NNFL Design Project

Shristi Kumari 2016B4A70574P
Akriti Garg 2016B4A70480P

Research Paper

The title of our research paper is :

Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network

By : Jin Yamanaka, Shigesumi Kuwashima, Takio Kurita

Problem Statement

- A highly efficient and faster Single Image Super-Resolution (SISR) model with Deep Convolutional neural networks (Deep CNN) is proposed in the paper which achieves state-of-the-art performance at 10 times lower computation cost

- Single Image Super-Resolution (SISR) is used in many fields like security video surveillance and medical imaging, video playing, websites display.

Importance of the Problem Statement

- The current Single Image Super Resolution models involve large computations and are not suitable for network edge devices like mobile, tablet and IoT devices.

- DCSCN model provides state-of-the-art performance with 10 times lower computation cost using parallelized 1x1 CNNs which not only reduces the dimensions of the previous layer for faster computation with less information loss, but also adds more nonlinearity to enhance the potential representation of the network

Data Description

- For training purpose, publicly available datasets are taken and the distribution is 91 images from Yang and 200 images from the Berkeley Segmentation Dataset
- Adjacent image is a sample from the dataset which is used for training and testing

Data Description

- Data augmentation is performed to obtain different sets of data like SET 5, 6 etc.
- In the training phase, SET 5 dataset is used to evaluate performance and check if the model will overfit or not
- Each training image is split into 32 by 32 patches with stride 16 and 64 patches are used as a mini-batch.
- For testing, SET 5 , SET 14 , and BSDS100 have been used .

Data Processing

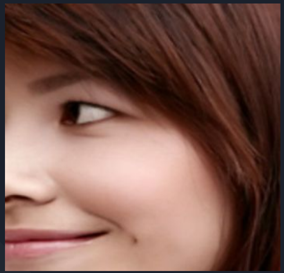
- Data Processing has been done to increase the dataset by 3 times for better results and to not overfit the model.
- Data is Augmented using 3 techniques: Horizontal Flip, Zoom and Rotate by 30 degrees.
- The next slide shows a few sample augmented images produced from a sample input image

Augmented Images

These images are obtained after augmenting the given sample image using flip, zoom and rotate technique.

Original Image

Augmentation



Flipped Image

Zoomed Image

Rotated Image

Model

The DCSCN model consists of the following 2 parts :

- Feature Extraction Network
- Image Detail Reconstruction Network

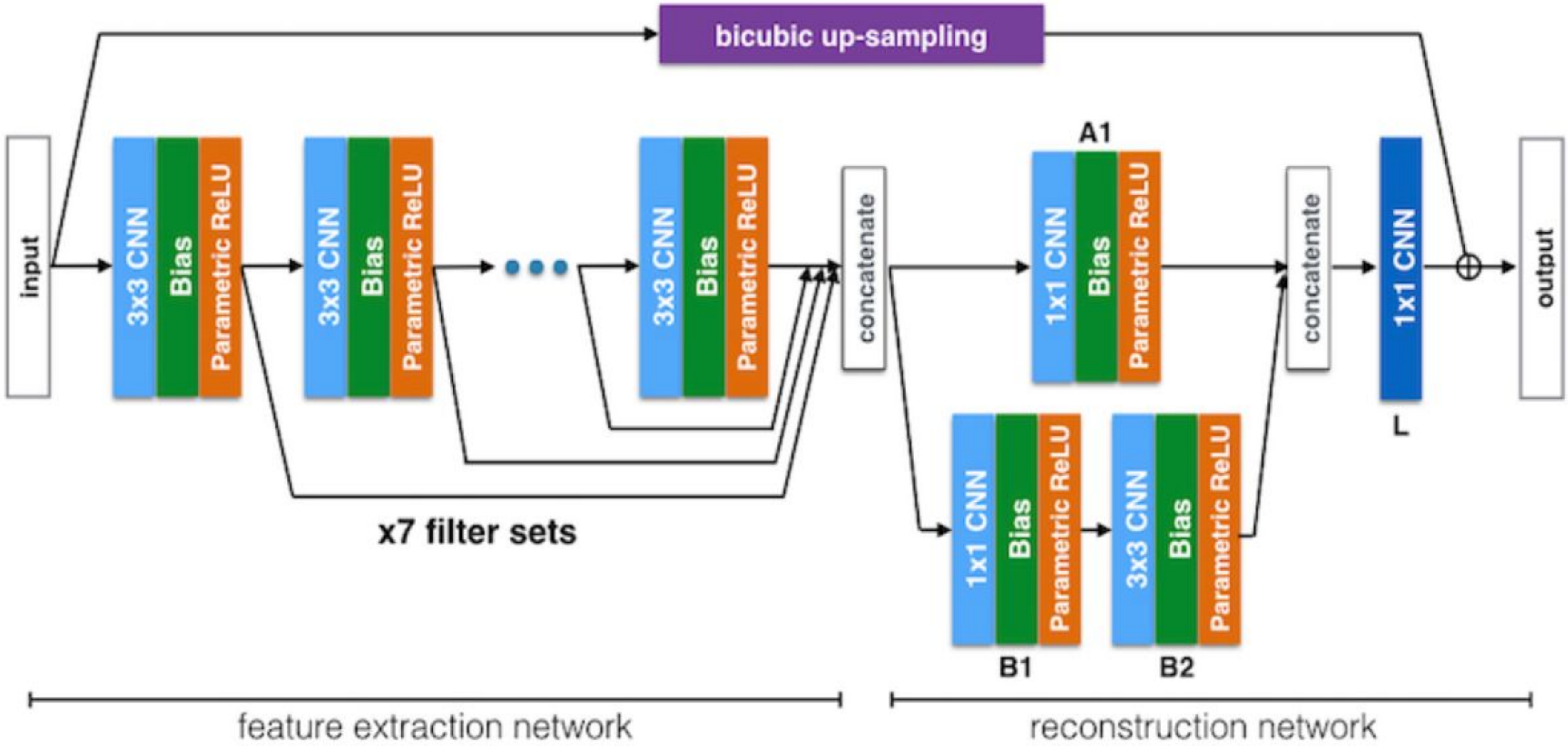
The following slides explain each part separately along with pictorial representation of the structure.

