Online Super-Resolution using Deep Internal Learning

Mohit Goyal
UIUC
mohit@illinois.edu

Video: https://drive.google.com/file/d/1BaINhmM8GKSS5OZK5qQyTlDTV8Zophrb/view?usp=sharing (First 2 minutes can be skipped if you are acquainted with Super Resolution.)

1. Introduction

Recently, Convolutional Neural Networks (CNNs) have been shown to be extremely effective at several visual tasks such as Semantic Segmentation [8], Image Inpainting [16], Image Denoising [12] as well as *SISR* [2]. We discuss two streams of research developed around SISR techniques based on deep learning in the past few years:

Supervised: Supervised SISR method learn a non linear mapping, through as a deep CNN, from LR images to HR images. These methods typically require large datasets with paired images (LR and HR) and computational resources to train Neural Networks so as to generalize well on unseen images. LapSRN [7] is one such CNN based supervised SISR method which utilizes a Laplacian pyramid network to perform efficient super resolution. It uses a deep CNN which takes a LR image as input and predicts the difference between the HR image and the bicubic upsampled LR image. This is different other methods such as SRCNN[2], FSRCNN[3] which takes a bicubic umsampled image as input and output the HR image.

Unsupervised: Unsupervised SISR based methods exploit the property of self similarity in natural images. Self similarity approaches assume that small patches recur inside an image at various locations and scales. Therefore, these patches can be extracted, augmented and then utilized in classical SR approaches [4][15]. Recently, Assaf et al. proposed Zero Shot Super Resolution (ZSSR) [1] which trains a small image specific CNN on each image by extracting LR-HR pairs from the image itself which serves as the training data. Note that the generation of these pairs requires a downsampling kernel which can be estimated directly from LR image [9] thus improving the quality of generated SR images. ZSSR can also utilize any externally available information such as noise or downsampling kernel that can be incorporated during image pair generation. Since this method operates on a single image only, overfitting can severely degrade the quality of generated SR image. To circumvent this issue, training data is augmented with different down sampled versions (self supervision) of the original image. To generate the training inputs, the test image and its augmented versions are down sampled to produce LR image (down-sampling kernel information can be incorporated here). Deep image prior [13] is another unsupervised SISR method that performs the same task with a single image but a different mathematical formulation utilizing CNNs.

While ZSSR doesn't require any external datasets, it is still unclear if using external images (even with different sensor noise) is strictly going to degrade the performance of ZSSR. Moreover, since natural images indeed share a common prior [13], this prior can certainly improve the performance or reduce the training time of ZSSR. Therefore, in this work, we dig deeper into these question and provide evidence that using external datasets is certainly useful for run time as well as the quantitative reconstruction of SR images. We also introduce a new loss function that equally weighs the reconstruction of each frequency band instead of just better reconstruction of only lower frequencies (typically favored by pixel wise mean squared errors). Using these results, we finally present a novel algorithm based on deep internal learning to perform online super resolution (OSR). Specifically, given a stream of images (not necessarily videos) a super resolution deep network can be trained dynamically in an online fashion while quickly adapting to each image. The weights are transferred over as the new image arrives followed by adaptation using back propagation. The learnt prior on previous images allows faster convergence and better reconstruction.

Our contributions/results can be summarized as follows:

1. We introduce a robust loss function using the difference of gaussian pyramids for minimising the distance between images.
2. We show that external data can be utilized along with ZSSR which strictly improves the performance in the case of clean data.
3. We present an online algorithm for super resolution

(OSR) which can transfer the knowledge learned from the previous image onto the next image while improving the reconstruction for every image in comparison to ZSSR.

2. Description of Proposed Method

2.1. Image Pair Generation

Given an image, ZSSR requires paired LR-HR images to learn the mapping from LR $\rightarrow$ HR domain. Let's assume a down sampling kernel $\mathcal{K}$ used to obtain the provided LR image from the true HR image. In the case where $\mathcal{K}$ is unknown, we use bicubic downsampling by default. Given the LR image, we perform several augmentations such as Scaling, Translation, Random Cropping while also sometimes preserving the original image generating a set $\{\mathcal{Y}_i\}$. The images serve as the HR ground truths. Then for each Y_i, we obtain the low resolution input image using $\mathcal{X}_i = \mathcal{K}(\mathcal{Y}_i)$. Then the dataset $\{(\mathcal{X}_i, Y_i)\}$ is then used to train our deep CNN for SISR. We generate 3000 images in all our experiments, the original implementation also uses the same number of augmented images.

