Single Image Super-resolution using Vectorization and Texture Synthesis

Kaoning Hu[1], Dongeun Lee[1] and Tianyang Wang[2]

[1]Department of Computer Science and Information Systems, Texas A&M University - Commerce, Commerce, Texas, U.S.A.

[2]Department of Computer Science and Information Technology, Austin Peay State University, Clarksville, Tennessee, U.S.A.

Keywords: SISR, Image Vectorization, Texture Synthesis, KS Test.

Abstract: Image super-resolution is a very useful tool in science and art. In this paper, we propose a novel method for single image super-resolution that combines image vectorization and texture synthesis. Image vectorization is the conversion from a raster image to a vector image. While image vectorization algorithms can trace the fine edges of images, they will sacrifice color and texture information. In contrast, texture synthesis techniques, which have been previously used in image super-resolution, can reasonably create high-resolution color and texture information, except that they sometimes fail to trace the edges of images correctly. In this work, we adopt the image vectorization to the edges of the original image, and the texture synthesis based on the Kolmogorov–Smirnov test (KS test) to the non-edge regions of the original image. The goal is to generate a plausible, visually pleasing detailed higher resolution version of the original image. In particular, our method works very well on the images of natural animals.

1 INTRODUCTION

Image super-resolution is a technique that enhances the resolution of digital images. It has a various range of applications including medical image processing, satellite image processing, art, and entertainment. In this paper, we focus on single image super-resolution (SISR), which is a classic computer vision problem to create a visually pleasing high-resolution image from a single low-resolution input image. In many situations, a single low-resolution image is the only source of data when a video sequence or high-resolution footage does not exist, thus SISR techniques are highly demanded. However, since the ground truth (i.e. the high-resolution information) does not exist in the low-resolution image, SISR is still a challenging problem.

In general, there are four types of super-resolution methods (Sun et al., 2008): interpolation-based, learning-based, reconstruction-based, and edge-directed. The interpolation-based methods, such as bilinear and bicubic interpolations, are simple and fast, and have been widely used in commercial software. However, these methods typically make the edges of the image blurry. The learning-based methods first learn the correspondences between low-resolution and high-resolution image patches from the training dataset, and then apply the learned model on the test set of low-resolution images to create a high-resolution image (Yang et al., 2019; Zhang et al., 2019). With deep neural networks, they can perform very well on general images. However, these methods rely largely on the quality of the similarity between the training set and the test set, and also require a large training set (Wang et al., 2013). The reconstruction-based methods assume that the downsampled version of the target high-resolution image is similar to the low-resolution image that we have (Wang et al., 2013; Damkat, 2011). However, it is observed that the reconstructed edges sometimes are unnaturally sharp and are often accompanied by undesired artifacts such as ringing (Wang et al., 2013). The edge-directed methods place emphasis on processing edges in the target high-resolution image by tracing the edge with the gradient of the input low-resolution image (Wang et al., 2013). While they can create smooth and natural edges in the target image, these methods cannot always properly handle certain texture regions such as grass and animal hair. Therefore, the edge-directed methods work better in upscaling of video game pixel arts (Stasik and Balcerek, 2017) and depth images (Liu and Ling, 2020; Xie et al., 2015), where there is not much texture information. Since each type of super-resolution methods has advantages and disadvantages, methods that combine two or more types also exist (Yang et al., 2017; Liu et al., 2019).

Hu, K., Lee, D. and Wang, T.
Single Image Super-resolution using Vectorization and Texture Synthesis.
DOI: 10.5220/0010325505120519
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 4: VISAPP*, pages 512-519
ISBN: 978-989-758-488-6

Due to the different types of image capturing devices, and objects in the images, there are many different categories of images. As such, SISR methods have been developed to particularly work on certain types of images. Besides the previously mentioned game pixel arts and depth images, there are also methods that work on remote sensing images (Jiang et al., 2019), medical images (Chen et al., 2018), and document and text images (Wang et al., 2019).

We have observed that most of the previous work emphasize enhancing edges in the images. Few methods exist that put emphasis on texture regions (Sajjadi et al., 2017). In this paper, we propose a novel approach that combines texture synthesis and image vectorization to upscale a single digital image. In particular, we focus attention on SISR for natural animal images, as our method works very well in processing the hair and feather parts of those images.

