A Project Report on Generative Super-Resolution using Capsule Networks

Sujit Rai
2017csm1006@iitrpr.ac.in

Department of Computer Science and Engineering,
Indian Institute of Technology Ropar

Abstract

This paper presents the use of capsule network and the concept of dynamic routing for the generation of high resolution images. The concept of routing by agreement is used in order to learn good part-whole relationships. The entire network is trained in an independent manner in order for the efficient training of each component individually. Also Variational Capsule Encoders and GAN is used in order to overcome the issues associated with the Mean Squared Error.

1 Introduction

The goal of Single-Image super resolution is to generate a high resolution image from a low resolution input image. This problem is of great importance and numerous algorithms are proposed in the recent years. However, it is an underdetermined inverse problem, of which a solution is not Unique [2].

2 Literature Review

Generic Single Image Super Resolution algorithms exploit certain image priors. Depending on this Image priors the Single Image Super-Resolution algorithms can be categorized into several types.

2.1 Prediction Models

The algorithms use a mathematical formula for generating the high resolution images. Interpolation based methods such as lanczos, bilinear and bicubic [11] belong to this category. They generate an high resolution image by evaluating the weighted average of neighbouring pixel values of low resolution image. According to local strength and directions, different ways are used to interpolate missing pixels. we first estimate edge strength and direction through local image gradients.

Pros This methods produces smooth images and are flexible to interpolation along different arbitrary directions.

Cons It suffers from artifacts such as blocking, blurring and ringing.

2.2 Edge Based Methods

This methods use varioius edge features such as width and depth of an edge [3] for learning priors in-order to reconstruct high resolution images. Due to this priors, the high resolution images have rich edges with good sharpness. But this methods fail in maintaining textures.

2.3 Statistical Methods

Image properties such as sparsity of large gradients [10] and heavy-tailed gradient distribution [8] in images have been exploited in this methods to reduce the computational load and to regularize the input images.

2.4 Patch Based Methods

This methods require a set of paired Low resolution as well as high resolution images for training. Then patches are extracted from this images, this patches are then used for learning the mapping between the low resolution patch with its high resolution counterpart. Various methods that are used for learning the mapping functions are weighted average [12], Kernel regression [10], Sparse dictionary representation [9], Neighbor Embedding [1], Anchored Neighbourhood Regression [13], Beta Process Joint Dictionary [7], Gaussian process regression [6]. In-order to combine the overlapping patches, the methods used are weighted averaging [5], Markov Random Fields [4] and conditional random fields [14].

2.4.1 Sparse Representation

In this method we approach this problem from the perspective of compressed sensing.The principle of compressed sensing ensures that the sparse representation can be correctly recovered from the downsampled signal, i.e high resolution images can be easily obtained from the downsampled counterpart. We further show that a small set of randomly chosen raw patches from training images of similar statistical nature to the input image generally serve as a good dictionary, in the sense that the computed representation is sparse and the recovered high-resolution image is competitive or even superior in quality to images produced by other SR methods.

2.4.2 Neighbor Embedding

This method has been inspired by recent manifold learning methods, particularly locally linear embedding (LLE). Specifically, small image patches in the low- and high-resolution images form manifolds with similar local geometry in two distinct feature spaces.As in LLE, local geometry is characterized by how a feature vector corresponding to a patch can be reconstructed by its neighbors in the feature space. Besides using the training image pairs to estimate the high-resolution embedding, we also enforce local compatibility and smoothness constraints between patches in the target high-resolution image through overlapping.More specifically, generation of a high-resolution image patch does not depend on only one of the nearest neighbors in the training set. Instead, it depends simultaneously on multiple nearest neighbors in a way similar to LLE for manifold learning. An important implication of this property is that generalization over the training examples is possible and hence we can expect our method to require fewer training examples than other learning-based super-resolution methods.

