# SISR Using GAN

Mentor: Prof. Ahlad Kumar

Ruchit Vithani (201701070)
Purvil Mehta (201701073)
Bhargey Mehta (201701074)
Kushal Shah (201701111)

# Timeline

31st July, 2015

Image
Super-Resolution
Using Deep
Convolutional
Networks

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE

# Image Super-Resolution Using Deep Convolutional Networks

- The method is represented as a deep convolutional neural network (CNN) that takes the low-resolution image as the input and outputs the high-resolution one.
- SISR problem is inherently ill-posed since a multiplicity of solutions exist for any given low-resolution pixel. In other words, it is an underdetermined inverse problem, of which solution is not unique. Such a problem is typically mitigated by constraining the solution space by strong prior information.
- Thus, the sparse-coding-based method  is one of the representative external example-based SR methods which allow us to build the dictionary having prior information.
- Sparse coding was the first method developed to solve this problem.

# Image Super-Resolution Using Deep Convolutional Networks

- In this paper, they showed that the aforementioned pipeline was equivalent to a deep convolutional neural network. Motivated by this fact, they consider a convolutional neural network that directly learns an end-to-end mapping between low- and high-resolution images.
- This method differs fundamentally from existing external example-based approaches. This method does not explicitly learn the dictionaries, or manifolds for modeling the patch space. These are implicitly achieved via hidden layers.
- Benefits of this method:
  - Better Accuracy observed from the result
  - Faster because it is fully feed-forward and does not need to solve any optimization problem on usage.
  - It can also handle larger dataset by adding/defining more deep network.
  - It is also applicable on R,G,B channel parallely.

# Image Super-Resolution Using Deep Convolutional Networks

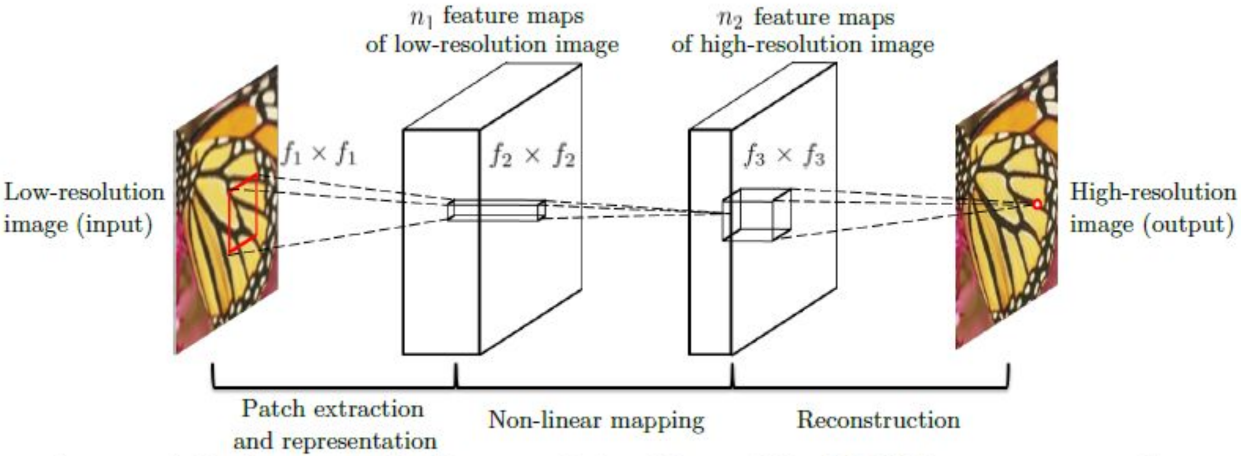



Fig. 2. Given a low-resolution image $\mathbf{Y}$, the first convolutional layer of the SRCNN extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image $F(\mathbf{Y})$.

Source: Image Super-Resolution Using Deep Convolutional Networks [Paper]

