



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 3, March 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Image Super-Resolution Using an Efficient Sub-Pixel CNN

Munnangi Dinesh Reddy¹, Kottu Sai Kumar², Meda Leela Naga Prasad³

U.G. Student, Department of Electronics and Communication Engineering, Vasireddy Venkatadri Institute of Technology, Nambur, Guntur, Andra Pradesh, India^{1,2,3}

ABSTRACT: This project implements a deep learning model using convolutional neural network with an efficient Sub-pixel convolution layer as "Image Super Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". Increasing a picture's resolution by creating a high-resolution (HR) image from a low-resolution (LR) image is known as single-image super-resolution. Lately, a number of neural network-based models have excelled in terms of single picture super-resolution reconstruction accuracy and computing performance. These techniques use a single filter, often bicubic interpolation, to upgrade the low resolution (LR) input picture to the high resolution (HR) space prior to reconstruction. The super-resolution (SR) operation is therefore carried out in HR space, according to this. We show that doing so increases computing complexity and is not ideal. The first convolutional neural network (CNN) is presented in this research, and in order to do so, we provide a CNN architecture in which the feature maps are extracted in the LR space. The final LR feature maps are upscaled into the HR output using an effective sub-pixel convolution layer that learns a variety of upscaling filters.

KEYWORDS: High-Resolution, Low-Resolution, Feature maps, Upscaling filters

I. INTRODUCTION

To create a higher quality image from a lower resolution image is the main goal of super-resolution (SR). A high pixel density high resolution picture provides more information about the original situation. High resolution is frequently required in computer vision applications to improve performance in pattern recognition and picture analysis. In medical imaging, a high resolution is crucial for diagnosis. For many applications, such as surveillance, forensic, and satellite imaging, zooming onto a particular region of interest in the picture becomes necessary.

During the image reconstruction process, the Super Resolution image reconstruction method aims to rebuild HR imagery by merging the incomplete information present in a series of under sampled low-resolution (LR) photographs of the scene. Up sampling of under-sampled pictures for super-resolution image reconstruction removes defects like noise and blur [3]. In compared to other image enhancement methods, the super-resolution image reconstruction methodology aims to eliminate distortions while also enhancing the quality of low-resolution, under-sampled photos.

The common methodology assumes that the low-resolution pictures are the result of resampling a high-resolution image. The next step is to recover the high-resolution picture that, after being resampled using the input photos and the imaging model, would result in the low resolution observed images. As a result, for super-resolution, the correctness of the imaging model is crucial, and improper modelling, like that of motion, can actually cause the image to get worse. The observed pictures may have come from a single camera, a group of cameras, or individual frames from a video series [1]. It is necessary to map these pictures to a common reference frame. This is the registration procedure. The aligned composite image's region of interest can then be subjected to the super-resolution process.

Hence, resolution improvement is still required, particularly in areas like video surveillance, medical diagnostics, and remote sensing applications. Super-resolution (SR) signal processing techniques have emerged as a potential means of obtaining high-resolution (HR) images due to the high cost and restrictions of resolution augmentation using "hardware" techniques, particularly for large-scale imaging systems. Instead of enhancing the technology, SR approaches allow us to go beyond the limits of low-resolution (LR) observations[2].

There are numerous ways to define the resolution. such as spectral resolution, pixel resolution, spatial resolution, and temporal resolution. The term "resolution" in terms of pixels refers to the overall pixel count in a digital image. For instance, an image's resolution can be described as M X N if it has M rows and N columns [3]. The image quality rises with increasing pixel resolution.

An image's resolution can be divided into two categories.

- a) Low-Resolution image
- b) High-Resolution image



Fig. 1. Types of images (a) Low-resolution image, (b) High-resolution image

II. PROPOSED MODEL

A. Efficient Sub-Pixel Convolutional Neural Network:

To address SRCNN's complexity problem as it quadratically increased, ESPCN was created. When upscaling factor, n , was applied to LR image using bicubic interpolation, this flaw in SRCNN resulted in excessive computation cost with a factor of n^2 [2]. Moreover, the interpolation technique did not contribute any new information to help with the poorly stated reconstruction problem. With the exception of the up-sampling module, the design was the same as that of SRCNN. Instead of bicubic interpolation sub-pixel convolution is used[7].

