# A Deep Convolutional Neural Network with Selection Units for Super-Resolution

Jae-Seok Choi and Munchurl Kim
School of EE, Korea Advanced Institute of Science and Technology, Korea
{jschoi14, mkimee}@kaist.ac.kr

## Abstract

*Rectified linear units (ReLU) are known to be effective in many deep learning methods. Inspired by linear-mapping technique used in other super-resolution (SR) methods, we reinterpret ReLU into point-wise multiplication of an identity mapping and a switch, and finally present a novel nonlinear unit, called a selection unit (SU). While conventional ReLU has no direct control through which data is passed, the proposed SU optimizes this on-off switching control, and is therefore capable of better handling nonlinearity functionality than ReLU in a more flexible way. Our proposed deep network with SUs, called SelNet, was top-5$^{th}$ ranked in NTIRE2017 Challenge, which has a much lower computation complexity compared to the top-4 entries. Further experiment results show that our proposed SelNet outperforms our baseline only with ReLU (without SUs), and other state-of-the-art deep-learning-based SR methods.*

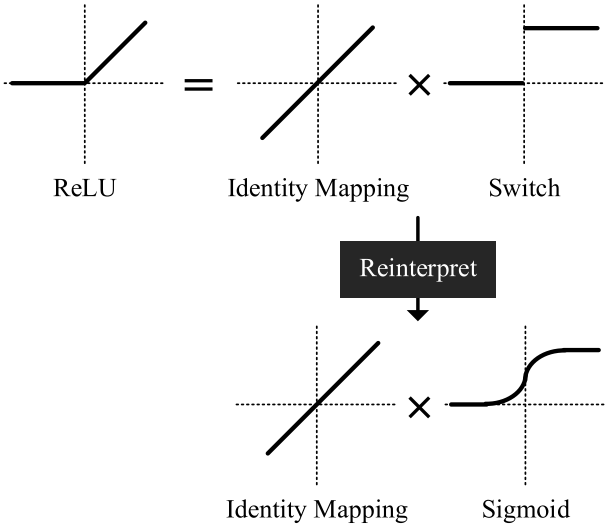


Figure 1: ReLU can be re-defined as a point-wise multiplication of an identity mapping and a switch. This motivates us to create a novel nonlinear unit: selection unit (SU).

## 1. Introduction

With the advent of 4K displays, super-resolution (SR) technique has become more crucial, due to the lack of available 4K contents. Specifically, single image SR is able to reconstruct high-quality high-resolution (HR) images from their low-resolution (LR) counterparts.

SR methods vary from simple methods such as bicubic interpolation [1], to sophisticated methods including example-based SR methods [1]-[29] that utilize external and/or internal image patches for learning LR-to-HR mappings.

Among them, linear-mapping-based SR methods [1]-[11] (LMSR) have been proposed to obtain HR images of comparable quality but with much lower computational complexity. These SR methods mostly comprise of two parts: 1) classifying each LR patch into one of multiple classes; 2) Applying an LR-to-HR linear mapping of the corresponding class to the current LR patch to obtain its HR patch.

Recently, SR methods using deep learning [26]-[29] have shown state-of-the-art performance. Their networks consist of multiple convolutional layers, with rectified linear units (ReLU) between convolutional layers. Here, ReLU is often used to ensure nonlinearity between two adjacent convolutional layers. By using ReLU, networks can learn piece-wise linear mappings between LR and HR images, which results in faster training convergence and higher reconstruction quality [30], compared to networks using other nonlinear functions such as a sigmoid.

Interestingly, we found that ReLU used in deep learning works very similar to linear mapping technique used in LMSR. ReLU can be re-defined as a point-wise multiplication of an identity mapping and a switch. Here, a switch refers to a function where the output of negative inputs is 0 and the output of positive inputs is 1. This switch function acts somewhat similar to how the classification is done in LMSR. However, while LMSR can control how LR patches are classified, ReLU does not perform such operation. This is because the derivative of the switch function is 0, and thus training error cannot be back-propagated through the switches when training the networks. This means ReLU has a very limited control over which data is to be passed or not.