A digital image of natural animals or objects typically contains edge regions, where the brightness changes sharply along contours, and non-edge regions, where there is no significant edge. Although texture regions still have small edges as the colors of neighboring pixels could be different, these edges are usually too tiny and irregular to be traced. For this reason, we consider texture regions as non-edge regions. Edge regions and non-edge regions show significantly different visual characteristics. Therefore, it is reasonable to treat them separately and apply different upscaling operations.

Specifically, for the edge regions, we will adopt Potrace (Selinger, 2003), an image vectorization method, to trace the edge and upscale the regions. Potrace approximates the edges in an image first with polygons, and then sharp corners are approximated with sharp angles, while non-sharp corners are approximated with Bézier curves. For the non-edge regions, we will assume they contain texture information, and will exploit the self-similarity of the texture and apply a texture synthesis method based on Kolmogorov–Smirnov Test (KS test) (Massey Jr, 1951) to upscale the texture regions. Our work uses the KS test since it can naturally compare the probability distributions of an upscaled texture patch and a low-resolution texture patch whose sizes are different.

The paper is organized as follows. In Section 2, related work, including conventional algorithms and deep learning algorithms, is discussed. In Section 3, we will introduce the proposed method in detail. In Section 4, we will demonstrate results and evaluate the performance of our image super-resolution method. The conclusion and future work will be stated in Section 5.

2 RELATED WORK

In recent years, deep learning has been successfully applied to SISR, and some non-deep learning methods have also been successfully implemented in commercial software. Successful commercial software includes DCCI2x (Zhou et al., 2012) and ON1 Resize. DCCI2x is an edge-directed interpolation method that adapts to different edge structures of images. ON1 Resize uses the technology of the U.S. patent "Genuine Fractals®" (Faber and Dougherty, 2007), which exploits the self-similarity of an image to increase its size while preserving its details. Typical deep learning-based methods include (Zhang et al., 2019; Zhang et al., 2018), and (Ledig et al., 2017). In (Zhang et al., 2019), a novel degradation model for SISR was proposed, leveraging the classical blind deblurring methods. In (Zhang et al., 2018), the authors aimed to improve framework practicability by designing a single convolutional network that can take both blur kernel and noise level as input. Ledig et al. (Ledig et al., 2017) pioneered to propose a generative adversarial network (GAN) for SISR, namely SRGAN. All these methods achieved competitive quantitative results (in PSNR/SSIM) and appealing qualitative results. However, most of these methods emphasized processing of the edge regions, but not the texture regions. In 2017, Sajjadi et al. (Sajjadi et al., 2017) proposed a learning-based method that uses convolutional neural networks in an adversarial training setting with a texture loss function. Their goal was to upscale low-resolution images by a 4×4 factor with synthesized realistic textures.

It is understandable that in many applications, the primary goal is not to faithfully reconstruct a high-resolution version of an input low-resolution image, but to improve its appearance (Damkat, 2011). In 2011, Damkat (Damkat, 2011) proposed an SISR method using example-based texture synthesis. The assumption is that even in a low-resolution image, there are still a number of scale-invariant elements and self-similarity across scale. For example, animal hair looks similar at different scales, so upscaling may be achieved by generating more hairs instead of making hairs thicker. Their method achieved relatively good result in the texture regions except some salt-and-pepper noises.

The idea of self-similarity can be also explained by fractals, whose applications include fractal compression (Jacquin et al., 1992; Wohlberg and De Jager, 1999). Fractal compression takes advantage of self-similarity in images by analyzing and representing image blocks in terms of parametric models. Since parts of an image often resemble other parts of the same image thanks to the self-similarity, those parametric models can significantly reduce redundant information in images. Later, approximation to original images can be regenerated by synthesizing from the parametric models (Sayood, 2012). Our approach analyzes the low-resolution image to search for self-similar elements and synthesize new texture regions from those.

