Image Super-Resolution and Colorization in a single Generative Adversarial Network

Erez Yosef
erez.yo@gmail.com

Shay Shomer Chai
shay5510@gmail.com

School of Electrical Engineering, Faculty of Engineering,
Tel-Aviv University, Israel.

Abstract

Image colorization and image super-resolution are two of the challenging and interesting tasks in computer vision. Many solutions presented for each task over the last years. Generative adversarial networks (GAN) based solutions proven to gain better results in recent papers compare to previous studies. In this work we introduce a novel GAN architecture to apply both image colorization and super-resolution by a single network (CSRGAN). To our knowledge, it is the first attempt to perform both tasks in a single GAN. The main difference between the two application is the type of the used information. While the colorization uses the global information (objects) in the image, the SR uses the local information (textures) in the image. We needed to consider it in the network structure development process. We perform modifications and changes to the network and apply different training methods based on previous works to achieve best results. Our CSRGAN[1] result example is shown in Figure 1.

1. Introduction

Super resolution (SR) problem receives significant attention in the computer vision research community. In super resolution problems the high-resolution (HR) image is reconstructed from the given low-resolution (LR) image. The estimated result image named super resolution (SR) image. The main problem in SR is the absent of the high frequencies content and the texture details. When using high upscaling factors this lack of information is more significant.

A pioneer work in the field of super resolution is SRCNN by Dong et al. [17]. Since then many studies improved the super resolution performances by using various network architectures and different methods [2,18,19, 20,21,22].

Image colorization is an attractive research area in the field of computer vision and machine learning for the recent years. An automatic colorization of a grayscale image is the desired result in image colorization tasks. This is due to the large variety of applications such color restoration and image colorization for animations. Models for the colorization of grayscales began back in the early 2000s. In 2002, Welsh et al. [4] proposed an algorithm that colorized images through texture synthesis. Levin et al. [5] proposed an alternative formulation to the colorization problem in 2004. This formulation followed an inverse approach, where the cost function was designed by penalizing the difference between each pixel and a weighted average of its neighboring pixels. Both proposed methods still required significant user intervention which made the solutions less than ideal. In [6], a colorization method was proposed by

Input x4 **Bicubic** **CSRGAN** **HR**

Figure 1: Best result example of CSRGAN model.

[1] The code is available at https://github.com/shay5510/CSRGAN

comparing colorization differences between those generated by convolutional neural networks and GAN.

In this work we propose super resolution and colorization in a single generative adversarial network (CSRGAN) to perform both tasks in one network. By combining both tasks in a single network we wish to gain optimal utilization of the input information compared to separate networks and obtain better results. We integrate different architectures and training methods based on previous works and examine their effect on the network performance. the network designed for input images in 64x64 pixels resolution and 4× upscaling factor such that the output size is 256x256 pixels. An example result image is shown in Figure 1.

2. Related Work

Generative adversarial networks represented in 2014 by Goodfellow et al. [1]. This type of generative model composed of two trainable networks: generator and discriminator. The generator trained to produce new data samples, whereas the discriminator trained to distinguish (classify) if the data sample is real or fake generator result. Both networks trained simultaneously. In training the generator learns the distribution function of the produced data to create new data samples from this distribution and to generate new results that are indistinguishable from a real data. The discriminator classification result is used such that the generator tries to fool the discriminator. Since our study is in the field of image translation and analyzing, both networks are convolutional neural networks (CNNs).

2.1. Image Super Resolution

Image super-resolution is a research field with a high interest, and such tasks were researched intensively over the last years. New developments in neural networks area and deep learning technics enabled a significant advance in super resolution problems and obtained better quality results. A pioneer work for deep neural networks solutions for SR problems is Dong et al. [11,17], proposed SRCNN that achieved better performance compared to previous studies.

A milestone paper in the single image super resolution research field is Ledig et al. [2]. Generative adversarial network was introduced for solving SR problem, to encourage the network to generate more realistic images. The system designed such that the generator input is a low-resolution (LR) image and it generates a super-resolution (SR) image. The generator designed as a fully convolutional network, and it was built of residual blocks.

The study also proposed a new content loss based on the VGG features of the pre-trained 19 layers VGG network described in Simonyan and Zisserman [3].

Some improvement methods where suggested in Wang et al. [10]. This study was based on the SRGAN paper by Ledig et al. [2], and the new represented methods aim to improve the results of the generator and enhance the super resolution output image.

First, the residual block [2] replaced by the suggested Residual-in-Residual Dense Block (RDDB). Also, the Batch Normalization (BN) [23] layers were removed as in [24] and residual scaling [24,25] was used.

Another suggested improvement is using Relativistic average GAN (RaGAN) [26] for the adversarial loss of the discriminator. Such relativity creates dependency in the loss between the fake and real images and enables to share an information between them. In this way, the generator learns to create more realistic images using information from the real images.