2.2. Network Architecture

We experimented with various architectures other than the one originally used by ZSSR. The original architecture contains seven two dimensional convolution layers each followed by ReLU activation with a kernel size of three and padding of one unit. Each layer outputs 128 feature maps. These seven layers are followed by one more convolutional layers identical to those before but outputs 3 feature maps. There is no downsampling involved and the CNN is fully convolutional. The input to the architecture is a bicubic upsampled version of $U(\mathcal{X}_i)$ and the task is to learn the residual information $\mathcal{Y}_i - U(\mathcal{X}_i)$, where U is the bicubic upsampling kernel. We experimented with UNet style architecture [10], Residual Fully Convolutional architectures but none of them seemed to provide any significant improvements while sometimes rather severely overfitting or underfitting to the task. We observed that using batchnormalization layers [5] before or after ReLU acitvations does not accelerate the training at all. Originally ZSSR uses a single image to perform each update to the CNN model. However, even with increased batch size and more number of iterations, we consistently observed high variance in training and test loss sometimes failing to converge. It can also be because the training dataset size is much smaller or there is not enough diversity among images. We also observed that small dropout in the first two layers also helped in improving generalization along with a batch size of 8 in comparison to the original architecture. Also, instead of predicting the residual linearly, we use hyperbolic tangent activation after the final convolutional layer such that the predicted residual is restricted to $[-1, 1]$.

ZSSR*: We will denote batching and dropout along with hyperbolic tangent activation on the residual with an asterisk (ZSSR*) in the experimental section.

2.3. Loss function

ZSSR originally uses pixel wise mean square error (MSE) to train the CNN, however we found out that using a *DoG Loss*, defined below, results in better generalization on the test image. For an input image X, we denote the gaussian kernel with $\mathcal{K}_g$, and $\mathcal{K}_g^r(X) = \mathcal{K}_g(\mathcal{K}_g^{r-1}(X))$ same as r composite convolution with $\mathcal{K}_g$. Then DoG_i can be written as

$$DoG_i(X) = \mathcal{K}_g^i(X) - \mathcal{K}_g^{i+1}(X) \tag{1}$$

Now, DoG pyramid of k levels is the set $\{DoG_1(X), DoG_2(X) \ldots, DoG_k(X), \mathcal{K}_g^{k+1}(X)\}$.
Then DoG Loss with k levels between images X and Y can be written as

$$\mathcal{L}_{DoG}^k(X, Y) = W \sum_{i=1}^{k} (MSE(DoG_i(X), DoG_i(Y)) + MSE(\mathcal{K}_g^{k+1}(X), \mathcal{K}_g^{k+1}(Y))$$

where MSE is the mean squared error and W is a hyper parameter. Since typically, DoG have smaller magnitude than the blurred image, we set W to be 10 to scale up the gradient backpropagted for errors in higher frequency bands.

Since pixel wise distances do not correctly describe the perceptual difference between images [17], we also experiment with Perceptual Loss [6] to train the CNN model for SISR. This loss computes the distance between representations learned by VGG net [11] trained on large scale image classification, instead of the RGB pixel space. We refer the readers to the original paper for a detailed description of this loss. For our experiments, we minimize the mean squared error between the first six layers of pretrained VGGNet followed by averaging across all six layers. We also scale the Perceptual Loss by a factor of 2 during training. This is to ensure that approximate range of the DoG Loss and Perceptual Loss is similar to allow a fair comparison.

2.4. Online Super resolution (OSR)

We consider a stream of LR images, where the task is to perform super resolution on each image. In this case, the images might or might not be correlated and is not restricted to videos. OSR (refer to Figure 1) starts from a pretrained or randomly initialised model (depending on the availability) and then update the model for the first image. For each image, several LR-HR image pairs are generated and then the model is updated using mini batch gradient descent. This model is then used to generate SR image by

