# Seven ways to improve example-based single image super resolution

Radu Timofte
CVL, D-ITET, ETH Zurich
radu.timofte@vision.ee.ethz.ch

Rasmus Rothe
CVL, D-ITET, ETH Zurich
rrothe@vision.ee.ethz.ch

Luc Van Gool
KU Leuven, ETH Zurich
vangool@vision.ee.ethz.ch

## Abstract

*In this paper we present seven techniques that everybody should know to improve example-based single image super resolution (SR): 1) augmentation of data, 2) use of large dictionaries with efficient search structures, 3) cascading, 4) image self-similarities, 5) back projection refinement, 6) enhanced prediction by consistency check, and 7) context reasoning.*

*We validate our seven techniques on standard SR benchmarks (*i.e. *Set5, Set14, B100) and methods (*i.e. *A+, SRCNN, ANR, Zeyde, Yang) and achieve substantial improvements. The techniques are widely applicable and require no changes or only minor adjustments of the SR methods.*

*Moreover, our Improved A+ (IA) method sets new state-of-the-art results outperforming A+ by up to 0.9dB on average PSNR whilst maintaining a low time complexity.*

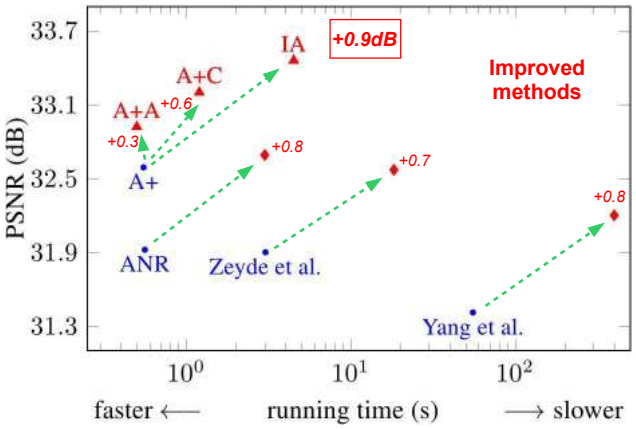


Figure 1. We largely improve (red) over the original example-based single image super-resolution methods (blue), *i.e.* our IA method is 0.9dB better than A+ [31] and 2dB better than Yang *et al.* [40]. Results reported on Set5, $\times 3$. Details in Section 4.1.

## 1. Introduction

Single image super-resolution (SR) aims at reconstructing a high-resolution (HR) image by restoring the high frequencies details from a single low-resolution (LR) image. SR is heavily ill-posed since multiple HR patches could correspond to the same LR image patch. To address this problem, the SR literature proposes interpolation-based methods [29], reconstruction-based methods [3, 15, 25, 39, 41], and learning-based methods [18, 10, 30, 31, 7, 5, 42, 6].

The example-based SR [13] uses prior knowledge under the form of corresponding pairs of LR-HR image patches extracted internally from the input LR image or from external images. Most recent methods fit into this category.

In this paper we present seven ways to improve example-based SR. We apply them to the major recent methods: the Adjusted Anchored Neighborhood Regression (A+) method introduced recently by Timofte *et al.* [31], the prior Anchored Neighborhood Regression (ANR) method by the same authors [30], the efficient K-SVD/OMP method of Zeyde *et al.* [41], the sparse coding method of Yang *et al.* [40], and the convolutional neural network method (SRCNN) of Dong *et al.* [7]. We achieve consistently significant improvements on standard benchmarks. Also, we combine the techniques to derive our Improved A+ (IA) method. Fig. 1 shows a comparison of the large relative improvements when starting from the A+, ANR, Zeyde, or Yang methods on Set5 test images for magnification factor $\times 3$. Zeyde is improved by 0.7dB in Peak Signal to Noise Ratio (PSNR), Yang and ANR by 0.8dB, and A+ by 0.9dB. Also, in Fig. 8 we draw a summary of improvements for A+ in relation to our proposed Improved A+ (IA) method.

The remainder of the paper is structured as follows. First, in Section 2 we describe the framework that we use in all our experiments and briefly review the anchored regression baseline - the A+ method [31]. Then in Section 3 we present the seven ways to improve SR and introduce our Improved A+ (IA) method. In Section 4 we discuss the generality of the proposed techniques and the results, to then draw the conclusions in Section 5.

## 2. General framework

We adopt the framework of [30, 31] for developing our methods and running the experiments. As in those papers, we use 91 training images proposed by [40], and work in

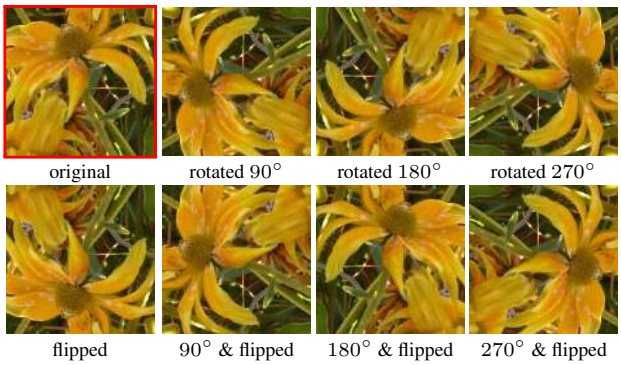

Figure 2. Augmentation of training images by rotation and flip.

the YCbCr color space on the luminance component while the chroma components are bicubically interpolated. For a given magnification factor, these HR images are (bicubically) downscaled to the corresponding LR images. The magnification factor is fixed to $\times 3$ when comparing the 7 techniques. The LR and their corresponding HR images are then used for training example-based super-resolution methods such as A+ [31], ANR [30], or Zeyde [41]. For quantitative (PSNR) and qualitative evaluation 3 datasets Set5, Set14, and B100 are used as in [31]. In the next section we first describe the employed datasets, then the methods we use or compare with, to finally briefly review the A+ [31] baseline method.