2.4.3 Anchored Neighbourhood Regression

This paper proposes fast super-resolution methods while making no compromise on quality. First, we support the use of sparse learned dictionaries in combination with neighbor embedding methods. In this case, the nearest neighbors are computed using the correlation with the dictionary atoms rather than the Euclidean distance. We obtained similar or improved quality and one or two orders of magnitude speed improvements.

2.4.4 Beta Process Joint Dictionary

This paper addresses the problem of learning overcomplete dictionaries for the coupled feature spaces, where the learned dictionaries also reflect the relationship between the two spaces. A Bayesian method using a beta process prior is applied to learn the over-complete dictionaries. Compared to previous couple feature spaces dictionary learning algorithms, our algorithm not only provides dictionaries that customized to each feature space, but also adds more consistent and accurate mapping between the two feature spaces. This is due to the unique property of the beta process model that the sparse representation can be decomposed to values and dictionary atom indicators. The proposed algorithm is able to learn sparse representations that correspond to the same dictionary atoms with the same sparsity but different values in coupled feature spaces, thus bringing consistent and accurate mapping between coupled feature spaces. Another advantage of the proposed method is that the number of dictionary atoms and their relative importance may be inferred non-parametrically.

2.4.5 Deep Coupled Autoencoders

One of the recent patch based method for generating high resolution images is by using deep coupled autoencoders.

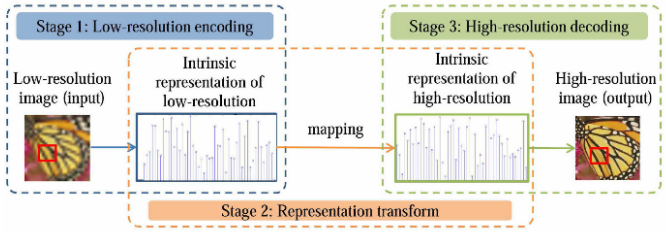

Figure 1: Obtaining Intrinsic Representation [15]

Stage 1 uses an autoencoder for learning the intrinsic representation of low resolution images by encoding the LR images to a latent space and then reconstructing original image from this latent space. The latent encoding of the LR input image is the intrinsic representation. Similarly Stage 3 uses another autoencoder for learning the intrinsic representation of high resolution images. Stage 2 is a normal Neural network which learns the mapping between LR intrinsic representation and HR intrinsic representation.

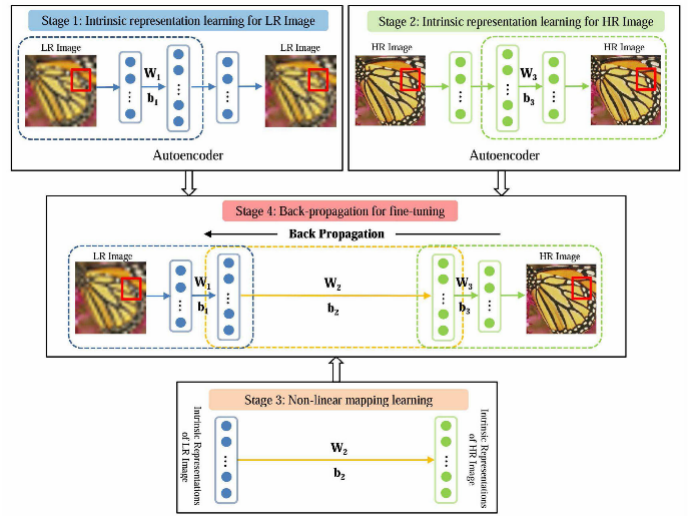

Figure 2: Coupled Deep Autoencoder Fine Tuning [15]

Stage 4 is the fine tuning process in which the learned parameters obtained from the Stage 1, Stage 2 and Stage 3 are used for constructing a model which converts the input LR image to its LR intrinsic representation using encoder of Stage 1 then a predicted HR intrinsic representation is obtained from this LR representation using model of Stage 3 and finally decoder of Stage 2 is used for reconstructing the HR image associated with this HR intrinsic representation.
The procedure of obtaining the HR image from LR image using Deep coupled Autoencoders involves 2 steps.

