“Zero-Shot” Super-Resolution using Deep Internal Learning
236860 Final assignment

Gal Peretz , Adir Rahamim

March 2020

1 Introduction

Super resolution (SR) is the process of enhancing the resolution of an image to create high resolution (HR) version of the low resolution (LR) one. A clear motivation of super resolution is to expose more details from the degraded images, for example understand the license number of a car from the low resolution image created by the zoom in process on the license plate. Deep Learning models, in particular CNN-based models, have proven to be effective for Super Resolution. In the Zero-Shot paper the authors introduce new technique to train simple CNN-based neural network in test time by extracting examples from the input image itself. As opposed to models that train on preprocessed images down sampled with specific kernel from the training data set, the Zero-shot SR model can adapt to different settings for each image thus preform better in a real world situations where the HR-LR relations are unknown.

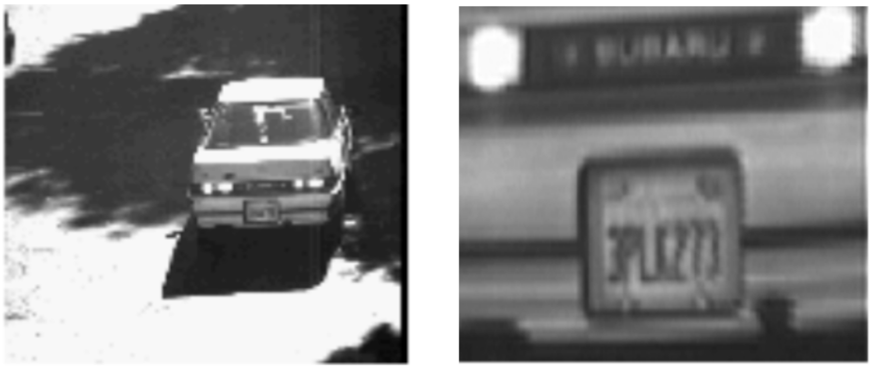


Figure 1: Super resolution technique applied to license plate image after the zoom in process

2 Problem Settings

Let HR be the high resolution image we want to restore from LR the low resolution input. We can define LR = d(HR) where d is the degradation function. If the degradation function was known and invertible we would be able to apply the inverse function on the LR image to find the original HR image. However, usually the inverse function of the degradation function is unknown if exists at all thus we strive to find $\hat{d}$ the best estimation of the inverse of the degradation function.

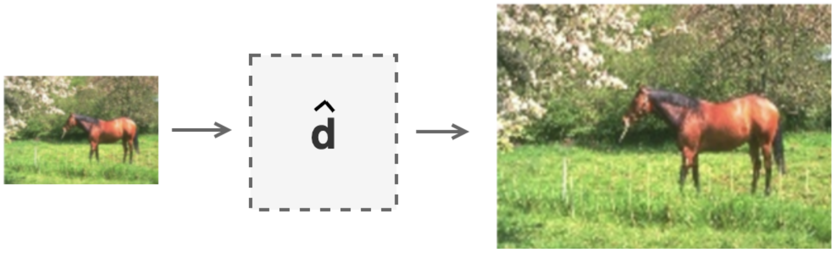


Figure 2: $\hat{d}$ is the best estimation of the inverse degradation function.

3 Related work

Deep learning models have achieved state of the art result biting the previous non deep learning models. most of the non deep learning models based on applying some sort of interpolation (KNN, Bilinear) to patches and as a result enhance the missing pixels by the existing ones.

