Available online at: http://www.ijmtst.com/vol5issue07.html
International Journal for Modern Trends in Science and Technology
ISSN: 2455-3778 :: Volume: 05, Issue No: 07, July 2019

A Trained CNN Based Resolution Enhancement of Digital Images

D.Sowmya Krishna[1] | Bibekananda Jena[2] | Ch.Mahesh[3]

[1,2,3]Department of ECE,Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam, Andhra Pradesh, India.

To Cite this Article
D.Sowmya Krishna, Bibekananda Jena and Ch.Mahesh, "A Trained CNN Based Resolution Enhancement of Digital Images", *International Journal for Modern Trends in Science and Technology*, Vol. 05, Issue 07, July 2019, pp.-09-16.

Article Info
Received on 29-May-2019, Revised on 20-June-2019, Accepted on 28-June-2019.

ABSTRACT

Image Resolution Enhancement (RE) is a technique to estimate or synthesize a high resolution(HR) image from one or several low resolution (LR) images . Resolution Enhancement (RE) technique reconstructs a higher-resolution image or sequence from the observed LR images. In this project we are going to present about the methods in resolution enhancement and the advancements that are taking place, since it has lot many applications in various fields. Most resolution enhancement techniques are based on the same idea, using information from several different images to create one upsized image. Algorithms try to extract details from every image in a sequence to reconstruct other frames.

KEYWORDS: *Pixel, Resolution, Interpolation, Bicubic Interpolation, Resolution Enhancement , Neural Network, Convolutional Neural network, Filter, Patch, Pooling, feature map.*

I. INTRODUCTION

Resolution enhancement image reconstruction is a promising technique of digital imaging which attempts to reconstruct HR imagery by fusing the partial information contained within a number of under-sampled low-resolution (LR) images of that scene during the image reconstruction process. Resolution enhancement image reconstruction involves up-sampling of under-sampled images thereby filtering out distortions such as noise and blur. In comparison to various image enhancement techniques, resolution enhancement image reconstruction technique not only improves the quality of under-sampled, low-resolution images by increasing their spatial resolution but also attempts to filter out distortions. The central aim of Resolution enhancement is to generate a higher resolution image from lower resolution images. High resolution image offers a high pixel density and thereby more details about the original scene. The need for high resolution is common in computer vision applications for better performance in pattern recognition and analysis of images. High resolution is of importance in medical imaging for diagnosis. Many applications require zooming of a specific area of interest in the image wherein high resolution becomes essential, e.g. surveillance, forensic and satellite imaging applications . Resolution enhancement is based on the idea that a combination of low resolution (noisy) sequence of images of a scene can be used to generate a high resolution image or image sequence. Thus it attempts to reconstruct the original scene image with high resolution given a set of observed images at lower resolution. The general approach considers the low resolution images as resulting from re-sampling of a high resolution image. The goal is then to recover the high resolution image which when re-sampled based on the input images and the imaging model, will produce the low resolution observed images. Thus the accuracy of imaging model is vital for Resolution enhancement and an incorrect modeling,say of motion, can actually degrade the i mage further.

