SwiftSRGAN - Rethinking Super-Resolution for Efficient and Real-time Inference

Koushik Sivarama Krishnan
Email: koushik.nov01@gmail.com

Karthik Sivarama Krishnan
Email: ks7585@rit.edu

Abstract—In recent years, there have been several advancements in the task of image super-resolution using the state of the art Deep Learning-based architectures. Many super-resolution-based techniques previously published, require high-end and top-of-the-line Graphics Processing Unit (GPUs) to perform image super-resolution. With the increasing advancements in Deep Learning approaches, neural networks have become more and more compute hungry. We took a step back and, focused on creating a real-time efficient solution. We present an architecture that is faster and smaller in terms of its memory footprint. The proposed architecture uses Depth-wise Separable Convolutions to extract features and, it performs on-par with other super-resolution GANs (Generative Adversarial Networks) while maintaining real-time inference and a low memory footprint. A real-time super-resolution enables streaming high resolution media content even under poor bandwidth conditions. While maintaining an efficient trade-off between the accuracy and latency, we are able to produce a comparable performance model which is one-eighth (1/8) the size of super-resolution GANs and computes 74 times faster than super-resolution GANs.

Index Terms—Image Super-Resolution; GANs; Depth-wise Separable Convolutions; Swift-SRGAN; Perceptual Loss Function; MobileNet Loss; Content Loss; Adversarial Loss; up-sampling; PSNR; SSIM;

Fig. 1. Image on the left is the original high-resolution image and the image on the right is the super-resolution image from SwiftSRGAN. The super-resolution(right) image is almost as good as the high-resolution image(left).

I. INTRODUCTION

Image super-resolution has gained the attention of many researchers in the recent days, since the introduction of Convolutional Neural Networks for computer vision based tasks. Especially the reconstruction of a single high resolution image from a low quality low-resolution image has been a key focus among many research communities. Image super-resolution is the process of restoring and reconstructing the resolution of a noisy low-quality image into a very high quality high resolution image. For example, a reconstruction of a 256x256 pixels image into a 1024x1024 pixels image. As shown in Figure 1, this technique is very useful in various applications like up-scaling a low-resolution webcam image to perform facial recognition, up-sampling the medical images for identifying very small anomalies, reconstructing the world-war video feed into current super high resolution 4k frames, etc.

With the recent advancements in deep learning, various newly proposed techniques and architectures like attention[1], transformers and GAN [2] have become de-facto standard. There has been various updates and improvements to these proposed techniques since then and they have found their way into almost all the domains in deep learning. With the increasing advancements in Deep Learning approaches, there is also an increasing demand for high-end computing capabilities to perform computations of such complex neural network architectures[3]. In today's real-world scenario, there is a very high demand for Mobile and embedded vision applications

that performs in real-time. For applications such as Robotics and Wearable mobile devices, Augmented reality etc., the super-resolution tasks need to be performed in real-time with minimal latency and very low footprint requirements.

This paper focuses on improving the latency of the Generative model by reducing the footprint and maintaining an efficient trade-off between the accuracy and the latency. Improving latency is benefited by reducing the computation size and introducing Depth-wise Separable Convolutions in place of standard Convolutions. This results in very few parameters required to be trained.

This technique has a significant application in many fields. When considering the medical image super-resolution, Capturing a very high resolution Magnetic Resonance Imaging (MRI) are complicated when factors like scanning time, spatial convergence and signal-to-noise ratio are being involved. Hence applying the concepts of image super-resolution eliminates these problems by up-scaling the low-resolution image into high resolution images. The same case applies to X-rays [4] and CT scans.

Image super-resolution can also be applied to cloud gaming and media streaming services, where videos can be transmitted at a lower resolution and then up-scaled to a higher resolution on the edge. Streaming very high quality images often comes with a huge cost and high latency that will have an effect on the quality of the game-play. Image super-resolution can also be applied to surveillance and security cameras where the low-resolution footage can be up-scaled to a higher resolution for detecting latent features present in the image frame that could be helpful in identifying suspect in real-time.

Tasks like cloud gaming and media streaming happens

in real-time, so we need a real-time and efficient solution that does not require top-of-the-line GPU to perform super-resolution. Cloud gaming and media streaming are typically processed on a data center outside the consumer's home and directly streamed to their at-home devices. Streaming very high-quality media content often comes with a huge cost and high latency that will affect the quality of the game-play. A real-time super-resolution approach would enable users to view high-resolution media content even on a poor internet connection without compromising on the clarity of the content.

