Accurate Image Super-Resolution Using Very Deep Convolutional Networks

Manolache Andrei
Department of Mathematics and Computer Science
University of Bucharest
Academiei 14, Bucharest, 010014
andrei_mano@outlook.com

Abstract

We will reproduce a classical state-of-the-art Super Resolution (SR) approach proposed by Jiwon Kim et al. that was published at CVPR 2016 [1].

The approach proposed by the authors is building a deep convolutional network inspired by VGG-net [2] used for ImageNet classification. The model has a depth of 20 layers and is cascading small filters many times in a deep network structure. The slow convergence speed problem is solved by learning residuals only and using very high learning rates that are enabled by gradient clipping and adjustable learning rates.

The proposed network also supports multi-scale training - typically, one network is created for each scale factor, but we will train models that can take x2, x3 or x4 scaling factors and produce higher resolution outputs.

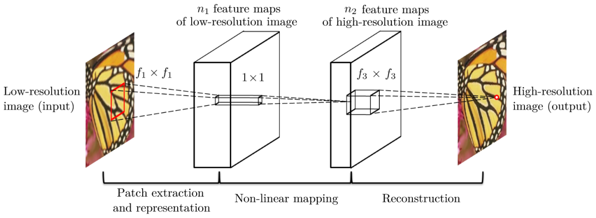
1. Introduction

Single image super-resolution (SR/SISR) is a classical problem in Computer Vision that aims to reconstruct a high-resolution (HR) image from one single low-resolution (LR) input image. SISR are a widely used set of Computer Vision techniques that have applications in radar and sonar imaging, surveillance, image post-processing, entertainment and even in medical imaging where often higher resolution, low-noise images are desired.

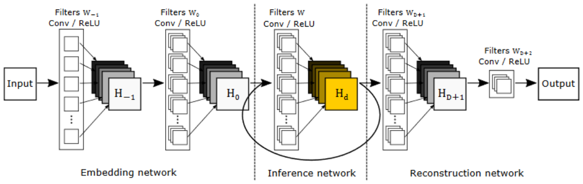
The SR problem was tackled by Harris [3] as early as 1964 when he established the theoretical foundation for the SR problem by introducing the theorems of how to solve the diffraction problem in an optical system [4]. Nowadays a common SR technique is bicubic interpolation, thus we shall consider it the baseline for our experiments. Other modern techniques are using internal patch recurrence [5] and recently learning methods are producing state-of-the-art results using models that are mapping the LR patches to the HR ones - the most notable approaches are using random forests [6] and Convolutional Neural Networks (CNN) [7].

2. Related work

SRCNN (Super-Resolution Convolutional Neural Network) [7] is representative for deep learning-based SR, the network structure is not deep - only three layers used for patch extraction, non-linear mapping and reconstruction: the LR input is upscaled to the desired size using bicubic interpolation, the first convolutional layer then extracts a set of feature maps, the second layer maps the feature maps to HR patch representations and the last layer combines the predictions to produce the HR output.


n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
Low-resolution image (input)
$f_1 \times f_1$
1×1
$f_3 \times f_3$
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

DRCN (Deeply-Recursive Convolutional Network) [8] is a method proposed by the VDSR authors. As in VDSR, 20 layers of 3x3 convolutions with the same number of filters are used but the network consists of three sub-networks: embedding, inference and reconstruction networks. The embedding network takes an interpolated LR image and represents it as a set of feature maps with two convolutions, the inference network takes the output from the embedding network and goes through a single recursive layer, the same filter being applied to feature maps recursively, followed by ReLU; once inference is done, the feature maps are fed into the reconstruction network to produce the HR output.



3. VDSR - Setup and Experimental Results

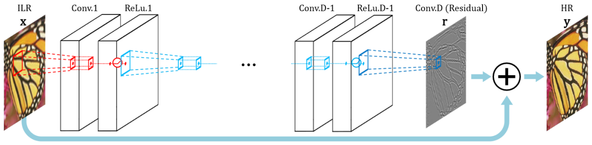
3.1. Setup

We are implementing VDSR (Very Deep Super-Resolution) in *Python* using the *PyTorch* framework.

Training: Our models are using a dataset[1] composed of 91 images from Yang et al. [9] and 200 images from Berkley Segmentation Dataset [10]. The images are cropped into disjointed 41x41 patches so we can enlarge our dataset to 2490 samples[2]. The images are converted to the *YCbCr* color space and only the downscaled, bicubic-interpolated Y channel is fed into the networks since the structural information is preserved into the greyscale copy of the image. We are using just random horizontal and vertical flips for data augmentation since it seems that random rotations are hurting the convergence rate of the network. We are training a single model for multiple scales (x2, x3, x4), this is accomplished by randomly shuffling images of multiple scales into every batch.

Testing: We are using four datasets: 'Set5' [11], 'Set14' [12], 'Urban100' [15] and 'B100' [13] [14].

Training parameters: We use a network of depth 20 and a deeper one of depth 25. We're training the networks in batches of size 64, the optimizer used is Stochastic Gradient Descent with the momentum set to 0.9 and weight decay set at 10^{-3}, the loss function is $\frac{1}{2}\|y-x-f(x)\|^2$ averaged over the training set. We're using rectified linear units (ReLU) as our activation function, thus we are using the procedure described by He et al. [16] for weight initialization. We're training the networks in 80 epochs, the learning rate is initially set to 10^{-1} and it decays by 10^{-1} every 20 epochs, gradient clipping is also used to avoid exploding gradients.



