multiple atrous convolutions in parallel. We carefully tune the atrous rate of each of these convolutions to achieve complete coverage of a rectangular area of the input. In this way we obtain a large receptive field at a high spatial resolution. The second stage aggregates information across multiple consecutive frames of a video sequence. Here we maintain a high-resolution, but also use multi-resolution features to mitigate the effects of large movements of objects between images. The presented models rank first and fourth in the NTIRE2020 challenges for single image deblurring and video deblurring, respectively. We apply our framework on current benchmarks and challenges and show that our model provides state-of-the art results.*

## 1. Introduction

Motion blur in images is one of the most common and noticeable artifacts that can occur during image capture. Due to technical constraints, an image captured by a camera represents the scene over a short period of time. Camera shake and moving objects during the exposure time of the image sensor can cause significant motion blur. Reversing the motion blur in an image can be very difficult because the exact reason for it, such as the movement of all objects in the image and the camera itself, is usually unknown. As a result, image deblurring is a challenging problem in computer vision. Blurred images can be problematic in many computer vision applications, such as object detection or structure-from-motion. For example, in camera-based applications such as head motion tracking in virtual reality applications where a camera is used, rapid head movements can lead to a high degree of motion blur and consequently to a loss of tracking. In this work we try to recover the details of blurred images given a set of pairwise data with blurred images and correspondingly sharp images.

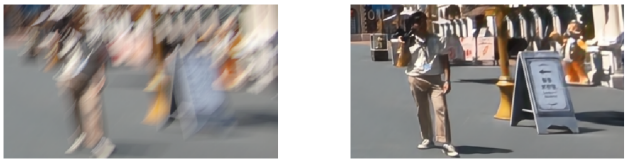


Figure 1: **[Best viewed in color]** Results of the proposed image deblurring method. **Left:** Blurred Image. **Right:** Deblurred Image.

Early work on image deblurring used preliminary information such as the type of blur kernel and additive noise [9]. In real world scenarios and applications, however, this information is unknown, which leads to poor performance of such methods. Recently, methods based on deep learning have shown significant success in image deblurring [24, 36, 35, 33, 16]. Common Convolutional Neural Networks (CNNs) use multiple down-sampling layers to expand their internal receptive field. To extract features from larger and more semantically more meaningful areas of the input, many methods use some sort of sub-sampling internally. Common options are pooling operations like max-pooling and average-pooling as well as strided convolution. For image-to-image tasks such as motion deblurring, dense spatial output at full input resolution is required. However, sub-sampling leads to a loss of spatial information and therefore has a negative effect on predictions when detailed spatial information is required. Furthermore, up-sampling layers usually follow down-sampling layers to recover the

*indicates equal contribution

spatial dimensions. However, current up-sampling techniques such as deconvolution can produce checkerboard artifacts caused by uneven overlap [28]. This has a negative effect on the performance of deep image-to-image models. To overcome these disadvantages, we simply omit most of the down-sampling and therefore do not need most of the up-sampling layers. Figure 1 shows an exemplary output of our proposed model. We analyze the use of atrous convolution for the task of image deblurring. Atrous convolution has already been used effectively in other dense-prediction tasks such as semantic image segmentation [38]. Atrous convolutions is a method that allows increasing the spatial area covered by a convolutional kernel without increasing the number of parameters. It is also known as dilated convolution. Instead of down-sampling, we use atrous convolution to increase the receptive field without reducing the spatial dimensions of the deep features of our CNN. Thus we are able to maintain a high-resolution representation of our data in all layers of our proposed model. Our contributions are as follows:

1. We propose a high-resolution model that heavily utilizes atrous convolution to obtain a large receptive field achieving state-of-the-art performance for the task of single image deblurring.

2. We propose a Multi-Level Feature Aggregation method for video deblurring that improves upon our single image performance.

3. We conduct an ablation study, that confirms that high-resolution is the primary reason for our outperformance.

We evaluate our proposed models for single image as well as video deblurring on the REDS dataset [23] provided by the NTIRE 2020 workshop challenges on image and video restoration and enhancement [25]. We further provide results on the GoPro dataset [24] and compare our results to state-of-the art approaches.

