ECDN: Efficient Cascading Dense Network for Super Resolution

Tao Lu
luttul@umich.edu

Siyuan Xu
xsyarctg@umich.edu

Tianyue Li
umlty@umich.edu

1. Introduction

Image super-resolution (SR) refers to the techniques that increase the resolution from some low-resolution (LR) images. SR techniques are needed in various fields due to the limits of hardware like sensors and cameras. For example, in medical imaging, SR is applied to stabilize Magnetic Resonance Imaging [2]. In the application of surveillance, SR is used to improve the facial recognition result from the LR image obtained by security cameras [8].

There have been many researches focusing on SR and many methods has been well developed. In general, these methods can be divided into two categories: classical computer vision methods and deep learning methods. Typically, the classical method includes prediction-based methods, edged-based methods, statistical methods, etc [7]. However, in recent years, deep learning based SR have received more attention and have shown promising performance on various benchmarks of SR. With the development of deep learning, the techniques used to SR are also upgraded. Early researches typically used Convolutional Neural Networks(CNN), while more recent researches use Generative Adversarial Nets(GAN) [4].

For our project, we build our Efficient Cascading Dense Network (ECDN) mainly on CARN [1]. We choose this method because compared with other CNN for SR task, CARN achieved a more ideal balance between the training speed and accuracy [1]. But still, CARN is not so effective. It requires more than 4 GPU-days to reach a good performance. Also, their parameter size is too large and can be reduced while preserving the performance to the large extent. Our ECDN, as well as its slimmer version, ECDN_M, achieves its performance with almost half of the original parameters.

In order to improve the CARN, we chose methods from [7], and refine the design of CARN. For example, we use iterative up-and-down sampling as our SR framework. Inspired by its cascading design, we develop our network based on denseNet, as it's a good choice to reduce model size and reuse feature maps. Finally, we try to integrate the perceptual similarity (Learned Perceptual Image Patch Similarity [9]) into our model, using it as the loss function instead of traditional pixel-based metric like peak signal-to-noise ratio(PSNR).

Our main contribution in this research can threefold:

- Based on CARN, we develop two new models, ECDN and ECDN_M, that can achieve same performance while reducing model size considerably.
- We use perceptual similarity to measure the model performance, which is a more reasonable selection mechanism.
- We tried iterative up-and-down sampling frameworks, which improves the performance without adding too much overheads.

2. Approach

In order to learn a mapping function from low resolution image I_{LR} to high resolution image I_{HR}, we have two main models: ECDN and ECDN-M. ECDN is designed to be a high-performance SR model, but with less parameters and convolution layers compared with traditional DenseNet SR model. ECDN-M is adopted from ECDN with good performance but much less parameters and faster training speed.

As shown in Figure 1(A), ECDN is composed of several parts: the entry and exit layers for matching input and output dimensions, three dense blocks for learning features, and one upsampling layer for reconstructing high resolution details. We use ReLu as activation function, and each convolution block is followed by a ReLu except for the exit layer. Notice that the outputs of intermediary dense blocks are cascaded and sent as inputs into higher dense blocks. The cascaded outputs will go through a 1×1 convolution layer before being used as inputs of next dense block, so that even in a deep network, the depth of dense block inputs is reasonable. This cascading network structure is adopted from the model provided by [1], which is a

