Evaluating Loss Functions for Illustration Super-Resolution Neural Networks

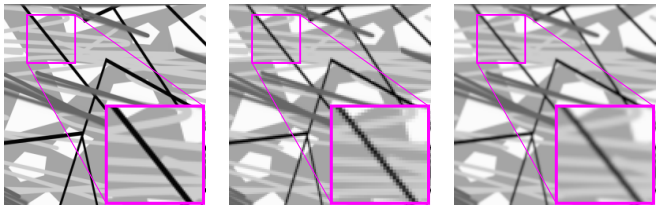
Raphael Nepomuceno and Michel M. Silva
Department of Informatics, Universidade Federal de Viçosa (DPI-UFV), Viçosa – MG, 36.570-900, Brazil
E-mails: {raphael.nepomuceno, michel.m.silva}@ufv.br

***Abstract*—As display technologies evolve and high-resolution screens become more available, the desirability of images and videos with high perceptual quality grows in order to properly utilize such advances. At the same time, the market for illustrated mediums, such animations and comics, has been in steady growth over the past years. Based on these observations, we were motivated to explore the super-resolution task in the niche of drawings. In absence of original high-resolution imagery, it is necessary to use approximate methods, such as interpolation algorithms, to enhance low-resolution media. Such methods, however, can produce undesirable artifacts in the reconstruct images, such as blurring and edge distortions. Recent works have successfully applied deep learning to this task, but such efforts are often aimed at real-world images and do not take in account the specifics of illustrations, which emphasize lines and employ simplified patterns rather than complex textures, which in turn makes visual artifacts introduced by algorithms easier to spot. With these differences in mind, we evaluated the effects of the choice of loss functions in order to obtain accurate and perceptually pleasing results in the super-resolution task for comics, cartoons, and other illustrations. Experimental evaluations have shown that a loss function based on edge detection performs best in this context among the evaluated functions, though still showing room for further improvements.**

I. Introduction

Nowadays, high-definition screens are becoming increasingly available due to advances in display technologies: statistics show a nine fold growth in the number of ultra-high-definition televisions from 2014 to 2019 [1]. To make the best use of high-definition displays, the availability of high-resolution imagery is desirable. However, while new content may be produced in high-resolution, previously recorded media may only be avaiable in low-resolution.

Enlarging images requires the use of some method to fill in pixels with unknown values. The most naïve method, referred to as nearest neighbor interpolation, is to repeat the intensity of the closest known pixel. However, this method introduces artifacts on the image, creating aliasing. A more elaborated method to fill in the unknown values is to interpolate the intensity based on the neighbors value and a polynomial function: *e.g.*, linear and bicubic interpolation. The choice of the interpolation method also impacts the perceptual quality of the result by potentially producing unwanted artifacts: in this case, there is a loss in the definition of the edges as the image becomes blurry [2]. Figure I exemplifies the artifacts introduced by the nearest neighbor and bicubic interpolation methods.



(a) Ground Truth (b) Nearest neighbor interpolation (c) Bicubic interpolation

Fig. 1. Examples of distortions produced by interpolation methods when applied to a low-resolution version of image 1a. Regions outlined in magenta are magnified to aid the comparison between methods. The edges in image 1b have aliasing artifacts not present in the Ground Truth. In the result of the interpolation method 1c, it is possible to observe the blur along the edges. The Ground Truth image is from the SYNLA dataset [3].

Single-Image Super-Resolution (SISR) is a classical task in computer vision of recovering a high-resolution image from a single low-resolution sample. It is inherently a ill-posed problem, as several possible solutions exist for a given low-resolution image [4]. Due to the details present in high-resolution images with high perceptual quality, the task is widely used in works involving high-definition television, medical, satellite and security imagery [5], [6].

One of the previous representative solutions for this problem employed a pipeline based on sparse coding pipeline with steps such as patch-extraction, dictionary look-ups and reconstruction in order to perform this task [7], [8]. However, it was shown that the process could be made more efficient, and potentially more accurate, by employing Convolutional Neural Networks (CNNs) with internal layers equivalent to such steps [4].

In fact, CNNs achieved memorable results in the task of creating super-resolution images [4], [5], [9], [10]. However, their application is focused on real world images. Therefore, one of the media type often reproduced in the high-quality screens, the illustrations, are underrepresented in the deep learning literature.

