Single Image Super-Resolution Using a Channel Attention Generative Adversarial Network

Siyuan Yan
Australian National University
u7050317
u7050317@anu.edu.au

Gefei Zhang
Australian National University
u7138112
u7138112@anu.edu.au

Hao Wang
Australian National University
u7195922
u7195922@anu.edu.au

Abstract

Despite the recent development in Single Image Super-Resolution (SISR) tasks attributed to deeper neural networks, some problems remain to solve: (i) we have not exploited the full potential of low-level and high-level features on SISR tasks. (ii) Many previous methods treat equally among different channels, while the input low-resolution (LR) image contains mostly useless low-frequency information and less useful high-frequency information. (iii) MSE loss function is not suitable for SISR tasks as they are optimized to compute the pixel-wise error on images, causing overly smoothed results. In this paper, we proposed our novel network called Channel Attention GAN (CAGAN). Residual blocks are applied to our method to fuse low-frequency and high-frequency feature information. Besides, our Channel Attention block can capture both long-range pixel information by a global pooling operation and important channel information by the channel attention mechanism. Meanwhile, we replace the MSE loss with VGG loss during network training and get perceptually better visual results. Finally, GANs are unstable and hard to train. We apply spectral normalization and the imbalanced learning rate to tackle it. Our Channel Attention GAN model outperforms the state-of-the-art baseline SRGAN method and Self Attention GAN (SAGAN) on three benchmark datasets, which significantly improved PSNR, SSIM, and MOS scores (representing perceptually better visual results).

1. Introduction

Single Image Super-Resolution (SISR) is targeted at producing a high-resolution (HR) image from its low-resolution (LR) counterpart. This task can be applied in many areas like old movie reconstruction, magnetic resonance imaging (MRI) [18], and etc. In recent years, deep learning is gaining much popularity because it is achieving marvelous performance. Dong *et al*. [3, 4] present a lightweight convolutional network structure, achieving good performance on SR tasks. However, SR tasks with highly upscaling factor (like $4\times$) are still challenging. The difficulty is the reconstructed SR image loses the texture information. Some researchers solved the drawbacks of traditional CNN hardly produce high-frequency details by Generative Adversarial Network(GAN) [5]. SRGAN [14] is the first GAN based model that can produce photo-realistic images. However, it generates details by only taking local information and does not consider the channel relationship, which fails to learn consistent information in long-range and capture high-frequency details.

In the report, we present a novel Channel Attention GAN to tackle these existing problems and verify that our model outperforms SRGAN and SAGAN on three benchmark datasets. An example of our model's effect is shown in Figure 1. The main contributions of our work can be summarized below as:

- We use a novel Channel Attention block that uses a global average pooling operation to get global information and then use channel attention to capture the interdependencies among different channels.
- We use the skip-connection [6] to bypass low-frequency information in the first few layers to deep layers.
- We use a VGG loss that computes the pixel-wise difference in feature maps of a pretrained VGG19 network among with an adversarial loss to produce perceptually better results.
- We use spectral normalization and imbalanced learning rate in our model, which has been shown that they can produce better results and stabilize training.

2. Related work

Residual network. Many prior CNN based super-resolution algorithms showed good performance. Basic

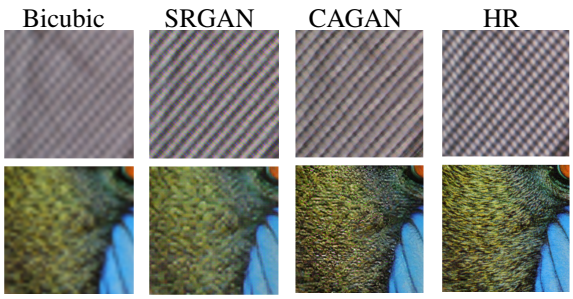

Figure 1. Super-resolution results of our proposed CAGAN model, compared with SRGAN model. We can see that our CAGAN performs better than SRGAN [14] in high-frequency regions.

convolutional network structure accomplished SR tasks with less satisfying results [3, 4]. Tremendous works tried to improve the network's representation ability to recover high-frequency details. Theoretical speaking, deeper neural network trends to have better performance. However, due to gradient vanishing, it is difficult to train a very deep neural network. Residual block [8] tries to solve the gradient vanishing and exploding problem in deep neural networks. When it comes to the SR problem, we found the skip connections in residual networks valuable in another way. Skip connections can bypass low-level features into high-level features in deep layers, providing more detailed texture information. Ledig *et al*. [14] proposed SRResNet, which introduced residual blocks into SR tasks, and it achieved very high PSNR and MOS scores in the SR tasks. Residual blocks are also used in our network to facilitate our deep network training. Skip connection makes the network bypass low-frequency information and makes the network focus on more high-frequency information.