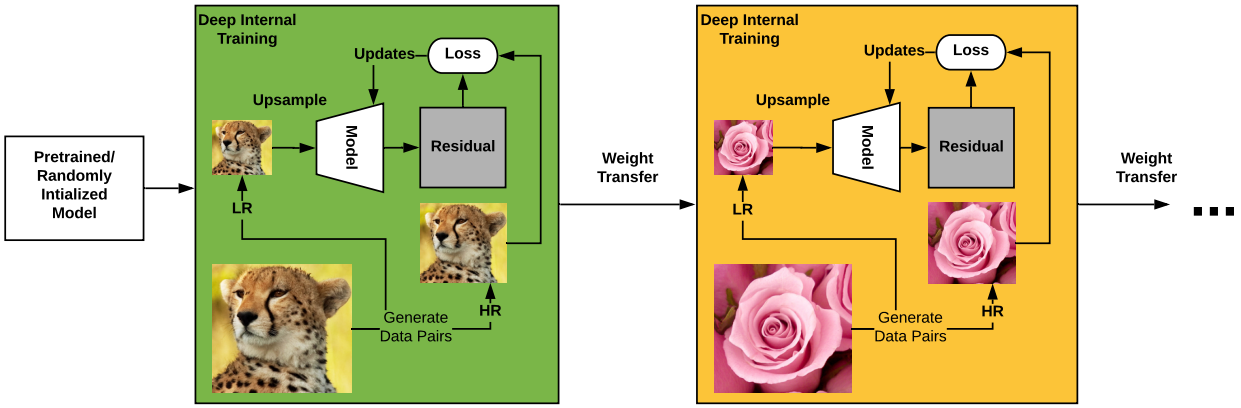

Figure 1: Description of Online Super Resolution using Deep Internal Learning for first two images. A randomly initialized/pretrained model is then trained on the first image using deep internal learning by generating data pairs from the LR image. The model takes in the upsampled image and learns to predict the residual from HR image. This trained model is then saved and used for the next image to initialize the Model parameters.

taking the original LR image as input. This model is then transferred over to the next image and the same process is repeated. OSR has multiple benefits over ZSSR or a pretrained CNN. Firstly, since the image contents might vary, a pretrained CNN would not be able to generalize to unseen images. ZSSR, being an unsupervised method, can adapt to each image but it is not efficient because ZSSR cannot exploit information from either external datasets. OSR does not suffer from these issues. Since, each image is considered individually and a separate model is trained, it can generalize better than supervised methods. Moreover, since we are adaptively training the model over images, it can also capture relationships between images by transferring weights learned from one image to the other. OSR can also utilize external datasets by pretraining a model on that dataset and using it for initialisation.

Training Details For OSR, we use the ZSSR* (with DoG Loss, dropout and batching) model. For each image, 3000 LR-HR pairs are generated (cropped to 128 $\times$ 128 shape) and then our model is trained for two epochs with a mini batch of 8 image pairs. We used Adam optimizer to update our model with a initial learning rate of 0.001. We decrease the learning rate by a factor of 10 after 250 and 500 iterations. Using a mini batch for training is slightly slower as compared to original ZSSR by 3-4 times. However, using batches results in better optima, ensures stable training and generally gives better test time performance. For all the experiments, we work with a scaling factor of 2 between LR and HR images. Since, in this work, we experiment with clean data only, we use bicubic downsampling to generate image pairs in all experiments. All the experiments are performed on Google Colaboratory and the code is available at `https://github.com/mohit1997/ZSSR/tree/pos`

3. Experiments and Results

3.1. Datasets

Since, OSR is optimized on each image for approximately 700 iterations, we only experiment with subsets of image datasets. We evaluate OSR enabled SISR methods on T91 and General-100 datasets. These datasets are briefly described below:

T91: This dataset contains 91 RGB images ranging from flowers, fruits to human faces etc. The average resolution of images is 264 $\times$ 204.

General-100: This dataset contains 100 images rangings from animals, food, humans and indoor scenes. The average resolution of images is 485 $\times$ 381.

For evaluation, we compute PSNR (Peak Signal to Noise Ration) and SSIM (Structural Similarity Index). A brief description for the metrics is provided by Zhihao et al.[14].

3.2. External Code Sources and Data Availability

Datasets can be accesed at: `http://vllab.ucmerced.edu/wlai24/LapSRN/`
ZSSR Method was downloaded from: `https://github.com/HarukiYqM/pytorch-ZSSR`
Perceptual Loss was adopted from : `https://github.com/richzhang/PerceptualSimilarity/`

3.3. Model and Loss Selection

We now discuss the impact of two design choices made over the original ZSSR. Firstly, we change the loss function from MSE to DoG pyramid with 6 levels obtained with a gaussian filter with a standard deviation of 3 pixels. Then we further change the underlying CNN model with a hyperbolic tangent activation, dropout after first and the second layer along with a mini batch training (denoted as ZSSR*). Table 1 shows the comparison among three approaches on T91 dataset. We did not include the results of Perceptual Loss because in most cases no visible difference is obtained and perceptual loss doesn't minimize MSE Loss and PSNR and SSIM metrics do not apply there. We also note that ZSSR* is 3-4 times slower than original ZSSR. However, a larger batch size converges faster so in practice less updates are sufficient. However, we do not change these hyperparameters across the three methods for fair analysis. Note that we do not tune our hyperparameters such as batch size or dropout rate on the entire dataset. Only the first 4-5 images (out of total 91 images) are used to find an acceptable range for these parameters.