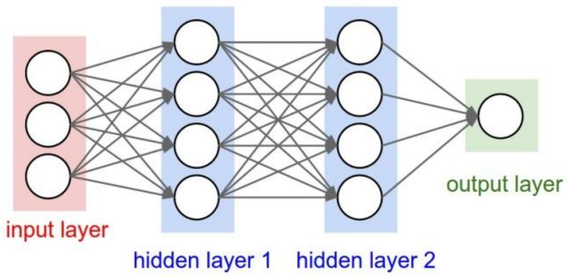


FIG 1: A simple Neural Network

II. RELATED WORK:

Based on the image priors, single-image resolution[1,4] enhancement algorithms can be categorized into four types. Those.are.prediction models, edge based methods, image statistical methods and patch based meth ods.Among them, the patch based methods achieve the high performance .The internal patch based methods exploit the self similarity property and generate exemplar patches from the input image. This method first proposed in Glasner's work, and several improve variants are proposed to accelerate the implementation. The external patch based or example-based methods learn a mapping between low/high-resolution patches from external datasets. These methods depend on how to learn the compact dictionary or manifold space to relates low/high resolution patches and how Representation schemes conducted in spaces. In the pioneer work of freemantle. the dictionaries are directly presented low/high resolution patch pair and the nearest- neighbor (NN) of the input patch is found in the low resolution space with its corresponding high resolution patch used for re-construction. In these methods the patches are the focus on the optimization. The patch extraction and aggregation steps are considered as pre/post-processing and handled separately .The majority of CNN algorithms focus on gray-scale or single channel image resolution enhancement. For colour images, the aforementioned methods first transform the problem to a different colour space (YCbCr orYUV), and CNN is applied only on the RGB channel. There are attempting to enhance resolution of all channels simultaneously.

Bicubic Interpolation Method:[2]

In image processing, bi-cubic interpolation is often chosen over bilinear interpolation[3] or nearest neighbor in image resembling, when speed is not an issue. In contrast to bilinear interpolation, which only takes 4 pixels (2x2) into account, bi-cubic interpolation considers 16 pixels (4x4).Images resembled with bi-cubic interpolation are smoother and have less interpolation distortion.

$$f(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}\, x^i y^j \quad \text{---------} \ (1)$$

Bi-cubic interpolation is often chosen over bilinear interpolation or nearest neighbour in image resembling, when speed is not an issue. Because it provide a less interpolation distortion. Bi-cubic interpolation makes use of more data, hence its results are generally smoother. Bicubic interpolation creates smoother curves than bilinear interpolation, and introduces fewer "artifacts," or pixels that stand out as conspicuously deteriorating the apparent quality of the image. Complex calculation compared to other two method described above. Greater time need to generate the output compared to bilinear and nearest neighbour methods. Appropriate color intensity values of that pixel. It then takes a weighted average of these 4 pixels to arrive at its final, interpolated value. The weight on each of the 4 pixel values is based on the computed pixel's distance (in 2D space) from each of the known points.

Experimental Evolution Stages Of CNN[6]

By stacking multiple and different layers in a CNN, complex architectures are built for classification problems. Four types of layers are most common: convolution layers, pooling /sub sampling layers, non-linear layers, and fully connectedlayers.

Convolution Layer:

The convolution operation extracts different features of the input. The first convolution layer extracts low-level features like edges and lines. Higher-level layers extract higher -level features. The input is of size h x w x d and is convolved with filter (fh x fw x d). Convolution of an input with one kernel produces one output feature, and with H kernels independently produces H features

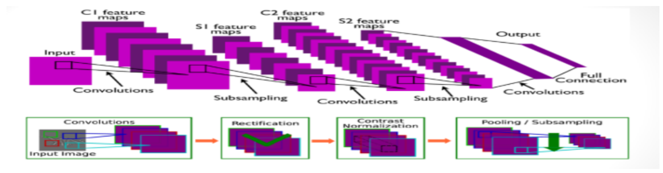


FIG2: A Schematic representation of adopted CNN

.

Starting from top-left corner of the input, each kernel is moved from left to right, one element at a time. Once the top-right corner is reached, the kernel is moved one element in a downward direction, and again the kernel is moved from left to right, one element at a time. This process is repeated until the kernel reaches the bottom-right corner. Convolution is the first layer to extract features from an input image. Convolution features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filte r or kernel Outputs a volume dimension (h-fh+1) x (w-fw+1) x1.Consider a5x5 whose image pixel values are 0, 1 and filter matrix 3 x 3 as shown in below with a filter example

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called **"Feature Map"** as output shown in below

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

*

1	0	1
0	1	0
1	0	1

FIG 3: convolution

1	1	1	0	0
0	1	1	1	0
0	**0**	**1 X1**	**1 X0**	**1 X1**
0	**0**	**1 X0**	**1 X1**	**0 X0**
0	**1**	**1 X1**	**0 X0**	**0 X1**

4	3	4
2	4	3
2	3	4

FIG 4 **:** Output of Convolution layer

IMAGE CONVOVLED FEATURE

Pooling / Sub-Sampling Layers :

The pooling /sub sampling layer reduces the resolution of the features. It makes the features robust against noise and distortion.