Attention mechanism. Attention mechanism was first invented to capture the context relationships for long sentences in the Nature Language Processing field [1]. The notion of attention showed its advantages after experimenting on SR tasks. Zhang *et al*. [24] presented a CNN network with attention blocks for the SR task, their channel attention blocks with two convolutional layers generated a channel attention map that can reflect each channel's importance. Channel attention mechanisms considered the interdependencies among feature channels, while other researchers explored the application of attention mechanisms on spatial relationships. Self-attention mechanism [23] abstracted inter-pixel attention maps to reflect spatial relationships. We briefly compared the influence of these two kinds of attention mechanisms on SR tasks and adapted the channel attention mechanism into our model to capture more useful information.

Generative Adversarial Network. Generative Adversarial Network (GAN) [5] is generated from the notion of adversarial training. It consists of a generator and a discriminator. Generator and Discriminator are trained competitively, resulting in a better performance compared to a solely CNN network. SRGAN [14] adapted Generative Adversarial Networks (GAN) into the SR model, which showed its superiority concerning SRResnet [14]. Our method is based on GAN architecture rather than solely CNN structure, which can produce perceptually better visual results.
Mode collapse is a common problem that GAN often encounter. Miyato *et al*. [17] used spectral normalization that could stabilize the training process of the GAN based model. Imbalanced learning rate [9] used different learning rates for the generator and discriminator, which produced better results. We implemented spectral normalization and an imbalanced learning rate to avoid model collapse.

VGG loss. Traditionally, Mean Square Error (MSE) was used as an objective function that produces images with high PSNR scores. However, MSE-optimized images lack high-frequency details, such as textures. SRGAN [14] used a loss function called VGG loss. It has been proven to produce images with better perceptual results. We adapt the VGG loss rather than MSE loss as our generator's content loss function to produce perceptually better images.

3. Method

Super Image Super-Resolution (SISR) aims to estimate a high-resolution image I^{SR} from its corresponding low-resolution input I^{LR}. In our project, we downscale the original high-resolution image I^{HR} with a factor of k[1] to obtain I^{LR}. So, given an input image I^{LR} that contains C channels, its size is $C \times W \times H$, the output of our model is I^{SR} with a size of $C \times kW \times kH$. The objective of our model is to learn a function G that minimize the error between I^{SR} and I^{HR}. The function can be expressed as

$$\arg\min_{\theta_G} \sum_{m=1}^{M} \mathcal{L}_{total}(G_{\theta_G}(I_m^{LR}), I_m^{HR}) \tag{1}$$

where G_{θ_G} denotes the generator of our model, θ_G denotes all parameters of the generator, M is the total number of training sets, and $\mathcal{L}_{total}$ is our total loss, which will be discussed in detail in section 3.3.

3.1. Architecture of our network

Following Goodfellow *et al*. [5], we develop a GAN based model. Our model consists of two parts, a generator to produce the high-resolution super-resolved image I^{SR},

[1] We set the scale factor k to be 4 for all our experiments.

which is similar to traditional super-resolution models, and an additional discriminator to distinguish the I^{SR} produced from the original I^{HR}. Based on the adversarial training strategy, our model can produce more realistic images as the discriminator will give suggestions to the generated image, and the generator will try to produce images that can fool the discriminator. During the training process, our model will spend more time because of the additional discriminator model. But during the test and evaluation part, our model has the same inference time as traditional models because the discriminator part can be removed.

For our generator, it starts with a convolution layer with a large kernel size 9 and 64 feature maps followed by spectral normalization [17], batch normalization [11], and ParametricReLU [7]. We define the architecture as a convolution block. The depth of the network is a crucial factor in the SR tasks. However, a very deep network is hard to train and suffers from gradient vanishing or gradient exploding. To solve the problem and do better feature fusion, our model then uses 16 residual blocks that each block contains two convolution blocks just mentioned before but with a kernel size of 3, and the residual block [6] is used for each residual block. Further, we use an additional skip-connection to fuse low-level features at the start of the first residual block and high-level features at the end of the 16th residual block. It can make our model bypass low-frequency feature information, forcing the network to focus on more useful high-frequency feature information. Finally, we use two sub-pixel convolution layers [19] followed by a convolution layer to recover the encoded feature maps and output the high-resolution super-resolved images.

