Pushbroom satellite image super-resolution using generative adversarial networks

Chao Xu
The Chinese University of Hong Kong
Shatin, New Territories, Hong Kong SAR
1155160618@lin.cuhk.edu.hk

Abstract

In order to improve the resolution or visual effect of imageries from pushbroom sensors, this report proposes to use super-resolution generative adversarial networks (SRGAN) [5]. Some improvements are achieved: firstly, the VGG-19 [8] network is modified to extract the features of input images, and thus reduce the training time; secondly, the loss function is updated as new evaluation metric for the generated images; finally, three models with different hyperparameters are trained and compared to find the optimal results. In this report, the pushbroom image super-resolution is achieved with a scale factor of 4, and the result shows a better visual perception compared with that recovered by bicubic interpolation. We call the network model in this report as pbiSRGAN, which is Push-Broom Images SRGAN.

1. Introduction

Pushbroom imaging has been one of the most common satellite scanning methods over the past decades, which features in low manufacturing cost and relatively high maneuvering [10]. However, its spatial resolution is usually degraded due to strong atmospheric turbulences, fierce temperature fluctuation or the Earth rotation, et al. As the development of CMOS and CCD technology, the size of single detector pixel is getting smaller and the density of detector array is increasing, which help improve the satellite image resolution but also greatly level up the hardware manufacturing cost. Therefore, software methods such as imaging super-resolution algorithms have been proposed recently.

Conventional super-resolution methods include nonuniform interpolation, maximum a posteriori, projection on convex sets, and iterative back projection [6]. These methods have their own constraints, such as multiple solutions, instability or image blurring. Since the beginning of this decade, deep learning has been keeping advancing itself into various areas, including object detection, image classification, speech recognition and so on. Imaging super-resolution has been a hot topic in the deep learning community since it is first introduced in 2015 as SRCNN [1]. In 2017, a perceptual loss is additional proposed in SRGAN [5] as a complement to the conventional pixel-based loss metrics, and it is another landmark in the area of deep learning image super-resolution. Essentially, the SRGAN method is also a variant of GAN proposed in 2014 [2].

In this report, we will first talk about the basic principle of SRGAN, which is different from those methods generating images pixelwisely, but creates more visually perceptual images with high resolution. Secondly, a neural network architecture for pushbroom images is described, including how it is fit for gray-scale remote sensing imageries, how to reduce training time and improve training efficiency. Finally, three experimental groups will be set up with different hyper-parameters to find the optimal results for the pbiSRGAN.

2. Key concept of SRGAN

As for SRGAN, the low-resolution images are fed into the generation network and "high-resolution images" are generated as the input of the discriminant network. The discriminator needs to determine whether the input images are generated "high-resolution image" or the original ground truth value. The super resolution process is completed when the equal probabilities are obtained of "high-resolution images" being judged as true or false. At this time, the generative network can be used to generate the "high-resolution images" closest to the ground truth value. Here, it is also necessary to mention the way of "summarizing experience" for both networks, namely, loss function. The loss function used in SRGAN is crit-

ical to the final network, which is different from the traditional per-pixelwise MSE loss. It evaluates the generated image from the perspective of human visual perception [5]. The advantage of pixelwise MSE loss is that it achieves a very high PSNR, but the reconstructed images are lack of high-frequency information, resulting in over-smooth image texture and poor visual effect. Specifically, the perceptual loss function L^{SR} is composed of content loss L^{SR}_{Con} and weighted generative loss L^{SR}_{Gen}:

$$L^{SR} = \underbrace{\underbrace{L^{SR}_{\text{Con}}}_{\text{content loss}} + \underbrace{10^{-3} L^{SR}_{\text{Gen}}}_{\text{generative loss}}}_{\text{perceptual loss}} \tag{1}$$

Content loss uses VGG-19 [8] as the feature extractor of the input image. The pre-trained network has strong feature extraction capability. Content loss is defined as the Euclidean distance between the reconstructed image $G\big(I^{LR}\big)$ and the reference image I^{HR} feature map:

$$L^{SR}_{Con} = \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left(\phi_{i,j}\big(I^{HR}\big)_{x,y}, \phi_{i,j}\big(G\left(I^{LR}\right)\big)_{x,y} \right)^2 \tag{2}$$

where $W_{I,j}$ and $H_{I,j}$ respectively represent the dimensions of each feature map in VGG-19 network; $\phi_{i,j}$ refers to the feature map [5] obtained by the convolution layer j in front of the max pooling layer i in VGG-19 network.

