An analysis of Scale Recurrent Network for video deblurring

HAERINCK Guillaume
Universite Gustave Eiffel
Champs-sur-Marne, Ile-De-France, France
haerinck.guillaume@gmail.com

LAFONTAINE Laurine
Universite Gustave Eiffel
Champs-sur-Marne, Ile-De-France, France
laurine.lafontaine@outlook.fr

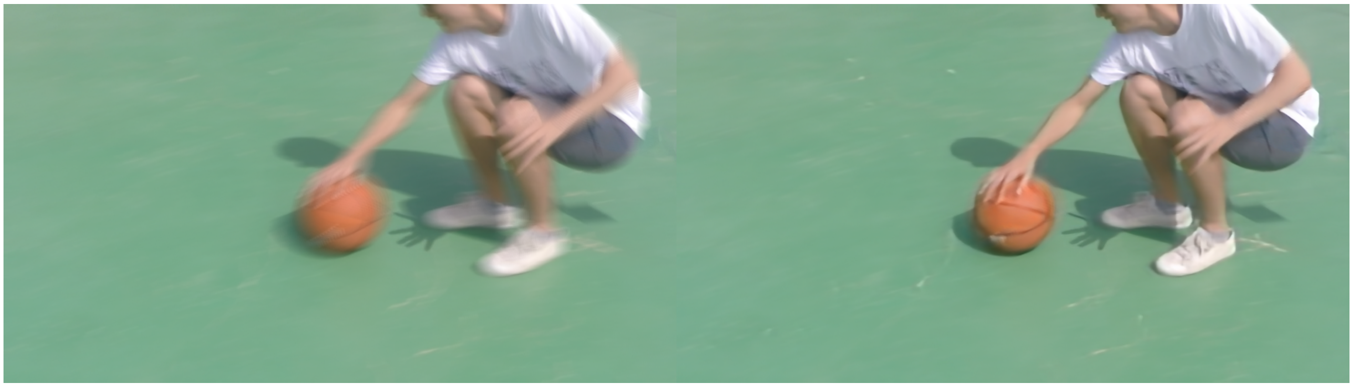

Figure 1: Left is blurred original image. Right is deblurred image. Extracted from GoPro dataset. 2017.

ABSTRACT

Video deblurring is a heavily researched topic that has seen further improvements with the introduction of machine-learning. From the recovery of car plates numbers to the improvements of handheld footage, it is a key field in constant evolution. We will review the best ranked open-sourced algorithm of this field, the Scale Recurrent Network[24] (2018), and compare its results against our own simpler implementation. Since its release, a few algorithms have outperformed it, so we will propose an analysis of key points of these papers as well. As they are not open-sourced, we weren't able to compare them with a local dataset.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Reconstruction**.

KEYWORDS

neural networks, deblur

ACM Reference Format:
HAERINCK Guillaume and LAFONTAINE Laurine. 2021. An analysis of Scale Recurrent Network for video deblurring. In . ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

The aim of video deblurring is to recover sharp latent images from blurred video frames. With the recent evolution of smartphones, security cameras and social media we now have an incredible amount of low-quality footage that needs such improvements for a similar amount of needs. One of the difficulties of the field is that the notion of blur encompasses multiple phenomena and acts in the final image in multiple ways.

In our case study the most prominent blur is the one caused by a movement of the camera going too fast relative to its shutter speed. In a frame we see it through the spread of edges and the colors being averaged-out. It is commonly called "motion blur". Another common case is the out-of-focus blur, less common in smartphones as they offer a focus to infinity, only objects too close to the camera are subject to this. Finally, we cannot ignore other environmental effects, such as dirt or steam on the lens, there are also some artifacts that can be caused by lens distortion on the border of the image or the video compressing algorithm.

The company GoPro offers a dataset[12] matching and mixing these situations taken from their action-cameras. Most of the algorithms we present have been evaluated against this dataset with measures of SSIM and PSNR. The PSNR is a quality measurement between the original and the reconstructed image, the higher is the PSNR, the better is the quality of the reconstruction algorithm. While the PSNR is a pixel to pixel comparison, the SSIM is focused on the image structure. This means that both of these measurements are valuable as they do not have the same response to the same artifacts.

In this paper, we have selected a few of the best-ranked neural networks in terms of their average PSNR and SSIM for the GoPro and related dataset. The Scale-Recurrent Neural network (SRN) being both open source and still in the top list, we will focus our attention on its structure and performance while still covering key points of other papers. We will also present a simple implementation of a CNN done for this paper and compare its performance against the SRN.

The inverse problem of video deblurring shares many similarities to other fields such as video supersampling, image denoising, tomographic reconstruction and even image segmentation. Network architecture made for one field can often be reused and adapted to another set of problems, which is why we will cite some resources concerning other areas during the following section.

