B. Tech Capstone Project Report

On

Single Image Super Resolution Imaging using Deep Learning

Group Members:

1. **Mihir Dalal (2019eeb1171)**
2. **Saiyam Bharara (2019eeb1189)**
3. **Somil Jain (2019eeb1202)**

Department of Electrical Engineering

Course Coordinator - Dr. Devarshi Das

Under the Guidance of Dr. Jyotindra Singh Sahambi and Mr. Inderjeet Singla

Indian Institute of Technology, Ropar

Ropar – 140001

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Supervisor **Dr. Jyotindra Singh Sahambi** as well as our institution **IIT Ropar** for giving us the golden opportunity to work on this project on Single Image Super resolution imaging using deep learning for the course CP302-Capstone project. This project motivated us to do research on Deep Learning and also learnt a lot about different and interesting topics that my fellow classmates took up for this project.
I would also like to express our heartfelt gratitude to **Mr. Inderjeet Singla** for guiding us whenever we faced any difficulty in the project throughout the semester and for their immense support throughout the semester and making this project possible.

Date: 29 November 2022

Group members :
Mihir Dalal (2019EEB1171)
Somil Jain (2019EEB1202)
Saiyam Bharara (2019EEB1189)

Abbreviations

SISR - Single-image super-resolution

PRLSR - Progressive Residual Learning based Super Resolution model

PRB - Progressive Residual Block

HR - High Resolution

LR - Low Resolution

HFP - High Frequency Preserving module

RLA - Residual Learning Architecture

PRB - Progressive Residual Block

PSNR - Peak Signal to Noise Ratio

SSIM - Structural Similarity Index

ResNet - Residual Neural Network

ANN - Artificial Neural Network

DL - Deep Learning

Content

INTRODUCTION

Technology has advanced significantly worldwide in the past ten years, both in terms of hardware and software. Industries used cutting-edge technology to make electronic devices like computers, mobile phones, PDAs, and many more at competitive costs. The camera sensor production facilities improved their manufacturing processes to create high-resolution (HR) digital cameras of excellent quality. Despite the availability of HR digital cameras, many computer vision applications, including target identification, medical imaging, and many others, still had a strong demand for higher resolution pictures, which frequently exceeded the capability of these HR digital cameras. These applications looked to image-processing techniques for a way to provide high-resolution imagery that would meet the enormous demand for it.

A potential method of digital imaging called super-resolution image reconstruction seeks to recreate high-resolution (HR) imagery by combining the fragments of data from several undersampled low-resolution (LR) photographs of the same scene. In order to remove distortions like noise and blur, super-resolution image reconstruction requires upsampling of undersampled images. In comparison 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.

Fig 1: Figure illustrating the difference between a Low and a High-resolution image.

Applications of Super Resolution Imaging

Some applications of super resolution images are:

(i) **Medical Image Processing:** Medical images (like MRI, CT, PET, etc) are extremely valuable for the diagnosis of a disease. Sometimes these scans don't help due to resolution limitations. SR-techniques have been used to generate better images.

(ii) **Video Enhancement and Optical Image Stabilization:** Super-resolution techniques have been used on low-resolution images in videos (Standard definition — SDTV) to create close to High Definition Videos (HDTV) to serve HD screens. Super-resolution can also be used to reduce blurring caused due to motion associated with the camera. To resolve such issues, SR-techniques are hoped to be employed in phones and tablets soon.

(iii) **Surveillance:** Surveillance cameras have become extremely common for traffic and security supervision. Needless to say, it's impossible to install high-resolution cameras everywhere. Since videos contain more data (requiring more computational power) than images and are subject to camera motion artifacts, super-resolution in this domain gets even more complex. Most of the video super-resolution approaches are still in the research stage to be implemented in real life.

(iv) **Astronomical Observation:** Astronomical devices' resolution is limited by its hardware. In such cases, Super-resolution is a boon to researchers exploring and studying outer space.