## 2. Related Work

Many approaches tried to estimate unknown blur kernels in order to reverse their effects [2, 3, 29]. Nowadays, due to the creation of synthetic data-sets using high-frame-rate cameras, we are able to use supervised techniques to train deep deblurring networks in an end-to-end fashion. Various approaches to image deblurring using CNNs have been proposed in recent years. Encoder-decoder architectures and multi-scale networks have been particularly successful [24, 36, 35, 33, 16]. These networks typically make use of representations on at least 3 different resolutions internally. Nah *et al*. [24] proposed a multi-scale architecture for blind image deblurring using a residual network structure. Their model takes a blurred image pyramid as input and residual blocks are used on every level. The output is again an image pyramid representing an estimate for the sharpened image on every level. Tao *et al*. [36] used encoder-decoder residual blocks. They alternate down-sampling and up-sampling of deep features with additional residual connections. Thus, they are able to increase the receptive field while keeping a high-resolution representation. In contrast, we omit all but one down-sampling operation as well as one up-sampling operation to keep a high-resolution representation throughout the network.

Atrous convolution, or dilated convolution, allows to enlarge the receptive field of a single convolution kernel to incorporate a larger context. Atrous convolution has been successfully employed in many spatially-dense tasks [4, 38, 42]. Yu and Koltun [38] applied atrous convolution for semantic segmentation and significantly improved segmentation performance. They showed that for tasks requiring high-resolution results, high-resolution operations throughout the network are feasible and promising. Their introduced context module consists of 7 layers that apply $3 \times 3$ convolutions with different dilation factors of 1, 1, 2, 4, 8, 16 and 1, resulting in a receptive field of $67 \times 67$ pixels. Recently, Zhou *et al*. [42] proposed a full resolution CNN for medical image segmentation. They use cascaded atrous blocks, which are similar to standard residual blocks but with different atrous rates of 1 and 3 in the convolution layers. This setting of the atrous rate was used to force each pixel within the receptive field to be covered such that there are no missing gaps. An atrous rate setting of 1 and 4 in two successive convolutions would result in a receptive field without complete coverage of all pixels. They showed that networks without down- and up-sampling layers and reasonable receptive field through atrous settings can outperform U-Net architectures with less trainable parameters. Similarly to their work, we carefully tune atrous rates to obtain a receptive field with almost full coverage.

Others analyzed the deblurring of images in videos, where the main challenge is to find corresponding content in multiple frames. Recent deep learning methods incorporate this search in networks that are trainable in an end-to-end fashion [41]. Sim and Kim [33] stacked several frames and fed them into a residual network to deblur a single image. Their model is equipped with an adaptive per-pixel kernel module to restore image details for small motion blur. Wang *et al*. [37] learn a model that utilizes deformable convolutions [5]. Deformable convolutions allow to perform convolution with adaptive kernel shapes. This allows aggregation of information taken from different locations. Our model instead aggregates spatio-temporal information on multiple resolutions to mitigate the effects of inter-frame movements.

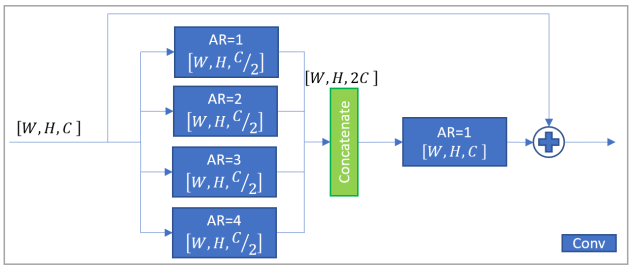



Figure 2: Residual block with atrous convolution used in our model. It performs four parallel convolutions on the input features with different atrous rates. The outputs are concatenated and now have twice the size of the input channels. The second convolution reduces the channel dimension to the same as the input, and the residual is added.

## 3. Methodology

We use a two-stage strategy where the first stage tries to remove blur in a single image and the second stage tries to align different features of a sequence of images to improve the deblurring result. We argue that the aggregation of spatio-temporal information is easier on images that have already been deblurred to a certain degree. Especially for images that are heavily blurred, aggregating details from different consecutive frames can be very difficult. Therefore, our work consists of two different networks, one with the task to deblur only a single image and one with the task to aggregate features of predictions from the first stage.

### 3.1. Stage 1 - Single Image Deblurring

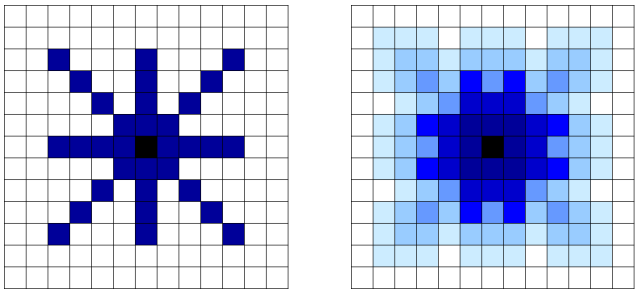
In our single image deblurring model we try to avoid down-sampling in favour of atrous convolution. However, there is no standard way for setting the atrous rate. Nevertheless, choosing a suitable atrous rate is crucial for performance. In this section, we describe our proposed model for single image deblurring and give motivation and intuition leading to the proposed atrous configuration within the presented residual block.

