Fuzzy-Rule Based Approach for Single Frame Super Resolution

Pulak Purkait and Bhabatosh Chanda
Electronics and Communication Sciences Unit
Indian Statistical Institute
Kolkata, India
Email: {pulak_r, chanda}@isical.ac.in

***Abstract*—We propose a novel high-quality image zooming technique using fuzzy rule based prediction framework. Our approach is based on the fundamental idea that a low-resolution (LR) image patch could be generated from any of the many possible high-resolution (HR) image patches. Therefore it would be natural to assign certain fuzzyness to each of the possibilities of HR patches. We develop a prediction system that learns the LR-HR patch correspondence and also the natural image patch prior from an external database. We do so by collecting a large amount of the LR-HR natural image patch pairs from an existing database, grouping them into different clusters and then generating fuzzy rules to get an efficient mapping from LR patch space to HR patch space. Experimental results show the efficacy of our method over existing state-of-art methods.**

***Index Terms*—Fuzzy Rules, Super Resolution, Image Zooming.**

I. Introduction

The goal of Super Resolution (SR) methods is to generate/reconstruct HR image from multiple or a single input LR images. Accordingly, SR image reconstruction methods may be broadly categorized into two classes: (i) Single-frame SR methods, and (ii) Multi-frame SR methods. In the classical multi-frame SR [1], [2], [3], [4], multiple LR images of the same scene with different sub-pixel shifts are taken as input. A set of constraints is imposed on the unknown HR image along with a regularization/prior term to convert an ill-posed SR problem to a well-posed one, since there exist infinitely many HR images that may result in the same LR image. It has direct application in video SR where multiple frames are available. These methods have been proven to give very good results theoretically, as well as for synthetic images. However, they do not work well most of the cases on real data.

The other class of SR methods generates the HR image from a single LR image or frame and is quite popular these days. These methods are also called 'Example-Based Super-Resolution' [5], [6], [7], [8] or 'Image Hallucination' [9]. In example-based SR, correspondence between LR and HR image patches are learned from a collection of LR and HR image pairs and then the knowledge is applied to a new LR image to reconstruct its most likely HR version. These algorithms are in general based on image edge prior [6], [10] or gradient profile prior [11]. Methods based on learning edge models are also proposed [10], [12], [13]. The goal of these methods is to magnify (zoom) an image while maintaining the sharpness of the edges and the texture details in the image. Glasner *et al*. [7] merge the concepts of both single-frame SR and multi-frame SR for HR image reconstruction from a single-frame. They down-scale the input LR image to further coarser scale and then learn the correspondence between the LR and the HR patch pairs. Another variation of SR algorithms deal with sparse image prior [9], [14]. The effectiveness of such sparsity prior is demonstrated by Yang *et al*. [8] and Purkait *et al*. [4].

Now for natural images different HR patches can result in identical LR patch when blurred and down-sampled. Hence, it would be more effective if we model each of the HR possibilities with different fuzzy membership corresponding to given a LR patch and then to preserve spatial coherence those HR patches are aggregated according to their fuzziness. In this work, We learn these LR-HR mapping from a database of LR-HR patch pairs by some fuzzy rules. We assume that LR patch feature space and HR patch feature space lie on the same manifold and use similar kind of mechanism for learning nonlinear mapping between input and output space as suggested by Pal *et al*. [15] for structure preserving dimensionality reduction. The rest of the paper is organized as follows. In the next section we describe the main intuition behind the proposed algorithm, followed by a detail description. In Section IV, we show some results. Finally, we conclude with some pros and cons of the proposed algorithm and give the direction of future work in Section V.

II. Proposed fuzzy model for Super Resolution

In this section we will describe details of the proposed Super resolution technique. It contains two phase (i) Offline phase: where we generate a rule base system and tune the parameters of the system to have the better performing learned system and (ii) Online phase: where we apply trained rule based system on the patches of LR image to have a HR version of LR ones. In the following subsections, we discuss step-(i) in detail and in the next section we discuss about the step-(ii).

A. Database Generation

We have downloaded a number of sharp natural images from 'flickr' and consider them as HR images to build our database of image patches. Some of those HR images are shown in figure 1. We generate LR versions corresponding to each of these HR images in the following way:

- Apply a Gaussian blur operator on each of these images in the database with a small kernel (size $[3 \times 3]$ and $\sigma = 0.8$).