Illustrations, *i.e.* images from comic books, manga, cartoons and anime, are produced by an entirely different process from photography and generally focus only on salient details while abstracting the rest: edges are overly emphasized and complex textures are often replaced with flat regions or simplified patterns. In applications such as style transfer, these intrinsic difference causes the neural network to produce undesirable

artifacts likened to watercolor paintings, where noisy transition textures are generated instead of the expected simplified patterns [11]. The proposed work explores the application of deep learning techniques — specifically, CNNs — to the problem of single-image super resolution within the niche of illustrations, which potential applications include enhancing drawing, comics or — when combined with video processing techniques — animations [12].

The main contribution of this work is a quantitative, through the SSIM and PSNR metrics, and qualitative evaluation of four loss functions in order to determine which produces the most accurate and perceptually pleasing images in the SISR task.

II. Related Work

In this section, we present an overview of the Single-Image Super-Resolution (SISR) area by exploring remarkable works. For the sake of clarity, we divide the works in the section related to CNN architectures, quality metrics, and loss functions applied in the SISR problem.

A. Super-resolution neural networks

The concept of Convolutional Neural Networks has been around since the late 1980s [13], with its resurgence in recent years justified by advances in hardware and algorithms and by the favorable results exhibited in tasks such as image classification and object recognition [4], [6].

A breakthrough in the application of deep learning for single-image super-resolution was the SRCNN [4], which preprocessed images with the bicubic interpolation, forwarded their Y-channel through three convolutional layers. The idea is to first interpolate the image and then use the convolution operations to remove the created artifacts. The results achieved by the approach either surpassed or matched the state of the art at the time.

Shi *et al*. proposed the Efficient Sub-Pixel Convolutional Neural network (ESPCN) [5] architecture, which has shown superior results over the SRCNN while being more time efficient and with fewer learnable weights. This was enabled by the use of the sub-pixel convolution layer as the network output, allowing to avoid the bicubic interpolation step, which increased feature map sizes while not producing an equivalent amount of additional information.

Deeper architectures, such as SRResNet [9], EDSR [10] and RDN [14], have been enabled by the use of residual networks [15], which address the vanishing gradients problem [16]. Haris *et al*. [17] proposed an iterative up and down-sampling method with units combining intermediate features to error maps calculated by upsampling the error in an internal input reconstruction pipeline.

B. Loss functions for image reconstruction

The loss function serves as the objective in the deep learning optimization process, guiding its training and, in image reconstruction tasks, determining how to compare a synthesized sample to its ground truth [18]. Given the differences between illustrations and photographs, we are interested in examining the effects that the choice of a loss function on the characteristics of the reconstructed images. The focus of this work is in the analysis of four loss functions: Mean Squared Error, Mean Absolute Error, the usage of the Structural Similarity Index Measure as a loss function [18], and, based on the emphasis on contours that illustrations commonly exhibit, a mixed loss function which accounts for image gradients through the Sobel operator [19].

The mean squared error (MSE) loss is considered a popular choice [18], being employed in the works we build upon [4], [5]. The use of mean absolute error (MAE) was proposed as an attempt to overcome limitations of the MSE, said to produce splotchy artifacts [18]. Despite the MAE providing improvements over the MSE, its results were said to be sub-optimal, leading to the exploration of other loss functions [18]. The structural similarity index measure (SSIM) [20] is a metric motivated by the human visual perception, which evaluates images accounting for perceived changes in structural information. Zhao *et al*. proposed the use of the SSIM as a loss function for image restoration neural networks.

Alternatives have been proposed in order to produce images with higher perceptual quality for humans. Johnson *et al*. proposed to use a feature extractor from a classifier, *e.g*., VGG-16 [21], to describe the ground truth and reconstructed image, and then calculate the distance between the feature maps. The authors demonstrate that this perceptual loss embed domain knowledge in the training process [22]. Ledig *et al*. [9] introduced the use of Generative Adversarial Networks (GANs) [23] for SISR in order to produce photo-realistic images.

Given that illustrations place emphasis on lines, a method that optimizes for that should intuitively perform better. The use of an edge detection operator provides another means of embedding such domain knowledge in the context of illustrations. To that end, we explored the mixed gradient error, which is composed by MSE and a weighted mean gradient error [19], calculated using the classic edge detection filter proposed by Sobel [24].

C. Evaluating perceptual quality

The structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR) are widely used metrics for quantitatively estimating the effectiveness of image reconstruction methods [4], [5], [18], [19]. However, their adequacy for the human perception has been questioned, with works exhibiting restored images with better perceptual quality despite lower metric scores [9], [22]. Thus, we also direct our focus in qualitative comparisons across loss functions.