**Atrous Rate Setting** We propose to use multiple parallel convolutions with different atrous rates. This allows each convolutional layer to learn specific parameters for a given viewing distance, thus simulating a multi-scale approach without reducing spatial resolution. The layout of the proposed residual block is shown in Figure 2. Each block performs four $3 \times 3$ convolutions in parallel with 128 filters each and atrous rates of 1, 2, 3 and 4. We concatenate the output of all four convolutions to obtain a feature block that consists of 512 feature maps. Subsequently, we use another $3 \times 3$ convolution to combine these features and reduce the number of feature maps to 256. We add the output of this layer to the input of the block. The intuition behind this is that expanding feature depth within the residual block allows more information to pass through and can improve performance [39]. Each atrous block has a receptive field of size $11 \times 11$ pixels on the input feature map. Through the stacking of blocks the receptive field grows iteratively.

Figure 3 visualizes the receptive field after four parallel convolutions with atrous rate 1, 2, 3 and 4, respectively, with missing gaps in between (a) and the receptive field after the last $3 \times 3$ convolution (b). The resulting receptive field now has an almost square shape.

A similarly shaped receptive field could be achieved with five sequential convolution layers with a kernel size of $3 \times 3$ and atrous rate of 1 per residual block. In such a setting every convolution layer incorporates information from a growing region as the receptive field grows from layer to layer. Our approach instead allows individual layers to specialize for a specific viewing distance. This is comparable to sub-sampling the input features and applying convolution on multiple scales. This way, our approach preserves high-resolution information while simulating multi-scale processing.

The last convolution layer in every block combines the outputs of the four parallel atrous convolution layers. In Figure 3 we show that using a kernel size of $3 \times 3$ is a good choice because it combines the input features in a way that results in an almost completely filled square receptive field. The resulting weight for individual pixels is indicated by the color intensity in Figure 3. Compared to the work of Zhou *et al*. [42], where all pixels are weighted equally, we found the increased weight towards the center of the receptive field to be beneficial for image deblurring.



(a) (b)

Figure 3: Receptive field of our proposed atrous residual block. (a) shows the receptive field after concatenating the output of 4 different convolutions with atrous rates of 1, 2, 3 and 4. All layers are applied to the same input feature map. (b) shows the final receptive field of the residual block after the second $3 \times 3$ convolution.

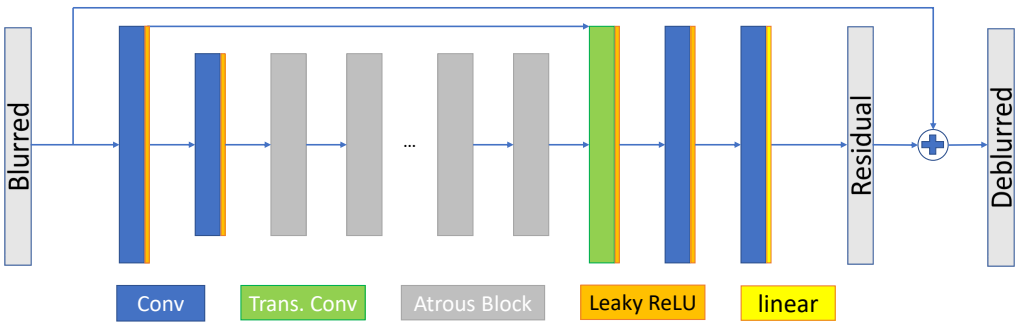



Figure 4: Proposed model architecture for atrous convolution.

**Network Architecture** Figure 4 gives an overview of the architecture of our first-stage model. We first employ two consecutive convolution layers, where the first layer uses a large convolutional kernel with a size of $9 \times 9$ to extract low-level features. In the second layer we perform down-sampling by strided convolution with a stride of 2 and kernel size of $3 \times 3$. This is followed by a total of 20 of our proposed atrous residual block to iteratively increase the size of the receptive field of the network. We up-sample the resulting features with a single deconvolution followed by 2 standard convolution layers. The output represent a residual image that we add to the blurred input image. Due to the high resolution of images from the REDS dataset and limited time for the experiments during the NTIRE 2020 challenge we used a single down-sampling layer.

