**Single Image-Super Resolution**

***A DISSERTATION***

|  |
| --- |
| *Submitted in the partial fulfilment* |
| *Of the requirements for the award of the degree of* |

|  |
| --- |
| **Bachelor & Master of Technology in Information Technology** |
| **With Specialization in Robotics**Image result for iiita |

***By:***

|  |
| --- |
| **Atul Yadav** |
| (IRM2013007) |

|  |
| --- |
| *Under the Guidance of:* |
| **Dr. Ranjna Vyas**  IIIT-Allahabad |

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD**

**(A UNIVERSITY ESTABLISHED UNDER SEC.3 OF UGC ACT, 1956 VIDE NOTIFICATION NO.F.9-4/99-U.3 DATED 04.08.2000 OF THE GOVT. OF INDIA)**

***A CENTRE OF EXCELLENCE IN INFORMATION TECHNOLOGY ESTABLISHED BY Ministry of H.R.D., GOVT. OF INDIA***

**Candidate Declaration**

I do hereby declare that the work presented in this thesis entitled “Single Image Super-Resolution”, submitted in the partial fulfilment of the degree of Bachelor of Technology & Master of Technology with specialization in Robotics Engineering at the Indian Institute of Information Technology, Allahabad is an authentic record of my original work carried out under the guidance of Dr. Ranjna Vyas due acknowledgements have been made in the text of the thesis to all other material user. This thesis work was done in full compliance with the requirements and constraints of the prescribed curriculum.

|  |  |
| --- | --- |
| Place: | Atul Yadav |
| Date: | IRM2013007 |

**Certificate from Supervisor**

I do hereby recommend that the thesis work prepared under my supervision by Atul Yadav titled “Single Image Super-Resolution” be accepted in the partial fulfilment of the requirements of the degree of Bachelor of Technology & Master of Technology with specialization in Robotics.

|  |  |
| --- | --- |
| Place: | Dr. Ranjana Vyas |
| Date: | IIIT Allahabad |
|  |  |
|  |  |

**Certificate of Approval**

The foregoing thesis is hereby approved as a credible study in the area of Information Technology and its allied areas carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

Signature & Name of the Committee Members:

Plagiarism Report

(Coming Soon)

**Acknowledgements**

The satisfaction and euphoria that accompany the successful completion of any project work would be impossible without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts. This project was not only a endeavour but also an interesting learning experience for me and it bears the imprint of a number of persons who directly or indirectly were a source of help and constant encouragement.

I would like to express my sincere and heartly thanks to my mentor Dr. Ranjana Vyas for her continuous motivation and guidance. Her valuable suggestions, comments and support were an immense help for me. I am grateful to her for taking out time from his busy schedule and being very supportive in guiding my work.

Atul Yadav

**Abstract**

This thesis addresses the problem of image super-resolution from a low-resolution image.

Basically, single image super-resolution is a task of getting a high-resolution image from a single low resolution input. Despite increasing in accuracy and speed of single image super-resolution using faster and deeper convolution neural networks, one major problem is remains unsolved. How can we recover the finer texture details when we super-resolve at higher scaling factor. Traditionally, performance of algorithms used for such task is measured using pixel-wise reconstruction measures like peak signal-to-noise ratio (PSNR) and mean-square error(MSE). But problem with these metric is that they are missing high frequency details and they hardly match with human perception of image quality.

I have used two approaches for this problem.

First one is deep convolutional network. This method directly learns and end to end mapping from low to high resolution images. The mapping between low resolution image(input) and

high resolution image(output). Second approach is using generative adversarial network.

This approach is capable of inferring high resolution image up to scale factor of 4.

