Super Resolution Using GANs

Image Super Resolution is can be classified into 3 categories :

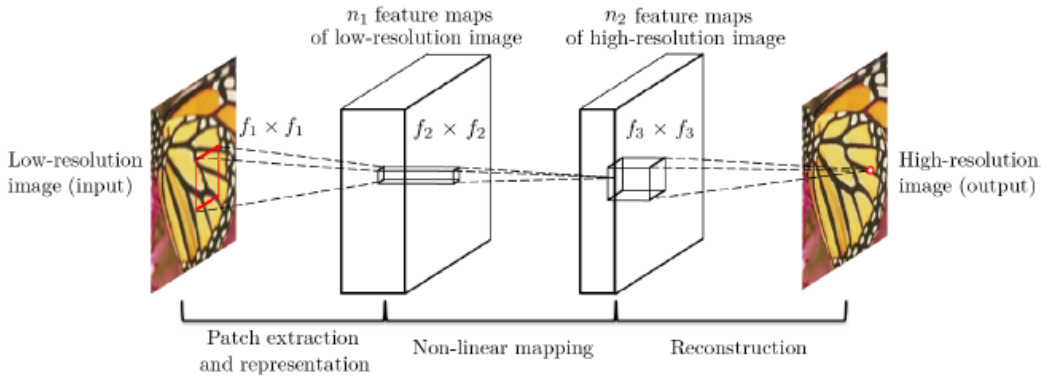
- Interpolation based
- Reconstruction based
- Learning based

Below listed are the models which we are considering for our project :

Super-Resolution Convolution Neural Network

This is a learning based SR model.

Low res image is upscaled to the desired size using bicubic interpolation and is the only pre-processing step for image.



Given a low-resolution image 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(Y).

This consists of three major steps :

- Patch Extraction and representation

Here the patches are extracted and represented by a set of pre-trained bases such as PCA, DCT, Haar, etc. This is equivalent to convolving the image by a set of filters, each of which is a basis. The bases are optimized into the optimization of the network. this operation extracts (overlapping) patches from the low-resolution image Y and represents each patch as a high-dimensional vector. These vectors comprise a set of feature maps, of which the number equals to the dimensionality of the vectors.

- Non-Linear Mapping

This operation nonlinearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch. These vectors comprise another set of feature maps.

- Reconstruction

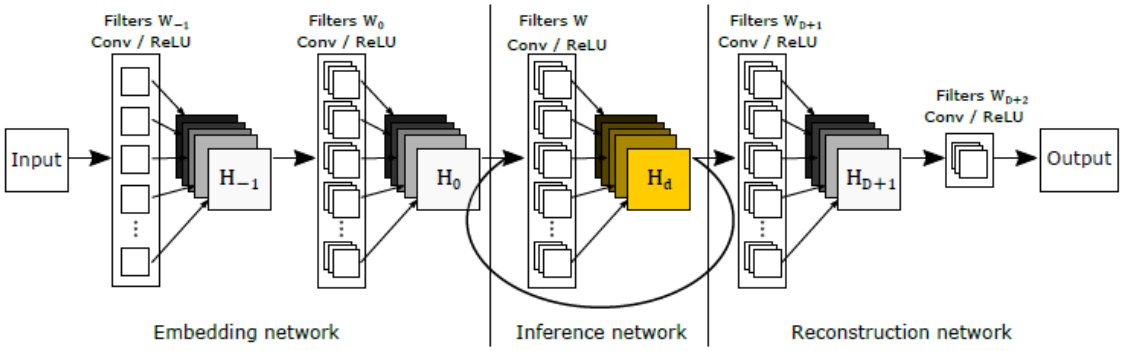
  This operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be similar to the ground truth X

Super Resolution Using GANs

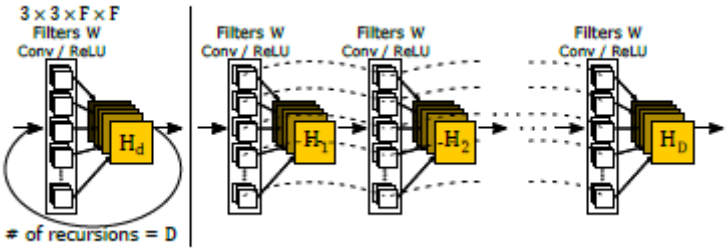
Deeply Recursive Convolution Network

Usually, deeper networks lead to higher accuracies but it also could cause overfitting and a huge model. The Deeply Recursive Convolution Network is a special kind of reconstruction based deep network, which helps us overcome the above stated issues of deep networks by applying the same

convolution network recursively for over 16 times. This way addition of newer parameters with increasing depth and exploding/vanishing gradient is reduced.

