Course Name- Project Lab (CSN-504)

**REPORT ON**

SINGLE IMAGE SUPER RESOLUTION USING GENERATIVE ADVERSARIAL NETWORK

Under the supervision of

**Prof. Sandeep Kumar Garg &**

**Prof. Pradumn K Pandey**

**Group members-**

Bhanu 19535008

Mini Agarwal 19535019

Venkateshwara Packya Janeefa S 19535030

# 

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE**

**ROORKEE- 247667 (INDIA)**

**AUTUMN, 2019**

**Undertaking**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Bhanu 19535008

Mini Agarwal 19535019

Venkateshwara Packya Janeefa S 19535030

**Signature:**

**Date:**

SINGLE IMAGE SUPER RESOLUTION USING GENERATIVE ADVERSARIAL NETWORK

ABSTRACT………………………………………………………………………. 3

1. INTRODUCTION……………………………………………………………… 4

1.1 Super resolution……………………………………………………….... 4

1.2 Convolutional neural network………………………………………...... 4

1.3 Super resolution generative adversarial network………………………. 5

1.4 Perceptual loss…………………………………………………………. 7

2.LITERATURE SURVEY……………………………………………………….. 8

3.WORK DONE..………………………………………………………………… 9

3.1 Approach used….……………………………………………………….9

3.2 Implementation details…………………………………………………12

3.2.1 Data set processing……………………………………………12

3.2.2 Training details………………………………………………..14

4. RESULTS AND OBSERVATION……………………………………………..15

4.1 Result…………………………………………………………………..15

4.2 Observation…………………………………………………………… 24

5.CONCLUSION………………………………………………………………... 26

6.FUTURE WORK……………………………………………………………… 27

7.REFERENCES………………………………………………………………… 29

**Abstract**

Lots of work has been done in super resolution domain by using deep neural networks, but the main problem still remains. How to recover the finer textual details of an Image.Recent work has focused on minimizing the pixel wise mean squared error, but the resulting images don’t have high frequency details despite of having higher PSNR values. In this work, we have used the concept of Perceptual loss which is the combination of adversarial and content loss. Adversarial loss helps the discriminator to differentiate between the generated images and real images and content loss is used to measure the perceptual similarity instead of similarity in pixel space between both the images.

**Chapter 1**

**Introduction**

**1.1 Super Resolution**

Super Resolution is the task in which High-Resolution(HR) images are obtained from its Low-Resolution(LR) Counterpart. These images should be rich in textual details.

Super-resolution can be Muti Image super-resolution in which multiple images are used to produce the Super Resolution(SR) image[1]

Or it can be single image super-resolution(SISR) in which only a single LR image is used to produce an SR image.

In this report, we will focus on the second type i.e SISR.

**1.2 Convolutional Neural Networks**

Convolution neural networks are the deep learning networks that take images as input and produces results according to the requirements. For example, for a classification problem, CNN will return a class to which the input image belongs to and for a super-resolution problem it will take an LR image and outputs its corresponding SR image.

