Super-resolution

Sebastien Strban
Faculty of Computer and Information Science
Email: ss4774@student.uni-lj.si

I. Abstract

Many breakthroughs in speed and accuracy of single image super-resolution (SISR) have been achieved. One of the biggest challenges is how to recover finer texture details when super-resolution is applied at large upscaling factors. A typical solution to SISR involves using a convolutional neural network (CNN), however new approaches using a generative adversarial network (GAN) are now also popular. The behavior of optimization-based super-resolution methods is principally driven by the choice of the objective function. In this work, we present an evaluation of SRResNet and SRGAN. SRResNet is a deep residual network and SRGAN is a generative adversarial network for image super-resolution (SR). SRResNet is able to recover reasonable quality photo-realistic textures from heavily downsampled images. SRGAN is capable of inferring photo-realistic natural images with high upscaling factors. This is achieved using a perceptual loss function which consists of an adversarial loss and a content loss. The adversarial loss pushes the solution to the natural image manifold using a discriminator network that is trained to differentiate between the super-resolved images and original photo-realistic images. The content loss is motivated by perceptual similarity instead of similarity in pixel space.

II. Introduction

The challenging task of estimating and predicting a high-resolution (HR) image from its low-resolution (LR) counterpart is referred to as super-resolution (SR). SR received substantial attention from within the computer vision research community and has a wide range of applications [1], [2], [3].

The optimization target of a supervised SR algorithms is typically the minimization of the mean squared error (MSE) between the recovered HR image and the ground truth. This is convenient as minimizing MSE also maximizes the peak signal-to-noise ratio (PSNR), which is a common measure used to evaluate and compare SR algorithms [4]. However, the ability of MSE to capture perceptually important differences, such as high level texture detail, is very limited as they are defined based on pixel-wise image differences [5], [6], [7].

In this work we present an evaluation of a super-resolution deep residual network with skip-connections (SRResNet) and super-resolution generative adversarial network (SRGAN) which also employs the use of SRResNet. SRResNet uses the MSE as the primary optimization target, while SRGAN is able to diverge from MSE as the sole optimization target and improve on performance and visual quality of results. New perceptual loss is also defined which uses high-level feature maps of the VGG network [8], [9], [10] in combination with a discriminator that encourages solutions perceptually hard to distinguish from the HR reference images.

A. Related work

1) Image super-resolution: Some of the recent overview articles on image SR include Nasrollahi and Moeslund [3] or Yang et al. [4]. In our analysis we are focusing on single image super-resolution (SISR). However, there exist many approaches that recover HR images from multiple images [11], [12].

One of the first methods to tackle SISR were prediction-based methods. These are usually filtering approaches, such as linear, bicubic or Lanczos [13] filtering. Mentioned approaches can be very fast, however they oversimplify the SISR problem and usually yield solutions with overly smooth texture results. Methods that put particularly focus on edge-preservation have also been proposed [14], [15].

More powerful and advanced approaches aim to establish a complex mapping between low-resolution and high-resolution image information. These approaches usually rely on the training data. Many other methods, which are based on example-pairs, rely on LR training patches for which the corresponding HR counterparts are known. Early work was done and presented by Freeman et al. [16], [17].

Recently, convolutional neural network (CNN) based SR algorithms have shown excellent performance. In Wang et al. [18] the authors encode a sparse representation and use their feed-forward network architecture which is based on the learned iterative shrinkage and threshold algorithm [19]. In Dong et al. [20], [21] authors used bicubic interpolation to upscale an input image and trained a three layer deep fully convolutional network in an end-to-end fashion to achieve good performance in SR.

2) Loss functions: It is notable that pixel-wise loss functions (such as MSE) struggle to handle the uncertainty which is inherent in recovering high-frequency details (e.g. a texture). Minimizing MSE encourages finding of pixel-wise averages of plausible solutions which are generally smooth and thus give us poor perceptual quality [10], [22], [9]. The problem of minimizing MSE is illustrated in figure 1, where multiple potential solutions, with high texture details, are averaged to create a smooth reconstruction.

In Mathieu et al. [22] and Denton et al. [23] the authors tackled this problem by employing generative adversarial networks (GANs) for the application of image generation.

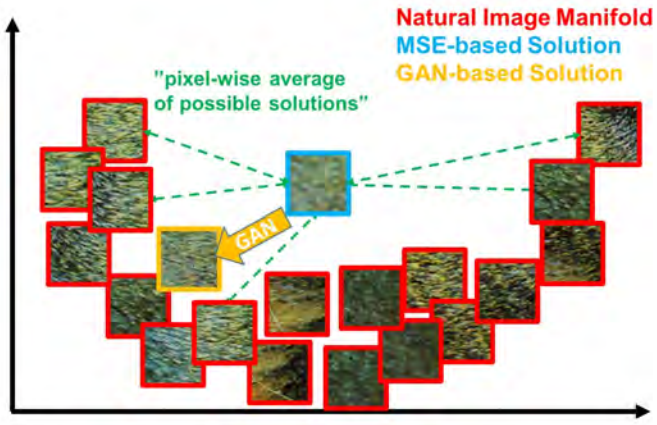


Fig. 1. Depiction of image patches from the natural image manifold (red), super-resolved image patches obtained with MSE (blue) and super-resolved image patches obtained with GAN (orange). The MSE-based solution appears overly smooth due to the pixel-wise average of possible solutions in the pixel space, while GAN-based solution drives the reconstruction towards the natural image manifold producing perceptually more convincing solutions.

Work of Dosovitskiy and Brox [24] uses loss functions based on Euclidean distances computed in the feature space of neural networks in combination with adversarial training. It is shown that the proposed loss allows visually superior image generation and can be used to solve the ill-posed inverse problem of decoding nonlinear feature representations. Similarly, Johnson et al. [9] and Bruna et al. [10] propose the use of features extracted from a pre-trained VGG network instead of low-level pixel-wise error measures. The authors formulate a loss function, which is based on the euclidean distance between feature maps that are extracted from the VGG19 network. Perceptually more convincing results were obtained for both super-resolution and artistic style-transfer [25], [26].

B. Contribution

GANs provide a powerful way for generating plausible-looking natural images which give us high perceptual quality. The GAN procedure encourages the reconstructions to move towards regions of the search space with high probability of containing photo-realistic images. This brings us closer to the natural image manifold as shown in figure 1.

In this work we describe and analyze the first very deep ResNet architecture using the concept of GANs to form a perceptual loss function for photo-realistic SISR [27]. The key aspects of this work are:

- a new state of the art for image SR with high upscaling factors ($4\times$) as measured by PSNR and structural similarity (SSIM) with 16 blocks deep ResNet (SRResNet) optimized for MSE;
- proposed SRGAN which is a GAN-based network optimized for a new perceptual loss. Here the MSE-based content loss was replaced with a loss calculated on feature maps of the VGG network, which are more invariant to changes in pixel space [28].

We describe the network architecture and the perceptual loss in section III. An evaluation is provided in section IV. The work concludes with section V

III. Method

The aim of SISR is to create a super-resolved image I^{SR}, which is an approximation of a high-resolution image I^{HR} from a low-resolution input image I^{LR}. Here I^{LR} is the low-resolution image version of its high-resolution image counterpart I^{HR}. The high-resolution images are only available during training. In training, I^{LR} is obtained by applying a Gaussian filter to I^{HR} followed by a downsampling operation with downsampling factor r. For an image with C color channels, we describe I^{LR} by a real-valued tensor of size $W \times H \times C$, I^{HR} by $rW \times rH \times C$ and I^{SR} by $rW \times rH \times C$.

The ultimate goal is to train a generating function G that estimates for a given LR input image its corresponding HR counterpart. This is achieved by training a generator network as a feed-forward CNN G_{θ_G} parameterized by θ_G. Here $\theta_G = \{W_{1:L}; b_{1:L}\}$ denotes the weights and biases of a L-layer deep network and is obtained by optimizing a SR-specific loss function l^{SR}. For training images $I_n^{HR}, n = 1, ..., N$ with corresponding $I_n^{LR}, n = 1, ..., N$, we solve:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR}) \tag{1}$$

In this work we will use perceptual loss l^{SR} which is specifically designed as a weighted combination of several loss components that model distinct desirable characteristics of the recovered SR image.

1) Adversarial network architecture: A discriminator network D_{θ_D} is defined and based on Goodfellow et al. [29] which is optimized in an alternating manner along with G_{θ_G} to solve the adversarial min-max problem:

$$\begin{aligned} \min_{\theta_G} \min_{\theta_D} & \mathbb{E}_{I^{HR} \sim p_{train}(I^{HR})}[log D_{\theta_D}(I^{HR})] + \\ & \mathbb{E}_{I^{LR} \sim p_G(I^{LR})}[log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))] \end{aligned} \tag{2}$$

The general idea behind this formulation is that it allows one to train a generative model G with the goal of fooling a differentiable discriminator D that is trained to distinguish super-resolved images from real images. With this approach this generator can learn to create solutions that are highly similar to real images and thus difficult to classify by discriminator D. This encourages the manifestation of perceptually superior solutions that reside in the subspace of natural images (the manifold). This approach is in contrast to SR solutions obtained by minimizing pixel-wise error measurements, such as the MSE.

The generator architecture is illustrated in figure 2 and the discriminator architecture in figure 3. At the core of the very deep generator network G are B residual blocks with identical layout. Two convolutional layers are used with small 3×3 kernels and 64 feature maps followed by batch-normalization layers and ParametricReLU as the activation function. The resolution of the input image is increased with two trained sub-pixel convolution layers.