Our approach also enables streaming graphics-intensive video games at low-resolution and then perform super-resolution on them without affecting the FPS(frames per second). We need an approach that works well even on low-end compute devices as top-of-the-line GPUs cannot be used everywhere. Our proposed approach can work even on low-end compute devices at 60 FPS.

II. RELATED WORK

Deep Learning has been growing exponentially in the past few years, with various newly proposed techniques and architectures like attention, transformers[1] and GAN [2] etc,. There has been various updates and improvements to these proposed techniques since then and they have found their way into almost all the domains in deep learning.

A. Depth-wise Seperable Convolutions

Francois Chollet [5] introduced Depth-wise Separable Convolutions. It is a two step operation: Depth-wise Convolution and Point-wise Convolution operations. The depth-wise Convolution is applied to a single image channel at a time, followed by the Point-wise Convolution, which is a 1x1 Convolution operation performed on M image channels. The standard Convolution operation both filters and combines input into a single step. The Depth-wise Convolution splits this into two steps, one for filtering and one for combining them. This results in very few parameters to tweak as compared to the normal Convolutions. They are also computationally cheaper and lighter on the disk.

Since then, several other researchers have adapted this type of Convolution into their architectures. Howard et al. [6] proposed a class of MobileNet architectures that are purely based on the Depth-wise Separable Convolutions. They added two new hyper-parameters, one for the width multiplier and another for the resolution multiplier. Width multiplier thins the network uniformly at each layer. For a selected layer with M channels and width multiplier α, the number of input channels becomes αM and output channels become αN. The second hyper-parameter resolution multiplier ρ, is used to reduce the dimensions of the input image.

B. GANs

Ian et al. [2] introduced the now, well known Generative models in 2014 that are trained simultaneously with discriminator model through adversarial process. The Generator tries to create fake images and the discriminator critics these images. As they progress through the training, the generator finally produces images that are indistinguishable from the real ones by the discriminator. Since then, many variations and updates has been made to this architecture for applications in different fields.

C. CNN based approaches

1) SRCNN: Chao et al. [7] introduced a Convolution based network architecture named as SRCNN that is structured into three stage process: Patch Extraction and Representation - this operation extracts overlapping patches from the low-resolution image and projects it into a high dimensional vector space. Non Linear Mapping - this operation non-linearly maps the resultant high dimensional vector from the previous operation onto another high dimensional vector. This results in a high resolution patch. Reconstruction - the generated high resolution patch is then aggregated to obtain the super-resolution output image. This proposed architecture achieves 32.52dB PSNR.

2) DRCN: Jiwon et al. [8] introduced a recursion based Convolution approach that used upto 16 recursions. And the main advantage of this approach is that, increasing the number of recursion can largely improve performance without introducing new parameters. They also observed that this approach is really hard to train with traditional gradient descent methods due to vanishing and exploding gradients. To tackle this problem, they introduced recursive-supervision and skip-connections to the network.

3) ESPCN: Wenzhe et al. [9] proposed a CNN architecture where the feature maps are extracted from the low-resolution images rather than the high-resolution images. They also proposed an efficient sub-pixel level Convolution operation that learns various up-sampling filters to up-scale the low-resolution feature maps into high resolution images. This also reduced the complexity of the super-resolution operation.

D. GAN based approaches

1) SRGAN: Christian et al. [10] introduced a Generative Adversarial Network-based architecture and proved this approach was better than using traditional methods previously proposed for image super-resolution. They introduced the Perceptual loss function based on the Mean Squared Error loss, which helped in removing the over-smoothing effects of MSE loss.

2) ESRGAN: Xintao et al. [11] modified the SRGAN[10] architecture by introducing Residual-in-Residual Dense Block for easier training. They have also improved the Discriminator architecture by using Relativistic Average GAN (RaGAN) [12], which learns to find the more realistic image rather than distinguishing the fake image.

3) DSCSRGAN: Zetao et al. [13] proposed the use of Depth-wise Separable Convolutions in the intermediate layers and regular Convolutions in the first and last blocks. They used regular Convolutions for the discriminator and removed batch normalization from them. They also proposed a new loss function, frequency energy similarity loss function, which converts the spatial domain of the image into frequency domain. Then they calculate the similarity of frequency domain energies between the super-resolution image and the high-resolution target image.

