
# SINGLE IMAGE SUPER RESOLUTION USING SRCNN

**Presented By : Spandan Chatterjee**
**Roll no. : 25MA60R06**

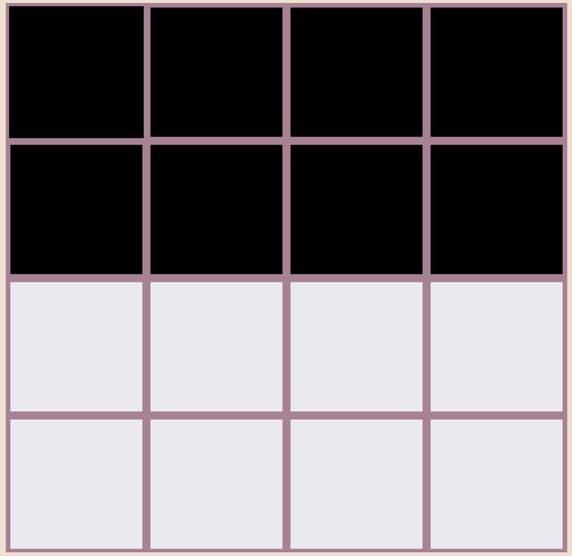
**IIT Kharagpur | 2025**

# OVERVIEW

- What is Image?
- Super Resolution
- SRCNN Overview
- Convolution Operation
- Padding
- SRCNN Architecture
- Image Processing
- Evaluation Metric
- Results
- Conclusion

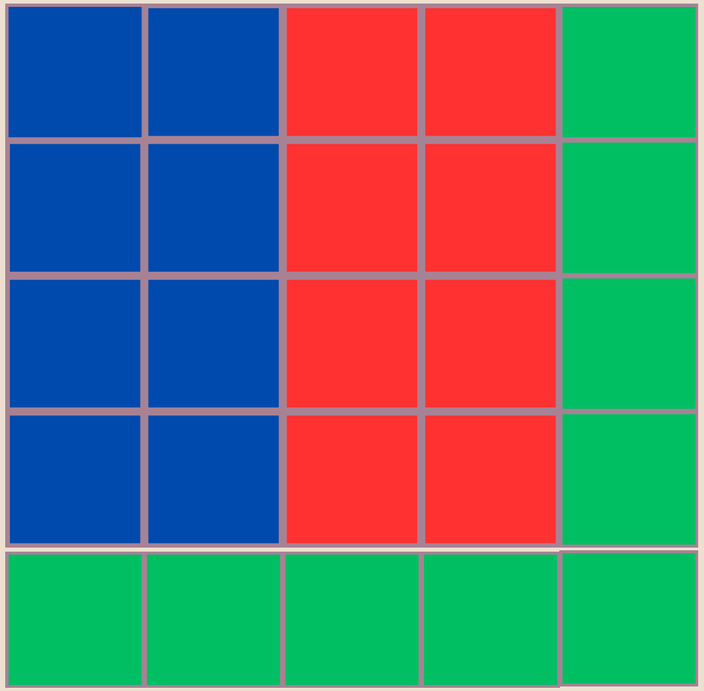
# What is Image ?

- An image is a 2D representation of visual information.
- It is made up of tiny square elements called pixels.
- Each pixel stores a value representing brightness or color.



| | | | |
|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 |

# What is Image ?



5x5

RGB Channel

| | | | | |
|---|---|---|---|---|
| 255 | 255 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

| | | | | |
|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 255 |
| 0.0 | 0.0 | 0.0 | 0.0 | 255 |
| 0.0 | 0.0 | 0.0 | 0.0 | 255 |
| 0.0 | 0.0 | 0.0 | 0.0 | 255 |
| 255 | 255 | 255 | 255 | 255 |

| | | | | |
|---|---|---|---|---|
| 0.0 | 0.0 | 255 | 255 | 0.0 |
| 0.0 | 0.0 | 255 | 255 | 0.0 |
| 0.0 | 0.0 | 255 | 255 | 0.0 |
| 0.0 | 0.0 | 255 | 255 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