### 2.1. Datasets

We use the same standard benchmarks and datasets as used in [31] for introducing A+, and in [40, 41, 30, 24, 7, 27, 8] among others.

**Train91** is a training set of 91 RGB color bitmap images as proposed by Yang *et al*. [40]. Train91 contains mainly small sized flower images. The average image size is only $\sim 6,500$ pixels. Fig. 2 shows one of the training images.

**Set5** is used for reporting results. It contains five popular images: one medium size image ('baby', $512 \times 512$) and four smaller ones ('bird', 'butterfly','head', 'women'). They were used in [2] and adopted under the name 'Set5' in [30].

**Set14** is a larger, more diverse set than Set5. It contains 14 commonly used bitmap images for reporting image processing results. The images in Set14 are larger on average than those in Set5. This selection of 14 images was proposed by Zeyde *et al*. [41].

**B100** is the testing set of 100 images from the Berkeley Segmentation Dataset [20]. The images cover a large variety of real-life scenes and all have the same size of $481 \times 321$ pixels. We use them for testing as in [31].

**L20** is our newly proposed dataset. [1] Since all the above mentioned datasets have images of medium-low resolution, below 0.5m pixels, we decided to created a new dataset, L20, with 20 large high resolution images. The images, as seen in Fig. 10, are diverse in content, and their sizes vary from 3m pixels to up to 29m pixels. We conduct the self-similarity (S) experiments on the L20 dataset as discussed in Section 3.6.

[1] http://www.vision.ee.ethz.ch/~timofter/

### 2.2. Methods

We report results for several representative SR methods.

**Yang** is a method of Yang *et al*. [40] that employs sparse coding and sparse dictionaries for learning a compact representation of the LR-HR priors/training samples and for sharp HR reconstruction results.

**Zeyde** is a method of Zeyde *et al*. [41] that improves the Yang method by efficiently learning dictionaries using K-SVD [1] and employing Orthogonal Matching Pursuit (OMP) for sparse solutions.

**ANR** or Anchored Neighborhood Regression of Timofte *et al*. [30] relaxes the sparse decomposition optimization of patches from Yang and Zeyde to a ridge regression which can be solved offline and stored per each dictionary atom/anchor. This results in large speed benefits.

**A+** of Timofte *et al*. [31] learns the regressors from all the training patches in the local neighborhood of the anchoring point/dictionary atom, and not solely from the anchoring points/dictionary atoms as ANR does. A+ and ANR have the same run-time complexity. See more in Section 2.3.

**SRCNN** is introduced by Dong *et al*. [7], and is based on Convolutional Neural Networks (CNN) [19]. It directly learns to map patches from low to high resolution images.

### 2.3. Anchored regression baseline (A+)

Our main baseline is the efficient Adjusted Anchored Neighborhood Regression (A+) method [31]. The choice is motivated by the low time complexity of A+ both at training and testing and its top performance. A+ and our improved methods share the same features for LR and HR patches and the same dictionary training (K-SVD [1]) as the ANR [30] and Zeyde [41] methods. The LR features are vertical and horizontal gradient responses, PCA projected for 99% energy preservation. The reference LR patch size is $3 \times 3$ while the HR patch is $s^2$ larger, with $s$ the scaling factor.

A+ assumes a partition of the LR space around the dictionary atoms, called anchors. For each anchor $j$ a ridge regressor is trained on the local neighborhood $\mathbf{N}_l$ of fixed size of LR training patches (features). Thus, for each LR input patch $\mathbf{y}$ we minimize

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{N}_l \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2. \quad (1)$$

Then the LR input patch $\mathbf{y}$ is projected to the HR space as

$$\mathbf{x} = \mathbf{N}_h (\mathbf{N}_l^T \mathbf{N}_l + \lambda \mathbf{I})^{-1} \mathbf{N}_l^T \mathbf{y} = \mathbf{P}_j \mathbf{y}, \quad (2)$$

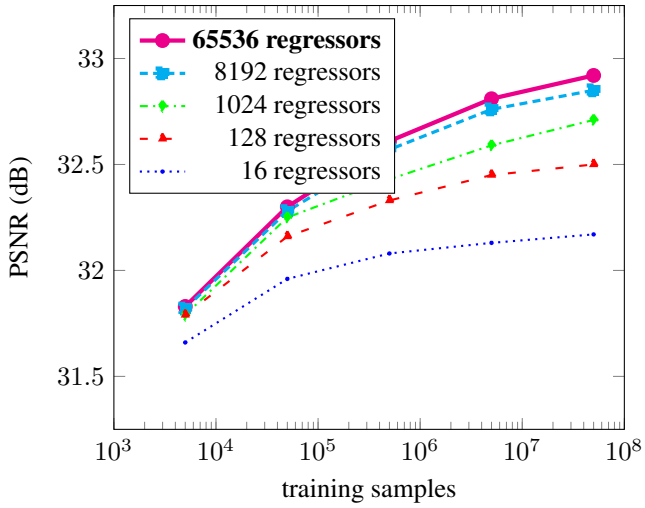


Figure 3. Average PSNR performance of A+ on (Set5, $\times 3$) improves with the number of training samples and regressors

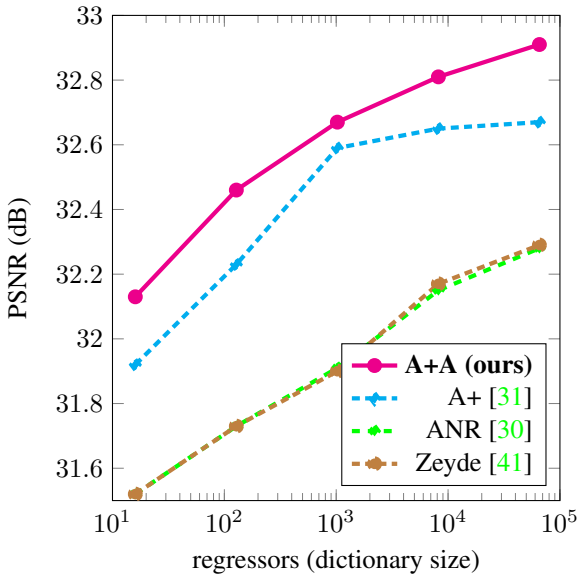


Figure 4. Performance (Set5, $\times 3$) improves with the number of regressors/atoms (dictionary size).