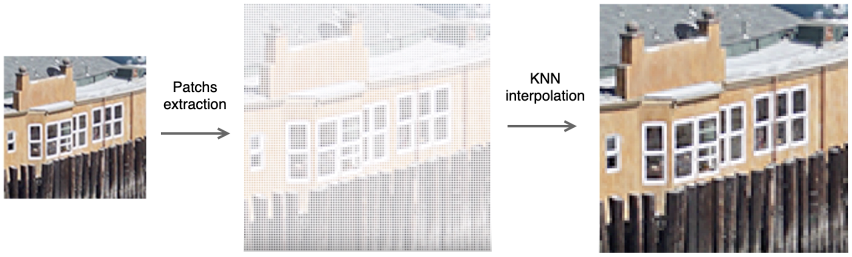


Figure 3: Patches based interpolation

Deep learning based models try to estimate the mapping function between the low resolution image to the high resolution image directly by learning the relations from examples in the training phase. SRCNN [1] is one example of such neural network architecture however, as appose to more recent architectures, SRCNN pipeline still compose of patch extraction phase and then applying features extraction and non-linear mapping using two CNN layers with Relu activation. A more recent approach published in 2016 CVPR called VDSR [4] (Very deep super resolution) uses 20 CNN layers, 64 filters with 3x3 kernel size for each layer. first the LR image is interpolated as ILR image and flows to the network and because the network is deep and there is a clear correlation between the LR image and the HR image the ILR image was added as a residual connection to the last layer to construct the output of the network. In contrast to the upsample interpolation of the input before feed it to the network. EDSR, another model based on deep CNN architecture introduce in [8], takes different approach and uses the network itself to upsample the image from low resolution to high resolution. These models are supervised which were trained exhaustively on external dataset and learn the interpolation kernel of this dataset and as a result perform well on new input that has the same properties as the training images. As appose the supervised methods unsupervised methods in the context of super resolution use only the LR input image to understand the HR output. ZSSR (Zero shot super resolution) is the model suggested in the reviewed paper [7] outperforms SelfExSR [2], another unsupervised based model, by large margin and get a state of the art result among the unsupervised based models.

4 The authors' main contributions

The main contribution of this paper is the introduction of new training procedure. instead of training the SR model on external data the authors suggest to train a substantially smaller model in test time using only the input image, that eliminates the need of collecting external dataset and more importantly it makes the model learn the specific properties of the HR-LR relations of the image that been tested instead of the relations of the images in the external training dataset. the idea of training a simpler model on augmented versions of the input image lead to more robust model for new and unseen images.

5 Model

The ZSSR model suggested in the paper we are reviewing doesn't need external training dataset to understand the LR-HR relations. Instead it uses data augmentation to create training dataset from the input image in the inference stage. to create the training dataset of specific image I at test time the authors downscaled I to many smaller versions of itself $(I_0, I_1, I_2..., I_n)$.

These are called ”HR fathers” because they play the role of the high resolution images in the ”supervised” training. each of the HR fathers downscaled again by the desired scale factor s to obtain the ”LR sons” that could be use as the training dataset. after forming the training dataset the authors enriched it by creating 4 rotated versions $(0°, 90°, 180°, 270°)$ for each LR-HR pair and their mirror reflections in the vertical and horizontal directions. The authors argue that the diversity of the LR-HR relations within single image is small thus the CNN constructed in test time can be much smaller than the CNN models that train on external dataset. The authors used fully convolutional network with 8 layers, each layer has 64 filters and Relu activation function.

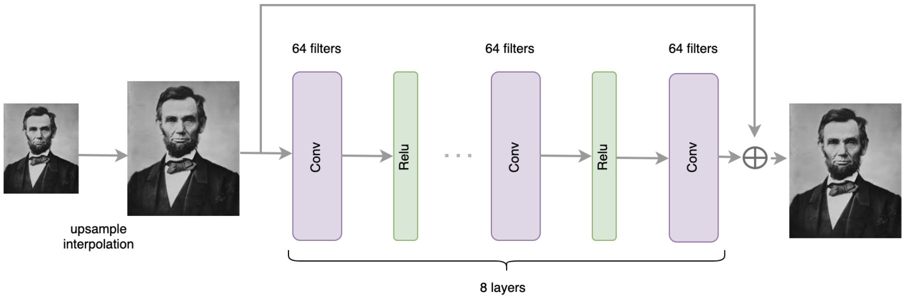


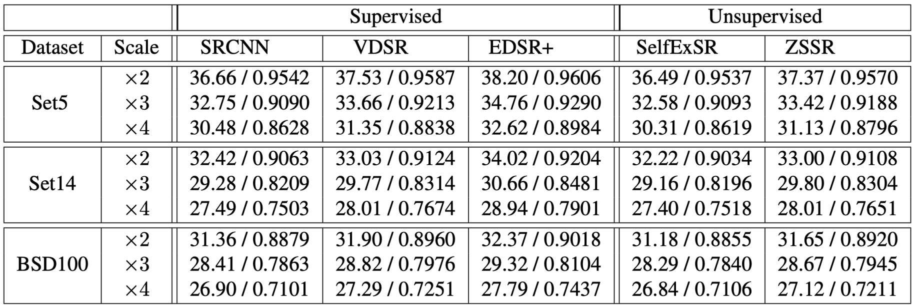
Figure 4: Small CNN model to understand the LR-HR relations in inference stage