# Image Super-Resolution Using Deep Convolutional Networks

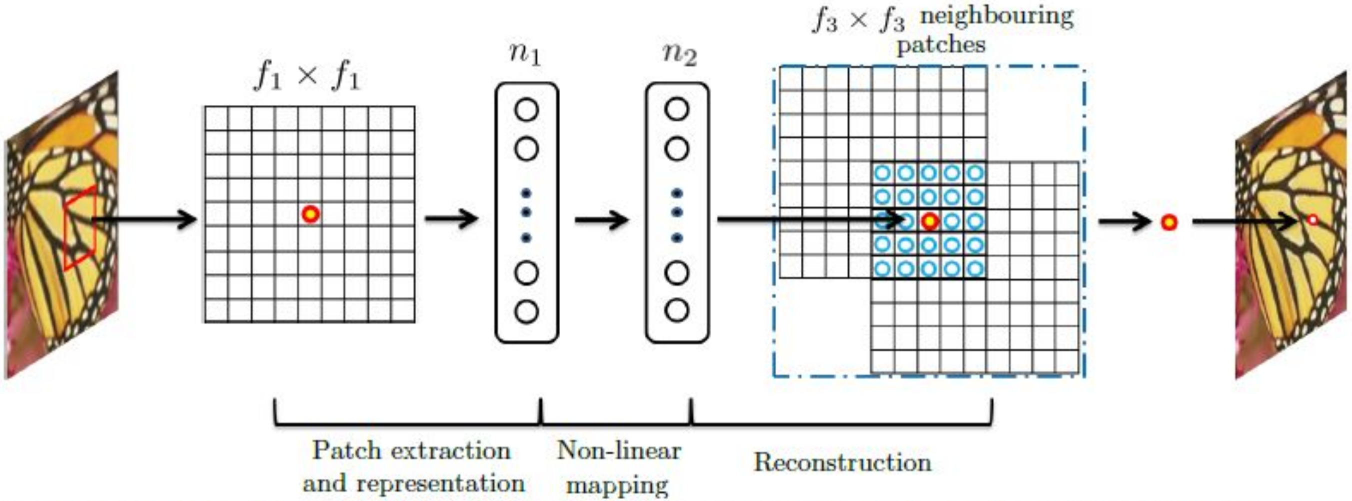



Fig. 3. An illustration of sparse-coding-based methods in the view of a convolutional neural network.

Source: Image Super-Resolution Using Deep Convolutional Networks [Paper]

# Timeline

31st July, 2015

Image Super-Resolution Using Deep Convolutional Networks

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE

15th October, 2015

Deep Networks for Image Super-Resolution with Sparse Prior

Zhaowen Wangyz Ding Liuy Jianchao Yangx Wei Hany Thomas Huangy

# Deep Networks for Image Super-Resolution with Sparse Prior

Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, Thomas Huang

- The authors argue : Domain expertise represented by the conventional sparse coding model is still valuable, and it can be combined with the key ingredients of deep learning to achieve further improved results.

- The authors argue : Domain expertise represented by the conventional sparse coding model is still valuable, and it can be combined with the key ingredients of deep learning to achieve further improved results.
- Using **sparse coding** for Single Image SR :
  - Given a low resolution patch $\boldsymbol{y}$.

- The authors argue : Domain expertise represented by the conventional sparse coding model is still valuable, and it can be combined with the key ingredients of deep learning to achieve further improved results.
- Using **sparse coding** for Single Image SR :
  - Given a low resolution patch $\boldsymbol{y}$.
  - **Step 1** : Obtain sparse encoding of this resolution patch by optimising this equation ⇒ $$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{D}_y \boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \qquad (1)$$

    Where $\boldsymbol{D}_y$ : Overcomplete dictionary trained from low resolution images.

    $\lambda$ : Coefficient of Sparsity

    $\boldsymbol{\alpha}$ : Sparse encoding of patch $\boldsymbol{y}$ obtained after optimisation.

- The authors argue : Domain expertise represented by the conventional sparse coding model is still valuable, and it can be combined with the key ingredients of deep learning to achieve further improved results.
- Using **sparse coding** for Single Image SR :
  - Given a low resolution patch $\boldsymbol{y}$.
  - **Step 1** : Obtain sparse encoding of this low dimensional patch by optimising this equation ⇒ $$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{D}_y \boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \qquad (1)$$

    Where $\boldsymbol{D}_y$ : Overcomplete dictionary trained from low resolution images.

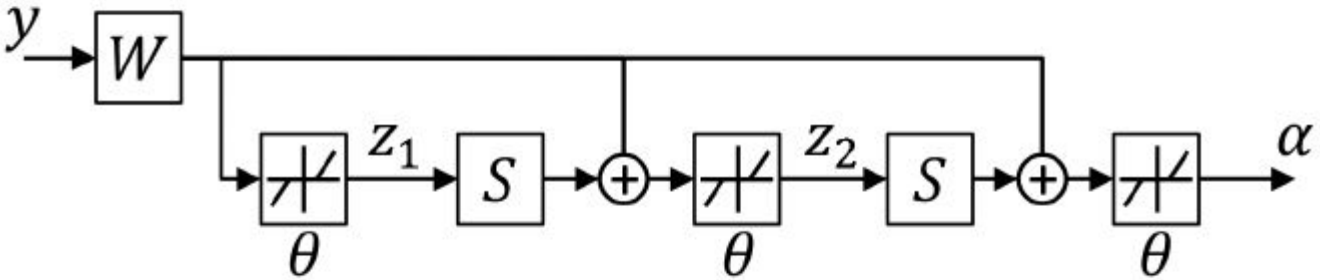
    $\lambda$ : Coefficient of Sparsity

    $\boldsymbol{\alpha}$ : Sparse encoding of patch $\boldsymbol{y}$ obtained after optimisation.
  - **Step 2** : Extract the high resolution patch using high resolution dictionary $\boldsymbol{D}_x$ .
    - This is done as follows : $\boldsymbol{x} = \boldsymbol{D}_x \boldsymbol{\alpha}$ .
      - $\boldsymbol{x}$ is generated high resolution patch.