3.2. Experimental Results

Peak signal-to-noise ratio (PSNR) is used for evaluating the models.

Scale	VDSR Paper	VDSR20	VDSR25	Bicubic
2x	37.06	35.41	35.39	33.66
3x	33.27	33.80	33.77	30.39
4x	30.95	32.90	32.92	28.42

Table 1: Performance in PSRN on 'Set5', the higher the better. Color red indicates best performance and blue indicates second best performance. VDSR20 and VDSR25 are our models.

Scale	VDSR Paper	VDSR20	VDSR25	Bicubic
2x	33.03	33.54	33.50	30.24
3x	29.77	32.42	32.41	27.55
4x	28.01	31.82	31.81	26.00

Table 2: Performance in PSRN on 'Set14'. The 20-layer network performs just slightly better than the deeper one.

Scale	VDSR Paper	VDSR20	VDSR25	Bicubic
2x	31.90	33.20	33.17	29.56
3x	28.82	32.18	32.17	27.21
4x	27.29	31.66	31.66	25.96

Table 3: Performance in PSRN on 'B100'.

Scale	VDSR Paper	VDSR20	VDSR25	Bicubic
2x	31.90	32.57	32.52	26.88
3x	28.82	31.75	31.75	24.46
4x	27.29	31.28	31.28	23.14

Table 4: Performance in PSRN on 'Urban100'. Performance between VDSR20 and VDSR25 is almost identical.

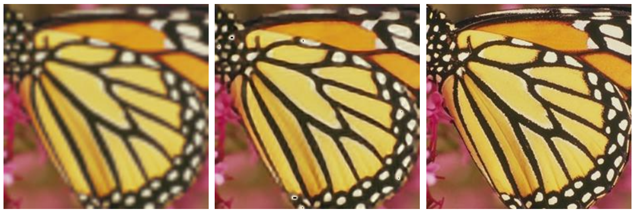


Figure 1. Bicubic Interpolation, VDSR and Ground Truth

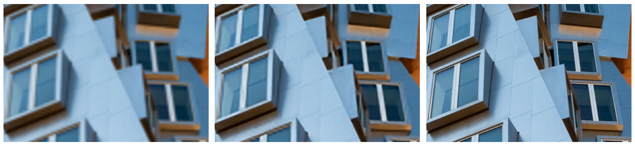


Figure 2. Bicubic Interpolation, VDSR and Ground Truth

4. Conclusions

We have explored the super-resolution method proposed by Jiwon Kim et al. that uses a very deep, residual-learning convolutional network. An interesting remark is the fact that by removing the random rotation from our data augmentation it appears that we obtain a model that converges faster and has a better performance at higher scales, but lower scale performance can sometimes decrease quite dramatically. It is also to be noted that by increasing the number of hidden layers without changing anything else in the network architecture doesn't seem to improve on the model's performance and can even lead to a model that infers slightly worse results in certain conditions.

[1] https://cv.snu.ac.kr/research/VDSR/

[2] Random crops before every training epoch were also tried but the results were average at best.

References

[1] J. Kim, J. K. Lee, K. M. Lee, ”Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In *CVPR* (2016).

[2] K. Simonyan, A. Zisserman. ”Very deep convolutional networks for large-scale image recognition”. In *ICLR* (2015).

[3] J. L. Harris, ”Diffraction and Resolving Power” J. Opt. Soc. Am. 54, 931-936 (1964).

[4] Yue et al., ”Image super-resolution: The techniques, applications, and future. Signal Processing” (2016).

[5] D. Glasner, S. Bagon, M. Irani, ”Super-resolution from a single image. In *ICCV* (2009).

[6] S. Schulter, C. Leistner, H. Bischof, ”Fast and accurate image upscaling with super-resolution forests”. In *CVPR* (2015).

[7] C. Dong, C. C. Loy, K. He, X. Tang, ”Image super-resolution using deep convolutional networks”. In *TPAMI* (2015).

[8] J. Kim, J. K. Lee, K. M. Lee, ”Deeply-Recursive Convolutional Network for Image Super-Resolution”. In *CVPR* (2016).

[9] J. Yang, J. Wright, T. S. Huang, Y. Ma., ”Image super-resolution via sparse representation”. In *TIP* (2010).

[10] D. Martin, C. Fowlkes, D. Tal, J. Malik., ”A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”. In *ICCV* (2001).

[11] C. G. Marco Bevilacqua, Aline Roumy, M.-L. A.Morel., ”Low-complexity single-image super-resolution based on nonnegative neighbor embedding”. In *BMVC* (2012).

[12] R. Zeyde, M. Elad, M. Protter. ”On single image scale-upusing sparse-representations”. In *Curves and Surfaces, pages 711730. Springer* (2012).

[13] R. Timofte, V. De Smet, L. Van Gool. ”A+: Adjusted anchored neighborhood regression for fast super-resolution”. In *ACCV* (2014).

[14] C.-Y. Yang, M.-H. Yang. ”Fast direct super-resolution by simple functions”. In *ICCV* (2013).

[15] J.-B. Huang, A. Singh, N. Ahuja. ”Single image super-resolution using transformed self-exemplars”. In *CVPR* (2015).

[16] K. He, X. Zhang, S. Ren, J. Sun. ”Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015