Inspired by this limitation of ReLU operation, we propose a novel nonlinear unit, called selection unit (SU),

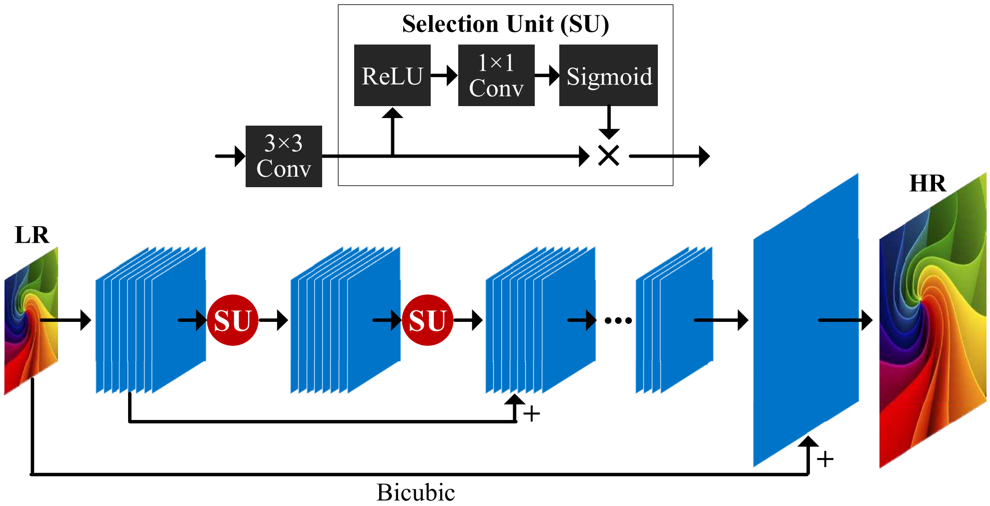


Figure 2: Network architecture of our proposed SelNet using selection units (SU).

which works as a trainable switch. The proposed SU is a multiplication of two modules: an identity mapping and a selection module (SM). Here, SM is a cascade connection of a ReLU, a 1×1 convolution and a sigmoid in a row. Contrary to the switch in ReLU, SM is able to optimize the whole selection control such that training error can be back-propagated through itself. By incorporating SU into a deep CNN, we propose a 22-layered deep CNN structure (SelNet), which can reconstruct HR images of higher quality with a slightly increased complexity, compared to the baseline only with ReLU. Our proposed SelNet was ranked in the 5th place in NTIRE2017 Challenge, with much lower testing time compared to the top-4 entries. Additionally, experiment results show that our proposed SelNet outperforms state-of-the-art deep learning SR methods.

## 2. Related work

### 2.1. Super-resolution

Reconstructing HR images from their corresponding LR input images is well-known as an ill-posed inverse problem [12], [13]. Nevertheless, various SR methods have been proposed to reconstruct HR images of high quality from LR images.

Sparse-representation-based SR methods [13], [15], [17] exploit sparsity and find a sparse combination of pre-trained and complex LR-HR dictionary sets. R. Fattal [14] utilized edge statistics of LR images to reconstruct their sharper HR counterparts. Other SR methods [18], [23], [25] search self-examples within LR images to extract LR-HR relationship.

Linear-mapping-based SR methods [1]-[11] (LMSR) have been also proposed to obtain HR images of comparable quality but with much lower computational complexity. Adjusted anchored neighborhood regression (A+, APLUS) [3] and (ANR) [2] methods search for the best linear mapping for each LR patch, based on the correlation with pre-trained dictionary sets from [13]. Jointly optimized regressors (JOR) [4] method employs an expectation-maximization algorithm with tree to learn and apply the best linear mappings to LR patches. Our previous work SI [6] employs simple edge classification to find suitable linear mappings, which are applied directly to small LR patches to reconstruct their HR version. These LMSR methods [1]-[11] mostly comprise of two parts: 1) classifying each LR patch into one of classes; 2) Applying an LR-to-HR linear mapping of the corresponding class to the current LR patch to obtain its HR patch.

### 2.2. Convolutional neural network for super-resolution

Recently, SR methods using convolutional neural network (CNN) [26]-[29] have shown high PSNR performance. Dong et al. [26], [27] first utilized a 3-layered CNN for SR (SRCNN), and reported a remarkable jump compared to previous SR methods. Recently, Kim *et al.* [28] proposed a very deep 22-layered CNN (VDSR), and by incorporating gradient clipping and residual learning, VDSR reconstructed HR images of even higher PSNR compared to SRCNN.