**Table of Contents**

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .1
   1. Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   2. Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   3. Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   4. Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   5. Organization of Thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. Literature Review . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   1. Approaches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2. Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
3. Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   1. Dataset Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
   2. Software and Hardware Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
4. Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .14
      1. Word Embedding Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
      2. Transfer Learning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
      3. Convolutional Neural Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
      4. Recurrent Neural Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .17
      5. Long Short Term Memory Networks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
      6. BLEU-N Evaluation Metric . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
   2. Flow Chart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
   3. Preparing Image Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
   4. Visual Features’ Extractor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
   5. Preparing Text Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
   6. Embedding Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
   7. Learning Architectures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
   8. Feature Map Visualization & Attention Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
5. Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
   1. Implant versus Merge Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
   2. One vs. Three vs. Five Captions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
   3. Ensemble Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
6. Results & Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
7. Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

References

**List of Figures**

* 1. Implant Mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  2. Merge Mode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  3. Images from FLICKR 8K . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

4.1 Regular Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4.2 Convolutional Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

4.3 Training in CNN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

4.4 Unfolded Recurrent Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

4.5 Working of Recurrent Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

4.6 Basic Structure of LSTM Network. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

4.7 Cell State in LSTM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

4.8 Forget Gate Layer in LSTM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

4.9 Input Gate and tanh layer in LSTM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

4.10 Updating Cell State in LSTM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

4.11 Output Gate Layer in LSTM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

4.12 Flow Chart of Proposed Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

4.13 VGG Architectures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

4.14 Visual Representation of VGG-16 Layer Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

4.15 VGG-16 with Soft-Max removed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

4.16 Inception Module . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

4.17 InceptionV3 Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

4.18 Example of LSTM Input & Output . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

4.19 Embedded Word Vector Example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

4.20 Implant Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

4.21 Merge Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

4.22 Ensemble Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

4.23 Image Captioning without Attention . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

4.24 Image Captioning with Attention . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30 4.25 Attention Module . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

4.26 Feature Map Visualization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

5.1 Implant Architecture Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33 5.2 Merge Architecture Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

5.3 Ensemble Implant Architecture Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

5.4 Ensemble Merge Architecture Neural Network . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

6.1 Graph representing BLEU Scores . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

6.2 Training Rate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

6.3 Tested Samples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

**List of Tables**

3.1 GPU Specs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

6.1 BLEU Scores noted on Single Caption . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

6.2 BLEU Scores noted on 1, 3 & 5 Captions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

6.3 BLEU Scores noted on Ensemble Models . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

1. **Introduction**
   1. **Overview**

High-Resolution (HR) images are useful in many applications such as surveillance video, video frame freezing, medical diagnostics and military information gathering etc.

The need for high resolution images came from two main application areas. First is improvement of pictorial information for human perception and second one is helping representation for automatic machine perception.

Basically Image resolution tells that how much amount of information contained by images. Lower resolution image would contain lesser information, higher resolution image would contain more information.

Due to its high cost and physical limitations of the image sensors and high precision optics, every time we cannot get a desired HR image for every scene. So we use Super Resolution for this purpose. Here comes the concept of super resolution (SR) image reconstruction which refers to reconstruction of a High-resolution (HR) image from one or multiple low-resolution (LR) images. Basically resolution of an image can be divided in five ways: spatial resolution,pixel resolution,temporal resolution,spectral resolution and radiometric resolution.

**Spatial resolution:** A image is made of pixels. Spatial resolution is how many pixels are in image per unit area. Current image capturing devices are using CCD and CMOS sensors. These sensors are orchestrated in two dimensional array to catch two dimensional picture signals.Number of pixels in image is determined by two factors.

1. Number of sensors per unit area.
2. Sensor size.

So one way to increase the resolution of imaging device is to increase the sensor density by reducing the size of sensors. Reducing size of sensor beyond a limit causes sht noise

In captured images because reducing size of sensor also reduces amount of light incident on it. Hardware cost will also increase.

* 1. **Motivation**

A lot of work has been done for multiple image super-resolution. But single image super-resolution has still enough scope to work.

* 1. **Problem Definition**
  2. **Organization of Thesis**

The work has been presented in seven chapters. The detailed description and summary of these chapters are as follows.