### 3.2. Stage 2 - Video Deblurring

Qualitative analysis of the output of stage 1 showed that many details such as fine textures or facial expressions are not recovered equally well in different images. We observed huge local differences in reconstruction quality between consecutive frames. In such cases we should be able to improve the deblurring from stage 1 by aggregating information across multiple neighboring images in a video. However, depending on the movement of the camera and the movement of objects in the scene, finding corresponding image contents can be very difficult for CNNs. This is mainly due to the local operation of convolutions. If an object has moved too far between frames, it is almost impossible to use the additional information. We approach this problem with our proposed Multi-Level Feature Aggregation (MLFA) method. We aggregate information from consecutive frames on multiple different resolutions in the feature space of a CNN. This allows the aggregation of information over long distances at low resolutions while also maintaining high resolution details.

**Network Architecture** Figure 5 shows the high level architecture of our second stage model. Processing of video data can be categorized into three phases: feature extraction, intra-resolution feature aggregation and inter-resolution feature aggregation.

We extract features from the image at the current time-step $t$ as well as the previous image $t-1$ and the subsequent image $t+1$ separately. We use shared weights here. This is achieved by 4 consecutive convolutions with stride $s = 2$ and increasing feature depth. Note that, all images are pre-deblurred by our stage 1 model.

In phase two we aggregate image features of a certain resolution across time-steps $t-1$, $t$ and $t+1$. Here we scale all features based on their similarity to the features of time-step $t$. Like Wang *et al*. [37] we use the dot product of pixel-wise feature vectors to measure the similarity. Note that, unlike Wang *et al*. [37] we do not squash the resulting features to a range between 0 and 1, but instead calculate the dot product on unit vectors. Given two vectors $a$ and $b$ the dot product $a \cdot b$ is defined as

$$a \cdot b = |a||b| \cos(\theta) \tag{1}$$

Here $\theta$ is the angle between $a$ and $b$. Thus, if $a$ and $b$ are both unit vectors, i.e., $|a| = 1$ and $|b| = 1$, the result is the cosine of the angle between the vectors. Hence, our similarity measure gives values between -1 and 1. A value of 1 means that features from both images are identical. A value of zero means that feature vectors are orthogonal, and as such are completely different. A value of -1 means that feature vectors are pointing in opposite directions. Scaling features by this similarity measure means reducing the impact of time-series features that do not describe locally similar data. If the content of a region described by the current pixel in the feature space has changed completely from one frame to another, this information is not relevant for the output at the current position. Reducing the weight for such pixels makes it easier to focus on locally relevant features from neighbouring images. Note that it is impor-

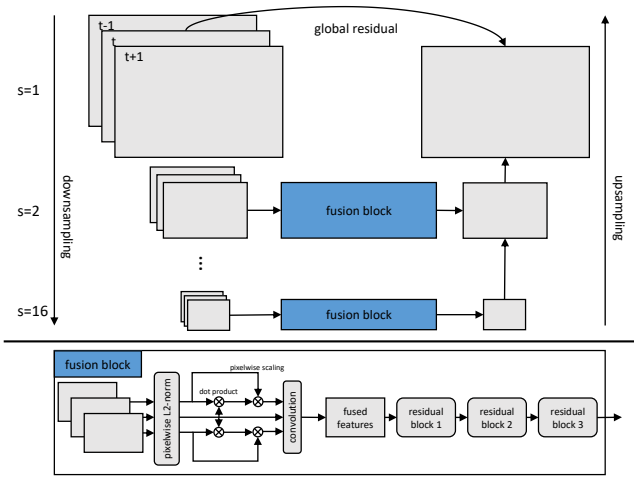



Figure 5: **Top**: High level representation of our multi-level feature aggregation method. We use strided convolutions with a stride of 2 for down-sampling. All frames are processed independently using identical weights. The result is a feature block per image and per spatial resolution. First we fuse features with identical resolution. Aggregation across multiple resolutions is achieved by up-sampling lower resolutions by a factor of two and subsequent element-wise addition with features of the higher resolution. **Bottom**: Detailed fusion method. We scale pixel-wise feature vectors to unit length. Features from time-steps $t-1$ and $t+1$ are then re-weighted by their similarity measured by dot-product with time-step $t$. We fuse the resulting features with a convolution layer that is followed by three residual blocks (two consecutive convolutions with kernel size 3).

tant to perform this re-weighting scheme at multiple feature resolutions that represent different areas of the input image in order to adapt to different amounts of movement between images.

We merge the similarity-weighted features by concatenating them along the channel-axis. We use a single convolution layer followed by three residual blocks to further improve the combined features.

In the third phase we use deconvolution to up-sample the features from phase two. We achieve inter-resolution aggregation by adding the up-sampled features to features of the new resolution in an element-wise fashion.

### 3.3. Experimental Framework

In this section we describe our experimental framework and detail relevant information regarding the used datasets, optimization methods, data processing methods as well as some insight on different learning objectives that we experimented with.

