Members:	**Taimoor Tariq**	**20174577**	**Neural network EE538**
	Juan Luis Gonzalez	**20178196**	**Final report**
	Thang Vu	**20174587**	**22/Dec/2017**

Real-time Video Super-Resolution Using Deep Convolutional Neural Networks

1 Introduction and related work

Super Resolution is the enhancement of image/video resolution. Super Resolution is inherently an ill posed problem. As we have to increase the number of pixels, there are a great number of possible outcomes. SR algorithms can be divided into two categories, model based and learning based algorithms. On the most popular model based learning approach the Bayesian approach which used Monte-Carlo methods for optimization. It had a disadvantage of being very slow. Model based algorithms such as Bayesian approaches have been used for a long time but with the advent of Deep Learning and Convolutional Neural Networks, there is great potential for learning based approaches for video and image super resolution.

Image super resolution using CNNs has been a topic of a great amount of research for some time. In Theory, a CNN is able to learn a non-linear mapping from input to output producing high-resolution mappings of low-resolution inputs [1]. As SR is inherently an ill posed problem, the nonlinearity of the CNN can provide a very good inverse mapping. Some methods use input images of smaller sizes and learn mappings to high resolution. Whereas some techniques first make use of bicubic interpolation to get an input image of the required size and then use the CNN to improve resolution [1]. A network architecture called SRCNN has been proved efficient in single image super resolution [1]. It is considered a benchmark while designing SR algorithms. We will also compare our networks performance with it.

Video or multi-frame super resolution (MFSR) has not made nearly as much progress as Single image super resolution. The reason for the difference is that videos have some spatio-temporal relationships in between successive frames that can be used to get even better results than image super resolution. These spatio-temporal relationships can be represented in terms of optical flow vectors between frames. Existing architectures have been inefficient in efficiently harnessing these relationships to produce precision architectures [2]. In short, alignment of successive frames is an issue. If we use computationally expensive frame alignment techniques, the frame rate of the video cannot be conserved. Therefore, it is better to design a network that provides good performance without any requirement of pre-processing. In theory, a CNN should be able to learn difference between frames automatically.

In [1], Dong et. al are the first to proposed a CNN-based model (SRCNN) to address the ill-posed problem in SR. Although the results were very promising, the network architecture is relatively shallow (3 layers). The authors of the SRCNN stated that they performance of their network decreases with a deeper network. They also use a very large kernel size of 9x9. Our understanding is that a deeper network with smaller kernels should have a greater receptive field and more non-linearity. In short, it should give better performance. In [3], the authors presented the MFCNN method which is deeper than SRCNN (5 layers) and uses multiple frames as the input. The experiment results showed a significant improvement compare to SRCNN, so "the deeper the better" seems still true in SR problem. In this project, we propose a Video Super Resolution using very deep convolutional networks. We will show that the performance of our method is improved as the network depth increase. Our final model has 20 layers with the same kernel size of 3x3. Although our kernel size is smaller than that of SRCNN and MFCNN, the receptive field in our model is much larger. To overcome the difficulties of training deep networks, we use the residual method. Besides, most SR algorithms make use PSNR only for their loss functions. But in many cases, PSNR is not enough to evaluate video perceptual qualities. We therefore incorporate a human visual system inspired quality metric called SSIM as another optimization objective for the network. We will show that this multi-task learning inspired approach not only improves the visual perceptive quality of the output, but also helps improve PSNR.

2 Proposed method

2.1 Network architecture

We design a very deep network with N layers as shown in Figure 1. We group the center frame with 2 previous and 2 successive frames which are use as the input. The last layer delivers a high-resolution output corresponding to the center (3rd) input frame. In the intermediate layers, we have 64 filters of size 3x3x64, where a filter operates on a 3x3 spatial region across 64 channels. The first layer operates on 5 consecutive input frames. The last layer, used for image construction, consists of a single filter of size 3x3x64. We also introduced a residual connection between the center frame and the output layer. One problem with a very deep network is that the size of the output is reduced due to applications of the convolutional layers. To resolve this issue, we pad zeros before the convolutions to keep size of all feature maps same. It turns out that zero padding works surprisingly well and the boundary pixel are also predicted accurately. Now we will explain the intuition and theory behind our design.