5x5x3

# What is Super Resolution ?

- Enhancing the Resolution of an Imaging System
  - *a measure of amount of details in an image*

***LR***

***HR***

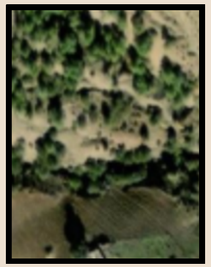
## *Application*

- Satellite Imaging
- Medical imaging
- Surveillance
- Old Image Restoration

# Some Mathematical Methods for Super Resolution ?

- *Nearest Neighbor*
- *Bilinear Interpolation*
- *Bicubic Interpolation*

# Super Resolution Convolutional Neural Network (SRCNN)



*LR Image*

*Bilinear Interpolation*

*High Resolution + Low Quality*

*Applied SRCNN*

*High Resolution + Good Quality*

*Ground Truth*

# Convolution Operation

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

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

# Convolution Operation

-1x0 -1x0 -1x0 + 0x0 + 0x0 + 0x0 + 1x0 + 1x0 + 1x0

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

| | | | |
|---|---|---|---|
| 0 | | | |
| | | | |
| | | | |
| | | | |

# Convolution Operation

-1x0 -1x0 -1x0 + 0x0 + 0x0 + 0x0 + 1x0 + 1x0 + 1x0

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

| | | | |
|---|---|---|---|
| 0 | 0 | | |
| | | | |
| | | | |
| | | | |

# Convolution Operation

-1x0 -1x0 -1x0 + 0x0 + 0x0 + 0x0 + 1x0 + 1x0 + 1x0

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | |
| | | | |
| | | | |
| | | | |

# Convolution Operation

-1x0 -1x0 -1x0 + 0x0 + 0x0 + 0x0 + 1x0 + 1x0 + 1x0

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

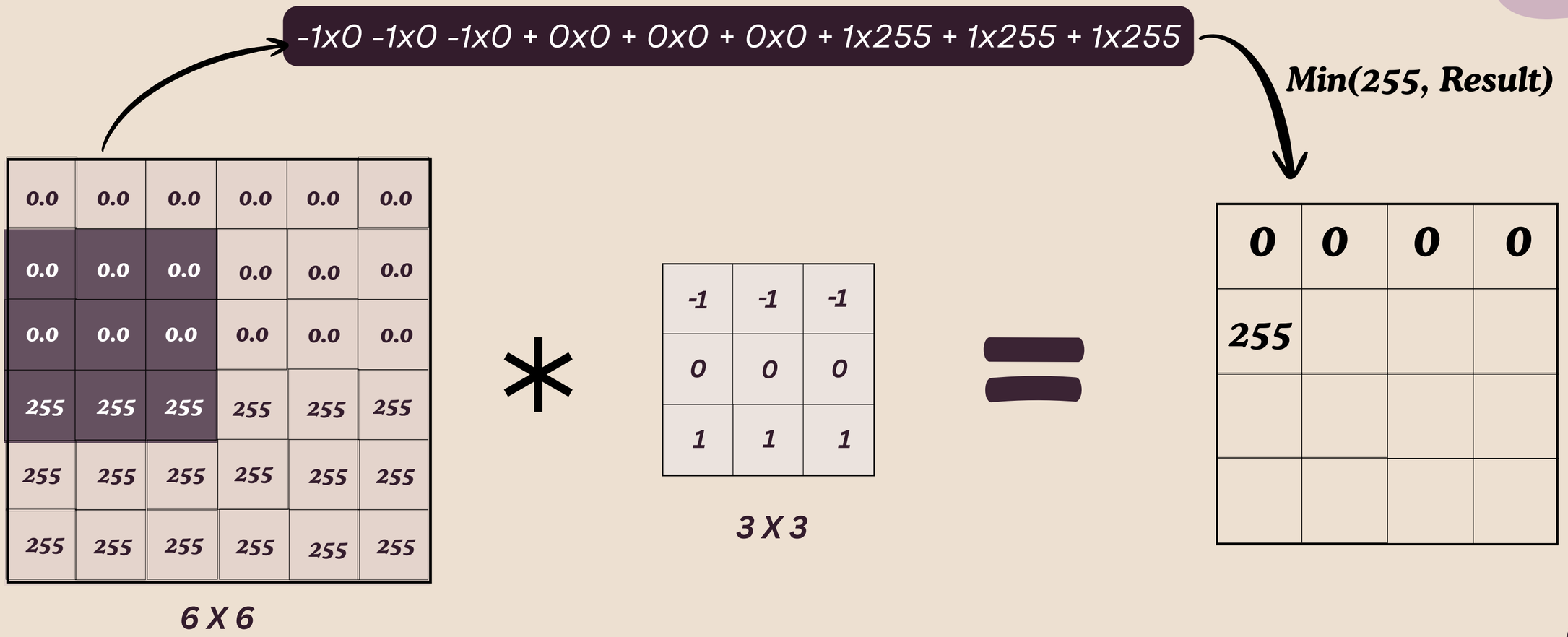
| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| | | | |
| | | | |
| | | | |