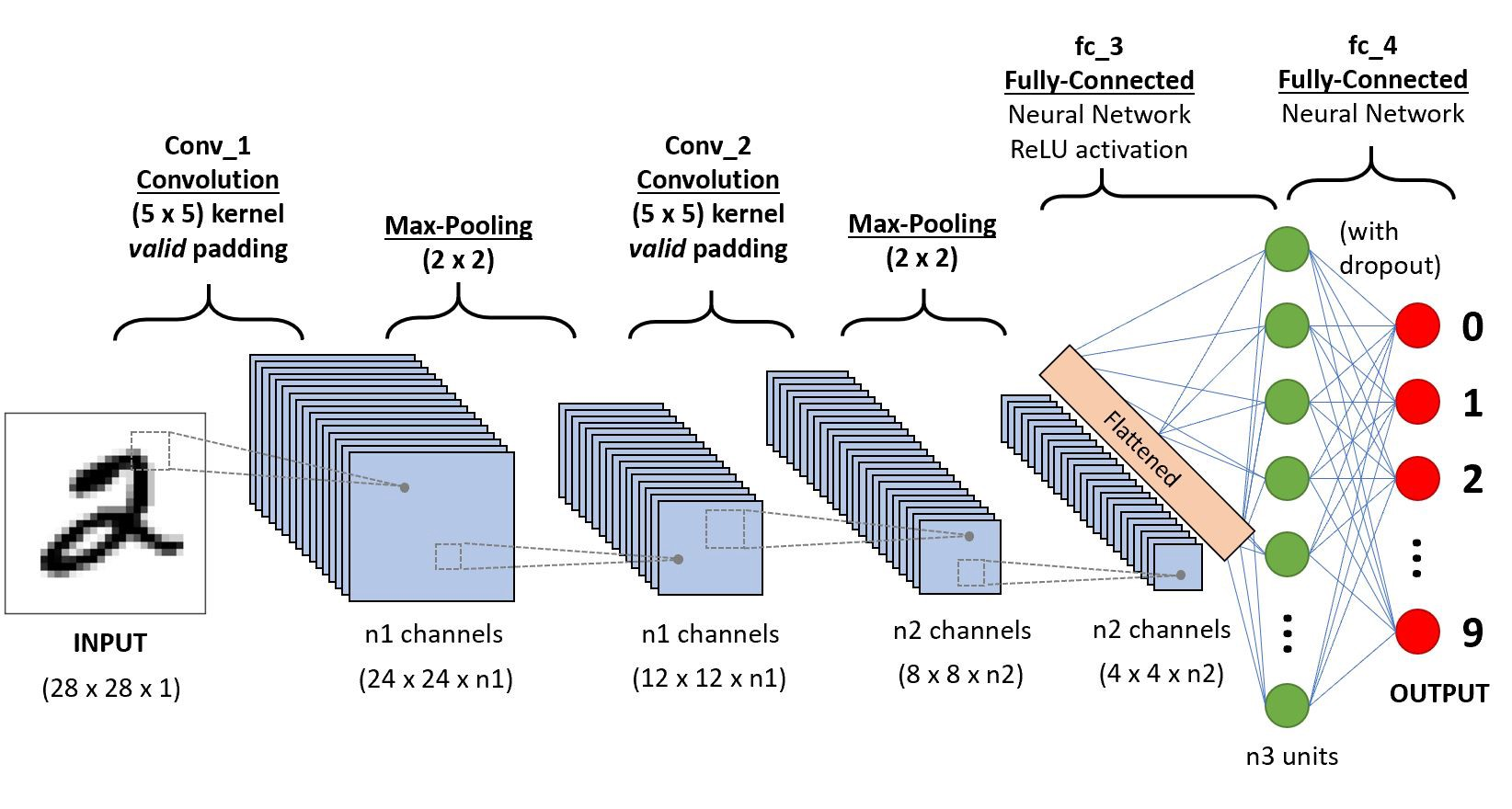


Figure 1: A CNN to classify images of numbers.

CNN has three types of layers:

1. **Convolution layer:**

Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.