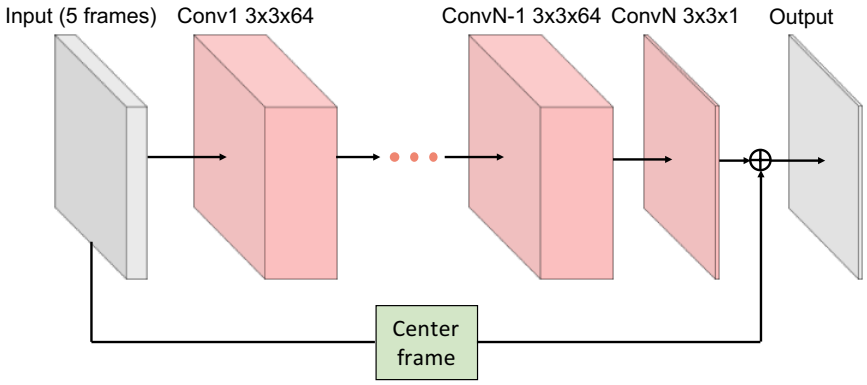


Figure 1: Our network architecture. 5 consecutive frames are used as the input and the model infer the high resolution for the center frame.

Convolutional neural networks exploit spatially local correlation by enforcing local connectivity between neuron of adjacent layers. By using a deeper architecture and a small kernel size of 3x3, the receptive field increases keeping the number of learnable parameters small. In the task of SR this corresponds to the amount of contextual information that can be exploited to infer high frequency components. As SR is an ill posed problem, using a deeper network with larger receptive field can detect more important patterns from neighboring pixels, which aid in delivering a better quality output. Moreover using a deeper layer introduces more non-linearity, which improves the mapping ability of the network.

2.2 Residual learning

In CNN based SR, an exact copy of the image goes through all convolutional layers giving us the output. With a deep network, this becomes an end-to-end relation requiring long-term memory. The vanishing/exploding gradients problem becomes an issue in this case. This can be solved simply be residual learning. In this project, we proposed a deep network architecture that learns residual images. As the input and output images are largely similar, we define a residual image $\mathbf{r} = \mathbf{y} - \mathbf{x}$ where most of the values are likely to be zero or very small. Therefore, our network actually learns to predict this residual image. We observed that a residual network converges faster and gives much better results for the SR problem upon convergence. In our opinion, the residual network will improve performance and convergence in all problem where the input and output are highly correlated.

2.3 Loss function

The common metric used to optimization is PSNR. It is widely used as a quality assessment metric in lossy compression. It is very closely related to mean square error. Minimization the mean square error between the output of the network and the ground truth high-resolution images corresponds to the maximization of PSNR.

$$\begin{aligned} \mathbf{MSE}(\mathbf{y}, \hat{\mathbf{y}}) &= \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 \\ \mathbf{PSNR}(\mathbf{y}, \hat{\mathbf{y}}) &= 20\log(s) - 10\log \mathbf{MSE}(\mathbf{y}, \hat{\mathbf{y}}) \end{aligned} \tag{1}$$

where s is the maximum possible pixel value (255 in our case). The problem is the same PSNR output presents the different quality perception in many cases. The structural similarity index (SSIM) was inspired by the human visual system [4]. It measures the perceived change in structural information while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image. The structural similarity index (SSIM) between two images x and y is defined as follows:

$$\mathbf{SSIM}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{(2\mu_{\mathbf{y}}\mu_{\hat{\mathbf{y}}} + c_1)(2\sigma_{\mathbf{y}\hat{\mathbf{y}}} + c_2)}{(\mu_{\mathbf{y}}^2 + \mu_{\hat{\mathbf{y}}}^2 + c_1)(\sigma_{\mathbf{y}}^2 + \sigma_{\hat{\mathbf{y}}}^2 + c_2)} \tag{2}$$

where $\mu_{\mathbf{y}}$, $\mu_{\hat{\mathbf{y}}}$, $\sigma_{\mathbf{y}}$, $\sigma_{\hat{\mathbf{y}}}$, $\sigma_{\mathbf{y}\hat{\mathbf{y}}}$ are the local means, standard deviations and cross-covariances for the images $\mathbf{y}$ and $\hat{\mathbf{y}}$, and c_1, c_2 constants are typically set to $(0.01s)^2$ and $(0.03s)^2$.. We adopted a Multi-Task Learning approach for our network [5]. We simultaneously optimized the MSE and the SSIM to provide a visually appealing output as close to the ground truth as possible. The net loss function we used is as follows.

$$L = \alpha\mathbf{MSE} + (1 - \alpha)(1 - \mathbf{SSIM}) \tag{3}$$

Multi-Task Learning has shown to improve performance and generalization of networks for a variety of tasks and it shown good results for our problem as well. We will explain in the result section that the incorporation of an extra learning objective not only improves the PSNR further but also improves the perceptive visual quality of the output.