15	8	4	11
3	1	5	6
9	7	2	8
10	3	6	4

7	6
8	5

15	11
10	8

AVERAGE POOLING MAX POOLING

FIG 5 : AVERAGE AND MAX POOLING

There are two ways to do pooling: max pooling and average pooling.

Max Pooling:

In max pooling, the input is divided into non-overlapping two-dimensional spaces .From fig :stages of CNN Each input feature is 28x28 and is divided into 14x14 regions of size 2x2.For max pooling, the maximum value of the four values is selected

Average Pooling:

For average pooling, the average of the four values in the region are calculated

The input is of size 4x4. For 2x2 sub sampling, a 4x4 image is divided into four non-overlapping matrices of size 2x2. In the case of max pooling, the maximum value of the four values in the 2x2 matrix is the output. In case of average pooling, the average of the four values is the output. T he figure is shown above,.

Fully Connected Layer:

Fully connected layers are often used as the final layers of a CNN. These layers mathematically sum a weighting of the previous layer of features, indicating the precise mix of "ingredients" to determine a specific target output result. In case of a fully connected layer, all the elements of all the features of the previous layer get used in the calculation of each element of each output feature.

In the below diagram, feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function to classify the outputs .

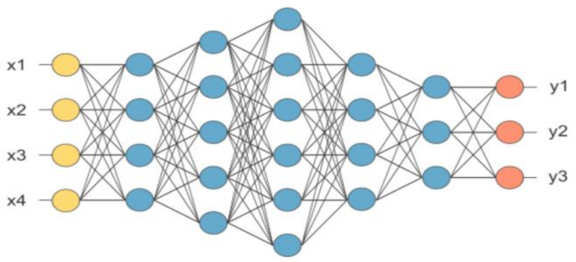


FIG 6 **:** Convolutional Neural Network With Multiple Layers

RESOLUTION ENHANCEMENT IMAGING[5]

The RECNN has several appealing properties:

1. First, its structure is intentionally designed with simplicity in mind, and yet provides superior accuracy compared with state-of-the-art example-based methods.

2. With moderate numbers of filters and layers, our method achieves fast speed.Our method is faster than a number of example-based methods, because it is fully feed-forward and does not need to solve any optimization problem on Resolution enhancement usage.

Overall, the contributions of this study are mainly in two aspects:

1) We present a fully convolutional neural network for image super-resolution. The network directly learns an end-to-end mapping between low and high-resolution images, with little pre/post processing beyond the optimization.

2) We demonstrate that deep learning is useful in the classical computer vision problem of, and can achieve good quality and speed.

Firstly, we improve the RECNN by introducing larger filter size in the non-linear mapping layer, and explore deeper structures by adding nonlinear mapping layers. Secondly, we extend the RECNN to process three color channels (either in YCbCr or RGB color space) simultaneously. The majority of RE algorithms focus on gray-scale or single-channel image super-resolution. For color images, the aforementioned methods first transform the problem to a different color space (YCbCr or YUV), and RE is applied only on the luminance channel. There are also works attempting to super-resolve all channels simultaneously.

Several factors are of central importance in this progress:

(i) the efficient training implementation on modern powerful GPUs

(ii) the proposal of the Rectified Linear Unit (ReLU) which makes convergence much faster while still presents good quality and

(iii) the easy access to an abundance of data for training larger models. Our method also benefits from these progresses.

The convolutional neural network is applied for natural image de-noising and removing noisy patterns (dirt/rain). These restoration problems are more or less de-noising -driven.

Consider a single low-resolution image, we first upscale it to the desired size using bicubic interpolation, which is the only pre-processing we perform . Let us denote the interpolated image as Y. Our goal is to recover from Y an image G(Y) that is as similar as possible to the ground truth high-resolution image X. For the ease of presentation, we still call Y a “low-resolution” image, although it has the same size as X. We wish to learn a mapping G, which conceptually consists of three operations:

Patch extraction and representation:

A popular strategy in image restoration is to densely extract patches and then represent them by a set of pre-trained bases. This is equivalent to convolving the image by a set of filters, each of which is a basis. In our formulation, we involve the optimization of these bases into the optimization of the network. Formally, our first layer is expressed as an operation G1:

$$G1(Y)=\max(0,W1*Y+B1) \text{------------------} (2)$$

where W1 and B1 represent the filters and biases respectively, and '∗' denotes the convolution operation. Here, W1 corresponds to n1 filters of support c×f1 ×f1, where c is the number of channels in the input image, f1 is the spatial size of a filter.