2 RELATED WORK

2.1 Traditional Methods

Traditional methods for deblurring are less performant than their machine-learning counterparts but they have their own advantages. These algorithms offer a better reproducibility, we clearly understand each step taken, and they are generally more robust in front of a diametrically different dataset. These methods work with the assumption that, when an image is blurred, the information is not lost as it just becomes redistributed in accordance with some rules (except for the borders of the image). When you know or retrieve this redistribution rule, you can revert the effect to restore the image.

2.1.1 Non-blind deconvolution. First proposed by the mathematician Norbert Wiener during the 1940s, non-blind deconvolution is a reconstruction method which tries to obtain a sharp image f having as input a blurred version g and a convolution kernel h. The reconstruction can be expressed as follows :

$$g(x, y) = h(x, y) * f(x, y)$$

The kernel h is called the point spread function (PSF), it represents how the blur impacts each pixel, and will change accordingly to the types of blur. For this method to work, the PSF needs to be known in advance and selected manually. An out of focus blur can be expressed with a gaussian function, while a motion blur is more of a line-spread for a pixel.

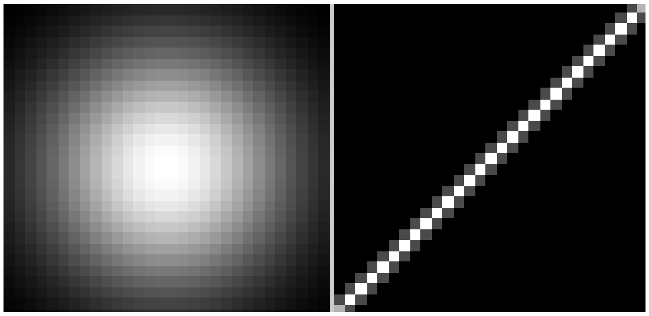

Figure 2: PSF for out of focus and motion blur [26]

The Wiener filter is not a simple inverse filter though as it takes the noise of the image into account to create a better reconstruction. The image and the noise are considered as random processes and the filter only outputs the values for which the frequency response is the Minimum Mean Square Error (MMSE). The formula can be expressed as follows :

$$H_w = \frac{H}{|H|^2 + \frac{1}{SNR}}$$

Other non-blind deconvolution filters such as Tikhonov regularization, and Lucy Richardson filter (1972) are available to obtain similar results. It can be useful to vary the method depending on the image to see which one performs the best.

2.1.2 Blind deconvolution. Blind deconvolution can be used when no information about the blur and noise are known. This method selects a first approximation of the PSF and applies a deconvolution using one of the methods presented above. The degree of quality of the resulting image is identified according to some criterion, and based on this score the PSF is tuned. This process repeats until the required result is achieved[2].

The function to create the PSF needs to be malleable enough to match many kinds of blur, but at the same time the number of parameters need to be manageable. It is important to note that this problem is underdetermined, which means that the solution is non unique. To ensure the uniqueness and the quality of the result, the PSF needs to be constrained by another set of criterias.

One well known representation of the PSF has been proposed by Markam[9] in 1999. The definition has been derived from the optical properties of a pupil:

$$h(r_j, z) = \left| \sum_k F_{j,k} a_k(z) \right|^2$$

With r_j the lateral position of pixel j, F the discrete Fourier transform and $a_k(z)$ a pupil function at frequel k and depth z. We define the pupil function as follows :

$$\begin{aligned} a_k(z) &= p_k exp(i2\pi(\phi_k + z\psi_k)) \\ p_k &= \sum_n \beta_n Z_k^n \\ \phi_k &= \sum_n \alpha_n Z_k^n \\ \psi_k &= \sqrt{(n_i/\lambda)^2 - |k_k|^2} \end{aligned}$$

With Z_k^n the n-th Zernike polynomial and n_i the refractive index of immersion medium. This all means that the PSF is parametrized by (n_i, α, β).

2.2 Machine learning methods

With an impressive breakthrough[7] made during the 2012 ILSVR challenge, machine learning based methods became the new go-to for image restoration and segmentation algorithms. While these methods are not new, the novel access to large dataset and the increase of computational power through GPU gave machine learning the much-needed tools to shine. Machine learning methods share the same process (dataset transformation, training and testing) but they differ on their neural network architecture. In this section, we will review and compare popular architecture used for image deblurring.

2.2.1 Convolutional Neural Networks (CNN). Given an input image, the role of the CNN is to reduce it into a shape that is easier to process, while conserving the features of the image used for predictions. The CNN architecture[4] is composed of a series of convolution and activation layers. One of the first successful usage of a CNN for image deblurring was used by Microsoft Research[3] in 2014. In this work, the network is composed of 3 groups of convolution-activation which is pretty lightweight and can run easily in real time.

The role of the first layer is to extract the feature of the input image (edges, corners, ridges, etc) and outputs high-resolution patches. These patches are then mapped to a lower-resolution representation and finally reconstructed to the 3-channels output image.

Another successful CNN architecture was used in 2016 for license plate motion deblurring[23]. Their network is made of 15 groups of convolution + ReLU activation layers. This network is able to handle saturation, non-uniform blur, compression artifacts and non-gaussian noise. In their paper, they compared the performance of their architecture with varying filter size and numbers of channels for each layer and found out that the higher it was the stronger their results remained against more difficult data (larger dimension and blur).