- Obtain trained models from Stage 1,2 and 3.
- Fine-Tuning The model in Stage 4.

3 Implementation

3.1 Capsule Network Encoder and Decoder

Convolutional Neural Network are one of the reasons that the deep learning is so popular today. But they have lot disadvantages such as they are not variant to affine transformations such as rotations and translations Also the pooling layer of CNN leads to loss of vital information. Therefore, there was a need for some sort of algorithm which is variant to affine transformation and maintans the part-whole based relationship in an image.

Capsule Networks tries to learn this part-whole using dynamic routing algorithm. This dynamic routing algorithm has a property of explaining away the properties of an incoming activation vector, This property makes the capsule networks invariant to affine transformations such as rotation, translation and scaling.

Therefore, we hypothesized that the capsule networks can be used for learning the part-whole relationships between the edges and colors for generating the high resolution images from the low resolution counterpart.

The same concept of Deep coupled autoencoders is used for training a capsule encoder decoder network for construction of high resolution images.

3.1.1 Model

The model consists of three networks.

- Capsule Encoder Decoder for Low Resolution Representation
- Capsule Encoder Decoder for High Resolution Representation
- Dense Layers Neural Network for Mapping from Low resolution to high resolution

Capsule Encoder Decoder For Low Resolution Representation :

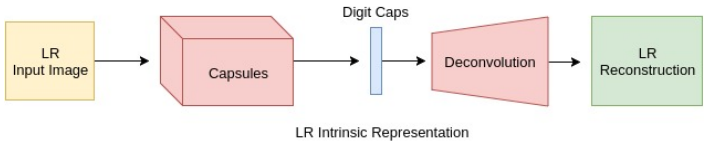

Figure 3: Capsule Encoder Decoder For Mapping Low Resolution

This network consists of Capsule network with dynamic routing algorithm as the encoder for learning the representation of low resolution images. The representation is learned in the digit caps layer. Then an decoder with deconvolutional layers is applied in order to convert the output of this digit caps layer to representation of an image.

Capsule Encoder Decoder For High Resolution Representation :

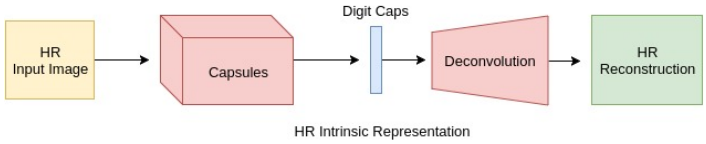

Figure 4: Capsule Encoder Decoder For Mapping High Resolution

This network uses the same concept of using capsule networks with dynamic routing algorithm as an encoder and deconvolutional layers as the decoder.

Dense Layer Artificial Neural Network for mapping LR to HR :

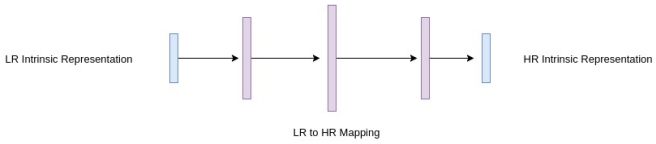

Figure 5: Fully Connected Layers for learning HR representation from LR representation

This network uses artificial neural network in order to learn the mapping from low resolution representation to high resolution representation

Encoder LR Architecture :