KS test is a popular test for checking similarity in time series (Lee et al., 2019; Lee et al., 2016) and images (Khatami et al., 2017; Hatzigiorgaki and Skodras, 2003). It is a statistical tool to measure whether two underlying one-dimensional probability distributions follow the same statistical distribution, which is simple to implement and faster than other non-parametric statistical hypothesis testing methods.

3 PROPOSED METHOD

The proposed method contains two parts: Image vectorization on the edge regions and texture synthesis on the non-edge regions. The two parts are then fused to form the final result.

Image vectorization, also known as image tracing or raster-to-vector conversion, has been used in computer graphics to convert a raster image to a vector image. Because vector images only keep the mathematical expressions of points, lines and curves, they are scalable. However, real-world photos cannot be represented as mathematical expressions alone. Therefore, real-world photos are typically stored as raster images, whereas vector images are typically used to store certain art works, such as logos, fonts and some cartoons. Due to these characteristics, image vectorization techniques cannot be directly used in image super-resolution.

On the other hand, an important aspect of image super-resolution is edge enhancement. To produce fine and clear edges in high resolution from a low-resolution image, edge extraction and tracing have been used in image super-resolution algorithms. Therefore, edge tracing techniques adopted from image vectorization algorithms can be very useful in processing the edges for the purpose of image super-resolution. In our paper, we will apply the algorithm of Potrace (Selinger, 2003) to scale the edge regions of the low-resolution image, as Potrace is one of the most promising image vectorization methods.

Besides the edge regions, an image also contains many non-edge regions. Some of these regions are plain regions, where color and brightness do not change very much. Examples are blue sky, white walls, blackboard, etc. Because there is not much information in these regions, any image scaling method can work well. However, there are also many non-edge regions with important texture information. Examples are animal hair, bird feather, lawns, wood surface, etc. Since these regions do not have significant edges, it is usually impossible to represent them with mathematical expressions. Therefore, an algorithm that processes the texture should be used instead. With a single low-resolution image, we will assume it contains a considerable amount of self-similar elements (Damkat, 2011) that can be reused to synthesize fine texture for a high-resolution image.

To measure the similarity between pieces of image texture elements, we will combine two measurements. One is the basic Euclidean distance, which is used to compare the color similarity of two pieces of image blocks. We consider each image block as a multi-dimensional vector, where the color of each pixel in the block is a component, and the Euclidean distance between two image blocks is computed in such a hyperspace. The other similarity measurement is based on the KS test. When we compute the similarity of two image blocks, we use the KS test to measure the probability distributions of the colors of their pixels.

With those image vectorization and texture synthesis techniques, the framework of our proposed algorithm is described as follows. Given a low-resolution raster image I_L, we first use a colored variant of Potrace algorithm (Selinger, 2003), to convert it to a vector image I_V. Then we use a texture synthesis method based on the KS test (Massey Jr, 1951) to upscale the low resolution raster image I_L to a high-resolution raster image I_H by the factor of 2×2. We then fuse I_V and I_H to get a final high-resolution image with fine edges I_F. To fuse the two images, we use edge regions from I_V and non-edge regions from I_H. To distinguish the edge regions and the non-edge regions, we use a Sobel edge detector on the original image I_L to extract the edge regions from the image. A flow chart of the proposed method is illustrated in Figure 1.

The details of image vectorization and texture synthesis will be discussed in the following subsections.

3.1 Image Vectorization using Potrace

Potrace is a robust image vectorization method that has been successfully used in many programs, including some commercial software. The original Potrace works only with binary images. Its algorithm transforms a black-white bitmap into a vector outline in three steps (Selinger, 2003).

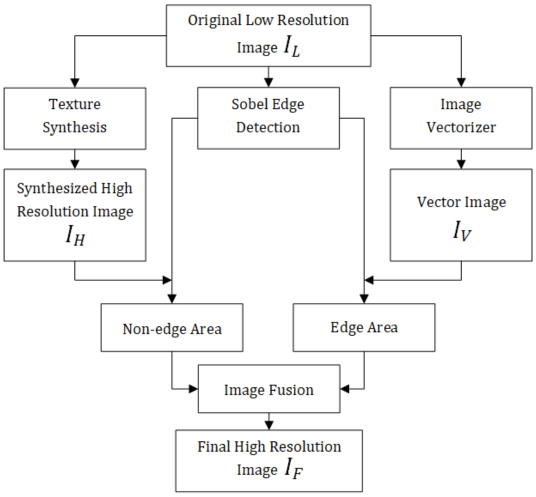

Figure 1: Flowchart of the proposed algorithm.