The other part of the perceptual loss is generative loss, which makes the generated results more consistent with the real images of human visual perception.

$$L^{SR}_{Gen} = \sum_{n=1}^{N} -\log D\left(G\left(I^{LR}\right)\right) \tag{3}$$

where $D\big(G\left(I^{LR}\right)\big)$ is defined as the probability that the discriminator D determines the reconstructed images as real high resolution images, and N is the number of generated samples.

3. From SRGAN to pbiSRGAN

Compared with natural images, there is a serious problem applying SRGAN to remote sensing pushbroom images, that is, the lack of real training sets containing high-low resolution image pairs. Specifically, if the real image obtained by the satellite is regarded as a low-resolution image, and then it is impossible to obtain the corresponding ground truth as a high-resolution image. It is obvious that the satellite cannot avoid atmospheric turbulence, temperature fluctuation and other factors in the real-scene shooting, therefore, it is impossible to obtain a higher resolution image. The only way to get high-low resolution image pair is to take the real satellite image as a high-resolution image (i.e., ground truth) and then simulate a low-resolution version of it. In addition, SRGAN is originally proposed to solve the super-resolution problem of naturally colorful images, and can not be directly used for remote sensing data. Remote sensing images can be divided into panchromatic, multispectral, hyperspectral and even hyperspectral data, whose image channels range from one to hundreds. The remote sensing imageries own the characteristics of large scanning scale and great data volume, and the coverage of a single shot can reach dozens or even hundreds of square kilometers. Therefore, we proposed pbiSRGAN to achieve the purpose of super-resolution for pushbroom scanning remote sensing images.

4. Training and reconstructing of pbiSRGAN

The section aim to training and reconstructing process of remote sensing image super-resolution.

4.1. pbiSRGAN training flow

The training flowchart is shown in Fig. 1 for model of pbiSRGAN.

In the first step, a panchromatic remote sensing image used and preprocessed to make it as the initial ground truth of network input. The steps of preprocessing include histogram equalization and random cutting. The former aims to improve the contrast of the image and is beneficial to the extraction of the feature map. The latter is to generate sufficient data sets, including training sets, validation sets, and testing sets.

Second, after the training set enters the network, bicubic interpolation with a certain scale factor is conducted to generate the corresponding low-resolution images (namely, the high-resolution images are downsampled), and thus obtaining the HR and LR image pairs in the training process.

The third step is to train the generative network G. The low-resolution image LR is passed through G to get "generated high-resolution image" $G\big(I^{LR}\big)$. Since the network G is weak at first, the quality of $G\big(I^{LR}\big)$ may not be as good as LR. Here, we need to evaluate the loss of the generated image. The loss of the network G L^{SR}_{G} is mainly divided into two parts, namely the content loss L^{SR}_{Con} and generative loss L^{SR}_{Gen}. L^{SR}_{Con} is

evaluated by Eq.(4):

$$L_{Con}^{SR} = \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left\| \phi_{i,j}\left(I^{HR}\right)_{x,y}, \phi_{i,j}\left(G\left(I^{LR}\right)\right)_{x,y} \right\|_1 \tag{4}$$

That is, the feature maps of $G\left(I^{LR}\right)$ and HR are extracted through the modified VGG-19 network, and the L_1 distance between them is calculated. The L_2 norm is mainly related with the Gaussian distribution error, while the L_1 norm is related to the Laplace error. When the image contains non-Gaussian errors, the confidence of L_1 norm is higher than that of L_2 norm [9]. Only when the error of the model is Gaussian white noise distribution, the solution of L_2 model is optimal, which is difficult to achieve in real world.