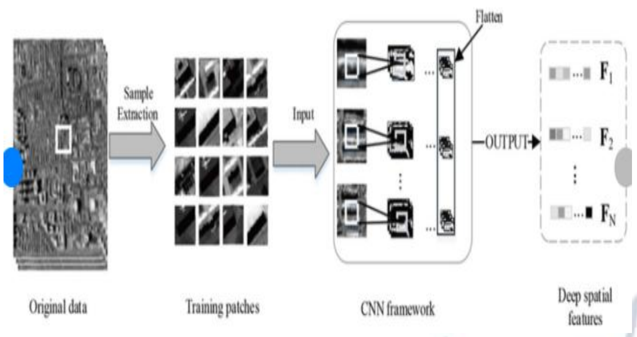


FIG 7 : Example for patch extraction

Non-linear mapping:
The first layer extracts an n1-dimensional feature for each patch. In the second operation, we map each of these n1-dimensional vectors into an n2-dimensional one. This is equivalent to applying n2 filters which have a trivial spatial support 1 × 1. This interpretation is only valid for 1×1 filters. But it is easy to generalize to larger filters like 3 × 3 or 5 × 5. In that case, the non-linear mapping is not on a patch of the input image; instead, it is on a 3 × 3 or 5 × 5 patch of the feature map.
The operation of the second layer is:

$$G2(Y)=\max(0,W2*G1(Y)+B2)\text{------------}(3)$$

Here W2 contains n2 filters of size n1 ×f2 ×f2, and B2 is n2-dimensional.
Each of the output n2-dimensional vectors is conceptually a representation of a high-resolution patch that will be used for reconstruction. It is possible to add more convolutional layers to increase the non-linearity. But this can increase the complexity of the model (n2 × f2 × f2 × n2 parameters for one layer), and thus demands more training time.

Reconstruction:
In the traditional methods, the predicted overlapping high-resolution patches are often averaged to produce the final full image.
The averaging can be considered as a pre-defined filter on a set of feature maps (where each position is the flattened vector form of a high resolution patch).
Here, we define a convolutional layer to produce the final high-resolution image:

$$G(Y)=W3*G2(Y)+B3)\text{---------}(4)$$

Here W3 corresponds to c filters of a size n2 × f3 × f3, and B3 is a c-dimensional vector.
If the representations of the high-resolution patches are in the image domain (we can simply reshape each representation to form the patch), we expect that the filters act like an averaging filter.
If the representations of the high-resolution patches are in some other domains (like coefficients in terms of some bases), we expect that W3 behaves like first projecting the coefficients onto the image domain and then averaging. In either way, W3 is a set of linear filters. Interestingly, although the abo ve three operations are motivated by different intuitions, they all lead to the same form as a convolutional layer. We put all three operations together and form a convolutional neural network (Figure). In this model, all the filtering weights and biases are to be optimized. Despite the succinctness of the overall structure, our RECNN model is carefully developed by drawing extensive experience resulted from significant progresses in super-resolution.

Training:
Learning the end-to-end mapping function G requires the estimation of network parameters U = {W1, W2, W3, B1, B2, B3}. This is achieved through minimizing the loss between the reconstructed images G(Y; U) and the corresponding ground truth high resolution images X. Given a set of high-resolution images {Xi} and their corresponding low-resolution images {Yi}, we use Mean Squared
Error (MSE) as the loss function:

$$L(U) = \frac{1}{n}\sum_{i=1}^{n} \| (G(Y_i;U) - X_i) \|^2 \quad \text{------}(5)$$

where n is the number of training samples. Using MSE as the loss function favours a high PSNR. The PSNR is a widely-used metric for quantitatively evaluating image restoration quality, and is at least partially related to the perceptual quality. It is worth noticing that the convolutional neural networks do not preclude the usage of other kinds of loss functions, if only the loss functions are derivable. If a better perceptually motivated metric is given during training, it is flexible for the network to adapt to that metric. On the contrary, such a flexibility is in general difficult to achieve for traditional “handcrafted” methods. Despite that the proposed model is trained favouring a high PSNR, we still observe satisfactory performance.