1. The bitmap is decomposed into a number of paths. A path is a sequence of connected pixels at the boundaries between black and white areas.
2. Each path is approximated by an optimal polygon. A polygon with fewer segments (edges) is considered optimal than one with more segments. Among the polygons with the same number of segments, a polygon with path points closer to the segments is considered optimal than one with path points strayed away from the segments.
3. Each polygon is transformed, so that each sharp corner is approximated by a sharp angle, while each non-sharp corners is approximated by a smooth Bézier curve.
4. An optional fourth step is to join consecutive Bézier curve segments together when possible.

To make Potrace work with color images, the algorithm is slightly modified. The edge at the boundary between each pair of different colors should be traced. However, in a true color image, there are 2^{24} different colors. As a result, it is impractical to trace the edges of all colors, because the computation will be too complex, and there will also be a lot of false and noisy edges. Therefore, the number of colors must be reduced. In our experiments, we have found that it is the best to reduce the number of colors to 64. The loss of colors will have little effect to the final result, because only significant edge regions from the vectorized image are used in the final fused image, while the rest of the final fused image come from the texture synthesized image.

(a) a 4x4 super block (b) a 2x2 block that is downscaled from the super block in (a)

Figure 2: Comparison after downscale may lead to salt-and-pepper noise in the texture synthesized image.

3.2 Texture Synthesis based on Kolmogorov–Smirnov Test

In this step, we upscale the original I_L using self-similarity to obtain image I_H. The scaling factor is 2×2. The original I_L is divided into small blocks of 2×2. For example, a 320×240 image should be divided into 160×120 blocks. To best preserve the geometrical structure and the texture information of I_L in I_H, each 2×2 block in I_L is replaced by a suitable 4×4 super block from the same image I_L. The self-similarity of these textures across scale makes their re-synthesis on a finer scale look plausible (Damkat, 2011). There are two criteria to choose a suitable 4×4 super block $S_{4\times4}$ that matches a 2×2 block $B_{2\times2}$:

1. When $S_{4\times4}$ is downscaled to 2×2, it should look like $B_{2\times2}$.
2. The color of the pixels in $S_{4\times4}$ and $B_{2\times2}$ should follow the same probability distribution.

The measurement of the first criterion is straightforward. We can downscale $S_{4\times4}$ to a 2×2 block $S_{2\times2}$, and compute its Euclidean distance (L2 distance) $Dist_{L2}$ to $B_{2\times2}$ in terms of pixel intensity (i.e. color).

However, the above criterion alone cannot guarantee that $S_{4\times4}$ matches $B_{2\times2}$. Here we have a special case that $S_{4\times4}$ does not match $B_{2\times2}$ but the downscaled $S_{2\times2}$ looks like $B_{2\times2}$. Given the super block in Figure 2(a), after it is downscaled, it would look like the block in Figure 2(b). However, (a) and (b) have distinctive texture patterns. If (a) is used to replace (b) in the texture synthesized image I_H, it will create some salt-and-pepper noises.

To avoid the above issue, we must ensure that the pixel intensity (color) of $S_{4\times4}$ should have a similar probability distribution to $B_{2\times2}$. Therefore, we apply the second criterion that requires the comparison of two blocks whose sizes are different. Here, we should not downscale $S_{4\times4}$, otherwise the above issue may still appear. We should not upscale $B_{2\times2}$ either, as upscaling will modify the probability distribution. To this end, we take advantage of the KS test, which is capable of comparing the probability distributions of two blocks, even if their sizes are different. In statistics, the KS test is a nonparametric test of the equality of continuous or discontinuous, one-dimensional probability distributions. It can be used to compare a sample with a reference probability distribution, or to compare two samples.