**Dataset** We trained our network on the REDS dataset [23], which has been provided by the NTIRE2020 Image Deblurring Challenge [25]. It consists of 300 videos with 100 images of size $720 \times 1280$ each, where 240 videos are used for training, 30 for validation and 30 for testing. The blurred images were synthesized by overlaying multiple sharp frames captured by a high-frame-rate camera. The sharp images of the test set are unknown. We report results on the validation set and test set if available.

We further trained our models on the GoPro dataset [24], which is a standard benchmark for evaluation of deblurring algorithms. It consists of 3214 pairs of blurred and sharp images with the same image resolution as in the REDS data set (2103 pairs for training and 1111 pairs for testing). Like the REDS dataset, the GoPro dataset is a synthetically created dataset for single image as well as video deblurring.

**Optimization** We trained our models using the Adam method [15] with initial learning rate $\eta = 0.0001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We constantly reduced the learning rate manually by a factor of 10 whenever training progress, measured on the validation set, stalled for a longer time. We trained both stage 1 and stage 2 models for about 2.5 million iterations each, independently of each other.

**Data Augmentation and Preprocessing** For training we use random image crops with a size of $320 \times 320$ pixels. This is mainly due to the enormous run-time and memory consumption when using full sized images. Preliminary experiments have shown that we are able to achieve identical results in $\approx 70\%$ of the time by simply using image crops instead of full sized images during training. We varied image brightness, hue, saturation, and image contrast by small amounts for every image crop. We also flip and rotate images at a $50\%$ rate each. Due to the combination of all these image augmentations, it is almost impossible to get the exact same image twice during training. All images were normalized to a range between -1 to 1. Each mini-batch consisted of 2 random image crops taken from different images for our stage 2 model and a batch size of 1 has been used for our stage 1 model.

**Training Objectives** Training of both stage 1 and stage 2 networks was guided by pixel-wise absolute error in RGB-Space. In both stages we used the RGB image of the current time-step $t$ as label. However, we have conducted preliminary experiments using different error formulations, which we will detail below. The baseline model for these experiments achieved 32.3dB PSNR and a SSIM of 0.901. All of these experiments were conducted in the single image deblurring setting.

GAN Learning pixel-wise tasks in an adversarial setting

has become very popular lately [12, 18]. However, our experiments showed no useful results. We tried various discriminator architectures and multiple GAN formulations such as StandardGAN [10], LSGAN[22] and Relativistic Average LSGAN[14]. None of these experiments converged to anything meaningful. These results are in line with results from Kupyn *et al*. [16], who found their GAN-based approach to not converge without adding an additional perceptual loss. In contrast to [16], we were able to learn a model that converged to a score of 29.00dB PSNR and 0.869 SSIM by combining a StandardGAN with an absolute error. This is still considerably worse than our baseline of 32.3dB PSNR and 0.901 SSIM.

VGG We added a perceptual loss [13] based on the *conv3_3* layer of the VGG16 [34] network pre-trained on ImageNet [6]. The resulting model showed faster convergence in terms of the number of parameter updates during training. However, we did not see any improvement to our baseline model in terms of PSNR and SSIM.

Edge Edges are very important for a visually pleasing reconstruction. Recent work tried to learn edges and deblurred images subsequently [7] and showed improvements. To emphasize this, we tried to improve our baseline by adding an additional absolute error between the edges of the output image and the edges of the corresponding sharp image. However, the resulting model performed worse than our baseline with 31.32dB PSNR and 0.880 SSIM.

It is commonly known that both adversarial and perceptual loss improve visual quality at the cost of quantitative performance measured by PSNR and SSIM [1]. It should be noted that we have not made a qualitative analysis of the above mentioned preliminary results.

**Other Details** In all reported models we use Leaky ReLU [21] activation functions in all convolution layers except for the output layer. Here, we use a linear activation function. At test time we simply clip the output to the correct range of values. Despite being very similar to the conventional ReLU [27] activation function, Leaky ReLU allowed us to train for a much longer time with continuous improvement in validation performance. This is probably because it does not suffer from the "dying ReLU" problem [20], as it does not have zero-slope parts. In contrast, ReLU activation functions stop gradient flow for all negative values. In our tests, training with ReLU converged much earlier to stable but lower results. Further note that we do not use any common feature normalization technique such as batch-normalization [11], but instead simply include a learned scaling factor for each individual feature map right before adding the bias term. In our experiments this technique was sufficient to keep the training stable.