The discriminator uses a similar structure to the SRGAN [14], which contains 8 convolution blocks but uses LeakyReLU ($\alpha = 0.2$) as the activation function. The difference is we change the final two dense layers to two convolution blocks with LeakyReLu and a convolution block without any activation functions. The advantage of this change is that it dramatically decreases the number of parameters, and we find that it will not change our model's performance in practice. Table 3.1 shows the comparison of parameters in the last few layers in two methods. Method 1 is the discriminator in the original SRGAN, and method 2 is the discriminator in our channel attention GAN.

Besides, we add channel attention blocks to all 16 residual blocks of the generator and 8 convolution blocks of the discriminator to capture the channel relationship among channels.

method 1	fc1	fc2	-	Total
	18874368	1024	-	18875392
method 2	conv1	conv2	conv3	Total
	589824	36864	288	**626976**

Table 1. Number of parameters comparison for the last few layers of the network

3.2. Channel attention block

SRGAN has two main disadvantages. The first disadvantage lies in the shallow layers of the neural network. Most channels in the shallow layers have abundant low-frequency feature information; however, high-frequency information is possessed by only a few channels. SRGAN treats equally among different channels, which results in the loss of high-frequency information and an abundance of low-frequency information in feature maps. The low-frequency information is not very useful for SR tasks and limits the representative ability of the model. The second is that SRGAN produces high-quality textures and details by only taking local information using convolution operation, which fails to learn information outside its local receptive field. This problem is also the drawback of most CNN based models.

Inspired by the Squeeze and Excitation networks (SENet) [10] and the very deep residual channel attention networks (RCAN) [24]. They use the channel attention mechanism to selective emphasize informative channels with mostly high-frequency feature information and suppress less useful channels with mostly low-frequency feature information. Based on their ideas, we use a channel attention block to improve the original SRGAN further. Our channel attention block is shown in Figure 3. It first aggregates each channel's global spatial information into a scalar to get a large receptive field and then captures long-range dependency between pixels. The formula can be expressed as:

$$H_{GP}(x_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{2}$$

where $x_c(i, j)$ is the value at position (i, j) of the c-th feature map, H and W are the height and width of the input feature map x_c , and H_{GP} is the global average pooling function. Then, we try to fully capture the interdependencies among channels with the help of the aggregated information. It must satisfy two criteria: Firstly, it must learn the non-linear relationship between channels. Secondly, it must learn a non-mututually-exclusive relationship as we hope multiple channels can be emphasized rather than only enforcing one channel. To achieve these objectives, we use a gating mechanism. The formula can be expressed as

$$s = f(W_u \delta(W_D z)) \tag{3}$$

where f(·) and $\delta(\cdot)$ are the sigmoid and ReLU function, respectively. W_D is the weight of the convolution layer, and the layer will reduce the channel dimension of z with a downscaling factor of r. After being activated by a ReLU function, a convolution layer with the upscaling factor of r will recover the channel's dimension where its weights are W_U. After applying the sigmoid function, we get the results. Our channel attention block's output is gotten by