where $\mathbf{N}_h$ are the corresponding HR patches of the LR neighborhood $\mathbf{N}_l$. $\mathbf{P}_j$ is the stored projection matrix for the anchor $j$. The SR process for A+ (and ANR) at test time then becomes a nearest anchor search followed by a matrix multiplication (application of the corresponding stored regressor) for each input LR patch.

## 3. Proposed methods

### 3.1. Augmentation of training data (A)

More training data results in an increase in performance up to a point where exponentially more data is necessary for any further improvement. This has been shown, among others, by Timofte *et al*. in [30, 31] for neighbor embedding methods and anchored regression methods and by Dong *et al*. in [7, 8] for the convolutional neural networks-based methods. Zhu *et al*. [43] assume deformable patches and Huang *et al*. [16] transform self-exemplars.

Around $\sim$0.5 million corresponding patches of $3\times3$ pixels for LR and $9 \times 9$ for HR are extracted from the Train91 images. By scaling the training images in [31] 5 million patches are extracted for A+ and improve the PSNR performance from 32.39dB with 0.5 million to 32.55dB on Set5 and magnification $\times3$. Inspired by the image classification literature [4], we consider also the flipped and rotated versions of the training images/patches. If we rotate the original images by 90°, 180°, 270° and flip them upside-down (see Fig. 2 and [14]), we get 728 images without altered content. Using an interpolation for other rotation angles can corrupt edges and impact the performance.

In Fig. 3 we show how the number of training LR-HR samples affects the PSNR performance of the A+ method on the Set5 images. The performance of A+ with 1024 regressors varies from 31.83dB when trained on 5000 samples to 32.39dB for 0.5 million and 32.71dB when using 50 million training samples. Note that the running time at test stays the same as it does not depend on the number of training samples but on the number of regressors. By **A+A** we mark our setup with A+, 65,536 regressors and 50 million training samples, improving 0.3dB over A+.

### 3.2. Large dictionary and hierarchical search (H)

In general, if the dictionary size (basis of samples/anchoring points) is increased, the performance for sparse coding (such as Zeyde [41] or Yang [40]) or anchored methods (such as ANR [30] or A+ [31]) improves as the learned model generalizes better, as shown in Fig. 4. We show in Fig. 3 on Set5, $\times3$ how the performance of A+ increases when using 16 up to 65,536 regressors for any fixed size pool of training samples. In A+ each regressor is associated with an anchoring point. The anchors quantize the LR feature space. The more anchors are used, the smaller the quantization error gets and the easier is the local regression. On Set5 the PSNR is 32.17dB for 16 regressors, while it reaches 32.92dB for 65536 regressors with 50 million training samples (our **A+A** setup). However, the more regressors (anchors) are used, the slower the method gets. At running time, each LR patch (feature) is linearly matched to all the anchors. The regressor of the closest anchor is applied to reconstruct the HR patch. Obviously, this linear search in $\mathcal{O}(N)$ can be improved. Yet, the LR features are high dimensional (30 after PCA reduction for $\times3$ for A+) and the speedup achievable with data search structures such as kd-trees, forests, or spherical hashing codes

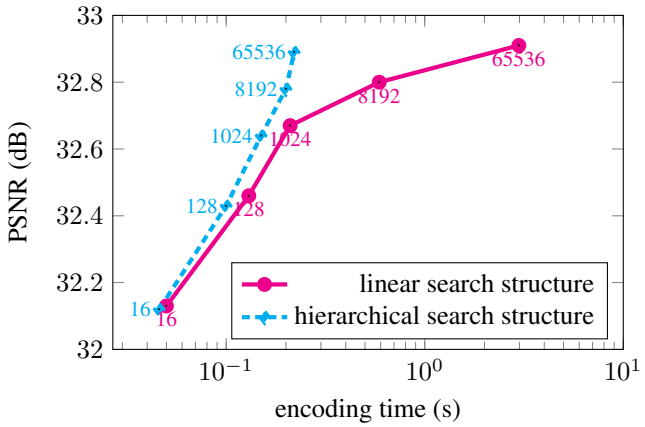


Figure 5. Performance (A+A on Set5, $\times 3$) depends on the number of regressors and the data search structure.

are rather small (3-4 times in [24, 27]).

Instead, we propose a hierarchical search structure in $\mathcal{O}(\sqrt{N})$ with very good empirical precision, that does not change the training procedure of A+. Given $N$ anchors and their $N$ regressors, we cluster them into $\sqrt{N}$ groups using k-means, each with an $l_2$-normalized centroid (on a unit $l_2$-norm hypersphere). To each centroid we assign the $c\sqrt{N}$ most correlated anchors. This results in a 2-layers structure. For each query we linearly search for the most correlated centroid ($1^{st}$ layer) to then linearly search within the anchors assigned to it ($2^{nd}$ layer). $c = 4$ is fixed in all our experiments, so that one anchor potentially can be assigned to more centroids to handle the cluster boundary well.

In Fig. 5 we depict the performance of A+A with and without our hierarchical search structure in relation to the number of trained regressors. The hierarchical structure looses at most only 0.03dB but consistently speeds up above 1,024 regressors. A+A with hierarchical search (H) and 65,536 regressors has a running time comparable to the original A+ with linear search and 1,024 regressors, but is 0.3dB better.