2.2. Image Colorization

Image colorization is a research field aims to color grayscale images, this field attracts attention because it allows to revive pictures and videos from the past. In the past most algorithms were based on user intervention in order to color the images, but in the last years new algorithms using CNN was introduced [27]. These algorithms use neural network in order to learn how to color grayscale images based on existing data. The state-of-the-art models uses Generative Adversarial Networks for such task [6,7]. The main idea is to insert a grayscale image to the generator and the generator try to generate a colored image out of this input image. While the generator is trying to create colored images, the discriminator is trying to distinguish between the original colored images and the fake ones generated by the generator. The generator and discriminator are trained simultaneously until convergence, when the discriminator can't distinguish between real colored images and generated ones.

In 2018 Nazeri et al. [7] proposed image colorization solution using GAN. They proposed a generator based on U-net architecture [8] while the discriminator based on convolutional neural networks (CNN). Isola et al [6] proposed image to image translation using GAN for multiple translation tasks such as edges to photo, day image to night and image colorization. They introduced a general architecture using a variant of U-net architecture for the generator network and discriminator based on neural networks (CNN).

Each paper uses encoder-decoder network for the generator in order to learn global features from the grayscale images and generate color images from those learned features. The discriminator gets the generated image and try to determine whether it real or fake, encourage the generator to the produce real images. One of the main problem with training GANs is the mode collapse issue, the generator learns to generate only handful of colored images for all grayscale images. In order to prevent mode collapse both papers use conditional GAN [7], adding the input grayscale image to the discriminator, forcing a dependency between the input grayscale image and the generated color image.

Colorization GANs add content loss to the generator loss to obtain realistic images that similar to the target image. In Nazeri et al. [7] L2 loss was used between the generated image and the real image, while Isola et al [6] used L1 loss for better edge preserving and less blurring.

3. Methods

In this section we introduce our methods for CSRGAN. We define a baseline model relied on the latest studies in the field of image colorization and image super resolution using Generative Adversarial Networks. We will test new approaches and we will introduce our best model as CSRGAN.

3.1. Generator Architecture

The Generator is used to create SR colored images from grayscale LR images. Our baseline Generator architecture (Figure 2) is a combination of U-net architecture for colorization, the colorization task same as Isoal et al. [6] and residual blocks for super resolution task same as Ledig et al [2]. The input image is 64X64 grayscale image, the L component of the L*a*b down sampled 4x original 256X256X3 image. We concatenate the output of the U-net last layer to the input grayscale image (Figure 2), and then enter the 64X64X3 shape data to the residual blocks for super resolution same as Ledig et al. [2]. Although the input image to the residual block is not BGR image like the input in the original SRGAN paper [2] the residual block's output is still SR BGR image.

This network structure creates a bottleneck in the data flow since the features dimension is reduced to 3 between the network components. We refer this bottleneck in our experiments and test its effect on the results.

In Nazeri et al. [7] the generator network is consisted of another 2 convolutional layers one at the beginning and the other at the end of the network, with stride 1 and kernel size of 3X3. During our training we tested this approach and it outperformed all tested models.

Our generator consisted 2 main blocks colorization and SR which trained together end-to-end. We used this specific structure because the colorization task is based on global feature learning [6,7] whereas the super resolution task is based on local feature learning [2,10]. This important noting force us to build generator capable of learning global and local features, allowing us to generate colored SR images.

Following this observation our generator was trained on the whole image whereas SRGAN/ESRGAN [2,10] were trained on crops from the input image (learning local feature learning).

3.2. Content Losses

The pixel-wise MSE loss is calculated as:

$$l_{MSE}^{CSR} = \frac{1}{r^2W*r^2H}\sum_{x=1}^{r*W}\sum_{y=1}^{r*H}\left(I^{HR}-G_{\theta_G}\left(I_{grayscale}^{LR}\right)\right)^2 \quad (1)$$

MSE loss are widely used on which many state-of-the-art approaches rely [11, 12], including Nazeri et al. [7]. However, while achieving particularly high PSNR, solutions of MSE optimization problems often lack high frequency content which results in perceptually unsatisfying solutions with overly smooth textures. Instead of relying on pixel-wise losses we build on the ideas of Gatys et al. [13], Bruna et al. [14] and Johnson et al. [15] and use a loss function that is closer to perceptual similarity. In Ledig et al [2] they defined the VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network described in Simonyan and Zisserman [3]- φi,j indicate the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network:

$$l_{VGGi,j}^{CSR} = \frac{1}{W'*H'}\sum_{x=1}^{W'}\sum_{y=1}^{H'}\left(\varphi_{i,j}\left(I^{HR}\right)-\varphi_{i,j}\left(G_{\theta_G}\left(I_{grayscale}^{LR}\right)\right)\right)^2$$

The MSE loss of the VGG 19 features is calculated in features space thus we define W' and H' as the relevant size of the feature map.

