Real-Time Single Image and Video Super-Resolution Using Sub-Pixel Convolution (A Reproducibility Effort)

Anwesh Marwade, Dekel Viner, Jinwan Huang

Electrical Engineering, Mathematics and Computer Science, Delft University of Technology

Abstract

- We look to investigate and in-turn reproduce the work of Shi et al. on image *Super-Resolution* using the novel *ESPCN* architecture as proposed in their paper *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network* [1].
- In analysing the work posited in the paper, we form a deeper understanding of the ill posed problem of Image Super-Resolution (SR).
- Building on the code effort by Jeffrey Yeo (yjn870) [2], we suggest certain improvements by carrying out ablation studies on the model structure and parameters.
- Futhermore, we also reproduce the experiments by training on dataset with 4K images [Kaggle] and present our results with a step towards video SR.
- We additionally built a pipeline for video Super-Resolution from the paper (by Shi et al.), which was not a part of Yeo's [2] previous work.

Efficient Sub-Pixel Convolution

- Up-scaling the resolution of low-resolution (LR) images before the image enhancement step adds to computational complexity. In convolution networks, this complexity severely influences the speed of their implementation. Previous interpolation methods fail to capture crucial information required to solve this ill-posed problem of super-resolution!

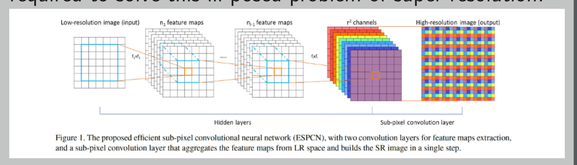


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

- In the proposed architecture, an L layer convolution neural network is directly applied to the LR image after which a *sub-pixel convolution layer* up-scales the LR feature maps to generate the super-resolved image. Exhaustve computation is hence done in the smaller LR space.
- Additionally, not using an explicit form of the interpolation filter, allows the proposed network to capture a more complex mapping (LR to HR) through learning.
- The authors propose using a *periodic shuffle* operation with fractional stride of $\frac{1}{r}$. This operation converts an $H \times W \times C.r^2$ tensor to a tensor with dimensions equal to $rH \times rW \times C$ (which is described in the figure).

Terms explained in the next section.

$$I^{SR} = f^L\left(I^{LR}\right) = PS\left(W_L * f^{L-1}\left(I^{LR}\right) + b_L\right)$$

Proposed Architecture

The L layered ESPCN network can be described as follows:

$$f^1(I^{LR}; W_1, b_1) = \phi(W_1 * I^{LR} + b_1)$$
$$f^l(I^{LR}; W_{1:l}, b_{1:l}) = \phi(W_l * f^{l-1}(I^{LR}) + b_l)$$

- $W_l, b_l, l \in (1, L-1)$: learnable weights and biases. W_l is a tensor of size $n_{l-1} \times n_l \times k_l \times k_l$: where n_l is features per layer l, $n_0 = C$ and k_l is the filter size at layer l.
- b_l is bias vector of length n_l and ϕ is the activation function (non-linearity like ReLU, Tanh and GELU).
- After processing in the LR space, the final layer f^L converts (up-scales) the LR feature-maps to a super-resolved image I^{SR}.
- Additionally, the following loss function (MSE) is optimized:

$$loss(W_{1:L}, b_{1:L}) = \frac{1}{r^2 HW} \sum_{x=1}^{rH} \sum_{x=1}^{rW} \left(I^{HR}_{x,y} - f^L_{x,y}(I^{LR})\right)^2$$

Experiments and Analysis

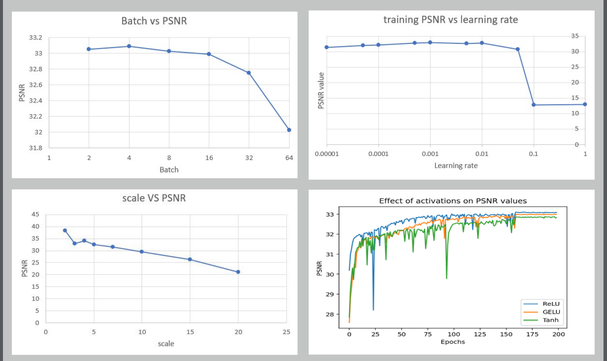


Figure: Experiments on *Set5* image data-set using model: ESPCN(ReLU) trained on 91 images data-set)

1. We observe optimal performance for **batch-size** $= 4$.
2. The performance (PSNR value) drops significantly for **learning rate,** $\alpha > 0.08$.
3. Setting, **scale** $= 3$ gives optimal performance i.e. the highest PSNR value (but it seems to be quite data dependent).
4. Observed performance of activation functions based on **PSNR scores:** *ReLU* $>$ *GELU* $>$ *Tanh*.