* 1. Introduction
  2. Literature Review

This chapter includes data about all the previous work done in this research field and builds a foundation for the work presented ahead.

* 1. Requirements

This chapter includes the description of the dataset used in the research work. Tools and software that are used in this work are also explained.

* 1. Methodology

The basic procedure of how the proposed work is achieved and all its modules are explained in this chapter. A short discussion on feature map visualization and attention model is also performed in this chapter.

* 1. Experiments

As the name suggests, this chapter describes the experiments that are performed. Experiments include comparison between merge and implant architecture, exploiting recurrent network behaviour on increasing number of captions per image, embedded word vector length and the combined feature extractors’ model.

* 1. Results & Analysis

After experimenting through the different architectures of the image captioning systems, results obtained are presented in this chapter. These results are further analysed and inferences are obtained.

* 1. Discussion

The study of what is done and what can be done builds the base of this chapter. A discussion is performed on the conclusion and future scope of the work.

1. **Literature Review**
   1. **Approaches**

According to image priors there are four approaches for single-image super resolution.

* Prediction Models :

SISR calculations in this class produce HR image from LR image through a predefined numerical equation without training data. Interpolation based techniques like bilinear, bicubic interpolation generate high pixel intensities by calculating weighted averaging of neighbouring of LR pixel values. These algorithms generate very good smooth regions but not effective along edges and high-frequency regions since interpolation methods considers intensities locally similar to neighbouring pixels.

* Edge Based Models :

Edges are critical crude image structures that play a prime part in visual perception.

Many SISR algorithms have been proposed to learn priors from edge features for reconstructing HR images. Various edge features such as width and depth of an edge

have been introduced. Reconstructed HR images from edge based models have high- quality edges with proper sharpness and limited artifacts [1]. Since HR images obtained this model priors are learned from edges. But edge priors are not good for modelling other high-frequency structures such as texture details.

* Patch Based Models :

In such methods basically set of paired LR and HR training images are given. For learning mapping functions patches are cropped from training images.

There are many learning methods such as weighted average, kernel regression, support vector regression, sparse dictionary representation.

* Image Statistical Methods :

* 1. **Related Work**

1. **Requirements**
   1. **Dataset Description**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. **Software & Hardware Requirements**

To keep workflow in fast pace, some already existed machine learning libraries are exploited

in the proposed work. These libraries that are availed in this work and their hardware

requirements are described in detail as follows.

* NVidia GPU

A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly

manipulate and alter memory to accelerate the creation of images in a frame buffer

intended for output to a display device. Several applications of GPU are in mobiles,

laptops, embedded circuits and play stations. Modern GPUs are very efficient at

manipulating computer graphics and image processing. This work also requires GPU

for fast processing and computation. Specifications of this GPU are given in detail as

such.

|  |  |
| --- | --- |
| Name | Persistence-M |
| Version | Quadro K6000 |
| Cores | Sixteen |
| Memory | 257838 MB |

Table 3.1: Specs of GPU

* TensorFlow

To do dataflow programming over a variety of tasks, a library was introduced by

Google Brain team for Google’s personal use. This library is known as TensorFlow. On

November 9, 2015, TensorFlow was released under the Apache 2.0 open source

license. This library is a symbolic math library which has spans its domain for

applications used in machine learning such as neural networks.

* Keras

For the purpose of built-in neural networks, a library written in Python called Keras

was developed by Google Engineer Francois Chollet. This library is open source. This

library has its backend in Theano, TensorFlow as well as CNTK. It contains several

implementations of typical neural networks’ modules like layers, activation

functions, optimization algorithms etc. It also allows training on Graphical Processing

Units.

* Anaconda

Anaconda is an open source and free distribution of the R and Python programming

languages for applications of data science and machine learning, predictive analytics

and scientific computing that simplifies the package management and deployment

by using package management utility *conda*.