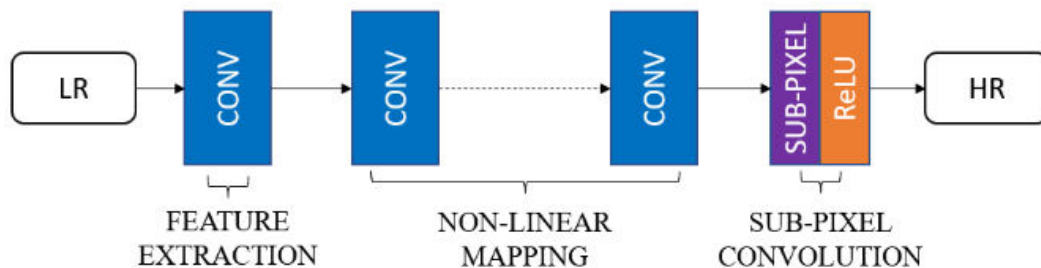


Fig. 2. ESPCN network model

B. Activation function-ReLU:

A rectified linear unit (ReLU) is an activation function that gives a deep learning model the ability to be non-linear and addresses the vanishing gradients problem. It interprets the conclusive aspect of its case. One of the most well-liked deep learning activation functions is ReLU. Sigmoid and Tanh are quite popular as they were monotonous, differentiable. But, these functions experience saturation, which causes issues with vanishing gradients. The Rectified Linear Unit is a substitute and the most widely used activation function to get around this problem (ReLU) [1]. ReLU was not widely employed until recently primarily due to the fact that it was not differentiable at zero. Differentiable functions like sigmoid and tanh were frequently used by researchers until they knew how ReLU is efficient. ReLU is now recognised as the ideal activation function for deep learning, though[4].

C. ESPCN Model:

Fundamentally, the SR model takes the LR blurred and noisy image as its input data. By applying a down sampling procedure to HR images obtained from the datasets, the LR images can be produced [5]. And the rebuilt SR pictures with the chosen upscale factor will be the result. Assuming the network has L layers, the input LR pictures are first transformed into feature maps by the first $L-1$ convolutional layers [7]. The effective sub-pixel convolutional layer is

the final layer, and it recovers the output image size with a predetermined upscale factor. The network typically has three layers:

- The input image has the following shape: $[B, C, N, N]$
- The first layer is a convolutional layer with 64 filters and a 5×5 kernel size, followed by a ReLU activation layer.
- The second layer is a 3×3 convolutional layer with 32 filters, followed by a ReLU activation layer.
- The third layer is a convolutional layer with a kernel size of 3×3 and a set number of output channels $C * r * r$.
- At the end, sub-pixel shuffle function is applied such that the output SR picture will have the shape $[B, C, r * N, r * N]$.

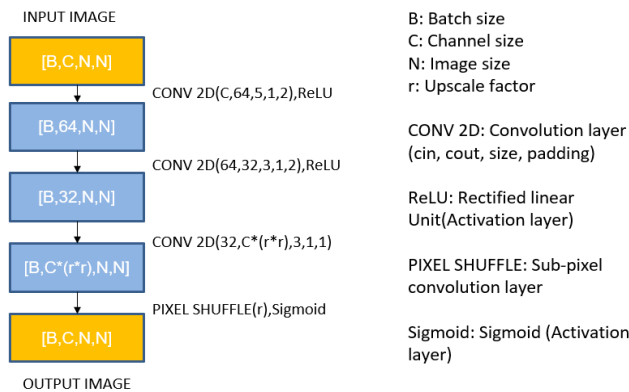


Fig. 3. ESPCN flowchart

D. Sub-Pixel Convolution:

Pixel shuffle is another name for sub-pixel convolution. Images are confined to their original pixel resolution because of the light sensor's limitations; in other words, each pixel on the photographs represents a relatively small region of colour in the real world. While there are many tiny pixels between the two actual pixels in the microscopic world, there are only a few connected pixels in a digital image. Sub-pixels are the name for the tiny pixels [5]. The pixel on the camera's imaging plane is each square area bordered by four tiny red squares; the black dots are its sub-pixels. The interpolation between the adjacent pixels can be used to change the precision of sub-pixels. Sub-pixel interpolation enables the mapping of small square areas to large square areas in this manner.