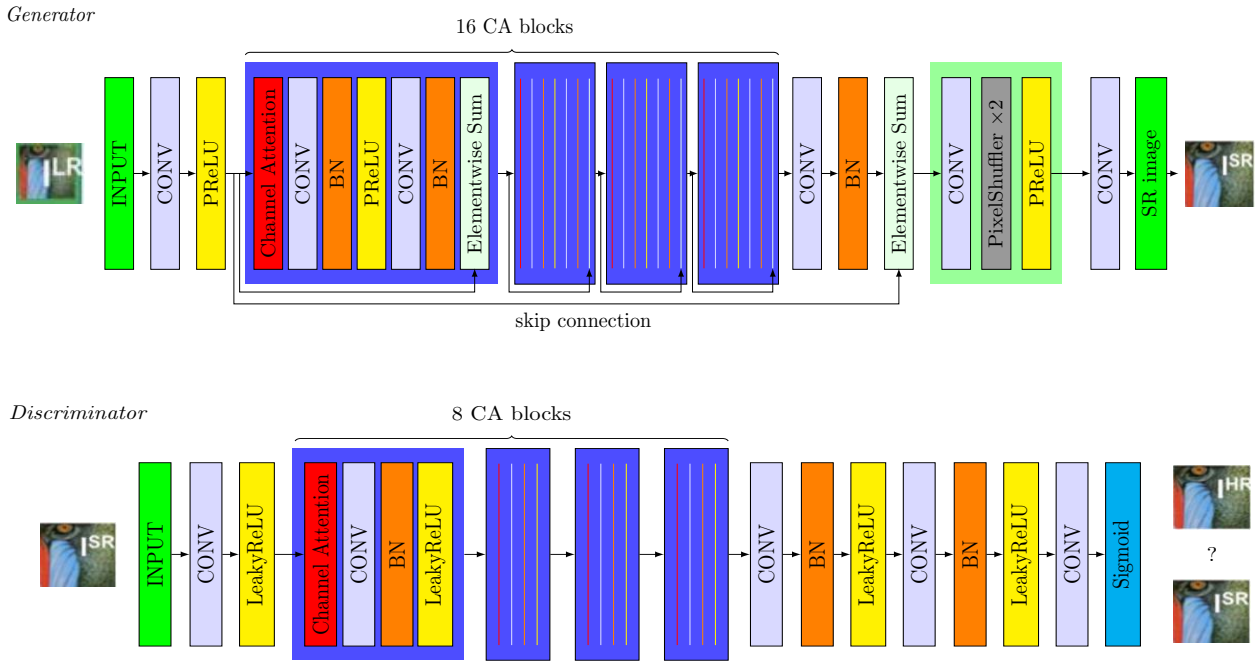

Figure 2. The architecture of our network

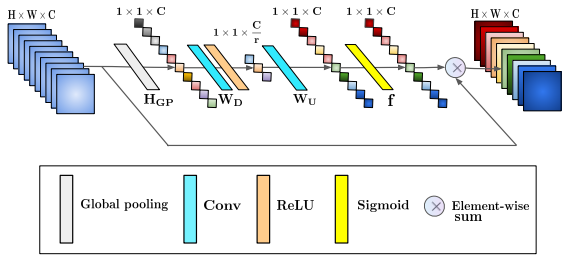

Figure 3. :The structure of our channel attention block

rescaling x_c with s_c, which is an adaptive recalibration process. The formula is expressed as

$$\hat{x_c} = s_c \cdot x_c \tag{4}$$

3.3. Adversarial training strategy

Our training strategy can be summarized by optimizing the min-max problem:

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D}&\mathbb{E}_{I^{HR}\sim p_{data}(I^{HR})}[logD_{\theta_D}(I^{HR})]+\\&\mathbb{E}_{I^{LR}\sim p_G(I^{LR})}[log(1-D_{\theta_D}(G_{\theta_G}(I^{LR})))]\end{aligned} \tag{5}$$

where G_{θ_G} is the generator and D_{θ_D} is the discriminator. The generator will try to produce I^{SR} to fool the discriminator and make it unable to distinguish the SR image from the HR image. And the discriminator tries to classify reference image I^{HR} to be real and classify generated image I^{SR} to be fake. By the adversarial training strategy, the output images will be perceptually superior and highly similar to real images.

Perceptual VGG loss. Our generator is trained with a content MSE loss $\mathcal{L}_{content}$ and an adversarial loss $\mathcal{L}_{adv}$ that is back-propagated from the discriminator. The total loss of the generator is

$$\mathcal{L}_G = \mathcal{L}_{content} + \lambda_G \mathcal{L}_{adv} \tag{6}$$

where $\mathcal{L}_{content}$ is the content loss, $\mathcal{L}_{adv}$ is the adversarial loss that is back-propagated from the discriminator, and λ_G is a hyper-parameter that is used to control the weights of $\mathcal{L}_{adv}$. In the following paragraph, we will discuss content loss and adversarial loss in detail.

Content loss. Many traditional SR models [4, 19] often choose MSE loss that computes the pixel-wise error between the original image I^{HR} and the generated image I^{SR}. These models can produce images that have very high PSNR scores but lack high-frequency detail information and textures. The reason is the pixel-wise MSE loss makes the result overly smooth.