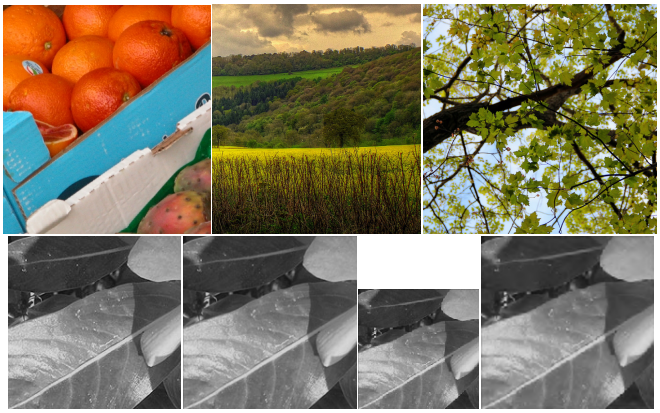


Fig. 1. Top row: Some images from our database of natural images. These color images are mapped into YCbCr space and only the intensity Y plane is stored. Bottom row (left to right): LR image formation process in the order as in the text. Bottom-right image is the LR version of the bottom-left HR image.

- Down-sample each image with resolution factor s(=1.6). This is our LR image.
- Then up-sample the LR image with the same resolution factor by bicubic interpolation. Thus original HR and corresponding LR images are now of same size. Only difference will be the LR one containing blurring artifacts and the ringing noises whereas HR one will be containing sharp edges without any artifacts.

In Figure 1 we show the formation of LR image in the same order as stated above.

The recent works [8], [11], [16] suggest that the derivative features can represent the patch more efficiently than the actual intensities. In this work, we use four 1-D derivative filter as $f_1 = [-1,0,1]$, $f_2 = f_1^T$, $f_3 = [1,0,-2,0,1]$ and $f_4 = f_3^T$. We apply these filters to the all LR images of the database and then divide the HR images and corresponding filtered LR images into overlapping patches of size $(n \times n)$. Each of the HR patch is normalized to mean zero and standard deviation one. It is assumed that a LR and HR patch has same mean. Moreover, in [8], it is shown that the standard deviation of HR patch is appropriately 1.2 times the standard deviation of LR patch. Therefore we incorporate this statistics of natural image into our model and during prediction we get mean and standard deviation of the HR patch from the LR patch itself.

After normalization, LR and HR patch pairs are concatenated into a single vector comprising LR and HR feature patch-pair $x^* = (x;y)$ for learning the fuzzy rules for prediction. For $(n \times n)$ size patch the length of the LR patch feature vector is $4 * n^2$ and that for HR patch vector is n^2.

B. Fuzzy rule learning

Let our database consists of input LR patches $X = x_1, x_2, \ldots, x_N \subset \mathbb{R}^p$ and the corresponding HR patches $Y = y_1, y_2, \ldots, y_N \subset \mathbb{R}^q$. Typically, $p = 4 * n^2$, $q = n^2$ for $n \times n$ patch and N is the number of patches in the database. We concatenate those patch pairs as follows:

$$X^* = \{x_i^* = (x_i; y_i) \in \mathbb{R}^{p+q}, i = 1,2,\ldots,N\}$$

where x_i^* is the i^{th} concatenated patch.

We partition our patch-pair database X^* by k-means clustering into C clusters with cluster centroids

$$V^* = \{v_c^* = (v_c^x; v_c^y) \in \mathbb{R}^{p+q}, c = 1,2,\ldots,C\}$$

Here, we assume that input LR feature space and output HR space lie on the same manifold. Therefore, after clustering if there is a cluster in the input LR patch feature space with centroid v_c^x, then the corresponding points in the output HR patch space are likely to form a cluster around v_c^y. In other words, if there is a good cluster around v_c^* in patch-pair space, then for any patch-pair $(x_i; y_i)$ in that cluster if $\|x_i - v_c^x\|$ is small, $\|y_i - v_c^y\|$ would also be small. i.e. if v_c^x be a rough estimate of an LR patch feature x_i then v_c^y would be a rough estimate of the corresponding HR patch y_i. Therefore, for each cluster a rule can be written expressing this concept and combining all those rules we get a locally continuous and smooth predictor that maps input LR patch to approximate HR patch. In figure 2, we have shown some cluster centers (only normalized v_c^ys are displayed)

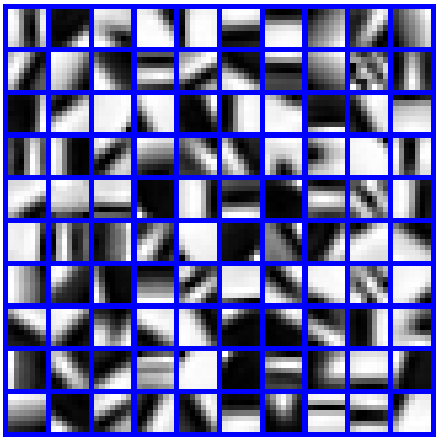


Fig. 2. This represents the cluster center of LR-HR patch after clustering. Only HR patches are displayed.