A general convolutional operation is followed by the rearrangement of pixels in the process of sub-pixel convolution. Thus, that the overall pixel count is consistent with the desired HR image, the output channel of last layer must be $C * r * r$ [5]. The interpolation mechanism is implicitly included in the convolutional layers of the ESPCN network, and the network can automatically learn it. The efficiency is substantially higher because convolution procedures are used on LR images that are smaller in size.

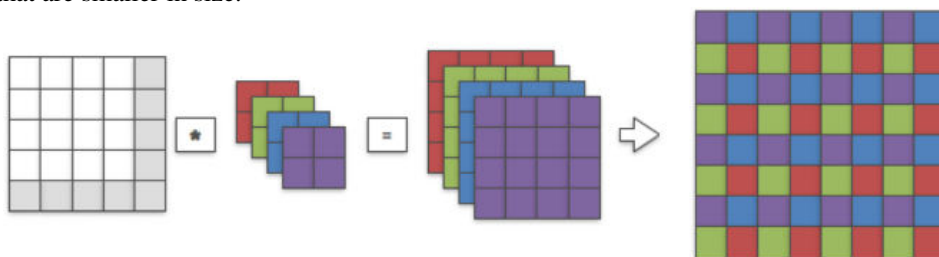


Fig. 4. Sub-Pixel Convolution (Pixel Shuffle)

III. IMPLEMENTATION

A. Dataset, Importing modules:

We have combined many datasets and used them to train our model. They are BSDS-500, BSDS-300, Set-5, Set-14, DIV-2K, Urban-100, Flower Image Dataset from Kaggle and some of our own images. We have collected nearly two thousand images for training and validation. There are 200 images for testing. Importing Keras from TensorFlow, NumPy, OS, Math modules in python. Also, some other modules for printing or displaying picture etc. Giving the path of the data set library as root directory. After retrieving the data set and created training and validation sets separately from root directory containing our dataset.



B. Changing image colour and size:

Converting pictures to YUV colour space from RGB colour space. Cropping the image to get the y channel (luminance) and resizing it using the area approach were done for the input data (low-resolution photographs). Humans are more sensitive to luminance shift; hence we solely take the luminance channel into account in the YUV colour system[3].

C. Creation of low-resolution image:

When given an LR image that has been downsampled from the corresponding original HR image, the single image super representation (SISR) task is to estimate an HR image I^{SR} [1]. The down sampling process is known and deterministic; to create I^{LR} from I^{HR} , we first convolve I^{HR} through a Gaussian filter [11], imitating the point spread function of the camera. by reducing the image's sampling size by r. Taking into account the scaling factor r. Both I^{LR} and I^{HR} are real-valued tensors of size $H*W*C$ and $rH*rW*C$, respectively, and both can have C colour channels [6].

D. Upscaling the image using ESPCN:

Due to the SISR issue and from a version of I^{LR} that has been upsampled and interpolated in place of I^{LR} . A 3-layer convolutional network is utilised to retrieve I^{SR} . To prevent upscaling I^{LR} before feeding it into the network, new network architecture is shown in this section using a network model. In the architecture, the LR picture is first subjected to an L-layer convolutional neural network followed by a sub-pixel convolution layer that upscales the LR feature maps to yield I^{SR} [3].

$$G_1(I^{LR}; W_1, \beta_1) = O(W_1 * I^{LR} + \beta_1) \dots\dots\dots (1)$$

$$G_1(I^{LR}; W_{1:L}, \beta_{1:L}) = O(W_L * g^{L-1} I^{LR} + \beta_1) \dots\dots (2)$$

Where $W_L, \beta_L, l \in (1, l-1)$ are weights and biases the network model respectively. W_L is a 2D convolution size $n_{L-1} \times n_L \times k_L \times k_L$, where n_L is the number of features at layer L, $n_0 = C$, and k_L is the filter size at first layer. The biases β_L are vectors of length n_L . The activation function which is non-linear and fixed, O which is ReLU is applied element-wise. The last layer G_1 has to convert the LR feature maps to a HR image which is the super-resolution image output.