In our application, we use the two-sample KS test to test whether two underlying probability distributions of blocks differ or not. Since it is nonparametric, it can compare two pixel blocks from any arbitrary distributions without the restriction of parametric distribution assumption.

Specifically, the distance of the KS test is defined as follows:

$$Dist_{\mathrm{KS}}(B,S) := 1 - D_{n_B,n_S}\sqrt{\frac{n_B n_S}{n_B + n_S}}, \qquad (1)$$

where n_B and n_S are the numbers of samples for two pixel blocks $B_{2\times2}$ and $S_{4\times4}$; D_{n_B,n_S} is the maximum distributional distance between the two pixel blocks. This distance, also called the test statistic is defined as follows:

$$D_{n_B,n_S} := \sup_x |F_{B,n_B}(x) - F_{S,n_S}(x)|, \qquad (2)$$

where $F_{B,n_B}(\cdot)$ and $F_{S,n_S}(\cdot)$ are empirical (cumulative) distribution functions of $B_{2\times2}$ and $S_{4\times4}$; sup is the supremum.

In addition to the above two criteria, to better preserve the continuity of the brightness of the original image, it is reasonable to take the surrounding neighborhood of a block into consideration when we search for a matching super block. However, if too many neighborhoods are taken, not only the computational complexity will be increased, but too much irrelevant information will be brought into the computation. Our experiments showed that using a single-pixel wide neighborhood of a block can effectively preserve the continuity.

As a result, the algorithm to synthesize the texture using self-similarity is described as follows.

For each block $B_{2\times2}$ in I_L:

1. Extract its 4×4 neighborhood from I_L. Denote it as N_B.
2. Scan the entire image I_L. For each 8×8 patch N_S whose center is a 4×4 super block $S_{4\times4}$:
   (a) If $Dist_{\mathrm{KS}}(B,S)$ is greater than a threshold T, disqualify $S_{4\times4}$ because its distribution is very different from $B_{2\times2}$.
   (b) Otherwise, compute $Dist_{\mathrm{L2}}(N_B,N_S)$.
3. Among the qualified super blocks, find the one S_{match} whose $Dist_{\mathrm{L2}}$ is the smallest. Place S_{match} on I_H at the location where $B_{2\times2}$ maps to. If there is no qualified super block, use bilinear interpolation on $B_{2\times2}$ to generate a super block and place it on I_H.

Because the KS test is used to process one-dimensional probability distributions, in our experiments, when we compute $Dist_{\mathrm{KS}}(B,S)$, we vectorize $B_{2\times2}$ to a vector of 4 components. In addition, to reduce the sensitivity to image variances, we extend the vector to 16 components with padding 0s. For the same reason, we also vectorize $S_{4\times4}$ and extend it to a vector of 64 components with padding 0s. Then we compute $Dist_{\mathrm{KS}}$ between the 16-component vector and the 64-component vector.

To speed up the algorithm, in the Step (2), instead of scanning the entire image, we could only scan a neighborhood of the block $B_{2\times2}$. In our experiments, we found that scanning a neighborhood of 80×80 produces very similar results to scanning the whole image.

After the texture-synthesized image I_H and the vector image I_V are generated, the edge regions of I_V and the non-edge regions of I_H are merged to obtain the final image I_F. Anti-aliasing is used on the borders between the two types of regions to provide a smooth appearance, and a high threshold is used to extract the edges, so that only the significant edges from I_V is taken to I_F.

4 EVALUATION

We have tested the effectiveness of the proposed method using the free and public images from https://www.peakpx.com, together with a free turaco picture by D. Demczuk from Wikipedia and a picture of a cat taken by ourselves. The process of our algorithm is illustrated in Figure 3.

We compared our methods with the latest version of two commercial software packages explained in Section 2: DCCI2x (2016) (Zhou et al., 2012) and ON1 Resize 2020 (Faber and Dougherty, 2007). The results are shown in the following figures. In our experiments, we downscaled high-resolution images to low-resolution images, and then used our method as well as the commercial methods to upscale the low-resolution images, so that we can compare the upscaled results with the ground truth. It should be noted that our goal is not to faithfully reconstruct the high-resolution image, but to synthesize visually pleasing textures.