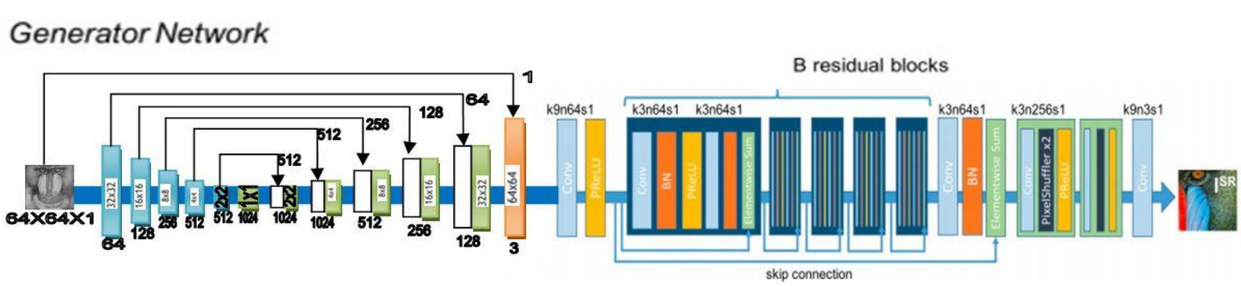

Figure 2: Generator base line architecture. Source: *partially from Ledig et al. [2]*

3.3. Discriminator Architecture

The discriminator is used to distinguish between the real images and the fake generated images. It performs as a classifier, where the input is an image in BGR color space, and the output is the predicted probability that the input image is a real image.

The discriminator network designed the same as Ledig et al. [2], and it defined as our baseline (in Figure 3). On this architecture we perform modifications and changes to obtain best results based on previous studies.

3.4. Adversarial Loss

According to Goodfellow et al.[1] We want to solve the min-max problem for the discriminator network parameters θ_D and the generator network parameters θ_G as shown in Eq.2, when I^{HR} is a real high resolution image, and I^{SR} is a super resolution image generated by the generator.

$$\min_{\theta_G}\max_{\theta_D}\begin{pmatrix} \mathbb{E}_{I^{HR}}\left[\log D_{\theta_D}(I^{HR})\right] + \\ \mathbb{E}_{I^{LR}}\left[\log\left(1 - D_{\theta_D}\left(G_{\theta_G}(I^{LR})\right)\right)\right] \end{pmatrix} \quad (2)$$

The adversarial loss encourages the generator to fool the discriminator and to create more realistic images. It computed as shown in Eq.3 for better gradient behavior, instead of the form specified in Eq.2.

$$l_{GEN,adversarial} = -\mathbb{E}_{I^{LR}}\left[\log D_{\theta_D}\left(G_{\theta_G}(I^{LR})\right)\right] \quad (3)$$

Note that the first term in Eq.2 is independent of the generator parameters θ_G, therefore it absents from the adversarial loss.

3.5. Conditional GAN

One of the main problems with GANs is training them. This type of reinforcement learning is known as tricky, especially when training deep networks. One of the main concerns in GANS is mode collapse. Mode collapse is a situation when the generator learns how to produce a handful of images and thus not learning the whole data distribution. In order to prevent such cases conditional GAN is used. When using conditional GAN we supply the discriminator the input image and generated image allowing the discriminator to get decision with respect to input image. Both papers use conditional GAN [6,7], adding the grayscale input images to the discriminator.

$$l_{cadversarial} = \min_{\theta_G}\max_{\theta_D}\begin{pmatrix} \mathbb{E}_{I^{HR}}\left[\log D_{\theta_D}(I^{HR}|I^{LR})\right] + \\ \mathbb{E}_{I^{LR}}\left[\log\left(1 - D_{\theta_D}\left(G_{\theta_G}(I^{LR})|I^{LR}\right)\right)\right] \end{pmatrix} \quad (4)$$

In our task, due to the super resolution the input and the output of the generator (LR and SR respectively) are in different dimensions therefore we cannot concatenate them together to create the wanted input of the discriminator.

The baseline architecture of the discriminator consists of convolutional blocks with stride equals two/one alternately. Namely the spatial dimension is reduced through the convolutions every other block in the discriminator, while the depth of the feature map is increased. Therefore, while using upscale factor of 4, after the fourth block the spatial dimension is equal to the LR image. To add this information (LR) to the model we reduced the output features of the fourth convolution block by 1 and concatenated the one-dimension (gray-scale image) LR to the feature map dimension. There are 256 features in this stage and the rest of the architecture is same as the baseline.