Method	PSNR	SSIM
ZSSR (MSE)	33.921	0.943
ZSSR (DoG)	34.222	0.945
ZSSR (DoG)*	34.327	0.946

Table 1: Comparing ZSSR with newly proposed model and DoG loss function.

We observe that adding DoG Pyramid Loss increases both PSNR and SSIM by 0.3 and 0.002 respectively. Similarly doing minibatch training, dropout and tanh activation also slightly improve both the metrics. Figure 2 shows the comparison among generated images using ZSSR and ZSSR*. We typically do not observe significant differences visually among the generated images of these methods.

3.4. Online Super Resolution

This experiment is conducted on the first 30 images of General 100 Dataset and the first 30 images of T91 dataset are considered as external datasets. We evaluate 8 methods which are described below:

- **ZSSR:** Original ZSSR with a batch size of 1. This method is 3-4 times faster than methods using a batch size of 8 and 5-6 times faster than methods optimizing perceptual loss with a batch size of 8.
- **ZSSR*:** This method uses dropout, mini batch training, tanh activation and optimises DoG Loss.
- **ZSSR/online*:** This method is evaluated on General-100 dataset in an online fashion using ZSSR* training scheme.

(a) ZSSR (b) ZSSR Scaled (c) ZSSR* (DoG) (d) ZSSR* (DoG) Scaled (e) HR Image (f) HR Image Scaled

Figure 2: Comparison of images generated using ZSSR and ZSSR* with DoG Loss. Note: Zoom onto the edges of the petals to observe noticeable difference.

- **ZSSR/ext*:**: This method is pretrained on T91 dataset using ZSSR* training scheme in an online fashion and is then evaluated on General-100 dataset without any further training.
- **ZSSR/ext/ftune*:** This method is pretrained on T91 dataset using ZSSR* training scheme in an online fashion and is finetuned (DoG Loss is optimized) on each image of General-100 dataset before evaluation.
- **ZSSR/ext/online*:** This method is pretrained on T91 dataset using ZSSR* training scheme in an online fashion and is further evaluated on General-100 dataset in an online manner where the model constantly keep updating (DoG Loss is optimized).
- **ZSSR+:** This method uses dropout, mini batch training, tanh activation and optimises Perceptual Loss.
- **ZSSR/ext+:** This method is pretrained on T91 dataset using ZSSR+ training scheme in an online fashion and is then evaluated on General-100 dataset without any further training.
- **ZSSR/online+:** This method is evaluated on General-100 dataset in an online fashion using ZSSR+ training scheme.

(a) HR Image (b) ZSSR (c) ZSSR/ext/ftune* (d) ZSSR/ext/ftune+

(e) ZSSR/online* (f) ZSSR/online+ (g) HR Image Scaled (h) ZSSR Scaled

(i) ZSSR/ext/ftune* Scaled (j) ZSSR/ext/ftune+ Scaled (k) ZSSR/online* Scaled (l) ZSSR/online+ Scaled

Figure 3: Comparison of ZSSR, pretrained and online methods for SISR. *ext* represents whether external data (T91) has been used or not.

- **ZSSR/ext/ftune+:** This method is pretrained on T91 dataset using ZSSR+ training scheme in an online fashion and is finetuned (Perceptual Loss is optimized) on each image of General-100 dataset before evaluation.

- **ZSSR/ext/online+:** This method is pretrained on T91 dataset using ZSSR+ training scheme in an online fashion and is further evaluated on General-100 dataset in an online manner where the model constantly keep updating (Perceptual Loss is optimized).

Method	DIL	OSR	PSNR	SSIM
ZSSR	✓	✗	34.927	0.953
*ZSSR**	✓	✗	35.092	0.955
ZSSR+	✓	✗	34.381	0.949
*ZSSR/ext**	✗	✗	34.129	0.947
*ZSSR/online**	✓	✓	35.776	0.959
*ZSSR/ext/ftune**	✓	✗	35.588	0.958
*ZSSR/ext/online**	✓	✓	35.772	0.959
ZSSR/ext+	✗	✗	33.92	0.945
ZSSR/online+	✓	✓	35.348	0.956
ZSSR/ext/ftune+	✓	✗	35.139	0.955
ZSSR/ext/online+	✓	✓	35.322	0.955

Table 2: Comparison of ZSSR, pretrained and online methods for SISR. *ext* represents whether external data (T91) has been used or not. * indicates that DoG Loss is employed for optimisation and + indicates that Perceptual Loss is chosen for optimization. DIL indicates that deep internal learning has been employed on each image and OSR indicates whether model has been tested in an online fashion.