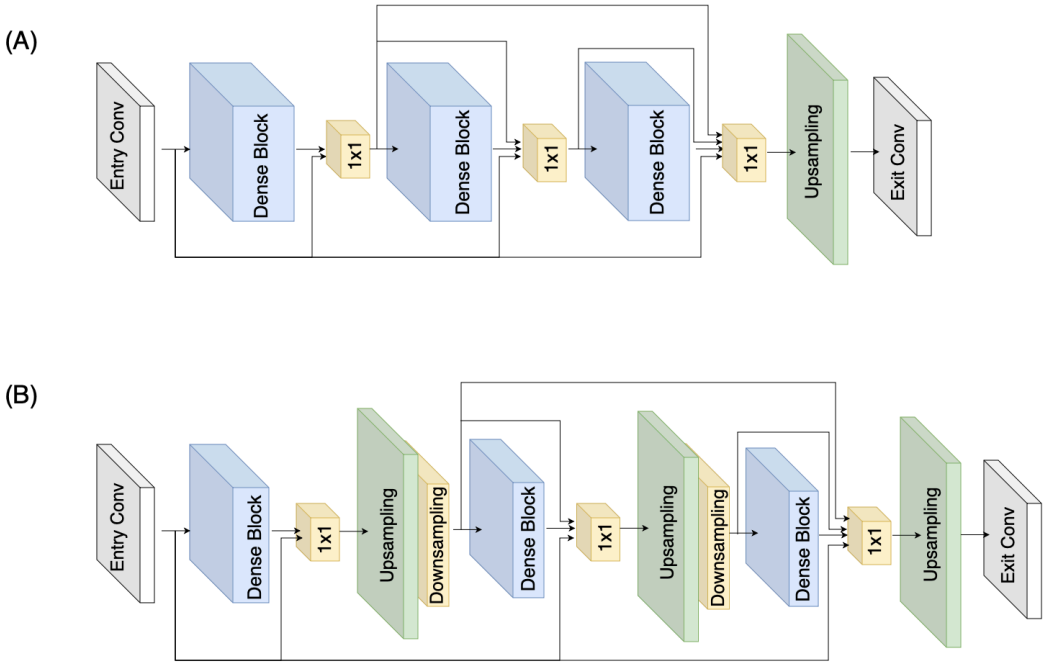

Figure 1. Structure of Proposed Networks

cascading residual network containing three local cascading blocks. This cascading residual network can quickly propagate information from lower to higher layers with multiple shortcuts. We apply post-upsampling framwork in ECDN to save parameters.

As shown in Figure 1(B), ECDN-M is adopted from ECDN, which still use the entry and exit layers for matching input and output dimensions, and three dense blocks for learning features. The main difference is that the dense blocks of ECDN-M have less convolution layers, and ECDN-M applies iterative up-and-down sampling framework where an up-and-down layer is applied after each intermediary dense block. Since less convolution layers usually weaken the performance with less parameters, we use iterative up-and-down sampling framework to keep the performance good. Notice that we use the same upsampling layers in up-and-down sampling and the final upsampling layer, so that the upsampling parameters are reused.

2.1. Dense Blocks

As shown in Figure 2, inside each dense block, the higher layer will use the output previous layer as input, and concatenate its input and convolution result as output:

$$H_0 = X$$
$$H_i = Concat(f(H_{i-1}), H_{i-1})$$

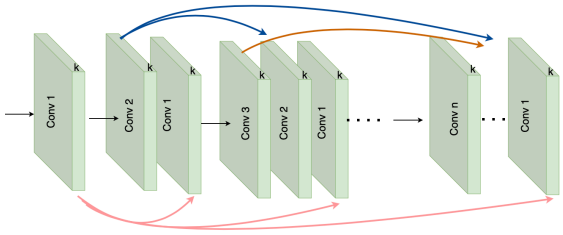

Figure 2. Structure of Dense Block

where H_i is the output of ith layer, X is the block input, $Concat$ is the concatenation function, and f is the convolution function. Notice that if there are n layers inside the dense block, and the output depth of each layer is k, then the final output depth will be nk.

For DENSEBLOCK, the input depth is set to be 64, and we set $n = 8$, $k = 16$, which means the output depth will be 128. For DENSEBLOCK-M, the input depth is set to be 64, and we set $n = 4$, $k = 16$, which means the output depth will be 64.

2.2. Upsampling

Instead of using interpolation, we apply the Sub-Pixel Convolution as the upsampling methond. Given a low resolution $I_{LR} : H \times W \times C$, after some convolution operation we may get $f(I_{LR}) : H \times W \times Cr^2$. As proposed in [5], an periodic shuffling operator can