From the figures, we can see that our method is able to provide very sharp and clear textures on these pictures of natural animals. We have also conducted quantitative comparisons between our method

Figure 3: (a) The original low-resolution image (b) Vectorized image I_V (c) Texture synthesized image I_H (d) Edge regions from I_V (e) Non-edge regions from I_H (f) Fused high-resolution image I_F - final result.

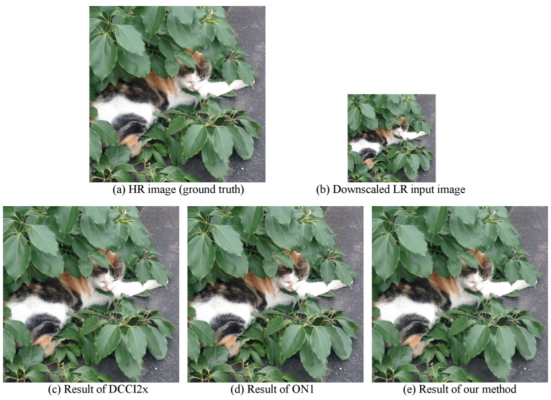

Figure 4: Comparison of results of SISR: (a) Original HR image (b) Downscaled LR input image (c) Result of DCCI2x (d) Result of ON1 (e) Result of our method.

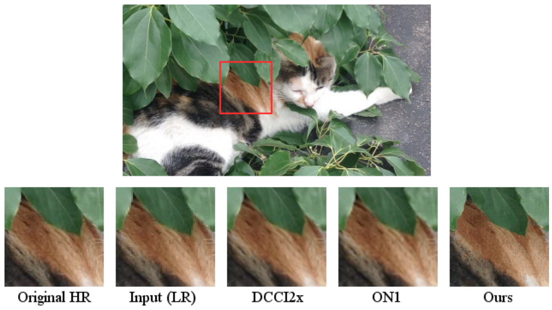

Figure 5: Comparison of results of SISR with enlarged details.

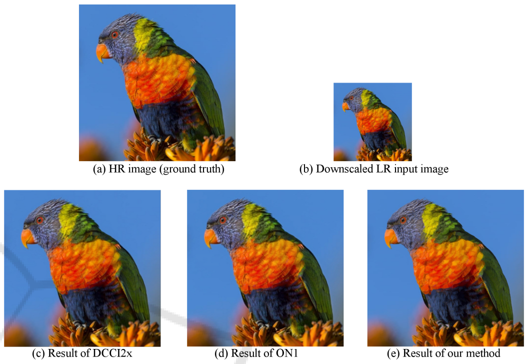

Figure 6: Comparison of results of SISR: (a) Original HR image (b) Downscaled LR input image (c) Result of DCCI2x (d) Result of ON1 (e) Result of our method.

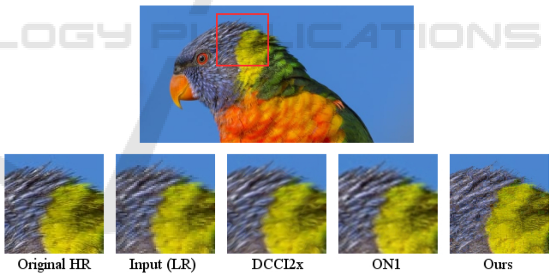

Figure 7: Comparison of results of SISR with enlarged details.

and the two commercial methods. We have chosen to use two image quality metrics: Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) (Mittal et al., 2012) and Perception based Image Quality Evaluator (PIQE) (Venkatanath et al., 2015). A BRISQUE model is trained on a database of images that have subjective quality scores. PIQE is not a trained model, but it provides local measures of quality as well as the global quality. Both metrics evaluates the image without any reference, i.e. without the original high-resolution image. This is reasonable, because in a real task of super-resolution, the original high-resolution image is unknown. We did not

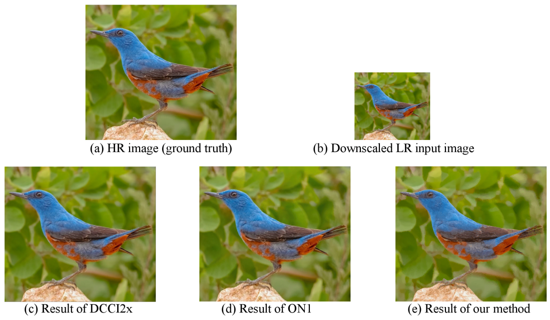

Figure 8: Comparison of results of SISR: (a) Original HR image (b) Downscaled LR input image (c) Result of DCCI2x (d) Result of ON1 (e) Result of our method.

