

Super-resolution Using Constrained Deep Texture Synthesis

Libin Sun*
Brown University

James Hays†
Georgia Institute of Technology

Abstract

Hallucinating high frequency image details in single image super-resolution is a challenging task. Traditional super-resolution methods tend to produce oversmoothed output images due to the ambiguity in mapping between low and high resolution patches. We build on recent success in deep learning based texture synthesis and show that this rich feature space can facilitate successful transfer and synthesis of high frequency image details to improve the visual quality of super-resolution results on a wide variety of natural textures and images.

Keywords: detail synthesis, texture transfer, image synthesis, super-resolution

1 Introduction

Single image super-resolution (SISR) is a challenging problem due to its ill-posed nature—there exist many high resolution images (output) that could downsample to the same low resolution input image. Given moderate scaling factors, high contrast edges might warrant some extent of certainty in the high resolution output image, but smooth regions are impossible to recover unambiguously. As a result, most methods aim to intelligently hallucinate image details and textures while being faithful to the low resolution image [Freeman et al. 2002; Sun and Tappen 2010; HaCohen et al. 2010; Sun and Hays 2012]. While recent state-of-the-art methods [Yang and Yang 2013; Timofte et al. 2014; Dong et al. 2014; Wang et al. 2015] are capable of delivering impressive performance in term of PSNR/SSIM metrics, the improvement in visual quality compared to earlier successful methods such as [Yang et al. 2008] are not as apparent. In particular, the amount of image textural details are still lacking in these leading methods. We build on traditional and recent deep learning based texture synthesis approaches to show that reliable texture transfer can be achieved in the context of single image super-resolution and hallucination.

Being able to model and represent natural image content is often a required first step towards recovering and hallucinating image details. Natural image models and priors have come a long way, from simple edge representations to more complex patch based models. Image restoration applications such as image super-resolution, deblurring, and denoising, share a similar philosophy in their respective framework to address the ill-posed nature of these tasks. A common strategy is to introduce image priors as a constraint in conjunction with the image formation model. Natural image content spans a broad range of spatial frequencies, and it is typically easy to constrain the restoration process to reliable recover information in the low frequency bands. These typically include smoothly varying regions without large gradients (edges, sky). In fact, a Gaussian or Laplacian prior would suit well for most image restoration task. This family of image priors have been shown to work in a variety of settings, in [Fergus et al. 2006; Levin and Weiss 2007; Levin et al. 2009; Cho and Lee 2009; Xu and Jia 2010], to name a few. More advanced prior models have also been developed such as FRAME [Zhu et al. 1998], the Fields of Experts model [Roth and Black 2009], and the GMM model [Zoran and

Weiss 2011]. It is known that the filters learned in these higher order models are essentially tuned low high-pass filters [Weiss and Freeman 2007]. As a result, no matter how these priors are formulated, they work under the same principle by penalizing high frequency image content, imposing the constraint that “images should be smooth” unless required by the image reconstruction constraint. When these priors are universally applied to every pixel location in the image, it is bound to yield over-smoothed output. But smoothness is just another form of blur, which is exactly what we are trying to avoid in the solution space in super-resolution.

To achieve sharpness in the upsampled image, successful methods usually learn a statistical mapping between low resolution (LR) and high resolution (HR) image patches. The mapping itself can be non-parametric [Freeman et al. 2002; Huang et al. 2015], sparse coding [Yang et al. 2008], regression functions [Kim and Kwon 2010; Yang and Yang 2013], random forest [Schulter et al. 2015], and convolutional neural networks [Dong et al. 2014; Wang et al. 2015; Johnson et al. 2016]. There are pros and cons of both parametric and non-parametric representations. Parametric methods typically offer much faster performance at test time and produce higher PSNR/SSIM scores. But no matter how careful one engineers the loss function during training, the learned mapping will suffer from the inherent ambiguity in low to high resolution patch mapping (many-to-one), and end up with a conservative mapping to minimize loss (typically MMSE). This regression-towards-the-mean problem suppresses high frequency details in the HR output. Non-parametric methods are bound to the available example patch pairs in the training process, hence unable to synthesize new image content besides simple blending of patches. As a result, more artifacts can be found in the output image due to misalignment of image content in overlapping patches. However, non-parametric methods tend to be more aggressive in inserting image textures and details [HaCohen et al. 2010; Sun and Hays 2012].

More recently, deep learning based approaches have been adopted with great success in many image restoration and synthesis tasks. The key is to use well-established deep networks as an extremely expressive feature space to achieve high quality results. In particular, a large body of work on image and texture synthesis have emerged and offer promising directions for single image super-resolution. By constraining the Gram matrix at different layers in a large pre-trained network, Gatys *et al.* showed that it is possible to synthesize a wide variety of natural image textures with almost photo-realistic quality [Gatys et al. 2015b]. Augmenting the same constraint with another image similarity term, they showed that artistic styles can be transferred [Gatys et al. 2015a; Gatys et al. 2016] from paintings to photos in the same efficient framework. Recent work [Sajjadi et al. 2016; Johnson et al. 2016] show that by training to minimize perceptual loss in the feature space, superior visual quality can be achieved for SISR. However, their success at synthesizing natural textures is still limited as shown in their examples.

In this work, we build on the same approach from [Gatys et al. 2015a] and adapt it handle SISR. We focus on synthesis and transfer aspect of natural image textures, and show that high frequency details can be reliably transferred and hallucinated from example images to render convincing HR output.

*e-mail:lbsun@cs.brown.edu

†e-mail:hays@gatech.edu

2 Related Work

2.1 Single Image Super-resolution (SISR)

Single image super-resolution is a long standing challenge in computer vision and image processing due to its extremely ill-posed nature. However, it has attracted much attention in recent research due to new possibilities introduced by big data and deep learning. Unlike traditional multi-frame SR, it is impossible to unambiguously restore high frequencies in a SISR framework. As a result, existing methods *hallucinate* plausible image content by relying on carefully engineered constraints and optimization procedures.

Over the past decade, SISR methods have evolved from interpolation based and edge oriented methods to learning based approaches. Such methods learn a statistical model that maps low resolution (LR) patches to high resolution (HR) patches [Yang et al. 2008; Kim and Kwon 2010; Yang and Yang 2013; Timofte et al. 2013; Timofte et al. 2014; Schuler et al. 2015], with deep-learning frameworks being the state-of-the-art [Dong et al. 2014; Wang et al. 2015]. While these methods perform well in terms of PSNR/SSIM, high frequency details such as textures are still challenging to hallucinate because of the ambiguous mapping between LR and HR image patches. In this respect, non-parametric patch-based methods have shown promising results [Freeman et al. 2002; Sun et al. 2010; HaCohen et al. 2010; Sun and Hays 2012; Huang et al. 2015]. These methods introduce explicit spatial [Freeman et al. 2002] and contextual [Sun et al. 2010; HaCohen et al. 2010; Sun and Hays 2012] constraints to insert appropriate image details using external example images. On the other hand, internal image statistics based methods have also shown great success [Freedman and Fatfal 2011; Glasner et al. 2009; Yang et al. 2013; ?; Huang et al. 2015]. These methods directly exploit self-similarity within and across spatial scales to achieve high quality results.