2.2.2 Auto-Encoders. Auto-Encoders are made of two parts : an encoder which extracts the features of an image and a decoder which reconstructs a higher dimensional image based on the extracted features. The center of the network is notably small in dimension as the outputed features are smaller than the input image. Auto-Encoders, just like CNN, relies heavily on convolutional layers, but they also often add pooling layers to further reduce the image dimension during the encoding stage.

This architecture is used in many image restoration fields, if not many of them with the same network[1, 8]. The auto-encoder architecture is also often modified to include skip connections for the network to conserve the details of the image. This method, popularized with the U-Net[20] paper in 2015, consists of connecting layers of the encoder to the corresponding layers of the decoder. As the encoder is reducing the dimension of the image, some details are inevitably lost in the procedure. By concatenating results from previous stages of the encoder to the decoder, we can recover this kind of data.

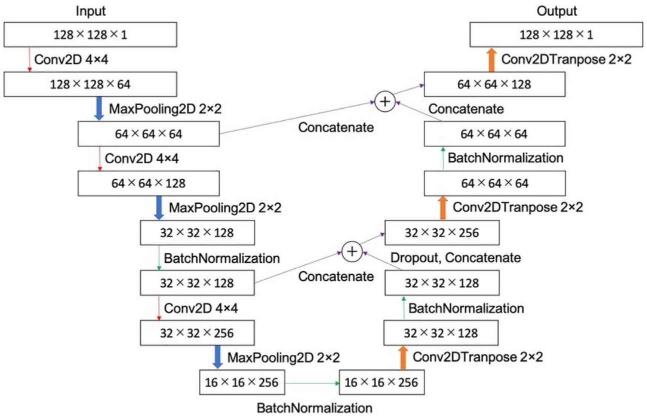

Figure 3: U-Net network schema[20]

2.2.3 Recurrent Neural Networks (RNN). Recurrent neural networks are used to improve the prediction in a sequence of data such as tracking a moving subject in a video. The basic idea[17] is to keep in memory a result from a specific step of the network and mix it with new input during the next epoch. This approach is great for short-term memory but lacks the ability to remember too old data. This problem is called the Vanishing Gradient and some architecture such as long short term memory (LSTM) and gated recurrent units (GRU) have been created to handle it. We won't cover these details here as such structures are not used in our analyzed papers.

The LSTM[6] were introduced by Hochreiter and Schmidhuber as a way for the network to remember information for long periods of time. It is composed of 4 layers and 3 gates that have the role to store or remove some of the information that goes through the cell. Further information can be found on colah's blog[15].

RNN mechanisms are necessary for video based datasets as they provide further temporal stability and more information about the overall context of the sequence. It was used on dynamic scenes by Zhang and Pan[27] for the CVPR 2018 and further improved by Nah and Son[14]. Both of their works are structured on a U-Net, for which they provide information from previous epochs at multiple stages of the pipeline.

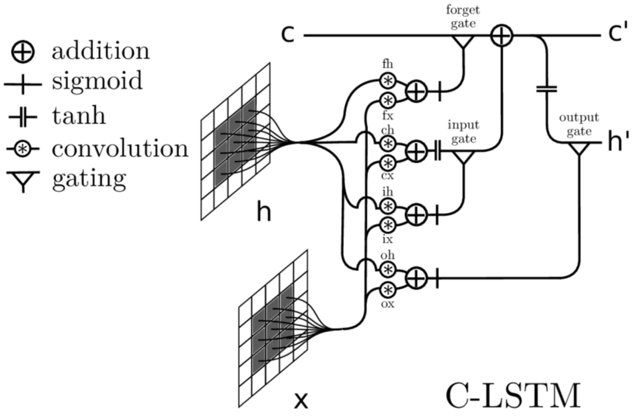

Figure 4: ConvLSTM cell schema[6]

More recent papers have refined the RNN structure[11, 19] into self-attention modules. The role of these modules is to take into account all of the previous inputs instead of the few last ones in order to increase the weight of the most valuable input areas. There are multiple ways to introduce this mechanism in computer vision and it will be further described during our Scale Recurrent Network analysis.

2.2.4 Hybrid models. SRN is no longer state of the art and has been outperformed by a few algorithms. We will present the broad ideas and structures of these hybrid networks. First and foremost we see that almost all of these papers are using a modified version of the U-Net auto-encoder, as well as improved recursivity through multiple attention modules. Cascaded entries similar to the SRN fashion are also present in some algorithms which clearly shows that these works are on the continuity of previous papers.

RADNet[18] (2019) is, at the time of writing, the 2nd ranked algorithm in terms of SSIM and 3rd for the PSNR. A rough summary would be to present it is a U-Net with a self-attention module at the bottleneck and convolutional layers capable of extracting depth motion information. These layers are called dense deformable modules (DMM) and allow the network to understand and reconstruct images where the blur acts differently depending on the distance to the camera of the objects.