# Convolution Operation



# Convolution Operation

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 |
| 0 | 0 | 0 | 0 |

# Convolution Operation

**Image**

| | | | | | |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 |

6 X 6

*

**Filter**

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3 X 3

=

**Feature Map**

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 |
| 0 | 0 | 0 | 0 |

4 X 4

**Image Size : n x n,**
**Filter Size : m x m,**
**Then Output size : (n-m+1) x (n-m+1)**

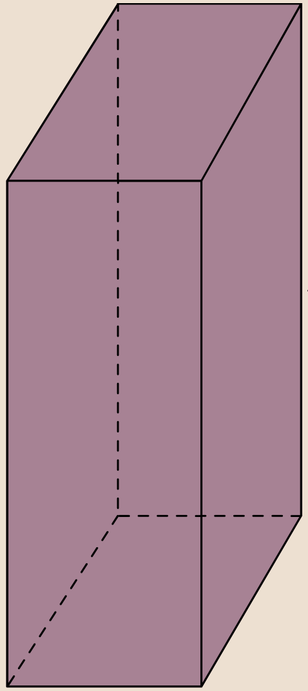
# Padding

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 255 | 255 | 255 | 255 | 255 | 255 | 0.0 |
| 0.0 | 255 | 255 | 255 | 255 | 255 | 255 | 0.0 |
| 0.0 | 255 | 255 | 255 | 255 | 255 | 255 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

# Architecture of SRCNN

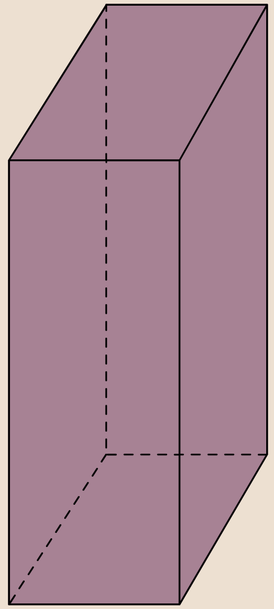
Bilinear Interpolation

Patch Extraction



128 Filters of size 9 x 9

Non Linear Mapping



64 Filters of size 3 x 3

Reconstruction

```python
def model():

    # define model type
    SRCNN = Sequential()
    # SRCNN.add(Input(shape=(None, None, 3)))
    # add model layers
    SRCNN.add(Conv2D(filters=128, kernel_size = (9, 9), kernel_initializer='glorot_uniform',
                     activation='relu', padding='valid', use_bias=True, input_shape=(None, None, 1)))
    SRCNN.add(Conv2D(filters=64, kernel_size = (3, 3), kernel_initializer='glorot_uniform',
                     activation='relu', padding='same', use_bias=True))
    SRCNN.add(Conv2D(filters=1, kernel_size = (5, 5), kernel_initializer='glorot_uniform',
                     activation='linear', padding='valid', use_bias=True))

    # define optimizer
    adam = Adam(learning_rate=0.0003)

    # compile model
    SRCNN.compile(optimizer=adam, loss='mean_squared_error', metrics=['mean_squared_error'])

    return SRCNN
```