More recently, new SISR approaches have emerged with an emphasis on synthesizing image details via deep networks to achieve better visual quality. Johnson *et al.* [Johnson et al. 2016] show that the style transfer framework of [Gatys et al. 2015a] can be made real-time, and show that networks trained based on perceptual loss in the feature space can produce superior super-resolution results. Sajjadi *et al.* [Sajjadi et al. 2016] consider the combination of several loss functions for training deep networks and compare their visual quality for SISR.

2.2 Texture and Image synthesis

In texture synthesis, the goal is to create an output image that matches the textural appearance of an input texture to minimize perceptual differences. Early attempts took a parametric approach [Heeger and Bergen 1995; Portilla and Simoncelli 2000] by matching statistical characteristics in a steerable pyramid. Non-parametric methods [Bonet 1997; Efros and Leung 1999; Efros and Freeman 2001; Kwatra et al. 2003; Wei and Levoy 2000; Kwatra et al. 2005] completely sidestep statistical representation for textures, and synthesize textures by sampling pixels or patches in a nearest neighbor fashion. More recently, Gatys *et al.* [Gatys et al. 2015b] propose Gram matrix based constraints in the rich and complex feature space of the well-known VGG network [Simonyan and Zisserman 2014], and show impressive synthesized results on a diverse set of textures and images. This deep learning based approach shares many connections with earlier parametric models such as [Heeger and Bergen 1995; Portilla and Simoncelli 2000], but relies on orders of magnitudes more parameters, hence is capable of more expressive representation of textures.

Synthesizing an entire natural image from scratch is an extremely

difficult task. Yet, recent advances in deep learning have shown promising success. Goodfellow *et al.* [Goodfellow et al. 2014] introduced the Generative Adversarial Network (GAN) to pair a discriminative and generative network together to train deep generative models capable of synthesizing realistic images. Follow-up works [Denton et al. 2015; Radford et al. 2016; Nguyen et al. 2016] extended the GAN framework to improve the quality and resolution of generated images. However, the focus of this line of work has been to generate realistic images consistent with semantic labels such as object and image classes, in which low and mid level image features typically play a more crucial role, whereas the emphasis on high resolution image details and textures is not the primary goal.

2.3 Image Style and Detail Transfer

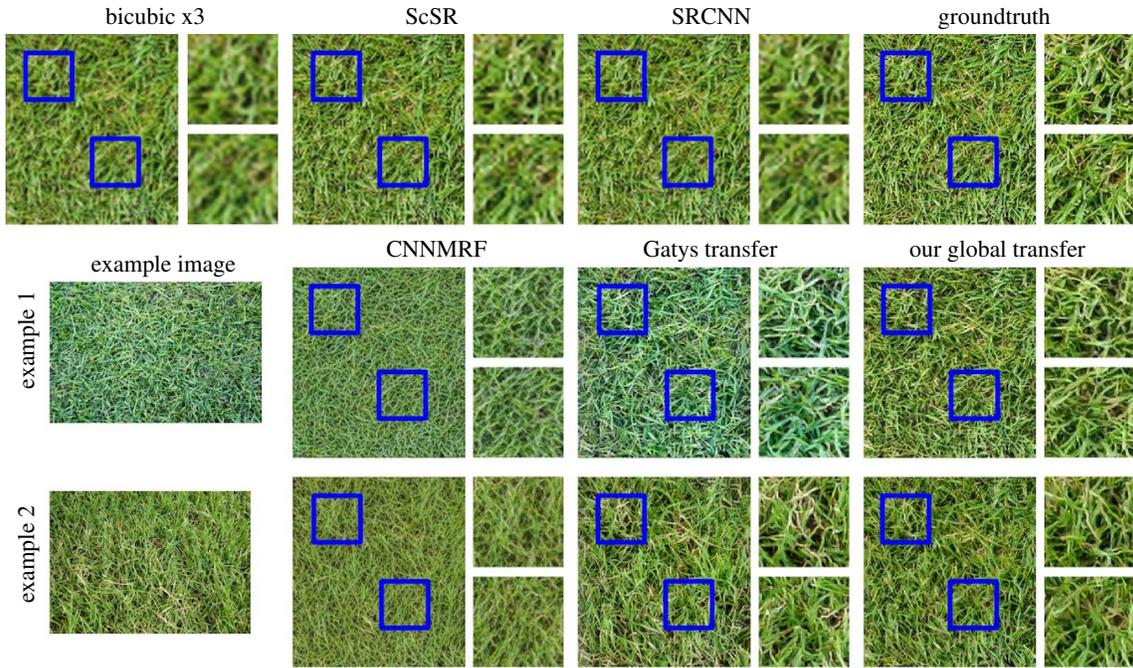
Many works exist in the domain of style and detail transfer between images. [Johnson et al. 2010] enhance the realism of computer generated scenes by transferring color and texture details from real photographs. [Shih et al. 2013] consider the problem of hallucinating time of day for a single photo by learning local affine transforms in a database of time-lapse videos. [Laffont et al. 2014] utilize crowd-sourcing to establish an annotated webcam database to facilitate transferring high level transient attributes among different scenes. Style transfer for specific image types such as portraits is also explored by [Shih et al.], in which multi-scale local transforms in a Laplacian pyramid are used to transfer contrast and color styling from exemplar professional portraits.

More recently, [Gatys et al. 2015a] propose a style transfer system using the 19-layer VGG network [Simonyan and Zisserman 2014]. The key constraint is to match the Gram matrix of numerous feature layers between the output image and a style image, while high level features of the output is matched that of a content image. In this way, textures of the style image is transferred to the output image as if painted over the content image, similar to Image Quilting [Efros and Freeman 2001]. Drawing inspirations from texture synthesis methods, [Li and Wand 2016] propose to combine a MRF with CNN for image synthesis. This CNNMRF model adds additional layers in the network to enable resampling ‘neural patches’, namely, each local window of the output image should be similar to some patch in the style image *in feature space* in a nearest neighbor sense. This has the benefit of more coherent details should the style image be sufficiently representative of the content image. However, this copy-paste resampling mechanism is unable to synthesize new content. In addition, this method is prone to produce ‘washed out’ artifacts due the blending/averaging of neural patches. This is a common problem to patch-based synthesis methods [Efros and Freeman 2001; Freeman et al. 2002; Kwatra et al. 2005]. Other interesting deep learning based applications such as view synthesis [Zhou et al. 2016] and generative visual manipulation [Zhu et al. 2016] have also been proposed. These methods allow us to better understand how to manipulate and transfer image details without sacrificing visual quality.