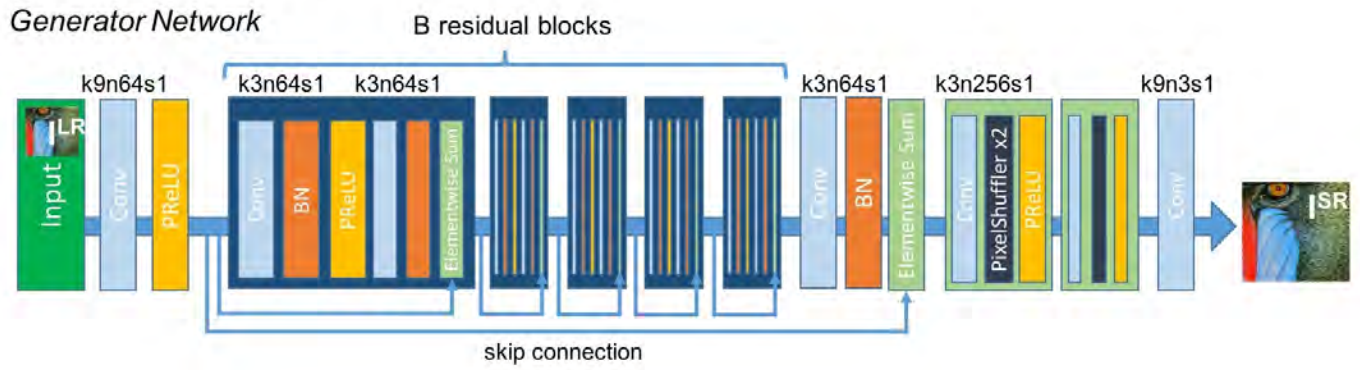


Fig. 2. Architecture of generator network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

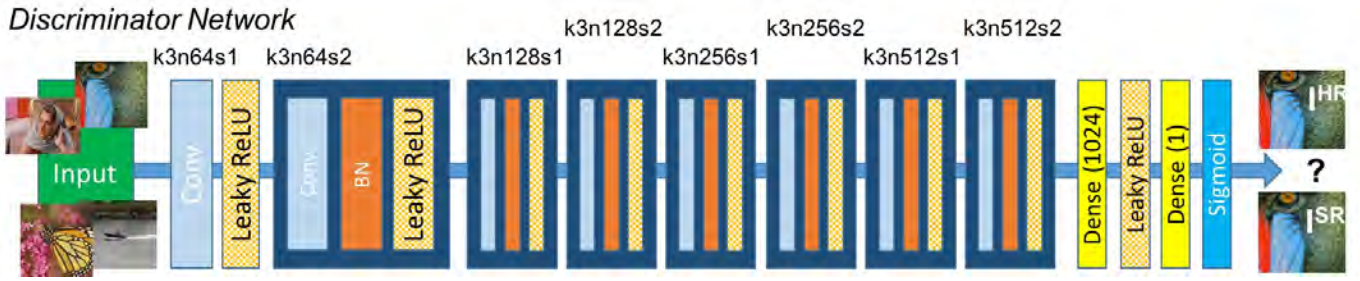


Fig. 3. Architecture of discriminator network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

To discriminate real HR images from generated SR samples a discriminator network is trained. LeakyReLU activation is used ($\alpha = 0.2$) and max-pooling is avoided throughout the network. The discriminator network is trained to solve the maximization problem in Equation III-1. It contains 8 convolutional layers with an increasing number of 3×3 filter kernels, increasing by a factor of 2 from 64 to 512 kernels. Strided convolutions are used to reduce the image resolution each time the number of features is doubled. The resulting 512 feature maps are followed by two dense layers and a final sigmoid activation function to obtain a probability for sample classification.

A. *Perceptual loss function*

The definition the perceptual loss function l^{SR} is critical for the performance of the generator network. This formulation of the perceptual loss is as the weighted sum of a content loss (l^{SR}_X) and an adversarial loss (l^{SR}_{Gen}) component as:

$$l^{SR} = l^{SR}_X + 10^{-3} l^{SR}_{Gen} \tag{3}$$

1) Content loss: The pixel-wise MSE loss is calculated as:

$$l^{SR}_{MSE} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2 \tag{4}$$

This is one of the most widely used optimization target for image SR on which many state-of-the-art approaches rely [20], [30]. However, while achieving particularly high PSNR, solutions of MSE optimization problems often lack high-frequency content which results in perceptually unsatisfying and unpleasing solutions with overly smooth texture details.

Better choice would be to use a loss function that is closer to perceptual similarity. The VGG loss is defined based on the ReLU activation layers of the pre-trained 19 layer VGG network.

With $\phi_{i,j}$ we indicate the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network.The VGG loss is then defined as the euclidean distance between the feature representations of a reconstructed image $G_{\theta_G}(I^{LR})$ and the reference image I^{HR}:

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \tag{5}$$

Here $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the respective feature maps within the VGG network.

2) Adversarial loss: In addition to the content losses described so far, the generative component of this GAN is also added to the perceptual loss. This encourages our network to favor solutions that reside on the manifold of natural images, when trying to fool the discriminator network.

The generative loss l^{SR}_{Gen} is defined based on the probabil-

ities of the discriminator $D_{\theta_D}(G_{\theta G}(I^{LR}))$ over all training samples as:

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -logD_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{6}$$

Here, $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is the probability that the reconstructed image $G_{\theta_G}(I^{LR})$ is a natural HR image. For better gradient behavior we minimize $-logD_{\theta_D}(G_{\theta_G}(I^{LR}))$ instead of $log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))$ [29].

IV. Experiments and evaluation

Training is done on the COCO training set 2014 and validation is performed with COCO validation set 2014. Testing and evaluation of experiments is performed on four widely used public benchmark datasets: COCO testing set 2014, Set5, Set14 and BSD100, the testing set of BSD300.

All experiments are performed with a scale factor of $4\times$ between low-resolution and high-resolution images. This corresponds to a $16\times$ reduction in number of image pixels.

A. Base model

The base model follows the architectural implementation of original model as described in [27] and is used as a base for other evaluations.

The base model uses Adam as an optimizer with $\beta_1 = 0.9$ The SRResNet network is trained with a learning rate of 10^{-4} and 10^6 update iterations, which yield 200 epochs in total on our training dataset. The trained SRResNet network is based on MSE loss and in used as initialization for generator when training the actual GAN to avoid undesired local optima. The SRGAN network is based on perceptual loss and trained with 10^{-5} update iterations at a learning rate of 10^{-4} and another 10^5 iterations at a lower learning rate of 10^{-5}, which yield 40 epochs in total on our training dataset Updates to the generator and discriminator network are alternating ($k = 1$). The generator network has 16 identical residual blocks ($B = 16$).

The evaluation results of the base model are presented in table I. We can observe that SRResNet gives better numerical results across the all testing datasets. This phenomenon is also observer in the original paper [27].

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet	**0.0219**	**27.3166**	**0.7766**	**0.0084**	**30.6233**	**0.8899**
SRGAN	0.1938	25.3691	0.6998	0.2755	27.6378	0.8098
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet	**0.0116**	**28.5642**	**0.7977**	**0.0140**	**27.3658**	**0.7509**
SRGAN	0.1935	25.9921	0.7149	0.1802	25.1075	0.6653

TABLE I
Evaluation results of the base model.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet can be seen on figure 4. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN can be seen on figure 5. We can observe a steady decrease in loss for both SRResNet and SRGAN. There is also a slight gap between validation and train set. For SRResNet it looks like a case of overfitting.

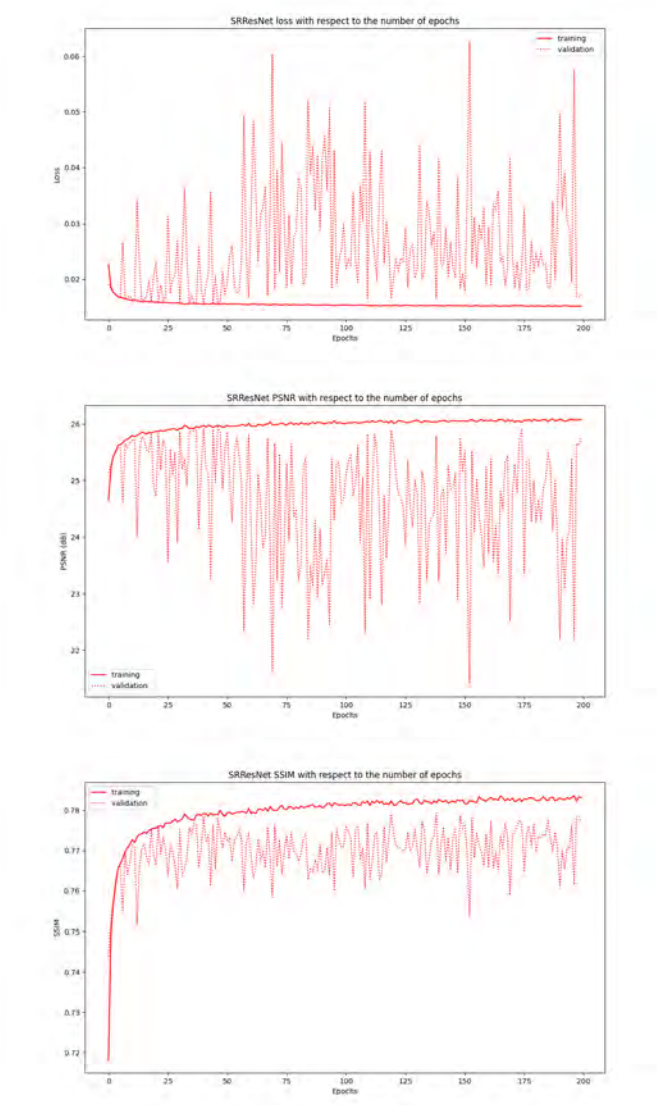


Fig. 4. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for the base model.

On figure 6 we present an example of super-resolved image. We can see that bot, SRResNet and SRGAN, give superior results compared to bicubic upsampling. Although SRResNet gives better results than plan bicubic upsampling, the final result is overly smooth. SRGAN give much more natural results and is also able to create fine texture details.

Similar results can be see in example on figure 7. SRResNet gives overly smooth yet better result than plain bicubic upsampling. SRGAN on the other hand presents us with much finer detail (note the grass on the ground and skin of the elephants).