Following Simonyan and Zisserman [20] and Ledig *et al.* [14], VGG loss has been proven useful on transfer learning, and SR tasks due to it can produce better perceptual

Method	Set5			Set14			BSD100		
	PSNR	SSIM	MOS	PSNR	SSIM	MOS	PSNR	SSIM	MOS
Bicubic	24.874	0.801	1.96	24.638	0.659	1.74	23.739	0.625	1.58
SRGAN	26.304	0.825	3.46	24.269	0.690	3.56	23.126	0.634	3.53
Self Attention	27.135	0.828	3.62	25.474	0.727	3.57	23.599	0.646	3.61
Channel Attention	**30.757**	**0.869**	**3.87**	**26.455**	**0.743**	**3.76**	**25.572**	**0.686**	**3.79**

Table 2. Performance comparisons for different models

results. Our VGG loss uses the j-th convolution layers' feature map after being activation, while before the i-th max-pooling layer within the VGG network, which is defined as $\beta_{i,j}$. And then, we compute the MSE loss between the feature maps of the original image I^{HR} and the super-resolved image $G_{\theta_G}(I^{LR})$. So, the content loss can be expressed as

$$\mathcal{L}^{SR}_{VGG_{i,j}} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\beta_{i,j}(I^{HR})_{x,y} - \beta_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \quad (7)$$

where $W_{i,j}$, and $H_{i,j}$ are the dimensions of feature maps with the VGG network.

Adversarial loss. We also add an adversarial loss to Perceptual VGG loss. It pushes the generator to produce better visual results by fooling the discriminator. The equation can be expressed as:

$$\mathcal{L}_{adv} = \sum_{n=1}^{N} -logD_{\theta_D}(G_{\theta_G}(I^{LR})) \quad (8)$$

Here, $G_{\theta_G}(I^{LR})$ denotes the generated image and $D_{\theta_D}(G_{\theta_G}(I^{LR})$ denotes the probability that the generated image is the original image I^{HR}.

3.4. Techniques to stabilize the training of GANs

As GAN based model is unstable and difficult to train, we research some stabilization strategies on our channel attention GAN model. We use the spectral normalization [12] in both the generator and the discriminator network. We also adapt the imbalanced learning rate [12] into our model to balance the training dynamic between the two networks.

Spectral normalization. Many previous works like [12] only use spectral normalization to the discriminator. We choose the same strategy as the self-attention GAN [23], which uses the spectral normalization to the two models. It can save the training time by reducing the discriminator update per generator update. Also, the spectral normalization can perform well without extra parameters, and the computation cost is very small. We empirically prove that the spectral normalization can improve the results and stabilize the training. The detail of our ablation study will be discussed in section 4.2.

Imbalanced learning rate. In practice, GANs often need more discriminator update steps per generator update step during the training process. To address it, Heusel *et al*. [9] use a different learning rate (TTUR) for both the generator and the discriminator. We use a larger discriminator learning rate (0.0004) and a smaller generator learning rate (0.0001) to make the discriminator updates fewer steps per generator update step. The imbalanced learning rate can avoid mode collapse and produce better results in the same training time.

4. Results

Dataset. The COCO Dataset [15] contains over 200,000 images. Our model was trained on 82k images from COCO train2014 dataset. We evaluated our model on 3 benchmark datasets: Set5 [2], Set14 [22], and BSD100 [16]. The images in the COCO training set are distinct from those in test sets, so we can fairly prove the results' effectiveness.

Evaluation metrics. We used three evaluation metrics to quantitatively reflect the results of our model: PSNR, SSIM [21], and MOS. PSNR means peak signal-to-noise ratio, SSIM means structural similarity index. These two metrics are evaluated on y-channel center-cropped, removing 4-pixels border images just like SRGAN [14]. MOS is the mean opinion score. We found 15 raters and asked them to do a survey on google forums to collect their feedback for images produced by different models. Raters rated the images based on image quality, and they scored from 1 (bad quality) to 5 (exceptional quality). MOS score is calculated as the mean of all raters' scores. In general, the three metrics will be high when the image quality is high. But we think both PSNR and SSIM cannot totally reflect the image quality; we prefer to use the MOS score to reflect the perceptual visual result. And we will discuss the reason later in section 4.2.