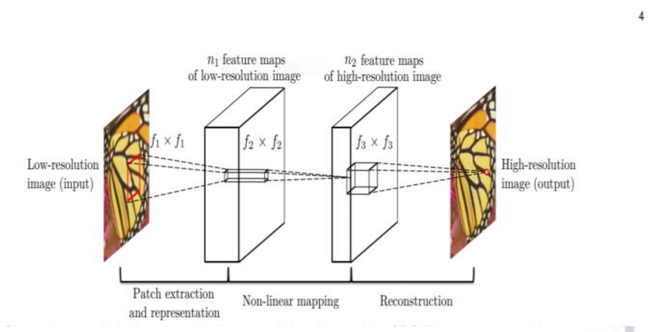

4
n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

FIG 8 Overall view of RECNN process

first, to guarantee the performance on the Y channel, we only use the MSE of the Y channel as the loss to pre-train the network. Then we employ the MSE of all channels to fine-t une the parameters.

• CbCr pre-train: we use the MSE of the Cb, Cr channels as the loss to pre-train the network, then fine-tune the parameters on all channels.

• RGB: training is performed The filter weights of each layer are initialized by drawing randomly from a Gaussian distribution with zero mean and standard deviation 0.001 (and 0 for biases). The learning rate is 10−4 for the first two layers, and 10−5 for the last layer. We empirically find that a smaller learning rate in the last layer is important for the network to converge (similar to the denoising case)In the training phase, the ground truth images {Xi} are prepared as fsub×fsub×c-pixel sub-images randomly cropped from the training images. By "sub-images" we mean these samples are treated as small "images" rather than "patches", in the sense that "patches" are overlapping and require some averaging as post-processing but "sub-images" need not. To synthesize the low-resolution samples {Yi}, we blur a sub-image by a Gaussian kernel, sub-sample it by the upscaling factor, and upscale it by the same factor via bicubic interpolation. To avoid border effects during training, all the convolutional layers have no padding, and the network produces a smaller output ((fsub − f1 − f2 − f3 + 3)^2 × c). The MSE loss function is evaluated only by the difference between the central pixels of Xi and the network output. Although we use a fixed image size in training, the convolutional neural network can be applied o n images of arbitrary sizes during testing.

Different channels:

Y only: this is our baseline method, which is a single-channel (c = 1) network trained only on the luminance channel.

The Cb, Cr channels are upscaled using bicubic interpolation.

YCbCr: training is performed on the three channels of the YCbCr space.

Y pre-trainon the three channels of the RGB space.

IV. TRAINING A NETWORK:

The learning process takes the inputs and the desired outputs and updates its internal state accordingly, so the calculated output get as close as possible from the desired output. The predict process takes input and generate, using the internal state**.**

1. *Model initialization:*

Let us assume that the desired relation between output Y and input X is Y=2.XNow, we need to optimize this function using a neural network. We are exploring which model of the generic form Y=W.X can fit the best the current dataset. Where W is called the weights of the network and can be initialized randomly

2. *Forward propagate*: The natural step to do after initializing the model at random, is to check its performance. We start from the input we have, we pass them through the network layer and calculate the actual output of the model straight forwardly. This step is called forward-propagation, because the calculation flow is going in the natural forward direction from the input -> through the neural network -> to the output.

3. *Loss function*: At this stage, in one hand, we have the actual output of the randomly initialized neural network. On the other hand, we have the desired output we would like the network to learn. The most intuitive loss function is simply loss = (Desired output—actual output). If we want the loss function to reflect an absolute error on the performance regardless if it's overshooting or undershooting we can define it as: The error function: E=∑ (desired—actual)².