Sometimes there is little difference between SRResNet and SRGAN. This can be partially observed on figure 8. Due to the smooth nature of the texture on the bananas, there is little difference between SRResNet and SRGAN. Note that the perceptually convincing reconstruction of text on the sticker is not well realized.

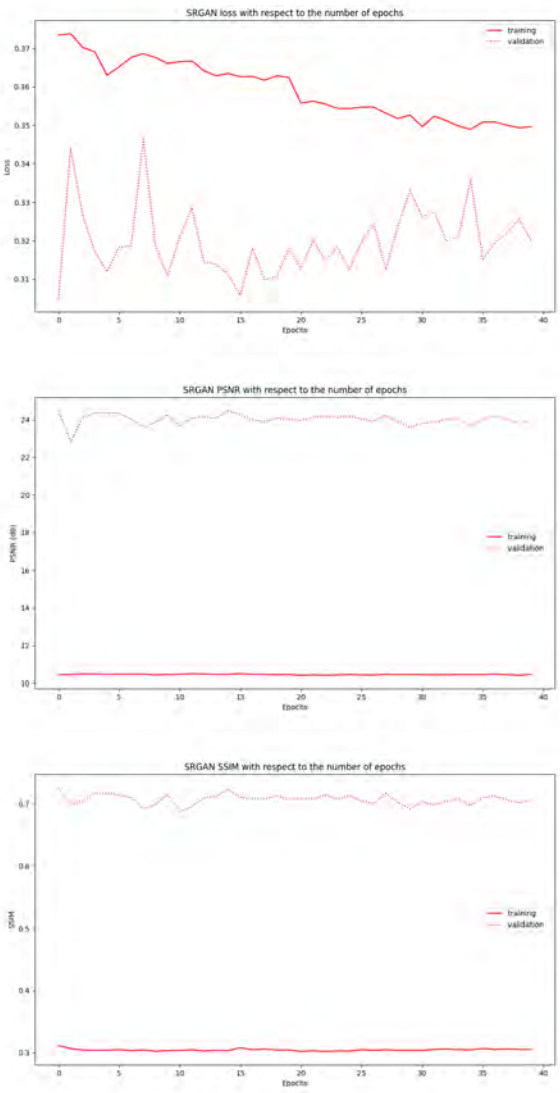


Fig. 5. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for the base model.

Final example is presented on figure 9. Similarly to previous examples SRResNet gives slightly smoother results than SRGAN. This time one might argue that SRResNet gives more pleasing results when compared to SRGAN.

B. Epochs

Here we explore the effect of increasing number of epochs during the training phase.

The evaluation results of the increasing number epochs are presented in table II. We can observe that SRResNet gives better numerical results across the all testing datasets.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet can be seen on figure 10. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN can be seen on figure 11. We can observe a steady decrease in loss for both SRResNet and SRGAN. We also see that increasing the number of epochs helped in improving the model and diminishing the gap between training and validation losses.

Fig. 6. An example of super-resolved image from Set14. Top left is LR image, top right is SR image using SRResNet, bottom left is SR image using SRGAN, bottom right is original image in HR.

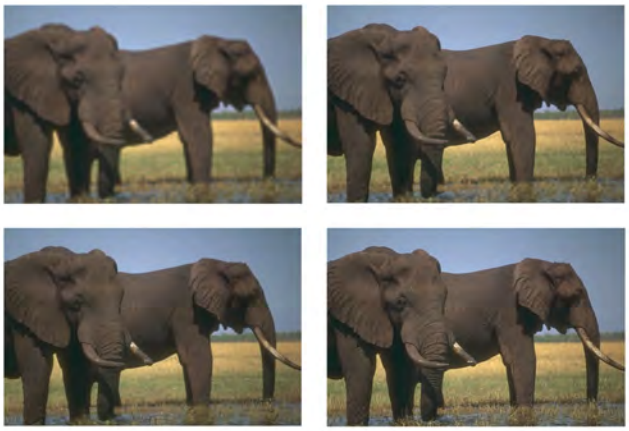


Fig. 7. An example of super-resolved image from BSD100. Top left is LR image, top right is SR image using SRResNet, bottom left is SR image using SRGAN, bottom right is original image in HR.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet	**0.0183**	**27.2895**	**0.7784**	**0.0088**	**30.1063**	**0.8914**
SRGAN	0.1945	25.2257	0.7048	0.2776	27.0301	0.8204
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet	**0.0114**	**28.5675**	**0.7990**	**0.0153**	**27.1376**	**0.7491**
SRGAN	0.1834	25.9390	0.7180	0.1787	25.1277	0.6666

TABLE II
EVALUATION RESULTS OF INCREASED NUMBER OF EPOCHS.

Fig. 8. An example of super-resolved image from COCO test 2014. Top left is LR image, top right is SR image using SRResNet, bottom left is SR image using SRGAN, bottom right is original image in HR.

Fig. 9. An example of super-resolved image from Set5. Top left is LR image, top right is SR image using SRResNet, bottom left is SR image using SRGAN, bottom right is original image in HR.

C. Batch size

Here we explore the effect of the different batch sizes used during the training phase.

The evaluation results of the different batch sizes are presented in table III. We can observe that SRResNet, with a batch size of 16 gives overall best numerical results across the all testing datasets.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet can be seen on figure 12. Loss, PSNR and SSIM

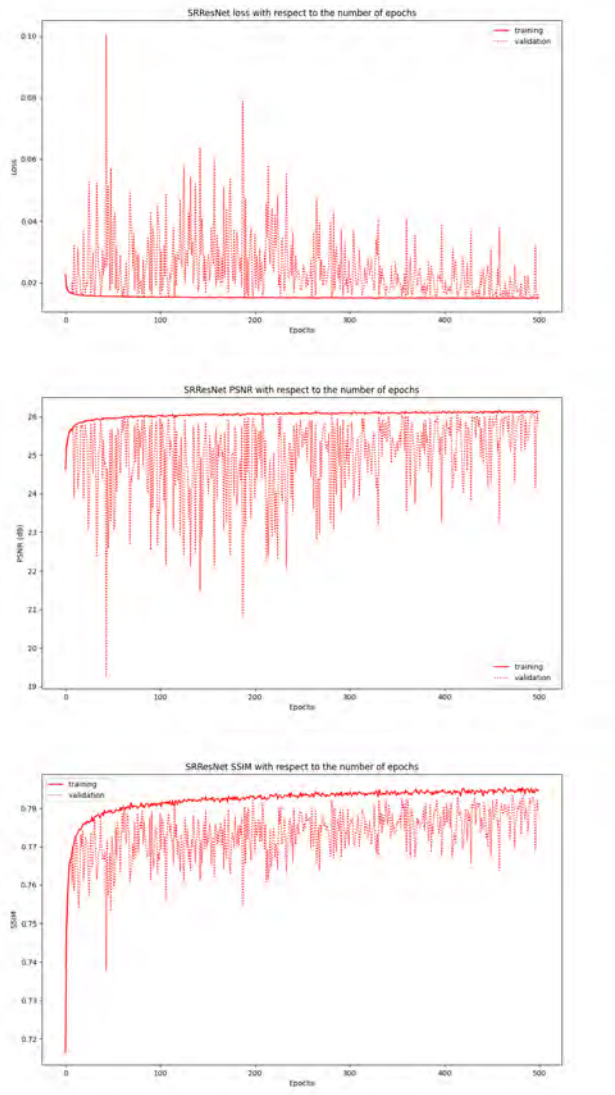


Fig. 10. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for increased number of epochs.

with respect to the number of epochs for SRGAN can be seen on figure 13. We can observe a steady decrease in loss for both SRResNet and SRGAN. The lower number of batch sizes seem to be beneficial for SRResNet loss and PSNR, while SRResNet SSIM seems best at around 16. Best SRGAN loss and SSIM is achieved with larger batch sizes, while the PSNR looks best with smaller batch sizes.

D. Optimizer learning rate

Here we explore the effect of the different optimizers and different learning rates used during the training phase.

The evaluation results of the Adam optimizer and different learning rates are presented in table IV. We can observe that using Adam optimizer with learning rate 10^{-4} yields best results.