- In **Learning fast approximations of sparse coding. In ICML, 2010. 1, 2** K. Gregor and Y. LeCun proposed "A feedforward neural network to efficiently approximate the sparse code $\alpha$ of the input signal $y$ as it would be obtained by solving equation (1) in the slides above."
- The algorithm (which used fully connected NN) is called "**Iterative shrinkage and thresholding algorithm (ISTA).**"

- This network has finite number of recurrent stages, each of which updates the intermediate sparse code according to $$\boldsymbol{z}_{k+1} = h_{\boldsymbol{\theta}}(\boldsymbol{W}\boldsymbol{y} + \boldsymbol{S}\boldsymbol{z}_k), \qquad (2)$$

  Where $h_{\boldsymbol{\theta}}$ is element-wise shrinkage function defined as $[h_{\boldsymbol{\theta}}(\boldsymbol{a})]_i = \mathrm{sign}(a_i)(|a_i| - \theta_i)_+$ and $\boldsymbol{W}$, $\boldsymbol{S}$ are the weights of linear layer of neural network.




Zhaowen Wang, Ding Liu, Jianchao Yang Wei Han, Thomas Huang : Deep Networks for Image Super-Resolution with Sparse Prior

# Sparse Coding based Network for Image SR

- Model input :
  - $\boldsymbol{I}_y$ : bicubic-upscaled low resolution image
- Model output :
  - $\boldsymbol{I}_x$ : Full high resolution image

# Sparse Coding based Network for Image SR

- Model input :
  - $\boldsymbol{I}_y$ : bicubic-upscaled low resolution image
- Model output :
  - $\boldsymbol{I}_x$ : Full high resolution image
- The input image $\boldsymbol{I}_y$ first goes through a convolutional layer $\boldsymbol{H}$ which extracts feature for each LR patch. There are $\boldsymbol{m}_y$ filters of spatial size $s_y \times s_y$ in this layer, so that our input patch size is $s_y \times s_y$ and its feature representation $\boldsymbol{y}$ has $\boldsymbol{m}_y$ dimensions.

# Sparse Coding based Network for Image SR

- Model input :
  - $\boldsymbol{I}_y$ : bicubic-upscaled low resolution image
- Model output :
  - $\boldsymbol{I}_x$ : Full high resolution image
- The input image $\boldsymbol{I}_y$ first goes through a convolutional layer $\boldsymbol{H}$ which extracts feature for each LR patch. There are $\boldsymbol{m}_y$ filters of spatial size $s_y \times s_y$ in this layer, so that our input patch size is $s_y \times s_y$ and its feature representation $\boldsymbol{y}$ has $\boldsymbol{m}_y$ dimensions.
- After this, each patch is fed into a LISTA network with finite number of k recurrent stages to produce its sparse code $\alpha$.

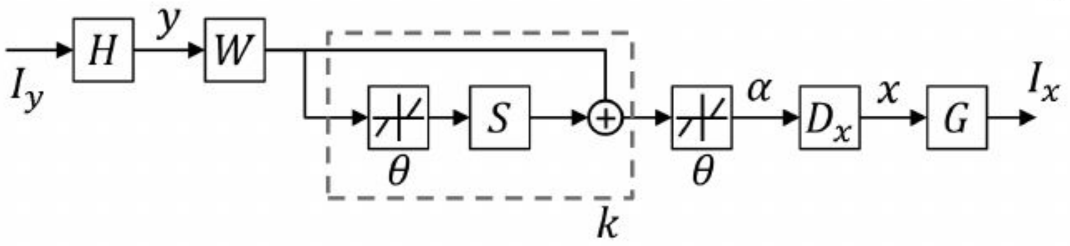
# Sparse Coding based Network for Image SR