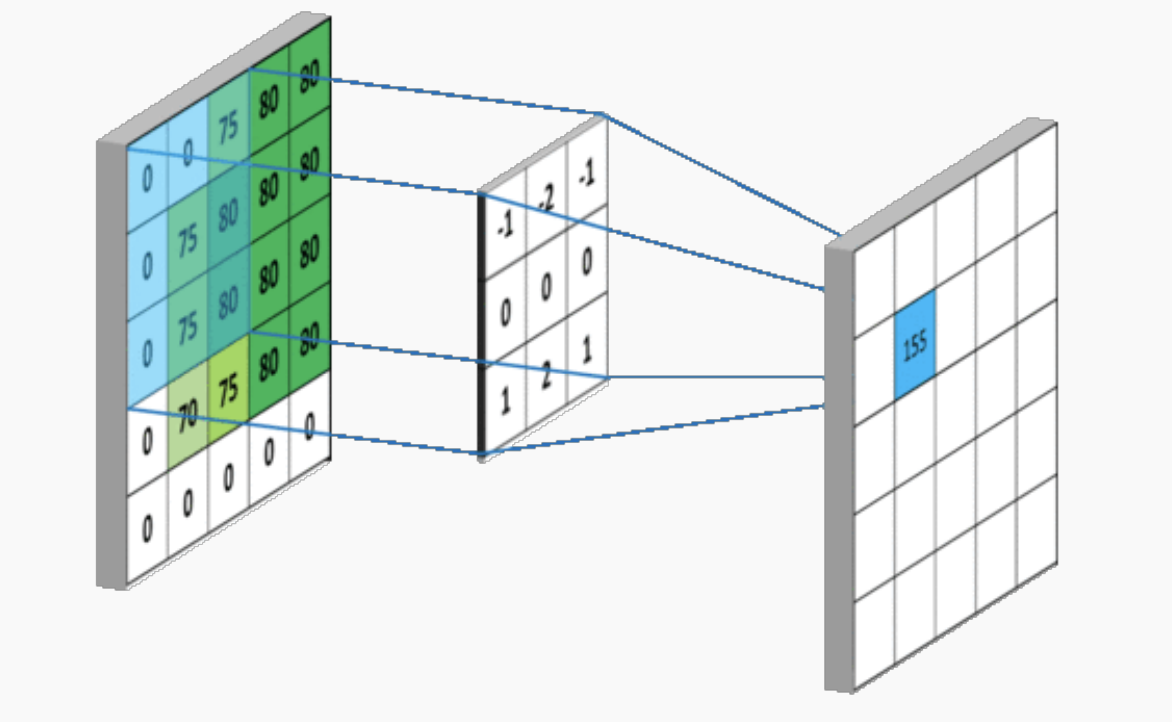


Figure 2: Figure showing how filters are applied to an image

1. **Pooling Layer:**

Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

1. **Fully Connected Layer:**

Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an (Multi-Layer Perceptron) MLP.

**1.3 Super-Resolution Generative Adversarial Network(GAN):**

Generative adversarial networks deep neural networks consisting of two main components a Generator and Discriminator. One is used to improve the performance of another.

While on one hand, given a label, generator tries to generate feature maps which lie in that label from random noise(In context to SR, Generator produces HR images(called as SR images) from LR images), on the other hand, given a feature map discriminator tries to put a label to it(In context to SR, Discriminator predicts whether the SR image generated by generator belongs to HR category or not).

Training steps of GAN:

1. The generator takes in LR Image and produces an SR image.
2. This image is fed to the discriminator along with the original HR images(Ground Truth) and the discriminator calculates the probability by which this SR image belongs to the HR image set.
3. The discriminator’s weights get updated with respect to the error in its prediction so that in future it performs better in detecting the fake data.
4. The generator weighs gets updated with respect to the prediction made by the discriminator so that in future it can produce more accurate results which can’t be differentiated by the discriminator.