3 Experiment

3.1 Experiment settings

For fair comparison, we use the same training and testing dataset for all the methods. We use Myanmar and India videos (we will call it the IndMya dataset) from a publicly available video database [6]. The original videos have 4K resolution (3840 × 2160 pixels), we extract 120 scenes (5 consecutive frames each scene) from the two videos and down-sample them to 960 × 540 pixels resolution, which are used as the ground truth. From the extracted scenes, 112 scenes is used for training, and the remaining is devoted for testing. Although the training and the testing scenes are different, they can share some common features because they are from the same videos. We therefore use the second test set Vid4 which is also used in [7]. All the experiments all employed with the upscaling factor ×2, ×3 and ×4.

To train our network, we use Adam optimizer momentum and weight decay are set to, 0.9, and 10^{-6}, respectively. The initial learning-rate is set to $10^{-}3$ and decay the learning-rate with the factor 0.5 every 20 epochs. In our loss function, α is set to 0.9. To enhance computational efficiency, during the training phase, we split the frame into 48×48 patches with stride 36 while keeping the frame order in each scene. There are total 60000 patches and 100 of them are used as our mini-batch size. Our model is implemented using Pytorch [8], which is widely used for training deep network. We train our model on Intel i7-4820k (3.47GHz) CPU with 64GB RAM and single Titan X GPU. It takes roughly 5 hours for the algorithm to converge with 100 epochs.

3.2 The deeper the better

We train our network with different network depths while the other parameters remain unchanged. As shown in Figure 2, we get higher PSNR when the network is deeper. This can be explained as follows. Super resolution can be consider as the learning non-linear mapping between low-resolution and high-resolution features. Increasing the network depth is referred to as increasing the model capacity, which enables the network to learn the feature maps better. Besides, with a larger receptive field, the model can infer new pixels more accurately. However, if data complexity is unchanged, the PSNR tends to saturate as the network depth (also number of parameters) increases. As the memory and processing capacity might be limited for some devices (e.g. mobile and embedded devices), the optimal network depth therefore should be selected to have a reasonable PSNR with a low number of parameters.