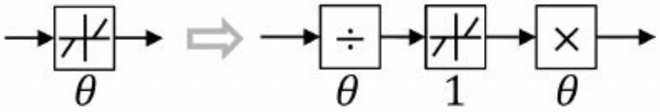
- Model input :
  - $\boldsymbol{I}_y$ : bicubic-upscaled low resolution image
- Model output :
  - $\boldsymbol{I}_x$ : Full high resolution image
- The input image $\boldsymbol{I}_y$ first goes through a convolutional layer $\boldsymbol{H}$ which extracts feature for each LR patch. There are $\boldsymbol{m}_y$ filters of spatial size $s_y \times s_y$ in this layer, so that our input patch size is $s_y \times s_y$ and its feature representation $\boldsymbol{y}$ has $\boldsymbol{m}_y$ dimensions.
- After this, each patch is fet into a LISTA network with finite number of k recurrent stages to produce its sparse code $\alpha$.
- This sparse code is then multiplied with high resolution dictionary $\boldsymbol{D}_x$ to obtain reconstructed high resolution image $\boldsymbol{I}_x$ of size $s_x \times s_x$ .

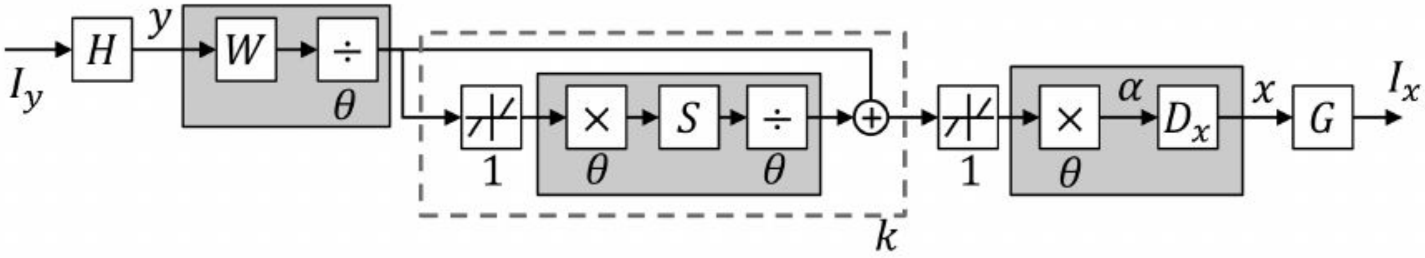
# Sparse Coding based Network for Image SR

$$[h_{\boldsymbol{\theta}}(\boldsymbol{a})]_i = \mathrm{sign}(a_i)(|a_i| - \theta_i)_+$$

$$[h_{\boldsymbol{\theta}}(\boldsymbol{a})]_i = \mathrm{sign}(a_i)\theta_i(|a_i|/\theta_i - 1)_+ = \theta_i h_1(a_i/\theta_i).$$




Zhaowen Wang, Ding Liu, Jianchao Yang Wei Han, Thomas Huang : Deep Networks for Image Super-Resolution with Sparse Prior

# Timeline

31st July, 2015

Image Super-Resolution Using Deep Convolutional Networks

Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE

15th October, 2015

Deep Networks for Image Super-Resolution with Sparse Prior

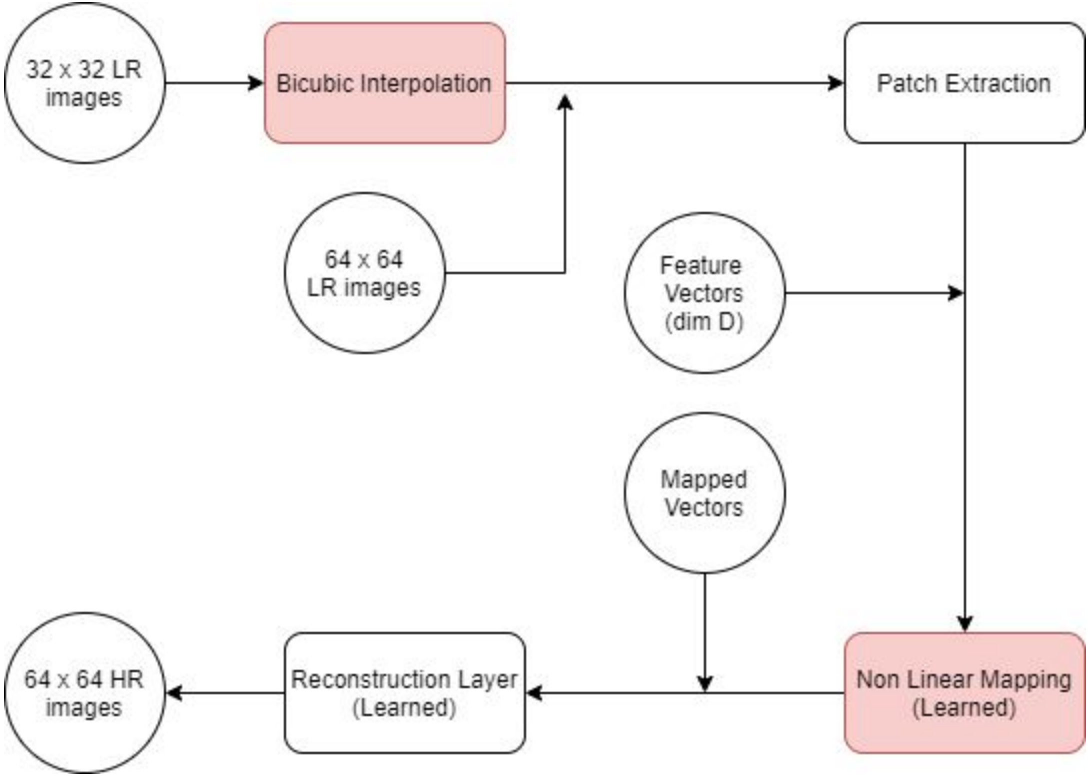
Zhaowen Wangyz Ding Liuy Jianchao Yangx Wei Hany Thomas Huangy