C. Fuzzy rule Identification scheme

We translate the c^{th} cluster into a rule (following Takagi-Sugeno (TS) model) of the form

$$R_c^{TS}: \text{ If x is CLOSE to } v_c^x \text{ then } y = u_c(x, v_c^y)$$

The behavior of the function $u_c(.)$ from input LR patch feature space to HR patch space lies in the neighborhood of v_c^y. This set of C rules forms an initial rule base for data prediction. Now as $x \in \mathbb{R}^p$, the antecedent part of R_c^{TS} can be viewed as a conjunction of p atomic clauses:

$$\text{If x is CLOSE to } v_c^x \approx \begin{pmatrix} \text{If } x_1 \text{ is CLOSE to } v_{c1}^x \\ \text{AND If } x_1 \text{ is CLOSE to } v_{c1}^x \\ \ldots \text{AND If } x_p \text{ is CLOSE to } v_{cp}^x \end{pmatrix} \quad (1)$$

Moreover, since $y \in \mathbb{R}^q$, the c^{th} rule R_c^{TS} corresponds to q different rules as

R_{cj}^{TS}: If x is CLOSE to v_c^x then $y_j = u_{cj}(x, v_{cj}^y), j = 1,2,\ldots,q.$

where we use the following nonlinear mapping:

$$u_{cj} = d_{cj0}.v_{cj}^y + d_{cj1}.x_1 + \ldots + d_{cjp}.x_p, c = 1,2,\ldots,C;\ j = 1,2,\ldots,q$$

where d_{cji} $(i = 0,1,\ldots,p)$ are the parameters to be identified. Thus the output HR patch pixels are linear combinations of centroids v_{cj}^y and the input LR patch feature vector x. Hence the output is computed by

$$y_j = \frac{\sum_{c=1}^{C} \alpha_c.u_{cj}(x, v_{cj}^y)}{\sum_{c=1}^{C} \alpha_c},\ j = 1,2,\ldots,q. \tag{2}$$

where, α_c is the firing strength of the rule R_c^{TS}, computed using a T-norm. Since v_{cj}^y is given (a constant), without loss of generality we denote $d_{cj0}.v_{cj}^y$ as d_{cj0} and choose it efficiently along with the other parameters d_{cji} $(i = 1,2,\ldots,p)$.

To estimate the appropriate set of consequent parameters d_{cji}, we formulate the problem as a linear least square optimization problem. Let, $\mu_c = \frac{\alpha_c}{\sum_{c=1}^{C} \alpha_c}$, for i^{th} point we rewrite eq. 2 as

$$y_{ij} = \sum_{c=1}^{C} \mu_{ic}.(d_{cj0} + d_{cj1}.x_{i1} + \ldots + d_{cjp}.x_{ip}) \tag{3}$$

where μ_{ic} is the normalized strength at i^{th} point. We can rewrite the above equation in the following form reshuffling the parameters:

$$y_{ij} = \sum_{c=1}^{C} (d_{cj0}.\mu_{ic} + d_{cj1}.x_{i1}\mu_{ic} + \ldots + d_{cjp}.x_{i_p}.\mu_{ic}). \tag{4}$$

If we fix the parameters of the antecedent membership functions of our initial rule base system, we can easily obtain the rest of prediction parameters:

$$D^j = (d_{1j0}\ d_{2j0}\ \ldots d_{Cj0}\ d_{1j1}\ d_{2j1}\ \ldots\ d_{Cj1}\ \ldots\ d_{1jp}\ d_{2jp}\ \ldots\ d_{Cjp})^T$$

by the least square error method using eq. 4. From our database of patch pairs X^*, we have a known set of N LR-HR patch pairs, $Y = (y_1, y_2, \ldots, y_N)^T$ and $X = (x_1, x_2, \ldots, x_N)^T$. Therefore, eq. 4 gives us a system of linear equations $Y = A * D$ where $D = (D^1 D^2 \ldots D^q)_{(p+1)C \times q}$ and

$$A = \begin{pmatrix} \mu_{11}\ldots\mu_{1C} & \mu_{11}.x_{11}\ldots\mu_{1C}.x_{11} & \ldots & \mu_{11}.x_{1p}\ldots\mu_{1C}.x_{1p} \\ \mu_{21}\ldots\mu_{2C} & \mu_{21}.x_{21}\ldots\mu_{2C}.x_{21} & \ldots & \mu_{21}.x_{21}\ldots\mu_{2C}.x_{2p} \\ \ddots & \ddots & & \ddots \\ \mu_{N1}\ldots\mu_{NC} & \mu_{N1}.x_{N1}\ldots\mu_{NC}.x_{N1} & \ldots & \mu_{N1}.x_{N1}\ldots\mu_{NC}.x_{Np} \end{pmatrix} \tag{5}$$