- SRCNN is trained on Y channel only

## ***About YCrCb Channel***

- ***Y Channel :*** *Black-and-white information of the image is stored here.*
- ***Cr Channel :*** *Measures how much red is in the pixel compared to brightness.*
- ***Cb Channel :*** *Measures how much blue is in the pixel compared to brightness.*

## ***RGB to YCrCb Conversion Formula***

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = 128 - 0.168736R - 0.331264G + 0.5B$$

$$Cr = 128 + 0.5R - 0.418688G - 0.081312B$$

# Data Preparation

```
def prepare_images(path, factor):

    # loop through the files in the directory
    for file in os.listdir(path):
        try:
          # open the file
          img = cv2.imread(path + '/' + file)

          # find old and new image dimensions
          h, w, _ = img.shape
          new_height = int(h / factor)
          new_width = int(w / factor)

          # resize the image - down
          img = cv2.resize(img, (new_width, new_height), interpolation = cv2.INTER_LINEAR)

          # resize the image - up
          img = cv2.resize(img, (w, h), interpolation = cv2.INTER_LINEAR)

          # save the image
          print('Saving {}'.format(file))
          cv2.imwrite('images/{}'.format(file), img)
        except:
          print('ERROR for file-', file, '!')
          pass

prepare_images('source/', 2)
```

# Layer by Layer Output

***Input :*** *Y channel of the degraded image*



*Outputs of Layer 1 Showing 3 images*



*Outputs of Layer 2 Showing 3 images*

*Third Layer : Reconstruction*

*Converting Back to RGB*

# Evaluation Metric Used

## *Peak Signal to Noise Ratio (PSNR)*

$$PSNR = 10 \cdot \log_{10}(255^2 / MSE)$$

- *Represents the ratio between the maximum possible signal (image) power and the error introduced by the model.*
- *Higher PSNR → better image quality*
- *PSNR > 30 implies Good image*

```python
def psnr(target, ref):

    target_data = target.astype(float)
    ref_data = ref.astype(float)

    diff = ref_data - target_data
    diff = diff.flatten('C')

    rmse = math.sqrt(np.mean(diff ** 2.))

    return 20 * math.log10(255. / rmse)
```