3.6. Relativistic Average GAN

The standard discriminator can be expressed as in Eq.5, where σ is the sigmoid function and C is the non-transformed discriminator output.

$$D(x) = \sigma\big(C(x)\big) \rightarrow 1\ (if\ real) \quad (5)$$

This standard discriminator estimates the probability that the input image is a real image.

Relativistic average GAN [26] is a method to create and train the adversarial network with more information that missing in the standard GAN. In a standard GAN, after enough training the discriminator will be able to correctly classify the input images. Eventually, the generator succeeds to fool the discriminator, such that the fake images classify as real images. In this case the discriminator classifies all input images as real, regardless the a priori knowledge that half of the images are real and half are fake.

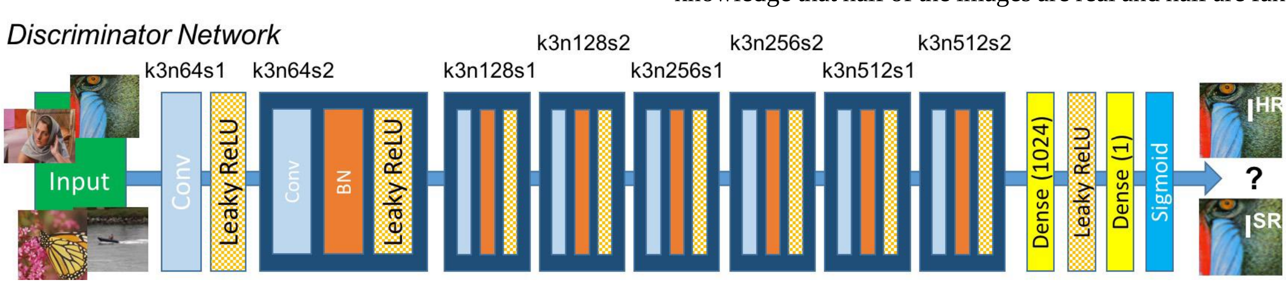

Figure 3: Discriminator architecture. Source: *Ledig et al. [2] Figure4*.

To add the needed dependency in the relativistic GAN the discriminator output for an image is compared to other discriminator outputs for images from the opposite class.

The relativistic average discriminator predicts the probability that an input image is more realistic than the average output of the counter class images. The discriminator function represented in Eq.6 for real input image x_r with respect to the average of the fake images x_f outputs, and the wanted result is 1 in this case of real image. The opposite case, for fake image represent in Eq.7. The $\mathbb{E}[\cdot]$ represent the average operator over all images in the mini-batch.

$$D_{Ra}(x_r, x_f) = \sigma\big(C(x_r) - \mathbb{E}[C(x_f)]\big) \rightarrow 1\ (real) \quad (6)$$

$$D_{Ra}(x_f, x_r) = \sigma\big(C(x_f) - \mathbb{E}[C(x_r)]\big) \rightarrow 0\ (fake) \quad (7)$$

The discriminator loss defined in Eq.8 based on these definitions.

$$l_D^{Ra} = -\mathbb{E}_{I^{HR}}\left[\log D_{Ra}(x_r, x_f)\right] - \mathbb{E}_{I^{LR}}\left[\log\left(1 - D_{Ra}(x_f, x_r)\right)\right] \quad (8)$$

The adversarial loss for the generator is shown in Eq.9. It different from the original adversarial loss (in Eq.3) due to the added dependency between the HR images to the SR images (the first term in the min-max in Eq.2).

$$l_{adversarial}^{Ra} = -\mathbb{E}_{I^{HR}}\left[\log\left(1 - D_{Ra}(x_r, x_f)\right)\right] - \mathbb{E}_{I^{LR}}\left[\log D_{Ra}(x_f, x_r)\right] \quad (9)$$

The RaGAN adversarial loss contains both HR and LR images, thus the generator parameters use not only the LR images (the trivial case), but also the real images HR to improve the results. This part is absent in the original adversarial loss.

4. Experiment

In order to get the best results we explore many architectures and tested them on the same data:

- Baseline as defined above – Isoal et al. [6] colorization combined with SRGAN [2] generator network as shown in Figure 2(using 6 residual blocks). The discriminator same as Ledig et al. [2] (Figure 3).
- Conditional GAN – baseline generator. Discriminator receives LR input and the generator output/HR image.
- Relativistic Average GAN – the baseline architecture. Changes in the discriminator output as Eq. 6,7.
- UBN - un-bottlenecked U-net, replacing the last bottleneck layer output to 64 feature channels (instead of 3) include grayscale image concatenated. In addition, the first residual block changed to input of size Bx64x64x64.
- RDDB generator - replacing the residual blocks with 10 RDDB blocks as specified in ESRGAN [10].
- Our best model: Feature extractor generator- adding another feature extractor in the beginning of the generator network and convolutional layer at the output of the last upsample layer (after the 32 to 64 layer) in order to get better inference. This approach is used at Nazeri et al. [7]. This generator architecture is shown in Figure 4.