III. METHODOLOGY

A. Dataset

To train this network, we merged two commonly available standard super-resolution datasets. DIV2K[14] [15] dataset and the Flickr2K [16] dataset. DIV2K[14] [15] contains 800 high-resolution images for training, 100 for validation, and 100
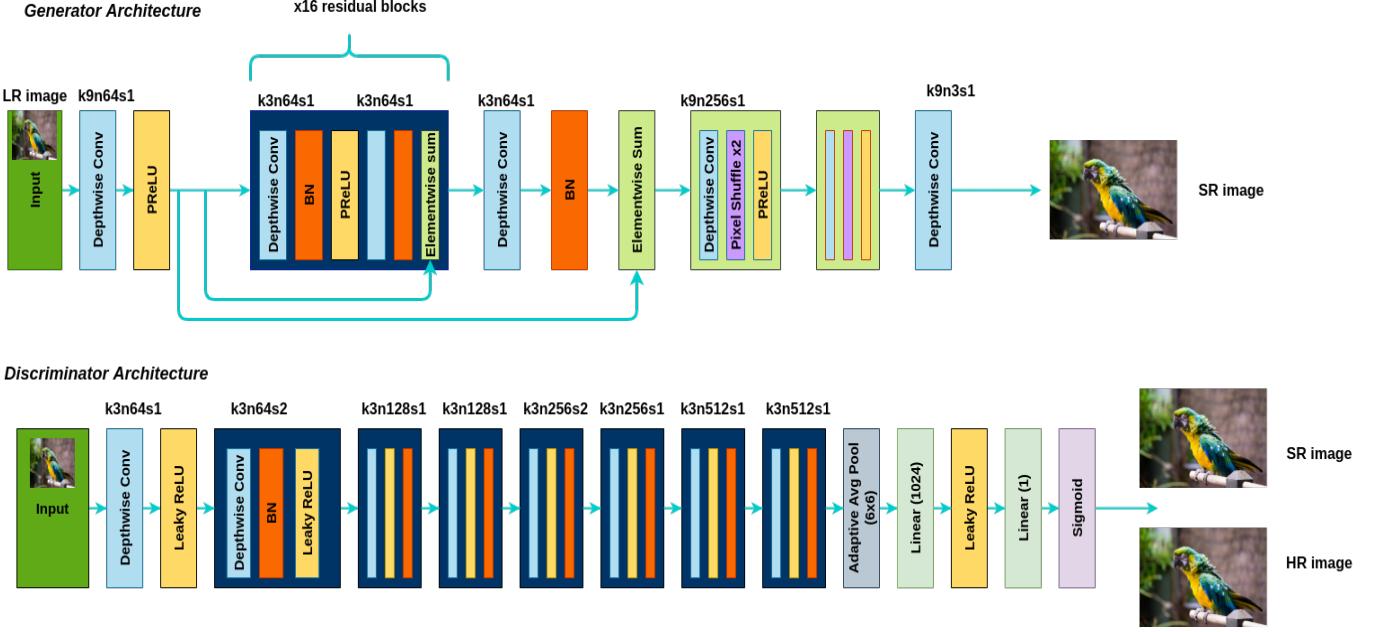
Fig. 2. Architecture diagram of the Generator and Discriminator where 'k' is the kernel size, 'n' is the number of output channels, 's' is the stride.

Split	Number of High-Resolution images
Train	800 (DIV2K) + 2650 (Flickr2K)
Validation	100
Test	100 + 5 (Set5) + 14 (Set14)
Total	**3669**

for testing. The Flickr2K [16] dataset has 2650 high-resolution images collected from the Flickr website. We used Set5[17] and Set14[17] test sets for evaluating our model. Set5[17] test set has 5 high-resolution images and Set14[17] has 14 high-resolution images in it.

B. Dataset Preprocessing

The high resolution images are augmented using the albumentations [18] library. This helps in making distinctive images that helps the deep learning model to learn much better and prevent overfitting. The high-resolution images are randomly cropped into 1024x1024 images. A random horizontal flip and random rotation to 90° is performed on these cropped images. This is then down-sampled using the bicubic operation to get low-resolution 256x256 images.