Training configurations and details. We trained our network on NVIDIA GTX 2080 Ti and NVIDIA GTX 2070 Super. We get low-resolution (LR) images by downsampling the high-resolution (HR) images using a bicubic interpolation with a scale factor of 4. The LR images are the input of our generator without further data augmentation. For the optimizer, both of our generator and discriminator

Figure 4. Visualization Comparisons with our CAGAN: SRGAN is our baseline model. SAGAN is the Self Attention GAN. CAGAN is our proposed Channel Attention GAN

Number of CA	Set5			Set14			BSD100		
	PSNR	SSIM	MOS	PSNR	SSIM	MOS	PSNR	SSIM	MOS
0	27.504	0.828	3.67	25.269	0.660	3.62	23.416	0.674	3.63
4	30.742	0.854	3.77	26.384	0.726	3.67	25.441	0.670	**3.78**
8	**30.811**	0.856	3.78	26.333	0.739	3.63	25.519	0.672	3.70
16	30.757	**0.869**	**3.91**	**26.455**	**0.743**	**3.76**	**25.572**	**0.686**	**3.78**

Table 3. Comparisons for different numbers of CA blocks

are optimized with Adam optimizer [13] with parameter: $\beta = 0.9$. We also apply the imbalanced learning rate to the generator and the discriminator. The generator is trained with a learning rate of 1e-4; the discriminator is trained with a learning rate of 4e-4. Further, we use the spectral normalization to stabilize GAN's training. For the channel attention blocks, we added 16 blocks into the generator and 8 blocks into the discriminator. Finally, we trained our CA-GAN for 100 epochs with a batch size of 16.

4.1. Performance of our final Channel Attention GAN

Comparisons with other methods. We compared the performance of bicubic interpolation, SRGAN, Self Attention GAN (SAGAN), and our Channel Attention GAN (CA-GAN). The quantitative results are shown in Table 2, and all models are trained with the same setting mentioned before. We can see our CAGAN outperforms bicubic, SRGAN, and SAGAN on three benchmark datasets. PSNR score of our model is improved by 3.622 compared to SAGAN and 4.453 compared to SRGAN. MOS score also outperforms all models, which shows our model has better perceptual visual results. To better illustrate our model's performance, we visualize images generated by these models in Figure 4. We can see the CAGAN has the best visual performance, which has more texture details. Our model outperforms SAGAN because the self-attention model only captures spatial relationships, which is not as useful as channel relationships on SR tasks. Intuitive speaking, high-frequent channel information should be extremely useful in reconstructing high-frequency image details. Thus, our channel attention model performs much better than the self-attention model as our model can capture both spatial information and high-frequency features.

Visualization of Channel Attention feature maps. Our Channel Attention GAN model shows better performance than SRGAN because channel attention can capture the interdependencies among channels. To better illustrate the effectiveness of our Channel Attention block, we plot the 64 Channel Attention feature maps for a particular Channel Attention block in our network. As we can see in Figure 5, on the top of each feature map is the learned channel attention weights. Feature maps with high quality often get large weights. Because Channel Attention Blocks selectively emphasize informative channels and suppress less useful channels, which is the superiority of our Channel Attention.

4.2. Ablation study

Different numbers of Channel Attention. We investigated the influence of different numbers of channel attention. Besides, the position of the channel attention block is also very important. As mentioned in section 3.2, in the first few layers, the receptive field is very small, failing to capture information outside its receptive field. We added channel attention blocks in both generator and discriminator. For the model with 4 channel attention blocks, we added two blocks on the first residual block layer and the last residual block layer. We also added two blocks on the first convolution block and the last convolution block on the discriminator. As for the model with eight blocks, we follow the same architecture. And the model with 16 channel attention blocks means we add the block on all 16 residual blocks in the generator and all 8 convolution blocks in the discriminator. By seeing Table 3, we found 16 channel attention model performs the best among the four models. And we

Content Loss	Set5			Set14			BSD100		
	PSNR	SSIM	MOS	PSNR	SSIM	MOS	PSNR	SSIM	MOS
MSE Loss	**33.335**	**0.923**	3.33	**28.588**	**0.800**	3.33	**27.187**	**0.722**	3.37
VGG Loss	30.942	0.854	**3.91**	26.384	0.726	**3.88**	25.441	0.670	**3.79**