MB2D[16] (2020) states that they have to blur more to deblur better. Their network is based on a CNN that is cascaded with the same inputs at different levels of blur. Split in two, the first module of their network adds and predicts more blurred images, this result is then passed unto the multi-scale deblurring module which is similar to a decoder and outputs a final image. The network reuses results from lower scales at higher scales similar to the skip connection fashion in U-Nets, and relies on multiple recurrent modules. With this novel architecture, they rank second for the GoPro dataset.

BANet[25] (2021) stands for Blur-aware Attention Networks for Dynamic Scene Deblurring and is now the best ranked algorithm in terms of video deblurring. It is structured as an asymmetric U-Net with multiple attention networks at its center and blur-aware modules right before the decoder. Their Blur-Aware attention modules are constructed in two parts, a multi-kernel strip pooling (MKSP) and attention refinement (AR) that allows them to detect the blurred areas of an image at multiple distances. The masks obtained are then used by the encoder to reconstruct high-quality images with impressive results (2db more than SRN).

3 METHOD

3.1 Problem Setting

Scale recurrent Network for Deep Image Deblurring (SRN) has been proposed in 2018 by the University of Hong Kong and was the best ranked algorithm against the GoPro Dataset at the date of contribution. The GoPro Dataset consists of multiple series of images extracted from video captured from an action camera. This footage contains all kinds of blurry footage that can be particularly difficult to restore. It is also important to remember that Image deblurring is an ill-posed problem : there is not enough information in the base image to produce a higher quality one so data extrapolation is necessary.

One way to increase valuable data for our reconstruction is to take into account multiple entries. The frames we are trying to restore do not exist in a vacuum, they belong to a set of frames making up the video. In machine learning such mechanisms are introduced through the use of recurrence and the concept of attention. The latter was a new concept in the time of publication so it wasn't properly introduced in the architecture as the latest deblurring papers did.

The complexity of the restoration is directly impacted by the size of the network. While adding layers can provide better results, it also poses problems for convergence, computation time and the inner limit of the auto-encoder architecture. As the encoder reduces the size of the input at each step, there is a certain limit that the feature map cannot go below. The SRN team decided for a fair but limited number of layers and selected a ResBlock structure in them, proven to be of higher quality than standard convolution as explained further below.

3.2 Network Architecture

The SRN is an hybrid architecture which can be summarized as a recurrent U-Net with ResBlocks layers. It takes as input a blurry image (B1) which is downsampled two times by half its quality (B2 and B3). The network is run 3 times, one for each level of quality starting with the lowest one. The result of each run is fed back into the network for the next run. At the end the network outputs a deblurred image.

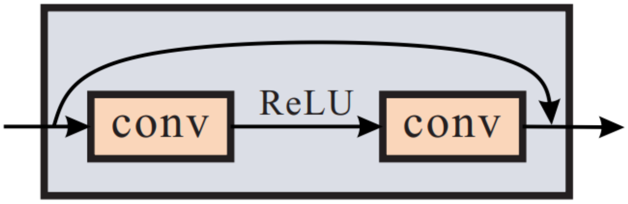

Figure 5: ResBlock schema

3.2.1 ResBlock. The base layer of SRN is a residual learning block, commonly called ResBlock. Presented by He and Zhang[5] from Microsoft Research, these double-convolution layers introduce skip connections as a way to reduce the degradation of the training accuracy. The skip connection is simply a concatenation of the entry tensor with the output tensor, it is seen as a way to blend different levels of extracted features. With this mechanism the error evolution is more stable and can be visualized as smoothing the gradient descent.

The decoder is made of 3 groups of 1 convolution followed by 3 ResBlocks. The decoder is symmetric with groups of 3 ResBlocks followed by 1 deconvolution layer. The kernel size is fixed at 5, and the stride is always 1 except for the 4 inner convolution and deconvolution for which the stride is set to 2. The entry and