C. Network Architecture

The generator architecture as shown in Figure 2 consists of Depth-wise Separable Convolutions followed by Batch Normalization and PReLU as the activation function. Since our main goal is to perform real-time inference even on low-end GPUs, we have replaced regular Convolutions with Depth-wise Separable Convolutions. This has shown a significant reduction in inference time. The generator consists of 16 residual blocks with Depth-wise Convolution, followed by batch normalization and PReLU as activation function. This

is followed by another Depth-wise Convolution, batch normalization, and lastly, an element-wise sum operation to sum the previous block's output and current output. We then up-sample the image by passing it through the up-sample Block twice. The up-sample Block contains Depth-wise Separable Convolutions followed by two Pixel Shuffle layers and PReLU as the activation function. This is then passed into the final Depth-wise Separable Convolution layer with a kernel size of 9, number of output channels as 3, and a stride of 1.

The design is based on standard SRGAN [10] architecture, and expands by the introduction of Depth-wise Separable Convolutions which benefits for faster inference and parameter efficiency. Given an input low-resolution image of dimensions 3x256x256, the generator outputs a 3x1024x1024 super-resolution image instantaneously.

The discriminator architecture consists of 8 Depth-wise Separable Convolution blocks. All blocks have a Depth-wise Separable Convolution, followed by batch normalization and LeakyReLU (negative slope=0.2) as activation function. The first block does not have a batch normalization layer. The output of the last Convolution block is passed to the adaptive average pooling layer that gives an output size of 6x6. The output of the adaptive average pooling layer is flattened and passed into a linear layer of 1024 neurons. The main objective of the discriminator is to classify super-resolution images as fake and high-resolution images as real.

D. Perceptual Loss Function with MobileNet

The perceptual loss function was introduced by Christian et al. [10] which is based on the MSE loss. As our goal was to reduce both training and inference time, We improved on this loss function by replacing VGG [19] network with MobileNetV2 [20] for efficient memory usage and training time. Since the definition of the perceptual loss function is very crucial for the performance of the generator, we only tweaked the network used. The perceptual loss function is formulated

as the weighted sum of adversarial loss and, the content loss as:

$$\underbrace{l^{SR}}_{\text{Perceptual loss}} = \underbrace{l^{SR}_X}_{\text{content loss}} + \underbrace{1e-3l^{SR}_{Gen}}_{\text{adversarial loss}} \quad (1)$$

1) Content Loss: We build upon the proposed content Loss by Christian et al. [10] by replacing VGG19 [19] with MobileNetV2 [20]. We define this MobileNet Loss based on the ReLU6 activation layers of the pre-trained network. ϕ_i denotes the output of i-th Convolution block after activation. We used the output of the 16 th block for our loss function i.e. i = 16. This MobileNet loss is defined as the euclidean distance between the target image I^{HR} and the output feature map of the MobileNet loss $G_{\theta_G}(I^{LR})$.

$$l^{SR}_{MobileNet} = \frac{1}{W_i H i} \Sigma^{W_i}_{x=1} \Sigma^{H_i}_{y=1} (\phi_i(I^{HR})_{x,y} - \phi_i(G_{\theta_G}(I^{LR}))_{x,y})^2 \quad (2)$$

Here W_i and H_i are feature map dimensions within the MobileNetV2 [20] network.

2) Adversarial Loss: We also add adversarial loss to our content loss. This encourages our network to produce images that are close to natural images, by trying to fool the discriminator. This generative loss is formulated as:

$$l^{SR}_{Generator} = \Sigma^N_{n=1} - \log D_{\theta_D}(G_{\theta_G}(I^L R)) \quad (3)$$

Where $D_{\theta_D}(G_{\theta_G}(I^L R))$ is the probability that the super-resolution image is a natural High-resolution image. And, instead of reducing $[1 - D_{\theta_D}(G_{\theta_G}(I^L R))]$, we minimize $-\log D_{\theta_D}(G_{\theta_G}(I^L R))$ for better gradients [21].

IV. EXPERIMENTS AND ANALYSIS

A. Experimental Setup

We trained our Swift-SRGAN on Tesla P100 GPU with high-resolution images of dimensions 1024x1024x3 and, low-resolution images of dimensions 256x256x3. We used the AdamW [22] optimizer with ReduceLROnPlateau learning rate scheduler and a batch size of 32. We also used the PyTorch's mixed-precision support to train our models. The low-resolution images are passed into the generator to get the super-resolution image. This is then passed into the perceptual loss function and the generator loss is calculated. This perceptual loss is simply adversarial loss + content loss. Then both high resolution and super-resolution images are passed into the discriminator to calculate the discriminator loss.