1. **Methodology**
   1. **Introduction**
      1. Convolutional Neural Networks

In neural networks, a single vector is taken as input which is transformed through a

series of hidden layers. Every neural layer is a set of neurons where each neuron is

linked to all neurons in a previous layer. Each layer neurons has weights. Each layer has

an activation function. Neurons of same layer are independent of each other. Last layer

is called output layer.

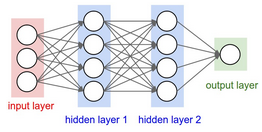


Figure 4.1: Regular Neural Network

Simple neural networks are not beneficial in terms of scaling for images. Consider

images of shape 200x200x3 RGB, so if a single neuron of input layer of neural network

is considered then it would have total of 200\*200\*3 i.e. 120,000 weights. Since there are several neurons in a single layer, so there will be more parameters. Therefore regular neural networks are not good choice for image learning tasks.

Convolutional neural network provides the solution. Layers of convolutional network

are arranged in 3D. Depth of network refers to third dimension of a layer not to the

number of layers. Neurons of convolutional network layer will only be connected to a

local region not the whole region. This type of architecture frames the full image into a

single feature vector.

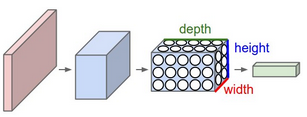


Figure 4.2: Convolutional Neural Network

Basically there are five types of layers in a convolutional network. The description of each layer is given below.

* Input Layer

This layer hold the input image, a raw pixel image can be of any dimension.

* Conv Layer

It is the core layer of a convolutional network. In this layer, activation value of neurons which are connected to the local regions of an image is computed. Computation is generally the dot product between the local region and the neuron weights.

* RELU Layer

This layer is responsible for applying element wise activation function. Choice of activation function depends on the designer of the neural net.

* POOL Layer

This layer is used to perform down-sampling in a convolutional network, reducing the spatial dimension.

* FC Layer

Fully connected layer of a convolutional network is just like a fully connected layer of a regular neural network. Each neuron in this layer is connected to every element of previous layer.

The working of a simple convolutional neural network is shown in figure 4.3. First input image is passed through convolutional layer then activations are applied using RELU layer. To down sample the data pooling layer is used. After using some combinations input is finally passed to the fully connected layer which provides us the final classification scores.

