Accelerating Super-Resolution for 4K Upscaling

Eduardo Perez-Pellitero*‡, Jordi Salvador*, Javier Ruiz-Hidalgo† and Bodo Rosenhahn‡
*Technicolor R&I Hannover
†Image Processing Group, UPC
‡TNT, Hannover University

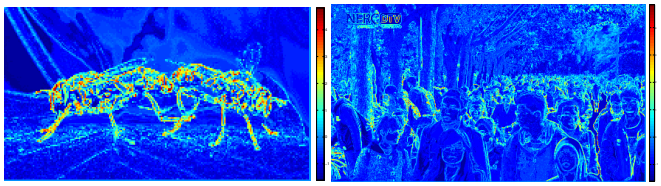
***Abstract*—This paper presents a fast Super-Resolution (SR) algorithm based on a selective patch processing. Motivated by the observation that some regions of images are smooth and unfocused and can be properly upscaled with fast interpolation methods, we locally estimate the probability of performing a degradation-free upscaling. Our proposed framework explores the usage of supervised machine learning techniques and tackles the problem using binary boosted tree classifiers. The applied upscaler is chosen based on the obtained probabilities: (1) A fast upscaler (e.g. bicubic interpolation) for those regions which are smooth or (2) a linear regression SR algorithm for those which are ill-posed. The proposed strategy accelerates SR by only processing the regions which benefit from it, thus not compromising quality. Furthermore all the algorithms composing the pipeline are naturally parallelizable and further speed-ups could be obtained.**

I. Introduction

The desire for higher resolutions has been constantly present in digital imaging, producing a fast succession of de facto standards with increasing resolutions. Nyquist for unidimensional signals and Petersen–Middleton for multidimensional signals (e.g. images) [1] theorems point out that, ideally, there is a maximum frequency we are able to reconstruct for each given lattice used in the sampling process. Violating this constraint would incur into aliasing, thus having to filter out those non-compliant frequencies.

While increasing resolution enables a wider bandwidth, capturing devices do not ensure acquiring a signal that exploits it. Two of the most common approaches to improve resolution in the capture side are: (a) Increasing the size of the sensor and (b) increasing the density of photodetectors. Although both of them will yield a greater pixel count, for the same lens conditions in the case of (a) the larger size of the sensor results in a wider field of view, which can be bypassed applying a certain magnification. This will result in shallower depth of field, and therefore, more content of the image unfocused and blurry. In industry, 4K cameras using 35mm *full-frame* sensors are a good example of this first approach. For (b), the amount of sensed light per pixel decreases, having to deal with a noisier output or correcting it with a larger lens aperture, resulting again in a shallower depth of field. In addition, when dealing with moving scenes (e.g. video sequences) usually a certain motion blur is introduced by the exposure time of the camera.

Super-resolution (SR) algorithms have improved the quality of upscaled images compared to the early interpolation-based methods (e.g. bicubic, Lanczos). We refer the reader to [2] for a current state of the art overview of SR methods. However, SR quality improvements come at a higher computational cost, which limits SR usage in fast applications. Relating this SR higher computational cost to the previously exposed optical and technological limitations in the capture side, whenever those unfocused and blurry areas are upscaled with SR there is little or no benefit. Such areas are oversampled enough to be well-posed for the common fast interpolation methods.



(a) Leibniz dataset (b) SJTU dataset

Fig. 1: Example of how the output of the classifier α is distributed in two images from two different datasets. Higher values of α (red colors) indicate higher likelihood of degradation appearing when moving to different scales, whereas low values (blue color) indicate the likelihood of realizing a degradation-free scaling.