Here $Y_{N \times q}$ is a matrix of output values and $A_{N \times C(p+1)}$ is a matrix with predefined x_{ij} and corresponding membership values μ_{cj}. The matrix $D_{C(p+1) \times q}$ contains all the parameters to be estimated. The estimate of D minimizing the square error $\|AD - Y\|_2^2$ is given by

$$D = (A^T A)^{-1} A^T Y \tag{6}$$

Now if dimension of $(A^T A)$ is very large (proportional to patch-size and number of cluster), computation of $(A^T A)^{-1}$ require a lot of memory and time consuming. On that scenario we use simply gradient descent for D which minimize the square error $\|AD - Y\|_2^2$. That give the results quite satisfactory.

D. Choice of antecedent membership functions

In order to get the firing strength of the rule base, we need to define the membership function of closeness for each of the clauses: 'x_j is CLOSE to v_{cj}^x' and a T-norm to combine the clauses. Here different choice of antecedent membership functions such as triangular, trapezoidal, and Gaussian are possible and choice of membership function effects the results of our algorithm in its own way though not much. Here we use a trivial membership function, that is, Gaussian function because it is differentiable everywhere and has only two parameters which can be tunable very easily. To be more explicit, 'x_j is CLOSE to v_{cj}^x' is modeled by:

$$\pi_{cj}(x_j; v_{cj}, \sigma_{cj}) = exp(-\frac{(x_j - v_{cj})^2}{\sigma_{cj}^2}).$$

The spread of the membership function σ_{cj} is initialized by the standard deviation of the j^{th} component of the LR feature vector in the training data that are included in the c^{th} cluster. We will further tune these parameters (v_{cj}, σ_{cj}) using the training data X^* as described in the next section.

For the intersection of the q clauses to get the firing strength of c^{th} rule, we may choose any one of the T-norm. The most trivial choice is product or min as the operator for intersection. As min is not differentiable, we could use the softer version of min that is differentiable. An way of defining the softmin is as follows:

$$\alpha_c := softmin(\pi_{c1}, \pi_{c2}, \ldots, \pi_{cp}, t) = \left(\frac{\pi_{c1}^t + \pi_{c2}^t + \ldots \pi_{cp}^t}{p}\right)^{\frac{1}{t}}$$

However, here issue of computational cost is involved. For a typical 7×7 patch, the number of clauses (i.e., dimension of feature vector) is $4 * 7^2 = 196$. Therefore, for one rule we need to compute $(196+1)$ exponentials and there are 100 such rules (for typical choice of $C = 100$). And we need to do this computation for every LR patch during training as well as testing. Which results very slow super resolution technique. Hence it would be more natural to use less computationally expensive product operator defined as:

$$\alpha_c = \Pi_{j=1}^{p} \pi_{cj}$$

Here instead of computing $4 * n^2 + 1$ exponentials, we need to compute only product of p numbers and we do not find any noticeable difference in the output. The main concern here is that product of those numbers (each of them is less than one) results a number close to zero. Then how we can use this fast operator? The answer lies in the defuzzification process of combining the consequents of C rules. As in equation 2, we are normalizing the rule strength $\mu_c = \sum_{c=1}^{C} \alpha_c$ during the combination of rules, the final output is, thus, computed as a

convex combination of the consequents of C rules. Therefore, even if the product is close to zero, during normalization process it would take a moderate value.

E. Further tuning of the parameters

In the section II-C, we estimated of the consequent parameters of the TS model by minimizing the LSE assuming that the antecedent parameters are predefined, we also discussed about the antecedent parameters of the initial rule base system. We take this as an initial choice for the consequent and then use the gradient descent method to further refine all the parameters (including the antecedent membership function) because the objective function minimized by the clustering algorithm and the objective function that a rule base should minimize are not the same. Here, we use an EM (expectation-maximization) algorithm for finding an optimal choice of the antecedent membership parameters (v_{cj}, σ_{cj}) as well as for the consequent parameters D^j. In this hybrid algorithm, first we maximize the antecedent parameters keeping the consequent parameters fixed and in the next stage maximize the consequent parameters with the modified antecedent parameters. We do these two steps iteratively. The stopping criterion could be either the number of iterations be reached upto a predefined number or the global error value (eq. 4) be less than a predefined threshold. Moreover, such a two stage hybrid scheme for further tuning of consequents along with membership parameters is justified because when the membership parameters are altered the LSE estimate of the consequents may not remain optimal. The rule base parameters v_{cj} and σ_{cj} are tuned using gradient descent method to minimize the error $E = \sum_{k=1}^{n} \|(\hat{y})_k - y_k\|^2$ as

