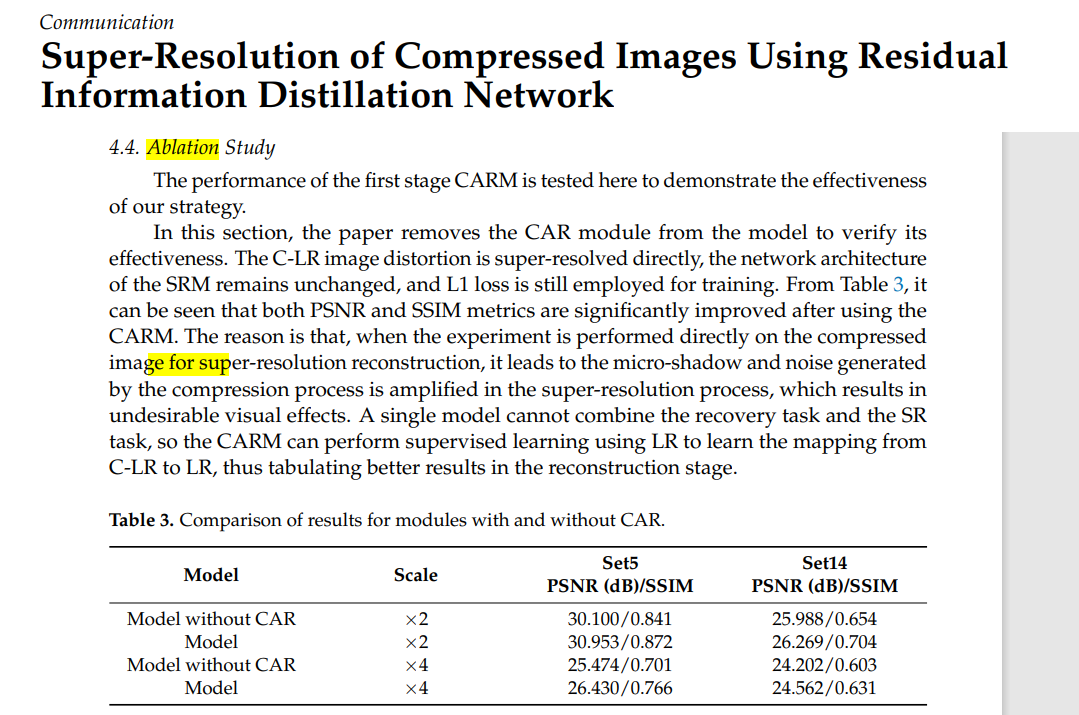
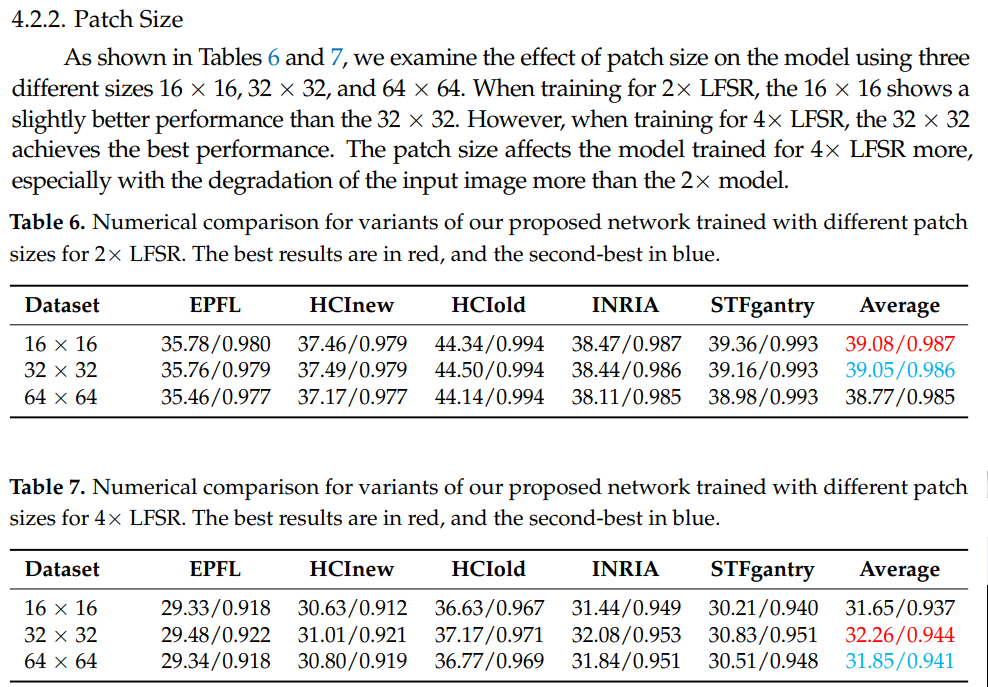