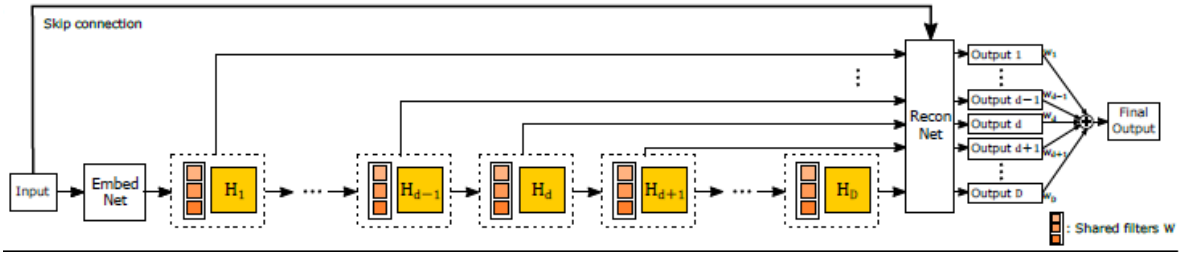


Base Model



Recursive Layer

Embedding Layer takes input images and represents them as feature maps which is then provided as in input to the inference layer. The Inference network is the most important section of Super resolution task assigned to the network overall. Each recursion applies the same convolution followed by a rectified linear unit. With convolution layers wider than 1x1, the receptive field is widened with every recursion. The feature maps from the final recursion represent the HR image. The reconstruction network transforms this HR image back into either gray-scale/rgb (same as the input). The **Inference layer** is the only layer that is recursive while the other networks are more like MLPs with a single hidden layer.



Advanced Model

Super Resolution Using GANs

The **Recursive Supervision method** used in the Advanced model of DRCN uses same recursion net for all convolutions in the inference net, which is used to predict the HR images for all recursions. All outputs from the recursion network are simultaneously supervised, the final prediction(output) is averaged during testing. The optimal weights are learned automatically during training. The adversarial effect of vanishing/exploding gradient along one back propagation path is alleviated.

Skip Connection

For super-resolution, the input and output images are very closely connected and it is necessary to carry most input values atleast until the end of the network. This is however not efficient, additionally because of gradient issues, learning a simple linear relation between input and output images becomes difficult.

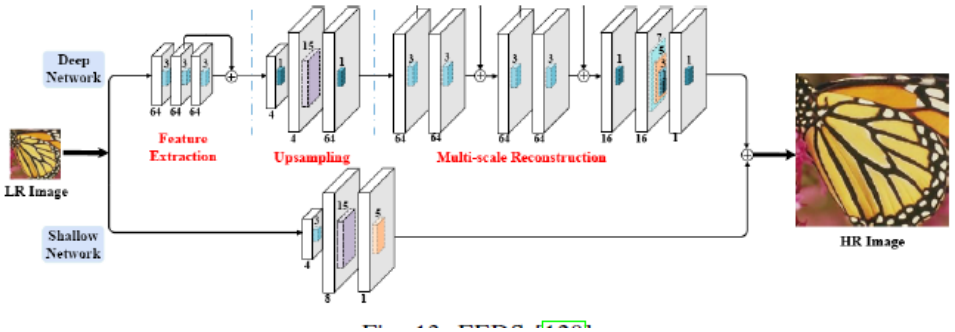
Adding a Skip connection from input to reconstruction layer helps in semantic segmentation network. This way, input image is directly fed to the reconstruction network whenever used during recursions. This benefits by – allowing the network to store the input signal during recursions and this copy of the input signal can be used as a reference during target prediction.

End to End Deep and Shallow Networks

This deep CNN consists of 13 trainable layers which perform the below tasks :

- Feature Extraction
- Up-sampling
- Multi-scale reconstruction

This enables to restore the HR content more accurately but makes the training challenging. The shallow CNN on the contrary, is simpler architecture enabling faster training. This leads to the faster convergence for this network and this acts like a stabilizing agent in the training process of the collective network.