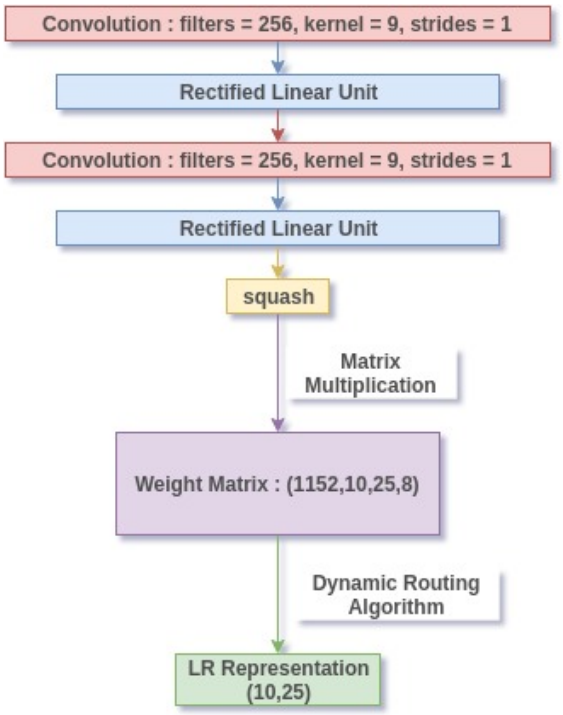

Figure 6: Capsule Encoder for learning the Low resolution latent representation

Above is the architecture for the encoder using capsule network for training the encoder decoder of Low resolution patches. The input to the network is an image of size 14x14 and it produces a latent space vector of size 250 which will be passed to the decoder after reconstruction to size of (5,5,10)

Decoder LR Architecture :

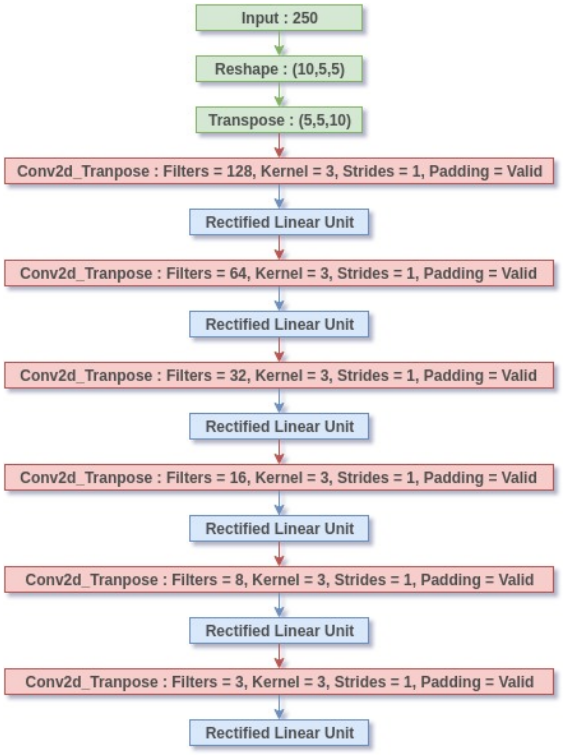

Figure 7: Decoder for reconstruction of images from this latent representation

Above is the decoder for reconstructing the low resolution images. The input to the decoder is a tensor of size 250 which will be reshaped to size (5,5,10) and then passed to several deconvolution layers. The output of the decoder is an image of size 14x14

Encoder HR Architecture :