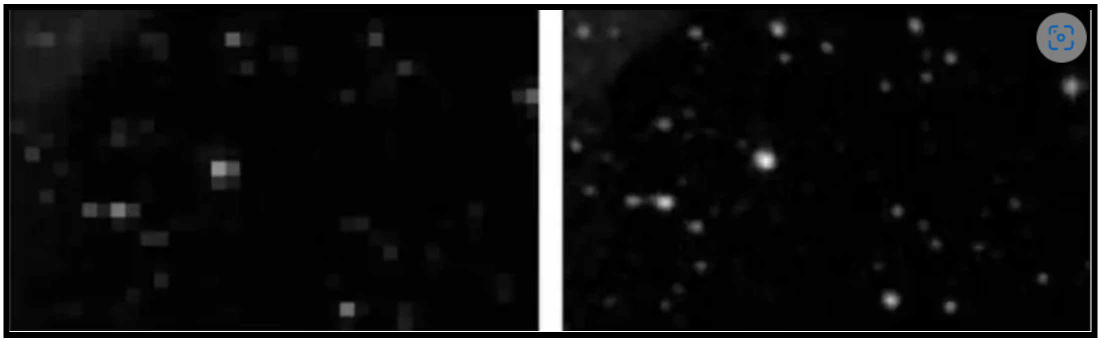


Fig 2: Left: LR astronomical image, Right: SR astronomical image.

Theoretical and Conceptual Aspects

The entire amount of pixels in an image, both horizontally and vertically, is essentially what is meant by spatial resolution. A digital image with dimensions of 300 pixels wide by 300 pixels high, for instance, has 90,000 total pixels and is therefore 0.1 megapixel (MP) in size. The image will have the following dimensions if it is tripled: 900 pixels wide by 900 pixels high, for a total of 810,000 pixels, or 0.8 MP. It is obvious that an image's ability to carry out detail depends directly on the amount of pixels it contains. The more pixels an image has, the more detail it can express. The acquired image typically provides an inadequate representation of the scene. An observed image merely depicts a degraded representation of the actual scene since real imaging technologies and imaging conditions are imperfect. These image degradations are brought on by a number of things, including aliasing, noise, and blur.

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 image provides more information about the original situation. High resolution is frequently required in computer vision applications to improve pattern recognition and image analysis performance.

Network Architecture and Methodology

Deep convolutional neural networks (CNNs) have recently made outstanding advancements in single picture super-resolution (SISR). To attain good performance, many of these techniques, however, utilize very deep or wide convolutional layers. However, the majority of current research is on using far deeper super-resolution networks, which are unfriendly to the limited computing power. This study suggests employing a lightweight network called PRLSR—progressive residual learning for SISR—to address this problem.

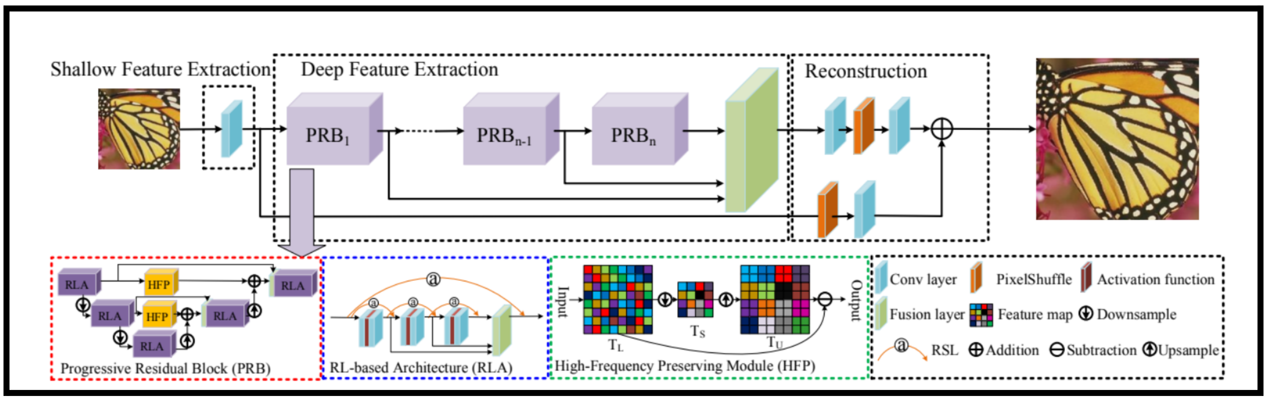


Fig 3: The overall architecture of PRLSR, including shallow feature extraction, deep feature extraction and reconstruction.