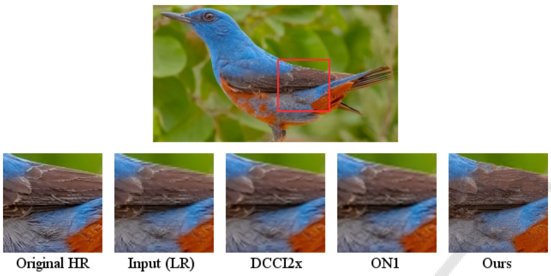

Figure 9: Comparison of results of SISR with enlarged details.

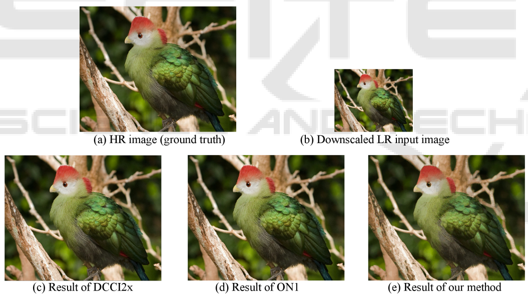

Figure 10: Comparison of results of SISR: (a) Original HR image (b) Downscaled LR input image (c) Result of DCCI2x (d) Result of ON1 (e) Result of our method.

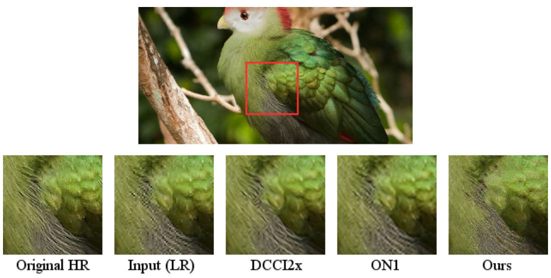

Figure 11: Comparison of results of SISR with enlarged details.

choose PSNR. As we discussed earlier, PSNR does not accurately represent the human perception of image quality (Sajjadi et al., 2017), especially when our images were generated with synthesized texture rather than reconstructed texture.

The quantitative results are shown in Table 1 and Table 2. Notice that for both metrics, a smaller score indicates better perceptual quality.

Table 1: Comparison of results of SISR using BRISQUE.

	DCCI2x	ON1	Ours
Cat with leaves	26.3153	29.7062	11.0070
Parrot	30.5968	36.7937	31.5079
Humming bird	37.8469	34.6144	23.6938
Turaco	24.4243	21.7355	26.2029

Table 2: Comparison of results of SISR using PIQE.

	DCCI2x	ON1	Ours
Cat with leaves	39.8146	40.6859	15.0370
Parrot	51.0482	49.8260	23.3640
Humming bird	41.9027	40.1564	34.3446
Turaco	34.5005	29.6453	22.9146

From the tables, we can see that when using BRISQUE, the two commercial methods outperforms our method slightly on two of the images, while our method outperforms them significantly on the other two images. When using PIQE, our method outperforms the other two methods significantly.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a two-phase novel single image super-resolution method for natural animal images. Our goal is to synthesize visually pleasing textures for the hair and feather of animals, as well as to preserve smooth and fine edges of the images. For the texture regions of the images, we propose a novel texture synthesis method based on KS test to produce fine and sharp textures. For the edge regions, we apply a traditionally successful image tracing method Potrace. The two intermediate results are then fused to produce the final result.

Experiments showed that our method outperforms two popular and successful commercial approaches in terms of synthesized texture quality. The synthesized texture from our method is sharp and clear, with little blurring effect. Quantitative comparisons also showed that our method generally outperforms these commercial methods.