### 3.3. Back projection (B)

Applying an iterative back projection (B) refinement [17] generally improves the PSNR as it makes the HR reconstruction consistent with the LR input and the employed degradation operators such as blur, downscaling, and downsampling. Knowing the degradation operators is a must for the IBP approaches and therefore they need to be estimated [21]. Assuming the degradation operators to be known, (B) improves the PSNR of A+ by up to 0.06dB, depending on the settings as shown in column **A+B** in Table 5, when starting from the A+ results.

In Table 1 we compare the improvements obtained with our iterative back projection (B) refinement when starting from different SR methods. The largest improvement is of 0.59dB when starting from the sparse coding method of Yang *et al.* [40], whereas for A+ it only improves 0.04dB. This behavior can be explained by the fact that the reference A+ is 1.18dB better than the reference Yang method. Therefore, A+'s HR reconstruction is much more consistent with the LR image than Yang's and improving by using (B) is more difficult. The refined Yang result is 0.59dB better than the baseline Yang method but still 0.59dB behind A+ without refinement. Note that generally the explicit form for the degradation operators is unknown and their estimation from prior train examples is imprecise [11, 26], therefore our reported results with (B) refinement are an upper bound for a practical implementation and difficult to reach.

Table 1. Back projection (B) improves the super-resolution PSNR results (Set5, $\times 3$).

| **(B)** | Yang [40] | ANR [30] | Zeyde [41] | SRCNN [7] | A+ [31] | IA (ours) |
|---|---|---|---|---|---|---|
| $\times$ | 31.41 | 31.92 | 31.90 | 32.39 | 32.59 | 33.46 |
| ✓ | 32.00 | 31.99 | 32.04 | 32.52 | 32.63 | 33.51 |
| **Improv.** | **+0.59** | **+0.07** | **+0.14** | **+0.13** | **+0.04** | **+0.05** |

### 3.4. Cascade of core SR method (C)

As the magnification factor is decreased, super-resolution becomes more accurate, since the space of possible HR solutions for each LR patch and thus the ambiguity decreases. Glasner *et al.* [15] use small SR steps to gradually refine the contents up to the desired HR. The errors are usually enlarged by the subsequent steps and the time complexity depends on the number of steps. Instead of super-resolving the LR image in small steps, we can go in one step (stage) and then refine the prediction using the same SR method again adapted to this input. We consider the output of the previous stage as LR image input and as target the HR image for each stage. Thus, we build a cascade of trained models, where each stage brings the prediction closer to the target HR image. The cascades and the layered or recurrent processing are broadly used concepts in vision tasks (*i.e.* object detection [34] and deep learning [4]). The method of Peleg and Elad [23] and the SRCNN method of Dong *et al.* [7] are layered by design and, also, the very recent CSCN method of Wang *et al.* [35]. While the incremental approach has a loose control over the errors, the cascade explicitly minimizes the prediction errors at each stage.

In our cascaded A+, called **A+C**, with 50 million training samples, we keep the same features and settings for all the stages but have models that have been trained per stage. As shown in Fig. 6 and Table 2 the performance improves from 32.92dB after the $1^{st}$ stage of the cascade and saturates at 33.21dB after the $4^{th}$ stage of the cascade. The running time is linear in the number of stages. The same cascading idea of A+ was applied for image demosaicing in [36] with two stages.

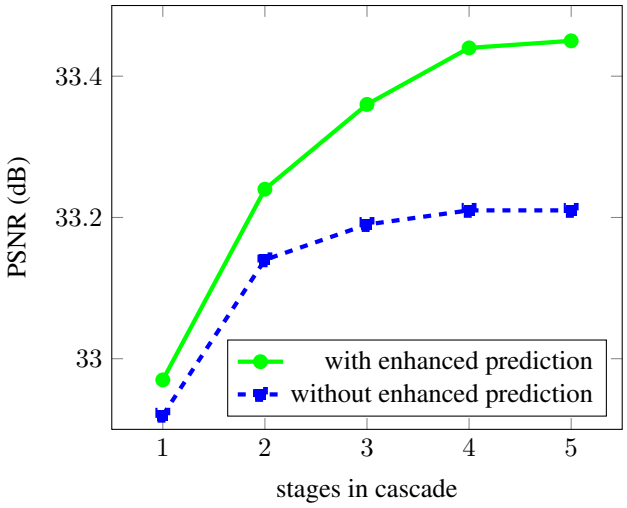


Figure 6. Average PSNR performance of IA improves with the number of cascade stages (Set5, $\times 3$).

Table 2. Cascading (C) and enhanced prediction (E) improve the super-resolution PSNR results (Set5, $\times 3$).

| **Cascade** | **(E)** | ANR [30] | Zeyde [41] | A+ [31] | IA (ours) |
|---|---|---|---|---|---|
| 1 stage | $\times$ | 31.92 | 31.90 | 32.59 | 32.77 |
| 1 stage | ✓ | 32.97 | 31.96 | 32.69 | 32.91 |
| 2 stages | $\times$ | 32.19 | 32.20 | 32.70 | 33.05 |
| 2 stages | ✓ | 32.25 | 32.26 | 32.81 | 33.21 |
| 3 stages | $\times$ | 32.22 | 32.23 | 32.76 | 33.18 |
| 3 stages | ✓ | 32.28 | 32.29 | 32.87 | 33.34 |
| 4 stages | $\times$ | 32.24 | 32.24 | 32.79 | 33.33 |
| 4 stages | ✓ | 32.30 | 32.31 | 32.89 | 33.46 |
| **Improvement** | | **+0.38** | **+0.41** | **+0.30** | **+0.69** |

Table 3. Enhanced prediction (E) improves the super-resolution PSNR results (Set5, $\times 3$).

| **(E)** | Yang [40] | ANR [30] | Zeyde [41] | SRCNN [7] | A+ [31] | IA (ours) |
|---|---|---|---|---|---|---|
| $\times$ | 31.41 | 31.92 | 31.90 | 32.39 | 32.59 | 33.21 |
| ✓ | 31.65 | 31.97 | 31.96 | 32.61 | 32.68 | 33.46 |
| **Improv.** | **+0.24** | **+0.05** | **+0.06** | **+0.22** | **+0.09** | **+0.25** |

### 3.5. Enhanced prediction (E)

In image classification [4] often the prediction for an input image is enhanced by averaging the predictions on a set of transformed images derived from it. The most common transformations include cropping, flipping, and rotations. In SR image rotations and flips should lead to the same HR results at pixel level. Therefore, we apply rotations and flips on the LR image as shown in see Fig. 2 to get a set of 8 LR images, then apply the SR method on each, reverse the transformation on the HR outputs and average for the final HR result.