1st August, 2016

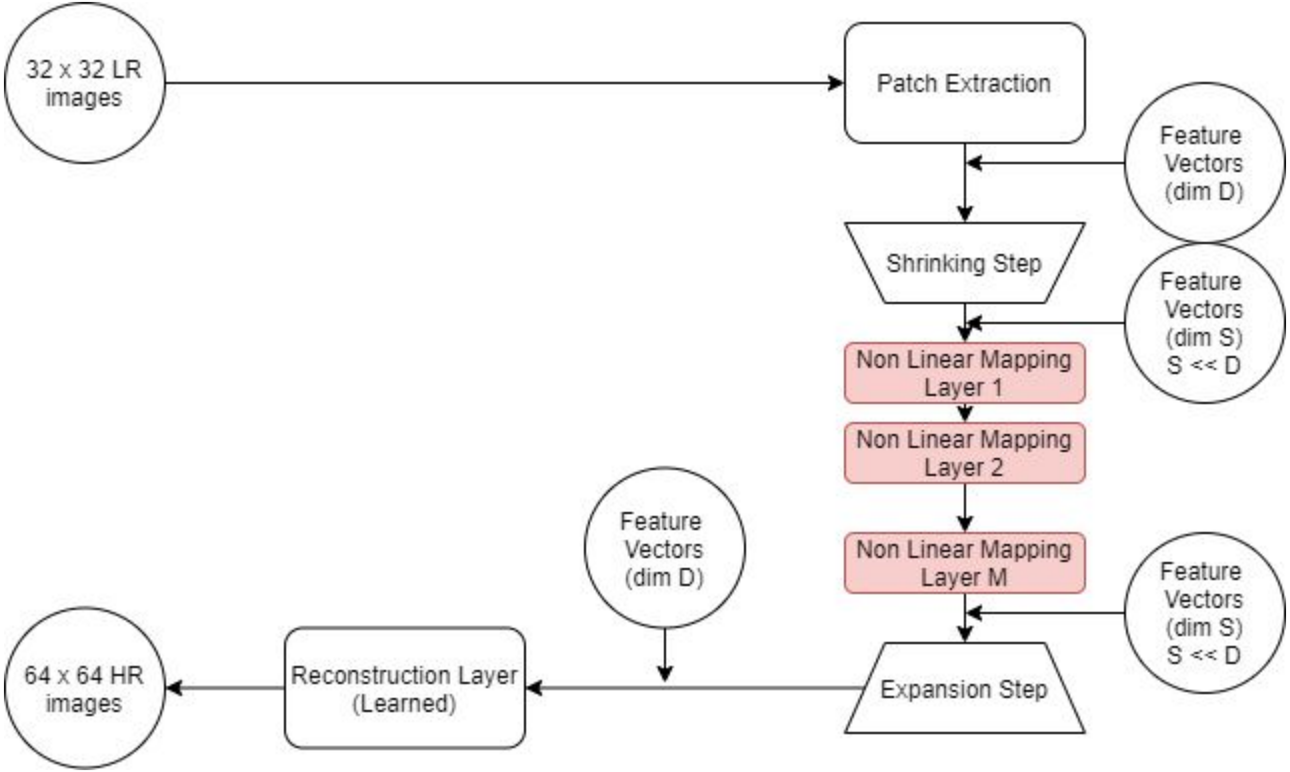
Accelerating the Super-Resolution Convolutional Neural Network