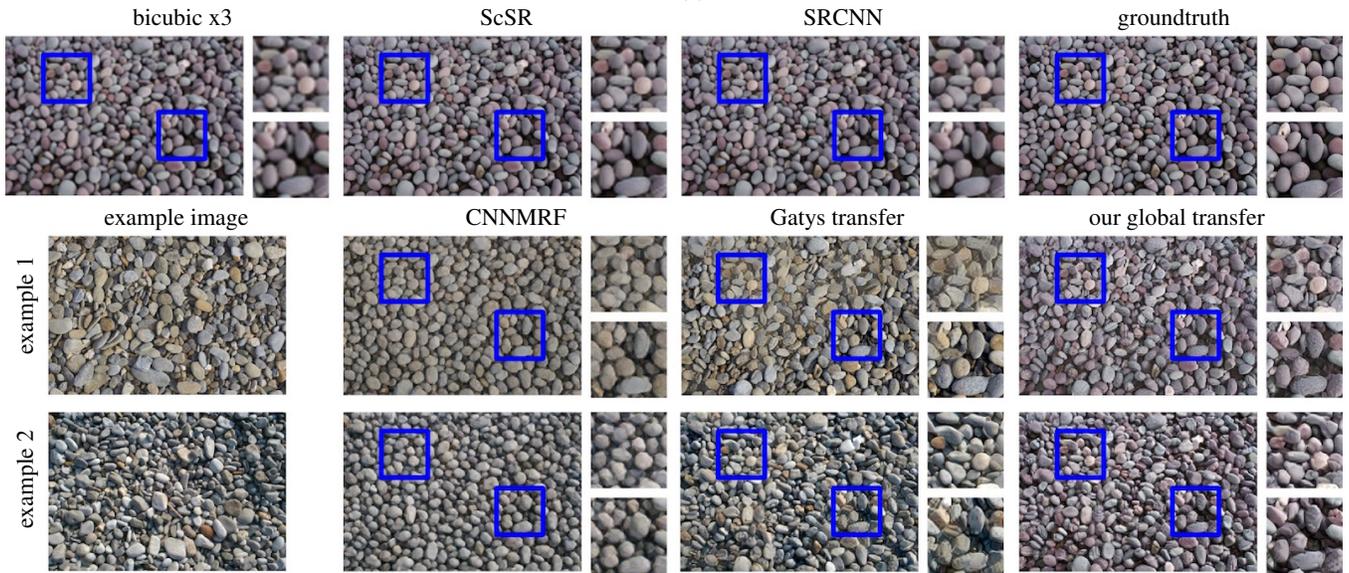
3 Method

Our method is based on [Gatys et al. 2015a; Gatys et al. 2015b], which encodes feature correlations of an image in the VGG network via the Gram matrix. The VGG-Network is a 19-layer CNN that rivals human performance for the task of object recognition. This network consists of 16 convolutional layers, 5 pooling layers, and a series of fully connected layers for softmax classification.

A latent image x is to be estimated given constraints such as content similarity and style similarity. We assume a style or example image s is available for the transfer of appropriate textures from s to x , and that x should stay similar to a content image c in terms



(a)



(b)

Figure 1: A sample comparison of various algorithms applied to upsampling texture images for a factor of $\times 3$. Two example images are provided in both (a) and (b) for example-based approaches. It can be seen that the example image has significant impact on the appearance of the hallucinated details in the output images, indicating effectiveness of the texture transfer process.

of mid to high level image content. The feature space representations with the network are X , S and C respectively. At each layer l , a non-linear filter bank of N_l filters is convolved with the previous layer’s feature map to produce an encoding in the current layer, which can be stored in a feature matrix $X^l \in \mathcal{R}^{N_l \times M_l}$, where M_l is the number of elements in the feature map (height times width). We use X_{ij}^l to denote the activation of the i^{th} filter at position j in layer l generated by image x .

In [Gatys et al. 2015a], the goal is to solve for an image x that is similar to a content image c but takes on the style or textures of s . Specifically, the following objective function is minimized via gradient descent to solve for x :

$$x = \arg \min_x (\alpha E_{content}(c, x) + \beta E_{style}(s, x)) \quad (1)$$

where $E_{content}$ is defined as:

$$E_{content}(c, x) = \frac{1}{2} \sum_l \sum_{ij} (C_{ij}^l - X_{ij}^l)^2 \quad (2)$$

The content similarity term is simply a L_2 loss given the difference between the feature map of the latent image in layer l and the corresponding feature map from the content image.

The definition of E_{style} is based on the the L_2 loss between the Gram matrix of the latent image and the style image in a set of chosen layers. The Gram matrix encodes the correlations between the filter responses via the inner product of vectorized feature maps. Given a feature map X^l for image x in layer l , the Gram matrix $G(X^l) \in \mathcal{R}^{N_l \times M_l}$ has entries $G_{ij} = \sum_k X_{ik}^l X_{jk}^l$, where i, j index through pairs of feature maps, and k indexes through positions in each vectorized feature map. Then the style similarity component of the objective function is defined as:

$$E_{style}(s, x) = \sum_l \frac{w_l}{4N_l^2 M_l^2} \left(\sum_{i,j} (G(S^l)_{ij} - G(X^l)_{ij})^2 \right) \quad (3)$$

where w_l is a relative weight given to a particular layer l . The derivatives of the above energy terms can be found in [Gatys et al. 2015a]. To achieve best effect, the energy components are typically enforced over a set of layers in the network. For example, the content layer can be a single conv4_2 layer, while the style layers can be over a larger set {conv1_1, conv2_1, conv3_1, conv4_1, conv5_1} to allow consistent texture appearances across all spatial frequencies.

This feature space constraint has been shown to excel at representing natural image textures for texture synthesis, style transfer, and super-resolution. We introduce a few adaptations to the task of single image super-resolution and examine its effectiveness in terms of transferring and synthesizing natural textures.

3.1 Basic Adaptation to SR

The objective function in Equation 1 consists of a content similarity term and a style term. The content term is analogous to the faithfulness term in SISR frameworks. The style term can be seen as a natural image prior derived from a single example image, which is assumed to represent the desired image statistics. A first step in our experiments is to replace the content similarity term $E_{content}$ with a faithfulness term $E_{faithfulness} = |G * x \downarrow_f - c|^2$, where f is the downsampling factor, G a Gaussian lowpass filter, and c the low resolution input image that we would like to upsample. These variables associated with the downsampling process are assumed known a-priori (non-blind SR). In the subsequent discussion, we refer to this basic adaptation as **our global**, since the Gram matrix

constraint is globally applied to the whole image. Formally, the **our global** method solves the following objective via gradient descent:

$$x = \arg \min_x (\alpha E_{faithfulness}(c, x) + \beta E_{style}(s, x)) \quad (4)$$

We further make the following changes to the original setup:

- All processing is done in gray scale. The original work of [Gatys et al. 2015a] computes the feature maps using RGB images. However, this requires strong similarity among color channel correlations between the example and input image, which is hard to achieve. For transferring artistic styles, this is not a problem. We drop the color information to allow better sharing of image statistics between the image pair.
- We use the layers {conv1_1, pool1_1, pool2_1, pool3_1, pool4_1, pool5_1} to capture the statistics of the example image for better visual quality, as done in [Gatys et al. 2015b].

We show that the above setup, while simple and basic, is capable of transferring texture details reliably for a wide variety of textures (see Fig.1 and Fig.6), even if the textures are structured and regular (see Fig.5). However, for general natural scenes, this adaptation falls short and produces painterly artifacts or inappropriate image details for smooth image regions, because their global image statistics no longer matches each other.

3.2 Local Texture Transfer via Masked Gram Matrices

Natural images are complex in nature, usually consisting of a large number of segments and parts, some of which might contain homogeneous and stochastic textures. Clearly, globally matching image statistics for such complex scenes cannot be expected to yield good results. However, with carefully chosen local correspondences, we can selectively transfer image details by pairing image parts of the same or similar textures via two sets of binary masks $\{m_s^k\}_1^K$ and $\{m_x^k\}_1^K$. To achieve this, we introduce an outer summation to the E_{style} term to loop over each corresponding pair of components in the masks (see Eq(5)).

In this setup, R_x^l is an image resizing operator that resamples an image (a binary mask in this case) to the resolution of feature map x^l using nearest neighbor interpolation. The normalization constant also reflects that we are aggregating image statistics over a subset of pixels in the images. The parameter β from Eq.1 is divided by the number of masks K to ensure the same relative weight between $E_{faithfulness}$ and $E_{stylelocal}$. Note that these binary masks are not necessarily exclusive, namely, pixels can be explained by multiple masks if need be.