To mitigate the effects of image boundaries on our results, we used reflection padding instead of zero-padding wherever possible.

| Method | PSNR | SSIM |
|---|---|---|
| ours (stage 1) | **34.44** | **0.9412** |
| Attentive Fractal Band Learning [1] | 34.20 | 0.9392 |
| DRU-prelu (ensemble)[1] | 33.35 | 0.9283 |
| Two-stage Edge-Conditioned Network[1] | 33.07 | 0.9242 |
| Reg.-Ada. Patch-hierarchical Net[1] | 32.61 | 0.9198 |
| Simplified SRN[1] | 30.04 | 0.8616 |
| V-Stacked Deep CNN[1] | 29.78 | 0.8629 |

Table 1: Comparison of methods on the REDS Motion Blur Dataset of the NTIRE2020 Challenge. Single Image Deblurring Test Data. [1] scores are taken from [25].

## 4. Evaluation

We give a comparison of various methods trained on the REDS dataset in Table 1 and Table 3 for single image and video, respectively. We achieved our best video score by separately deblurring all images with our stage 1 model and subsequently aggregating spatio-temporal information with our stage 2 model. We performed geometric self-ensemble [33, 19] to further improve performance by augmenting the input frames to four different versions by rotating and flipping. All combinations are fed into the network, transformed back to their original shape and the mean pixel value of all combinations is taken as final prediction.

**Ablation Study** To compare the effectiveness of our proposed atrous residual block, we implemented a simple encoder-decoder network with three down-sampling layers and skip connections, followed by 12 standard residual blocks with 512 filters each and a constant atrous rate of 1. The differences between this network and our atrous network are the two additional layers for down- and up-sampling and the four parallel convolutions with less filters and different atrous rates compared to a single convolution. Considering the two additional down-sampling layers, this network achieves a similar receptive field compared to our atrous network. Both networks were trained with the same data augmentation strategies, patch size and learning rate adjustment. Our proposed atrous residual network has fewer trainable parameters, but it takes longer to calculate a sharp image. Table 2 summarize the ablation study results.

This model achieved 32.3dB PSNR on the REDS validation set, while our atrous network achieved 33.9dB PSNR.

| Method | type | PSNR | SSIM |
|---|---|---|---|
| stage 1+stage 2 (ensemble) | video | **34.67** | **0.9422** |
| stage 1 (ensemble) | single | **34.19** | **0.9378** |
| stage 1 | single | 33.93 | 0.9352 |
| stage 1(w/o atrous conv) | single | 32.34 | 0.9019 |
| stage 2 | video | 33.17 | 0.9135 |
| stage 2 | single | 32.06 | 0.8971 |

Table 2: Quantitative comparison of model performance measured in PSNR and SSIM. We compare our atrous network (stage 1) to a model without atrous convolution. This model is built with standard residual blocks and additional down-sampling layers to achieve a similar receptive field. We also compare variants of MLFA (stage 2) trained on single images and video. In this way we are able to quantify the influence of spatio-temporal data provided by videos. The scores are measured on the validation set of the REDS dataset. Bold font indicates the full models for single image and video.

We find that operations on a higher resolution increase performance at the cost of computing time, which is caused by the higher spatial dimension used in all stages.

We also conducted an ablation study to show the influence of the time series data on our results. For this we compare our Multi-Level Feature Aggregation (MLFA, stage 2) model trained on videos to a model that is trained on single images. We keep the architecture identical and remove the feature extraction for time-step $t-1$ as well as time-step $t+1$. To keep things simple, we compare models that we have trained directly on the REDS data, i.e., we do not use our stage 1 model here. The model trained on video data achieves 33.17dB PSNR and 0.9135 SSIM, while the model trained on single images achieves 32.06dB PSNR and 0.8971 SSIM. This shows that MLFA utilizes spatio-temporal information provided by video data.

**Benchmark Results** We compare our model on the GoPro dataset with already known older methods and newer state-of-the art methods from previous work and the NTIRE 2019 challenge. The result of our stage 1 and stage 2 models on the GoPro dataset is shown in Table 4. We observe that both our stage 1 (single image) and stage 2 (video) achieve a lower mean squared error compared to other works (which can be seen by the PSNR score). When comparing the SSIM score, our models achieve comparable results, but cannot surpass previous work. The SSIM scores tries to model visual quality using various components such as luminance, contrast and structure.