4.1. Data

For training we used Stanford Dogs Dataset [16] containing 20K images of 120 dog breeds. For architecture testing we used 2K images from ~12 breeds. After examination of the results we picked our best performance model and trained it on 6K images from ~30 dog breeds. One important noting is that the resolution of the images is diverse but most of them are at least 224X224.

4.2. Training Details

We trained our models on single NVIDIA Tesla K80 GPU using google cloud services. We used batch size of 16 with random samples from our dataset, every HR image in the batch was pre-procced in order to fit our model. We resized the input images to 256X256 BGR images and down sampled 4x the images to 64X64 BGR images, we changed the color space of the LR images to L*a*b and extracted the luminance component, thus defining our

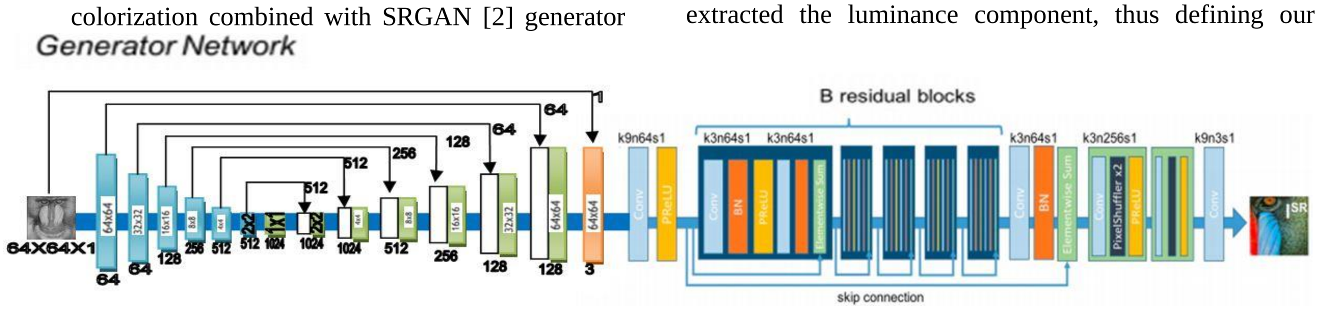

Figure 4: Generator chosen architecture including feature extractor. Source: *partially from Ledig et al. [2]*

generator input image as 64X64 grayscale image. We scaled the HR images to [0,1] and the LR input images to [-1,1] for better gradient flow. We split our dataset randomly to train and test (9:1 ratio) getting images from different dog breeds.

While the residual blocks are full convolutional and compatible with any input size as specified in Ledig et al. [2] the U-net architecture is fixed to specific size, thus training has been made on the whole image instead of crops, leading to longer duration of training, approximately 3 hours of training on the 2K images dataset.

We define our baseline losses as follow:

$$l_G^{CSR} = l_{MSE} + \lambda l_{g-adv*} + \xi l_{VGG} + \eta l_{TV} \quad (10)$$

- $\lambda = 10^{-3}$, $\xi = 6 * 10^{-3}$, $\eta = 2 * 10^{-8}$
- l_{TV} is the total variation loss.
- l_{g-adv*} is the adversarial loss in all models except the Relativistic Average GAN, where the loss is defined in 3.6.

$$l_d^{CSR} = l_{d-adv*} \quad (11)$$

- l_{d-adv*} is the adversarial loss in all models except the Relativistic Average GAN, where the loss is defined in 3.6.

Due to the big difference between generator and discriminator depth, we used different learning rates in order to allow the generator to learn and to prevent the training from failing. We trained our generator with learning rate of 10^{-4} whereas the discriminator was trained with learning rate of 10^{-5}. We trained our suggested architectures for 20 epochs on the 2K dataset specified above.

After we determine our best model (Feature extractor generator) we trained our model for 20 epochs on larger dataset of 6K images and ~30 dog breeds. This training duration was 8 hours long.

4.3. Qualitative Results

Our best model was selected based on the visual results of the different methods and architectures we mentioned and tested. Since there are no previous works of integrated SR and colorization tasks we compared the SR results to the bicubic interpolation of the colored low-resolution image. We evaluate the quality of the results mostly based on the visually appearance, and we computed PSNR and VGG-MSE for the results. The VGG-MSE metric computed as the mean squared error of the VGG-19 features as used in the perceptual loss. The results are shown in figure 5, and they obtained by the standard GAN (neither conditional GAN nor relativistic average GAN), with the added feature extractor layer to the baseline generator as shown in Figure 4.