B. Evaluation Criteria

In this study, we used Peak-Signal-to-Noise-Ratio(PSNR) and Structural Similarity Index Measure (SSIM) as scoring metrics. PSNR is simply a ratio between maximum power of image to power of noise that affects the image quality. The higher the PSNR, better the super-resolution output image when compared to the high resolution target image.

$$PSNR = 10 \cdot \log_{10}(\frac{MAX_1^2}{MSE}) \quad (4)$$

$$= 20 \cdot \log_{10}(\frac{MAX_1}{MSE}) \quad (5)$$

$$= 20 \cdot \log_{10}(MAX_1) - 10 \cdot \log_{10}(MSE) \quad (6)$$

Structural Similarity Index Measure(SSIM) is a perceptual image metric that quantifies image degradation between the

Fig. 3. Low-resolution Bicubic image (top), Super-resolution image (middle) and, High-resolution image (bottom). As you can see from the images, our model has reproduced all the lighting, reflections (in the eye) and texture details seen on the high resolution image.

TABLE II
PERFORMANCE ANALYSIS

Test Set	Model	PSNR	SSIM
Set5	ESRGAN	32.7	0.9011
Set5	SRGAN	29.40	0.8501
Set5	**Swift-SRGAN**(ours)	25.13	0.7940
Set5	ESRGAN	28.7	0.7917
Set14	SRGAN	26.02	0.7397
Set14	**Swift-SRGAN**(ours)	23.29	0.7012

two images. It is calculated based on 3 comparison measurements: structure, luminance and contrast. This metric lies between 0.0 to 1.0 where 1.0 signifies perfect replica of the high resolution image.

Luminance comparison function,

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2\mu_y^2 + c_1} \quad (7)$$

Contrast comparison function,

$$l(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2\sigma_y^2 + c_2} \quad (8)$$

Structure comparison function,

$$l(x,y) = \frac{2\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (9)$$

and,

$$c3 = c_2/2 \quad (10)$$

SSIM is the weighted combination of these measures,

$$SSIM(x,y) = [l(x,y)^\alpha \cdot c(x,y)^\beta \cdot s(x,y)^\gamma] \quad (11)$$

V. RESULTS

To evaluate the performance of our model, we compared the inference time of our proposed approach to the inference time of the SRGAN[10] architecture on Tesla K80 GPU. From the table II, we can clearly see that our proposed architecture has outperformed SRGAN[10] and ESRGAN[11] by a large magnitude in inference time. The inference time is calculated

TABLE III
COMPARISON OF INFERENCE TIME BETWEEN SRGAN AND
SWIFT-SRGAN

Upsampling Resolution	Model	Inference time per-frame (in ms)
270p to 1080p	SRGAN	812
270p to 1080p	ESRGAN	974
270p to 1080p	**Swift-SRGAN**(ours)	**5.605**
540p to 4K	SRGAN	1210
540p to 4K	ESRGAN	1330
540p to 4K	**Swift-SRGAN**(ours)	**16.18**

by the amount of time it takes to upsample a single image frame (in ms). Our proposed architecture achieves a swift inference time of about 5.605 ms to up-sample a single image while the SRGAN[10] took 812 milliseconds to up-sample the same. This result signifies that our proposed architecture can up-sample 100 frames a second which is upto a 100 times faster than SRGAN[10] and ESRGAN[11]. Hence our proposed method achieves an efficient up-sampling of videos and images on the fly.

We also compared the PSNR and the SSIM scores of ESRGAN[11], SRGAN[10] and Swift-SRGAN(ours) on the standard benchmark testing datasets Set5[17] and Set14[17]. These two test sets has diverse set of images that truly signifies the performance of these models. As we can see from table III, our approach has achieved comparable performance to the others. Our proposed architecture achieves PSNR of 25.13 and SSIM of 0.794 on Set5[17] and PSNR of 23.29 and SSIM of 0.701 on Set14[17] which is very similar to other architectures while also maintaining real-time inference.

Figure 3 depicts the comparison between the high-resolution, low-resolution and our SwiftSRGAN's super-resolution output. The resultant super-resolution image is comparable to the high resolution image and reproduces the lighting and reflections entirely while also maintaining the color accuracy.