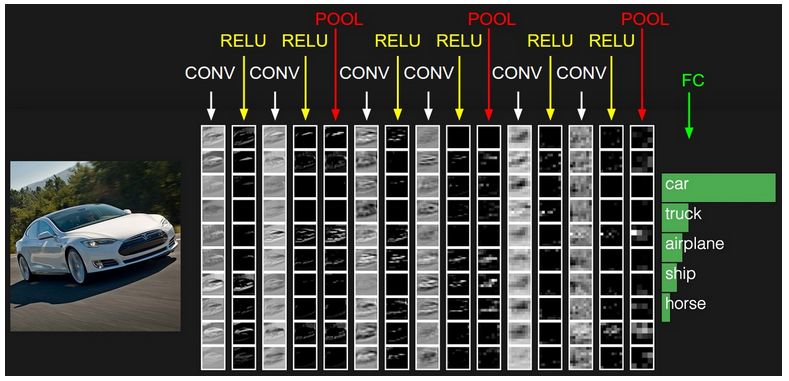
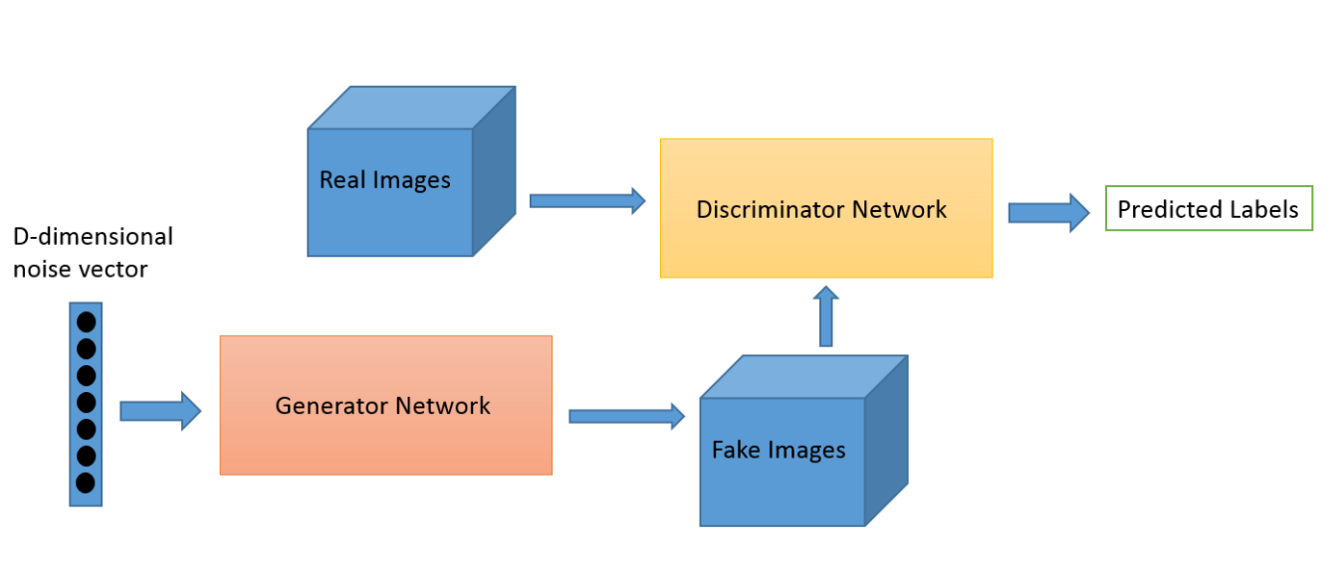