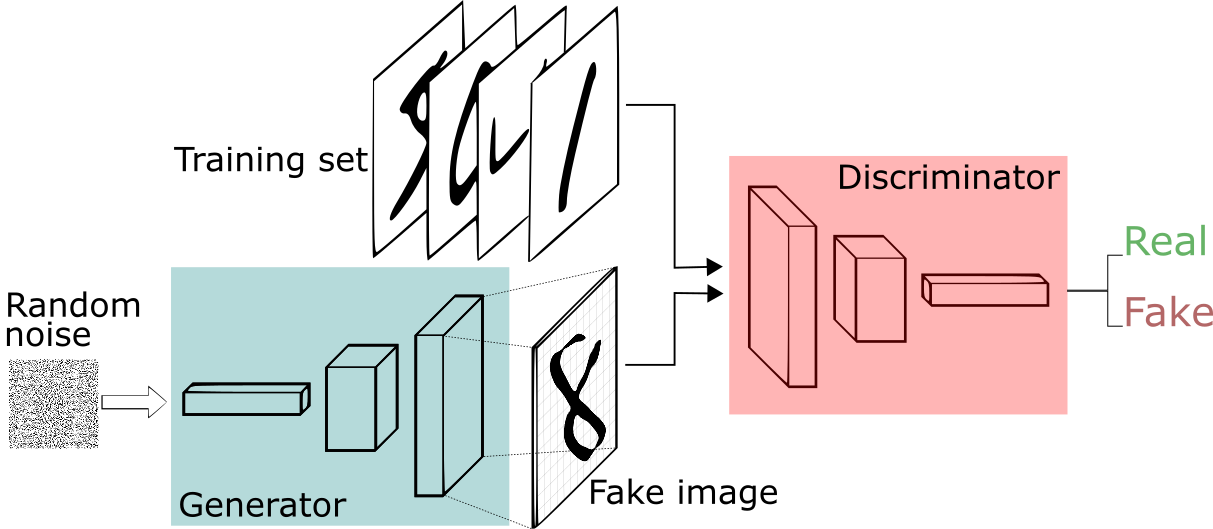


Figure 3: A basic GAN

**1.4 Perceptual Loss**

In general SR models, the loss function is mainly based on MSE but its ability to capture the perceptually relevant differences such as texture is limited as they are based on pixel-wise image difference.

Therefore, a perceptual loss which is a weighted combination of Content Loss and Adversarial Loss is used.

1. **Content Loss**

A VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network. With we indicate the feature map obtained by the jth convolution (after activation) before the ith max-pooling layer within the VGG19 network.

VGG loss is then defined as the euclidian distance between the reference image and the generated image().

1. **Adversarial Loss**

Adversarial loss is defined as the summation of probability that the generated image is a natural image(HR) over all the training examples(1 to N).

**Chapter 2**

**Literature Survey**

Chao Dongproposed a deep convolutional neural network to convert low resolution image to high resolution image.In which,first upscale is done to low resolution image using bicubic and then patch extraction and representation,non-linear mapping and reconstruction are done.All these three together form the convolutional neural network.In patch extraction and representation,the patches from the bicubic image is extracted and converted into a high dimensional vector.In non-linear mapping, the output from the patch extraction and representation is mapped into another high dimensional vector.Reconstruction aggregates the high resolution patches from the previous step into the final high resolution image.

Alahi[4] done experiment on two image transformation tasks, transformation of style and super resolution.This is mainly focused on how to increase the speed using perceptual loss instead of using per-pixel loss. And another reason is to use perceptual loss is that even though per-pixel loss gives the mathematically correct edge by maximizing the PSNR metrics value but in perceptual view it differs a lot from the ground truth image.Perceptual loss is used to train the network with the fixed loss value in the pre-trained network.