In this paper we introduce an algorithm that tackles this problem by learning how to recognize which regions of an image do benefit from more complex SR methods and which regions can be upscaled with common fast interpolation methods without compromising quality. We propose a fast, probabilistic machine learning framework to efficiently select the areas of the image to apply SR. Exploiting the widely used example-based prior [3], [4], we train a system able to recognize which patches of an image can be properly upscaled with a fast upscaler, modeled as a certain probability obtained in testing time, as shown in Figure 1. The proposed framework is flexible and allows to adapt the training for different applications and scenarios. It is also fast, applicable to any SR technique and easily parallelizable.

II. Proposed method

Let C denote a continuous-space image with limited-bandwidth spectrum $\mathcal{F}(C)$, from which we obtain a discrete high resolution image X whose discrete Fourier transform $\mathcal{F}(X)$ is analogous to $\mathcal{F}(C)$. Let a given image Y denote a degraded version of X which can be modeled as a downsampled and low-pass filtered image $Y = \downarrow (X * h_s)$. There are several fac-

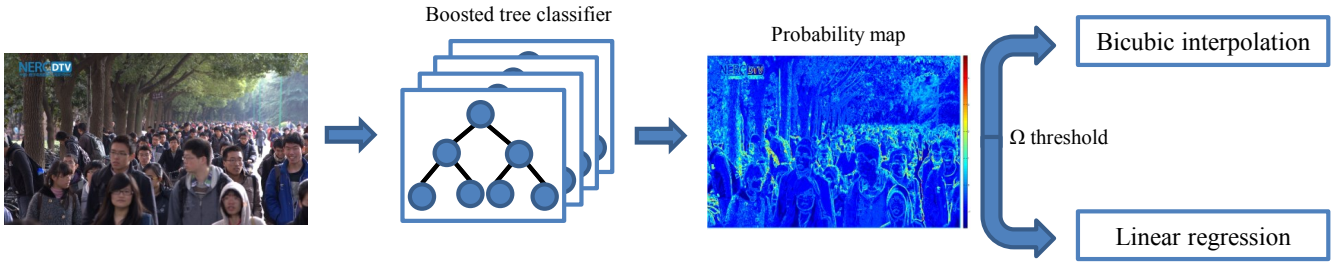


Fig. 2: Overview of the proposed method. For each patch in the testing image, a probability of potential degradation is estimated through a boosted tree classifier. Depending on this probability, a fast bicubic upscaler or a linear regressor is applied.

tors undermining the proper recovering of X from Y, the most remarkable being the shrinkage of spectral support due to the new sampling lattice, but also the attenuation of frequencies close to the Nyquist limit due to the non-ideal shape of h_s and the upscaling method $\upsilon(Y)$ used at the reconstruction stage. Outside the ideal mathematical framework, these factors are flexible, changing, and most importantly, unknown. Estimating them can be challenging since the observed image Y has already suffered the loss of information.

This paper tackles this problem by relying in learning to blindly detect (i.e. without the source image X) degradation in Y. By training the system with real-world examples where a good reconstruction is possible, we learn the sampling limits of our whole degradation-sampling-reconstruction pipeline $\tilde{X} = \upsilon(\downarrow (X * h_s))$.

A. Boosted Tree Classifier

Among the diverse machine learning approaches, we address the problem as a binary classification one, i.e. we must detect whether degradation is present or not. We find in recent object-detection literature the growing use of boosting algorithms when speed is an important concern [5]. This is a convenient choice since they have few parameters to tune and offer a simple and fast scheme while still providing state-of-the-art classification performance. Boosting algorithms create an ensemble of several weak learners (i.e. learners which are slightly better than random guessing) which are trained using weighted samples to focus on difficult examples and have a weighted classifier vote. From within the boosting algorithm family, we selected a modified Adaboost (see details in [6], [7]) with depth-2 decision trees as weak learners.