L_{Gen}^{SR} is also redesigned as shown in Eq.(5):

$$L_{Gen}^{SR} = \sum_{n=1}^{N} 1 - \log D\left(G\left(I^{LR}\right)\right) \tag{5}$$

Finally, we need to train the discriminator D. The power of network D can be strengthened by iteration of the loss function L_D^{SR}, which consists of two parts, L_{real}^{SR} to discriminate I^{HR} as true and L_{fake}^{SR} to discriminate $G\left(I^{LR}\right)$ as false.

$$L_{real}^{SR} = \sum_{n=1}^{N} 1 - \log D\left(I^{HR}\right) \tag{6}$$

$$L_{fake}^{SR} = \sum_{n=1}^{N} - \log D\left(G\left(I^{LR}\right)\right) \tag{7}$$

$$L_D^{SR} = \frac{1}{2}\left(L_{real}^{SR} + L_{fake}^{SR}\right) \tag{8}$$

4.2. pbiSRGAN reconstructing flow

While validating the generative networks using validation set, super-resolution reconstruction can also be accomplished, as shown in Fig. 2.

5. Network architecture of pbiSRGAN

The network is mainly divided into three parts: feature extraction network F, generative network G and discriminant network D.

5.1. Feature extraction network

The basic prototype of the feature extraction network comes from the VGG-19 network, which is a 19-layer pre-training network with very strong feature extraction ability, but it cannot be directly applied to the network model in this project. VGG-19 is used for feature extraction of colorful images. In this section, the previous 18 layers of VGG-19 are extracted, and the first layer of VGG-19 is modified into a structure suitable for single-channel gray remote sensing images. In this way, the feature extraction network F can obtain the corresponding features from the ground truth value and generated images, and then carry out loss calculation. The structure of network F is shown in Fig. 3.

5.2. Generative network

Generative network G is the core of the whole pbiSRGAN network, which will generate high-resolution images. The network structure we designed is shown in the Fig. 4, the core of which is the residual block. The role of residual blocks will be discussed later.

Specifically, a convolution layer with kernel size of 9 $times$9 is used first, and the input is a single-channel low-resolution image and the output is a 64-channel feature map. A Parametric Rectified Linear Unit (PRELU) is then used as the activation function, with the parameter of . As shown in Fig. 5, PRELU is actually a variant of Leaky Rectified Linear Unit (Lrelu). For LReLU, if $x < 0, y = 0.01x$;For PReLU, if $x < 0$, then $y = ax$. Compared with ReLU, the advantages of using PReLU are to prevent zero activation of the network and speed up the training process.

Next, we use 16 residuals blocks, all of which have the same structure. Using residuals blocks helps solve the optimization problem when the forward network is too deep [3]. The structure of each residual block is shown in Fig. 6. Its first layer is a convolution layer, with input and output channels of 64, and the size of the convolution kernel is 3 $times$3. What follows up is a batch normalization (BN) layer that speeds up training by reducing covariance shifts within the dataset, allowing us to use larger learning rates during training without decreasing output quality[4]. After activation by PRELU, there is another pair of convolution layer and batch normalization layer with the same parameters as above, and the final output is 64-channel feature map.

As shown in the Fig. 4, after passing the output of the PReLU1 layer and the 16 residual blocks to the BN33 layer, we used two trained sub-pixel convolution layers to increase the resolution of the input image [7]. The core of the sub-pixel convolution layer is the PixelShuffle layer, which can convert the 4-dimensional data in the training process from $(*, C \times r \times r, H, W)$ to $(*, C, H \times r, W \times r)$, where $*$ represents the batch size of the data set. C, H and W are the dimension of the images, and r is the scale factor. Finally, the generated high-resolution image is obtained through the

Conv37 layer.

5.3. Discriminant network

The main function of discriminant network D is to battle with generative network G, so as to improve the power of G. The discriminant network designed in this project is shown in the Fig. 7, with a total of 24 layers.The initial input channel and the final output channel are both designed as 1. In addition, the convolution layer of 64, 128, 256 and 512 channels is designed to gradually improve the feature extraction ability. The detailed parameters can be seen in the table 1.