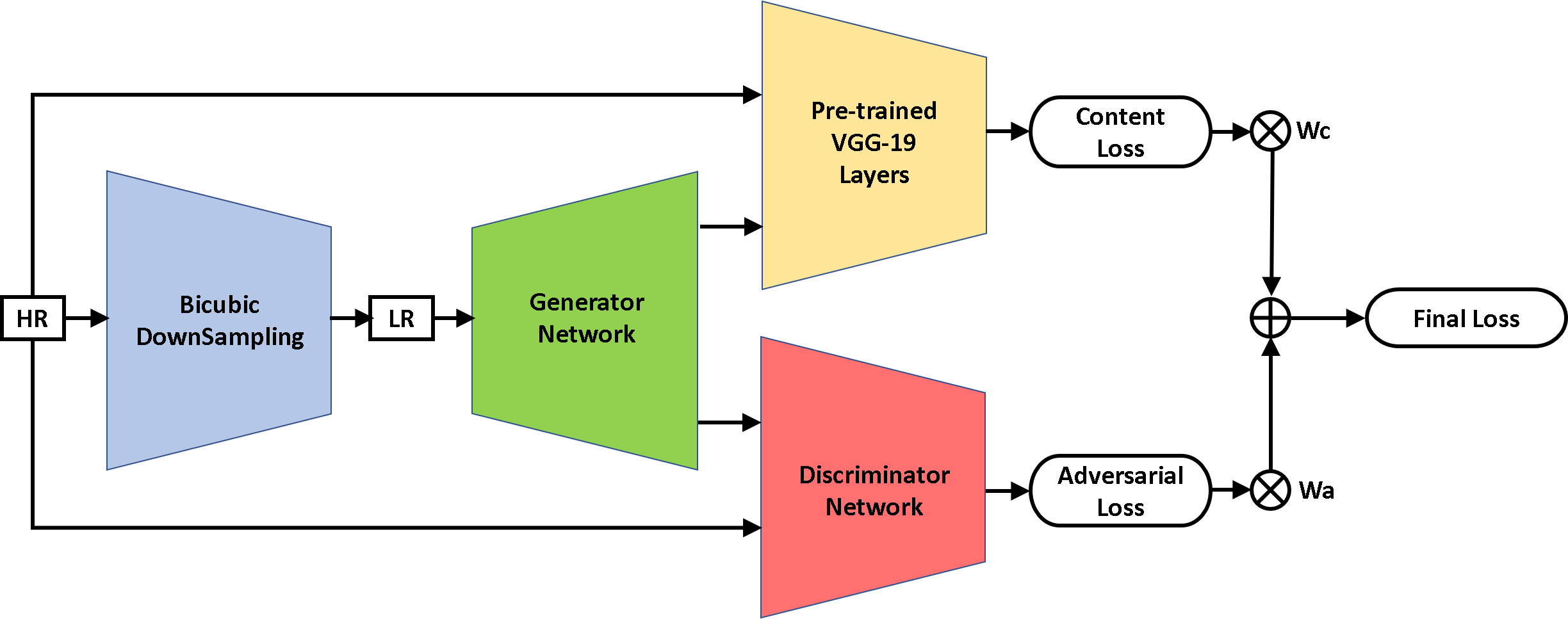
**Chapter 3**

**Work Done**

**3.1 Approach used:**

For the proposed problem, we will implement the paper by Ledig et al.[2] The paper solves the problem of single image super resolution. It focuses on producing a high resolution (HR) image corresponding input low resolution (LR) image such that texture details are not lost. To get satisfactory results, we use SRGAN for generating SR images. Output image is generated using a GAN, Generative Adversarial Network, which is trained to generate data from scratch using unsupervised machine learning. The system has two networks: Generator and Discriminator.

Generator generates new data which fits the generation of new data as output that fits the distribution of the data that is used for training. Discriminator is designed and trained to discriminate among two or more classes, which in this case are the ground truth image and output image which is generated by generator.

Figure 3: Basic Architecture for training of SRGAN.

Training is done in two Phases:

1. Generator Training

* HR images are downsampled to LR by using bicubic Downsampling.
* The generated LR image is passed through the generator which upsamples the image and gives SR image.
* Loss is calculated as Mean Squared Error(MSE) between the SR and HR image which is backpropagated back to train the generator.

1. GAN(Generative Adversarial Network) Training

* In this phase both generator and discriminator are alternatively trained.
* Firstly, the generator’s weights are updated according to the total loss(perceptual loss).
* Secondly, the discriminator’s weights are updated according to the Adversarial loss.
* The above two steps are repeated for the training of GAN.

Generator Architecture:



Figure 4: Generator Network (SRResNet or SRCNN).

Discriminator Architecture:



Figure 5: Discriminator Network

VGG19 Architecture:

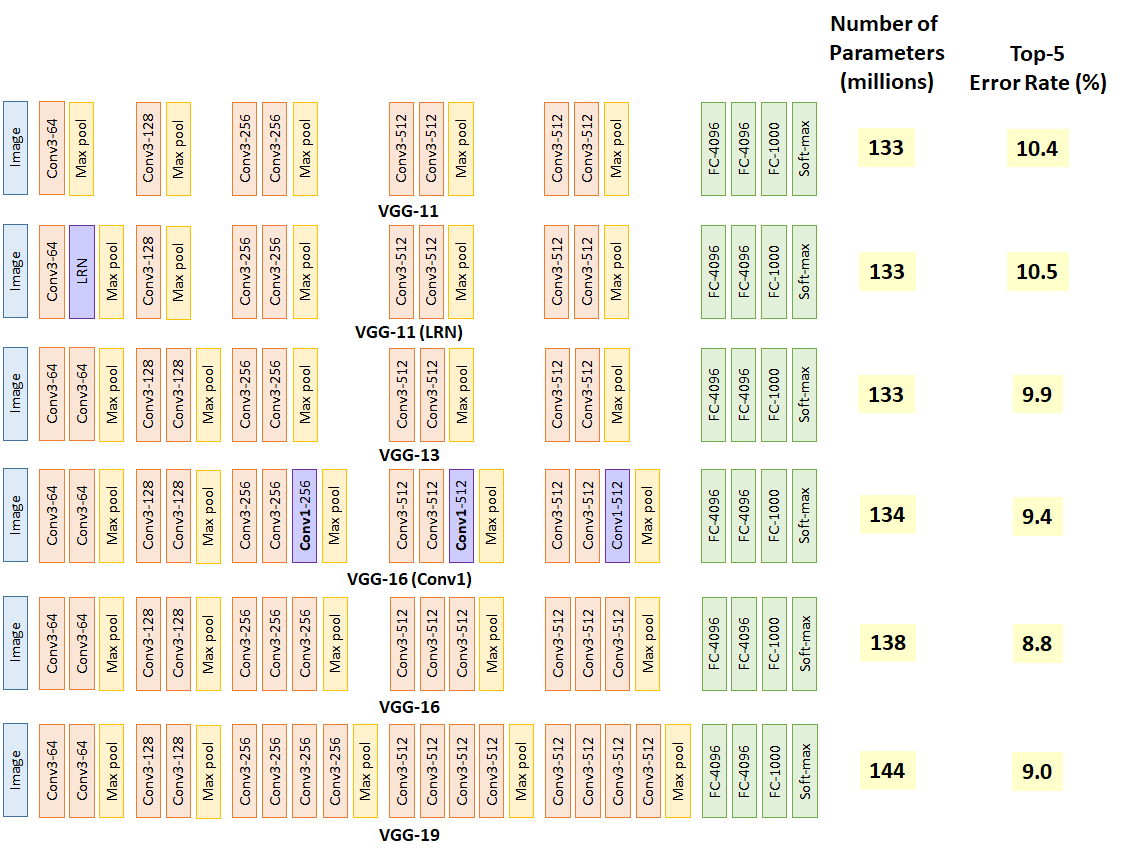


Figure 6: VGG19 Network

In this model, loss has two components: Adversarial loss and Content loss. Adversarial loss is calculated by discriminator by distinguishing between the generated SR images and ground truth HR image. Adversarial loss ensures that the constructed image looks real. Content loss is loss which is more related to perceptual similarity between SR and HR image. Content loss is focused on perceptuality rather than pixel space. Content loss ensures that high level content details of image are preserved.

Content Loss:

Here for calculating the content loss, MSE loss is calculated between the feature maps obtained by the 4th convolution(after activation) before 5th maxpooling layer of the VGG19 Network by inputting the SR and HR images.

Adversarial Loss:

This loss makes sure that the solutions(SR images) reside on the manifold of natural images.

We have to minimize this loss.

Here, probability that the generated image is a natural image(HR).

Final Loss:

In the proposed paper[1], final loss is calculated as :

But here in our work we have also taken some weighted() part of the MSE loss as well.

**3.2 Implementation Details:**

For the implementation part, scaling of x4 is there in between the LR and HR images.