Feature Extraction Network

- In this part 7 sets of 3x3 CNN, bias and Parametric ReLU units are cascaded.
- Each output of the units is passed to the next unit and simultaneously skipped to the reconstruction network.
- Parametric ReLu is used to solve “Dying ReLu “ problem as it prevents weights from learning a large negative bias term and leads to better performance.

Image Reconstruction Network

- The Image reconstruction network consists of 2 parallelized CNN blocks which are concatenated together
- The first consists of 1x1 convolution layer with PRelu and the second consists of a 1x1 layer followed by a 3x3 layer with PRelu as Activation function. After this a 1x1 CNN layer is added
- 1x1 CNNs are used to reduce the input dimension before generating the high resolution pixels



DCSCN Model Structure

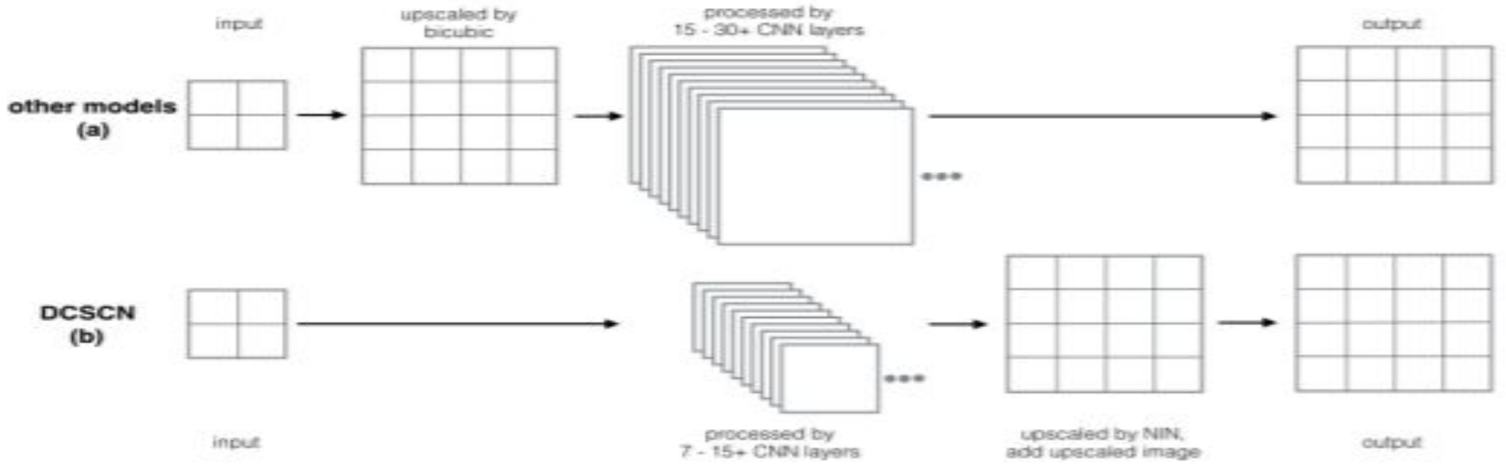


Fig. 2. Simplified process structures of (a) other models and (b) our model (DCSCN).

Table 1. The numbers of filters of each CNN layer of our proposed model

	Feature extraction network							Reconstruction network			
	1	2	3	4	5	6	7	A1	B1	B2	L
DCSCN	96	76	65	55	47	39	32	64	32	32	4
c-DCSCN	32	26	22	18	14	11	8	24	8	8	4

Figure shows simplified model structure. The table states the number of filters in different layers

Results

The following section describes the results obtained after training the model on the described dataset in terms of improvement in computational complexity and accuracy (measured in terms of PSNR and SSIM values) as compared to other models.