Please direct any questions to the production editor in charge of these proceedings at the IEEE Computer Society Press: https://www.computer.org/about/contact.

No.	Layer	Input channel	Output channe
1	Conv2d	1	64
2	LeakyReLU	-	-
3	Conv2d	64	64
4	BatchNorm2d	64	64
5	LeakyReLU	-	-
6	Conv2d	64	128
7	BatchNorm2d	128	128
8	LeakyReLU	-	-
9	Conv2d	128	128
10	BatchNorm2d	128	128
11	LeakyReLU	-	-
12	Conv2d	128	256
13	BatchNorm2d	256	256
14	LeakyReLU	-	-
15	Conv2d	256	256
16	BatchNorm2d	256	256
17	LeakyReLU	-	-
18	Conv2d	256	512
19	BatchNorm2d	512	512
20	LeakyReLU	-	-
21	Conv2d	512	512
22	BatchNorm2d	512	512
23	LeakyReLU	-	-
24	Conv2d	512	1

Table 1. Parameters of discriminant network.

References

[1] 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.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[3] Sam Gross and Michael Wilber. Training and investigating residual nets. Facebook AI Research, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[5] 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.

[6] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. IEEE signal processing magazine, 20(3):21–36, 2003.

[7] 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. In IEEE conference on computer vision and pattern recognition, pages 1874–1883, 2016.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[9] Huihui Song, Lei Zhang, Peikang Wang, Kaihua Zhang, and Xin Li. An adaptive l 1–l 2 hybrid error model to super-resolution. In 2010 IEEE International Conference on Image Processing, pages 2821–2824. IEEE, 2010.

[10] Chao Xu, Xiubin Yang, Tingting Xu, Lin Zhu, Lin Chang, Guang Jin, and Xiangdong Qi. Study of space optical dynamic push-broom imaging along the trace of targets. Optik, 202:163640, 2020.