6 Training

The authors used the Adam optimizer with starting learning rate of 0.001 and divided it by 10 if some conditions on the std of the reconstruction error have met (stopped when the learning rate is lower than 10^{-6}). In addition, to accelerate the training phase(that happens on inference stage) for each iteration one LR-HR pair is selected in a non-uniform way where the probability of choosing specific LR-HR pair is proportional to the size of the HR image (which gives higher probability to the non-synthesized images) and then applying L1 lose to minimized the reconstruction error between crop of 128x128 instead of the all image. To make the model more robust the authors trained the model gradually with several intermediate scale factors $(s_1, s_2 ..., s_m = s)$ and for each intermediate scale factor they added the downscaled and rotated images that have been created in this stage to the collection of training example to the next training stages. all the images at stage s_i downscaled by this scale factor.

7 Experiments and Results

The authors decided to conduct experiments with two different types of datasets. The first type is the "ideal case" datasets which mean datasets that contain images that have been ideally downscaled using MATLAB's imresize function (with a bicubic kernel). These datasets were separated to train and test sets and the images were downscaled with the same method using the same kernel. In addition, the authors decided to check different scale factors.



		Supervised			Unsupervised	
Dataset	Scale	SRCNN	VDSR	EDSR+	SelfExSR	ZSSR
Set5	×2	36.66 / 0.9542	37.53 / 0.9587	38.20 / 0.9606	36.49 / 0.9537	37.37 / 0.9570
	×3	32.75 / 0.9090	33.66 / 0.9213	34.76 / 0.9290	32.58 / 0.9093	33.42 / 0.9188
	×4	30.48 / 0.8628	31.35 / 0.8838	32.62 / 0.8984	30.31 / 0.8619	31.13 / 0.8796
Set14	×2	32.42 / 0.9063	33.03 / 0.9124	34.02 / 0.9204	32.22 / 0.9034	33.00 / 0.9108
	×3	29.28 / 0.8209	29.77 / 0.8314	30.66 / 0.8481	29.16 / 0.8196	29.80 / 0.8304
	×4	27.49 / 0.7503	28.01 / 0.7674	28.94 / 0.7901	27.40 / 0.7518	28.01 / 0.7651
BSD100	×2	31.36 / 0.8879	31.90 / 0.8960	32.37 / 0.9018	31.18 / 0.8855	31.65 / 0.8920
	×3	28.41 / 0.7863	28.82 / 0.7976	29.32 / 0.8104	28.29 / 0.7840	28.67 / 0.7945
	×4	26.90 / 0.7101	27.29 / 0.7251	27.79 / 0.7437	26.84 / 0.7106	27.12 / 0.7211

Figure 5: Results for the "ideal case"

From this table we can see that ZSSR almost match the result of VDSR and outperform SRCNN on the ideal case however it did not manage to reach the performance of EDSR on the ideal case. In the unsupervised regime ZSSR reached SotA performance and outperforms SelfExSR by large margin ($\sim 1db$).

The authors stated that although they checked their model's performance on standard datasets against known baseline models, ZSSR really shines when the test images are not satisfying specific settings like bicubic kernel for downsampling. This led the authors to modify the BSD100 dataset in order to test a more realistic settings. this "non-ideal" dataset consist images that were downsacled with non-ideal kernels(that deviate from the bicubic kernel) that simulate artifacts and noise due to real world situations like sensor noise, image compression and non-ideal PSF. They created this dataset by randomly degraded the BSD100 images using 3 types of degradations:

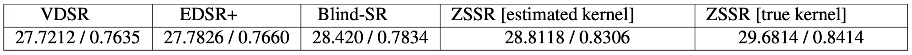
1. Gaussian noise

2. Speckle noise

3. JPEG compression

For the gaussian noise they chose the mean for the x and y axis randomly and a random angle and for the JPEG compression they used MATLAB. For the non-ideal case the authors conducted two types of experiments, the first one is with known kernel for each test image and the second is with unknown kernel.

for the experiment with the unknown kernel the authors compare their models to BlindSR [5] and also used the patches technique to estimate the kernel form the test image.



VDSR	EDSR+	Blind-SR	ZSSR [estimated kernel]	ZSSR [true kernel]
27.7212 / 0.7635	27.7826 / 0.7660	28.420 / 0.7834	28.8118 / 0.8306	29.6814 / 0.8414

Figure 6: Results for the "non-ideal case"

Note that all the supervised models cannot benefit from knowing the downsampling kernel of the test image. overall the ZSSR model outperformed SotA by large margin +1db for the estimated kernel and +2db to the known kernel.