Mean Inference Time

The inference time is an important factor of image super-resolution methods other than the SR performance. In this part of ablation we shows the inference test time on publicly available datasets such as Set5, Set14, BSD100, Urban and Manga109 with enlargement factor 2x, 3x, 4x and 8x as shown in Figure. Form the Figure, it is clearly seen that as the higher enlargement factor increasing the more processing time as compared to small scale factor, because input image having 2x enlargement factor is higher than 8x enlargement factor. Therefore, the computational cost of 2x input image is higher than 8x enlargement factor.







Patch Size

As shown in Table 6 and 7, we examine the effect of patch size on the model using three different sizes 16 x 16, 32 x 32, and 64 x 64

<https://tungmphung.com/elu-activation-a-comprehensive-analysis/>

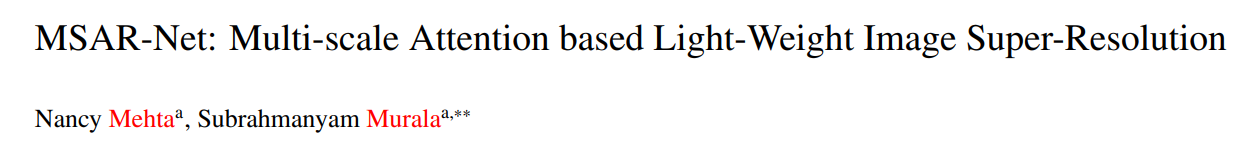
<https://arxiv.org/pdf/1604.04112.pdf>

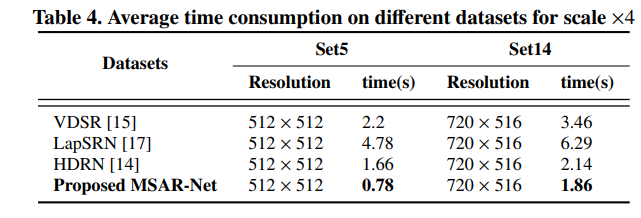
<https://towardsdatascience.com/deep-study-of-a-not-very-deep-neural-network-part-2-activation-functions-fd9bd8d406fc>

<https://www.askpython.com/python/leaky-relu-activation-neural-networks>

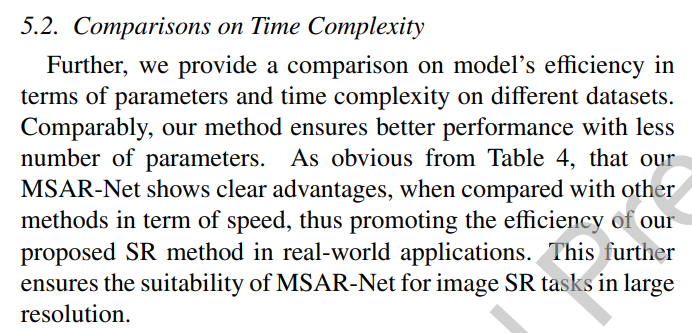
<https://medium.com/analytics-vidhya/relu-activation-increase-accuracy-by-being-greedy-6b93c7c40882>

<https://github.com/christianversloot/machine-learning-articles/blob/main/using-leaky-relu-with-keras.md>





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Set5 | Resolution | Bicubic | SRCNN | VDSR | LapSRN | SENext |
| Baby.png | 512 × 512 | 2.2 |  |  |  |  |
| Bird.png | 288 × 288 |  |  |  |  |  |
| Butterfly.png | 256 × 256 |  |  |  |  |  |
| Head.png | 280 × 280 |  |  |  |  |  |
| Woman.png | 228 × 344 |  |  |  |  |  |
| Average time(s) |  |  |  |  |  |  |