Loss, PSNR and SSIM with respect to the number of

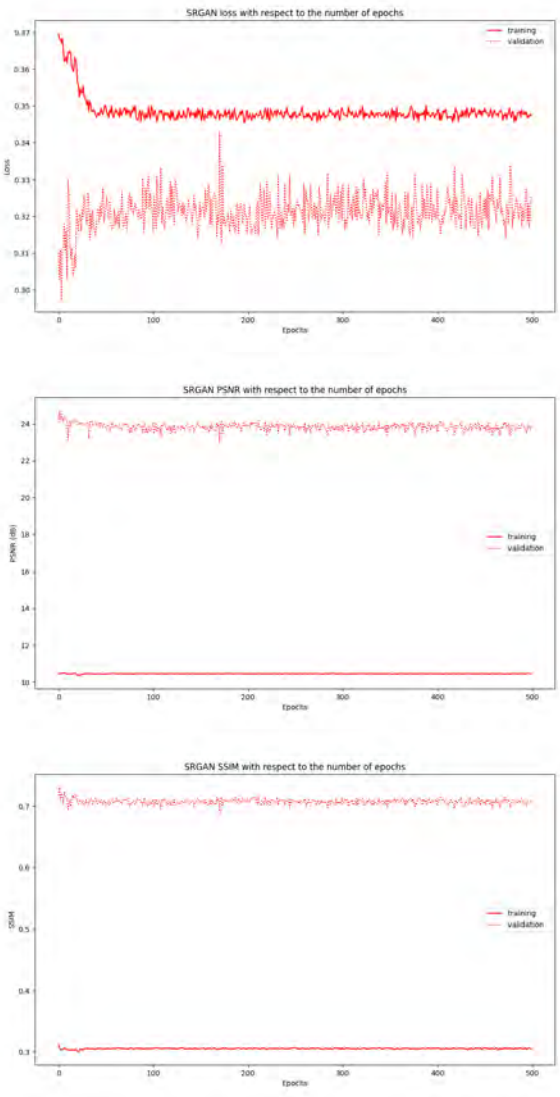


Fig. 11. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for increased number of epochs.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, 1	0.0143	27.8191	0.7782	0.0050	31.6714	0.8969
SRGAN, 1	0.2420	19.6681	0.6479	0.2208	28.5091	0.8620
SRResNet, 2	0.0139	27.9936	0.7858	0.0048	31.7809	0.9004
SRGAN, 2	0.2865	16.1535	0.5405	0.2465	24.8239	0.8378
SRResNet, 4	0.0137	28.0711	0.7868	**0.0046**	31.9375	0.9015
SRGAN, 4	0.3593	15.7794	0.5945	0.4470	13.3654	0.6416
SRResNet, 8	0.0137	28.0657	0.7877	**0.0046**	**31.9493**	0.9017
SRGAN, 8	0.2156	22.9547	0.6767	0.3313	20.6688	0.7466
SRResNet, 16	**0.0136**	**28.1025**	**0.7883**	**0.0046**	31.9345	**0.9023**
SRGAN, 16	0.1905	25.8082	0.7057	0.2529	29.1558	0.8324
SRResNet, 32	0.0276	27.0769	0.7799	**0.0046**	31.9011	0.9007
SRGAN, 32	0.1945	25.3633	0.7006	0.2484	29.0209	0.8382
SRResNet, 64	0.0137	28.0807	0.7873	0.0047	31.9276	0.9015
SRGAN, 64	0.1850	25.7564	0.7086	0.2374	29.1578	0.8449
SRResNet, 128	0.0137	28.0664	0.7872	0.0047	31.9168	0.9015
SRGAN, 128	0.1925	25.5932	0.7023	0.2397	29.1346	0.8411
SRResNet, 256	0.0139	28.0023	0.7852	0.0048	31.8391	0.9005
SRGAN, 256	0.1862	25.4428	0.7020	0.2419	28.9599	0.8410
SRResNet, 512	0.0141	27.9223	0.7832	0.0049	31.7255	0.8993
SRGAN, 512	0.1833	25.4843	0.7029	0.2449	29.1373	0.8444
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, 1	0.0118	28.4080	0.7926	0.0135	27.3937	0.7475
SRGAN, 1	0.2027	22.6563	0.7173	0.2393	20.1547	0.6132
SRResNet, 2	0.0115	28.5317	0.7977	0.0132	27.5089	0.7539
SRGAN, 2	0.2389	20.2460	0.6603	0.2649	18.7091	0.5400
SRResNet, 4	0.0115	28.5848	0.7980	**0.0130**	27.5693	0.7546
SRGAN, 4	0.2629	20.4511	0.6664	0.3252	19.7112	0.6073
SRResNet, 8	**0.0114**	28.6114	0.7989	**0.0130**	27.5384	0.7553
SRGAN, 8	0.1932	24.6153	0.7041	0.1901	24.2988	0.6561
SRResNet, 16	**0.0114**	28.6075	**0.7993**	**0.0130**	**27.5998**	**0.7559**
SRGAN, 16	0.1793	26.2713	0.7173	0.1781	25.3709	0.6676
SRResNet, 32	0.0119	28.5291	0.7982	0.0132	27.5172	0.7550
SRGAN, 32	0.1857	25.9926	0.7064	0.1851	25.1500	0.6614
SRResNet, 64	**0.0114**	**28.6219**	0.7989	**0.0130**	27.5743	0.7551
SRGAN, 64	0.1781	25.8796	0.7043	0.1775	24.9667	0.6583
SRResNet, 128	0.0115	28.6023	0.7987	0.0131	27.5668	0.7551
SRGAN, 128	0.1808	25.7765	0.6992	0.1882	24.8520	0.6498
SRResNet, 256	0.0116	28.5298	0.7967	0.0131	27.5262	0.7536
SRGAN, 256	0.1782	25.6421	0.7005	0.1818	24.5470	0.6457
SRResNet, 512	0.0118	28.4319	0.7953	0.0133	27.4700	0.7523
SRGAN, 512	0.1780	25.7628	0.7017	0.1787	24.4982	0.6435

TABLE III
EVALUATION RESULTS OF THE DIFFERENT BATCH SIZES.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, 1e-1	1.4228	13.1536	0.4244	1.5209	12.6921	0.3888
SRGAN, 1e-1	0.2787	12.1388	0.4128	0.3264	12.1870	0.3806
SRResNet, 1e-2	0.4239	12.7827	0.4007	0.3967	13.1758	0.4630
SRGAN, 1e-2	0.2787	12.1388	0.4128	0.3264	12.1870	0.3806
SRResNet, 1e-3	**0.0138**	28.0172	**0.7867**	**0.0047**	31.8344	0.9008
SRGAN, 1e-3	0.1995	25.3966	0.6964	0.2556	28.8420	0.8385
SRResNet, 1e-4	0.0153	**28.0184**	0.7862	**0.0047**	**31.9811**	**0.9012**
SRGAN, 1e-4	0.1969	24.8958	0.6969	0.2872	26.3803	0.8078
SRResNet, 1e-5	0.0141	27.9295	0.7833	0.0049	31.7169	0.8990
SRGAN, 1e-5	0.2128	24.4837	0.7093	0.3560	24.9329	0.8307
SRResNet, 1e-6	0.0155	27.4038	0.7657	0.0060	30.9854	0.8848
SRGAN, 1e-6	0.2115	24.3678	0.7325	0.3368	23.5866	0.8350
SRResNet, 1e-7	0.0179	26.6381	0.7394	0.0095	29.5271	0.8458
SRGAN, 1e-7	0.1977	25.7059	0.7565	0.2653	26.5974	0.8601
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, 1e-1	1.3285	13.8049	0.3816	1.4595	14.6831	0.3827
SRGAN, 1e-1	0.2395	12.4859	0.3669	0.2602	13.7321	0.3799
SRResNet, 1e-2	0.4709	12.1461	0.3631	0.4628	12.3999	0.3143
SRGAN, 1e-2	0.2395	12.4859	0.3669	0.2602	13.7321	0.3799
SRResNet, 1e-3	**0.0114**	28.6002	0.7989	0.0131	27.5206	0.7552
SRGAN, 1e-3	0.1925	25.5104	0.6952	0.1921	24.9167	0.6557
SRResNet, 1e-4	**0.0114**	**28.6433**	**0.7995**	**0.0130**	**27.5729**	**0.7557**
SRGAN, 1e-4	0.1941	25.6344	0.7101	0.1829	24.9255	0.6595
SRResNet, 1e-5	0.0120	28.3930	0.7942	0.0132	27.4897	0.7525
SRGAN, 1e-5	0.2106	25.9821	0.7258	0.1926	24.9634	0.6649
SRResNet, 1e-6	0.0129	27.9709	0.7810	0.0141	27.1546	0.7394
SRGAN, 1e-6	0.2056	26.2074	0.7590	0.1858	25.2086	0.6983
SRResNet, 1e-7	0.0159	26.9259	0.7522	0.0159	26.5549	0.7179
SRGAN, 1e-7	0.1812	27.3106	0.7838	0.1804	26.2667	0.7324

TABLE IV
EVALUATION RESULTS OF ADAM OPTIMIZER AT DIFFERENT LEARNING RATES.

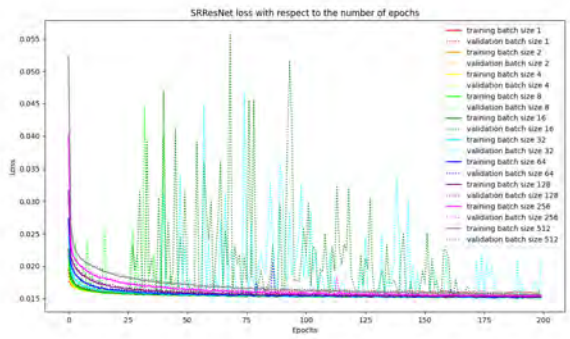
epochs for SRResNet using Adam optimizer can be seen on figure 14. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN using Adam optimizer can be seen on figure 15. We can confirm that using Adam optimizer with learning rate 10^{-4} yields best results for both SRResNet and SRGAN. Increasing the learning rate has a negative impact on performance, while decreasing the learning rate has a positive impact on performance.

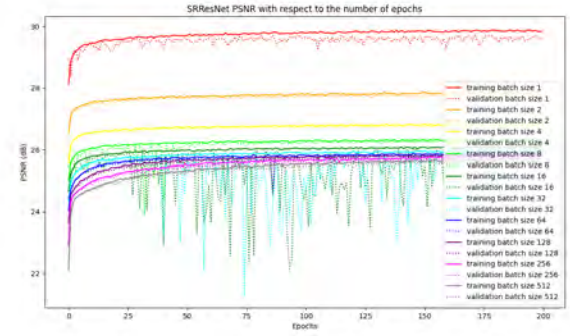
The evaluation results of the SGD optimizer and different learning rates are presented in table V. We can observe that using SGD optimizer with learning rate 10^{-1} yields best results on all testing datasets.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet using SGD optimizer can be seen on figure 16. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN using SGD optimizer can be seen on figure 17. We can confirm that using SGD optimizer with learning rate 10^{-1} yields best results for SRResNet. Here decreasing the learning rate has a negative impact on performance, while increasing the learning rate has a positive impact on performance. Similarly we can say about SRGAN, with an exception of learning rate 10^{-2}, which gives poor results.

The evaluation results of the ADAGRAD optimizer and different learning rates are presented in table VI. We can observe that using ADAGRAD we minimize loss when using learning rate 10^{-3}. PSNR is minimized when using learning rate 10^{-6} and SSIM is minimized when using learning rate 10^{-7}.