III. Methodology

Our work consisted in evaluating the effects of the loss function in the training of CNNs for the single-image super-resolution task for illustrations. Each loss was evaluated by training our chosen neural network architecture from scratch using an illustration dataset, then performing quantitative and qualitative analysis on their outputs. In this section, we present

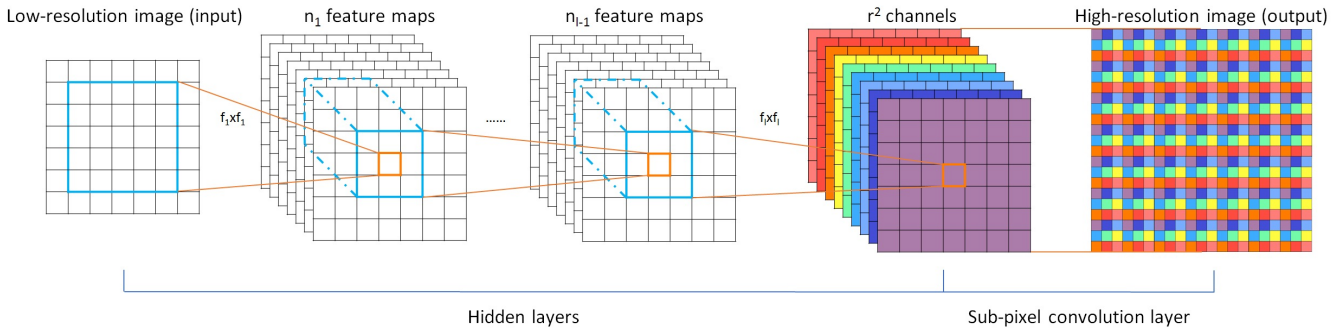


Fig. 2. Architecture overview of the ESPCN [5] network. Our application of the architecture differs from the original by inserting a normalization layer after the input and by using RGB images from end to end. Source: Shi *et al*. [5]

how we carried the training and the analysis of the CNN model.

A. Neural network architecture

We base our methodology upon the ESPCN [5] CNN architecture, presented in Figure 2. The choice of a shallower network architecture over the more recent deep residual networks [9], [10] is based on the goal of this work of making relative evaluations of loss functions, under the assumption that such relationships would be maintained if reevaluated on more complex networks. Thus, we default to training the simpler architecture, aiming faster experimentation cycles.

Following previous observations that networks trained on RGB images perform best on super-resolution tasks [4]. Thus, we modified the ESPCN architecture to operate from end-to-end on RGB images. To that end, we modify the number of input and output channels to 3, which raised the number of learnable weights of the network. We also added an additional non-parametric normalization layer at the start of the network to approximate the input into a standard normal distribution.

B. Loss Functions

We evaluate the impact of four loss functions in the training process of the CNN model based on the ESPCN architecture to create super-resolution images in the context of illustrations.

1) Mean Squared Error: The Mean Squared Error (MSE) for a high-resolution image y and its reconstructed counterpart $\hat{y}$ can be defined as:

$$MSE(\hat{y}, y) = \frac{1}{n} \sum_{p \in P} [y_p - \hat{y}_p]^2, \tag{1}$$

in which P is the set of the indices of the pixels and n is their amount.

Another metric to evaluate the quality of the reconstruction in the context of image reconstruction, is the Peak Signal-to-Noise Ratio (PSNR), that can be expressed as:

$$PSNR(\hat{y}, y) = 10 \cdot \log_{10} \left(\frac{1}{MSE(\hat{y}, y)} \right). \tag{2}$$

Analyzing the relation between MSE and PSNR, it can be seen that minimizing MSE maximizes the PSNR between y and $\hat{y}$. Therefore, as pointed in the literature [4], [22], conducting the training process using MSE leads to high values of PSNR. This MSE-PSNR relation motivates the designation of the MSE as the default choice [18] for image reconstruction if one considers the PSNR a suitable proxy for the human assessment of perceptual quality.

2) Mean Absolute Error: The use of the Mean Absolute Error (MAE) has previously been proposed as an attempt to reduce the artifacts introduced by the MSE loss [18]. Differently from the MSE (Equation 1), the errors are weighted uniformly in MAE formulation, as follows:

$$MAE(\hat{y}, y) = \frac{1}{n} \sum_{p \in P} |y_p - \hat{y}_p|. \tag{3}$$

3) Structural Similarity: By employing a loss function motivated by the human perception, one should expect yields in the perceptual quality of the generated images. To that end, we evaluate the use of the Structural Dissimilarity Index Measure (DSSIM) loss function [18] derived from the Structural Similarity Index Measure (SSIM) metric, defined as:

$$DSSIM(\hat{y}, y) = \frac{1 - SSIM(\hat{y}, y)}{2}. \tag{4}$$

where $SSIM$, in turn, is defined for a window of y and $\hat{y}$ as:

$$SSIM(\hat{y}, y) = \frac{(2\mu_{\hat{y}}\mu_y + c_1)(2\sigma_{\hat{y}y} + c_2)}{(\mu_{\hat{y}}^2 + \mu_y^2 + c_1)(\sigma_{\hat{y}}^2 + \sigma_y^2 + c_2)}, \tag{5}$$

in which c_1 and c_2 are stabilizing terms, μ_x and σ_x^2 are the mean and variance, respectively, for a given x.