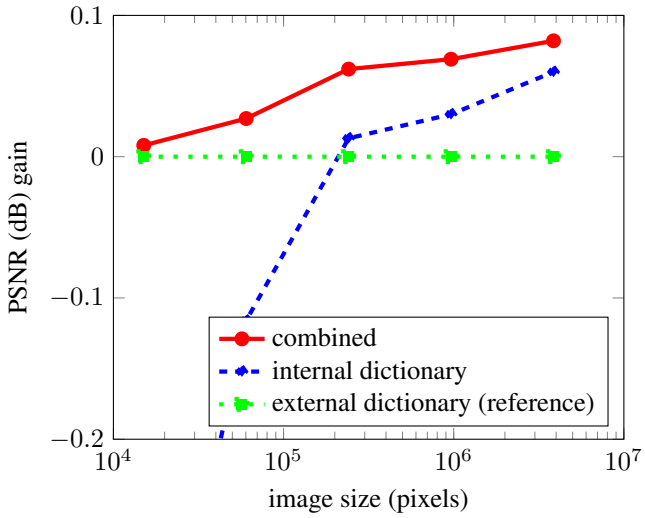


Figure 7. Average PSNR gain comparison of internal dictionary, external dictionary and combined dictionary with respect to the input LR image size (L20, $\times 3$).

On Set5 (see Fig. 6 and Table 2) the enhanced prediction (E) gives a 0.05dB improvement for a single stage and more than 0.24dB when 4 stages are employed in the cascade. The running time is linear in the number of transformations. In Table 3 we report the improvements due to (E) for different SR methods. It varies from +0.05dB for ANR up to +0.25dB for the Yang method.

### 3.6. Self-similarities (S)

The image self-similarities (or patch redundancy) at different image scales can help to discriminate between equally possible HR reconstructions. While we considered external dictionaries, thus priors from LR and HR training images, some advocate internal dictionaries, *i.e.* dictionaries built from the input LR image, matching the image context. Exponents are Glasner *et al.* [15] or Dong *et al.* [9], among others [37, 12, 38, 16]. Extracting and building models adapted to each new input image is expensive. Also, in recent works, and on the standard benchmarks, methods such as SRCNN [7] and A+ [31] based on external dictionaries proved better in both PSNR and running time. Mosseri *et al.* [22] discuss the combination of internal and external dictionaries for image denoising.

We point out that depending on the size of the input LR image and the textural complexity, the internal dictionaries can be better than the external dictionaries for SR. Huang *et al.* [16] with internal dictionaries report better results than A+ (external) for urban HR images with high geometric regularities. We bicubically downscale the L20 images and plot the improvements over an external dictionary in Fig. 7. Above $246,000$ LR pixels the internal dictionary improves over the external one. However, the best results are obtained

Table 4. Reasoning with context (R) improves the super-resolution PSNR results (Set5, $\times 3$).

| (R) Context | ANR [30] | Zeyde [41] | A+(0.5m) [31] | A+ [31] | IA (ours) |
|---|---|---|---|---|---|
| $\times$ | 31.92 | 31.90 | 32.39 | 32.59 | 33.46 |
| ✓ | 32.12 | 32.11 | 32.55 | 32.71 | 33.51 |
| **Improv.** | **+0.20** | **+0.21** | **+0.16** | **+0.12** | **+0.05** |

using both external and internal dictionaries. All the dictionaries were built with a fixed budget of 500000 patches and 1024 regressors.

### 3.7. Reasoning with context (R)

The immediate surrounding of a LR patch adds extra information helping SR. For example, Dong *et al*. [10] train domain specific models, Sun *et al*. [28] hallucinate using context constraints and Timofte *et al*. [32] employ the semantic information. We consider context images centered on each LR patch of size equal with the LR patch size times the scaling factor ($\times 3$). We extract the same features as for the LR patches but in the ($\times 3$) downscaled image and cluster them into 4 groups with 4 centroids. A small number that does not increase the time complexity much but it is still relevant for analyzing the context idea. We keep the standard A+ pipeline with 1024 anchors and 0.5 million training patches ( A+(0.5m) ). To each anchor we assign the closest patches and instead of training one regressor as A+ would, we train 4 context specific regressors. For each context we compute a regressor using the 1024 patches closest to both anchor (distance weighted by 10) and context centroid (distance weighted by 1). For patches of comparable distances to the anchor the distance to the context centroid makes the difference. At testing time, each LR patch is first matched against the anchors and then the regressor of the closest context is used to get the HR output. By reasoning with context we improve from 32.39dB to 32.55dB on Set5, while the running time only slightly increases. In Table 4 we report the improvements achieved using reasoning with context (R) over original SR methods. The (R) derivations were similar to the one explained for the A+ (0.5m) setup.