Loss, PSNR and SSIM with respect to the number of epochs





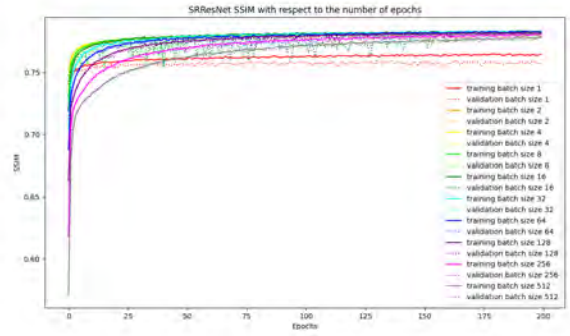
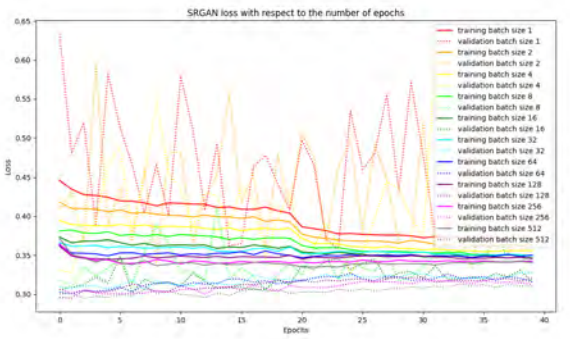
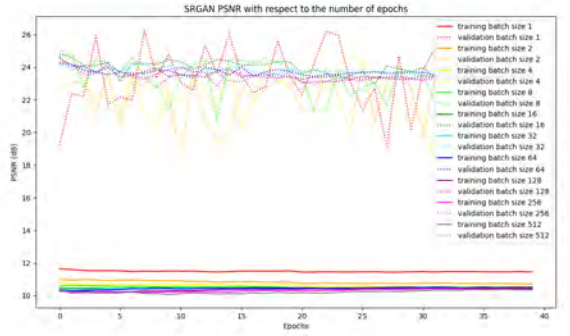


Fig. 12. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for different batch sizes.





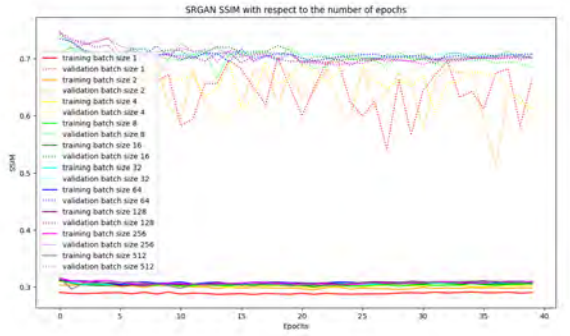


Fig. 13. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for different batch sizes.

for SRResNet using ADAGRAD optimizer can be seen on figure 18. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN using ADAGRAD optimizer can be seen on figure 19. We can confirm that using ADAGRAD optimizer with learning rate 10^{-3} yields best results for SRResNet and SRGAN. Using learning rate 10^{-1} has negative impact on performance.

The evaluation results of the RMSPROP optimizer and different learning rates are presented in table VII. We can observe that using RMSPROP optimizer with learning rate 10^{-4} yields best results across all testing datasets.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet using RMSPROP optimizer can be seen on figure 20. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN using RMSPROP optimizer can be seen on figure 21. We can confirm that using RMSPROP optimizer with learning rate 10^{-4} yields best results for SRResNet and SRGAN. Using learning rate 10^{-1} or 10^{-2} has negative impact on performance and causes unstable results.

E. Number of residual blocks

Here we explore the effect of different number of residual blocks in SRResNet and generator part or SRGAN.

The evaluation results of the different number of residual blocks are presented in table VIII. We can observe that using 16 residual blocks yields best results on all testing datasets.

Loss, PSNR and SSIM with respect to the number of epochs for SRResNet can be seen on figure 22. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN can be seen on figure 23. We observe that using 8 or 16 residual blocks gives the most table results ob both SRResNet and SRGAN. We can also see that increasing the residual blocks further increases the loss on the validation set.

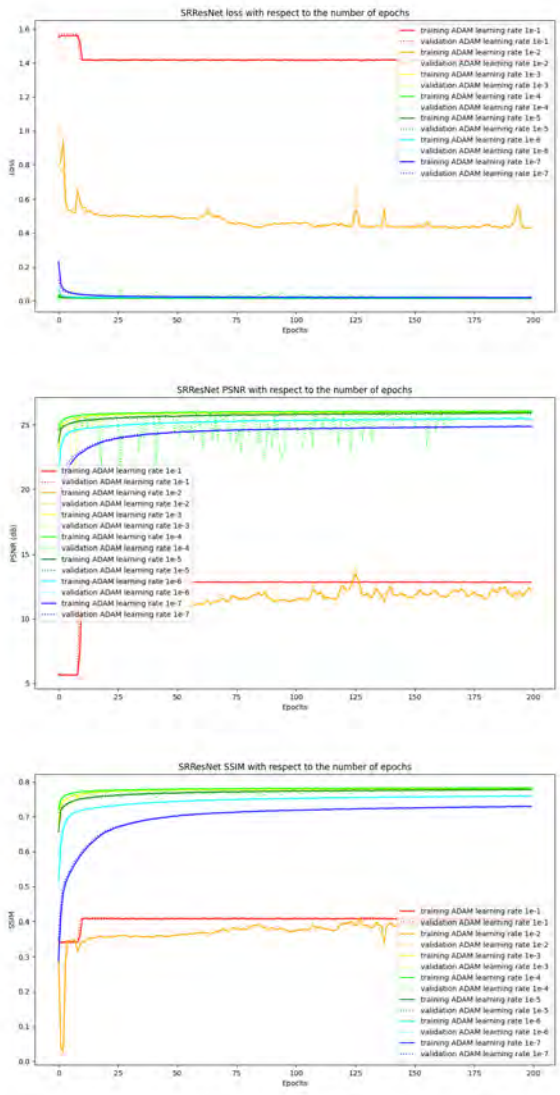


Fig. 14. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for using Adam optimizer at different learning rates.

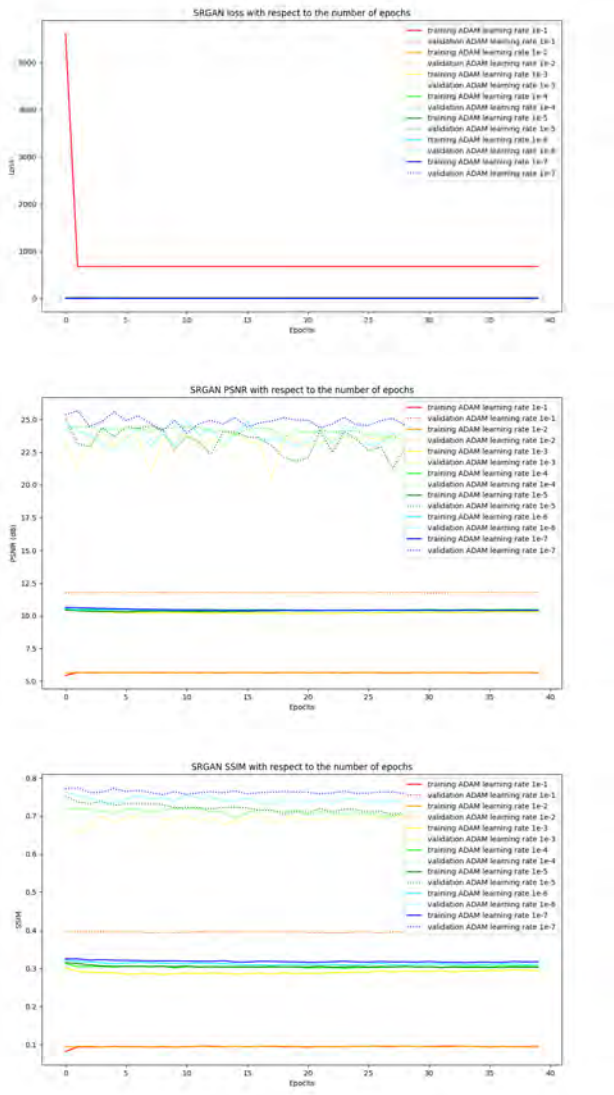


Fig. 15. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for using Adam optimizer at different learning rates.

V. Conclusion

We have described a deep residual network SRResNet and a generative adversarial network SRGAN. SRResNet can give us plausible results in certain circumstances, where an image in question is naturally smooth. SRGAN uses combination of content loss and adversarial loss to give us much more detail results.

We can note that the ideal loss function depends on the application. Approaches that try to create finer texture detail might be generally desirable, but less suited for medical applications. The perceptually convincing reconstruction of text is also very challenging to realize. The development of content loss functions that describe image spatial content, that are more invariant to changes in pixel space, will further improve photo-realistic image SR results.

It is also worth mentioning that standard quantitative measures (such as PSNR and SSIM) fail to capture and accurately assess the image quality with respect to the human visual system perception.

Of particular importance when aiming for photo-realistic solutions to the SR problem is the choice of the content loss

References

[1] Q. Yang, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:6788025

[2] W. Zou and P. C. Yuen, "Very low resolution face recognition problem," vol. 21, 10 2010, pp. 1 – 6.

[3] K. Nasrollahi and T. Moeslund, "Super-resolution: A comprehensive survey," *Machine Vision and Applications*, vol. 25, pp. 1423–1468, 08 2014.