4) Mixed Gradient Loss: With regards to the emphasis on lines exhibited by illustrations, we also explore the Mixed Gradient Loss (MixGE), which embeds the Sobel operator in order to guide the network to produce sharp edges which are close to those of the ground truth [19]. The Mixed Gradient Error (MixGE) can be defined as:

$$MixGE = MSE(\hat{y}, y) + \lambda_G MSE(G(\hat{y}), G(y)) \ , \tag{6}$$

where the hyperparameter λ_G is a weighting factor and $G(y)$ represents the gradient magnitude yielded by the Sobel operator, defined as:

TABLE I
QUANTITATIVE EVALUATION OF THE BASELINE METHODS AND LOSS FUNCTIONS OVER DIFFERENT ILLUSTRATION DATASETS. HIGH VALUES ARE BETTER AND THE BEST ONE IS PRESENTED IN BOLD FACE.

Methods	**Danbooru2020**		**Manga109**		**SYNLA (Color)**		**SYNLA (Greyscale)**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bilinear	21.42	0.7806	17.52	0.6922	21.19	0.7475	21.27	0.7498
Bicubic	21.97	0.8026	18.01	0.7190	22.29	0.7977	22.37	0.7996
Baseline	23.08	0.8274	19.31	0.7726	22.50	0.7638	24.23	0.8493
Baseline-RGB	22.47	0.7878	18.73	0.7328	22.39	0.7660	23.73	0.8410
Ours (MSE)	23.99	0.8514	20.12	0.8040	23.14	0.7816	24.84	0.8663
Ours (MAE)	23.14	0.8402	19.07	0.7751	22.58	0.7831	24.34	0.8655
Ours (DSSIM)	23.12	0.8600	19.34	0.7949	22.37	**0.8013**	23.93	0.8751
Ours (MixGE, $\lambda_G = 0.01$)	24.61	**0.8708**	20.62	**0.8235**	**23.30**	0.7864	**25.35**	**0.8791**
Ours (MixGE, $\lambda_G = 0.10$)	24.62	0.8700	**20.65**	0.8216	23.22	0.7858	25.31	0.8776
Ours (MixGE, $\lambda_G = 1.00$)	**24.66**	0.8707	20.63	0.8231	23.24	0.7839	25.17	0.8746

$$G(y) = \sqrt{G_X^2(y) + G_Y^2(y)}\ , \tag{7}$$

for the gradient maps G_X and G_Y defined in the X and Y direction, respectively.

IV. EXPERIMENTS

In this section, we provide details about the implementation, datasets, experimental evaluation, and its results.

A. Network configuration

To perform the experiments, we implement the modified version of the ESPCN CNN, presented in Section III-A and illustrated in Figure 2. The architecture is composed of 64 5×5 feature maps in the first layer, 32 3×3 in the second, and 27 3×3 in the last, totaling 31 thousand learnable parameters.

Each network was trained for up to 1500 epochs; the training process was halted after no improvements in the loss function were shown on the validation dataset for 100 epochs. The learning rate was set to $\alpha = 10^{-3}$ on the first two convolution layers of the network and 10^{-4} on the last, with no scheduling, as reported by the authors [5].

We trained three networks in order to observe the effects of the λ_G hyperparameter in the MixGE loss function.

B. Input and Output data

Our experiments were executed on RGB images set to be upscaled by a factor of 3. We used the central patch of each image as the high-resolution target, and prepared the low-resolution inputs by blurring the target with a Gaussian kernel of $\sigma = 1.0$ before downsampling with bicubic interpolation by a factor of 3.

C. Datasets

In an attempt to replicate the wide spectrum of illustrations, the following three datasets were used in this work.

1) Danbooru2020: A collection of approximately 4 million crowdsourced illustrations [11][1] of varying characteristics, ranging from line art to highly textured pictures. A subset of 40 thousand randomly sampled images were selected for use during the training phase, of which 8 thousand were used for validation at the end of each epoch. A second subset of 10 thousand images was used for testing. Due to hardware constraints for training, we used 96×96 central patches as the ground truth images.