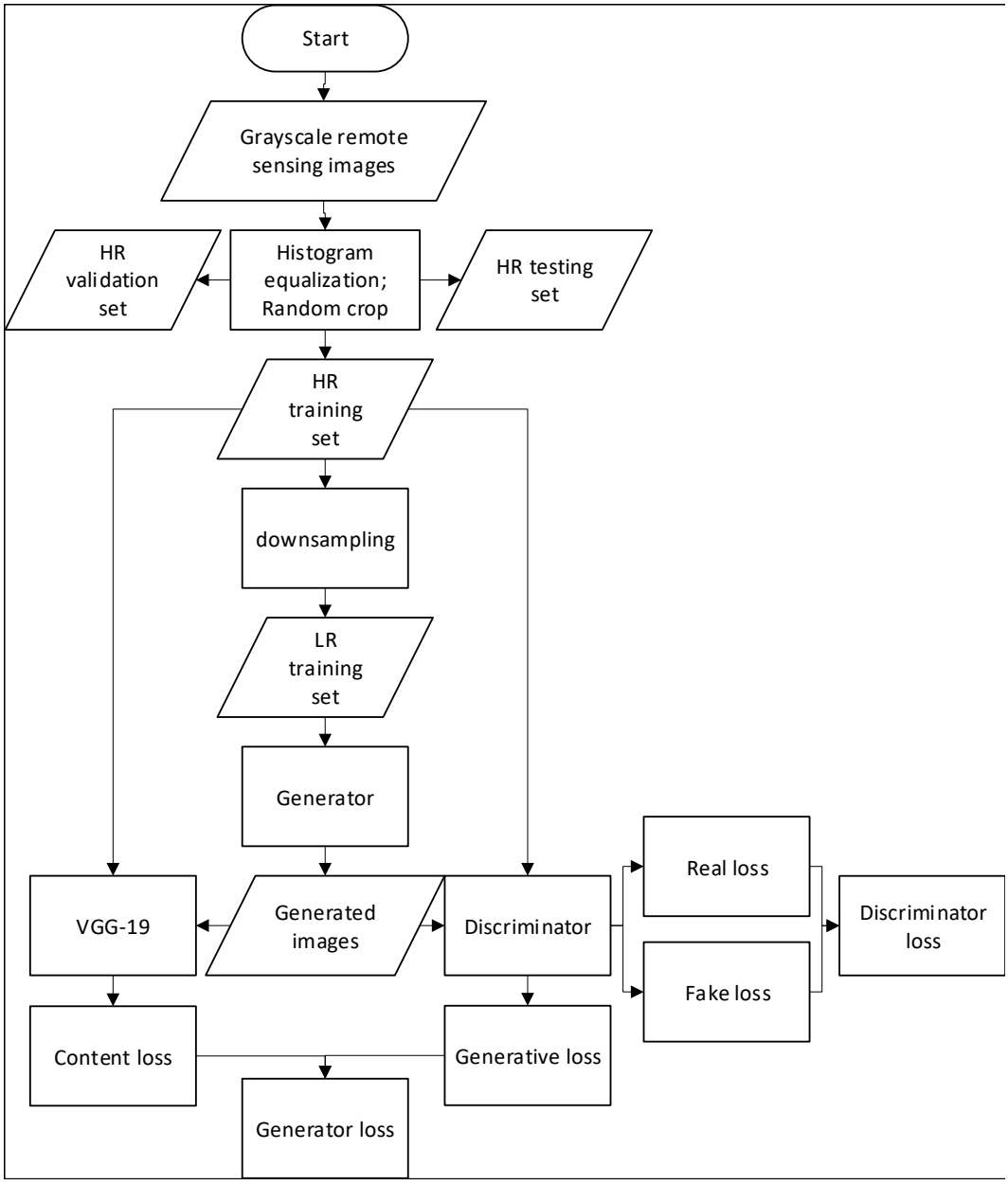

Figure 1. Training flow chart of pbiSRGAN.

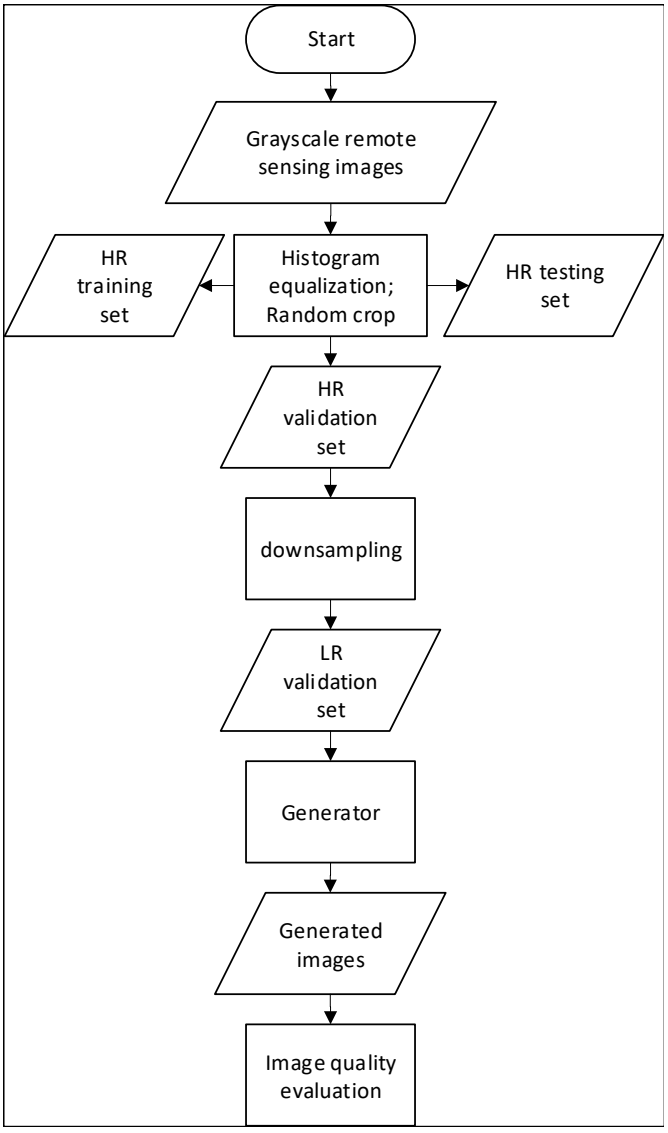

Figure 2. Reconstructing flow chart of pbiSRGAN.