A color image in low resolution serves as the model's input, and its output is an improved high resolution image of double its size. The three modules that make up the overall network architecture are as follows.

1. PRB module (Progressive Residual Block): The computational cost of earlier traditional SR models is substantially linear with the number of layers since the spatial resolution of the feature map is often constant in the pipeline in case of losing picture information. To eliminate feature redundancy in the proposed PRB, we gradually exploit the downsampling and upsampling layers, enabling PRLSR to reach very deep layers for excellent performance at a cheap computational cost.
2. HFP module (High Frequency Preserve) : The primary goal of SISR is to amplify resolution while preserving as many details as possible. Therefore, the earlier CNN-based SR algorithms frequently maintain the feature map's spatial resolution throughout the pipeline and upsample it at the conclusion. Our PRB reduces the size of the feature map

twice in order to lessen the computational cost, but this may result in the compressed feature losing too much HFI and the restored SR pictures seeming artificial. The HFI is retained by the proposed high frequency preserving module (HFP) before the size of feature maps is reduced.

3. RLA module (Residual Learning architecture) : It has been established that the model's depth is crucial to the SISR work. As in the case of the standard ResNet model, we can clearly see that by employing skip connections from one layer to the next layer, the RL-based architecture can significantly reduce the gradient vanishing problem as the depth grows and improve the model's ability to represent data. We will use this technique in our model as well for the fundamental feature extraction module in PRB for deepening the model.
4. Fusion layer : All the inputs to the Fusion layer represent some useful information of the low resolution image. Hence, to retain all the useful frequency info, this layer first concatenates all the inputs along the channel axis and then takes a convolution with a 3 x 3 kernel with learnable weights with 32 channels in the convolution layer.
5. Deep feature extraction block : This module consists of several PRB modules to effectively obtain the image information with less depth and easy computation.

Basically, Shallow feature extraction (the first layer), deep feature extraction, and reconstruction make up our PRLSR's three components.
A convolution layer with a kernel size of 3 x 3 is used for the shallow feature extraction portion to obtain the shallow feature x0, which is determined by:

$$x_0 = f_s(I_{LR}),$$

Where where f_s denotes the shallow feature extraction function for the LR image I_{LR}.
The deep feature extraction module consists of several PRBs and a fusion layer, which can be formulated as:

$$x_n = \zeta^n(\zeta^{n-1}(...(\zeta^1(x_0)))), n > 0,$$
$$\widehat{x} = \Theta([x_1, x_2, ..., x_n]),$$

Where ζ^n denotes the feature extraction of the n-th PRB, Θ is the fusion function (used in fusion layer). x_n is the output of the n-th PRB, and $\widehat{x}$ is the output of deep feature extraction module and [·] denotes the concatenation. The output $\widehat{x}$ is then fed into the reconstruction module. Meanwhile, a global residual path fup is designed by stacking a PixelShuffle layer and a convolution layer with x0 as input. The final restored SR image ISR can be obtained by,

$$I_{SR} = f_{rec}(\widehat{x}) + f_{up}(x_0),$$

Where f_{rec} is the reconstruction function, composed of two convolution layers and a PixelShuffle layer.

Five RLAs, two HFPs, and two fusion layers make up the U-Net structure of the PRB. These RLAs gradually pick up the SR knowledge from the LR characteristics. The first two RLAs use an average-pooling layer to reduce the spatial resolution of the feature map by half twice, and the latter two RLAs use bilinear interpolation to raise it twice. HFP is made to keep the high-frequency information (HFI) before resolution reduction, preventing the loss of HFI brought on by downsampling.