Dataset Description

Training
- 91 Image Data-set
- 4K images (train)

Evaluation
- Set5, Set14, BSD500
- 4K images (test)

Results

Dataset	*Scale*	*relu*	*tanh*	*gelu*	*paper*
Set5	3	**33.13**	32.88	32.99	33.00
Set14	3	**29.49**	29.33	29.40	29.42
BSD500	3	**28.87**	28.69	28.77	28.62
4K(test)	4	43.61		46.25	

Table: Avg. PSNR scores as evaluated using different activation functions (Trained using 91-Image Data-set)

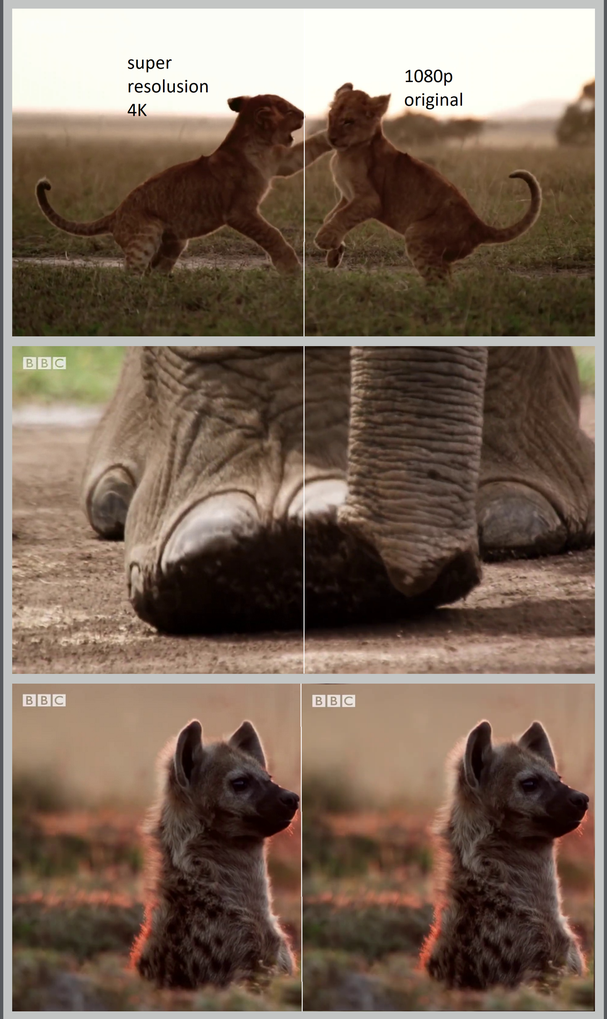
- **Higher** PSNR values were achieved **across the board** as compared to the results posited in the paper.
- Additionally, it should be noted that the model taken from the paper is trained on ImageNet (bigger data-set).

Super-resolution images

Video Pipeline

- Video pipeline takes in 1080p video file as input.
- Breaks down the video into frames (images) by sampling at 30 frames per second.
- Predicts each frame 4k super-resolution according to the trained model and combines the frames.
- **Potential Enhancement**: live video conversion.

Video Super-resolution



References

[1] Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. Shi et al. *http://arxiv.org/abs/1609.05158*

[2] ESPCN-pytorch. Jeffrey Yeo (yjn870) *https://github.com/yjn870/ESPCN-pytorch*

[3] Image Super-Resolution Using Deep Convolutional Networks. Dong et al. CoRR: abs/1501.00092. 2015

[4] Y. Chen and T. Pock, "Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1256-1272, 1 June 2017.

All Experiments were carried out using Tesla K80 GPU through Google Colab