Chao Dong, Chen Change Loy, and Xiaoou Tang

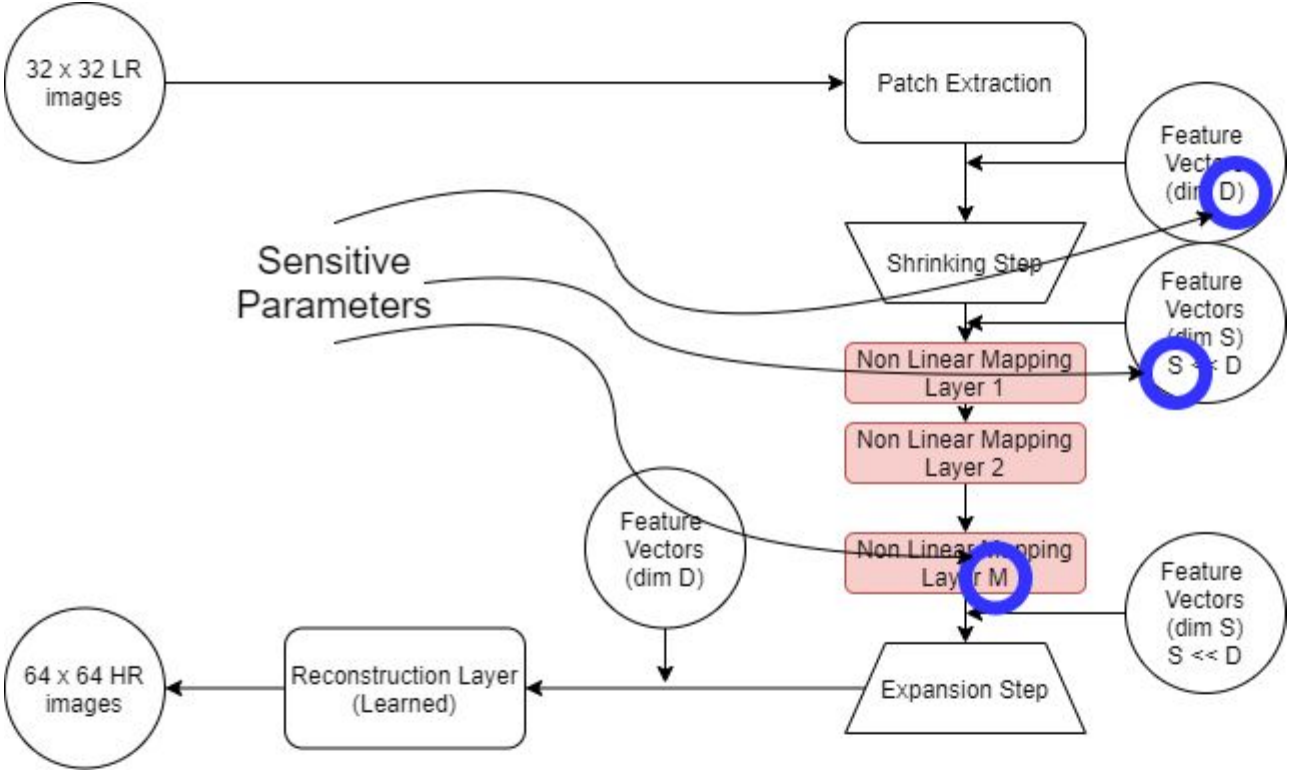
# Accelerating the Super-Resolution Convolutional Neural Network