[4] C.-Y. Yang, C. Ma, and M.-H. Yang, "Single-image super-resolution: A benchmark," 09 2014, pp. 372–386.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, 1e-1	**0.0148**	**27.6513**	**0.7742**	**0.0053**	**31.4667**	**0.8939**
SRGAN, 1e-1	0.2787	12.1388	0.4128	0.3264	12.1870	0.3806
SRResNet, 1e-2	0.0164	27.0814	0.7554	0.0071	30.4285	0.8719
SRGAN, 1e-2	0.2042	25.4762	0.7036	0.3073	27.3122	0.8139
SRResNet, 1e-3	0.0182	26.5140	0.7370	0.0098	29.3957	0.8430
SRGAN, 1e-3	0.1913	25.1054	0.7360	0.3038	25.6066	0.8414
SRResNet, 1e-4	0.0212	25.8920	0.7104	0.0145	28.0141	0.8009
SRGAN, 1e-4	0.1916	25.3612	0.7475	0.2547	25.8899	0.8467
SRResNet, 1e-5	0.0313	24.1183	0.6112	0.0319	24.6919	0.6651
SRGAN, 1e-5	0.2063	25.4319	0.7529	0.2679	26.1378	0.8535
SRResNet, 1e-6	0.0646	21.5710	0.4884	0.0863	21.0701	0.4909
SRGAN, 1e-6	0.2361	26.2418	0.7663	0.2705	27.4284	0.8719
SRResNet, 1e-7	0.2997	13.5937	0.2811	0.3314	12.8454	0.2308
SRGAN, 1e-7	0.2490	26.8058	0.7692	0.2638	28.9829	0.8782
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, 1e-1	**0.0123**	**28.2316**	**0.7890**	**0.0137**	**27.3240**	**0.7460**
SRGAN, 1e-1	0.2395	12.4859	0.3669	0.2602	13.7321	0.3799
SRResNet, 1e-2	0.0139	27.5859	0.7717	0.0147	26.9131	0.7321
SRGAN, 1e-2	0.2010	26.1120	0.7162	0.1895	25.2000	0.6605
SRResNet, 1e-3	0.0162	26.8108	0.7509	0.0161	26.4705	0.7175
SRGAN, 1e-3	0.1852	26.5645	0.7554	0.1757	25.6472	0.6993
SRResNet, 1e-4	0.0205	25.9539	0.7164	0.0188	25.9029	0.6891
SRGAN, 1e-4	0.1831	26.8729	0.7717	0.1754	25.9797	0.7218
SRResNet, 1e-5	0.0352	23.6456	0.5867	0.0286	24.0971	0.5708
SRGAN, 1e-5	0.1907	27.1484	0.7818	0.1855	26.1419	0.7313
SRResNet, 1e-6	0.0823	20.6619	0.4352	0.0589	21.6855	0.4433
SRGAN, 1e-6	0.2069	27.9260	0.7926	0.2082	26.7425	0.7429
SRResNet, 1e-7	0.2835	14.0807	0.2511	0.2420	14.7492	0.2710
SRGAN, 1e-7	0.2098	28.2456	0.7946	0.2197	27.0776	0.7458

TABLE V
EVALUATION RESULTS OF SGD OPTIMIZER AT DIFFERENT LEARNING RATES.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, 1e-1	1.5630	5.8904	0.3568	1.7727	5.4294	0.3280
SRGAN, 1e-1	0.2782	5.8904	0.3568	0.3234	5.4294	0.3280
SRResNet, 1e-2	0.0377	24.9989	0.7508	0.0207	27.5198	0.8659
SRGAN, 1e-2	0.3150	5.8904	0.3568	0.3427	5.4294	0.3280
SRResNet, 1e-3	**0.0150**	27.5786	0.7716	**0.0055**	31.3050	0.8913
SRGAN, 1e-3	0.1897	25.3809	0.7210	0.2537	27.5077	0.8447
SRResNet, 1e-4	0.0175	26.7673	0.7437	0.0089	29.7581	0.8522
SRGAN, 1e-4	0.1967	24.1881	0.7376	0.2758	23.0264	0.8322
SRResNet, 1e-5	0.0223	25.6923	0.6928	0.0167	27.5300	0.7757
SRGAN, 1e-5	0.1939	26.6645	0.7675	0.2507	29.2331	0.8768
SRResNet, 1e-6	0.0411	23.1019	0.5444	0.0532	22.9718	0.5862
SRGAN, 1e-6	0.2178	**27.6007**	0.7787	0.2380	**31.3681**	0.8931
SRResNet, 1e-7	0.3284	13.2122	0.2612	0.3868	12.3575	0.2214
SRGAN, 1e-7	0.2369	27.5398	**0.7789**	0.2346	31.2256	**0.8933**
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, 1e-1	1.4644	6.1837	0.3236	1.6350	5.6892	0.2978
SRGAN, 1e-1	0.2389	6.1837	0.3236	0.2588	5.6892	0.2978
SRResNet, 1e-2	0.0189	26.7962	0.7821	0.0175	26.8613	0.7437
SRGAN, 1e-2	0.2713	6.1837	0.3236	0.2870	5.6892	0.2978
SRResNet, 1e-3	**0.0125**	28.1517	0.7862	**0.0138**	27.2796	0.7437
SRGAN, 1e-3	0.1829	26.5904	0.7352	0.1815	25.4371	0.6764
SRResNet, 1e-4	0.0154	27.0977	0.7573	0.0156	26.6520	0.7212
SRGAN, 1e-4	0.1830	26.1493	0.7653	0.1756	25.3676	0.7118
SRResNet, 1e-5	0.0223	25.5590	0.6896	0.0200	25.6579	0.6647
SRGAN, 1e-5	0.1757	28.0251	0.7892	0.1783	26.8712	0.7416
SRResNet, 1e-6	0.0517	22.2720	0.5149	0.0380	23.2285	0.5021
SRGAN, 1e-6	0.1874	**28.5868**	0.7971	0.1938	**27.4829**	0.7524
SRResNet, 1e-7	0.2994	13.7605	0.2439	0.2763	14.2153	0.2592
SRGAN, 1e-7	0.1991	28.6120	**0.7985**	0.2146	27.4612	**0.7525**

TABLE VI
EVALUATION RESULTS OF ADAGRAD OPTIMIZER AT DIFFERENT LEARNING RATES.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, 1e-1	1.5630	5.8904	0.3568	1.7727	5.4294	0.3280
SRGAN, 1e-1	0.2815	13.1536	0.4244	0.3337	12.6921	0.3888
SRResNet, 1e-2	1.0984	7.3581	0.1873	0.9982	7.6160	0.2059
SRGAN, 1e-2	0.2863	7.8828	0.0066	0.3371	7.7273	0.0069
SRResNet, 1e-3	0.0164	27.1982	0.7746	0.0054	31.2938	0.8916
SRGAN, 1e-3	0.2084	25.2973	0.6969	0.2454	28.5261	0.8376
SRResNet, 1e-4	0.0144	27.8336	**0.7865**	**0.0046**	**31.9273**	**0.9012**
SRGAN, 1e-4	0.1979	24.4598	0.6942	0.3006	24.9009	0.7957
SRResNet, 1e-5	**0.0141**	**27.9284**	0.7830	0.0048	31.7686	0.8998
SRGAN, 1e-5	0.1919	25.6305	0.7144	0.2938	27.8801	0.8355
SRResNet, 1e-6	0.0155	27.4020	0.7661	0.0060	30.9602	0.8845
SRGAN, 1e-6	0.2113	24.0303	0.7315	0.3445	22.8058	0.8346
SRResNet, 1e-7	0.0179	26.6402	0.7396	0.0095	29.5686	0.8465
SRGAN, 1e-7	0.2013	25.2928	0.7547	0.2792	25.4219	0.8545
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, 1e-1	1.4644	6.1837	0.3236	01.6350	5.6892	0.2978
SRGAN, 1e-1	0.2434	13.8049	0.3816	0.2660	14.6831	0.3827
SRResNet, 1e-2	1.1400	7.2104	0.1471	0.9138	8.0171	0.1635
SRGAN, 1e-2	0.2473	8.0779	0.0070	0.2703	8.1470	0.0068
SRResNet, 1e-3	0.0133	27.9592	0.7897	0.0135	27.3762	0.7499
SRGAN, 1e-3	0.1990	25.4091	0.6955	0.2038	24.6655	0.6488
SRResNet, 1e-4	**0.0115**	**28.5690**	**0.7986**	**0.0131**	**27.5277**	**0.7551**
SRGAN, 1e-4	0.1927	25.3890	0.7061	0.1824	24.6674	0.6563
SRResNet, 1e-5	0.0118	28.4693	0.7956	0.0133	27.4794	0.7521
SRGAN, 1e-5	0.1857	26.5741	0.7300	0.1818	25.4256	0.6705
SRResNet, 1e-6	0.0130	27.9587	0.7811	0.0141	27.1562	0.7398
SRGAN, 1e-6	0.2072	25.8822	0.7551	0.1871	24.9091	0.6923
SRResNet, 1e-7	0.0160	26.9278	0.7529	0.0159	26.5542	0.7180
SRGAN, 1e-7	0.1884	27.0478	0.7827	0.1822	26.0861	0.7314

TABLE VII
EVALUATION RESULTS OF RMSPROP OPTIMIZER AT DIFFERENT LEARNING RATES.

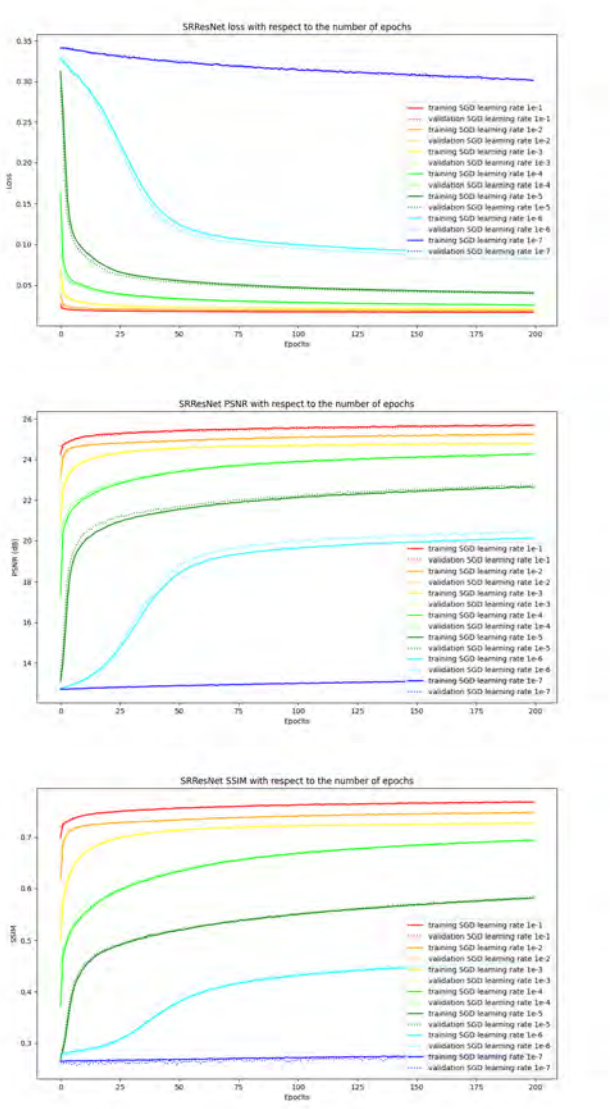


Fig. 16. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for using SGD optimizer at different learning rates.