Data Information and Pre Processing, losses and implementation details

The most crucial aspect of training a high performance model is to obtain an adequately large to cater to the computation needs. For the training, testing and validation purposes, we have used the commonly available DIV2K tensorflow dataset comprising 800 low and high resolution training images and 100 low and high resolution testing images.

We have used mini batch processing of batch size 2 in order to feed the low resolution images to the model. For the settings of hyperparameters, we've set 32 channels for each convolution layer and ReLU is used as the activation function in the Residual learning based architecture block.

We have implemented the model as a concatenation of 3 sections:

- Shallow feature extraction layer
- Deep feature extraction layer
- Reconstruction layer

Shallow feature extraction is just a simple Convolution layer to extract low level features. Deep feature extraction contains 3 PRBs (Progressive residual block) the results of which are merged with the help of a fusion layer. The aforementioned and implemented model has the capability to double scale the input LR image.

Loss function

Our model uses the L1 loss function for the training procedure. L1 Loss Function is used to minimize the error which is the sum of all the absolute differences between the true value and the predicted value.

$$L1LossFunction = \sum_{i=1}^{n} |y_{true} - y_{predicted}|$$

L1 loss function is preferred when there are outliers present in the dataset.

Metrics

In order to judge the performance of the model we have used SSIM and PSNR values

- SSIM: The structural similarity index measure (SSIM) is a method used for measuring the similarity between two images. The Structural Similarity Index metric extracts 3 key features from an image Luminance, Contrast and Structure

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

  μx: the pixel sample mean of x

  μy: the pixel sample mean of y

  σ_x^2 the variance of x

  σ_y^2 the variance of y

  σ_{xy} the covariance of x and y

- PSNR: The term peak signal-to-noise ratio is an expression for the ratio between the maximum possible value of a signal and the power of distorting noise that affects the quality of its representation.

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

Hyperparameter Tuning

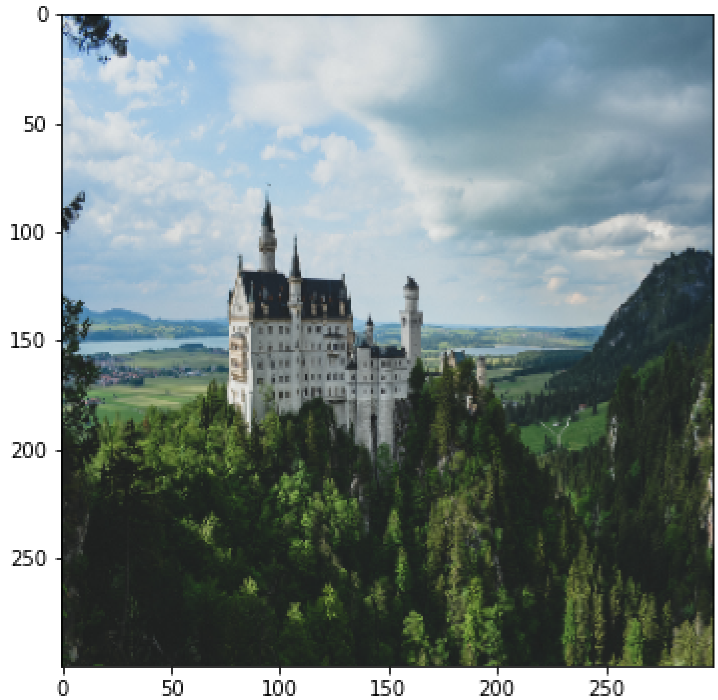
We have used Adam optimizer to minimize the loss function with a learning rate 2e -4 for execution over 20 epochs. Our model comprises 1,406,678 trainable parameters.

Results and observations

We trained our models on Kaggle due to memory restrictions on our systems, which do not have GPUs (Graphics Processing Units) with CUDA support. However, we could only run our model continuously for around an 2.5 hour on GPU enabled kaggle notebooks. Our model ran successfully for 20 epochs with a batch size of 2 and 400 steps per epoch. The results of the model are shown below as a table.