In these deep learning-based SR methods [26]-[29], rectified linear units (ReLU) are used to obtain nonlinearity between two adjacent convolutional layers. ReLU is a simple function, which has an identity mapping for positive values and 0 for negative. Unlike a sigmoid or Tanh, ReLU does not suffer from vanishing gradient problem, where back-propagated errors become vanished as they go backwards through layers for training. By using ReLU, networks can learn piece-wise linear mappings between LR and HR images, which results in higher reconstruction quality and faster training convergence.

# 3. Our proposed method

## 3.1. Reinterpreting ReLU

We found that ReLU used in deep learning can be interpreted in terms of two modules of linear mapping technique used in LMSR. While linear mapping technique comprises of classification and linear mapping, ReLU can be re-defined as a point-wise multiplication of a switch and an identity mapping. Here, a switch refers to a function where the output of negative inputs is 0 and the output of positive inputs is 1. Combined with a convolutional layer, ReLU selects which values in the feature maps from the previous convolutional layer can be input to the next layer. This is somewhat similar to how the classification is done for selecting which linear mapping is applied in LMSR.

However, while LMSR can control how LR patches are classified, ReLU cannot do so. This is because the derivative of the switch function is 0, and thus training error cannot be back-propagated through the switch when training networks. This means ReLU has a very limited control over which data is to be passed.

At first, this limitation of ReLU can be easily solved by changing the switch into other functions that have nonzero derivative such as a sigmoid. However, even though the back-propagated error may now be passed through the sigmoid part, this variant still cannot control the switch directly. This is because the error that is back-propagated through the sigmoid of ReLU and the other error that is back-propagated through the identity mapping of ReLU will both update the same convolutional filters in the previous layer. Thus, the filters in the previous layer would be greatly affected by the error back-propagated through the identity mapping. This motivates us to design a novel ReLU-like nonlinear unit, where two different filters are set before the sigmoid and the identity mapping of ReLU respectively. Fig. 1 illustrates this idea.

## 3.2. Proposed selection unit

We propose a novel nonlinear unit, called selection unit (SU), which now has control over which values in the feature maps from the previous convolutional layer can be input to the next layer. In order to use a second convolutional filter before the switch part of ReLU, we propose and utilize a selection module (SM): a cascade connection of one ReLU, a 1×1 convolution and a sigmoid in a row. Thus, the proposed SU is a multiplication of two modules: an identity mapping and an SM. Contrary to the switch in ReLU, SM is able to optimize whole selection control as training error can be back-propagated through itself, which will update the 1×1 convolutional filter to optimize which data is to be passed to the next layer.

## 3.3. Proposed network architecture

By incorporating SU, we propose a 22-layered deep network for SR (SelNet). Fig. 2 shows the network architecture of our proposed SelNet. Our proposed SU is inserted between every two adjacent convolutional layers. For better convergence in somewhat deep network architecture, we also utilize improved residual units using identity mappings [35], where the (n-2)-th feature map after convolution is simply added to the n-th feature map and forwarded to the (n+1)-th layer. Additionally, a technique for learning the residual between HR and a bicubic-interpolated image as in VDSR [28] is further incorporated to ensure faster convergence and better PSNR performance. An LR image is given to our network as input, and a sub-pixel layer [29] is added to the end of the network to convert a multi-channeled LR-sized image into an HR-sized output. In doing so, our network becomes quadratically faster than other conventional networks where bicubic-interpolated images are used as input.

In addition, instead of using gradient-hard clipping as in VDSR [28], we newly propose gradient switching for faster convergence in training. Gradient switching is a harsher version of gradient clipping, where positive gradients are mapped to a predefined threshold θ regardless of its magnitude, and negative ones to -θ. Experiments show that our gradient switching ensures continuous and faster learning even for very small back-propagated error, compared to other network counterparts with or without gradient-hard clipping.

Fig. 3 shows the performance curve of a toy network with SU and its ReLU counterpart. The basic architecture for the both networks are the same, and the two networks have 6 convolutional layers. Note this network is a toy example using SU, and is not our final network structure. As shown, our network with SU outperforms its ReLU counterpart.