	COCO Loss	COCO PSNR	COCO SSIM	Set5 Loss	Set5 PSNR	Set5 SSIM
SRResNet, # res blocks 1	0.0146	27.7220	0.7774	0.0052	31.4811	0.8954
SRGAN, # res blocks 1	0.1995	25.5735	0.7008	0.2426	28.5581	0.8304
SRResNet, # res blocks 2	0.0144	27.7865	0.7795	0.0051	31.5160	0.8961
SRGAN, # res blocks 2	0.1938	25.7625	0.7073	0.2220	28.8557	0.8408
SRResNet, # res blocks 4	0.0141	27.8920	0.7826	0.0049	31.7197	0.8989
SRGAN, # res blocks 4	0.1917	25.6810	0.7085	0.2319	28.7489	0.8414
SRResNet, # res blocks 8	0.0139	27.9864	0.7851	0.0048	31.8353	0.9003
SRGAN, # res blocks 8	0.1891	26.0564	0.7200	0.2375	29.4107	0.8515
SRResNet, # res blocks 16	**0.0136**	**28.1006**	**0.7882**	**0.0046**	**31.9512**	**0.9017**
SRGAN, # res blocks 16	0.1861	25.9949	0.7173	0.2426	29.2615	0.8454
SRResNet, # res blocks 32	0.0811	24.4021	0.7571	0.0717	21.9514	0.8473
SRGAN, # res blocks 32	0.2401	23.0640	0.6693	0.4164	20.0458	0.7253
	Set14 Loss	**Set14 PSNR**	**Set14 SSIM**	**BSD100 Loss**	**BSD100 PSNR**	**BSD100 SSIM**
SRResNet, # res blocks 1	0.0122	28.2579	0.7913	0.0136	27.3519	0.7482
SRGAN, # res blocks 1	0.1892	25.6392	0.6919	0.1956	24.7737	0.6367
SRResNet, # res blocks 2	0.0121	28.3104	0.7927	0.0135	27.3919	0.7496
SRGAN, # res blocks 2	0.1816	26.0606	0.7079	0.1875	25.1068	0.6536
SRResNet, # res blocks 4	0.0119	28.4147	0.7951	0.0133	27.4496	0.7520
SRGAN, # res blocks 4	0.1804	25.9324	0.7042	0.1834	25.0172	0.6559
SRResNet, # res blocks 8	0.0116	28.5236	0.7972	0.0131	27.5233	0.7537
SRGAN, # res blocks 8	0.1815	26.3066	0.7207	0.1799	25.4398	0.6731
SRResNet, # res blocks 16	**0.0114**	**28.6109**	**0.7990**	**0.0130**	**27.5807**	**0.7555**
SRGAN, # res blocks 16	0.1748	26.2480	0.7170	0.1767	25.3600	0.6697
SRResNet, # res blocks 32	0.0211	27.3702	0.7926	0.0477	25.9942	0.7383
SRGAN, # res blocks 32	0.2128	24.8214	0.7008	0.2024	24.4013	0.6523

TABLE VIII
EVALUATION RESULTS OF THE DIFFERENT NUMBER OF RESIDUAL BLOCKS.

[5] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," vol. 2, 12 2003, pp. 1398 – 1402 Vol.2.

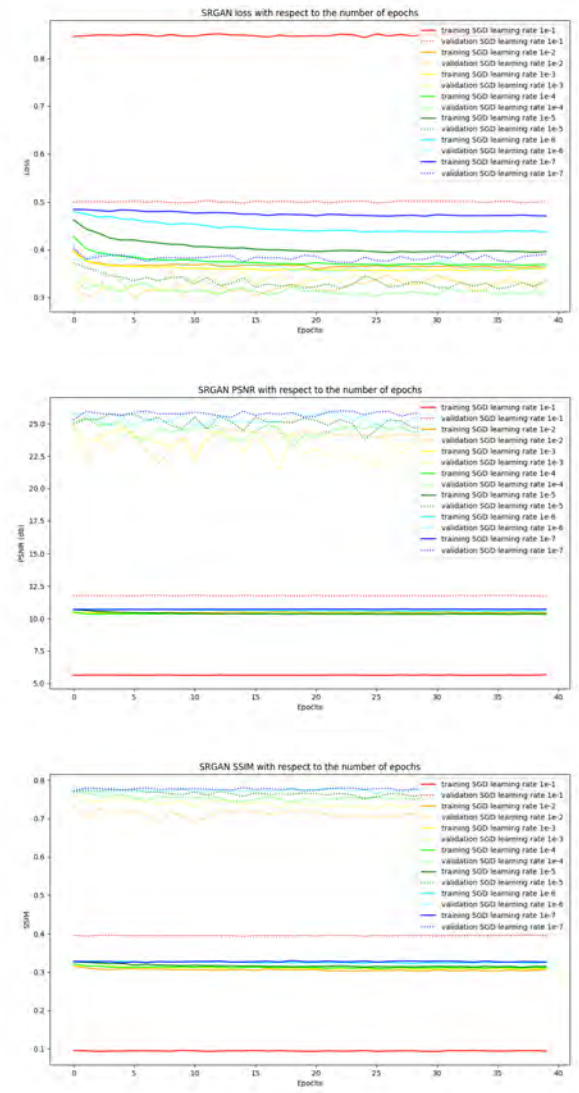


Fig. 17. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for using SGD optimizer at different learning rates.

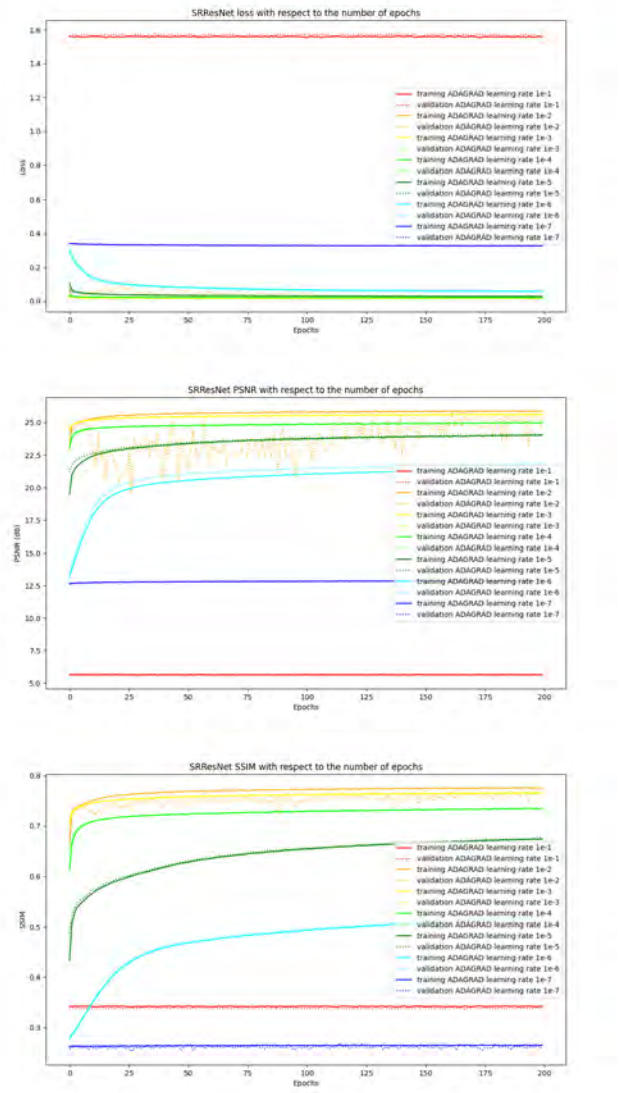


Fig. 18. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for using ADAGRAD optimizer at different learning rates.

[6] P. Gupta, P. Srivastava, S. Bhardwaj, and V. Bhateja, "A modified psnr metric based on hvs for quality assessment of color images," 12 2011.

[7] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 – 612, 05 2004.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[9] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," vol. 9906, 10 2016, pp. 694–711.

[10] J. Bruna, P. Sprechmann, and Y. Lecun, "Super-resolution with deep convolutional sufficient statistics," 11 2015.

[11] S. Borman and R. Stevenson, "Super-resolution from image sequences - a review," 09 1998, pp. 374 – 378.

[12] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 13, pp. 1327–44, 11 2004.

[13] C. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology - J APPL METEOROL*, vol. 18, pp. 1016–1022, 08 1979.

[14] J. P. Allebach and P. W. Wong, "Edge-directed interpolation," *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3, pp. 707–710 vol.3, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:16375380

[15] X. Li and M. Orchard, "New edge-directed interpolation," *Image Processing, IEEE Transactions on*, vol. 10, pp. 1521 – 1527, 11 2001.

[16] W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, pp. 25–47, 10 2000.

[17] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[18] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," 2015.

[19] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:13333501

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," 2015.