$$v_{cj}(t+1) = v_{cj}(t) - \alpha_v \frac{\partial E}{\partial v_{cj}} \tag{7}$$

$$\sigma_{cj}(t+1) = \sigma_{cj}(t) - \alpha_\sigma \frac{\partial E}{\partial \sigma_{cj}} \tag{8}$$

After modifying the antecedent membership parameters $(v_{cj}(t+1), \sigma_{cj}(t+1))$, we recalculate the matrix $A_{(t+1)}$ as in eq. 4, consequently the consequent parameters are updated as:

$$D_{(t+1)} = (A_{(t+1)}^T A_{(t+1)})^{-1} A_{(t+1)}^T Y \tag{9}$$

where t indicates the iteration number. In this hybrid scheme, since the initial antecedent membership are judiciously chosen based on cluster analysis, they are likely to form a reasonably good set of linguistic values. The LSE estimate of consequent parameters keeping the antecedent membership fixed will result in a fairly good rule base system.

III. Up-scaling of the test document

After tuning the antecedent and consequent parameters, our rule base system is ready to do upscaling for test images. We extract all LR feature vectors x corresponding to all overlapping patches of the given input test image as discussed in the section II-A. During extraction of the patches, we also store the mean and variance of each of the patch. Then compute the membership values of each of the patch x corresponding to each of We use the tuned fuzzy rule to generate the corresponding normalized HR patch for each LR patch feature vector by calculating the matrix A as in eq. 2 and multiplying with the trained antecedent parameters to get all the HR patches of the corresponding LR ones. We simply average out the overlapping portions of the HR patches. More detail description and the parameters used in the proposed algorithm are depicted in section IV.

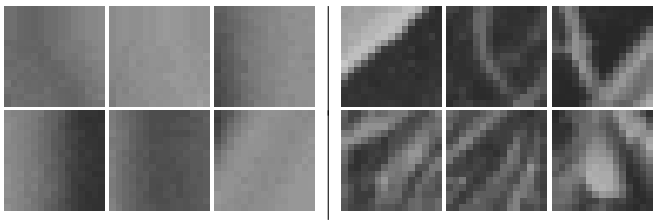


Fig. 3. The image patches on the left portions are the patches where bicubic interpolation and fuzzy rule base prediction gives comparable results and the patches where fuzzy rule base prediction gives a definite better results are displayed on right.

A. Selective patch processing

Natural images typically contain large smooth regions as well as strong discontinuities, such as edges and corners. Although simple interpolation methods for image upscaling, e.g., bilinear or bicubic interpolation, results in noticeable artifacts along the edges and corners, such as ringing, jaggies and blurring effects, however perform reasonably well on smooth regions. Figure 3 illustrates this fact, where we upscale (x2) each image patch of a small image (having ground truth) by bicubic interpolation and fuzzy rule base system respectively for RMSE comparison. The patches on the right part reveal that the fuzzy rule base outperform the bicubic interpolation in while the left part shows the patches where the result of bicubic interpolation is comparable with that of fuzzy rule base. Similar observation has also been reported by Yang *et al*. [8]. Based on the these observations, we may selectively process those highly textured regions using our fuzzy rule base technique and simply apply cheaper bicubic interpolation for the rest smooth regions. This will save a remarkable amount of computation. We compute the roughness of a patch by the variance of the LR patch feature vector x. If its high the corresponding patch probably belong in texture region and if it is low it belongs to smooth region. We choose a threshold judiciously, to choose the appropriate class for the patch and consequently the upscaling method.

IV. Experimental Results

We have implemented our algorithm using MATLAB 7.6 on a Linux OS with 8GB of RAM and a 3.6-GHz Intel processor. We have used mex code for some of the subroutines to speed up the process. It takes usually less than 10 seconds for a typical 200×200 LR image to upscale with resolution factor 4.

A. Detail description of parameters

We have chosen a small up-scaling factor ($s = 1.6$) for the proposed algorithm. HR image with higher up-scaling factor

can be generated by applying the algorithm required number of times. When working with color images, the image is first converted from RGB to YCbCr. Then the SR algorithm is applied on Y (intensity) channel only. For rest of the components (Cb and Cr which are usually low frequency channels and stores the chromatic information only) we use simply bicubic interpolation. The three modified components are then combined and converted back to RGB to get the estimated color SR image displayed.