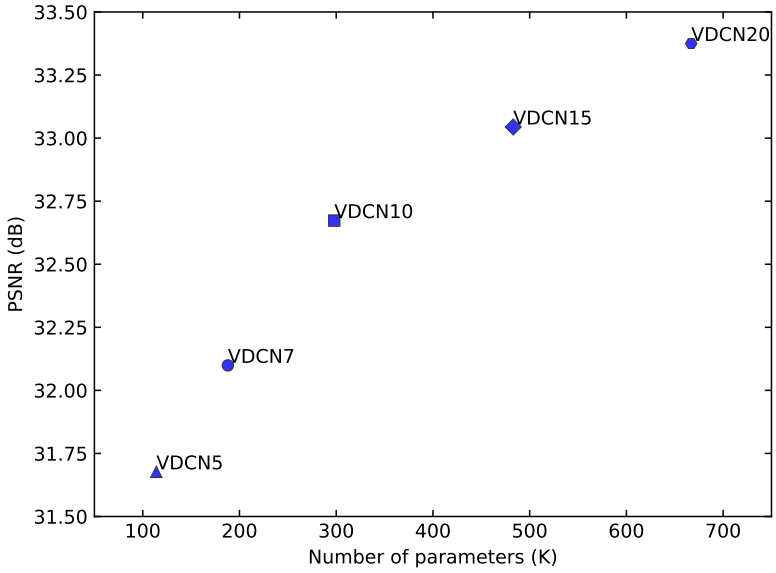


Figure 2: Performance of our model with different network depths

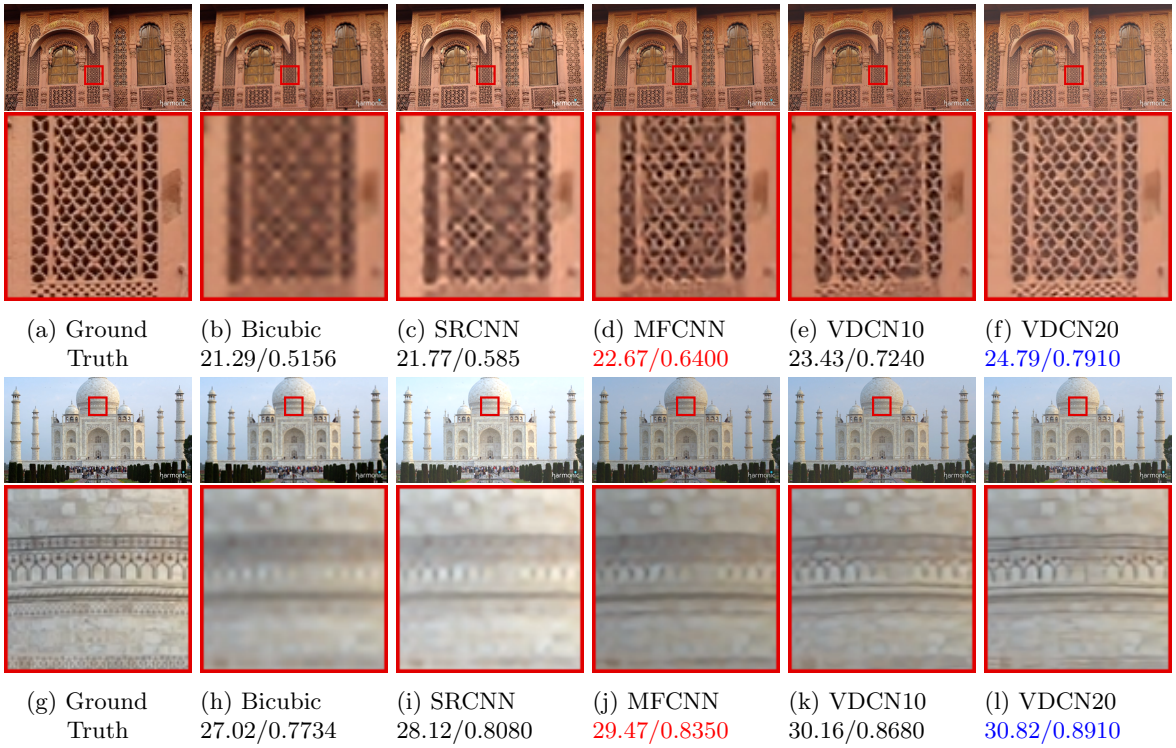
3.3 Comparison with existing models

We now present quantitative and qualitative comparison between the proposed and existing models under three upscaling factors ($\times 2$, $\times 3$, and $\times 4$). The reference models are SRCNN [1], MFCNN [3], and the bicubic interpolation is presented as the baseline. We choose the number of layers to 10 and 20 for our method. Table 1 and Table 2 summarize the quantitative performances in terms of PSRN and Structural SIMilarity index (SSIM) for IndMya and vid4 datasets. The two VDCN models outperform all existing methods in all datasets and scale factors, in both PSNR and SSIM. It is remarkable that for the the difficult dataset vid4, our method advances the previous method with large margins, which are 0.73, 0.68, and 0.34 dB on scale factor $\times 2$, $\times 3$, and $\times 4$, respectively.

Qualitative comparison among the methods are illustrated in Figure 3 and Figure 4. Our proposed method produces relatively clear and sharpen edges with respect to pattern in all figures, while other algorithm show blurry and corrupted HR reconstructed results. It is clear that our method outperforms the existing algorithms both qualitatively and quantitatively.

Table 1: Benchmark results. Average PSNR/SSIMs for scale factor ×2, ×3, and ×4 on IndMya dataset. (BLUE and RED indicate the best performance of our methods and the best performance of previous methods, respectively)

Scale factor	Metric	Bicubic	SRCNN [1]	MFCNN [3]	VRES10 (ours)	VRES20 (ours)
2	PSNR (dB)	32.25	34.95	36.13	37.71	38.43
	SSIM	0.9155	0.9489	0.9586	0.9711	0.9746
3	PSNR (dB)	28.93	30.57	31.62	32.67	33.36
	SSIM	0.8288	0.8726	0.8928	0.9168	0.9276
4	PSNR (dB)	27.29	28.32	29.36	30.02	30.54
	SSIM	0.7622	0.8026	0.8263	0.8578	0.8743