**3.2.1 Dataset Processing:**

**Collected data sets-**

* DIV2K dataset (800 Images) for training.
* Set5, Set14, BSDS100 for testing purposes

**Data reading & LR generation, with support for heterogeneous & multiformat images**

* Since in the above datasets both LR and HR images of required dimensions are not present (i.e differing by a scale of 4) therefore we have only collected HR images and then converted them in LR images using bicubic downsampling.
* And since the HR images can be of different sizes and different formats, therefore, our system should be robust enough to process them.
* During the training phase, we are dividing both LR and HR images into patches of 24\*24 and 96\*96 resolution respectively.
* The LR images are scaled in between the range of 0 to 1 and HR images are scaled in between the range of -1 to 1.
* VGG19 feature maps are also rescaled by a factor of 1/12.75\*\*2 (Different from proposed approach(1/12.75)).

There are two approaches for patching of images:

* The image is first downsampled and then broken down into patches for further processing.
* The HR image is first broken down into patches of small size and then is downsampled for further processing.



Figure : LR Patches(24\*24)



Figure : HR Patches(96\*96)

**3.2.2 Training Details:**

We have trained our network on Google Collaboratory which provides a single 12GB NVIDIA Tesla K80 GPU using 800 images(after preprocessing) obtained from DIV2K dataset.

Adam optimiser is used with . SRCNN network is trained with learning rate and iterations. Updations made to the generator and discriminator are in alternate manner.

During testing batch-normalisation update is removed so the input deterministically depends only on the input.

**Chapter 4**

**Results and Observations**

**4.1 Results**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SRCNN  PSNR: 32.50283199 | SRGAN  PSNR:31.87665999 | LR  PSNR:30.83307666 |
|  |  |  |
| HR |  |  |
|  |  |  |
| SRCNN | SRGAN | LR |
|  |  |  |
| HR |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SRCNN  PSNR: 31.08788635 | SRGAN  PSNR: 30.50603813 | LR  PSNR: 28.06814573 |
|  |  |  |
| HR |  |  |
|  |  |  |
|  |  |  |
| SRCNN | SRGAN | LR |
|  |  |  |
| HR |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SRCNN  25.87847057 | SRGAN  23.17956489 | LR  21.12223594 |
|  |  |  |
| HR |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SRCNN  PSNR: 31.06793457 | SRGAN  PSNR1: 30.74650215 | LR  PSNR: 30.01213618 |
|  |  |  |
| HR |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| SRCNN  PSNR: 28.67756786 | SRGAN  PSNR: 26.69596217 | LR  PSNR: 25.18124196 |
|  |  |  |
| HR |  |  |

**4.1.1 Set5 PSNR Results:**

**4.1.1.1 SRCNN Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| index | img\_name | initial\_psnr | final\_psnr | psnr\_gain |
| 1 | img\_001\_SRF\_3\_HR.png | 30.83307666 | 32.50283199 | 1.66975533 |
| 2 | img\_002\_SRF\_3\_HR.png | 28.06814573 | 31.08788635 | 3.019740621 |
| 3 | img\_003\_SRF\_3\_HR.png | 21.12223594 | 25.87847057 | 4.756234633 |
| 4 | img\_004\_SRF\_3\_HR.png | 30.01213618 | 31.06793457 | 1.055798388 |
| 5 | img\_005\_SRF\_3\_HR.png | 25.18124196 | 28.67756786 | 3.496325902 |

Test Statistics for SRCNN:

Unweighted initial PSNR = 27.04336729185581

Unweighted final PSNR = 29.84293826664055

Unweighted gain PSNR = 2.7995709747847415

**4.1.1.2 SRGAN Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| index | img\_name | initial\_psnr | final\_psnr | psnr\_gain |
| 1 | img\_001\_SRF\_3\_HR.png | 30.83307666 | 31.87665999 | 1.043583336 |
| 2 | img\_002\_SRF\_3\_HR.png | 28.06814573 | 30.50603813 | 2.437892402 |
| 3 | img\_003\_SRF\_3\_HR.png | 21.12223594 | 23.17956489 | 2.057328951 |
| 4 | img\_004\_SRF\_3\_HR.png | 30.01213618 | 30.74650215 | 0.7343659746 |
| 5 | img\_005\_SRF\_3\_HR.png | 25.18124196 | 26.69596217 | 1.514720212 |

Test Statistics for SRCNN:

Unweighted initial PSNR = 27.04336729185581

Unweighted final PSNR = 28.6009454

Unweighted gain PSNR = 1.55757818

**4.1.2 Set5 Mean Opinion Score(MOS) Results:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. No. | Image | No. Of people | SRCNN average score | SRGAN average score | MOS result |
| 1 | img\_001\_SRF\_3\_HR | 12 | 3.4 | 4.2 | SRGAN |
| 2 | img\_002\_SRF\_3\_HR. | 12 | 3.6 | 4.34 | SRGAN |
| 3 | img\_003\_SRF\_3\_HR. | 12 | 2.3 | 4.2 | SRGAN |
| 4 | img\_004\_SRF\_3\_HR. | 12 | 3.01 | 3.14 | SRGAN |
| 5 | img\_005\_SRF\_3\_HR. | 12 | 3 | 3.2 | SRGAN |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | Average score of Set15 |  | 3.062 | 3.816 |  |

**4.2 Observations**

From the results of Set5, we observe that PSNR results of SRCNN are better than that of SRGAN Results i.e,

Avg. SRCNN PSNR: 29.84293826664055

Avg. SRGAN PSNR: 28.6009454

But from the MOS results, we observed that SRGAN results are better than that of SRCNN results i.e,

Avg. SRCNN MOS: 3.062

Avg. SRGAN MOS: 3.816

In the original Paper, the results are as follows:

|  |  |  |
| --- | --- | --- |
| **Set5 Results** | **SRCNN** | **SRGAN** |
| **Avg. PSNR** | 32.05 | 29.50 |
| **Avg. MOS** | 3.37 | 3.58 |

**Chapter 5**

**Conclusion**

As human based results(MOS) has more perceptual assessment quality than PSNR results(analytical Results), we can conclude that SRGAN is performing better than SRCNN as SRGAN’s MOS is better despite of having less PSNR.

Also, in comparison to the original paper our results are lacking a little bit, this is due to the difference in datasets used for training.

**Chapter 6**

**Future Work**

The extension to the work is done by Wang[6] in the paper called ESRGAN. The methods used so far to generate the SR image uses various loss function and various approaches but in all this methods they only try to maximize the PSNR but increasing PSNR gives the over smooth result which disagrees with the human subjective view. In this paper some changes are made to the SRGAN.They are,

1. The Residual-in-Residual Dense Block (RDDB) is introduced which is easy to train, has high capacity and has dense connection.Next, residual scaling is used and all the Batch Normalization (BN) is removed. Removing BN has proven that it increases the performance and reduce the computational complexities.The RDDB can capture more semantic information.
2. The discriminator using Relativistic average GAN (RaGAN) is improved,previously discriminator is only used to find the image is fake or real,but now using RaGAN,discriminator learns to identify one image is more realistic than other.This helps the generator to focus on more realistic textual details and learn about the sharper edges.This is done by replacing the standard discriminator by the relativistic discriminator.
3. The perceptual loss is improved by using VGG features before activation instead of after activation in which the distance between two activated features is minimized.By adjusting the perceptual loss we get the more sharper edge and satisfied subjective view than all the methods proposed so far.

By doing this, the ESRGAN generates the better super resolution image from the low resolution image than any other SR methods.

Furthermore, this method cannot be used for x8 upscaling. x8 upscaling requires more details about semantic features about the image for which only using perceptual loss does not suffice. This requires extra parameters to take care of while super resolution.

**Chapter 7**

**References**

1. Xu, Jieping, et al. "Online multi-frame super-resolution of image sequences." *EURASIP Journal on Image and Video Processing* 2018.1 (2018): 136.
2. Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
3. Contour Detection and Hierarchical Image Segmentation P. Arbelaez, M. Maire, C. Fowlkes and J. Malik. *IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, May 2011*.
4. Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." *European conference on computer vision. Springer, Cham, 2016.*
5. C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV), pages 184–199. Springer, 2014*.
6. Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks." *Proceedings of the European Conference on Computer Vision (ECCV). 2018.*