In patch based technique, it is observed that usually 5×5 to 10 patch size are good choice for processing. In our experiment we have chosen $100,000$ patches of size 7×7 and then cluster them into $C = 100$ cluster. Thus the size of each LR feature vector $4 * 7^2 = 196$ and size of A for test data is $100,000 \times 100 * (196 + 1)$, so increasing number of clusters was not possible for us with our limited resources. However, with 100 clusters the proposed system is good enough to produce the results comparable to the state of the art techniques. For selective patch processing, we use the threshold in such a way that 30% of all the patches are predicted by the proposed fuzzy rule base system. The rest of the patches are considered to be smooth and are simply copied from the result of bicubic interpolation. In Figure 4, we compare our result with two recent techniques Glasner *et al.* [7] and Purkait *et al.* [17]. It is clear from the images that the result of our method contains less artifacts and more sharp boundaries. We also compare our method with the state-of-art techniques and got better or equivalent result. Some results are depicted in figure 5. In these experiments we experience that proposed method is moreover faster than most of the recent algorithms (viz. sparse coding) because in our case we need to compute some membership values and the defuzzification step which require only multiplication of two matrices (A and D). More results of the proposed method along with that of the other methods can be found in http://www.isical.ac.in/~lrg/?q=node/13. We have also **uploaded the source code of the proposed** method and made it publicly available for research purpose.

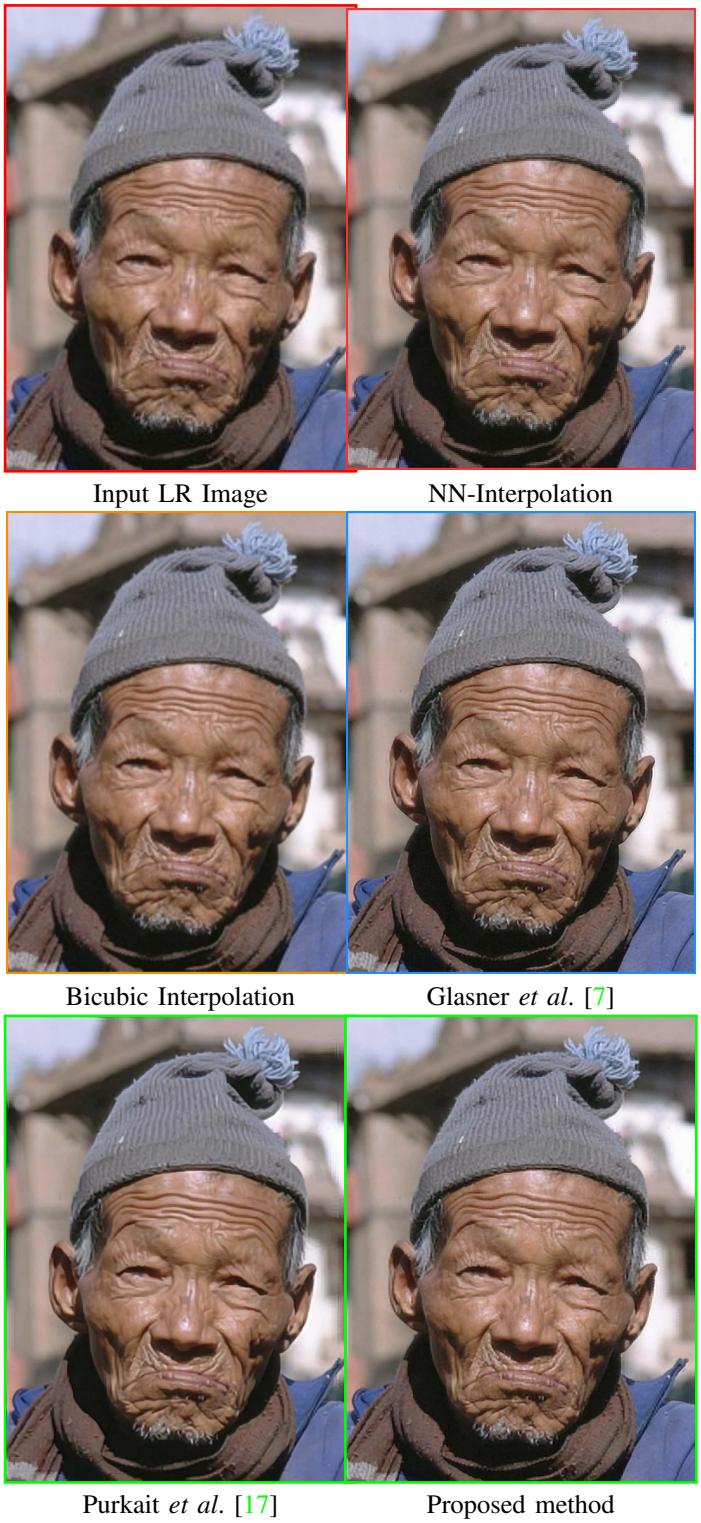
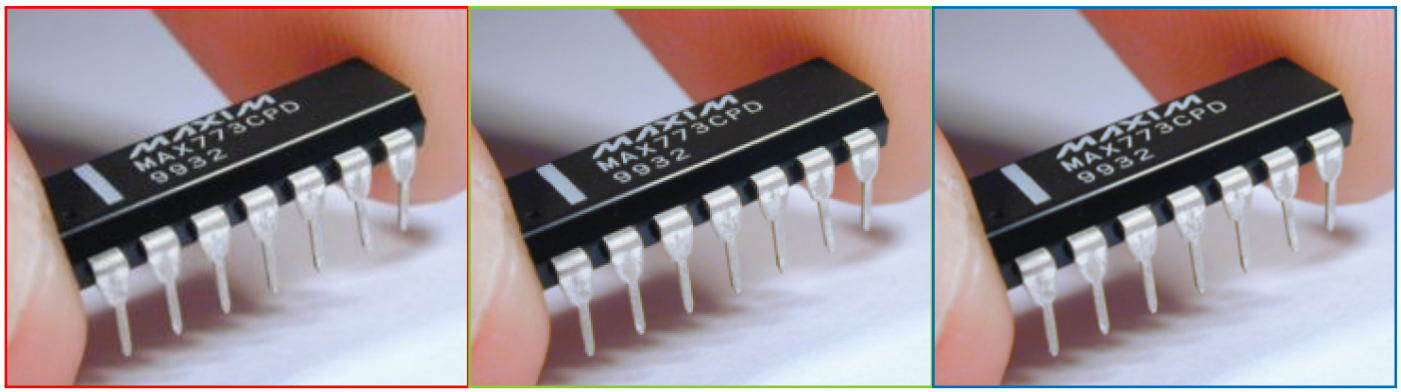