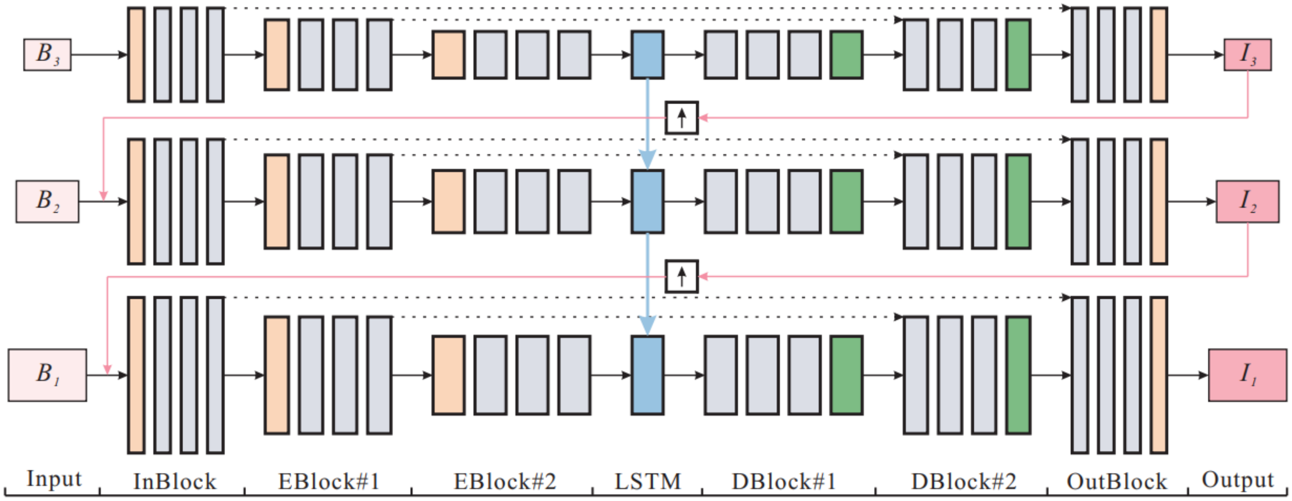

Figure 6: SRN-DeblurNet schema

output of the network is of dimension 32 (8 bits images with 4 channels), going through each group of the encoder this dimension is multiplied by 2, and reduced by 2 during the decoder, which means that the bottleneck is of dimension 128.

3.2.2 *U-Net.*

In comparison with the original U-Net, SRN is adjusting multiple parameters while maintaining the overall structure and ideas. SRN has 22 convolution operations in its encoder and decoder which is twice as much as the original U-Net. Resblocks have replaced the standard convolution and this allows the team to have more operation as explained above, and an added complexity.

However in terms of tensor dimension SRN is much more narrow with the maximum size being 128 in the bottleneck while it is 1024 on U-Net. It is explained by the fact that to make a valid reconstruction, the SRN team needs the feature map not to go through a lot of downsampling. U-Net being a segmentation network, their first goal is to extract as finer features as possible, the decoder is not reconstructing the image but trying to add a valid tag for the features extracted.

The main attribute of U-Net is skip connection, in a similar fashion as the ResBlock structure, and this attribute is respected by the SRN team. In practice, the output of each group in the decoder is kept in memory and concatenated with the input of the matching group in the encoder. Some cropping of the tensor can be necessary to respect the network dimension. This process in computer vision allows us to keep higher-level features which would be lost in the convolution steps.

3.2.3 *Recurrent U-Net.*

In order to bring information from the previous frames to the reconstruction, an LSTM was introduced at the bottleneck of the autoencoder. The implementation chosen is known as Conv LSTM[21] which is proven to perform well on image-based datasets. The particularity of this cell is to keep the input dimension through a padded convolution, so the tensor does not need to be flattened.

SRN is structured so that each frame is run through the network 3 times, one for each resolution. As the network is run from the lowest resolution to the highest, the LSTM cell is upsampled by 2 at each run. Similar to what is done with the skip connection, this process of using the same network multiple times with different levels of resolution is a way to extract different levels of features and further enhance the final reconstruction. As the input from the second run is concatenated with the last output of the network, the result is smoothed-out which has a tendency to provide better results for deblurring.

4 RESULTS

Our results are obtained with an NVIDIA GTX 960M GPU. We discarded the use of CPU in order to keep our computation time as short as possible. SRN is implemented with the TensorFlow framework while our custom CNN and AE uses PyTorch. All experiments are conducted on the same dataset with the same training configuration.

4.1 Dataset preparation

We focus on the GoPro dataset[12] which contains 3,214 blurry images with their sharp groundtruth. This dataset has created its blurry images by averaging consecutive short-exposure frames, which has proven to be realistic[13, 22].

For SRN we keep the input image size at 1280x720 pixels, but our custom CNN / AE has to resize the image to 1024x1024 during data loading in order to work more efficiently. We are using 2,103 pairs for training and the remaining 1,111 pairs for evaluation. It is a distribution of 25 percent for validation and 75 percent for training.

4.2 Training settings

SRN uses Adam solver as their optimization algorithm, the learning rate is computed with an initial value of 0.0001 and is diminished to 1e6 for 2000 epochs. According to their experiments 2,000 epochs are enough for convergence. At each iteration, they sample a batch of 16 blurry images and randomly crop 256 x 256-pixel patches as training input. Ground truth sharp patches are generated with the same method. Since their network is fully convolutional, images of arbitrary size can be fed in it as input. For testing images of size 1280 x 720, the running time of their proposed method is around 1.6 seconds. All trainable variables are initialized using the Xavier method[10]. The loss chosen is a simple L2-norm between the network output and the ground truth.

On the side of our custom CNN/AE, we also use Adam as the optimization algorithm, with a learning rate of 0.001. We are using a dynamic learning rate, reduced based on the ReduceLROnPlateau() learning rate scheduler. The patience is 5 and factor is 5. So, if the loss value does not improve for 5 epochs, the new learning rate will be old learning rate * 0.5. We achieve convergence at 500 epochs, and save the output of the network at this step.