Table 4. Comparisons for MSE Loss and VGG Loss

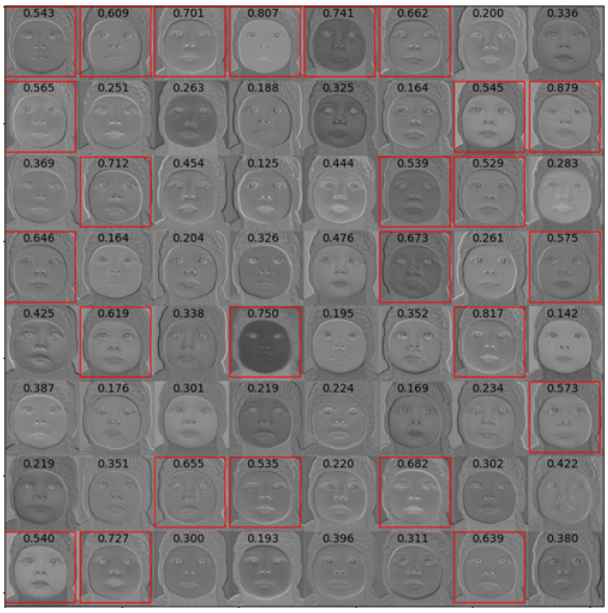

Figure 5. Visualizing Channel Attention: This image contains 64 subimages with each one as the feature map of one particular channel. The attention map weight of each channel is labeled on the top of each image. Images with weights larger than 0.5 are red-boxed.

conclude that more channel attention blocks lead to better performance on our channel attention GAN.

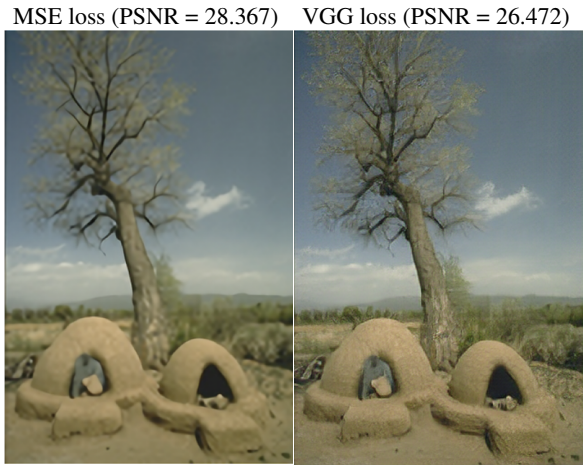

Figure 6. Visualization Comparisons for the MSE and VGG loss

VGG loss *vs.* MSE loss. We also investigated the impact of different content loss functions. This experiment used the same setting for fair comparison and did experiments on VGG loss and MSE loss. The quantitative results are in Table 4. This table shows that the model optimized by MSE loss has a higher PSNR and SSIM scores. However, the model with VGG loss has a much higher MOS score than the model with MSE loss, which means participants think the model with VGG loss has much better visual results. To prove it, we also show the results of the two models in Figure 6. We can see the image generated by MSE loss has a higher PSNR score, however worse visual quality, the image is overly smooth. This illustrates that larger PSNR does not necessarily represent the perceptually better result. In conclusion, MSE loss is optimized to achieve a higher PSNR score; however, VGG loss is optimized to produce better visual effects by computing pixel-wise differences between feature maps.

Stabilization strategies. In this experiment, we investigated the influence of the imbalanced learning rate and spectral normalization. We test three models: (i) the base model. (ii) the model with a fixed learning rate, (iii) the model with an imbalanced learning rate. These models are trained for 80 epochs on a small subset of the COCO dataset with 6k images to reduce computational costs. For the fixed learning rate, we use a 1e-4 learning rate for both the generator and the discriminator. For the imbalanced learning rate, we use a 4e-4 learning rate for the discriminator and 1e-4 learning rate for the generator. We can see from Table 5, the model with an imbalanced learning rate gets better performance than the model with a fixed learning rate, which proves the advantages of an imbalanced learning rate. The third experiment setting performs better than the second one, which proves the effectiveness of spectral normalization.
Also, from Figure 7, we can see the spectral normalization and imbalanced learning rate both stabilize the training process and get a high PSNR score in early epochs.
Thus, both quantitative results and qualitative results reflect the effectiveness of our training stabilization strategies. The two strategies can speed up and stabilize network training.