By setting $L = 3, (g_1, n_1) = (5, 64), (g_2, n_2) = (3, 32)$, and $g_3 = 3$ in evaluations, the ESPCN is represented. The 3-layer 9-5-5 model and equations (1) and (2) of SRCNN serve as inspirations for the parameter selection [8]. The training ground truth images I^{HR} are broken down into $17r*17r$ pixel sub-images, where r is the upscaling factor. We blur I^{HR} using a Gaussian filter and sub-sample it by the upscaling factor to synthesise the low-resolution samples I^{LR} . This makes sure that each pixel in the original image only appears once while serving as the training data's ground truth.

E. Defining call-backs to monitor training:

The ESPCN Callback object will compute and display the PSNR metric. This is the main metric we use to evaluate super-resolution performance [1].

In conjunction with training using `model.fit()`, the `ModelCheckpoint` callback is used to save a model or set of weights (in a checkpoint file) at regular intervals. The model or set of weights can then be loaded later to resume training from the state saved. When training using an iterative technique, `EarlyStopping` is employed to prevent overfitting. assuming a training's objective is to reduce loss. With this, "loss" would be the statistic to be tracked, and "min" would be the mode [9]. A `model.fit()` training loop will evaluate the patience and `min_delta` if necessary, when determining whether the loss is no longer decreasing at the conclusion of each epoch. `model.stop_training` is set to True and the training is stopped once it is determined that it is no longer decreasing [2].

F. Architectural information:

The model's architecture is very similar to that of the SRCNN, with the exception of the use of 4 convolution layers in place of 3 at the final stage of upscaling rather than the bicubic interpolation employed in the SRCNN. Between the Y component's original HR photos and the matching Y component's predicted SR images, the backpropagation mean squared error was minimised [4].



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 1)]	0
conv2d (Conv2D)	(None, None, None, 64)	1664
conv2d_1 (Conv2D)	(None, None, None, 64)	36928
conv2d_2 (Conv2D)	(None, None, None, 32)	18464
conv2d_3 (Conv2D)	(None, None, None, 9)	2601
tf.nn.depth_to_space (TFOpL ambda)	(None, None, None, 1)	0
=====		
Total params: 59,657		
Trainable params: 59,657		
Non-trainable params: 0		

Fig. 5. Model summary

G. Evaluation index:

A mathematical technique based on image pixel data is called PSNR [10]. By calculating the difference between the grey scale values of the pixels in the final image that match to the original image, it employs statistical methods to assess the quality of the image. Equation (3) is the PSNR calculation formula, where F and R are the original and resultant images, respectively, and MxN are their respective sizes. N denotes the image width where M is the image height.

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |R(i,j) - F(i,j)|^2} \dots \dots \dots (3)$$

The disparity between the final image and the original image is lower with a higher PSNR value, indicating greater image quality. The areas of image denoising and picture super-resolution can both benefit greatly from this technique, which is rather straightforward and simple to apply. The local visual factors of the human eye are not taken into account, however, as the PSNR is based on the global statistics of image pixel values. Human eyes have variable sensitivity to different locations, and the perception of a particular area is also influenced by its surroundings and nearby places, therefore the evaluation results of PSNR might have differed from how the human eye sees things [12].

IV. EXPERIMENTAL RESULTS

A. PSNR value after training:

The PSNR value after training the model after 100 epochs is improved and the loss value is decreased. By this we can say our model is efficient. The noise is gradually decreased and the loss also decreased.

```
Epoch 93/100
Mean PSNR for epoch: 28.22
125/125 - 5s - loss: 0.0015 - val_loss: 0.0014 - 5s/epoch - 43ms/step
Epoch 94/100
Mean PSNR for epoch: 28.53
125/125 - 7s - loss: 0.0015 - val_loss: 0.0014 - 7s/epoch - 54ms/step
Epoch 95/100
Mean PSNR for epoch: 28.26
125/125 - 5s - loss: 0.0015 - val_loss: 0.0014 - 5s/epoch - 41ms/step
Epoch 96/100
Mean PSNR for epoch: 28.47
125/125 - 5s - loss: 0.0015 - val_loss: 0.0014 - 5s/epoch - 42ms/step
Epoch 97/100
Mean PSNR for epoch: 28.67
125/125 - 6s - loss: 0.0015 - val_loss: 0.0014 - 6s/epoch - 50ms/step
Epoch 98/100
Mean PSNR for epoch: 28.58
125/125 - 6s - loss: 0.0015 - val_loss: 0.0014 - 6s/epoch - 52ms/step
Epoch 99/100
Mean PSNR for epoch: 28.36
125/125 - 5s - loss: 0.0016 - val_loss: 0.0014 - 5s/epoch - 40ms/step
Epoch 100/100
Mean PSNR for epoch: 28.45
125/125 - 5s - loss: 0.0015 - val_loss: 0.0014 - 5s/epoch - 40ms/step
<tensorflow.python.checkpoint.checkpoint.CheckpointLoadStatus at 0x7ff0d0548d30>
```