Activation Function Analysis

Maas et al. [1] evaluated a variant of ReLU with a gradient more amenable to optimization, which leads to Leaky ReLU [1]. Most common problems facing ReLU activation function is a Dying ReLU, which resolve by LReLU as shown in Figure 3 and 4. In figure 3 and 4 shows that network with LReLU has a more quick convergence and help networks to train faster. In Figure we observed that

Outputs can be either positive or negative. Studies showed that functions with 0-centered outputs help networks train faster. Although ELU’s outputs are not distributed around 0, the fact that it does produce negative values makes it be preferred in this sense compared to ReLU. In practice, it seems that networks with ELU converge more quickly than with ReLU, even though the exponential computation in ELU () requires longer processing time.

ELU is not piece-wise linear, this makes it model the non-linearity better. The plateau in the negative region helps in maintaining robustness and stability.

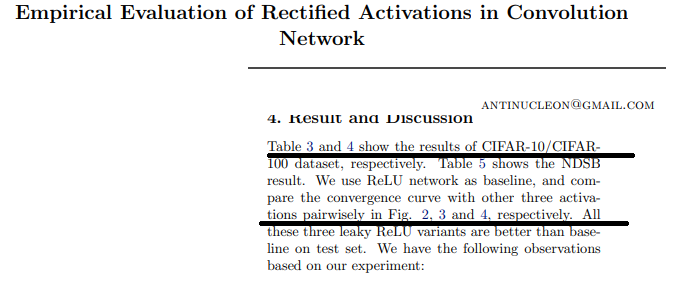
Disadvantages

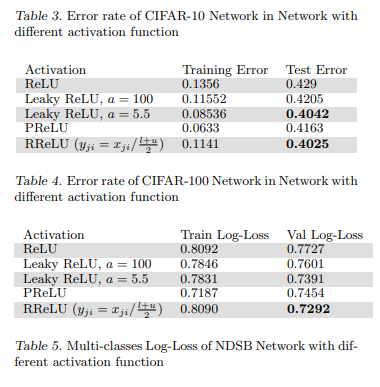
α is fixed, not learned.

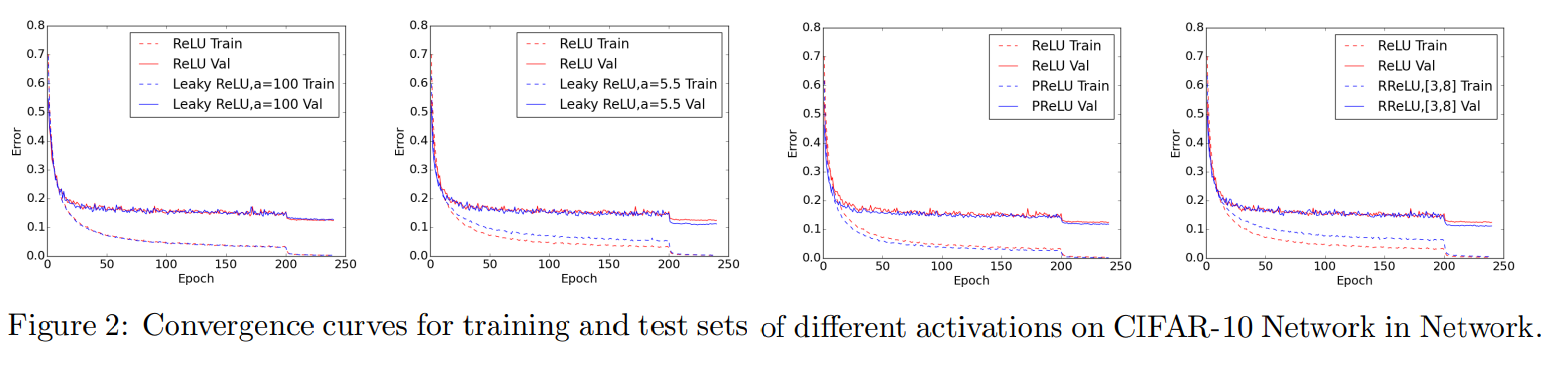
It still suffers from Gradient Exploding and Gradient Vanishing Problem, thus the help of normalization methods may remain necessary sometimes (e.g. [**this paper**](https://arxiv.org/pdf/1604.04112.pdf)). SELU, which was built from ELU with an innate ability to self-normalize, was introduced to address this problem.