# Results

***Original***

***Bilinear Interpolation***

PSNR : 24.74
MSE : 654

***SRCNN***

PSNR : 26.64
MSE : 420.5

PSNR : 26.87
MSE : 400.15

PSNR : 27.68
MSE : 332.24

# Results

***Original***

***Bilinear Interpolation***

PSNR : 33.01
MSE : 97.32

***SRCNN***

PSNR : 36.67
MSE : 41.98

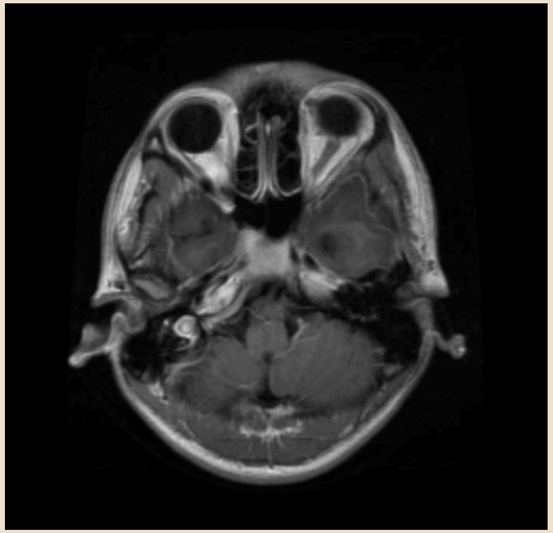
PSNR : 30.93
MSE : 157.46

PSNR : 34.88
MSE : 63.28

# Results

***Original***

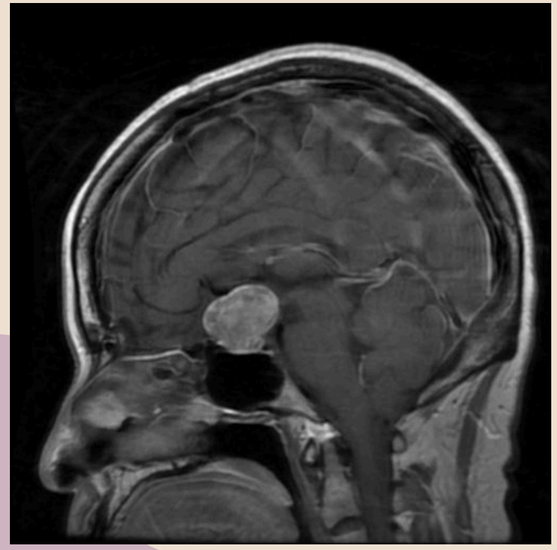
***Bilinear Interpolation***



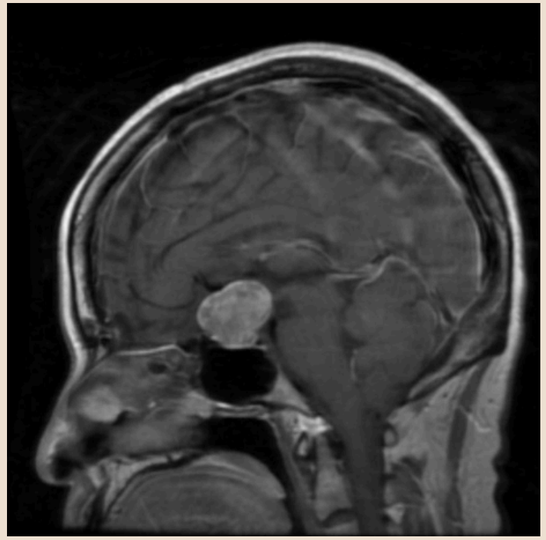
**PSNR : 36.72**
**MSE : 41.47**

***SRCNN***

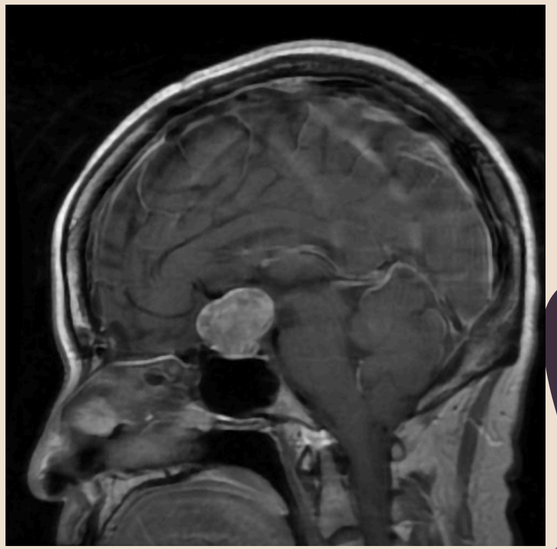
**PSNR : 40.44**
**MSE : 17.61**



**PSNR : 38.88**
**MSE : 25.22**



**PSNR : 42.86**
**MSE : 10.08**



# Conclusion

- *Super-Resolution is crucial for enhancing image details in fields like medical imaging, satellite imaging, and photography.*
- *SRCNN, proposed by Dong et al., is the first CNN-based model for image super-resolution and laid the foundation for many advanced deep learning methods.*
- *SRCNN outperforms traditional interpolation methods, producing sharper and more accurate reconstructions.*

# References


- *Dong, Chao, et al. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38.2 (2015): 295-307..*

- *Super Resolution Benchmark Dataset : https://www.kaggle.com/datasets/jesucristo/super-resolution-benchmarks*

- *Brain Tumor MRI Dataset : https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset*

# THANK YOU