Table 2 shows the comparison among the approaches described above. Note that for all methods training using perceptual loss (methods with '+' in their title), PSNR and SSIM do no correctly exhbit their visual quality. Among ZSSR and ZSSR*, we observe that ZSSR* give around 0.16 improvement in PSNR and 0.002 improvement in SSIM. In comparison, ZSSR/ext* which is pretrained on T91 dataset, and directly evaluated on General 100 dataset without any training do much worse than ZSSR and ZSSR*. Note that the pretrained models are not trained in a supervised manner. Instead, we use the OSR on T91 images to pretrain a model. This shows that while the model trained on one set of images can be transferred onto an unseen set of images, it still doesn't outperform vanilla deep internal learning. This is in agreement to the fact that internal recurrence is much more relevant that external recurrence [1]. We then experiment with ZSSR/ext/ftune* which is pretrained on T91 dataset and for each image the model has been finetuned individually. This means that using both internal and external information is more beneficial than any one alone. ZSSR/online* and ZSSR/ext/online* which do OSR starting with random intiialised model and model pretrained on T91 dataset both perform equally well. They exhibit a further 0.2 improvement in PSNR over ZSSR/ext/ftune*. It is also noted that pretraining on another dataset is not as beneficial as doing OSR on the same dataset. This might be because most of the relevent information common to both datasets can obtained only from a few examples. However, the information present in the same dataset i.e. General 100 is much more relevant since it can be transferred to the incoming images during online training. We observed approximately similar trends are within methods optimizing Perceptual Loss.

Visual Comparison: While PSNR/SSIM are popularly used as metrics for evaluating the task of SISR. These do not correlate very well with the perceptual quality of images. Therefore, it becomes important to qualitatively compare the performance of several methods mentioned in Table 2. Figure 3 shows the original HR Image and the generated SR image from five methods, ZSSR, ZSSR/ext/ftune*, ZSSR/ext/ftune+, ZSSR/online* and ZSSR/online+. We observe that ZSSR is the worst among all of them as shown in the table. ZSSR/online* and ZSSR/online+ are quite similar and comparable to the GT image. The models pretrained on external data and then finetuned on each image perform better than ZSSR and worse than OSR enabled methods.

Runtime Comparison: For fair comparison, we trained each model for the same number of iterations. However as compared to ZSSR, the online models involving weight transfer can quickly converge on the new image. In constrast, ZSSR always trains from scratch requiring a lot of iterations. Therefore OSR using Deep Internal Learning not only give better performance but they quickly adapt to the new image. In both the datasets, the images were not highly correlated (e.g. videos) and this trend is because of underlying weight transfer. This allows a much better initialisation than one used in ZSSR. Typically, OSR methods (ZSSR/online* or ZSSR/online+) both converge in 100-200 iterations. By making the batch size smaller, runtime can be further improved.

4. Future Work

Since Online Super Resolution is better applicable in images that are correlated to each other, videos can show much more improvements over the original ZSSR. Also, ZSSR can incorporate downsampling kernel information which is not yet incorporated in the OSR training. Similar to ZSSR, we can use the kernel information while generating LR-HR image pairs and improve performance on image streams with varying noise levels/artifacts/sensor noise as well. We must also note that OSR methods incorporate deep internal learning and therefore do not require a lot of different images to learn a useful prior. In some cases, it might also be the case that learning some of the final layers can improve the performance of OSR. Nevertheless, our work shows that OSR can be potentially useful over vanilla unsupervised super resolution methods.

References

[1] Michal Irani Assaf Shocher, Nadav Cohen. "zero-shot" super-resolution using deep internal learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015.

[3] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. *CoRR*, abs/1608.00367, 2016.

[4] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[6] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[7] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[9] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *2013 IEEE International Conference on Computer Vision*, pages 945–952, 2013.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[12] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview, 2019.

[13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv:1711.10925*, 2017.

[14] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey. *CoRR*, abs/1902.06068, 2019.

[15] Chih-Yuan Yang, Jia-Bin Huang, and Ming-Hsuan Yang. Exploiting self-similarities for single frame super-resolution. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision – ACCV 2010*, pages 497–510, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[16] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. *CoRR*, abs/1801.07892, 2018.

[17] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018.