[21] ——, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[22] M. Mathieu, C. Couprie, and Y. Lecun, "Deep multi-scale video predic-

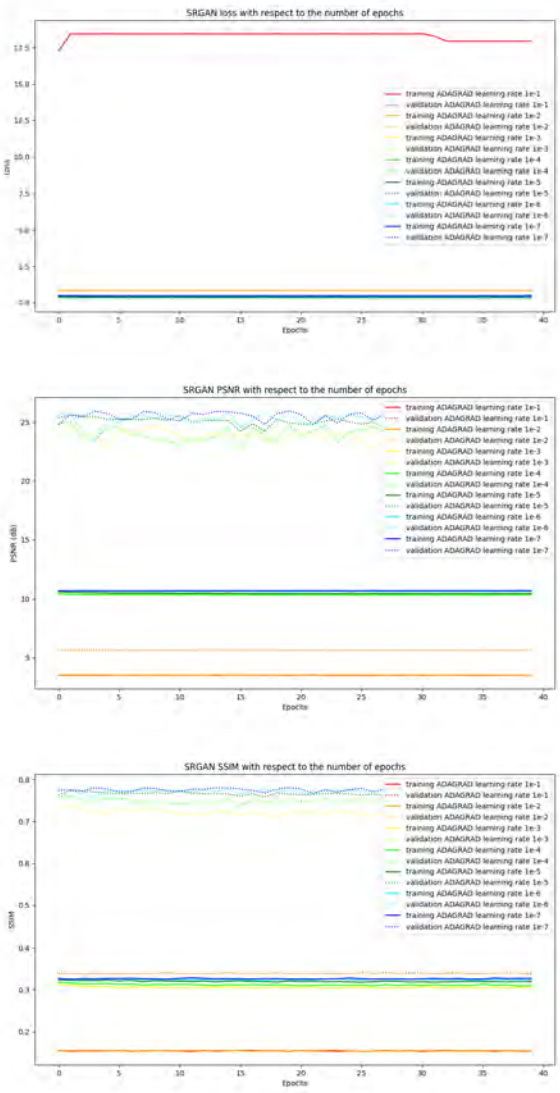


Fig. 19. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for using ADAGRAD optimizer at different learning rates.

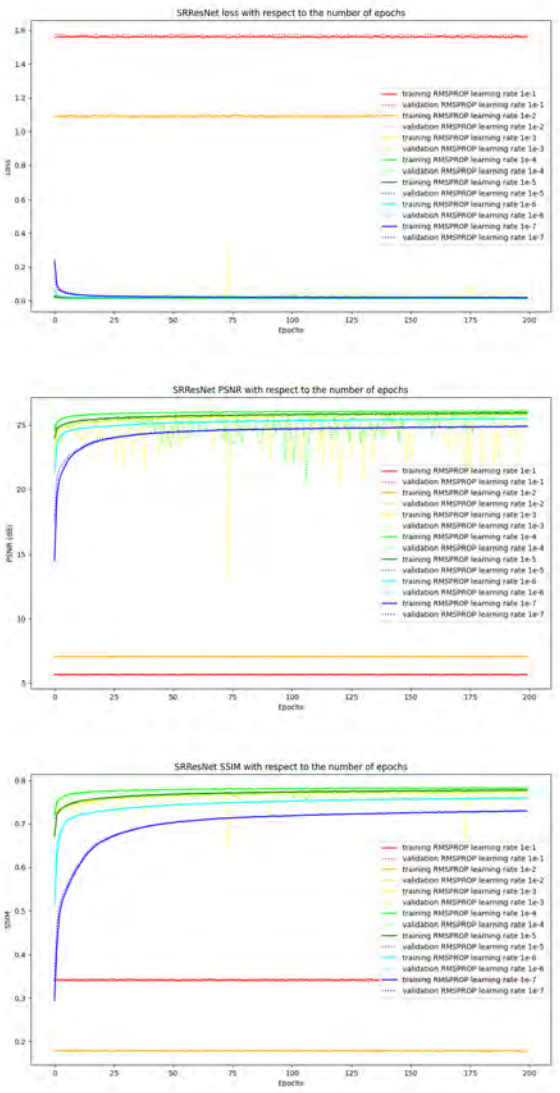


Fig. 20. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for using RMSPROP optimizer at different learning rates.

tion beyond mean square error," 11 2016.

[23] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," 2015.

[24] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," 2016.

[25] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," 2015.

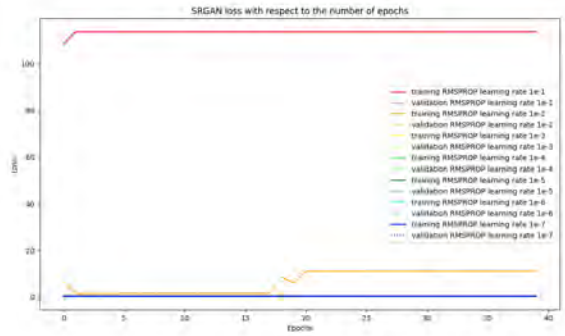
[26] L. Gatys, A. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," 06 2016, pp. 2414–2423.

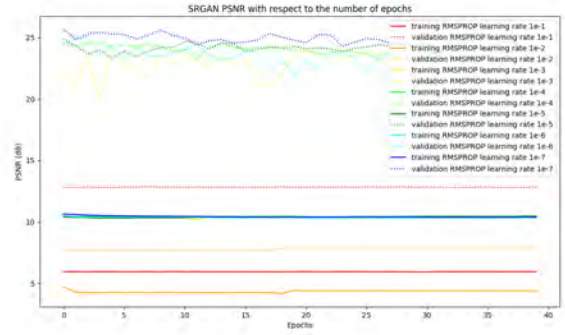
[27] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[28] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," 2016.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, 06 2014.

[30] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "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, pp. 1874–1883.





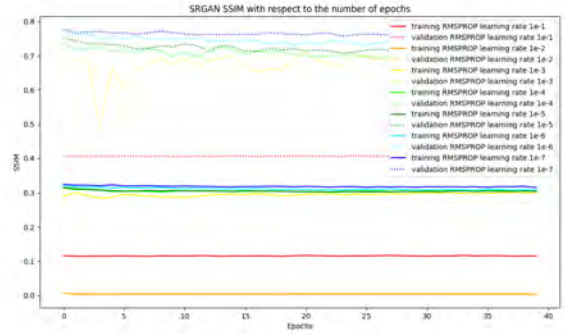
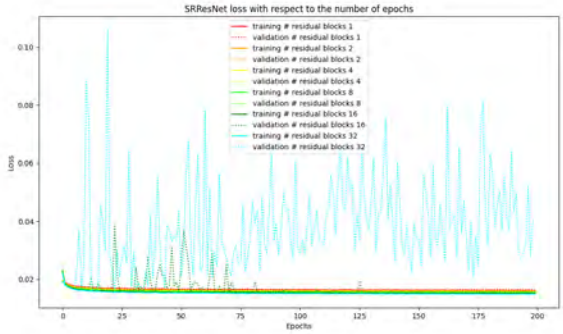
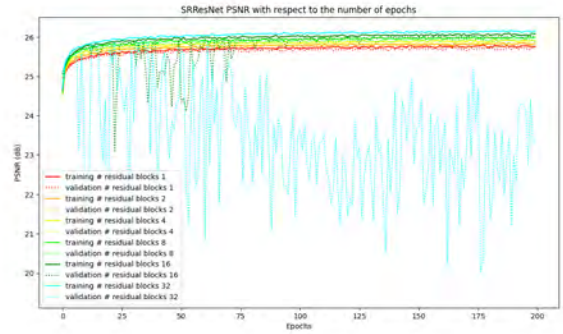


Fig. 21. Loss, PSNR and SSIM with respect to the number of epochs for RMSPROP for using Adam optimizer at different learning rates.





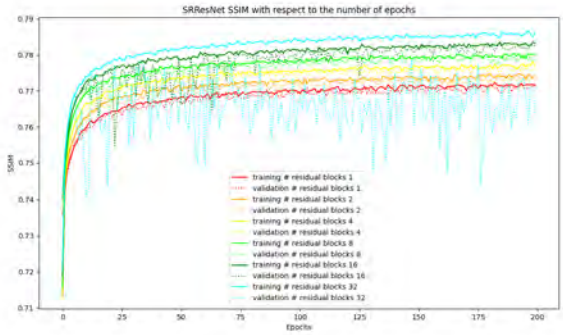


Fig. 22. Loss, PSNR and SSIM with respect to the number of epochs for SRResNet for the different number of residual blocks.

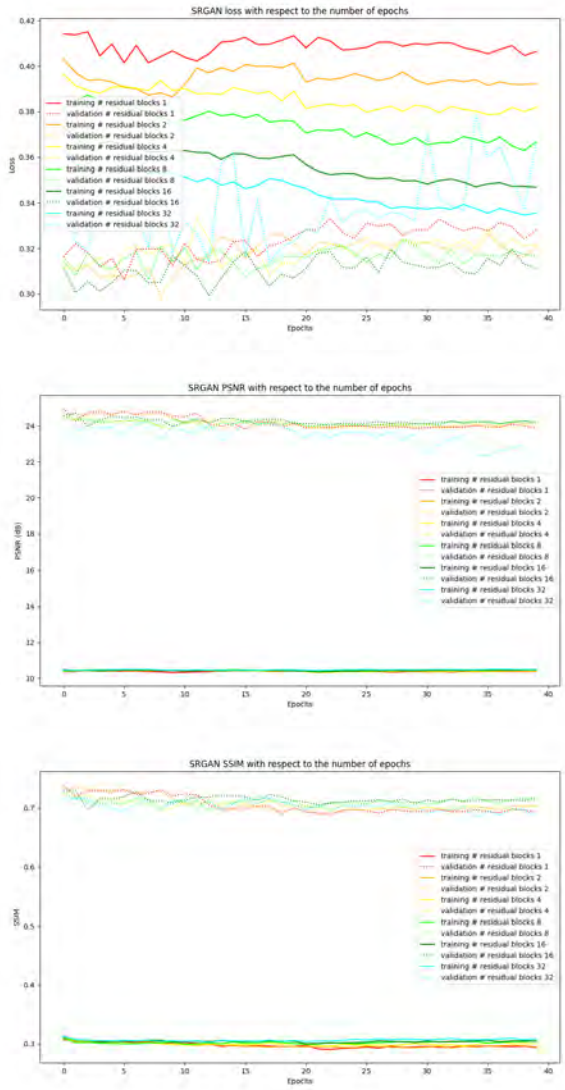


Fig. 23. Loss, PSNR and SSIM with respect to the number of epochs for SRGAN for the different number of residual blocks.