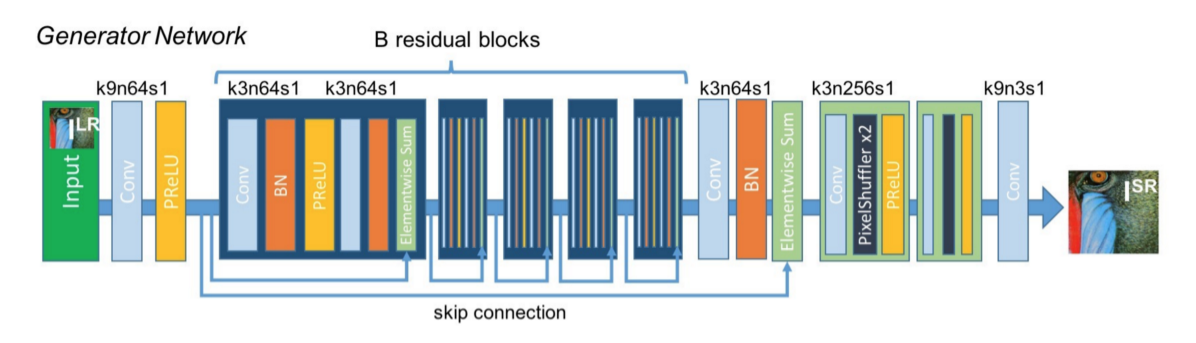
Figure 4.3: Training in CNN

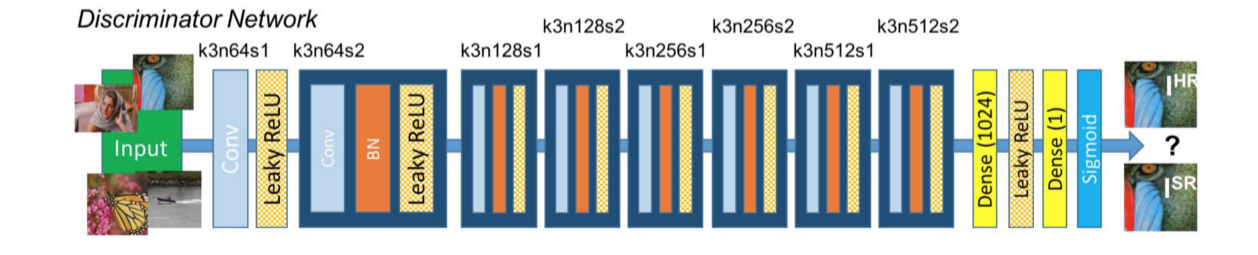
4.1..2 Generative Adversarial Networks

Generative Adversarial Networks comprise of a generator and a discriminator neural network system. The reason for the generator is to take in noise vectors and create pictures that looks like the input data distribution closely nearly and attempt to trick the discriminator into arranging a fake picture as a genuine picture. The function of the discriminator is to characterize a produced picture as genuine or fake. What is happening between the generator and the discriminator here is a 2 player zero sum game. Basically in every move, the generator is trying to maximize the chance of the discriminator misclassifying the image and the discriminator is in turn trying to maximize its chances of correctly classifying the incoming image.

Basically GAN has powerful framework for generating image look like natural images with high perceptual quality.







I have used SRGAN which is a GAN-based network optimized for a new perceptual loss.

In this instead of using MSE-based content loss I have used loss calculated on feature

maps of VGG network, we replace the MSE-based content loss with a loss calculated

on feature maps of the VGG network, which is more invariant to changes in pixel

space.

In SISR the aim is to estimate a high-resolution, super resolved image from a

low-resolution input image. The high resolution images are only available during

training.

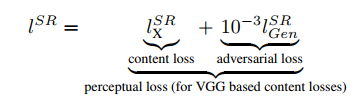
**Perceptual Loss Function:**

Basically loss function of generator network consists of content loss which is

also known as reconstruction loss and adversarial loss.

Perceptual loss can be represented as weighted sum of a content loss and an

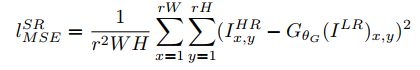
adversarial loss component.



**Content Loss:ss**

Content loss is basically pixel-wise mean squared error loss (MSE loss) between the HR

and SR images but it is not relevant to human perception.



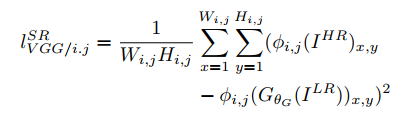
Instead of relying on pixel-wise losses I have used the ideas of Gatys[19], Bruna[5] and

Johnson[33] which is closer to perceptual similarity.

I have used VGG loss based on ReLU activation layers of pre-trained 19 layer VGG

network . Then VGG loss is defined as the Euclidean distance between the feature

representations of a reconstructed image Gθ (ILR) and reference image IHR.



Here Wi,j  and Hi,j  are dimensions of the respective feature maps within the VGG

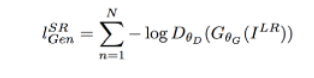
network.

**Adversarial loss:**

The adversarial loss basically favour solutions that reside on manifold of natural

images by trying to fool the discriminator network. The adversarial loss is defined by

following equation.



Here,  is the probability of reconstructed image 

to be natural image.

Discriminator uses typical cost function for training.

]

Let t = 1 mean real and t = 0 mean fake.

Then y = D(x) = p(image is real | image)

* 1. **Preparing Image Data**
  2. **Visual Feature Extractor**

Now to extract the visual information from the images, convolutional neural networks are exploited. Basically two types of convolutional networks are used in the experiments. These networks are discussed below.