### 3.8. Improved A+ (IA)

Any combination of the proposed techniques would likely improve over the baseline example-based super-resolution method. If we start from the A+ method, and (A) add augmentation (50 million training samples), increase the number of regressors (to 65536) and (H) use the hierarchical search structure, we achieve 0.33dB improvement over A+ (Set5, $\times 3$) without an increase in running time. Adding reasoning with context (R) slightly increases the running time for a gain of 0.1dB. The cascade (C) allows for another jump in performance, +0.27dB, while the enhanced prediction (E) brings another 0.25dB. The gain brought by

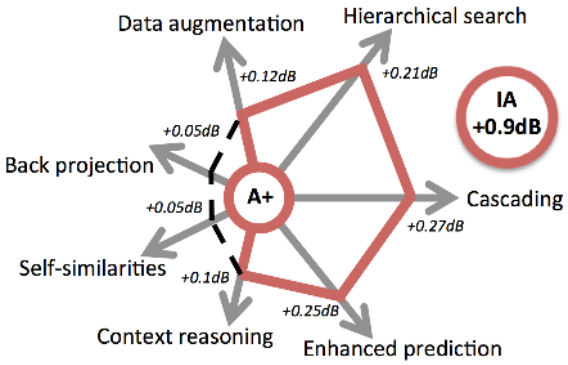


Figure 8. Seven ways to Improve A+. PSNR gains for Set5, $\times 3$.

(C) and (E) comes at the price of increasing the computation time. The full setup, using (A, H, R, C, E) is marked as our proposed Improved A+ (IA) method. The addition of internal dictionaries (S) is possible but undesirable due to the computational cost. Adding IBP (B) to the IA method can further improve the performance by 0.05dB.

The seven ways to improve A+ are summarized in Fig. 8. The Improved A+ (IA) method is 0.9dB better than the baseline A+ method by using 5 techniques (A, H, R, C, E).

Table 5 compares the results with A+ [31], Zeyde [41], and SRCNN [7] on standard benchmarks and for magnifications $\times 2$, $\times 3$, $\times 4$. Figs. 9 and 11 show visual results.

## 4. Discussion

### 4.1. Generality of the seven ways

Our study focused and demonstrated the seven ways to improve SR mainly on the A+ method. As a result, the IA method has been proposed, combining 5 out of 7 ways, namely (A, H, R, C, E). The effect of applying the different techniques is additive, each contributing to the final performance. These techniques are general in the sense that they can be applied to other example-based single image super-resolution methods as well. We demonstrated the techniques on five methods.

In Fig. 1 we report on a running time versus PSNR performance scale the results (Set5, $\times 3$) of the reference methods A+, ANR, Zeyde, and Yang together with the improved results starting from these methods. The A+A method combines A+ with A and H, while the A+C method combines A+ with A, H, and C. A+A and A+C are lighter versions of our IA. For the improved ANR result we combined the A, H, R, B, and E techniques, for the improved Zeyde result we combined A, R, B, and E, while for Yang we combined B and E without retraining the original model.

Note that using combinations of the seven techniques we are able to improve significantly all the methods considered in our study which validates the wide applicability of these techniques. Thus, A+ is improved by 0.9dB in PSNR, Yang

Table 5. Average PSNR on Set5, Set14, and B100 and the improvement (red) of our IA (blue) over **A+ (bold)** method.

| Benchmark | | Bicubic | NE+LLE [30] | Zeyde [41] | ANR [30] | SRCNN [7] | **A+** [31] | A+B (ours) | A+A (ours) | A+C (ours) | **IA (ours)** | **Improvement of IA over A+** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | x2 | 33.66 | 35.77 | 35.78 | 35.83 | 36.34 | **36.55** | 36.60 | 36.89 | 37.26 | **37.39** | **+0.84** |
| **Set5** | x3 | 30.39 | 31.84 | 31.90 | 31.92 | 32.39 | **32.59** | 32.63 | 32.92 | 33.20 | **33.46** | **+0.87** |
| | x4 | 28.42 | 29.61 | 29.69 | 29.69 | 30.09 | **30.29** | 30.33 | 30.58 | 30.86 | **31.10** | **+0.81** |
| | x2 | 30.23 | 31.76 | 31.81 | 31.80 | 32.18 | **32.28** | 32.33 | 32.48 | 32.73 | **32.87** | **+0.59** |
| **Set14** | x3 | 27.54 | 28.60 | 28.67 | 28.65 | 29.00 | **29.13** | 29.16 | 29.33 | 29.51 | **29.69** | **+0.56** |
| | x4 | 26.00 | 26.81 | 26.88 | 26.85 | 27.20 | **27.32** | 27.35 | 27.54 | 27.74 | **27.88** | **+0.56** |
| | x2 | 29.56 | 30.41 | 30.40 | 30.44 | 30.71 | **31.26** | 31.28 | 31.38 | 31.57 | **31.79** | **+0.53** |
| **B100** | x3 | 27.21 | 27.85 | 27.87 | 27.89 | 28.10 | **28.30** | 28.32 | 28.44 | 28.55 | **28.76** | **+0.46** |
| | x4 | 25.96 | 26.47 | 26.51 | 26.51 | 26.66 | **26.84** | 26.86 | 26.96 | 27.07 | **27.25** | **+0.41** |

and ANR by 0.8dB and Zeyde by 0.7dB.