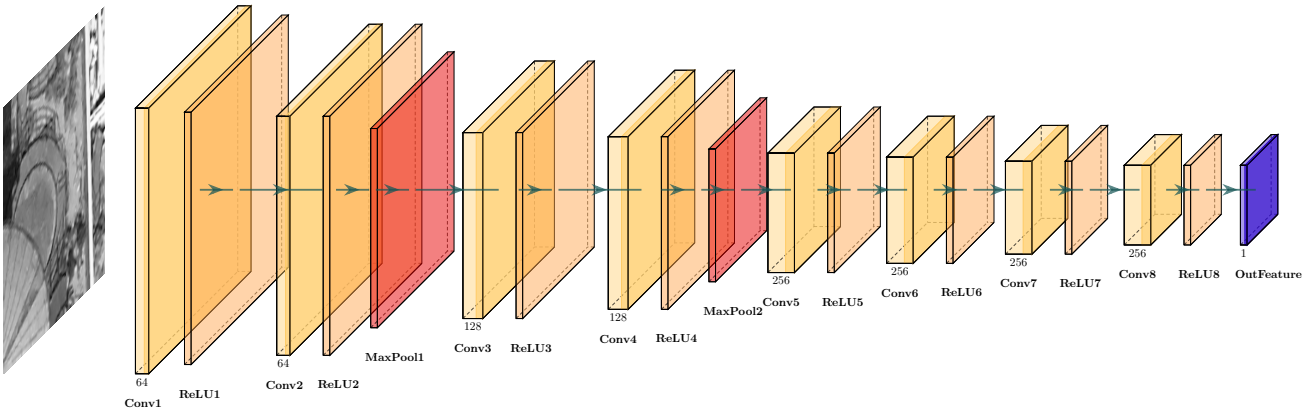

Figure 3. Architecture of feature extractor network.

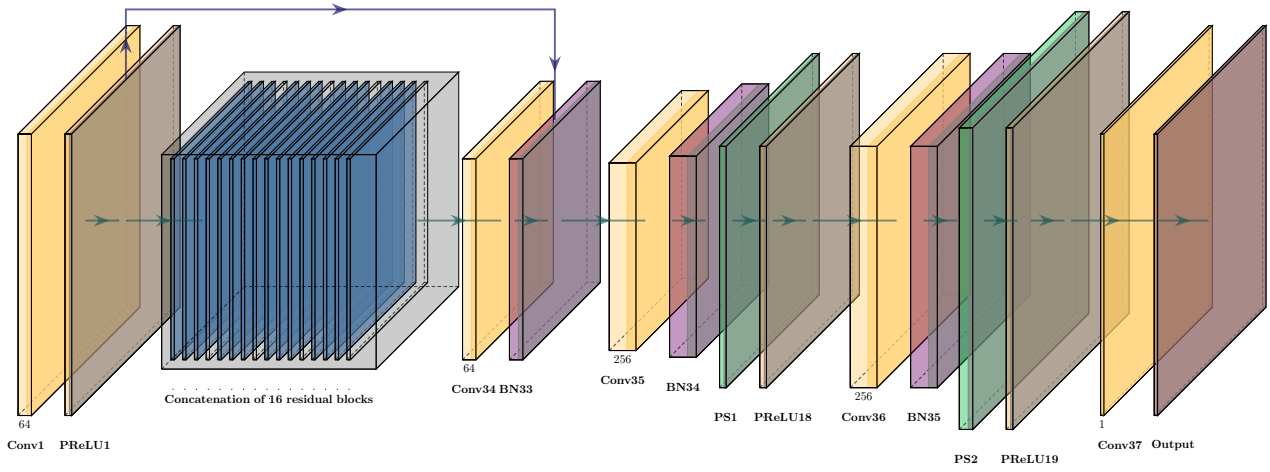

Figure 4. Architecture of generative network.