5. Discussion and future

For the SRGAN, we found that it does not exploit the high-frequency channel features well due to its shallow layers contain mostly useless low-frequency features. We use the channel attention mechanism to tackle it as it can selec-

	Set5			Set14			BSD100		
	PSNR	SSIM	MOS	PSNR	SSIM	MOS	PSNR	SSIM	MOS
basic model	26.039	0.808	3.33	24.848	0.721	3.41	24.093	0.670	3.54
w. imbalanced learning rate	26.146	0.812	**3.60**	25.077	0.727	**3.54**	24.517	0.667	3.66
w. both	**27.960**	**0.831**	**3.60**	**25.496**	**0.729**	3.51	**25.536**	**0.729**	**3.69**

Table 5. Comparisons for imbalanced learning rate and spectral normalization: The first row is the experimental result without an imbalanced learning rate and spectral norm. The second row adds an imbalanced learning rate. The third row adds both an imbalanced learning rate and spectral normalization.

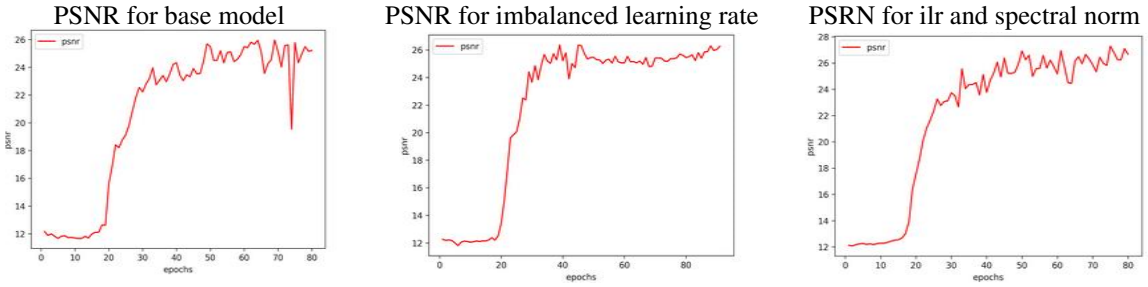

Figure 7. Training plots for imbalanced learning rate and spectral normalization.

tively capture the high-frequency information of channels. Also, shallow layers of SRGAN cannot get long-range pixel information due to the limited receptive field size. We use a global pooling operation to aggregate global information and to capture the spatial information well. As for the Self Attention GAN, it can only capture the spatial information while our model also can capture useful channel information, which performs much better visual results.

We proved our VGG loss could produce more texture details. The reason is that it is optimized based on feature representation rather than pixel-wise between the image in MSE loss. We also proved our adversarial training strategy's effectiveness, which can push the generator to produce more realistic images.

We explored the training stabilization strategies for GAN and found traditional GAN suffers from mode collapse and are hard to train. We use a large learning rate for the discriminator to reduce the discriminator update steps per generator update steps. We empirically proved that it could improve the performance and also can stabilize the training process.

Our model also has some problems and can further improve. In contrast to Zhang *et al*. [24], we found the depth of the model is a crucial factor in SR tasks. In our model, we use 16 residual blocks to build our deep generator network and achieve good performance, but we think a deeper network can further increase our CAGAN performance. For a very deep network, it is hard to train and easily suffer from gradient vanishing and exploding. Zhang *et al*. [24] shows general skip-connection cannot tackle the problem very well when the network is very deep. Instead, they proposed a novel residual structure called Residual in Residual (RIR) structure, which can learn a very deep network. We believe we can improve our model's performance by increasing the number of layers of the network and adapting the RIR structure into our network rather than simply skip-connection. Due to computation resources and time limits, we have not tried the idea. In the future, if we have enough computation resources, we will try it.

6. Conclusion

In this report, we proposed a Channel Attention GAN, which incorporates a channel attention mechanism into the GAN model. We qualitatively proved our channel attention block's effectiveness, which can produce more textures and high-frequency details by capturing both spatial and channel information. Besides, we showed by delicately designed experiments that the imbalanced learning rate and spectral normalization could improve results and stabilize the training of GANs. Our Channel Attention GAN experimentally outperforms both the state-of-the-art baseline model SRGAN and Self Attention GAN using PSNR, SSIM, and MOS testing on three benchmark datasets.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[3] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[10] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. corr abs/1412.6980, 2014.

[14] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[17] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks, 2018.

[18] Kamal Nasrollahi and Thomas B Moeslund. Super-resolution: a comprehensive survey. *Machine vision and applications*, 25(6):1423–1468, 2014.

[19] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[21] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[22] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[23] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[24] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.