Fig.6. PSNR after 100 epochs

B. Testing results:

The PSNR and prediction of an example low resolution image is observed, that the predicted image’s has better quality than the low-resolution image. The PSNR value is the comparison between two images which shows the signal to noise ratio (signal mean pixels in the image, noise mean lost pixels or blurred pixels). The ratio when compared an image to its original image is our metric in this project. If PSNR is more it has more quality when referred to the original image. In the testing results the PSNR of low-resolution image compared with original image is less than the PSNR of predicted image compared with original image. This mean our model is showing accurate results.

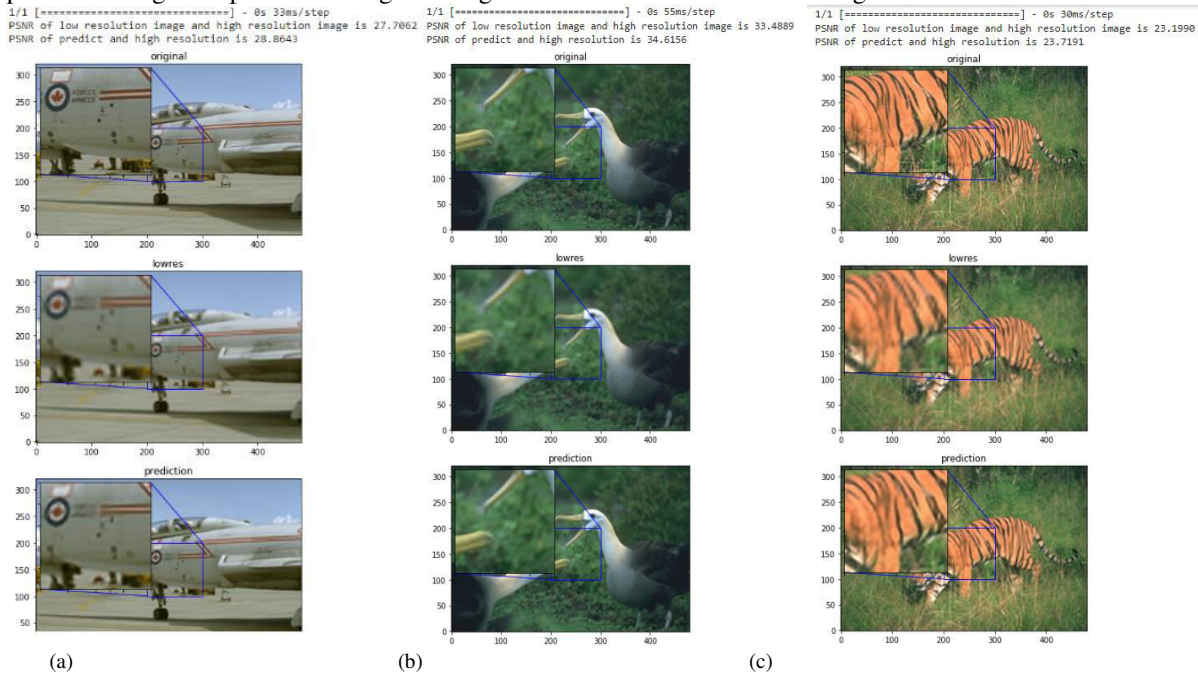


Fig. 7. (a), (b), (c) Testing results (Original, Low-resolution, High-resolution images)

C. Training with different datasets:

The model has been trained with various individual Datasets and the PSNR value is noted below.