8 Our contribution

After reviewing this paper we credit the authors about the following achievements:

1. The idea of training on augmented versions of the input instead of external data source increases the probability of estimating the correct kernel when dealing with unknown settings like in real world situation.
2. The fact that the convolutions network is small makes the training on inference stage a feasible idea.
3. We feel that the authors did a good job engineering the training phase by adding the probability function of sample with respect to the HR-father size and the gradual training with respect to the scale factor.
4. Overall the paper solution more suitable for real life situations where the downsample kernel is unknown than other methods.

However, we think the following points need to be revisited:

1. To upsample the lower resolution images in the "unknown kernel" case the authors used the non-deep method suggested in [5] and this separated the learning process to two parts. first they estimating the kernel and then they optimizing the ZSSR model with the l1 loss . In general separating the learning process is not a good practice because then the model cannot learn to optimize the pipeline to the general task as a whole.
2. However the main advantage of ZSSR is that it trains on augmented versions of the input on inference time we think the ZSSR model can benefit from "transfer knowledge" from pre-trained network and then fine tune the SR model on the augmented versions of the input.
3. We think it is a good practice to explore more options for the loss function (for example perceptual loss).

We will conduct three experiments to try to mitigate the affects of the issues above.

8.1 Upsample as part of the model

To deal with the first issue we need to review the possible options for upsampling the image. The first option, the one the authors used, is to upsample the image by applying some sort of interpolation before feed the input to the neural network. The other option is to use the model architecture to upsample the image. One option of upsampling during the model forward pass is to use transpose convolution. To use the transpose convolution one should first pad each pixel in the input and then applying convolution to each patch by calculating the convolution of wight matrix with the padded input.

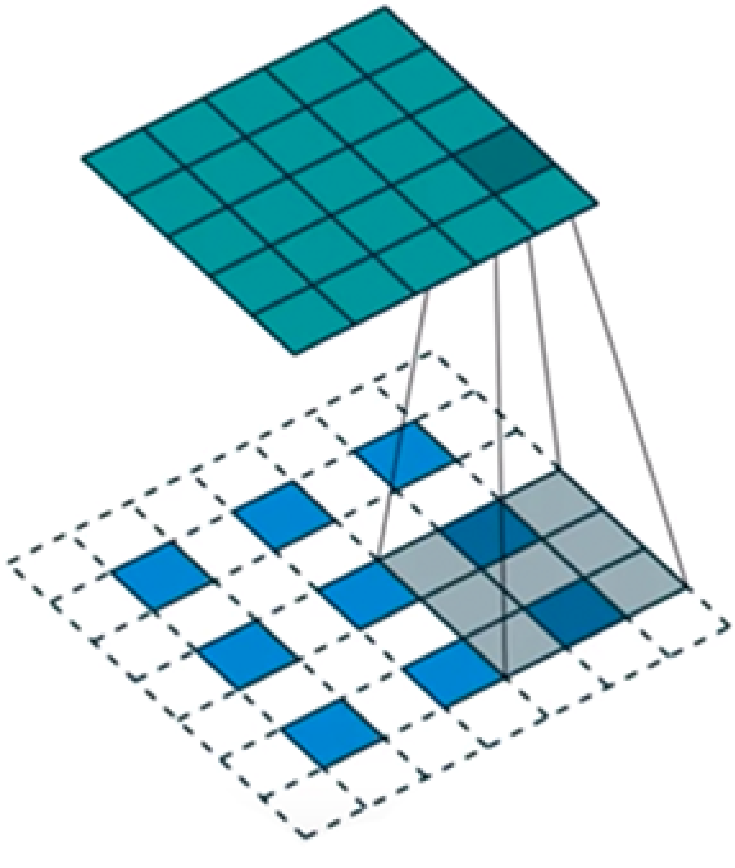


Figure 7: Transpose convolution process

The zero padding creates sparse patches which mean that the only relevant information for the upsample function is the original input pixels because all the surrounding pixels are zeros. In the paper [6] the authors suggest another method to upsample called pixel shuffle. In the pixel shuffle method first one should apply regular convolution layers and then use the activation map of the different channels to create the upsample matrix.