Result of PSNR

Peak Signal-to-Noise Ratio (PSNR) used to compare the accuracy of the proposed DCSCN (3 variants : normal, large and compact) with other Deep Learning-based SR algorithms

DataSet	Bicubic	SRCN	SelfEx	DRCN	VDSR	DCSCN (compact)	DCSCN (normal)	DCSCN (large)
Set5 x2	33.66	36.66	36.49	37.63	37.53	37.13	37.62	37.72
Set14 x2	30.24	32.42	32.22	33.04	33.03	32.71	33.05	33.15
BSD100 x2	29.56	31.36	31.18	31.85	31.90	31.59	31.91	32.03

Result of SSIM

SSIM used to compare the accuracy of the proposed DCSCN (3 variants : normal, large and compact) with other Deep Learning-based SR algorithms

DataSet	Bicubic	SRCN	SelfEx	DRCN	VDSR	DCSCN (compact)	DCSCN (normal)	DCSCN (large)
Set5 x2	0.9299	0.9542	0.9537	0.9588	0.9587	0.9569	0.9590	0.9589
Set14 x2	0.8688	0.9063	0.9034	0.9118	0.9124	0.9090	0.9126	0.9133
BSD100 x2	0.8431	0.8879	0.8855	0.8942	0.8960	0.8905	0.8956	0.8966

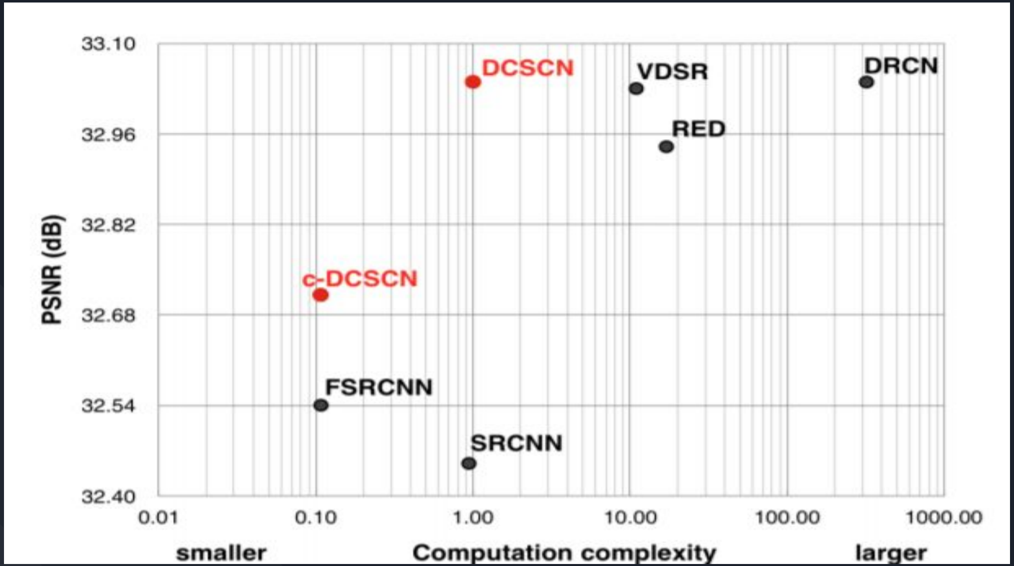
Accuracy Comparison

The figure shows DCSCN algorithm has either best (red text) or second best (blue text) in terms of PSNR and SSIM when compared with other models.

Dataset	SRCNN	DRCN	VDSR	RED30	DCSCN (ours)	c-DCSCN (ours)
Set5	36.66/0.9542	**37.63**/0.9588	37.53/0.9587	**37.66/0.9599**	37.62/**0.9590**	37.62/0.9569
Set14	32.45/0.9063	**33.04**/0.9118	33.03/0.9124	32.94/**0.9144**	**33.05/0.9126**	33.05/0.9090
BSD100	31.36/0.8879	31.85/0.8942	31.90/0.8960	**31.99/0.8974**	**31.91/0.8956**	31.91/0.8905

Computational Complexity Comparison

The adjacent figure gives an estimate of the improvement the model provides as compared to other available techniques in terms of computational complexity.



Qualitative Result

A sample output of our model on a test image is shown next. The figure shows the output generated by the model when given the first image as an input. As can be seen there has been significant improvement in the resolution and the result is comparable to the ground truth data.

Qualitative Result

input LR　　result of our model　　ground truth

Conclusion & Takeaways

NNFL Design project has been a great learning experience for both of us. Over the past 2 months we had an exponential learning curve, and learnt a lot by implementing the theoretical concepts grasped from the course. This project being our first project involving Neural Networks and Python, the amount of learning has been tremendous and we are really thankful to have such a wonderful learning experience.

Conclusion & Takeaways

Here, we are stating down some of our major learnings :

- Gained confidence by implementing a reputed research paper using Python. Practical exposure provided clear understanding and improved our grasp on the subject.
- Made neural network models using both TensorFlow and Keras.
- Work from home enhanced our remote collaboration capabilities. Used GitHub and VS code editor.

Thank You