### 4.2. Benchmark results

All the experiments until now used Set5 and L20 and magnification factor $\times 3$. In Table 5 we report the average PSNR performance on Set5, Set14, and B100, and for magnification factors $\times 2$, $\times 3$, and $\times 4$ of our methods in comparison with the baseline A+ [31], ANR [30], Zeyde [41], and SRCNN [7] methods. Also we report the result of the bicubic interpolation and the one for the Neighbor Embedding with Locally Linear Embedding (NE+LLE) method of Chang *et al*. [3] as adapted and implemented in [30]. All the methods used the same Train91 dataset for training. For reporting improved results also for magnification factors $\times 2$ and $\times 4$, we keep the same parameters/settings as used for the case of magnification $\times 3$ for our A+B, A+A, A+C, and IA methods. A+B is provided for reference as the degradation operators usually are not known and are difficult to estimate in practice. A+B just slightly improves over A+. A+A improves 0.13dB up to 0.34dB over A+ while preserving the running time. A+C further improves at the price of running time, using a cascade with 3 stages. IA improves 0.4dB up to 0.9dB over the A+ results, and significantly more over SRCNN, Zeyde, and ANR. [2]

### 4.3. Qualitative assessment

For qualitatively assessing the IA performance we depict in Fig. 11 several cropped images for magnification factors $\times 3$ and $\times 4$. Generally IA restores more sharp details with fewer artifacts than the A+ and Zeyde methods do. For example, the clarity and sharpness of the HR reconstruction for the text image visibly improves from the Zeyde to A+ and then to our IA result. The same for other face features such as chin, mouth or eyes in the image from the second row of Fig. 9.

In Fig. 11 we show image results for magnification $\times 4$ on Set14 for our IA method in comparison with the bicubic, Zeyde, ANR, and A+ methods.

[2]For B100, we report PSNR before saving the images. Prior works [31, 33] report lower PSNR values after lossy JPEG compression.

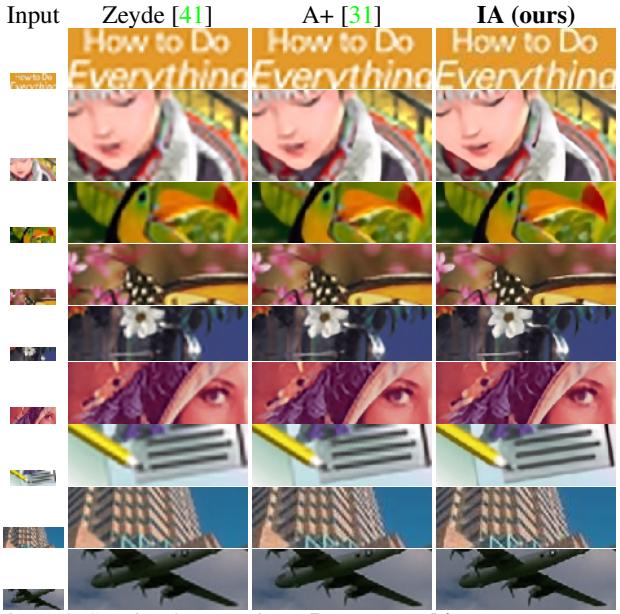

Figure 9. SR visual comparison. **Best zoomed in on screen.**

The supplementary material contains more per image PSNR results and HR outputs for qualitative assessment.

## 5. Conclusion

We proposed seven ways to effectively improve the performance of example-based super-resolution. Combined, we obtain a new highly efficient method, called Improved A+ (IA), based on the anchored regressors idea of A+. Non-invasive techniques such as augmentation of the training data, enhanced prediction by consistency checks, context reasoning, or iterative back projection lead to a significant boost in PSNR performance without significant increases in running time. Our hierarchical organization of the anchors in the IA method allows us to handle orders of magnitude more regressors than the original A+ at the same running time. Another technique, often overlooked, is the cascaded application of the core super-resolution method towards HR restoration. Using the image self-similarities or the context is shown also to improve PSNR. On standard benchmarks IA improves 0.4dB up to 0.9dB over state-of-

Figure 10. L20 dataset. 20 high resolution large images.