The sparse correspondences are non-trivial to obtain. We examine two cases for the correspondence via masks: manual masks, and automatic masks via the PatchMatch [Barnes et al. 2009] algorithm.

Manual Masks For moderately simple scenes with large areas of homogeneous textures such as grass, trees, sky, *etc.*, we manually generate 2 to 3 masks per image at the full resolution to test out the local texture transfer. We refer to this setup as **our local manual**. A visualization of the images and masks can be found in Figure 2.

PatchMatch Masks To automatically generate the masks, we apply the PatchMatch algorithm to the LR input image c and a LR version of the style image s after applying the same downsampling process used to generate c . Both images are grayscale. Once the nearest-neighbor field (NNF) is computed at the lower resolution, we divide the output image into cells and pool and dilate the interpolated offsets at the full resolution to form the mask pairs. Each m_x^k contains a square cell of 1’s, and its corresponding mask m_s^k

$$E_{stylelocal} = \sum_k E_{style}(s \otimes m_s^k, x \otimes m_x^k) = \sum_k \sum_l \frac{w_l}{4N_l^2 |R_x^l(m_x^k)|^2} \left(\sum_{i,j} \left(G(S^l \otimes R_s^l(m_s^k))_{ij} - G(X^l \otimes R_x^l(m_x^k))_{ij} \right)^2 \right) \quad (5)$$

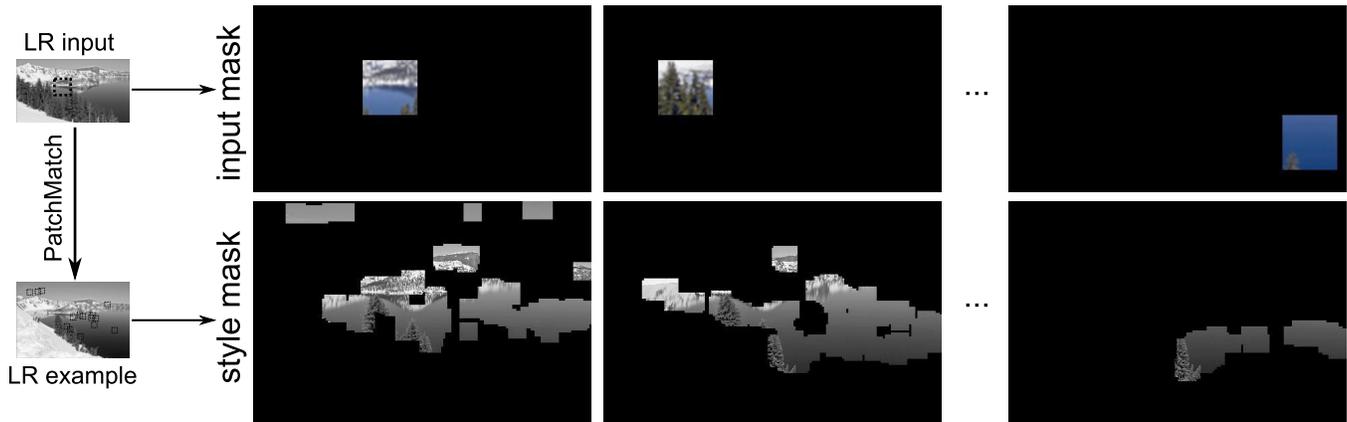


Figure 3: Visualization of the masks automatically generated using the PatchMatch algorithm. PatchMatch is applied to the low resolution grayscale input and example images to compute a dense correspondence. The HR output image is divided into cells, and all correspondences contained in the input cell are aggregated to form the example image mask.

will be the union of numerous of binary patches. We refer to this variation as **our local**. A sample visualization is given in Figure 3.

4 Experimental Results

4.1 Baseline Methods

For comparison, we first describe several baseline methods from recent literature on super-resolution and texture transfer, and compare to our methods. These baseline methods are representative of state-of-the-art performance in their respective tasks, and form the basis of comparison for Section 4.2.

ScSR [Yang et al. 2008; Yang et al. 2010] is one of the most widely used methods for comparison in recent SISR literature. It is a sparse coding based approach, using a dictionary of 1024 atoms learned over a training set of 91 natural images. Sparse coding is a well studied framework for image reconstruction and restoration, in which the output signal is assumed to be a sparse linear activation of atoms from a learned dictionary. We use the Matlab implementation provided by the authors¹ as a baseline method for comparison.

SRCNN [Dong et al. 2014] is a CNN based SISR method that produces state-of-the-art performance for PSNR/SSIM measures among recent methods. It combines insights from sparse coding approaches and findings in deep learning. A 3-layer CNN architecture is proposed as an end-to-end system. We can view this representation as a giant non-linear regression system in neural space, mapping LR to HR image patches. For subsequent comparisons, we use the version of SRCNN learned from 5 million of 33×33 subimages randomly sampled from ImageNet. The Matlab code package can be found on the author’s website².

Gatys [Gatys et al. 2015a; Gatys et al. 2015b] first consider reformulating the texture synthesis problem within a CNN framework.

¹We use the Matlab ScSR code package from <http://www.ifp.illinois.edu/~jyang29/codes/ScSR.rar>

²We use the SRCNN code package from <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>



Figure 2: Sample images and their corresponding masks, each one is manually generated.

In both work, the VGG network is used for feature representation and modeling image space, and the correlation of feature maps at each layer is the key component in encoding textures and structures across spatial frequencies. The Gram matrix representation is compact and extremely effective at synthesizing a wide variety of textures [Gatys et al. 2015b]. We use a Lasagne and Theano based implementation of [Gatys et al. 2015a] as a baseline method for comparison³.

CNNMRF [Li and Wand 2016] address the loss of spatial information due to the Gram matrix representation by introducing an MRF style layer on top of the VGG hidden layers to constrain local similarity of *neural patches*, where each local window in the output image feature map is constrained to be similar to the nearest neighbor in the corresponding layer of the style image feature maps. We use the `torch` based implementation from the authors⁴.

To adapt the code from Gatys *et al.* and CNNMRF for our experiments, we upsample the LR input image bicubically to serve as the content image. All other processing remain identical to their respective implementation.

We show a sample comparison of these methods in Figure 1, where a low resolution texture image is upsampled by a factor of 3. For the example based methods [Gatys et al. 2015a; Li and Wand 2016] and ours, we provide two example images to test the algorithm’s ability in transferring textures. Some initial observations can be made:

- ScSR [Yang et al. 2008] and SRCNN [Dong et al. 2014] produce nearly identical results qualitatively, even though their model complexity is orders of magnitude apart. This represents half a decade of progress in the SISR literature.
- CNNMRF [Li and Wand 2016] produces painterly artifacts due to averaging in neural space. The highest frequencies among different color channels can be misaligned and appear as colored halos when zoomed in.
- Our method produces convincing high frequency details while being faithful to the LR input. The effect of the example image can be clearly seen in the output image.