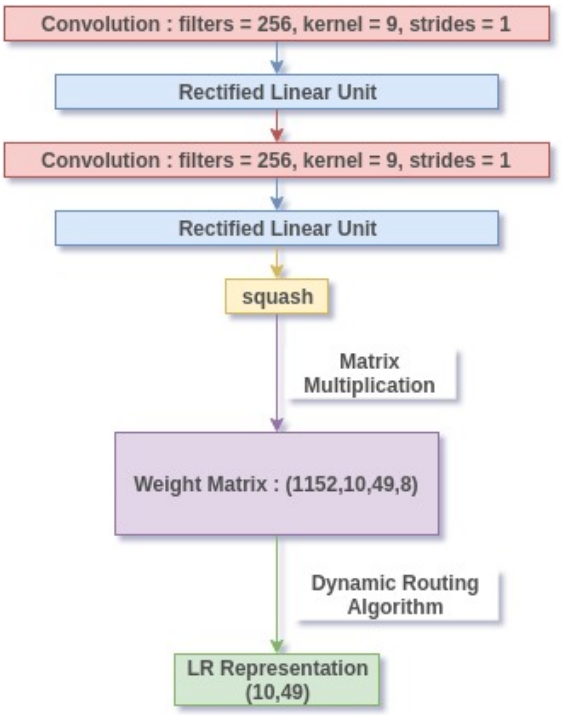

Figure 8: Capsule Encoder for Learning the High resolution latent Representation

Above is the architecture for the encoder using capsule network for training the encoder decoder of High resolution patches. The input to the network is an image of size 28x28 and it produces a latent space vector of size 490 which will be passed to the decoder after reconstruction to size of (7,7,10)

Decoder HR Architecture :

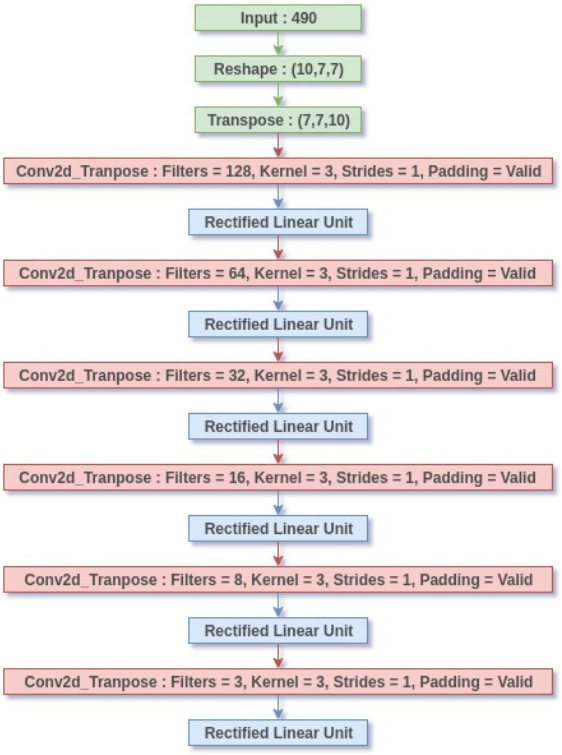

Figure 9: Decoder for reconstruction of images from this latent representation

Above is the decoder for reconstructing the low resolution images. The input to the decoder is a tensor of size 490 which will be reshaped to size (7,7,10) and then passed to several deconvolution layers. The output of the decoder is an image of size 28x28

Dense Layers :

Dense Layer : input = 250, output = 786
Rectified Linear Units
Dense Layer : input = 786, output = 1024
Rectified Linear Units
Dense Layer : input = 1024, output = 786
Rectified Linear Units
Dense Layer : input = 786, output = 490
Rectified Linear Units

Table 1: Encoder Architecture

Above is an architecture of the network which will be used for learning a mapping from the low resolution intrinsic representation to the high level intrinsic representation of the vector. It uses 4 dense layers for converting latent space vectors of size 250 to latent space vectors of size 490.

3.1.2 Training

The training is performed in 4 major steps similar to that of the deep coupled encoder decoder network used for single image super resolution. In the intial stage the Low Resolution Encoder and Decoder is trained using Mean Squared Error. Similarly in the next stage High Resolution Encoder and Decoder is trained using Mean Squared Error. In the next stage both the learned weights of the LR and HR encoders are taken and then the output of the LR encoder was passed to the Dense Layers and then the output of the dense layers was trained using Mean Squared Error to be similar to that of output of HR encoder.

In the last stage we combine all the parts consisting of the LR encoder, Dense Layers and HR decoder and then perform the finetuning of the entire network.