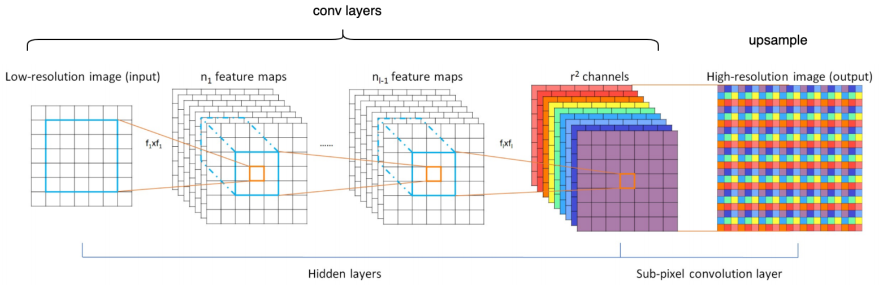


Figure 8: Pixel shuffle process

The main advantage of this method is that it increases the information on the grid before the upsample process. In our case, adding the pixel shuffle layer to the model is easy as appose to implementing the non-deep method suggested in [5], moreover and more importantly it unifies the training process and able us to optimize one model instead of two separate models. Furthermore, in the unknown kernel case this method can represent different upsample function for each input to optimize the overall super resolution task for the specific image. we implemented the suggested model with the pixel shuffle part replacing the non-deep method and tested it on the BSD100 dataset with unknown kernel(we randomly use 4 different kernels to down sample each image) with scale factor of 2 and 4.

Method	PSNR x2	PSNR x4
ZSSR	28.2554	24.2300
ZSSR with PixelShuffle	28.2550	25.0655

we can see that for the x2 super resolution task the pixel shuffle addition does not improve the PSNR value however in the x4 case it does improve the performance of the model by a small amount.

8.2 ZSSR with VDSR as a backbone model

To test the second issue we trained VDSR network on BSDS300 (a different dataset than BSD100) and then tested the combination of the pre-trained VDSR with ZSSR while training only the ZSSR part of the model on the BSD100 dataset.

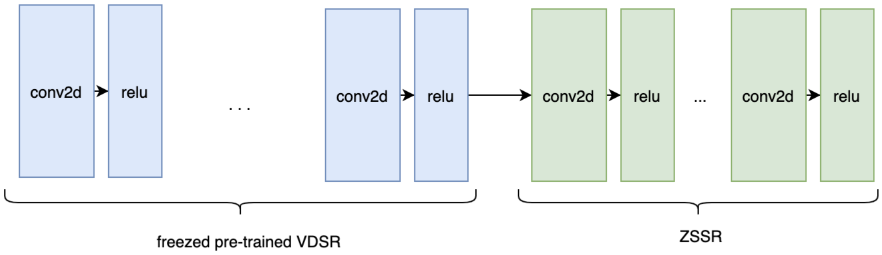


Figure 9: ZSSR with VDSR as backbone

we used the training method of ZSSR which mean training each input on augmented versions of the input.

Method	PSNR / SSIM x2
ZSSR	28.2492 / 0.8931
ZSSR with pre-trained VDSR	27.6091 / 0.8868

unfortunately, the pre-trained VDSR does not improve the performance of the model. it can be because we don't use enough images in the pre-trained task or because the distributions of the inputs between the original task and the pre-trained task are too different this can be due to the fact that we train on smaller images and randomly crop and flip them.

8.3 Integrating perceptual loss to improve SSIM

We implemented the perception loss (e.g content loss) to see if we can get better results when the model also ”knows” the content of the image and not only try to minimize the pixels difference between the predicted high resolution image to the ground truth image. The perception loss suggested in [3] is using pre-trained and freezed VGG network to extract the features in the inner hidden layers and penalize the model if it creates HR image with different features then the original HR image. the loss is L1 loss between the corresponding features of the predicted HR image and the original one.

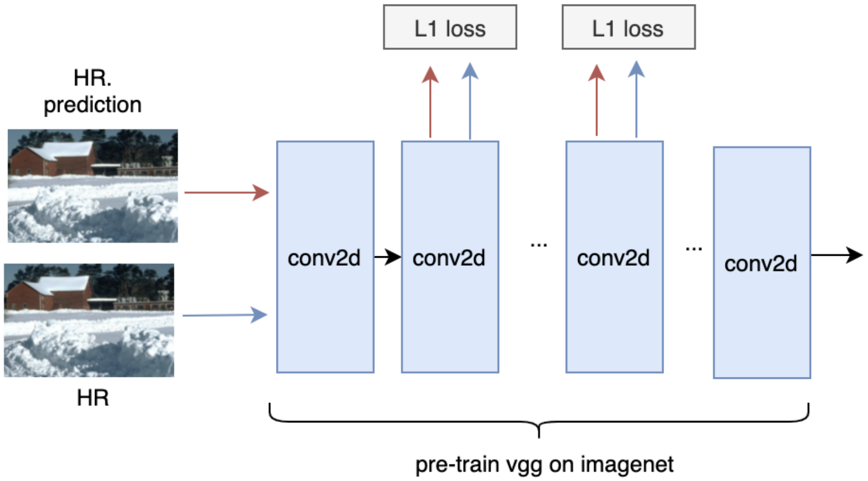


Figure 10: Compute the perceptual loss by weight sum all the L1 loss between the different layers

In order to incorporate the content loss in our setting we resized the low resolution image by the super resolution factor and then compute the content loss on those images for each iteration because the original training data set consist of random crop images that don't contain the original concepts of the image.