We further provide the test results from the NTIRE 2020 challenge [25] in Table 1 and Table 3 for single image and video deblurring, respectively. For single image deblurring,

| Method | PSNR | SSIM |
|---|---|---|
| HelloVSR[2][37] | **36.96** | **0.966** |
| PAFU[1] | 36.93 | 0.965 |
| UIUC-IFP[2] | 35.71 | 0.952 |
| WDVR+[1] | 35.58 | 0.950 |
| PROMOTION[1] | 35.42 | 0.952 |
| ours (stage 1+stage 2) | 34.68 | 0.944 |
| KAIST-VICLAB[2] | 34.09 | 0.936 |
| BMIPL$_U NIST_D J$[2] | 33.71 | 0.936 |
| (modified) DMPHN + GridNet[1] | 31.85 | 0.907 |
| (modified) DMPHN[1] | 31.43 | 0.895 |
| Multi-loss Optimization[1] | 29.44 | 0.853 |

Table 3: Comparison of methods on the REDS Motion Blur Dataset of the NTIRE2020 Challenge. Video Deblurring Test Data. [1] scores are taken from [25]. [2] scores are taken from [26].

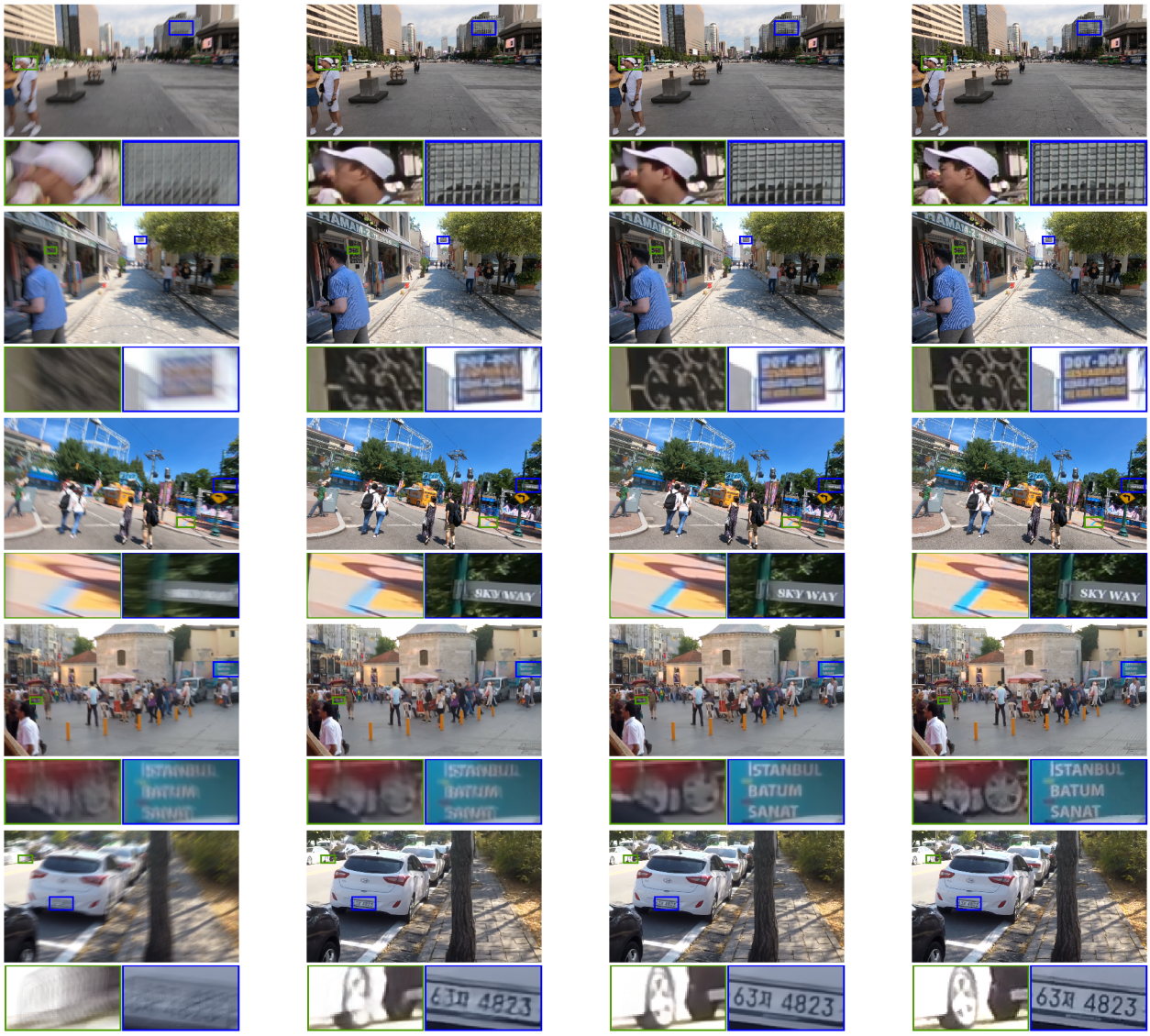
| Method | PSNR | SSIM |
|---|---|---|
| Nah *et al*. [24] | 29.23 | 0.916 |
| Kupyn *et al*. [17] | 29.55 | 0.934 |
| Shen *et al*. [32] | 30.26 | 0.940 |
| Tao *et al*. [36] | 30.26 | 0.934 |
| Purohit *et al*. [31] | 30.58 | 0.941 |
| Fu *et al*. [7] | 31.02 | 0.912 |
| Sim and Kim [33] | 31.34 | 0.947 |
| Zhang *et al*. [40] | 31.50 | 0.948 |
| Gao *et al*. [8] | 31.58 | 0.948 |
| Purohit [30] | 32.15 | **0.956** |
| ours (stage 1) | 32.61 | 0.935 |
| ours (stage 1+stage2) | **33.23** | 0.944 |