4.2 Comparison of Results

In this section we showcase the performance of the algorithm variants **our global**, **our local** (PatchMatch based) and **our local manual** on a variety of textures and natural images. We also compare against leading methods in single-image super-resolution such as ScSR [Yang et al. 2008] and SRCNN [Dong et al. 2014], as well as deep learning based style transfer methods including [Gatys et al. 2015a] and CNNMRF [Li and Wand 2016]

4.2.1 Test Data

We collect a variety of images from the Internet including natural and man-made textures, regular textures, black and white patterns, text images, simple natural scenes consisting of 2 or 3 clearly distinguishable segments, and face images. These test images are collected specifically to test the texture transfer aspect of the algorithms. As a result, we do not evaluate performance of single image super-resolution in its traditional sense, namely, measuring PSNR and SSIM.

³Our implementation is adapted from the art style transfer recipe from Lasagne: <https://github.com/Lasagne/Recipes/tree/master/examples/styletransfer>

⁴Chuan Li’s CNNMRF implementation is available at: <https://github.com/chuanli11/CNNMRF>

4.2.2 Black and White Patterns

The simplest test images are texts and black and white patterns. As shown in Figure 4, traditional SR algorithms do a decent job at sharpening strong edges, with SRCNN producing slightly less ringing artifacts than ScSR. As expected, the example based methods produce interesting hallucinated patterns based on the example image. CNNMRF yields considerable amount of artifacts due to averaging patches in neural space. Gatys and our global introduce a bias in background intensity but are capable of keeping the edges crisp and sharp. Much fine details and patterns are hallucinated for the bottom example.

4.2.3 Textures

For homogeneous textures, most SISR methods simply cannot insert meaningful high frequency content besides edges. On the other hand, we see that the Gram matrix constraint from [Gatys et al. 2015a; Gatys et al. 2015b] works extremely well because it is co-ercing image statistics across spatial frequencies in neural space, and ensuring that the output image match these statistics. However it is less effective when it comes to non-homogeneous image content such as edges and salient structures, or any type of image phenomena that is spatially unexchangeable. Finally, CNNMRF works reasonably well but still falls short in terms of realism. This is because linear blending of neural patches inevitably reduces high frequencies. Another artifact of this method is that this blending process can produce neural patches from the *null space* of natural image patches, introducing colored halos and tiny rainbows when zoomed in.

The main benefits of the **our global** method are (1) better faithfulness to the input LR image, and (2) less color artifacts. The Gatys transfer baseline operates in RGB color space, hence any correlated color patterns from the style image will remain in the output image. However, the style image might not represent the correct color correlation observed in the input image, *e.g.*, blue vs yellow flowers against a background of green grass. Our global transfer method operates in gray scale, relaxing the correlation among color channels and allowing better sharing of image statistics. This relaxation helps bring out a more realistic output image, as shown in Figure 5, 6, 7.

Comparisons on regular textures are shown in Figure 5. **our global** produces better details and color faithfulness, whereas traditional SISR methods do not appear too different from bicubic interpolation. Figure 6 shows results on numerous stochastic homogeneous textures. Example based methods exhibit strong influence from example images and can produce an output image visually different from the input, such as the fur image (third row). However, better details can be consistently observed throughout the examples. **Gatys** can be seen to produce a typical *flat* appearance in color (*e.g.*, rock, first row), this is because of the color processing constraint.

Going beyond homogeneous textures, we test these algorithms on simple natural images in Figure 7. Realistic textures and details can be reasonably well hallucinated by **our global**, especially the roots in the roil (first row) and the patterns on the butterfly wings (bottom row). The pipes (second row) are synthesized well locally, however, the out output image becomes too ‘busy’ when viewed globally. It is worth pointing out that CNNMRF essentially produces a painting for the forest image (third row), this is a clear example of the disadvantages of averaging/blending patches.

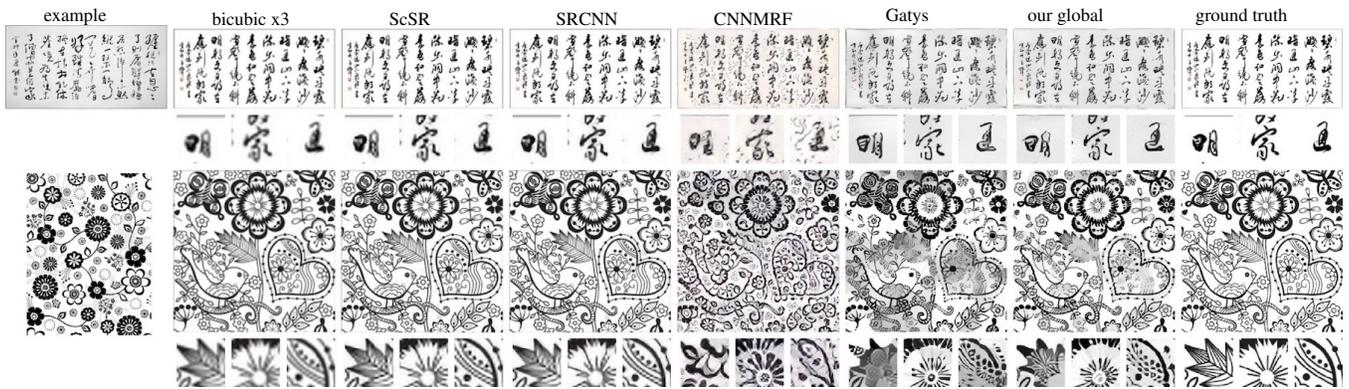


Figure 4: Example comparisons on a Chinese text image (top) and black and white pattern image (bottom). Example based methods can hallucinate edges in interesting ways, but also produce biases in background intensity, copied from the example image. Other artifacts are also present. Best viewed electronically and zoomed in.

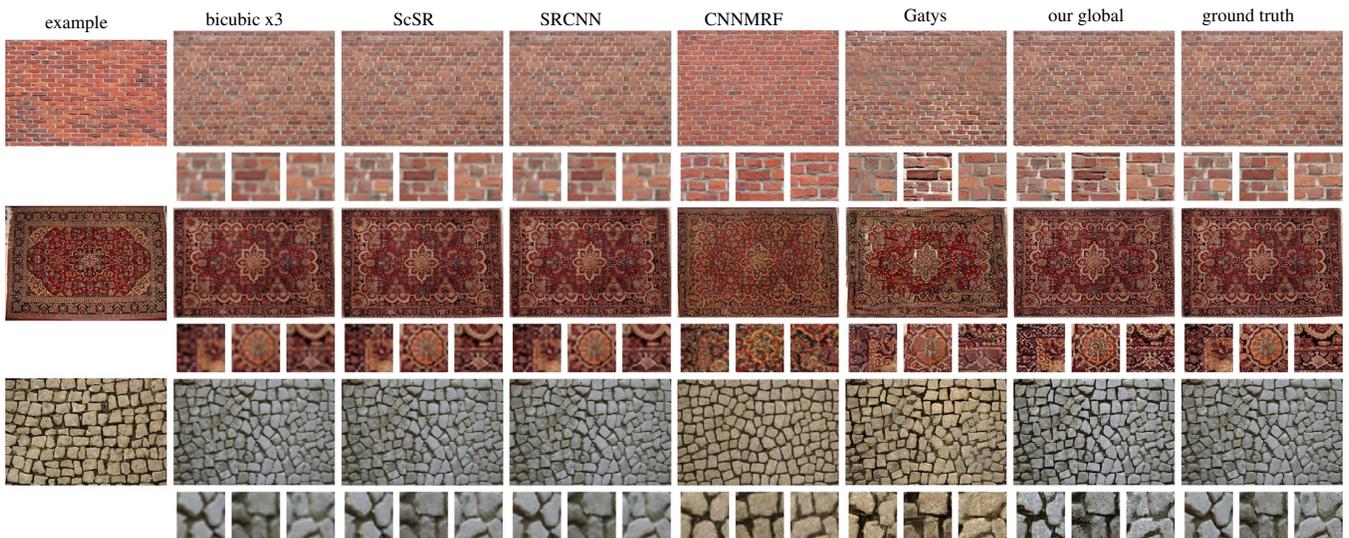


Figure 5: Example comparisons on regular textures. Best viewed electronically and zoomed in.