For training of all the parts of the network the input was passes as the patches of the image convolutionally

3.1.3 Observations

Since, the patches were passed convolutionally to network, it was observed that a single low resolution images consisted of around 3000 patches therefore it took around 5-10 minutes for the network to train on a single image. When kept on training it was observed that it took around 3 days on GTX 1060 GPU card and not even a single epoch was completed. On test the network it was found that it is not trained well and was producing blurry results. Some of the results and the respective PSNR values are given in the images below.

Output :

Figure 10: Ground truth image with shape 512x512x3

Figure 11: Produced Image by capsule network of shape 1024x1024x3

PSNR : Y = 12.890479749935409 **Cb** = 20.336101776157022 **Cr** = 20.510390277192244

3.2 Variational Capsule Encoder with Generative Adversarial Networks

Variational Capsule Encoder Network :

Since in the above model it was observed that the network took lot of time to train and the output produced were blurry. Therefore, We thought of experiment with the GAN so that the adversarial loss of GAN will consider the entire structural information of the image rather than just pixel wise loss that is obtained from the Mean squared error.

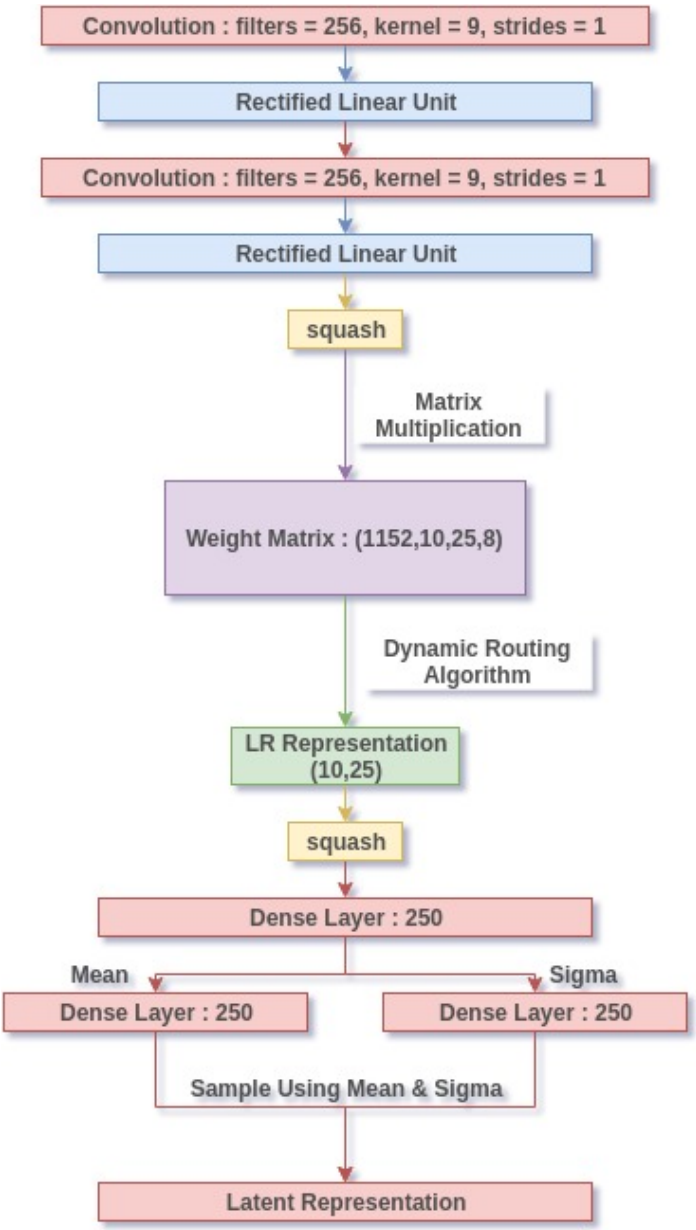

Figure 12: Variational Capsule Encoder

The concept of variational auto encoders were used in the encoder for the efficient training of the encoder.