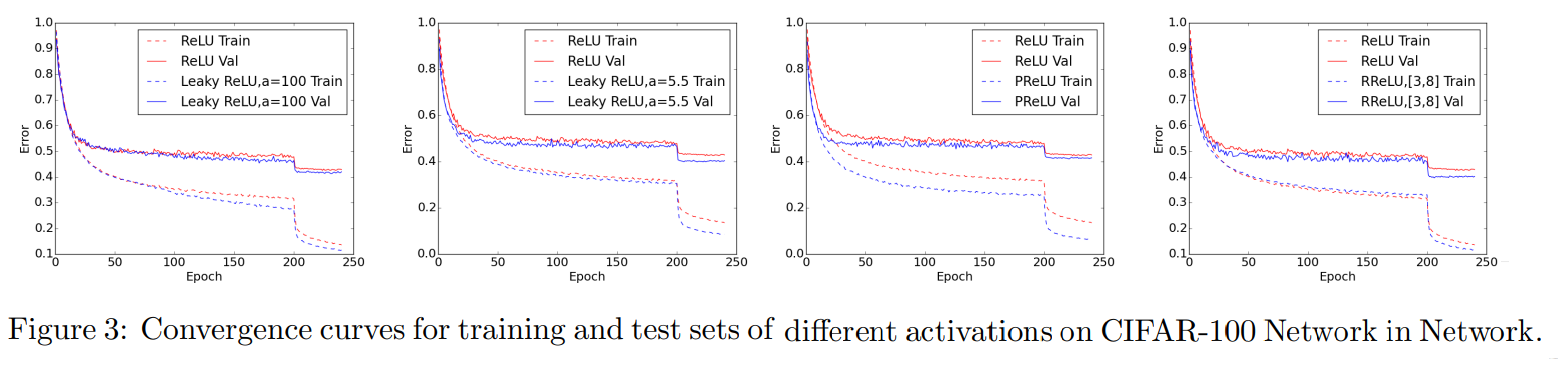
It is not 0-centered. Although ELU does produce negative outputs, the fact that it is not 0-centered makes it seemingly sub-optimal. Another nonlinearity called Parametric ELU is introduced to mitigate this issue.

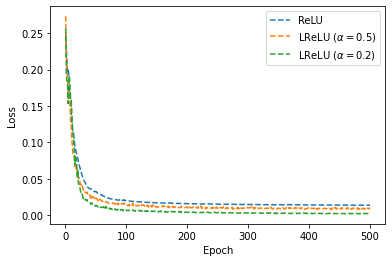
No sparsity. ReLU outputs 0 for negative inputs, which is bad in some senses but also good in some others, as elaborated in