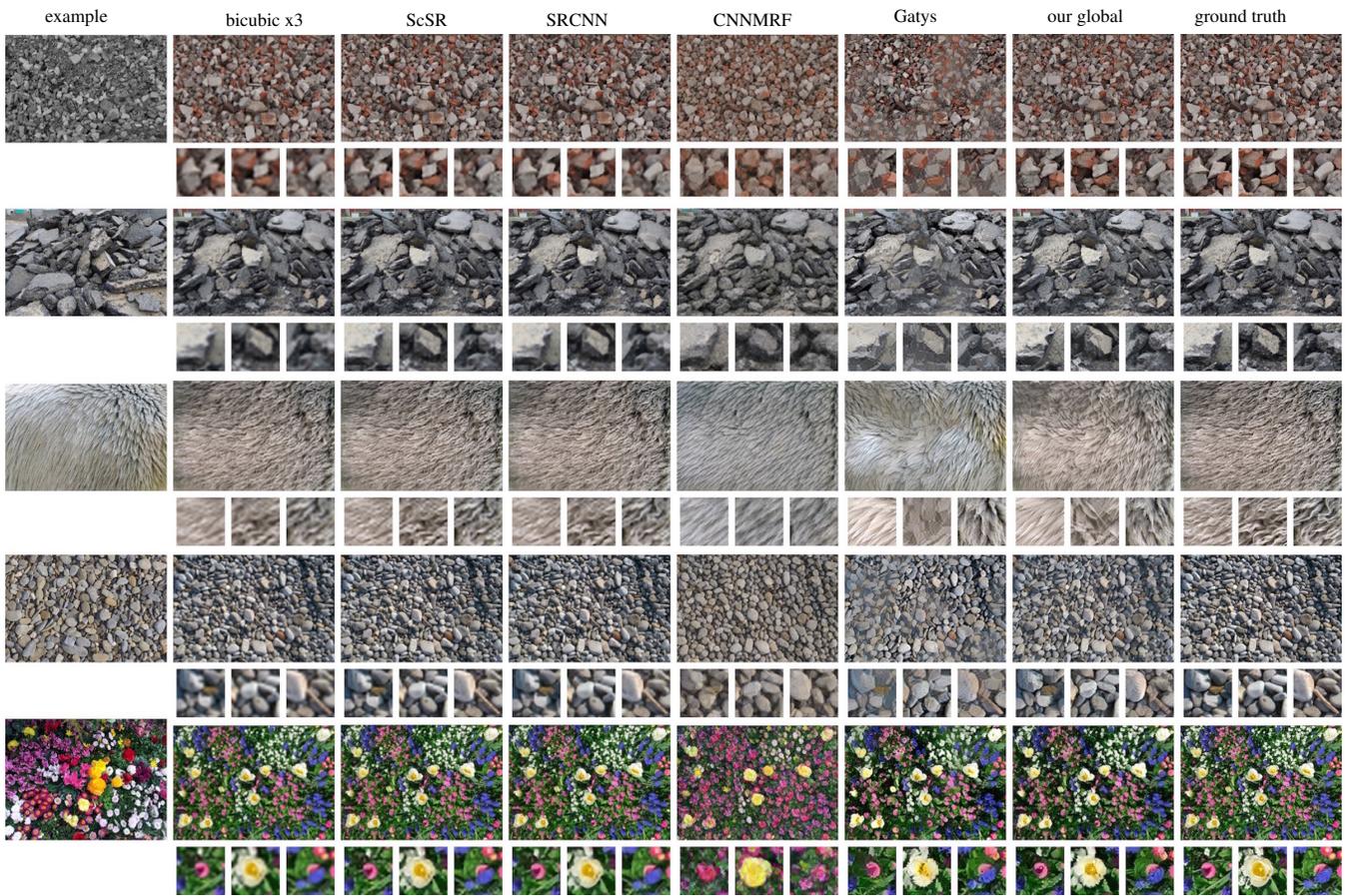


Figure 6: Example comparisons on various types of textures. Best viewed electronically and zoomed in.



Figure 7: Example comparisons on simple natural images. Best viewed electronically and zoomed in.