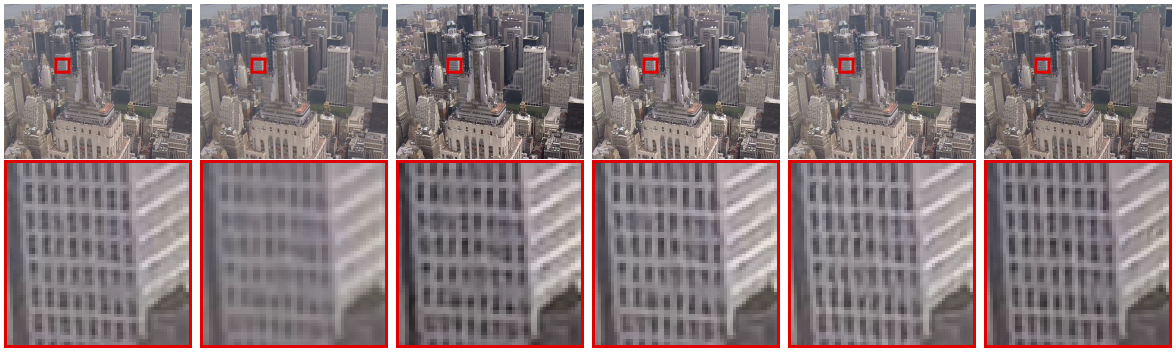
(a) Ground Truth (b) Bicubic 21.29/0.5156 (c) SRCNN 21.77/0.585 (d) MFCNN 22.67/0.6400 (e) VDCN10 23.43/0.7240 (f) VDCN20 24.79/0.7910

(g) Ground Truth (h) Bicubic 27.02/0.7734 (i) SRCNN 28.12/0.8080 (j) MFCNN 29.47/0.8350 (k) VDCN10 30.16/0.8680 (l) VDCN20 30.82/0.8910

Figure 3: Qualitative comparison. Frames are taken from IndMya test set with the up-scale factor ×4

Table 2: Benchmark results. Average PSNR/SSIMs for scale factor ×2, ×3, and ×4 on IndMya dataset. (BLUE and RED indicate the best performance of our methods and the best performance of previous methods, respectively)

Scale factor	Metric	Bicubic	SRCNN [1]	MFCNN [3]	VRES10 (ours)	VRES20 (ours)
2	PSNR (dB)	28.43	31.20	31.58	32.24	32.31
	SSIM	0.8676	0.9329	0.9397	0.9538	0.9555
3	PSNR (dB)	25.28	26.62	26.98	27.56	27.66
	SSIM	0.7329	0.8088	0.8270	0.8601	0.8628
4	PSNR (dB)	23.79	23.98	25.05	25.39	25.37
	SSIM	0.6332	0.7090	0.7281	0.7625	0.7637



(a) Ground Truth (b) Bicubic 30.01/0.8860 (c) SRCNN 32.42/0.9360 (d) MFCNN 35.57/0.9600 (e) VDCN10 36.43/0.9710 (f) VDCN20 37.16/0.9740

(g) Ground Truth (h) Bicubic 22.01/0.6976 (i) SRCNN 22.98/0.7680 (j) MFCNN 23.31/0.7900 (k) VDCN10 24.40/0.8450 (l) VDCN20 24.58/0.8510

Figure 4: Qualitative comparison. The frame in the first row is taken from vid4/city with up-scale factor ×2. The frames in the second row is taken from vid4/calendar with up-scale factor ×3.

3.4 Real-time processing

Real-time processing is a must for every video super resolution method. In this subsection, we examine the processing time, in the other word, the frame rate that the models can handle. All the algorithms are tested with a video of full HD 1920x1080 pixels resolution and the result is shown in Table 3. It is noted that we use single GPU for experiments, we measure the VDCN10 to fit the model in the GPU. As expected, the SRCNN has the best performance as it is the most shallow network. Although our model is deep, the processing time surpass the real-time processing (24 fps) by a huge margin.

Table 3: Processing time comparison

Scale	SRCNN	MFCNN	VDCN10 (ours)
Processing time (s)	5.7×10^{-4}	9.0×10^{-4}	1.7×10^{-3}
Frame-rate (fps)	1745	1107	599

4 Conclusion

In this project, we proposed a solution to the ill posed video super resolution problem using a very deep convolutional neural network, that we called VDCN. Multiple frames, very deep convolutional neural networks and residual connections were used. By first upscaling the low-resolution frames using bicubic interpolation, we let the deep convolutional neural network learn and reconstruct the high frequency residue from the low frequency up scaled frames. The residue is then added to the up scaled low frequency central frame to get the final high-resolution frame. We concluded that the PSNR was not a proper perceptive visual quality metric and introduced an additional learning objective to optimize the SSIM. The experiment result shows that our proposed method outperforms the existing methods by a large margin while ensuring the real-time processing.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[2] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," *arXiv preprint arXiv:1611.05250*, 2016.

[3] A. Greaves and H. Winter, "Multi-frame video super-resolution using convolutional neural networks."

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[5] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[6] "Harmonic inc, myanmar and india 4k video sequences [online]. avaiable: http://www.harmonicinc.com/resources/videos/4k-video-clip-center."

[7] C. Liu and D. Sun, "On bayesian adaptive video super resolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 346–360, 2014.

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.