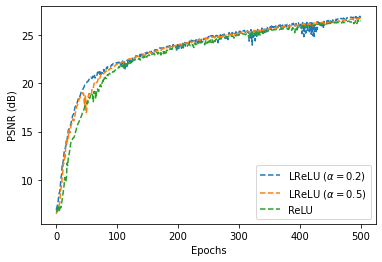




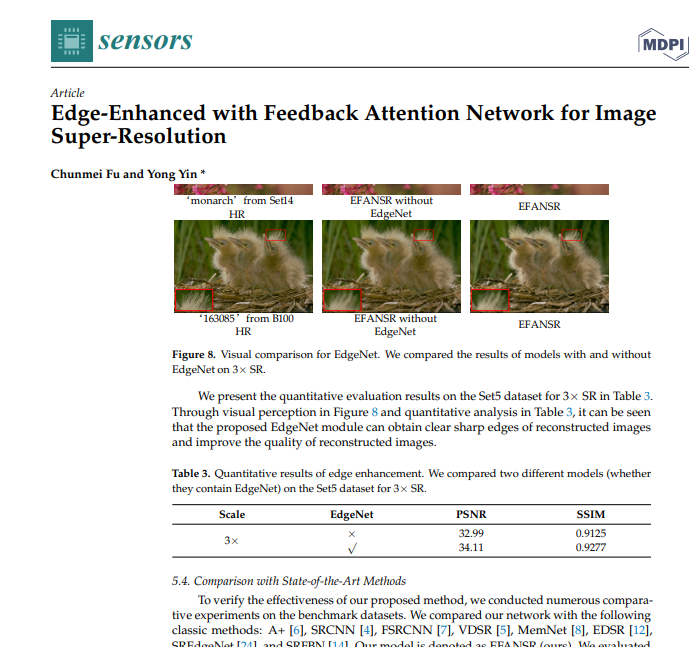


We observed that LReLU (alpha = 0.2) has very close to

**Performance of convergence curves with activation functions**



**Performance analysis of PSNR versus activation functions.**

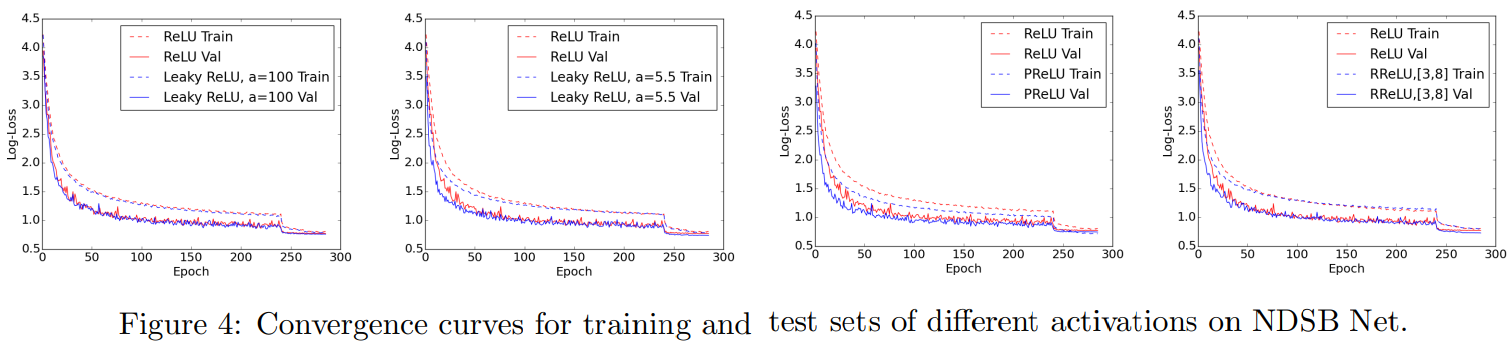
****

We discuss the quantitative comparisons on benchmark test datasets such as Set5, Set14, BSDS00, Urban100 and Manga109 with enlargement factor 4× as shown in Table 3. For quantitative ablation study purpose we have trained a two model separately such as SENext model without using the SFEB and with SFEB. Both model is trained on same specification of GPU discuss in Section IV. For Training purpose, we used only Yang91 image dataset. Table 3 clearly shows that model with SFEB achieved the better performance as compare to without SFEB.

evaluation Information observed from Table 3

Table 3. Quantitative evaluations with and without SFEB on enlargement factor 4×.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scale | Dataset | SFEB | PSNR (dB) | Parameters |
| 4× | Set5 | × | 29.79 | 52K |
|  |  | ✓ | 30.37 | 54K |
| 4× | Set14 | × | 27.99 | 52K |
|  |  | ✓ | 28.31 | 54K |
| 4× | BSD100 | × | 28.61 | 52K |
|  |  | ✓ | 28.91 | 54K |
| 4× | Urban100 | × | 25.89 | 52K |
|  |  | ✓ | 26.14 | 54K |
| 4× | Manga109 | × | 26.65 | 52K |
|  |  | ✓ | 26.99 | 54K |



Performance of convergence curves with activation functions

ReLU , Leaky ReLU having alpha = 0.1 and 0.3.

for traing

1. Maas, A.L., A.Y. Hannun, and A.Y. Ng. *Rectifier nonlinearities improve neural network acoustic models*. in *Proc. icml*. 2013. Atlanta, Georgia, USA.