The feature extraction consists of three convolution layers with ReLUs(kernel size 3x3 generate feature maps of 64 channels). A shortcut connection is adapted between the o/p of the first and the o/p of the third layer. This is done as it facilitates gradient flow through multiple layers, helping with faster training. **Upsampling** is where the deconvolution and unpooling takes place to increase spatial span of LR images to target HR size. The Upsampling layer connects the Feature extraction and Multi-scale Reconstruction. A larger deconvolution kernel size enables the up-sampling operation to consider a larger i/p neighborhood and enforce better spatial consistency. But this also increases computing complexity. For simplicity, the kernel sizes are set to 14x14, 15x15 and 16x16 for up-sampling factors of 2,3 and 4 respectively. Zero padding of 6px is conducted on the o/p feature maps

to preserve spatial maps. Two 1x1 convolutions are conducted before and after expensive deconv layer to reduce computational complexity, where the 64 channel input feature map is mapped to the 4 channel output feature map during upsampling and the last convolution where this is mapped back to the 64 channels unsampled feature map. **Multi-scale reconstruction** is the last component of the Deep network. It consists of 7 trainable layers interleaved by ReLU layers. The first 3 layers are 3x3 convolution layers which take input from the 64 channels and generate a new feature map. The 5th layer is a 1x1 layer for dimensionality reduction mapping the feature maps of 64 channels to 16 channels. Subsequent multi-scale convolution layer of kernel sizes 1x1,3x3,5x5,7x7 produce 4 feature maps of 16 channels each. These are then fused together and this concatenated feature map is then fed into another 1x1 convolution layer and serves as a weighted combination of multi-context feature and reconstruct the Final HR Images.

Shallow Network

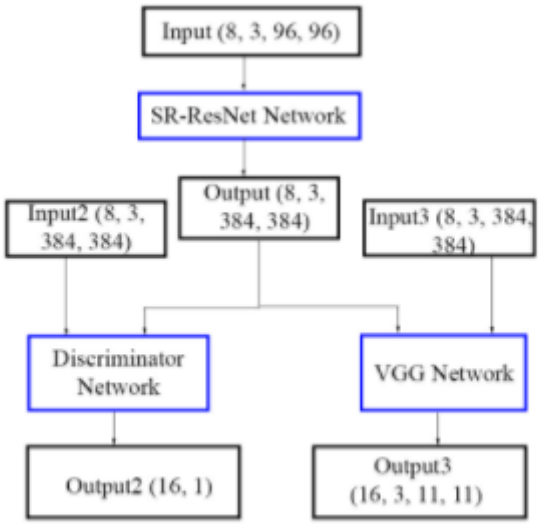
Our shallow network is of 3 layers. 1st layer takes the original LR image as input and conducts 3x3 convolutions, producing a feature map of 4 channels. Second is a deconv layer which upsamples the input feature map to the target spatial size. The final layer is the reconstructed HR image from the upsampled feature maps of 5x5 conv.

In comparison, the shallow CNN can restore the overall illumination but fails to capture high-frequency details. To combine the best of both worlds, we jointly train an ensemble comprising the proposed deep CNN and another shallow CNN. The shallow CNN facilitates faster convergence and predicts the major component of HR images, while the deep CNN restores high-frequency details and corrects errors of the shallow CNN.

Comparative Assessment of the models on Set-14 dataset (x3 upsampling)

Model Name	Accuracy
SRCNN	29.30
EEDS	29.60
DRCNN	29.76

Our model



Super Resolution Using GANs

The SR-GAN model is built in stages. The SR-ResNet model is created and a VGG network is added to it. The weights for VGG are frozen as we will not be updating the weights.
In the pre-train mode:

1. The discriminator model is not attached to the entire network. Therefore it is only the SR + VGG model that will be pretrained first.

2. During pretraining, the VGG perceptual losses will be used to train (using the ContentVGGRegularizer) and TotalVariation loss (using TVRegularizer). No other loss (MSE, Binary crosss entropy, Discriminator) will be applied.

3. Content Regularizer loss will be applied to the VGG Convolution 2-2 layer

4. After pre training the SR + VGG model, we will pretrain the discriminator model.

5. During discriminator pretraining, model is Generaor + Discriminator. Only binary cross entropy loss is used to train the Discriminator network.

In the full train mode:

1. The discriminator model is attached to the entire network. Therefore it creates the SR + GAN + VGG model (SRGAN)
2. Discriminator loss is also added to the VGGContentLoss and TVLoss.
3. Content regularizer loss is applied to the VGG Convolution 5-3 layer. (VGG 16 is used instead of 19 for now)

We have used a function to load datasets in aws. Then we run this network and plot graphs to analyze our results.