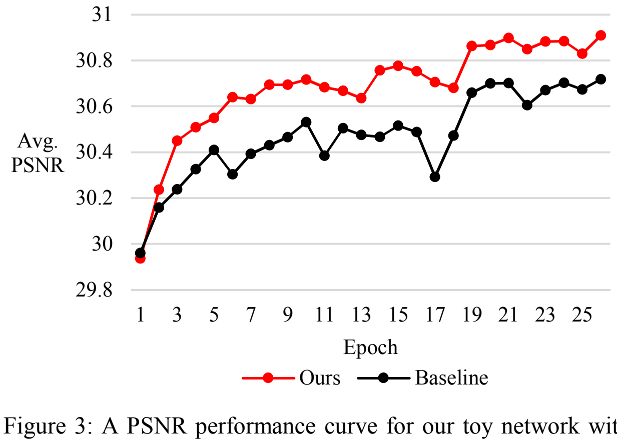


Figure 3: A PSNR performance curve for our toy network with SU and a baseline network with ReLU. The basic architecture for the both networks are the same.

# 4. Experiment result

## 4.1. Experiment setting

For training images, we used 800 high-quality images from the NTIRE2017 Challenge training dataset [37] for HR images. These training images are divided into 120×120-sized RGB subimages without overlapping for any scaling factors. LR training subimages are obtained by down-scaling HR subimages using bicubic interpolation. We do not use any data augmentation such as rotation. As a result, 162,946 LR-HR subimage pairs are used for training. Batch size is set to 32, learning rate is set to $10^{-1}$, weight decay is set to $10^{-5}$, and the number of epoch is set to 50. The network is learned using our gradient switching for faster and better convergence, and $\theta$ is set to $10^{-4}$.

We tested our SelNet on popular benchmark datasets including Set5, Set14 and BSD100 [33]. A down-scaled 3-channeled RGB LR image is used as input to our network. A 3-channeled residual image obtained from the network is added to a bicubic-interpolated image to finally construct an HR RGB image.

For comparison with other conventional SR methods, we follow the popular framework which is frequently used in most of SR methods [3], [27], [28] as follows. SR is applied to the Y-channel of LR inputs, while color components are simply enlarged using bicubic interpolation. PSNR and SSIM [32] is measured on Y-channels of HR images. Note our SelNet produces RGB HR images, and in order to measure PSNR on Y-channel, they are converted to YCbCr.

Our proposed SelNet was implemented using Matconvnet beta23 [36], which is a deep learning toolbox for Matlab, using GPU Nvidia Titan X Pascal. Training time for a scale factor of 2 is 30 hours, 16 hours for a scale factor of 3, and 10 hours for 4. Similar trend is observed for testing time. This is because the size of LR input of a larger scaling factor will always be quadratically smaller for the fixed HR image size. Likely, the size of feature maps in the network is smaller for a larger scaling factor, thus reducing

| Scale | Metric | Bicubic | APLUS [3] | SRCNN [27] | VDSR [28] | SelNet |
|---|---|---|---|---|---|---|
| 2 | PSNR | 33.68 | 36.55 | 36.66 | 37.53 | **37.89** |
| | SSIM | 0.9304 | 0.9544 | 0.9547 | 0.9587 | **0.9598** |
| | Time | 0.01 | 1.0 | 4.9 | 0.13 | 0.03 |
| 3 | PSNR | 30.40 | 32.59 | 32.75 | 33.66 | **34.27** |
| | SSIM | 0.8687 | 0.9088 | 0.9095 | 0.9213 | **0.9257** |
| | Time | 0.01 | 0.7 | 4.9 | 0.13 | 0.03 |
| 4 | PSNR | 28.43 | 30.30 | 30.49 | 31.35 | **32.00** |
| | SSIM | 0.8109 | 0.8603 | 0.8634 | 0.8838 | **0.8931** |
| | Time | 0.01 | 0.5 | 5.1 | 0.12 | 0.02 |