4.3 Quality evaluation

We compare SRN against our simpler CNN and AE implementations. Comparison against state of the art is available on the original SRN paper and the work cited for hybrid models in part 2.1.4. We evaluate the results through training time, peak to noise ratio (PSNR) and structural similarity index (SSIM). For all results in this section we have included the full-frame result images in a zip file next to this paper.

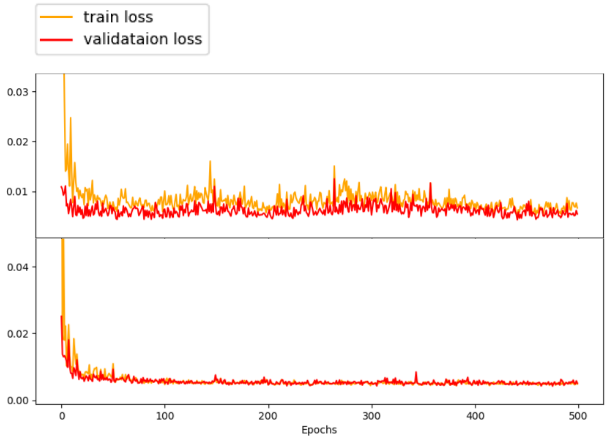

Figure 7: Loss evolution over training. Top is CNN, bottom is AE

4.3.1 Training time.

For our CNN and AE, we trained them with 500 epochs which took approximately 8 hours for the CNN and 4 hours for the AutoEncoder. We haven't trained SRN again due to large training time explained in their paper (72 hours for 2000 epochs on a higher-end graphics card than ours). As expected, the AE has a better training time than the CNN : only 50 to 100 epochs are necessary while the CNN needs at least 400.

4.3.2 Image to image comparison.

As a disclaimer, our custom implementation of CNN/AE saves the output images in black and white due to a problem happening during the conversion from tensors to RGB values. The calculation of PSNR and SSIM for our custom answer is done between the resized ground truth and the network output before black and white conversion. The results obtained by the CNN and AE are exactly the same so we will indicate only one.

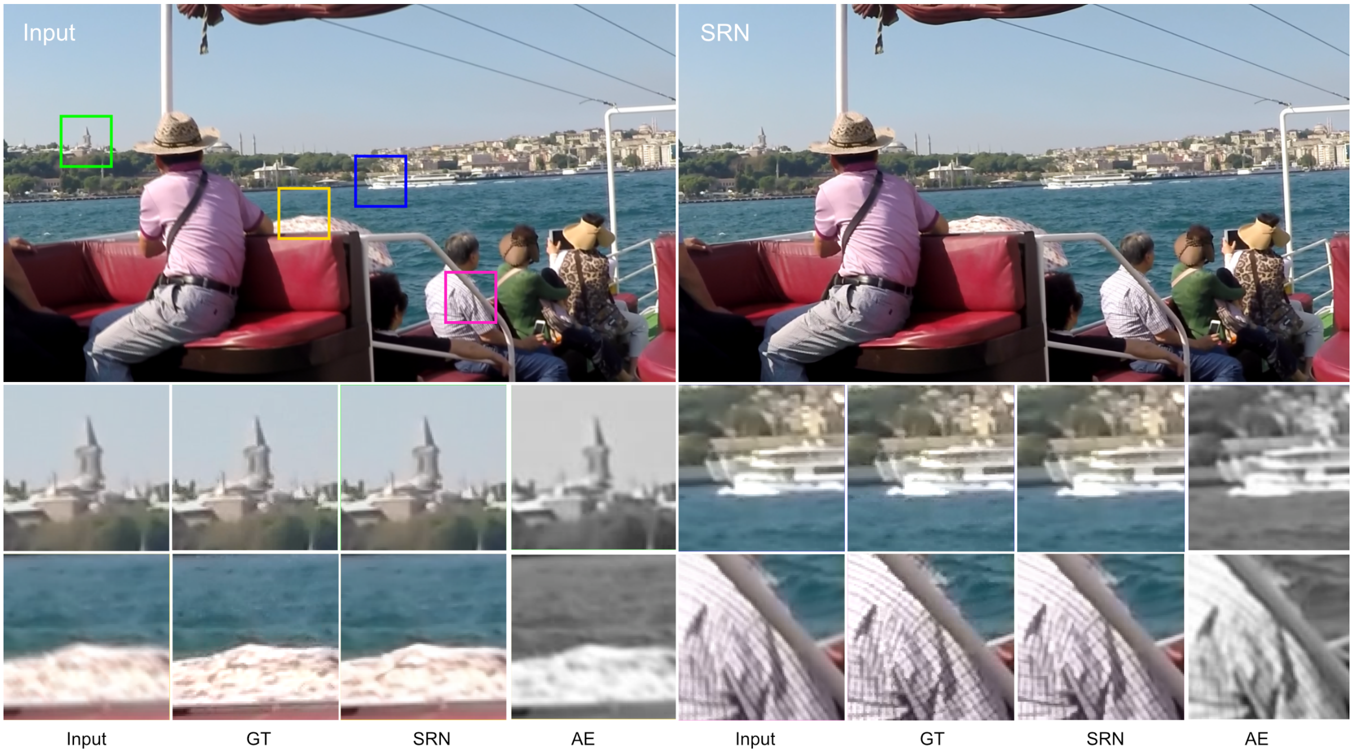

Figure 8: The results obtained. Input is the blurred image. GT is ground truth. SRN is SRN output. AE is auto-encoder output