VI. DISCUSSION

The proposed approach uses Depth-wise Separable Convolutions that has comparable performance with regular convolutions while also maintaining very low memory footprint and real-time inference. Depth-wise Separable Convolutions has very less number of trainable parameters compared to regular Convolutions. This reduces the size of the model and, number of computations during training and inference time. We used MobileNet instead of VGG in calculating loss which results in faster training time. While previously proposed approaches tend to focus more on the performance, we propose a real-time solution that can be run on low-end compute devices.

VII. CONCLUSION AND FUTURE WORK

Image super-resolution is a challenging task and, many researchers have proposed architectures for this application which performs well requiring very high computational capabilities and are not real-time. We propose a real-time efficient image super-resolution architecture that has comparable trade-off results with the previously proposed architectures and has the lowest inference time.

Though this model achieves efficient real-time inference on low-end GPU, there is still scope for improving it's super-resolution performance. Our training data is relatively small when compared to other approaches that used 50,000 images from the ImageNet [23] database for training. We are also planning to extend our work by improving our proposed network's performance.

REFERENCES

[1] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[2] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[3] Karthik Sivarama Krishnan and Ferat Sahin. "ORB-DeepOdometry - A Feature-Based Deep Learning Approach to Monocular Visual Odometry". In: *2019 14th Annual Conference System of Systems Engineering (SoSE)*. 2019, pp. 296–301. DOI: 10.1109/SYSOSE. 2019.8753848.

[4] Koushik Sivarama Krishnan and Karthik Sivarama Krishnan. "Vision Transformer based COVID-19 Detection using Chest X-rays". In: *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*. 2021, pp. 644–648. DOI: 10.1109/ ISPCC53510.2021.9609375.

[5] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357.

[6] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: 1704.04861 [cs.CV].

[7] Chao Dong et al. "Image Super-Resolution Using Deep Convolutional Networks". In: *CoRR* abs/1501.00092 (2015). arXiv: 1501.00092. URL: http://arxiv.org/abs/ 1501.00092.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Deeply-Recursive Convolutional Network for Image Super-Resolution". In: *CoRR* abs/1511.04491 (2015). arXiv: 1511.04491. URL: http://arxiv.org/abs/1511. 04491.

[9] Wenzhe Shi et al. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *CoRR* abs/1609.05158 (2016). arXiv: 1609.05158. URL: http://arxiv.org/abs/ 1609.05158.

[10] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *CoRR* abs/1609.04802 (2016). arXiv: 1609. 04802. URL: http://arxiv.org/abs/1609.04802.

[11] Xintao Wang et al. "ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks". In: *CoRR* abs/1809.00219 (2018). arXiv: 1809.00219. URL: http://arxiv.org/abs/1809.00219.

[12] Alexia Jolicoeur-Martineau. "The relativistic discriminator: a key element missing from standard GAN". In: *CoRR* abs/1807.00734 (2018). arXiv: 1807.00734. URL: http://arxiv.org/abs/1807.00734.

[13] Zetao Jiang, Yongsong Huang, and Lirui Hu. "Single Image Super-Resolution: Depthwise Separable Convolution Super-Resolution Generative Adversarial Network". In: *Applied Sciences* 10.1 (2020). ISSN: 2076-3417. DOI: 10.3390/app10010375. URL: https://www.mdpi.com/2076-3417/10/1/375.

[14] Eirikur Agustsson and Radu Timofte. "Ntire 2017 challenge on single image super-resolution: Dataset and study". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 126–135.

[15] Radu Timofte et al. "Ntire 2017 challenge on single image super-resolution: Methods and results". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 114–125.

[16] Bee Lim et al. "Enhanced deep residual networks for single image super-resolution". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 136–144.

[17] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. "Single Image Super-Resolution From Transformed Self-Exemplars". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5197–5206.

[18] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[19] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[20] Mark Sandler et al. "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation". In: *CoRR* abs/1801.04381 (2018). arXiv: 1801.04381. URL: http://arxiv.org/abs/1801.04381.

[21] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[22] Ilya Loshchilov and Frank Hutter. "Fixing Weight Decay Regularization in Adam". In: *CoRR* abs/1711.05101 (2017). arXiv: 1711.05101. URL: http://arxiv.org/abs/1711.05101.

[23] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.