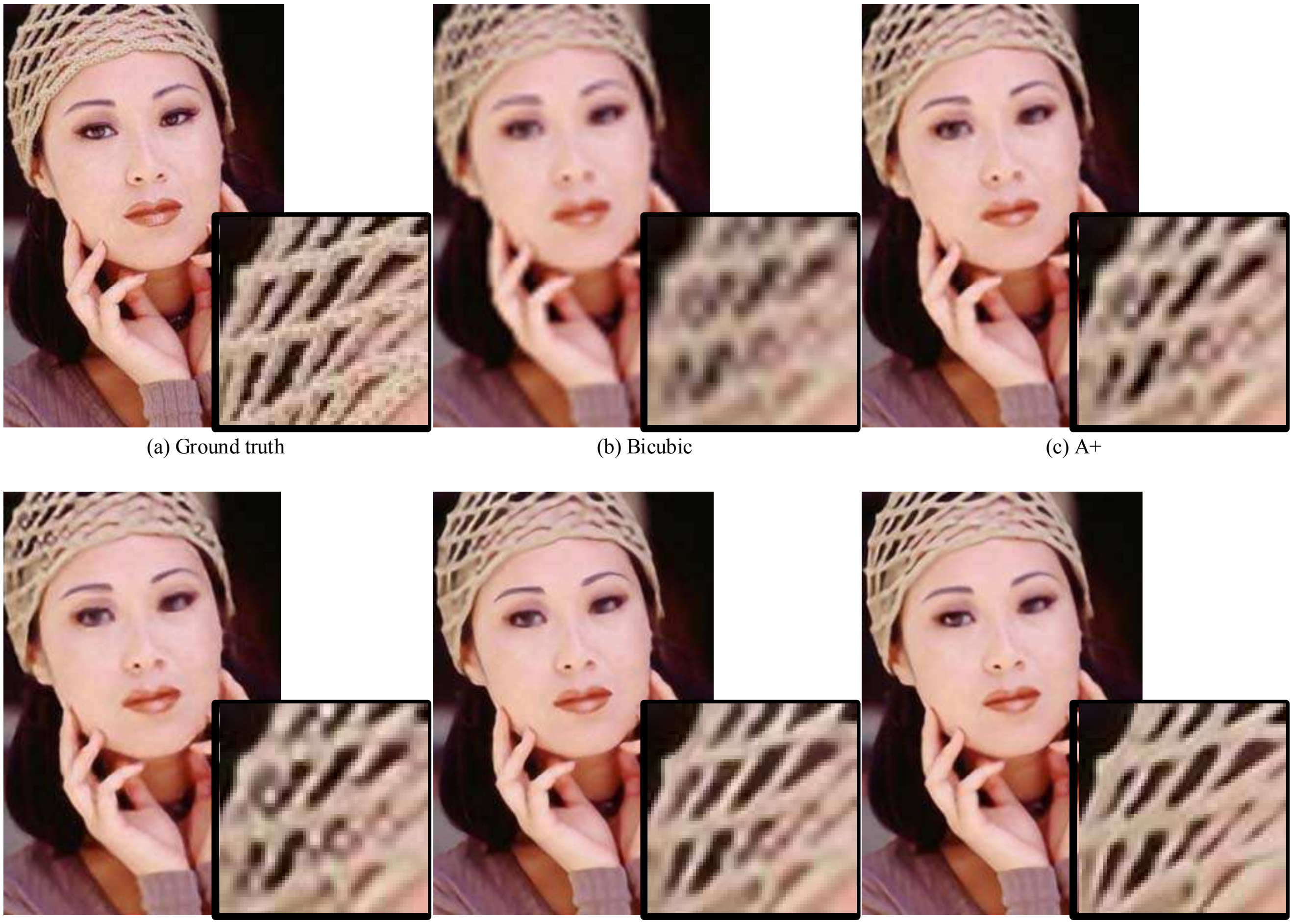
Table 1: Average performance comparison for various SR methods using the **Set5** test set. Time is recorded in seconds. The highest scores are in **red bold**.

| Scale | Metric | Bicubic | APLUS [3] | SRCNN [27] | VDSR [28] | SelNet |
|---|---|---|---|---|---|---|
| 2 | PSNR | 30.24 | 32.27 | 32.45 | 33.03 | **33.61** |
| | SSIM | 0.8691 | 0.9056 | 0.9072 | 0.9124 | **0.9160** |
| | Time | 0.01 | 2.3 | 9.6 | 0.25 | 0.04 |
| 3 | PSNR | 27.55 | 29.12 | 29.30 | 29.77 | **30.30** |
| | SSIM | 0.7741 | 0.8188 | 0.8219 | 0.8314 | **0.8399** |
| | Time | 0.01 | 1.3 | 9.6 | 0.26 | 0.03 |
| 4 | PSNR | 26.01 | 27.31 | 27.50 | 28.01 | **28.49** |
| | SSIM | 0.7023 | 0.7491 | 0.7517 | 0.7674 | **0.7783** |
| | Time | 0.01 | 1.0 | 9.6 | 0.25 | 0.03 |