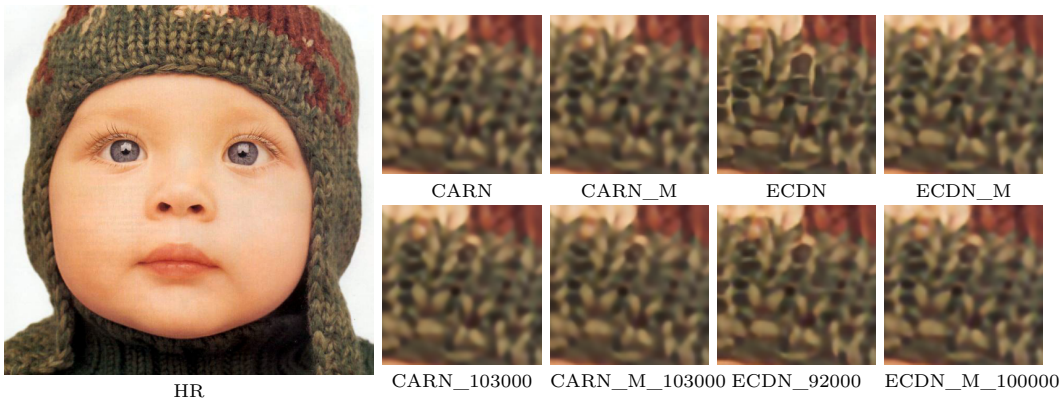

Figure 3. Result of example from Set5

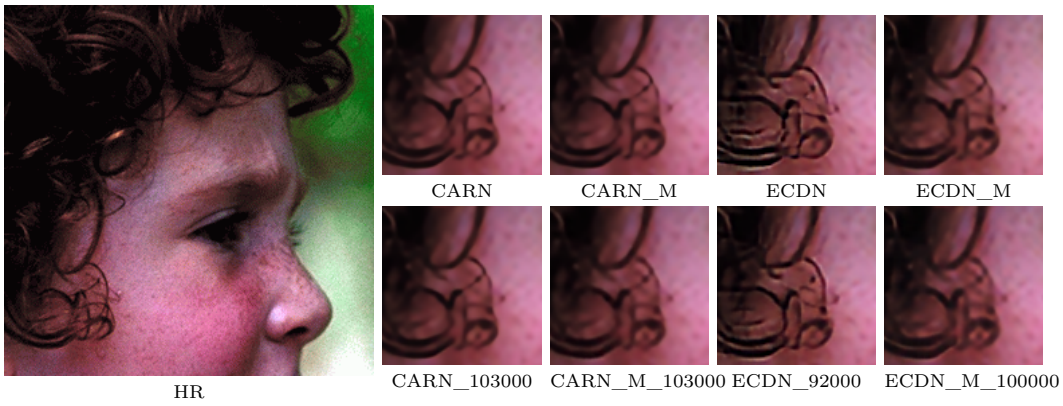

Figure 4. Result of example from Set14

be applied on $f(I_{LR}) : H \times W \times Cr^2$ to rearrange the tensor to $I_{HR} : rH \times rW \times C$. The advantage of sub-pixel convolution is computational efficient and make more use of previous convolution results.

In order to investigate the effect of upsample framework, we implemented Pre-upsampling SR and Iterative Up-and-down Sampling SR based on the same network architecture. For Pre-upsampling SR, we move the upsample layer to the beginning of the model. For Iterative Up-and-down Sampling SR, we add a upsample and downsample layer pair after each cascading blocks. The downsample is implemented by interpolation. The problem with Pre-upsampling is that it increase the memory usage substantially, so we decide to drop the pre-upsampling framework. The Iterative Up-and-down Sampling performs better if the network is relatively simple, so we decide to use Iterative Up-and-down Sampling in our ECDN-M model.

2.3. Optimizer, learning rate and loss function

We use ADAM as our optimizer with adaptive learning rate. The formula of learning rate is given by

$$lr_{new} = lr_{old} \times (\frac{1}{2})^{\frac{s}{d}}$$

where s is the number of steps it has refined and d is the decay factor provider by user, which is 400000 in our case. We choose $L1$ as our loss function. All of these settings are the same as CARN.

3. Experiments

3.1. Data

We use DIV2K [6] as our training data and Urban100 [3] as test data. We also use Set5, Set14 and B100 [3] as benchmark test dataset to evaluate CARN model and our customized models. These data are the popular choice of super resolution.