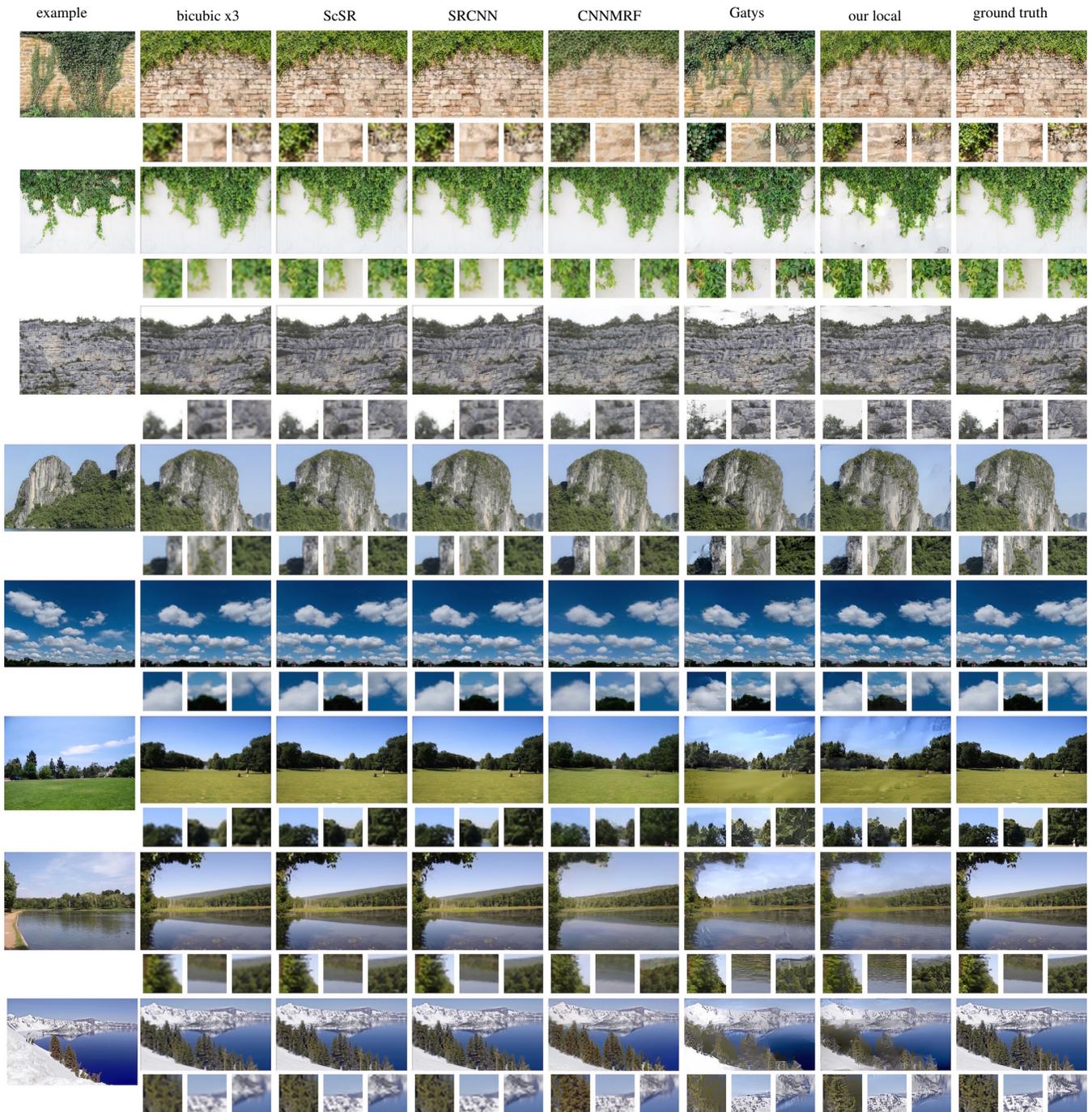


Figure 8: Example comparisons on moderately complex natural images. CNNMRF, Gatys and ‘our local’ consistently synthesize more high frequencies appropriate to the scene. CNNMRF and Gatys suffer from color artifacts due to mismatching colors between the example and the input image. CNNMRF also produces significant amount of color artifacts when viewed more closely, especially in smooth regions and near image borders. Gram matrix based methods such as Gatys and ‘our local’ outperform other methods in terms of hallucinating image details, however also produce more artifacts in a few test cases. Best viewed electronically and zoomed in.



Figure 9: Example comparisons on natural scenes with manually supplied masks. Best viewed electronically and zoomed in.

4.2.4 Natural Scenes

Natural images exhibit much more complexity than homogeneous textures, here we only consider scenarios where the image can be clearly divided into several types of textures, mostly homogeneous. In this way, we can better test the effectiveness of the algorithm’s performance on synthesizing and hallucinating texture details. One complication that arises here is that texture transitions and borders represent extremely non-homogeneous statistics that is not easily handled by synthesis methods. Since the image now contains different types of statistics, we will apply our masked variants using PatchMatch masks and manual masks to these test images. To better deal with texture transitions, we dilate the manually generated masks slightly to include pixels near texture borders.

In Figure 8, all results under **our local** are generated using our PatchMatch based variant. These test images consist of moderately complex natural scenes. It can be seen that CNNMRF, Gatys and **our local** consistently synthesize more high frequencies appropriate to the scene, traditional SISR methods appear similar to bicubic interpolation. CNNMRF and Gatys suffer from color artifacts due to mismatching colors between the example and the input image. Again, CNNMRF produces significant amount of color artifacts when viewed more closely, especially in smooth regions and near image borders. Gram matrix based methods such as Gatys and **our local** outperform other methods in terms of hallucinating image details, however also produce more artifacts in a few test cases.

PatchMatch is far from perfect for generating the masks suitable for our application. This can be seen in many regions in the output images. For example, the trees in the pond image (second last row) is hallucinated by water textures towards the left, even the tree on the far left shows much water-like textures, clearly due to bad correspondences generated by PatchMatch. Similar artifacts can be seen in the crater lake image (last row). For natural scenes, our method is capable of opportunistically inserting appropriate textures, but cannot produce a perfect flaw-free output.

One would expect manually generated masks to be more suitable than PatchMatch masks. Although there are two drawbacks:

- The entire masked example region would participate in the Gram matrix computation, forcing the output image to take on the exemplar statistics, even though it might be undesirable. For example, when matching sky with slow intensity gradient with a flat sky region. PatchMatch offers more freedom in this regard, allowing certain regions to be completely discarded (in the example image).
- Texture transitions are hard to account for. Even though we dilate the masks hoping to include the borders, the pyramid nature of the CNN architecture and pooling operations will eventually introduce boundary artifacts.

Figure 9 shows comparisons using our manually generated masks (c.r. Fig. 2). Clearly, there is less low frequency artifacts in color biases. However, ringing artifacts become more prominent near texture transitions and image borders.

4.3 Face Images

Another interesting scenario is to test the algorithms on face images. When the example image is sufficiently close to the input, such as in Figure 10, our method works well for hallucinating image details. In this particular example, the facial features in the output image remain similar to the input, and it is almost impossible to tell who the example image is given just the output. However, CNNMRF lacks the ability to synthesize new content (copy-paste

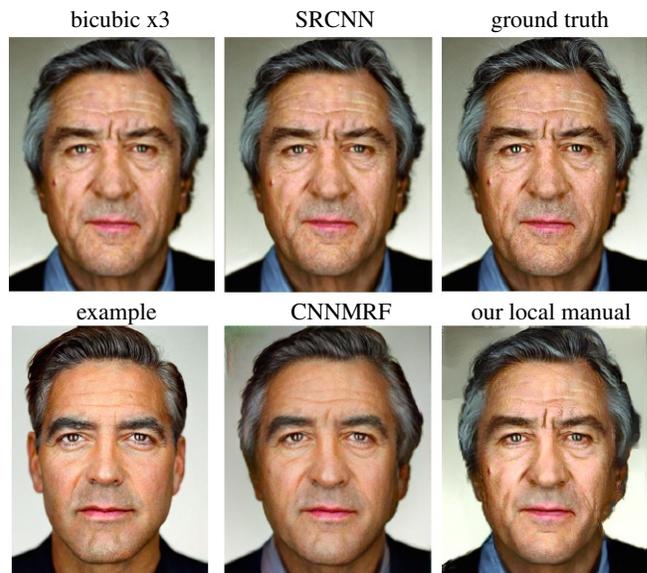


Figure 10: Example comparisons on a portrait image. Our method is able to hallucinate appropriate details given the well-matched image statistics. Most noticeably, plausible details are successfully introduced to the eyebrows, hair, and eyes. CNNMRF produces decent amount of details as well, however, it makes the output image less recognizable as the person in the input image. Best viewed electronically.

in neural space) and its output is more of a blend between the input and example. The final output image somewhat falls into the ‘uncanny valley’, and is almost unrecognizable as De Niro.

In Figure 11, CNNMRF is able to produce a natural looking output with decent high frequency details except for the mouth region, since the example image does not contain the best source patches. On the other hand, our Gram matrix based method (our global setup) fails completely for the face region, only synthesizing details on parts of the hat, which happens to be homogeneous textures. This is because human faces are highly structured and far from textures.

5 Discussion and Future Work

Recent works on the texture synthesis aspect of single image super-resolution provide a promising direction that complements existing methods which perform well in traditional image quality metrics. We have shown that deep architectures can provide the appropriate constraints in its rich feature space to model natural image content, especially textures. We have shown that the Gram matrix constraint from [Gatys et al. 2015b] can be easily adapted to achieve realistic transfer of high frequency details for wide variety of natural textures and images. With sparse spatial correspondences, more localized transfer of textures can be achieved to handle moderately complex natural scenes. However, it is non-trivial to handle texture transitions by matching statistics in neural space. Non-homogeneous textures, edges, and objects in natural images are also challenging to handle by this framework. Future work may focus on combining texture and object synthesis with traditional SISR approach for edge handling in a more unified framework.