Generator Network :

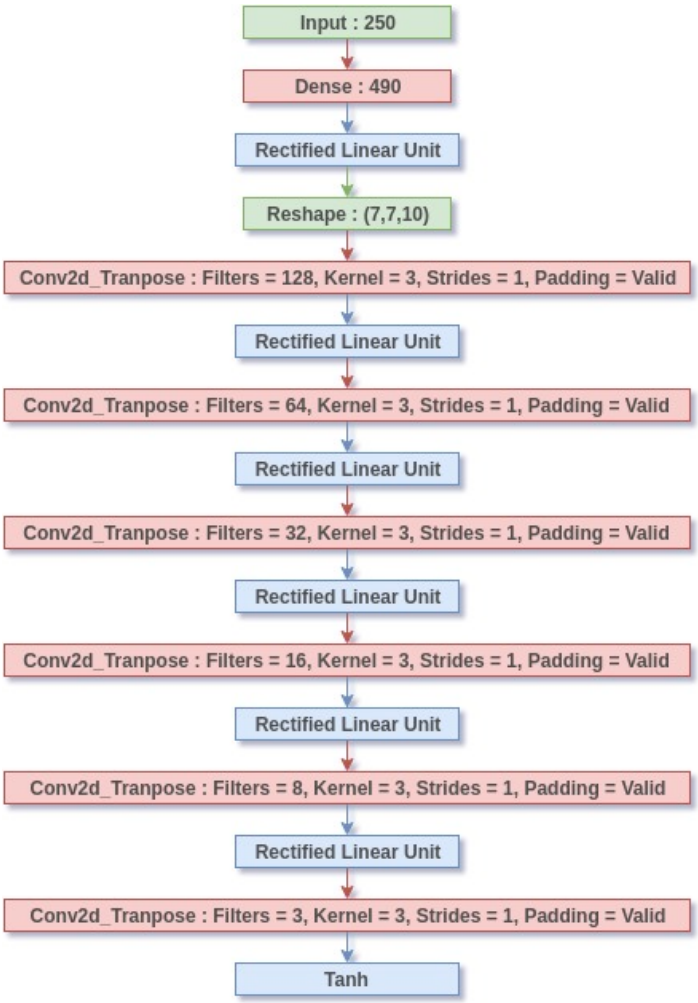

Figure 13: Generator Network

The Generator network reconstructs the images from this latent space.

Discriminator Network :