Although that the PSNR is usually lower in our results compare to the bicubic interpolation, it is noticeable that our results contain more textures and high frequencies content. That is a limitation of the PSNR and MSE metrics in evaluating super resolution results [2,10].

In addition, we also have a color differences of the HR image due to the automatic colorization. these differences decrease the PSNR, while that the bicubic result obtained by a colored-low-resolution image, and it does not suffer from this color noise.

4.4. Ablation Studies

We gradually modify our baseline model to compare the effect of each used method and the obtaining results. From those results we determine our best model. In order to study the effects of each component in the proposed CSRGAN, we gradually modify the baseline CSRGAN model and compare their differences. The overall visual comparison is illustrated in Figure 6. Each column represents a model with its configurations shown in the top.

The first model we tested was our baseline model defined as a combination of the Isoal et al generator for colorization and residual blocks for super resolution, discriminator same as Ledig et al. [2] as shown in Figure 4. The baseline model achieve good results but struggled in the colorization task and colored the images in light green. Consider the dataset is consisted of dog breeds we assumed that a substantial amount of images was pictured in a green background (grass, forest, garden etc.) causing the generator to color the images in light green.

Following this observation, we decided to use relativistic average loss, encourage the discriminator to better distinguish between real and fake images and making generator work harder in order to produce real images. The Relativistic GAN solve the colorization with light green but caused artifacts on the images making them unrealistic.

Following this results we decided to take a different approach in order to make the generator produce images more correlated to the input image. We used conditional GAN as specified in Isola et al. [6], encourage the discriminator to make decision with respect to the input grayscale image. This approach yields very good results, without artifacts or extra colors. Although the colors were good and the artifacts disappeared we got a bit blurry images. We decided to go back to the baseline architecture. When we observe the baseline results we noticed that pixel's color is approximately the same in the neighborhood of the pixel, we figured that the generator doesn't have enough global information in order to determine the pixel's color precisely, causing blurry images and mis-colored pixels.

We decided to add another convolutional layer with stride 1 and kernel 3X3 at the beginning and at the end of the U-net layers for better feature learning (as at Nazeri et al. [7]). This approach got the best results from all models tested.

We tested another 2 networks:

- RDDB blocks instead of residual blocks – because the generator's depth we were forced to remove layers from the original network from Wang et al [10] causing the results to be un realistic.
- UBN- our un-bottlenecked generator gave good results but struggled to color the images.