Table 2: Average performance comparison for various SR methods using the **Set14** test set.

| Scale | Metric | Bicubic | APLUS [3] | SRCNN [27] | VDSR [28] | SelNet |
|---|---|---|---|---|---|---|
| 2 | PSNR | 29.57 | 31.21 | 31.36 | 31.90 | **32.08** |
| | SSIM | 0.8436 | 0.8863 | 0.8884 | 0.8960 | **0.8984** |
| | Time | 0.01 | 1.6 | 6.3 | 0.16 | 0.03 |
| 3 | PSNR | 27.21 | 28.30 | 28.41 | 28.82 | **28.97** |
| | SSIM | 0.7389 | 0.7835 | 0.7867 | 0.7976 | **0.8025** |
| | Time | 0.01 | 0.9 | 6.4 | 0.21 | 0.02 |
| 4 | PSNR | 25.96 | 26.82 | 26.90 | 27.29 | **27.44** |
| | SSIM | 0.6678 | 0.7088 | 0.7107 | 0.7251 | **0.7325** |
| | Time | 0.01 | 0.7 | 6.3 | 0.21 | 0.02 |

Table 3: Average performance comparison for various SR methods using the **B100** test set.


(a) Ground truth

(b) Bicubic

(c) A+

(d) SRCNN

(e) VDSR

(f) SelNet

Figure 4: Reconstructed HR images of *woman* using various SR methods for a scale factor of 4.

overall computations. Note that while the time of other SR methods were measured on CPU, VDSR [28] reported of using GPU Nvidia Titan Z.

We compared our proposed SelNet with the following SR methods: bicubic interpolation, A+ [3], SRCNN [27], VDSR [28]. For A+ and SRCNN, we utilized open Matlab source codes that are publically available.

### 4.2. Results and discussion

Tables 1-3 show the average PSNR and SSIM values of reconstructed HR images using various SR methods, with their computation times in seconds, for Set5, Set14 and B100 test sets. As shown in the tables, our proposed SelNet outperforms all other SR methods for all scale factors an d for all test datasets, even with much lower computational time.

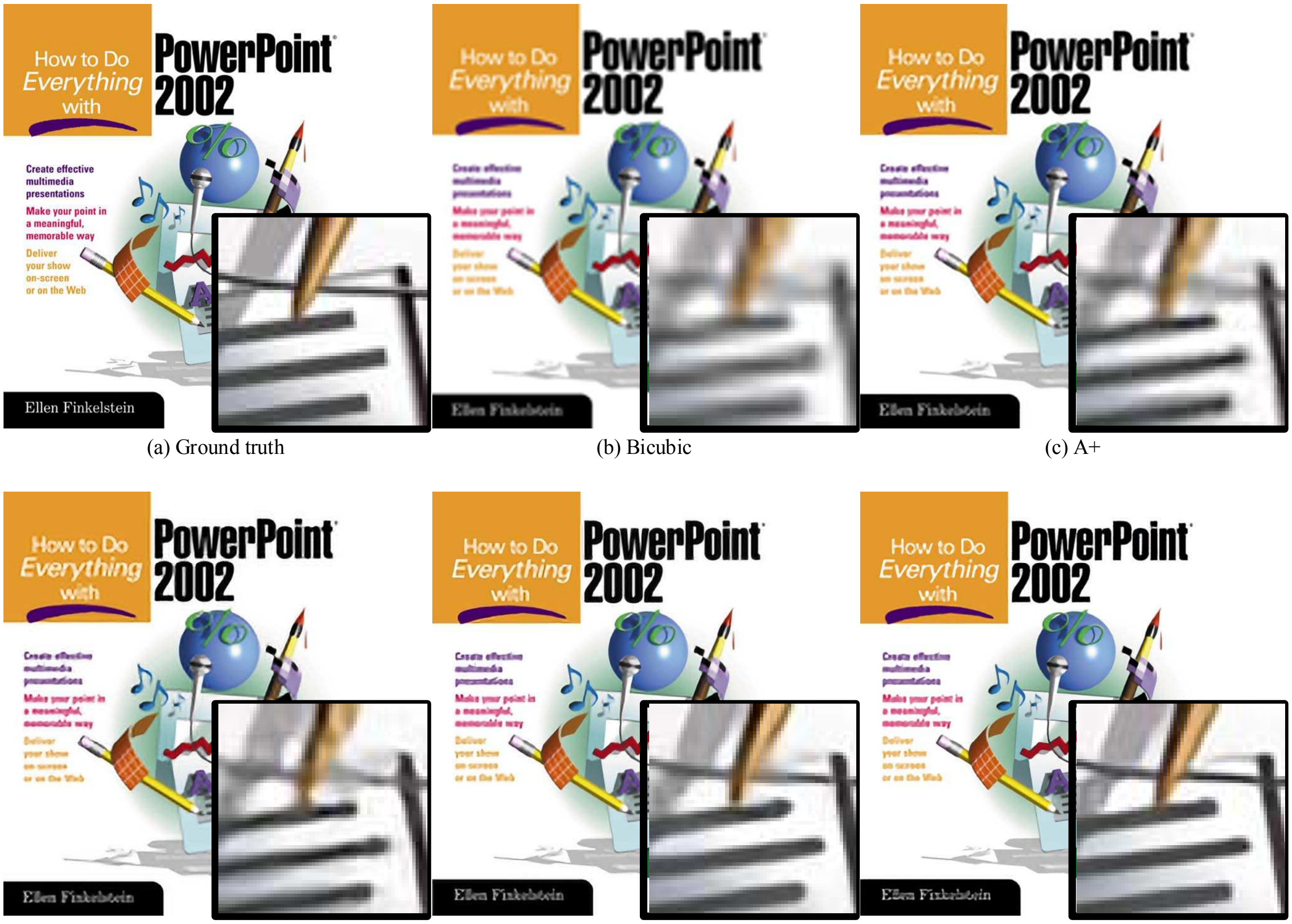
Fig. 4 shows the reconstructed HR images of *woman* using various SR methods for a scale factor of 4. As shown, our SelNet is able to separate hat strings, where other SR methods have difficulty. Fig. 5 shows the reconstructed HR images of *ppt3* using various SR methods for a scale factor of 4. Similarly, our SelNet reconstructs a sharper and clearer HR image, where a pencil and a microphone string can clearly be discerned.