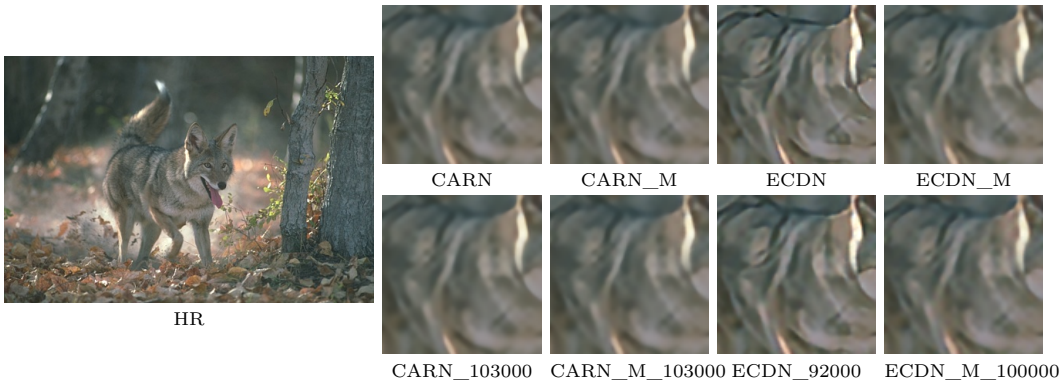

Figure 5. Result of example from B100

Model	LPIPS score	PSNR score	# parameter	training time (GPU-day)
CARN	35.7	8875	1591963	fully trained (4)
CARN_M	36.1	8854	414811	fully trained (4)
CARN_103000	36.0	8852	1591963	1
CARN_M_103000	37.3	8822	414811	1
ECDN (ours)	35.7	8702	694555	fully trained (2.5)
ECDN_M (ours)	36.8	8814	396955	fully trained (2.5)
ECDN (ours)	36.8	8741	694555	1
ECDN_M (ours)	37.8	8795	396955	1

Table 1. Quantitative result of CARN and our model

3.2. Metrics

Choosing an appropriate measure is necessary for a super resolution model to grow on a right track. Traditionally, pixel-based metrics like PSNR and SSIM are used since they are easy to set up in the deep learning scenario. However, such metrics assumes pixel independence and therefore do not work well in the super resolution tasks.

In our project, we use Learned Perceptual Image Patch Similarity (LPIPS) [9] metric. It trains on a special dataset that they collected via two tests: 2 alternative forced choice (2AFC) test and just noticeable difference (JND) test. 2AFC allows the data to be labelled close to human judgment, while JND alleviates the subjectivity of participants. The insight into perceptual similarity is the key to the success of this metric. Also, the use of deep neutral network like VGG and AlexNet helps quantify perceptual similarity as the distances.

Despite the good performance of selecting better results, LPIPS metric is fairly memory expensive. It needs about 10GB each time we set checkpoints every 1000 steps. Therefore, it might be a better choice to use it as a way of comparing different models instead of a loss function during training.

3.3. Results

The original high resolution image and super resolution image pairs are shown in Figure 3, 4 and 5, together with their PSNR and perceptual distance values in Table 1. Note that the LPIPS and PSNR score are the sum of scores for individual image in Set5, Set14 and B100.

4. Implementation

Since Densenet has the advantage of reusing feature maps from preceding layers and avoiding the relearning of redundant features, we implement and apply Densenet in our model. Instead of three local cascading blocks in [1], we use three densenet blocks, and at the end of each densenet block, a 1×1 convolution layer is applied so that the input and output dimensions are both 64. Also we use Iterative Up-and-down Sampling in our ECDN-M model. The way of connecting dense blocks in our model is the same as the way of connecting cascading blocks in [1].

Moreover, to use LPIPS, we write a compare program to evaluate the performance of models on Set5, Set14 and B100. the model with lowest LPIPS score is the best. It will also calculate PSNR score for reference.

It's lightweight and can run on our own computers.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European conference on computer vision (ECCV), pages 252–268, 2018. 1, 4

[2] Hayit Greenspan. Super-Resolution in Medical Imaging. The Computer Journal, 52(1):43–63, 02 2008. 1

[3] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5197–5206, 2015. 3

[4] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4681–4690, 2017. 1

[5] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016. 2

[6] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, July 2017. 3

[7] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. IEEE transactions on pattern analysis and machine intelligence, 43(10):3365–3387, 2020. 1

[8] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. Signal Processing, 90(3):848–859, 2010. 1

[9] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, 2018. 1, 4