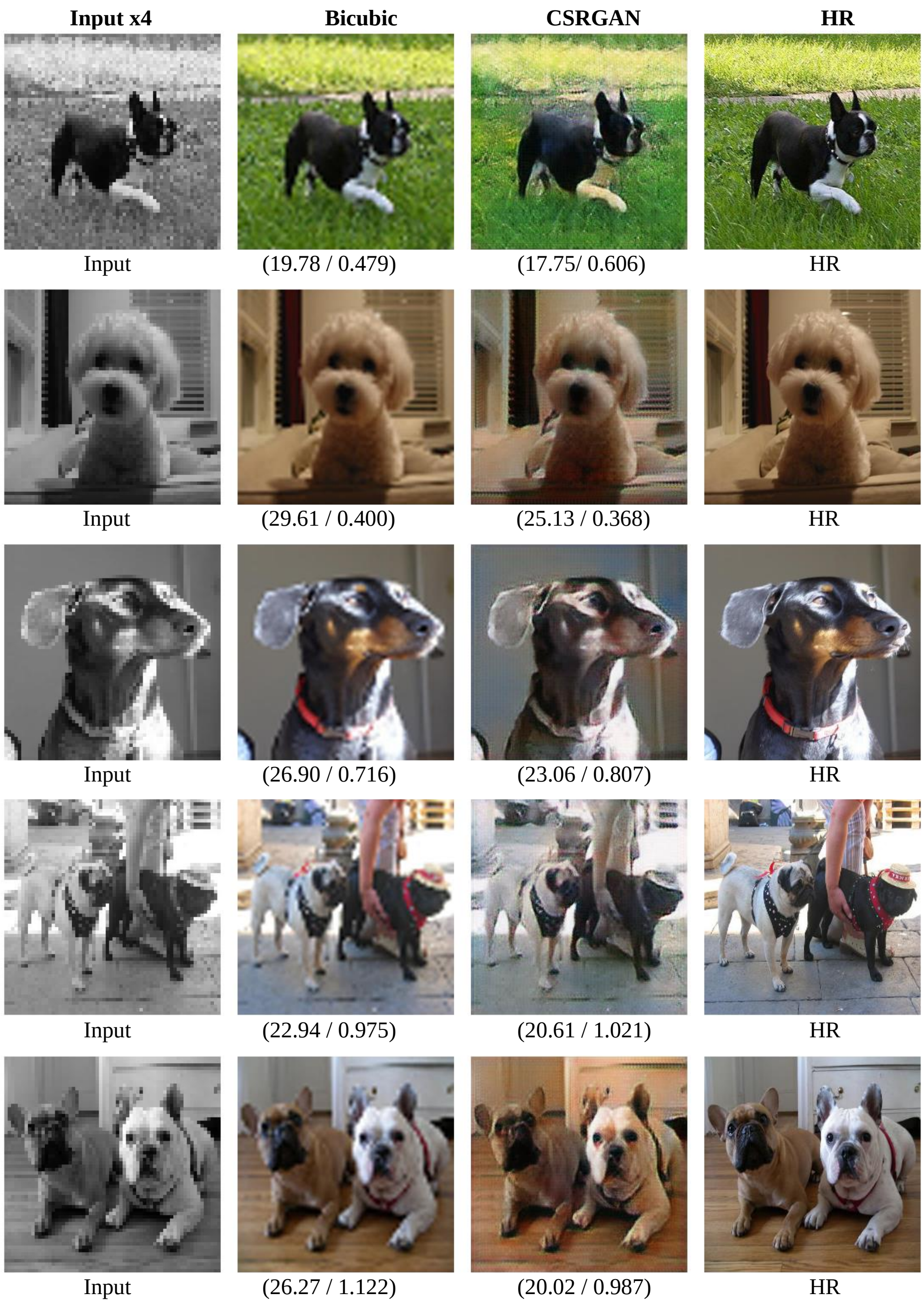

Figure 5: Qualitative results of the network. Our CSRGAN results and colored-LR images that bicubic interpolated are Compared to the ground truth (HR) visually and by PSNR and VGG features MSE values (PSNR / VGG-MSE)

We don't show the results of these two networks in this study because they didn't add any insights to our network and didn't contribute to our research.

5. Conclusion and Future Work

We presented our CSRGAN, the first colorization and super resolution generative adversarial network using

GAN?	Baseline GAN	Relativistic GAN	Conditional GAN	Feature GAN	Ground Truth
Extra Feature layer?	✗	✗	✗	✓	------------------------
VGG loss?	✓	✓	✓	✓	------------------------
Adversarial loss?	✓	✓	✓	✓	------------------------
Relativistic loss?	✗	✓	✗	✗	------------------------