## 5. Conclusion

By re-interpreting ReLU as a combination of an identity mapping and a switch, we proposed a novel selection unit (SU), which is a multiplication of an identity mapping and a sigmoid-based selection module. In doing so, our SU is capable of handling more nonlinearity compared to conventional ReLU. Furthermore, our SU-based deep SR network (SelNet) outperforms its ReLU counterparts and state-of-the-art deep learning SR methods.

## Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funde d by the Korea government (MSIP) (Project No.: 2017-0-0 0419, Project Title: Intelligent High Realistic Visual Proce ssing for Smart Broadcasting Media).


How to Do
Everything
with
PowerPoint
2002
Create effective multimedia presentations
Make your point in a meaningful, memorable way
Deliver your show on-screen or on the Web
Ellen Finkelstein

(a) Ground truth

(b) Bicubic

(c) A+

(d) SRCNN

(e) VDSR

(f) SelNet

Figure 5: Reconstructed HR images of *ppt3* using various SR methods for a scale factor of 4.

## References

[1] R. Keys. Cubic convolution interpolation for digital image processing. IEEE Int. Conf. Acoust. Speech Signal Proc.. 29(6):1153–1160, Dec. 1981.

[2] R. Timofte, V. De and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. IEEE Int. Conf. Comp. Vis., Sydney, Australia, Dec. 2013.

[3] R. Timofte, V. De and L. Van Gool. A+: adjusted anchored neighborhood regression for fast super-resolution. Asian Conf. Comp. Vis., Singapore, Singapore, Nov. 2014.

[4] D. Dai, R. Timofte and L. Van Gool. Jointly optimized regressors for image super-resolution. Eurographics, Zurich, Switzerland, May 2015.

[5] R. Timofte, R. Rothe and L. V. Gool. Seven ways to improve example-based single image super resolution. IEEE Conf. Comp. Vis. Pattern Recog., Nov. 2015.

[6] J.-S. Choi and M. Kim. Super-interpolation with edge-orientation-based mapping kernels for low complex 2x upscaling. IEEE Trans. Image Process.. 25(1):469-483, Dec. 2015.

[7] K. Zhang, X. Gao, D. Tao, and X. Li. Single image super-resolution with non-local means and steering kernel regression. IEEE Trans. Image Process.. 21(11):4544–4556, Nov. 2012.