References

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLD-

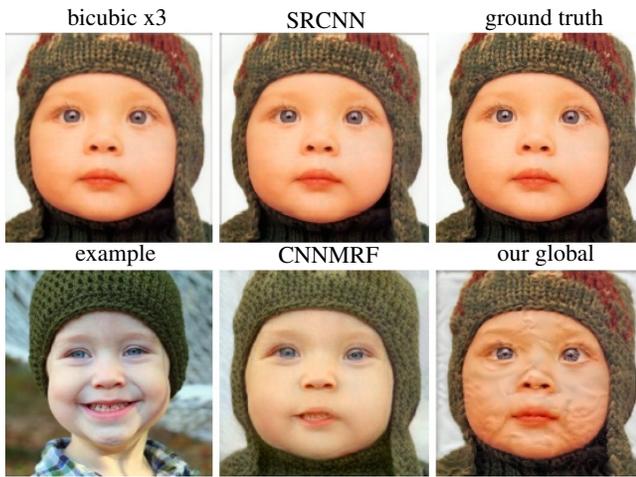


Figure 11: Example comparisons on a face image. Our method fails due to mismatch in global image statistics. It is interesting to note that CNNMRF works extremely well for face images, however, it cannot insert image details not present in the example image. In this case, it cannot synthesize a closed mouth of the baby. Best viewed electronically.

- MAN, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28, 3 (Aug.).
- BONET, J. S. D. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *ACM Transactions on Graphics*.
- CHO, S., AND LEE, S. 2009. Fast motion deblurring. In *ACM Transactions on Graphics*.
- DENTON, E. L., CHINTALA, S., SZLAM, A., AND FERGUS, R. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*.
- DONG, C., LOY, C. C., HE, K., AND TANG, X. 2014. Learning a deep convolutional network for image super-resolution. In *ECCV*.
- EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001* (August), 341–346.
- EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*.
- FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. T. 2006. Removing camera shake from a single photograph. In *ACM Transactions on Graphics*.
- FREEDMAN, G., AND FATTAL, R. 2011. Image and video upscaling from local self-examples. *ACM Trans. Graph.*
- FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. C. 2002. Example-based super-resolution. In *IEEE Computer Graphics and Applications*.
- GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2015. A neural algorithm of artistic style.
- GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2015. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems* 28.
- GATYS, L. A., ECKER, A. S., BETHGE, M., HERTZMANN, A., AND SHECHTMAN, E. 2016. Controlling perceptual factors in neural style transfer. *arXiv preprint arXiv:1611.07865*.
- GLASNER, D., BAGON, S., AND IRANI, M. 2009. Super-resolution from a single image. In *ICCV*.
- GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. C., AND BENGIO, Y. 2014. Generative adversarial nets. In *NIPS*.
- HACOHEN, Y., FATTAL, R., AND LISCHINSKI, D. 2010. Image upsampling via texture hallucination. In *ICCP*.
- HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*.
- HUANG, J.-B., SINGH, A., AND AHUJA, N. 2015. Single image super-resolution using transformed self-exemplars. In *CVPR*.
- JOHNSON, M. K., DALE, K., AVIDAN, S., PFISTER, H., FREEMAN, W. T., AND MATUSIK, W. 2010. Cg2real: Improving the realism of computer-generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*.
- JOHNSON, J., ALAHI, A., AND LI, F.-F. 2016. Perceptual losses for real-time style transfer and super-resolution. *ECCV*.
- KIM, K. I., AND KWON, Y. 2010. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32, 6.
- KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.* 22, 3 (July), 277–286.
- KWATRA, V., ESSA, I. A., BOBICK, A. F., AND KWATRA, N. 2005. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics*.
- LAFFONT, P., REN, Z., TAO, X., QIAN, C., AND HAYS, J. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.*
- LEVIN, A., AND WEISS, Y. 2007. User assisted separation of reflections from a single image using a sparsity prior. *TPAMI* 29, 9, 1647–1654.
- LEVIN, A., WEISS, Y., DURAND, F., AND FREEMAN, W. T. 2009. Understanding and evaluating blind deconvolution algorithms. In *CVPR*.
- LI, C., AND WAND, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. *CVPR*.
- NGUYEN, A., YOSINSKI, J., BENGIO, Y., DOSOVITSKIY, A., AND CLUNE, J. 2016. Plug & play generative networks: Conditional iterative generation of images in latent space. *arXiv preprint arXiv:1612.00005*.
- PORTILLA, J., AND SIMONCELLI, E. P. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision* 40, 1.
- RADFORD, A., METZ, L., AND CHINTALA, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*.
- ROTH, S., AND BLACK, M. J. 2009. Fields of experts. *International Journal of Computer Vision* 82, 2.

- SAJJADI, M. S. M., SCHLKOPE, B., AND HIRSCH, M. 2016. Enhancenet: Single image super-resolution through automated texture synthesis. *arXiv preprint arXiv:1612.07919*.
- SCHULTER, S., LEISTNER, C., AND BISCHOF, H. 2015. Fast and accurate image upscaling with super-resolution forests. In *CVPR*.
- SHIH, Y., PARIS, S., BARNES, C., FREEMAN, W. T., AND DURAND, F. 2013. Style transfer for headshot portraits. *ACM Trans. Graph.*
- SHIH, Y., PARIS, S., DURAND, F., AND FREEMAN, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.*
- SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- SUN, L., AND HAYS, J. 2012. Super-resolution from internet-scale scene matching. In *ICCP*.
- SUN, J., AND TAPPEN, M. F. 2010. Context-constrained hallucination for image super-resolution. In *CVPR*.
- SUN, J., ZHU, J., AND TAPPEN, M. F. 2010. Context-constrained hallucination for image super-resolution. In *CVPR*.
- TIMOFTE, R., SMET, V. D., AND GOOL, L. J. V. 2013. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*.
- TIMOFTE, R., SMET, V. D., AND GOOL, L. J. V. 2014. A+: adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*.
- WANG, Z., LIU, D., YANG, J., HAN, W., AND HUANG, T. 2015. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE International Conference on Computer Vision*.
- WEI, L., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000*.
- WEISS, Y., AND FREEMAN, W. T. 2007. What makes a good model of natural images? In *CVPR*.
- XU, L., AND JIA, J. 2010. Two-phase kernel estimation for robust motion deblurring. In *ECCV*.
- YANG, C., AND YANG, M. 2013. Fast direct super-resolution by simple functions. In *ICCV*.
- YANG, J., WRIGHT, J., HUANG, T. S., AND MA, Y. 2008. Image super-resolution as sparse representation of raw image patches. In *CVPR*.
- YANG, J., WRIGHT, J., HUANG, T. S., AND MA, Y. 2010. Image super-resolution via sparse representation. *IEEE Trans. Image Processing*.
- YANG, J., LIN, Z., AND COHEN, S. 2013. Fast image super-resolution based on in-place example regression. In *CVPR*.
- ZHOU, T., TULSIANI, S., SUN, W., MALIK, J., AND EFROS, A. A. 2016. View synthesis by appearance flow. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- ZHU, S. C., WU, Y. N., AND MUMFORD, D. 1998. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. In *International booktitle of Computer Vision*.
- ZHU, J.-Y., KRÄHENBÜHL, P., SHECHTMAN, E., AND EFROS, A. A. 2016. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*.
- ZORAN, D., AND WEISS, Y. 2011. From learning models of natural image patches to whole image restoration. In *ICCV*.