Let lower case letters (e.g. x, y) denote patches extracted from images in high case letters (e.g. X, Y). The training algorithm takes the training pairs $\{(y_j, x_j)\} = (y_1, x_1), \ldots, (y_m, x_m)$ as input. In order to transform the training pairs $\{(y_j, x_j)\}$ into labeled training instances $\{(y_j, \Lambda_j)\}$ where $\Lambda_j = \{-1, \ +1\}$, a comparison with a certain threshold ε is performed:

$$\tilde{X} = \upsilon(Y)$$
$$\Lambda_j = \begin{cases} -1, & \|x_j - \tilde{x_j}\|_2^2 \leq \varepsilon \\ 1, & \|x_j - \tilde{x_j}\|_2^2 > \varepsilon \end{cases} \quad (1)$$

The Adaboost classifier can be trained at the input scale (i.e. the example pairs $\{(y_j, \Lambda_j)\}$) or at the upper scale (i.e. the upscaled pairs $\{(\tilde{x}_j, \Lambda_j)\}$):

Input scale: By training the Adaboost classifier with the labeled input scale patches, we are able to work with a smaller patch size L_p and a lower number of image pixels. The main drawback of this approach is that subpixel shifts might occur when converting coordinates and patch sizes across scales.

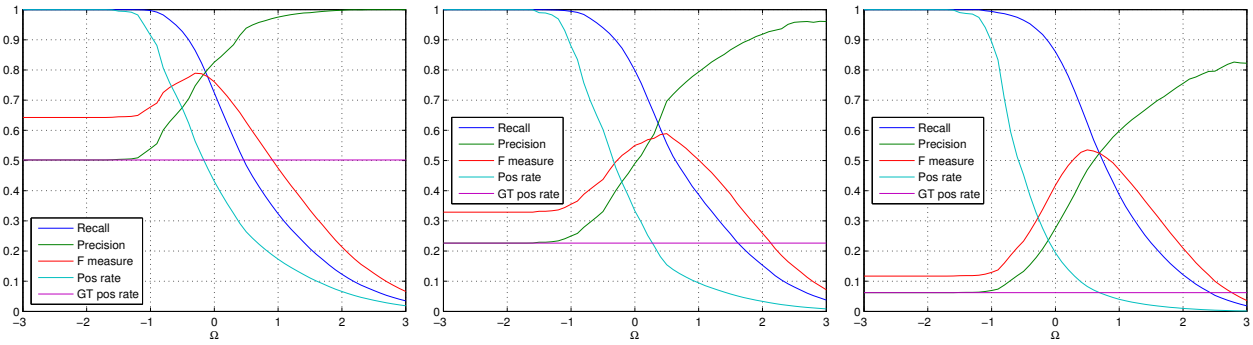
Upper scale: On the other hand, if the Adaboost classifier is trained with upscaled patches (e.g. patches upscaled via bicubic interpolation) the training takes place in the same scale grid. This also means that, for a given upscaling factor s, the dimensionality of the classification problem increases to $L_p^2 s^2$.

The choice of scale should be consistent with the application requirements, e.g. some applications as the cross-scale self-similarity SR usually work in the upscaled grid.

In testing time, the probability distribution $p(\Lambda \mid y)$ is obtained thanks to the leaf predictors of the boosted trees, and the decision rule is defined as the logarithmic ratio $\alpha = \log \frac{p(\Lambda=1|y)}{p(\Lambda=-1|y)} > \Omega$, where α values above threshold Ω indicate degradation in the scaling process and values below Ω indicate a proper scaling output.

B. Upscaling and Super-Resolution

A simple and fast upscaler υ_1 is applied then to regions of the image which are smooth and well-posed and the more computationally costly SR upscaler υ_2 only for textured regions where there is significant expected improvement. Figure 2 shows an overview of the complete process. As for the choice of the computationally inexpensive upscaler υ_1, we find in the bicubic kernel upscaler a fast and vastly used solution. Regarding the more costly SR upscaler υ_2, we select a linear regression-based SR approach [8] . In such approaches, the objective of training a given regressor R is to obtain a certain mapping function from LR to HR patches. From a more general perspective, low-resolution (LR) patches form an input manifold M of dimension m and high-resolution (HR) patches form a target manifold N of dimension n. Formally, for training pairs $(x_i, \ y_i)$ with $x_i \in M$ and $y_i \in N$, we would like to infer a mapping $\Psi : M \subseteq \mathbb{R}^m \rightarrow N \subseteq \mathbb{R}^n$. We obtain the regressor by solving the following minimization:



(a) Kodak dataset (b) Leibniz dataset (c) SJTU 4k dataset

Fig. 3: Recall, precision, F measure and positive rate averages for different decision rule thresholds Ω tested in three different datasets. The recall measure can be interpreted as a quality measure while the positive rate can be interpreted as a computational cost measurement (e.g. number of patches processed by the SR upscaler). The desired trade-off between these two measurements can be selected with the decision rule Ω. Note also that the presence of sharp content, represented by the GT positive rate, decreases fastly for increasing resolutions.

$$\min_{\beta} \|y - D_l\beta\|_2^2 + \lambda \|\beta\|_2 , \tag{2}$$

where D_l is the LR dictionary selected for the training, from which a HR counterpart D_h is known. This minimization problem has a closed-form solution $\beta = (D_l^T D_l + \lambda I)^{-1} D_l^T y$, which we can furthermore apply to the HR dictionary D_h to obtain a matrix-shaped linear regressor

$$R = D_h(D_l^T D_l + \lambda I)^{-1} D_l^T , \tag{3}$$

which in testing time only needs to be multiplied by every input patch as $x = R\,y$.

III. Results

In this section we present experimental results of the classification performance in order to support the viability of the chosen Adaboost approach. We also illustrate the benefits of our proposed SR algorithm, providing running times and objective quality measures of our method and comparing it to the well-known Sparse SR work of Yang. et al. [9]. Additionally, we present the usage of the framework as a tool to better understand and assess the performance of SR algorithms by plotting a map of the probabilities $p_i(\Lambda \mid y)$ in the upscaled image.

A. Cross-validation

We train the Adaboost classifier with 256 depth-2 trees with 60 images from the Berkeley Segmentation Dataset [10], using the full resolution as the original references $\{X\}$, applying a bicubic downscale by $s = 2$ to obtain the degraded versions $\{Y\}$ and generating the training labels with a threshold ε corresponding to a PSNR of 34 dB, which we obtained experimentally, i.e. the reconstruction quality is still good but some degradation can be observed. The patch size is $L_p = 5$ and the training and testing is performed on the input scale with the example pairs $\{(y_j, \Lambda_j)\}$, setting as positive training examples those which can not be properly reconstructed using bicubic as the upscaling method $\upsilon(Y)$. All the experiments were run on a Intel Xeon E5-1620 (10M Cache, 3.60 GHz), with a MATLAB and C++ implementation [7], where only the Adaboost code and the matrix multiplication are parallel. The training process with about 2M patches takes 50s. In order to support the generalization of the method we extract ground-truth (GT) labels from three different datasets: (a) Kodak dataset, (b) 9 sharp images obtained from the internet with a resolution of 1920x1080 (referred as Leibniz dataset) and (c) 10 images from the *Rush Hour* sequence of SJTU 4k dataset [11].

	Recall	Pos rate	GT pos. rate	time (s)
Kodak	0.71	0.43	0.50	0.02
Leibniz	0.80	0.33	0.23	0.04
SJTU	0.85	0.19	0.07	0.13

TABLE I: Cross validation and execution time. See Section III-A for more details.