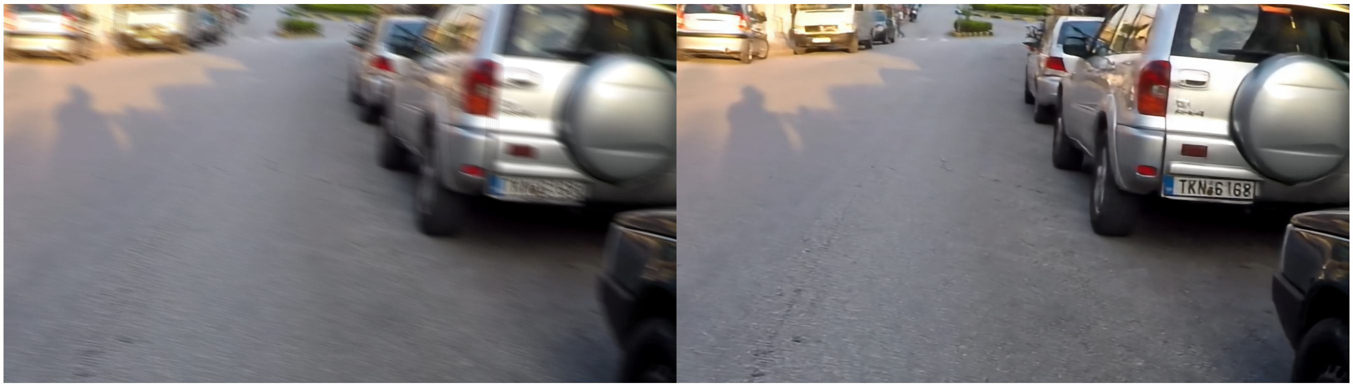

Figure 9: Blurred input on the left and SRN output on the right

It is clear by the results that SRN is much more powerful than a simple CNN/AE. While these simpler networks can provide good results on simple images, when asked for more difficult data such as with multiple moving objects or some depth the results are unconvincing.

SRN in comparison is much more robust against multiple kinds of blur and difficult setups, as it works with an LSTM module, it also handles way better a serie of frames and prevents many distortions from happening due to its awareness of the global context. We see a difference in terms of 5 to 10db for the PSNR and about 0.2 for the SSIM, which is huge. Each year, the state of the art is approximately improving by 1db. If we use this graduation we can say that SRN is a 5 to 10 years improvement over simple networks.

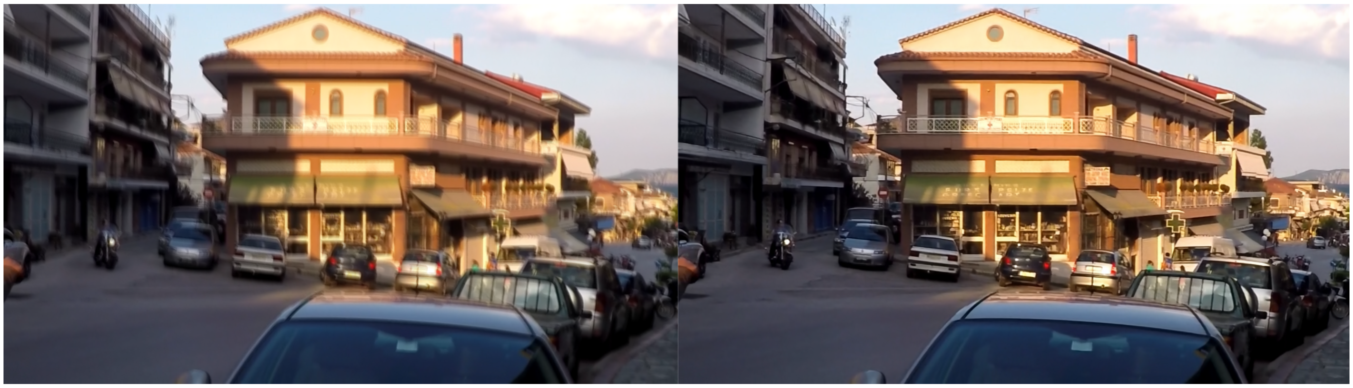

Figure 10: Blurred input on the left and SRN output on the right

Table 1: PSRN and SSIM of the multiple scenes

	SRN	AE
Boat (Fig.8)	37dB / 0.73	23dB / 0.49
Car (Fig.9)	35.2dB / 0.69	-
House (Fig.10)	31.75dB / 0.65	-
Boy (Start Fig.)	37.63dB / 0.97	-

Yet, SRN is now behind multiple state of the art networks, the gap between SRN and BaNET, the current winner on the GoPro dataset, is 2db on PSNR and 30 on SSIM for a much higher computation time (424ms against 26ms). We believe that the lack of powerful attention modules is a start to understand the difference in performance.