[8] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. Proc. IEEE Conf. Comp. Vis. Pattern Recog., Jun./Jul. 2004..

[9] B. Li, H. Chang, S. Shan, and X. Chen. Locality preserving constraints for super-resolution with neighbor embedding. IEEE Int. Conf. on Image Proc., Nov. 2009.

[10] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. in Proc. IEEE Conf. Comp. Vis. Pattern Recog., Jun. 2013.

[11] K. Zhang, D. Tao, X. Gao, X. Li and Z. Xiong. Learning multiple linear mappings for efficient single image super-resolution. IEEE Trans. Image Process.. 24(3):846-861, Jan. 2015.

[12] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. IEEE Trans. Pattern Anal. Mach. Intell., 32(6):1127–1133, Jun. 2010.

[13] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. IEEE Trans. Image Process., 19(11):2861–2873, Nov. 2010.

[14] R. Fattal. Image upsampling via imposed edge statistics. ACM Transactions on Graphics. 26(3), Jul. 2007, Art. ID 95.

[15] J.-S. Choi, S.-H. Bae and M. Kim. A no-reference perceptual blurriness metric based fast super-resolution of still pictures using sparse representation. Proc. IS&T/SPIE Elec. Imag. Comp. Imaging XIII, San Francisco, USA, Mar. 2015.

[16] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based superresolution. IEEE Comp. Graphics App.. 22(2):56–65, Mar./Apr. 2002.

[17] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. IEEE Trans. Image Process.. 21(8):3467–3478, Aug. 2012.

[18] C.-Y. Yang, J.-B. Huang, and M.-H. Yang. Exploiting self-similarities for single frame super-resolution. Proc. 10th Asian Conf. Comp. Vis., Nov. 2010.

[19] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. in Proc. IEEE Int. Conf. Comp. Vis., Oct. 2009.

[20] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. IEEE Trans. Image Process.. 23(6):2569-2582, Feb. 2014.

[21] L. Wang, H. Wu and C. Pan. Fast image upsampling via the displacement field. IEEE Trans. Image Process.. 23(12):5123-5135, Sept. 2014.

[22] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. in Proc. IEEE Conf. Comp. Vis. Pattern Recog., Jun. 2008.

[23] G. Freedman and R. Fattal. Image and video upscaling from local self-examples." ACM Trans. Graphics. 30(2), article 12, Apr. 2011.

[24] T. Michaeli and M. Irani. Nonparametric blind super-resolution. IEEE Int. Conf. Comp. Vis., Sydney, Australia, Dec. 2013.

[25] J.-B. Huang, A. Singh and N. Ahuja. Single image super-resolution from transformed self-exemplars. IEEE Conf. Comp. Vis. Pattern Recog., Boston, USA, Jun. 2015.

[26] C. Dong, C. C. Loy, K. He and X. Tang. Learning a deep convolutional network for image super-resolution. European Conference on Computer Vision, Zurich, Switzerland, Sept. 2014.

[27] C. Dong, C. C. Loy, K. He and X. Tang. Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell., 38(2):295-307, June 2015.

[28] J. Kim, J. K. Lee and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. IEEE Conf. Comp. Vis. Pattern Recog., 2016.

[29] W. Shi *et al.* Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. IEEE Conf. Comp. Vis. Pattern Recog., 2016.

[30] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[31] K. He, *et al.* Delving deep into rectifiers: surpassing human-level performance on imagenet classification. IEEE Int. Conf. Comp. Vis., 2015.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process.. 13(4):600–612, Apr. 2004.

[33] D. Martin *et al.* A database of human segmented natural images. In Proc. 8th IEEE Int. Conf. Comp. Vis., 2(1):416-423, July 2001.

[34] J.-S. Choi and M. Kim. Single image super-resolution using global regression based on multiple local linear mappings. IEEE Trans. Image Process.. 26(3):1300-1314, Mar. 2017.

[35] K. He *et al.* Identity mappings in deep residual networks. European Conf. Comp. Vis., 2016.

[36] A. Vedaldi *et al.* Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd ACM international conference on Multimedia. 2015.

[37] R. Timofte, *et al.* New trends in image restoration and enhancement workshop and challenge on image super-resolution. http://www.vision.ee.ethz.ch/ntire17, 2017.