Fig. 4. Comparison results of super resolution for chip image using proposed method over other state-of-arts methods. **For better comparison kindly view the soft version of the paper**.

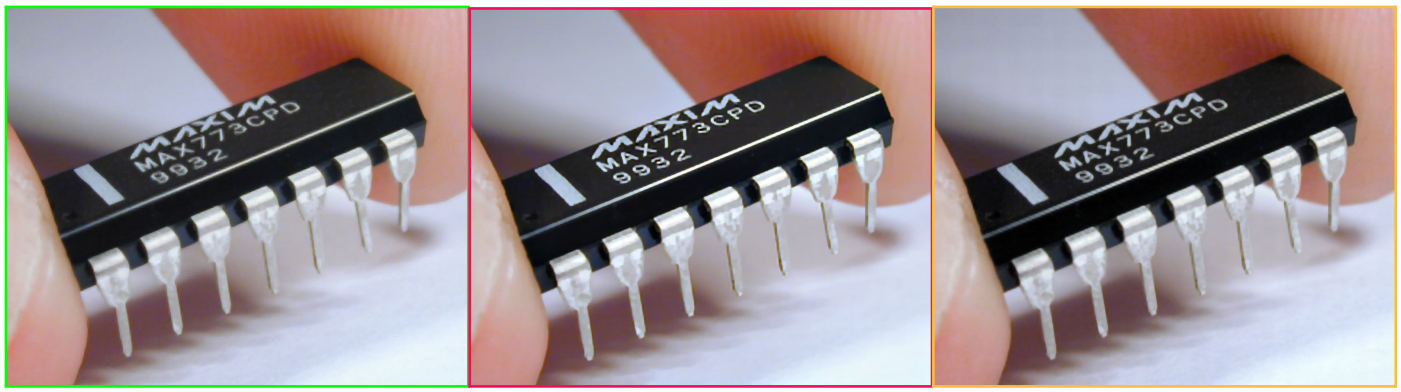
V. Conclusion

In this work, we have proposed a fuzzy rule based scheme for single frame super resolution. We have used the TS model with consequents expressed as a linear combination of the input variables. An initial rule base is extracted using cluster analysis. The consequent parameters of these rules are then estimated using LSE technique. The antecedent as well as the consequent parameters of the rule base thus obtained are further refined using gradient descent. We tested the proposed scheme over some benchmark images for single frame super resolution and obtained better or comparable results compared to very recent techniques. Also proposed method is faster than most of the state-of-the-arts Super Resolution methods.