Table 4: Comparison of methods on GoPro dataset [24]. Our methods have been trained on the GoPro dataset.

our model surpasses other works and shows best performances for both PSNR, and SSIM score. In video deblurring, our model could not surpass other work, which indicates that there is still room for improvement in combining the information in image sequences. Note that our final scores are determined with the geometric self-ensemble strategy.

We also provide a qualitative analysis of the deblurring performance of our model on the REDS validation and GoPro test dataset in Figure 6. We compare our results to the work of Sim and Kim [33], which is also a video deblurring method and one of the top rankings in the NTIRE 2019 video deblurring challenge [23]. From these figures we find that both networks are able to remove motion blur very well. However, our model is better at restoring fine details in e.g. faces, grid patterns and text information.

The proposed model requires 175 ms and 400 ms computing time for stages 1 and 2 without self-ensemble. Note



(a) (b) (c) (d)

Figure 6: **[Best viewed in color]** (a) Input blurred images. (b) Results of Sim and Kim [33]. (c) Results of our proposed method. (d) Ground truth sharp image. The images of the 1st-3rd row are from REDS dataset [23], those on 4th-5th row are from the GoPro dataset [24]. Our proposed method is able to deblur fine details better.

that, in stage 2 we need to feed 3 images at once. Timings were obtained on a single NVIDIA Tesla V100m.

## 5. Conclusion

We proposed a high-resolution motion deblurring network with novel atrous residual block for the task of single image deblurring. We have extended this model for the task of video deblurring by aggregating information of different frames. Our experiments on benchmarks demonstrate the superiority of our approach in comparison to previous work.

We assume that an atrous network without any internal down-sampling could achieve further improvements. Thus, future work could remove the single down-sampling layer that we have used due to memory and run-time limitations. First experiments have shown that a full resolution model converges faster in terms of the number of parameter updates. This prospect is promising for future work. Another promising perspective is to combine the benefits of the proposed stage 1 and stage 2 networks in a single network that could be learned in an end-to-end fashion.

## References


[1] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 6

[2] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2

[3] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 40(4). 2

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[7] Zhichao Fu, Yingbin Zheng, Hao Ye, Yu Kong, Jing Yang, and Liang He. Edge-aware deep image deblurring. *CoRR*, abs/1907.02282, 2019. 6, 7

[8] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[9] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 1

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. 6

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. 6

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 6

[14] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 6

[15] Diederik Kingma and Jimmy Ba. 12 2014. 5

[16] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1, 2, 6

[17] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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 (CVPR)*. 6

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshops*. 6

[20] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019. 6

[21] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 30. 6

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6

[23] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2, 5, 7, 8

[24] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1, 2, 5, 7, 8

[25] Seungjun Nah, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2020 challenge on image and video deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2, 5, 6, 7

[26] Seungjun Nah, Radu Timofte, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 7

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*. 6

[28] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 2

[29] Liyuan Pan, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Phase-only image based kernel estimation for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[30] Kuldeep Purohit and A. N. Rajagopalan. Spatially-adaptive residual networks for efficient image and video deblurring. *CoRR*, abs/1903.11394, 2019. 7

[31] Kuldeep Purohit, Anshul Shah, and A. N. Rajagopalan. Bringing alive blurred moments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[32] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7

[33] Hyeonjun Sim and Munchurl Kim. A deep motion deblurring network based on per-pixel adaptive kernels with residual down-up and up-down modules. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 1, 2, 6, 7, 8

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 6

[35] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1, 2

[36] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 7

[37] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2, 4, 7

[38] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 2

[39] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas S. Huang. Wide activation for efficient and accurate image super-resolution. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 3

[40] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[41] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[42] Xiao-Yun Zhou, Jian-Qing Zheng, and Guang-Zhong Yang. Atrous convolutional neural network (ACNN) for biomedical semantic segmentation with dimensionally lossless feature maps. *CoRR*, abs/1901.09203, 2019. 2, 3