Figure 3 shows the difference in performance for different probability ratio thresholds. The recall measure can be interpreted as a quality measure (i.e. how many degraded patches are actually classified as such) while the positive rate can be interpreted as a computational cost measurement. The logarithmic probability ratio threshold Ω should be selected application-wise as a trade-off between these two measures. It is important to remark that as stated in the introduction, when increasing the image resolution, the presence of sharp content decreases (less than 10% in Figure 3c) and higher recalls are obtained at little computational cost. Setting the probability ratio threshold to $\Omega = 0$ is a good trade-off across the three datasets, specially for our goal application which is 4k upscaling. Table I shows recall, positive rates and execution

	Yang et al.		Proposed algorithm			
	PSNR (dB)	time (s)	PSNR (dB)	time (s)	% of v_2 (SR) patches	speed up
SJTU 4k dataset	40.93	7085.6	42.50	5.98	19.07%	×1184.88

TABLE II: Running times and PSNR for SJTU dataset (*2K* to *4K* upconversion).

Fig. 4: Results of an interactive $\times 2$ zooming for a 4K image from the SJTU dataset (shown in the left red rectangle), accompanied by the upscaler decision mask (shown in the right red rectangle, black is bicubic and white is SR). Better viewed zoomed in.

times for this configuration.

B. Super-Resolution performance

In this section we test the performance of the proposed SR algorithm, using the presented selective patch processing stage based in boosting trees, linear regression and bicubic interpolation. We compare the advantages of using our selective scheme against the well-known sparse SR method of Yang et al. [9], which is a computationally intense method. We select the SJTU 4k dataset in an attempt to better represent a realistic scenario of high-resolution images. An upscaling step of $s = 2$ is performed, showing the current problem of upscaling legacy cinema content from 2K to 4K.

In Table II we show the PSNR and execution times.

C. Probability-ratio map

In Figure 1 we show the color mapping of the classifier output α for two images for a visual qualitative evaluation. The images have been processed in the same conditions stated in Section III-A and they are consistent and directly comparable with the results shown in Figure 3. Smooth and blurry regions appear with low values of α, while the sharp edges and textured areas show higher values.

This probability-ratio maps could be accompanied with PSNR values of an upscaled image in order to better understand where the method is performing better or the most challenging regions of the image and how the algorithm performs at such regions. In a similar way, a new metric can be obtained by weighting in a pixel basis the PSNR by the correspondent probability value, shifted to a range of $(0, 1)$ for this purpose.

IV. Conclusions

We proposed a new resolution assessment framework which learns to blindly discern if a patch will be degraded when converting it to an upper scale. We train binary boosted tree classifiers with examples of patches, evaluating their similarity when moving across scales and labeling them accordingly. By learning from examples, the framework additionally covers degradations which are complex to model analytically. Thanks to boosting classifiers' simple scheme, the classification stage runs in about 0.15s for a 2Mpixels image (e.g. industry standards *FullHD*, *2K*), which makes it suitable for fast applications. We then select between a bicubic upscaler and a linear regression-based SR algorithm. The experimental results reflect the viability of the selected machine learning approach and the flexibility obtained by modifying the threshold Ω, which controls a trade-off between speed and quality. The obtained PSNRs are competitive, and the measured times are orders of magnitudes faster, enabling interactive zoom-in for 2K images.

References

[1] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces," *Information and control*, vol. 5, pp. 279–323, 1962.

[2] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *ECCV 2014*, ser. Lecture Notes in Computer Science, 2014, vol. 8692.

[3] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *Computer Graphics and Applications, IEEE*, vol. 22, no. 2, pp. 56–65, 2002.

[4] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression." 2013, pp. 1059–1066.

[5] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," *CVPR*, 2012.

[6] C. Zhang and P. Viola, "Multiple-instance pruning for learning efficient cascade detectors," in *Advances in Neural Information Processing Systems*, 2008, pp. 1681–1688.

[7] P. Dollár, "Piotr's Image and Video Matlab Toolbox (PMT)," `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`.

[8] R. Timofte, V. D. Smet, and L. V. Goool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings IEEE International Conference on Computer Vision*, 2013, pp. 1920–1927.

[9] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Processing*, vol. 19, no. 11, 2010.

[10] 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 *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 2001, pp. 416–423.

[11] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The SJTU 4k video sequence dataset," *QoMEX*, 2013.