In the future, we plan to conduct experiments with more images. We will also focus on the image vectorization and improve the performance of edge tracing.

REFERENCES

Chen, Y., Shi, F., Christodoulou, A. G., Xie, Y., Zhou, Z., and Li, D. (2018). Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 91–99. Springer.

Damkat, C. (2011). Single image super-resolution using self-examples and texture synthesis. *Signal, Image and Video Processing*, 5(3):343–352.

Faber, V. and Dougherty, R. L. (2007). Method for lossless encoding of image data by approximating linear transforms and preserving selected properties for image processing. US Patent 7,218,789.

Hatzigiorgaki, M. and Skodras, A. N. (2003). Compressed domain image retrieval: a comparative study of similarity metrics. In *Visual Communications and Image Processing 2003*, volume 5150, pages 439–448. International Society for Optics and Photonics.

Jacquin, A. E. et al. (1992). Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on image processing*, 1(1):18–30.

Jiang, K., Wang, Z., Yi, P., Wang, G., Lu, T., and Jiang, J. (2019). Edge-enhanced gan for remote sensing image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5799–5812.

Khatami, A., Khosravi, A., Nguyen, T., Lim, C. P., and Nahavandi, S. (2017). Medical image analysis using wavelet transform and deep belief networks. *Expert Systems with Applications*, 86:190–198.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690.

Lee, D., Sim, A., Choi, J., and Wu, K. (2016). Novel data reduction based on statistical similarity. In *Proceedings of the 28th International Conference on Scientific and Statistical Database Management*, pages 1–12.

Lee, D., Sim, A., Choi, J., and Wu, K. (2019). Idealem: Statistical similarity based data reduction. *arXiv preprint arXiv:1911.06980*.

Liu, B. and Ling, Q. (2020). Edge-guided depth image super-resolution based on ksvd. *IEEE Access*, 8:41108–41115.

Liu, H., Fu, Z., Han, J., Shao, L., Hou, S., and Chu, Y. (2019). Single image super-resolution using multi-scale deep encoder–decoder with phase congruency edge map guidance. *Information Sciences*, 473:44–58.

Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708.

Sajjadi, M. S., Scholkopf, B., and Hirsch, M. (2017). Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500.

Sayood, K. (2012). *Introduction to data compression*. Newnes.

Selinger, P. (2003). Potrace: a polygon-based tracing algorithm, http://potrace.sourceforge.net/potrace.pdf.

Stasik, P. M. and Balcerek, J. (2017). Improvements in upscaling of pixel art. In *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 371–376. IEEE.

Sun, J., Xu, Z., and Shum, H.-Y. (2008). Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Venkatanath, N., Praneeth, D., Bh, M. C., Channappayya, S. S., and Medasani, S. S. (2015). Blind image quality evaluation using perception based features. In *2015 Twenty First National Conference on Communications (NCC)*, pages 1–6. IEEE.

Wang, L., Xiang, S., Meng, G., Wu, H., and Pan, C. (2013). Edge-directed single-image super-resolution via adaptive gradient magnitude self-interpolation. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(8):1289–1299.

Wang, Y., Su, F., and Qian, Y. (2019). Text-attentional conditional generative adversarial network for super-resolution of text images. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1024–1029. IEEE.

Wohlberg, B. and De Jager, G. (1999). A review of the fractal image coding literature. *IEEE Transactions on Image Processing*, 8(12):1716–1729.

Xie, J., Feris, R. S., and Sun, M.-T. (2015). Edge-guided single depth image super resolution. *IEEE Transactions on Image Processing*, 25(1):428–438.

Yang, W., Feng, J., Yang, J., Zhao, F., Liu, J., Guo, Z., and Yan, S. (2017). Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing*, 26(12):5895–5907.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q. (2019). Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121.

Zhang, K., Zuo, W., and Zhang, L. (2018). Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271.

Zhang, K., Zuo, W., and Zhang, L. (2019). Deep plug-and-play super-resolution for arbitrary blur kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1671–1681.

Zhou, D., Shen, X., and Dong, W. (2012). Image zooming using directional cubic convolution interpolation. *IET image processing*, 6(6):627–634.