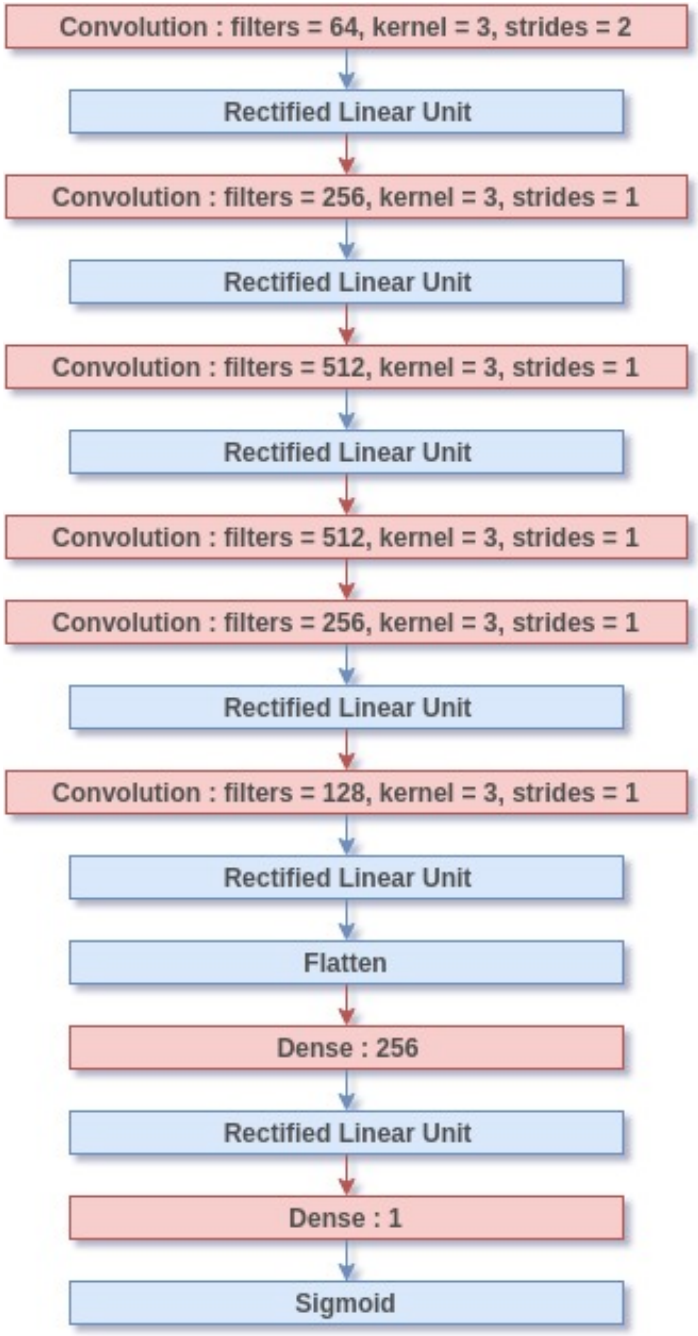

Figure 14: Discriminator Network

The Discriminator here is used as an loss function which will enforce the decoder to maintain the structure in the image and therefore produce sharper images.

3.2.1 Training

For training we used reconstruction loss, KL Divergence loss and Perceptual Loss on the encoder network and Adversarial Loss on the generator and discriminator Networks

3.2.2 Observations

During training the network it was observed that the discriminator loss would reduce very rapidly to a very low values. Once this phenomenon happens the encoder is not able to learn anything.

4 Conclusion & Future Work

From the extensive experimentation it was observed that the capsule networks are very hard to train. The use of dynamic routing algorithm is responsible for this low speed. Also it was observed that the reconstruction loss only considers the pixel wise information i.e it was only performing the average of the loss. Another thing that was observed was that the discriminator in the Vcegan was able to learn the data disctribution very easliy, Making the learning process of generator very slow.

The Future work can be of making the routing by agreement protocol more efficient so that the training becomes faster. Also there should be some sort of way in order to reduce the learning of the disciminator inorder to make the generator learn the intrinsic distribution.

[1] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2): 295–307, 2016.

[3] Raanan Fattal. Image upsampling via imposed edge statistics. In *ACM transactions on graphics (TOG)*, volume 26, page 95. ACM, 2007.

[4] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, 22(2):56–65, 2002.

[5] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 349–356. IEEE, 2009.

[6] He He and Wan-Chi Siu. Single image super-resolution using gaussian process regression. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 449–456. IEEE, 2011.

[7] Li He, Hairong Qi, and Russell Zaretzki. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 345–352. IEEE, 2013.

[8] Jinggang Huang and David Mumford. Statistics of natural images and models. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference On.*, volume 1, pages 541–547. IEEE, 1999.

[9] Yang Jianchao, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[10] Kwang In Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010.

[11] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.

[12] Jian Sun, Nan-Ning Zheng, Hai Tao, and Heung-Yeung Shum. Image hallucination with primal sketch priors. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–729. IEEE, 2003.

[13] Radu Timofte, Vincent De, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1920–1927. IEEE, 2013.

[14] Qiang Wang, Xiaoou Tang, and Harry Shum. Patch based blind image super resolution. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 709–716. IEEE, 2005.

[15] Kun Zeng, Jun Yu, Ruxin Wang, Cuihua Li, and Dacheng Tao. Coupled deep autoencoder for single image super-resolution. *IEEE transactions on cybernetics*, 47(1):27–37, 2017.