* + 1. VGG
* Introduction

VGG-Architectures were built by Visual Geometry Group [16]. It is a convolutional network proposed by K.Simonyan and A. Zisserman in their paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Given an image as input, the model was built to detect the object in that image. The input image is of shape [224\*224\*3]. It is R-G-B image. The model size is 528MB. The network attains 70.5% top-1 accuracy and 92.7% top-5 test accuracy in ImageNet competition. ImageNet is a vast dataset that is a collection of over 14 million images mapped to 1000 classes.

* Architecture

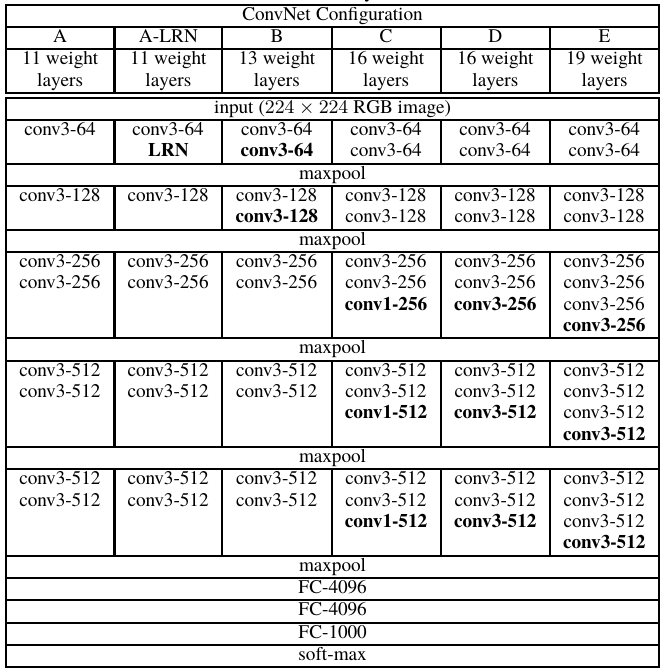
VGG-Architectures have several variations. These invariants are described in the below table. The column D indicates the VGG-16 convolutional neural network whereas column E indicates the invariant of VGG-16 known as VGG-19 network.

Figure 4.13: VGG Architectures

In the shown figure 4.13 , “conv3-512” is abbreviation of *“512 Convolutional Filters of Size 3x3”*, “FC-4096” is abbreviation of *“Fully Connected Feed Forward Layer of 4096 Neurons”*, “maxpool” indicates a “Max Pooling Layer” and “soft-max” points to the “Soft-Max Layer of A Convolutional Neural Network”.

The concise model of VGG-16 convolutional neural network can be seen in the given figure 4.14. A thing to notice is that there is a pre-processing layer which taken the RGB image as input in the range [0,255] and then perform feature-wise zero centring.

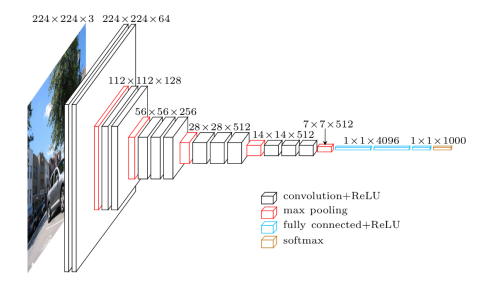


Figure 4.14: Visual Representation of VGG-16 Layer Network

* Application

In this work, VGG-16 is used as one of the visual feature extractor. The pre-processed input image as mention in section 4.3 is forwarded to the network. Now the last layer of VGG-16 is a softmax layer so this layer was removed from the network resulting into fully connected layer 4096 neurons as the last layer. Thus after passing image into the VGG-16 network, we get a fixed-size feature vector of 4096 activations as shown in figure 4.15.



Figure 4.15: VGG-16 with Soft-max Removed

* 1. **Preparing Text Data**
  2. **Learning Architectures**

1. **Experiments**

1. **Results & Analysis**
2. **Discussion**
3. **Future Scope**

**References**