Table 1. PSNR values with various datasets

Dataset	PSNR value
Set 5	33.13
Set 14	29.49
Our combined Dataset	26.12
BSDS 500	28.64
DIV 2k	26.70
Average	28.82

We have trained our model with different image datasets because to test the efficiency and accuracy while testing every type of data set like small images, large images, low quality, high quality etc. We got a PSNR value of 28.82 on average with every dataset we have trained, so we can say our model is an upgraded and useful model. This can be seen by the PSNR value we have obtained. It can be seen that ESPCN as super-resolution reconstruction method, maintain the image details very well, and the main details are blurred

D. Comparing with previous models:

Every model has been trained and noted down the PSNR values.

Table 2. PSNR values comparison

Model	PSNR value
SRCNN	23.76
Bicubic	23.98
ESPCN-Tanh	25.01
ESPCN-ReLU (our model)	26.12

Our model has been compared with other models and we can see the improvement in our model. These above models have been trained with our dataset and shown the value here. So, we can say our model produced efficient output.

V. CONCLUSION

In contrast to the conventional CNN model, we constructed a considerably smaller and faster ESPCN model in this study. We used ReLU, which is faster than Tanh, to develop our model in order to address the time constraints. Losses occurred while using Leaky ReLU are also prevented by our model. After comparison, we discovered that our model's generated image appears to be significantly more detailed to human perception than earlier models. This model can be used to train computer vision models. The ESPCN's performance is also at its peak, which can promote the development of the image recognition system's performance. The super-resolution using an efficient sub-pixel CNN is improved compared to all its previous versions by our different activation function and trained our model with more than two thousand images which is large data for a super-resolution model. Our model's predicted image shows the difference from the low-resolution image to the naked eye without referring the PSNR values.

VI. FUTURE WORK

We will improve our model accuracy by learning new methods and using new algorithms which are currently popular. We are looking ahead to improve our image super-resolution to video super-resolution. Our model will be tested in spatial-temporal network to super resolve multiple neighbouring frames at a time. Our model will be trained with more datasets and we will improve its speed and accuracy so it can be used as un-pixelation software.

REFERENCES

- [1] Single Image Super Resolution using ESPCN-With SSIM: Loss Dam Ja Ba Nay (Bachelor of Data Science, S P Jain School of Global Management Sydney, Australia) January 10, 2021.
- [2] Super-resolution Restoration of Single Vehicle Image Based on ESPCN-VISR Model: Mengxi Xu et al (IOP Conf. Ser.: Mater. Sci. Eng. 790 012107) 2020.
- [3] Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network: Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang (Magic Pony Technology Imperial College London) September 16, 2016.
- [4] Image Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network Soham Sheth (Department of Electronics and Telecommunications, Pune VidhyarthiGriha's College of Engineering and Technology, India) 2021 IJCRT volume 9, 11 November 2021.
- [5] An Overview of ESPCN: An Efficient Sub-Pixel Convolutional Neural Network: Zhuo Cen (<https://medium.com/@zhuocen93/an-overview-of-espcn-an-efficient-sub-pixel-convolutional-neural-network-b76d0a6c875e>) April 17, 2020.
- [6] Review: ESPCN-Real Time SR: SikHoTsang (<https://medium.datadriveninvestor.com/review-espcn-real-time-sr-super-resolution8dceca249350>) November 30, 2018.
- [7] Deep Learning Algorithms for Single Image Super-Resolution- A Systematic Review: Yoong Khang Ooi and Haidi Ibrahim, Electronics 2021, 10, 867. (<https://doi.org/10.3390/electronics10070867>), April 6, 2021.
- [8] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network: Ledig C, L Huszar F, Caballero J, Cunningham A, Acosta A, Shi W, (<https://arxiv.org/abs/1609.04802v5>)29 October 2020.
- [9] "Mean squared error.": Wikipedia, The Free Encyclopedia. (https://en.wikipedia.org/wiki/Mean_squared_error) 29 Jan. 2021.



- [10] Survey of single image super-resolution reconstruction: Kai Li, Shenghao Yang, Runting Dong, Xiaoying Wang, Jianqiang Huang July 23, 2020.
- [11] Super Resolution Reconstruction of Images Based on Interpolation and Full Convolutional Neural Network and Application in Medical Fields: Na Sun, HuinaLi December 23, 2019.
- [12] Deep Learning for Single Image Super-Resolution: A Brief Review: Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, Qingmin Liao July 12, 2019.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details