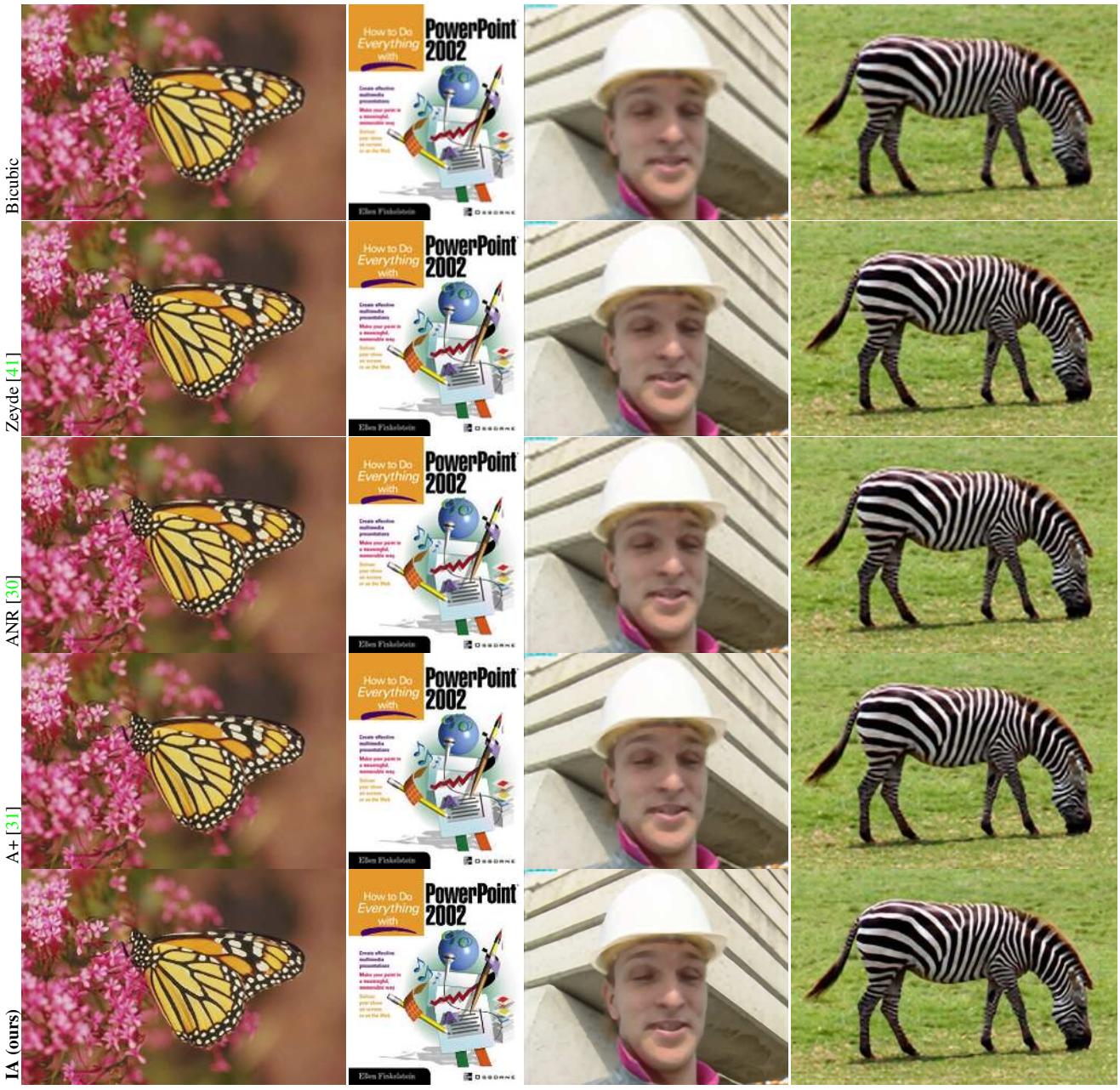


Figure 11. SR visual results for $\times 4$. Images from Set14. **Best zoomed in on screen.**

the-art methods such as A+ [31] and SRCNN [7]. While we demonstrated the large improvements mainly on the A+ framework, and several other methods (ANR, Yang, Zeyde, SRCNN), we strongly believe that the proposed techniques provide similar benefits for other example-based super-resolution methods. The proposed techniques are generic and require no changes to the core baseline method.

**Acknowledgement.** This work was supported by the European Research Council project *VarCity (#273940)*.

## References


[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11), November 2006. 2

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 2

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. *CVPR*, 2004. 1, 7

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 3, 4, 5

[5] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, chapter Deep Network Cascade for Image Super-resolution, pages 49–64. Springer International Publishing, Cham, 2014. 1

[6] D. Dai, R. Timofte, and L. Van Gool. Jointly optimized regressors for image super-resolution. *Computer Graphics Forum*, 34(2):95–104, 2015. 1

[7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[8] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 2, 3

[9] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *TIP*, 22(4):1620–1630, 2013. 5

[10] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *TIP*, 20(7):1838–1857, 2011. 1, 6

[11] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 4

[12] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Trans. Graph.*, 30(2):12:1–12:11, Apr. 2011. 5

[13] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002. 1

[14] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47. 3

[15] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 1, 4, 5

[16] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. June 2015. 3, 5

[17] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP*, 53(3):231–239, 1991. 4

[18] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1127–1133, June 2010. 1

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 2

[21] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *ICCV*, 2013. 4

[22] I. Mosseri, M. Zontak, and M. Irani. Combining the power of internal and external denoising. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–9, April 2013. 5

[23] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. *Image Processing, IEEE Transactions on*, 23(6):2569–2582, June 2014. 4

[24] E. Pérez-Pellitero, J. Salvador, I. Torres-Xirau, J. Ruiz-Hidalgo, and B. Rosenhahn. Fast super-resolution via dense local training and inverse regressor search. In *ACCV*. 2014. 2, 4

[25] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the nonlocal-means to super-resolution reconstruction. *Image Processing, IEEE Transactions on*, 18(1):36–51, 2009. 1

[26] G. Riegler, S. Schulter, M. Ruther, and H. Bischof. Conditioned regression models for non-blind single image super-resolution. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 4

[27] S. Schulter, C. Leistner, and H. Bischof. Fast and accurate image upscaling with super-resolution forests. In *CVPR*, pages 3791–3799, 2015. 2, 4

[28] J. Sun, J. Zhu, and M. F. Tappen. Context-constrained hallucination for image super-resolution. In *CVPR*, 2010. 6

[29] P. Thévenaz, T. Blu, and M. Unser. Image interpolation and resampling. In I. Bankman, editor, *Handbook of Medical Imaging, Processing and Analysis*, pages 393–420. Academic Press, 2000. 1

[30] R. Timofte, V. De Smet, and L. Van Gool. Anchored neighborhood regression for fast example-based super resolution. In *ICCV*, 2013. 1, 2, 3, 4, 5, 6, 7, 8

[31] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014. 1, 2, 3, 4, 5, 6, 7, 8

[32] R. Timofte, V. De Smet, and L. Van Gool. Semantic super-resolution: When and where is it useful? *Computer Vision and Image Understanding*, 142:1 – 12, 2016. 6

[33] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. *CoRR*, abs/1511.02228, 2015. 7

[34] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 4

[35] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 4

[36] J. Wu, R. Timofte, and L. Van Gool. Efficient regression priors for post-processing demosaiced images. In *ICIP*, 2015. 4

[37] C.-Y. Yang, J.-B. Huang, and M.-H. Yang. Exploiting self-similarities for single frame super-resolution. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *Computer Vision ACCV 2010*, volume 6494 of *Lecture Notes in Computer Science*, pages 497–510. Springer Berlin Heidelberg, 2011. 5

[38] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1059–1066, June 2013. 5

[39] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Process.*, 19(11):2861–2873, 2010. 1

[40] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008. 1, 2, 3, 4, 5

[41] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730, 2012. 1, 2, 3, 4, 5, 6, 7, 8

[42] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong. Learning multiple linear mappings for efficient single image super-resolution. *Image Processing, IEEE Transactions on*, 24(3):846–861, March 2015. 1

[43] Y. Zhu, Y. Zhang, and A. Yuille. Single image super-resolution using deformable patches. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2917–2924, June 2014. 3