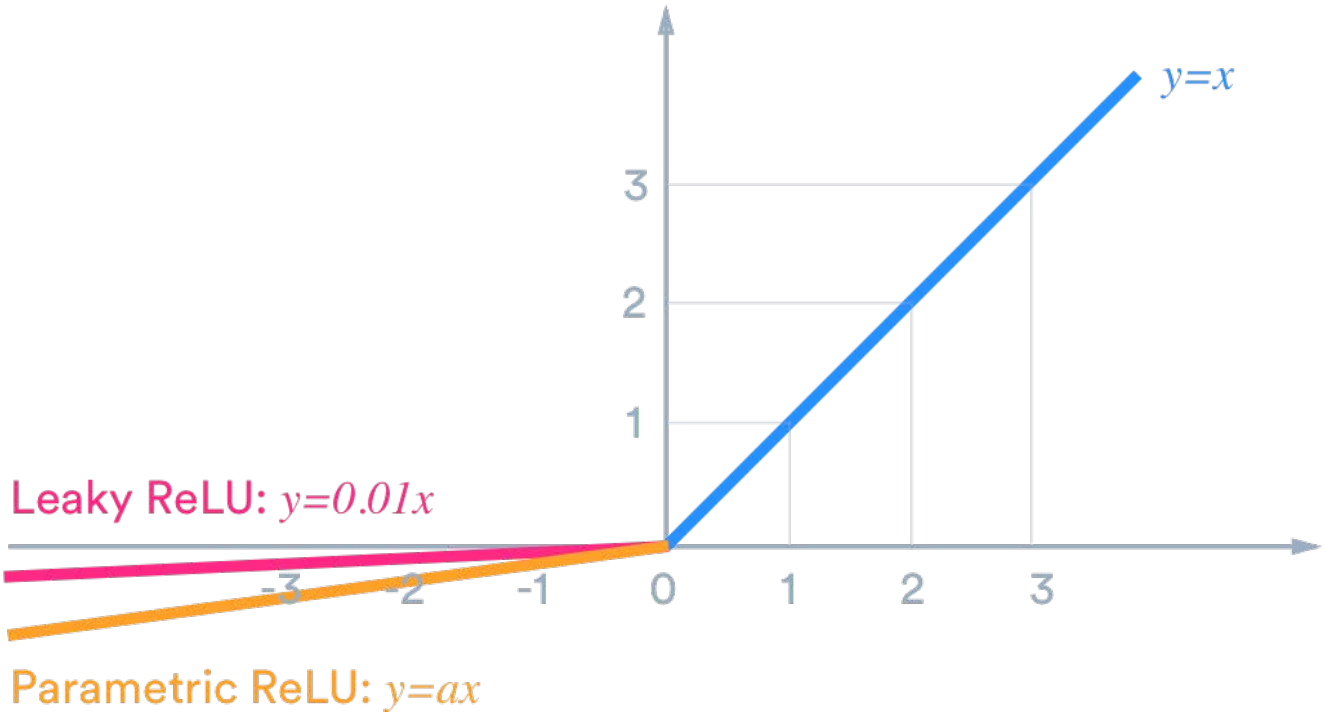

Figure 5. Representation of LReLU and PReLU.