4. *Differentiation*: In order to see the effect of the derivative, how much the total error will change if we change the internal weight of the neural network

with a certain small value δW. Derivatives represent a slope on a curve. Also, the derivative measures the steepness of the graph of a function at some particular point on the graph. In computational networks, the activation function of a node defines the output of that node given an input or set of inputs. When constructing Artificial Neural Network (ANN) models, one of the key considerations is selecting an activation functions for hidden and output layers that are differentiable. This is because calculating the back propagation error is used to determine ANN parameter updates that require the gradient of the activation function for updating the layer.
The advantage of using the mathematical derivative is that it is much faster and more precise to calculate.
In the figure, y-error function; w-weight

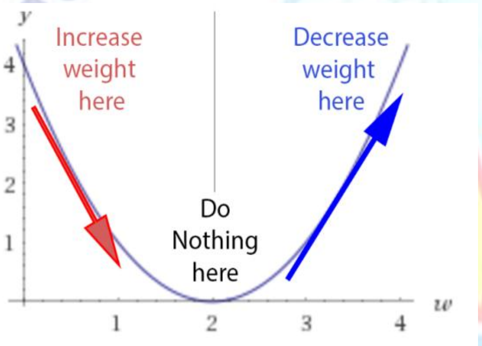
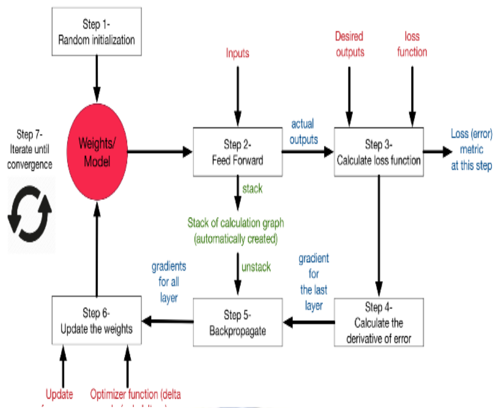


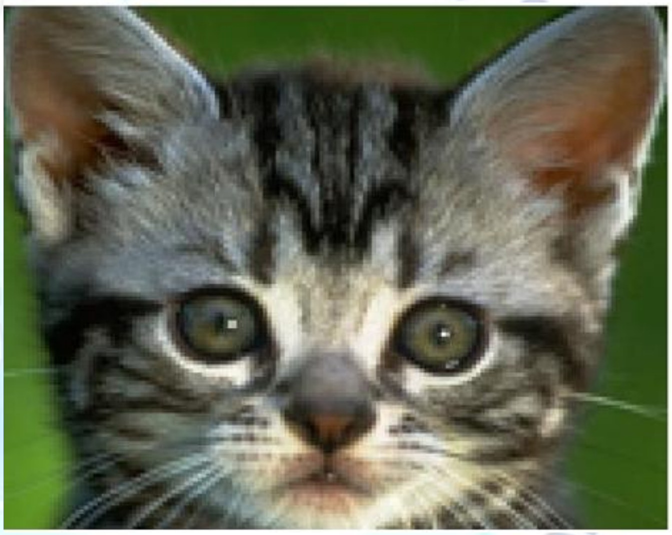
FIG 9: Weight and error relation

5- *Back-propagation*: Back propagation is shorthand for "the backward propagation of errors," since an error is computed at the output and distributed backwards throughout the network's layers. It is commonly used to traindeep neuralnetworks, a term referring to neur al networks with morrate is introduced as a constant
(usually very small), in order to force the weight to . Get updated very smoothly and slowly. In order to validate this equation if the derivative rate is positive, it means that an increase in weight will increase the error, thus the new weight should be smaller. If the derivative rate is negative, it means that an increase in weight will