5 CONCLUSION

Image deblurring is an actively researched field and it is no-surprise that SRN is no longer the best algorithm. Yet, it has proven to be effective and powerful and many of its ideas were reused and improved upon by the state of the art. Most of the recent evolution were done through attention modules for which we can see a glimpse in the recurrent architecture of SRN. These recent networks are hybrids, taking parts and bits from previous architecture and merging them into a coherent whole. We now even see popular softwares such as photoshop introducing deblurring tools through the use of deep learning, an opening that annonces a bright future and many more innovations in this field.

REFERENCES

[1] Siavash Arjomand Bigdeli and Matthias Zwicker. 2017. Image Restoration using Autoencoding Priors. arXiv:arXiv:1703.09964

[2] Loic Denis, Eric Thiebaut, and Ferreol Soulez. 2013. *Image deblurring*. ENS Lyon SIERRA. http://www.ens-lyon.fr/PHYSIQUE/Equipe3/SIERRA/old/pdf/DENIS_L_slides.pdf

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a Deep Convolutional Network for Image Super-Resolution. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 184–199.

[4] Vincent Dumoulin and Francesco Visin. 2016. A guide to convolution arithmetic for deep learning. arXiv:arXiv:1603.07285

[5] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[8] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf

[9] Joanne Markham and José Conchello. 1999. Parametric blind deconvolution: A robust method for the simultaneous estimation of image and blur. *Journal of the Optical Society of*

America. A, Optics, image science, and vision 16 (11 1999), 2377–91. https://doi.org/10.1364/JOSAA.16.002377

[10] Diganta Misra. 2017. *Xavier initialization and batch normalization, my understanding*. Medium. https://medium.com/@shiyan/xavier-initialization-and-batch-normalization-my-understanding-b5b91268c25c/

[11] Diganta Misra. 2020. *Attention Mechanisms in Computer Vision: CBAM*. paperspace. https://blog.paperspace.com/attention-mechanisms-in-computer-vision-cbam/

[12] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. 2017. Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[13] S. Nah, T. H. Kim, and K. M. Lee. 2017. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 257–265. https://doi.org/10.1109/CVPR.2017.35

[14] Seungjun Nah, Sanghyun Son, and Kyoung Mu Lee. 2019. Recurrent Neural Networks With Intra-Frame Iterations for Video Deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Christopher Olah. 2015. *Understanding LSTM Networks*. Colah's blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[16] Dongwon Park, Dong Un Kang, and Se Young Chun. 2020. Blur More To Deblur Better: Multi-Blur2Deblur For Efficient Video Deblurring. arXiv:2012.12507 [cs.CV]

[17] Michael Phi. 2018. *Illustrated Guide to Recurrent Neural Networks*. Medium. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

[18] Kuldeep Purohit and A. N. Rajagopalan. 2019. Spatially-Adaptive Residual Networks for Efficient Image and Video Deblurring. *CoRR* abs/1903.11394 (2019). arXiv:1903.11394 http://arxiv.org/abs/1903.11394

[19] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. 2019. Stand-Alone Self-Attention in Vision Models. *CoRR* abs/1906.05909 (2019). arXiv:1906.05909 http://arxiv.org/abs/1906.05909

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241.

[21] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Montreal, Canada) *(NIPS'15)*. MIT Press, Cambridge, MA, USA, 802–810.

[22] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. 2016. Deep Video Deblurring. *CoRR* abs/1611.08387 (2016). arXiv:1611.08387 http://arxiv.org/abs/1611.08387

[23] P. Svoboda, M. Hradiš, L. Maršík, and P. Zemcík. 2016. CNN for license plate motion deblurring. In *2016 IEEE International Conference on Image Processing (ICIP)*. 3832–3836. https://doi.org/10.1109/ICIP.2016.7533077

[24] Xin Tao, Hongyun Gao, Yi Wang, Xiaoyong Shen, Jue Wang, and Jiaya Jia. 2018. *Scale-recurrent Network for Deep Image Deblurring*. The Chinese University of Hong Kong. https://arxiv.org/pdf/1802.01770.pdf

[25] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. 2021. BANet: Blur-aware Attention Networks for Dynamic Scene Deblurring. arXiv:2101.07518 [cs.CV]

[26] Vladimir Yuzhikov. 2012. *Restoration of defocused and blurred images*. Retrieved February 9, 2021 from http://yuzhikov.com/articles/BlurredImagesRestoration1.htm

[27] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. H. Lau, and M. Yang. 2018. Dynamic Scene Deblurring Using Spatially Variant Recurrent Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2521–2529. https://doi.org/10.1109/CVPR.2018.00267