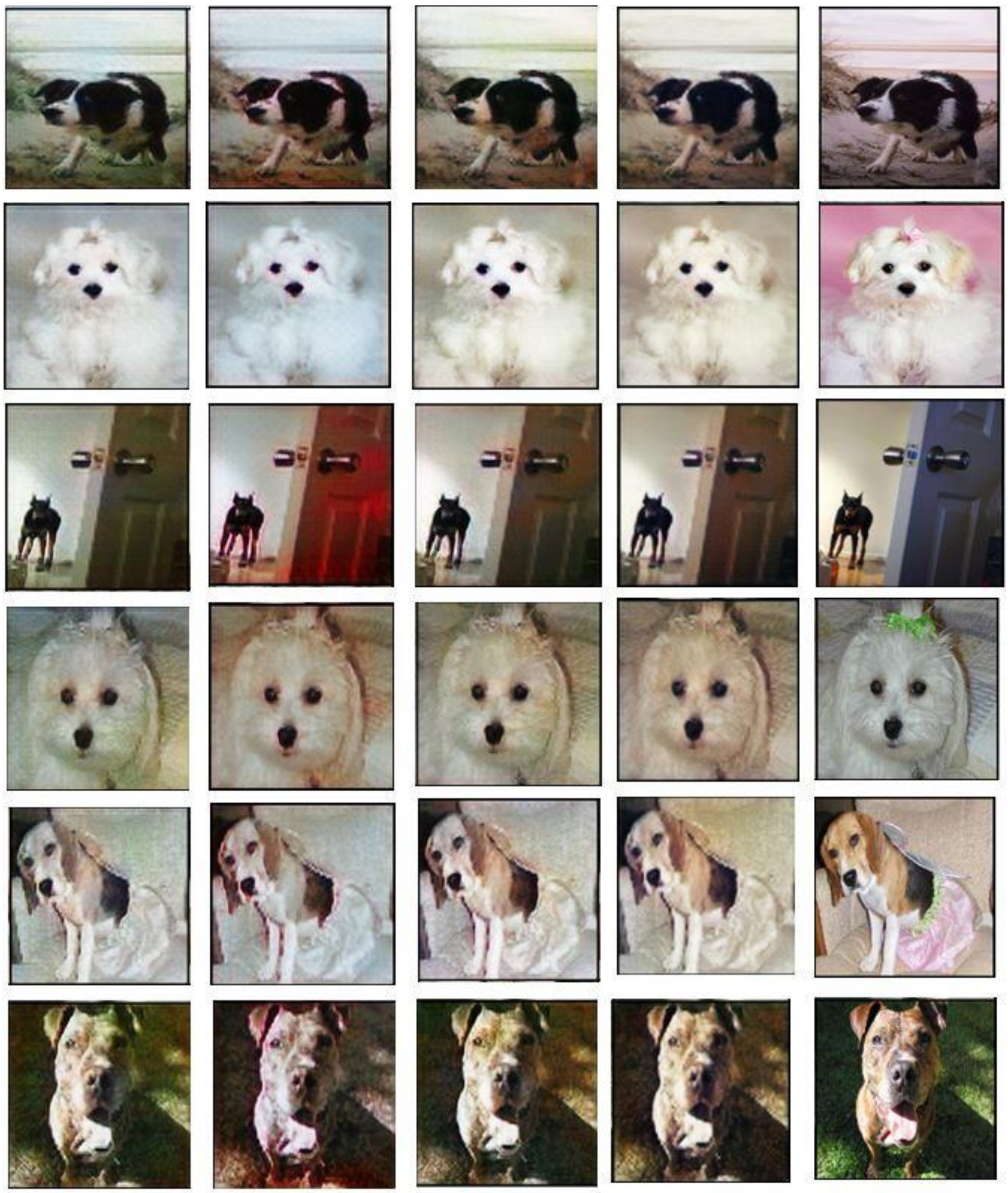

Figure 6: Generator chosen architecture including feature extractor. Source: Ledig et al. [2] figure

single network trained end-to-end. Our generated colored HR images were characterized with realistic colors and high frequency textures similar to the original images.

We saw that some of the methods we tested did not provide the improvement we expected. It includes the conditional GAN and the relativistic average GAN. When we open the bottleneck of the data flow in the generator we expect better results because of the increase in the feature map size, though we got worse results, we assume that the increase in the number feature add more noise to the system and make the learning difficult.

Following our results it can be seen that our CSRGAN is struggling to colorize complicated textures and to perform super resolution. Due to the generator depth and the limitation of resources (small dataset, single GPU) we couldn't perform long training with a big diverse dataset. Stanford dataset contains only 150 images for each dog breed.

We believe that longer training with larger dataset could lead to better results with better texture and colors. In addition, there is more research to do in unique architectures for integrating local features task with global features task to obtain better results.

While most GANs are specific for one task, in this paper we wished to implement a new type of GAN. We presented a network that is capable of colorize and preform super resolution at the same time. These tasks are different and rely on different type of learning(global vs local).While testing different architecture we learned that deep networks are sensitive to small changes, when we used relativistic average loss we suffered from artifacts, when we used RDDB blocks for super resolution we got unrealistic images. We conclude although it's challenging training GAN in order to learn both global and local feature is possible but need to be done with great care and with great research due to the high sensitivity of such networks.

We learned different GANs architectures and optimization methods such as Conditional GAN, relativistic GAN. We learned that U-net network have different variants and even small change in the network architecture can have a major impact on the results.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), pages 2672–2680, 2014.

[2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, ´ A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802, 2016.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations (ICLR), 2015.

[4] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In ACM TOG, volume 21, pages 277–280, 2002.

[5] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In ACM transactions on graphics (tog), volume 23, pages 689–694. ACM, 2004.

[6] P. Isola, J.Y. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. 2016.

[7] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image Colorization using Generative Adversarial Networks 2018.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation 2015.

[9] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets 2014.

[10] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In ECCVW, 2018.

[11] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2):295–307, 2016.

[12] W. Shi, J. Caballero, F. Huszar, 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1874–1883, 2016.

[13] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 262–270, 2015.

[14] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In International Conference on Learning Representations (ICLR), 2016.

[15] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super- resolution. In European Conference on Computer Vision (ECCV), pages 694–711. Springer, 2016.

[16] Stanford dogs dataset-http://vision.stanford.edu/ aditya86/ ImageNetDogs/.

[17] Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In ECCV. (2014)

[18] Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In CVPR. (2016)

[19] Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In CVPR. (2017)

[20] Kim, J., Kwon Lee, J., Mu Lee, K.: Deeply-recursive convolutional network for image super-resolution. In: CVPR. (2016)

[21] . Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR. (2018)

[22] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV. (2018)
[23] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICMR. (2015)
[24] Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In CVPRW. (2017)
[25] Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261 (2016)
[26] Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard gan. arXiv preprint arXiv:1807.00734 (2018)
[27] Richard Zhang, Phillip Isola, Alexei A. Efros Colorful Image Colorization.

Appendix

In the GitHub link there are all training graphs and all the trained models. All code is available at: https://github.com/shay5510/CSRGAN.