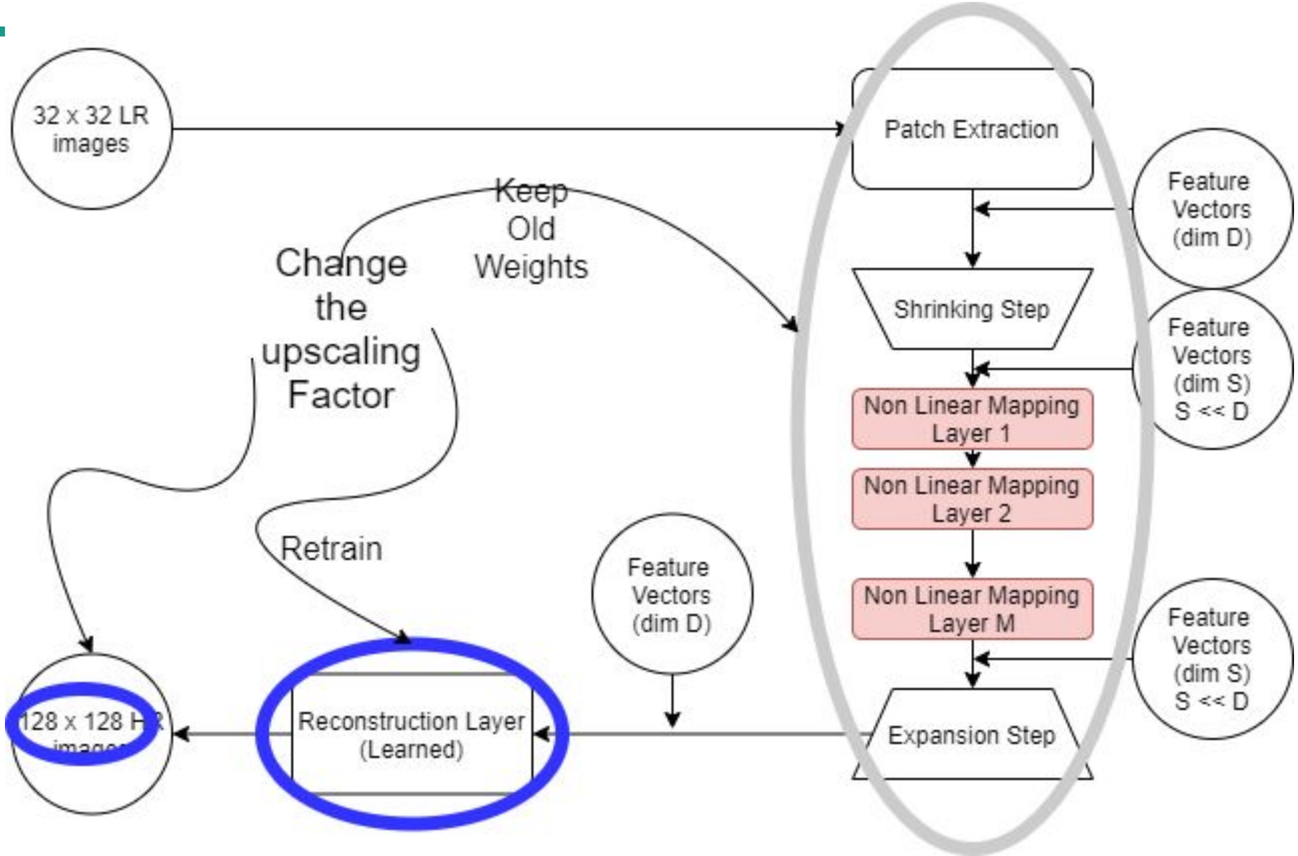
# Accelerating the Super-Resolution Convolutional Neural Network



# Accelerating the Super-Resolution Convolutional Neural Network



# Accelerating the Super-Resolution Convolutional Neural Network



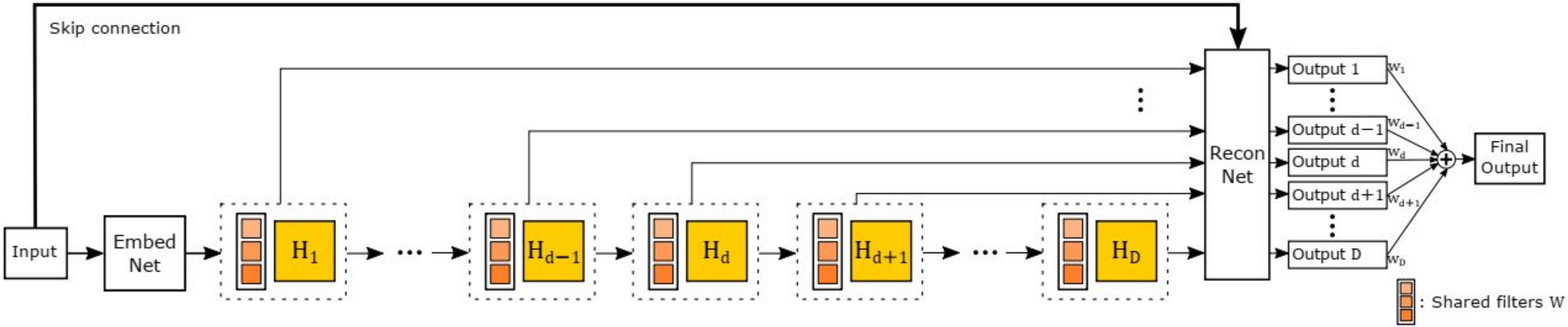
# Timeline

| 31st July, 2015 | 15th October, 2015 | 1st August, 2016 | 11th November, 2016 |
|---|---|---|---|
| Image Super-Resolution Using Deep Convolutional Networks<br>Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE | Deep Networks for Image Super-Resolution with Sparse Prior<br>Zhaowen Wangyz Ding Liuy Jianchao Yangx Wei Hany Thomas Huangy | Accelerating the Super-Resolution Convolutional Neural Network<br>Chao Dong, Chen Change Loy, and Xiaoou Tang | Deeply-Recursive Convolutional Network for Image Super-Resolution<br>Jiwon Kim, Jung Kwon Lee and Kyoung Mu Lee Department of ECE, ASRI, Seoul National University, Korea |