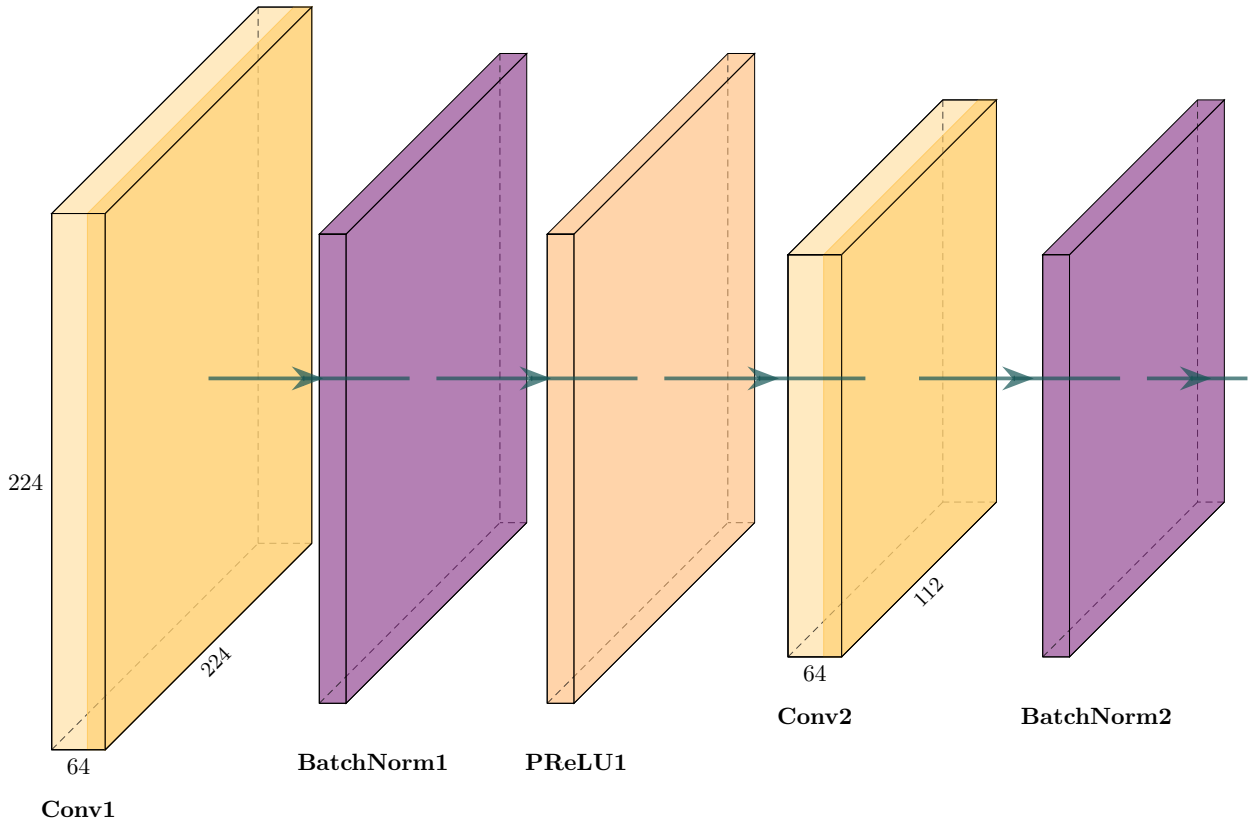

Figure 6. Architecture of residual block.

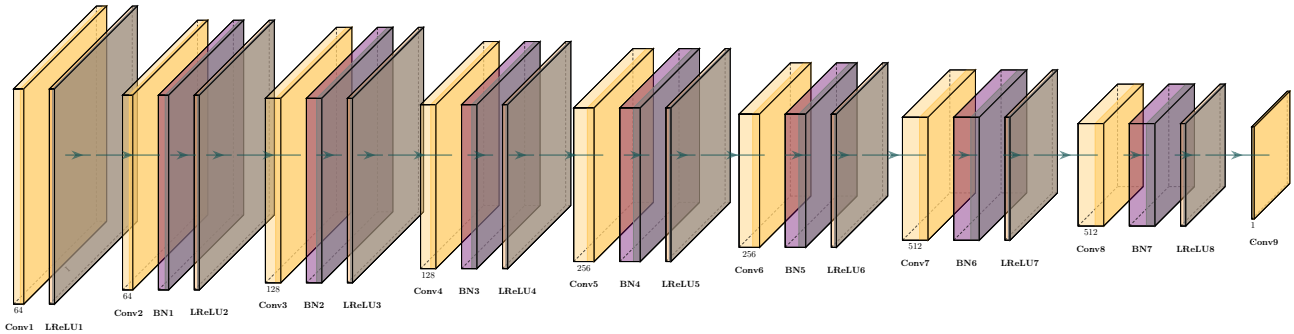

Figure 7. Architecture of discriminator network.