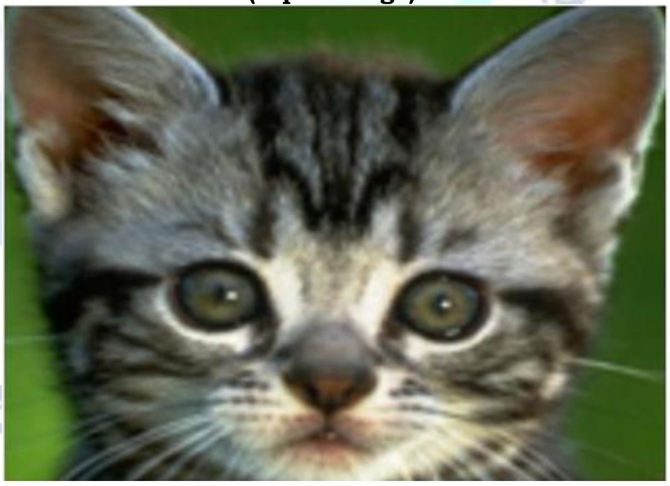


.
FIG 10: Block diagram of CNN using back propagation

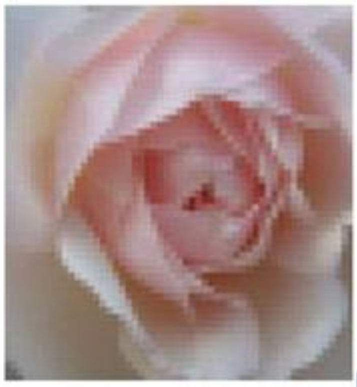
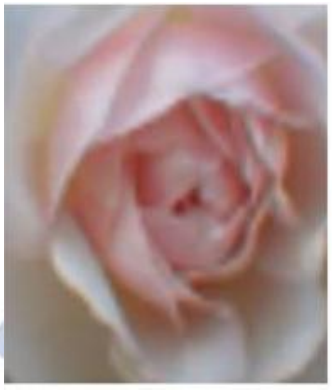
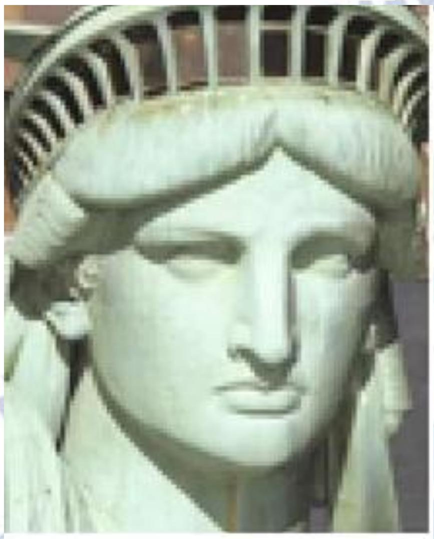
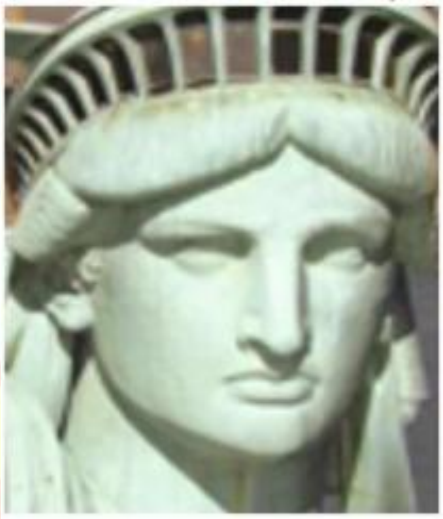
V. RESULTS



(Input Image)



PSNR: 55.7862

Low Resolution Image	Resolution Enhanced (Output of CNN)	PSNR
		51.033937
		59.099285

References

[1] Hassan Aftab ; Atif Bin Mansoor ; Muhammad Asim A New Single Image Interpolation Technique For Super Resolution , 2008 IEEE International Multitopic Conference

[2] Prachi R Rajarapollu, Vijay R Mankar Bicubic Interpolation Algorithm Implementation for Image Appearance Enhancement ," International Journal of Computer Science And Technology" Vol. 8, Issue 2, April - June 2017

[3] R. Matsuoka, M. Sone, N. Sudo and H. Yokotsuka, "Comparison of Image Interpolation Methods Applied to Least Squares Matching," *2008 International Conference on Computational Intelligence for Modelling Control & Automation,* Vienna, 2008

[4] An Introduction to Super-Resolution Imaging Jonathan Simpkins and Robert L. Stevenson Mathematical Optics: Classical, Quantum, and Computational Methods 2012.

[5] Azade Mokari Electrical and Robotic "An adaptive single image method for super resolution" SPIS2015, 16-17 Dec. 2015

[6] Dong, C., Loy, C.C., He, K., Tang, X.: Learning a Deep Convolutional Network for Image Super-Resolution. In: European Conference on Computer Vision, pp. 184–199 (2014)