# Deeply-Recursive Convolutional Network

- The approach is very similar to a traditional convolutional neural network. However, traditional CNN has 2 issues namely, overfitting and large model that is difficult to store and retrieve.
- In a DRCN, we use same convolutional layer multiple times.
- DRCN, however, is difficult to train with SGD methods.
- To ease the training, the  paper proposes 2 methods:
  - Recursive Supervision
  - Skip Connections.
- The final loss function is composed of 2 intermediate loss functions and a regression parameter $\beta$.
- Unlike the final paper that proposes perceptual loss, the DRCN uses MSE loss.

# Architecture



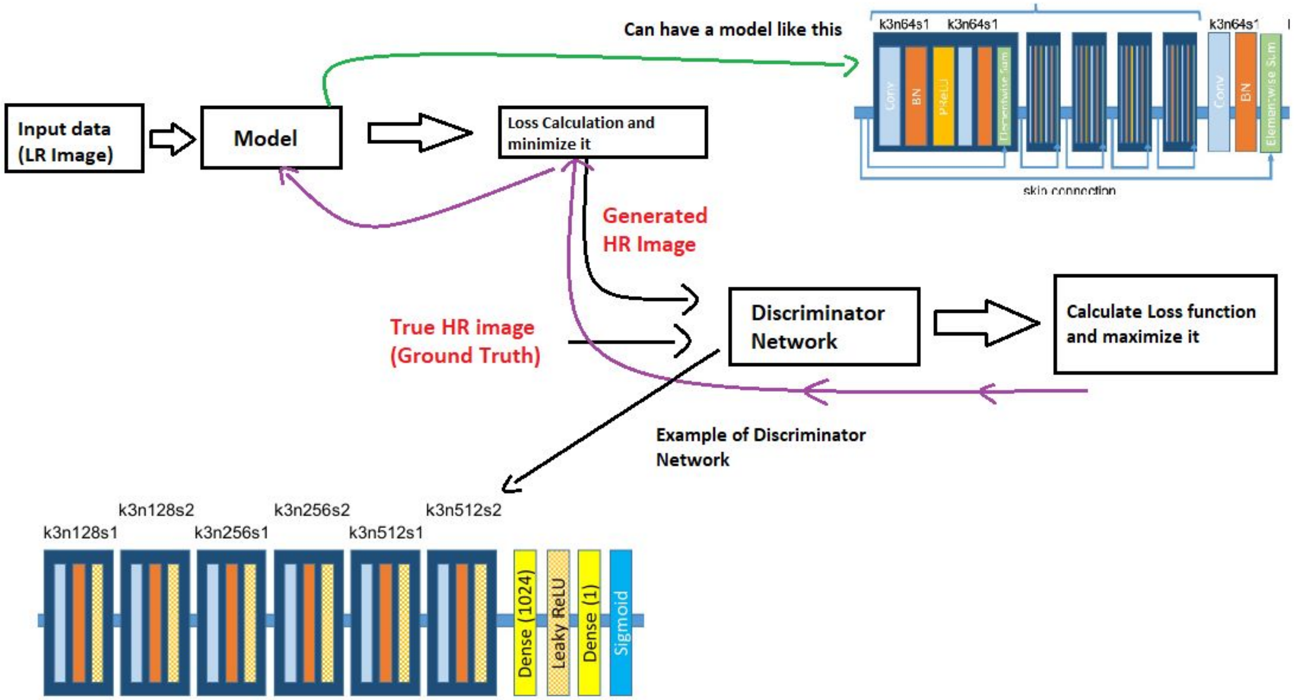

Embedding Network => Inference Network => Reconstruction Network

# Mathematical Model

- Let f1,f2 and f3 represent embedding network, inference network and reconstruction network recursively. The final prediction can be viewed as function composition f(x) = f3(f2(f1(x))).
- The inference layer is applied recursively so, zooming into above model and also applying skip connection, the final model can be viewed as;
- For a given depth of recursion *d*, we can have a prediction $\hat{\mathbf{y}}_d = f_3(\mathbf{x}, g^{(d)}(f_1(\mathbf{x})))$
- These predictions can merged with original image *X*.
- We can take weighted average of these recursions to get the final outcome. Weights are automatically trained.

$$\hat{\mathbf{y}} = \sum_{d=1}^{D} w_d \cdot \hat{\mathbf{y}}_d.$$

# GAN for Super Image Resolution



# Thank you