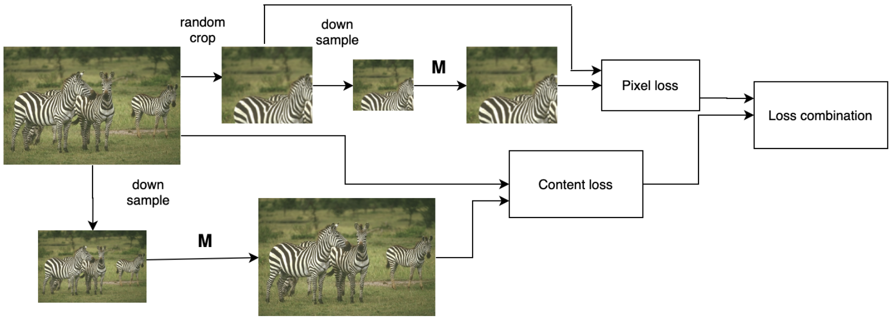


Figure 11: Forward pass with content loss and L1 loss. The M letter marks the computation of the model after the upsample process on the downsampled image

Method	PSNR / SSIM x2
ZSSR with L1 loss	28.2625 / 0.8903
ZSSR with hybrid loss (l1 + content loss)	27.5138 / 0.8938

we can see that although the L1 loss gets a better PSNR values combining this loss with the content loss gives better SSIM values. the SSIM metric represent better the structure of the image than the PSNR metric and that often lead to better visual results.

9 Conclusions

The ZSSR model can get good result in the ideal case when the down sampled kernel is known and great results when it is not known. We can benefit from unifying the upsample process and integrate it in the model architecture as a pixel shuffle layers. We can get better SSIM values combining perceptual loss with the L1 loss.

References

[1] Chao Dong et al. “Learning a Deep Convolutional Network for Image Super-Resolution”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 184–199. ISBN: 978-3-319-10593-2.

[2] Zheng Hui et al. “Lightweight Image Super-Resolution with Information Multi-Distillation Network”. In: *Proceedings of the 27th ACM International Conference on Multimedia*. MM ’19. Nice, France: Association for Computing Machinery, 2019, pp. 2024–2032. ISBN: 9781450368896. DOI: 10 . 1145 / 3343031 . 3351084. URL: https://doi.org/10.1145/3343031.3351084.

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual Losses for Real-Time Style Transfer and Super-Resolution”. In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*. Ed. by Bastian Leibe et al. Vol. 9906. Lecture Notes in Computer Science. Springer, 2016, pp. 694–711. DOI: 10 . 1007 / 978 - 3 - 319 - 46475 - 6 \ _43. URL: https://doi.org/10.1007/978-3-319-46475-6%5C_43.

[4] J. Kim, J. K. Lee, and K. M. Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 1646–1654. DOI: 10.1109/CVPR.2016.182.

[5] T. Michaeli and M. Irani. “Nonparametric Blind Super-resolution”. In: *2013 IEEE International Conference on Computer Vision*. Dec. 2013, pp. 945–952. DOI: 10.1109/ICCV.2013.121.

[6] Wenzhe Shi et al. “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 1874–1883. DOI: 10 . 1109 / CVPR . 2016 . 207. URL: https://doi.org/10.1109/CVPR.2016.207.

[7] Assaf Shocher, Nadav Cohen, and Michal Irani. ““zero-shot” super-resolution using deep internal learning”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3118–3126.

[8] Haimin Wang et al. “Deep Residual Network for Single Image Super-Resolution”. In: *Proceedings of the 2nd International Conference on Control and Computer Vision*. ICCCV 2019. Jeju, Republic of Korea: Association for Computing Machinery, 2019, pp. 66–70. ISBN: 9781450363228. DOI: 10 . 1145 / 3341016 . 3341030. URL: https://doi.org/10.1145/3341016.3341030.