Epoch	Training data L1 Loss	Training data PSNR	Training data SSIM	Testing data L1 loss	Testing data PSNR	Testing data SSIM
0	59.52411	14.1586	0.518967	24.99768	17.46581	0.670349
1	23.67502	18.06219	0.72756	22.21357	18.66451	0.756898
2	21.61755	18.89391	0.788998	24.17287	18.37399	0.805913
3	18.84573	19.98146	0.82917	17.49654	20.4547	0.840977
4	16.82943	20.80355	0.856239	18.04733	20.50474	0.86317
5	16.04855	21.23111	0.87619	14.70398	21.73606	0.886755
6	14.35767	21.97133	0.893588	13.8707	22.16459	0.901632
7	13.89341	22.29336	0.90663	13.13067	22.65737	0.916026
8	13.09132	22.74174	0.91954	12.12764	23.1909	0.926561
9	12.56177	23.09135	0.928481	11.31281	23.62389	0.935575
10	11.69884	23.56549	0.936985	11.19709	23.84703	0.941536
11	11.23364	23.85165	0.942981	10.99534	23.99973	0.947245
12	11.13764	23.99454	0.947207	11.19288	24.03049	0.95012

13	10.90631	24.1484	0.951272	10.09067	24.55332	0.955393
14	10.45936	24.40941	0.954857	10.3387	24.5247	0.957559
15	10.20837	24.56417	0.957777	9.971089	24.72161	0.96063
16	9.821872	24.78517	0.960081	9.304036	25.03495	0.963384
17	9.628832	24.90791	0.961864	9.395897	25.00731	0.964537
18	9.578769	24.9416	0.963521	9.586499	24.98149	0.965393
19	9.384283	25.04744	0.965017	8.832113	25.32041	0.967608



Low resolution image 300 x 300 px

Original high resolution image 600 x 600 px

<u>Predicted high resolution image 600 x 600 px</u>

Conclusions

These are the primary contributions of this work:

1. In contrast to state-of-the-art techniques, we offer a lightweight SISR model dubbed PRL-SR that penetrates very deep layers and delivers higher SR reconstruction performance at a reduced computational cost.
2. To implement PRLSR, PRB is suggested to use downsampling gradually in many stages to decrease the Mult-Adds. Additionally, HFP is built to keep the HFI before PRB resolution decreases. RLA with learnable weights is designed to more efficiently extract multilayer characteristics.

In this study, we present PRLSR, a quick and precise deep network for SISR. A progressive residual block (PRB), which is used in PRLSR, is intended to lessen feature redundancy in gradually utilizing downsampling and upsampling procedures to create multiple stages. We suggest a high-frequency scheme to alleviate the detail loss brought on by the resolution reduction in PRB. To keep the high-frequency information, use the high-frequency preservation module (HFP). In order to understand the SR information added for LR features, multilevel features are extracted using an RL-based architecture (RLA) in each step of PRB. Numerous tests demonstrate that our technique, while preserving great efficiency, obtains state-of-the-art results on four benchmarks.

References

- H. Liu, Z. Lu, W. Shi and J. Tu, "A Fast and Accurate Super-Resolution Network Using Progressive Residual Learning," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 1818-1822, doi: 10.1109/ICASSP40776.2020.9053890.
- M. Zou, J. Tang and G. Wu, "Low Complexity Single Image Super-Resolution with Channel Splitting and Fusion Network," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2398-2402, doi: 10.1109/ICASSP40776.2020.9053516.
- Capel, D.; Zisserman, A.; "Computer vision applied to super resolution," Signal Processing Magazine, IEEE , vol.20, no.3, pp. 75- 86, May 2003 doi:10.1109/MSP.2003.1203211
- Glasner, D.; Bagon, S.; Irani, M.; “Super-Resolution from a Single Image”, 12th International Conference on Computer Vision (ICCV), IEEE , pp. 349 – 356, 2009

****