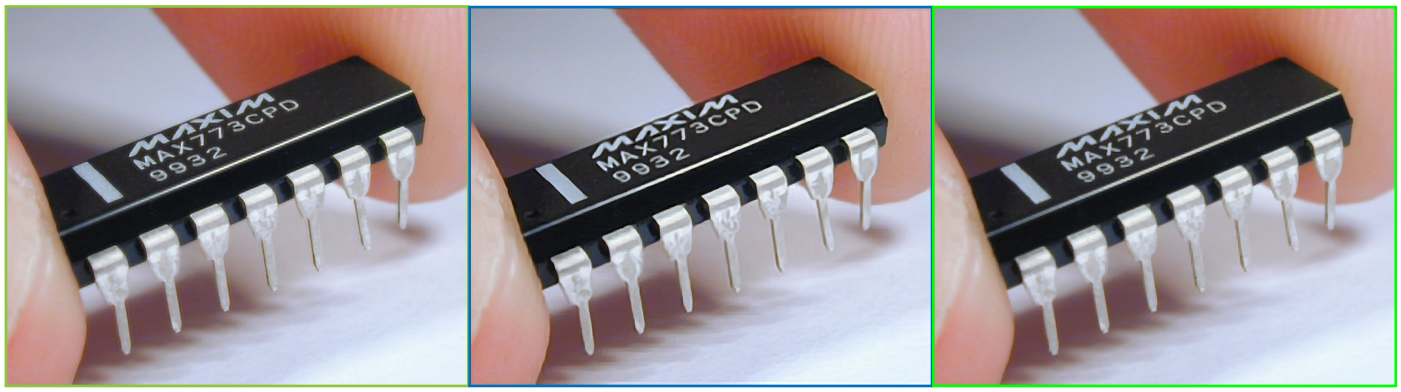
Our method can be directly extended into Video SR technique, where either we can apply SR technique on each and individual frame to get SR video or we can divide the video into overlapping 3D blocks (instead of 2D patches for a single-frame) and use exactly the same algorithm to generate SR video. Modeling explicit nonlinear fuzzy rules which can incorporate inherent correlation between output variables would also be another direction of future work.



Input LR Image | NN-Interpolation | Bicubic Interpolation



Genuine Fractals^{TM} | Sun *et al.* [11] | Fattal *et al.* [12]



Glasner *et al.* [7] | Purkait *et al.* [17] | Proposed method

Fig. 5. Comparison results of super resolution for chip image using proposed method over other state-of-arts methods. It is clear from the images that proposed fuzzy rule based technique can produce comparable results with state-of-art methods. **For better comparison kindly view the soft version of the paper**.

REFERENCES

[1] M. Elad and A. Feuer, "Restoration of a single super-resolution image from several blurred, noisy and under-sampled measured images," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1646–1658, Dec. 1997.

[2] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *The Computer Journal*, vol. 52, no. 2, pp. 15–30, Apr. 2009.

[3] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.

[4] P. Purkait and B. Chanda, "Super resolution image reconstruction through bregman iteration using morphologic regularization," *IEEE Trans. Image Process.*, vol. PP, no. 99, p. 1, 2012.

[5] W. T. Freeman and T. R. Jones, "Example-based super resolution," *IEEE Computer Graphics and Appl.*, vol. 22, no. 2, pp. 56–65, Mar. 2002.

[6] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *The Computer J.*, vol. 50, no. 4, pp. 1–16, Apr. 2007.

[7] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proceedings of ICCV*, Oct. 2009, pp. 349–356.

[8] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[9] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse

regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.

[10] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," in *Proceedings of CVPR*, Jun. 2007, pp. 1–8.

[11] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *Proceedings of CVPR*, Jun. 2008, pp. 1–8.

[12] R. Fattal, "Image upsampling via imposed edge statistics," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 56–65, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276496

[13] H. He and W.-C. Siu, "Single image super-resolution using gaussian process regression," in *Proceedings of CVPR*, june 2011, pp. 449–456.

[14] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 533–549, Jul. 2011.

[15] N. Pal, V. Eluri, and G. Mandal, "Fuzzy logic approaches to structure preserving dimensionality reduction," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 277 –286, jun 2002.

[16] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–10, 2010.

[17] P. Purkait and B. Chanda, "Image upscaling using multiple dictionaries of natural image patches," *Proceedings of The 11th Asian Conference on Computer Vision (ACCV-12)*, november 2012.