2) Manga109: A collection of approximately 10 thousand comic pages drawn by professional manga artists in Japan [25], [26][2] used as a benchmark for SISR tasks [14], [17]. This dataset is characterized by having mostly grayscale images with finer details such as text. We used 288×288 central patches from this dataset as the ground truth images. A larger patch size was used in order to capture a meaningful section of the illustration images present in this dataset.

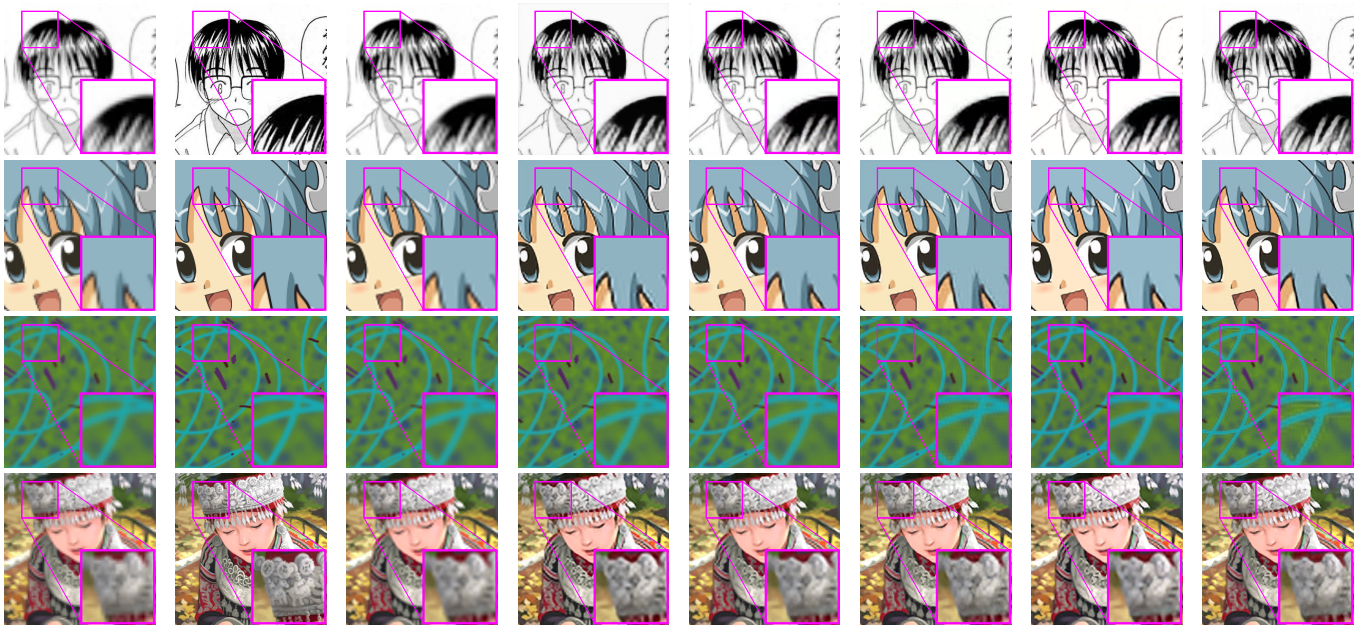
3) SYNLA: In order to further evaluate the generalization capabilities of the networks and find potential pathological cases, we also included a collection of synthetic line art images [3].[3] The dataset is available in two versions, each with roughly 2000 images, both which were used: one in greyscale, the other in color. As the original image sizes were smaller than the patch size 288×288, specified in Section IV-C2, and not an exact multiple of our scale factor, we used 192×192 central patches from this dataset as the ground truth images.

Danbooru2020 was used for training and testing, due to its wide range of illustrations in order to train networks able to generalize over style characteristics. Manga109 and SYNLA were used solely for testing.

D. Competitors

We compare our proposed models with the following approaches: (i) **Baseline**, the original ESPCN model [5], using as input the Y-channel of the image in YCbCr color space,

[1]Publicly available at https://www.gwern.net/Danbooru2020.
[2]Available upon request at http://www.manga109.org/en/.
[3]Publicly available at https://github.com/bloc97/SYNLA-Dataset.



(a) Input (b) Ground Truth (c) Bicubic (d) Baseline (e) Ours (MAE) (f) Ours (MSE) (g) Ours (DSSIM) (h) Ours (MixGE)

Fig. 3. From top to bottom rows, images are cropped samples of the following sources: 1) Manga109 [25], [26], © Ken Akamatsu 2) Wikipedia (https://en.wikipedia.org/wiki/File:Wikipe-tan_face.svg) 3) SYNLA dataset [3] 4) Set14 dataset [27]. Regions outlined in magenta are magnified to better visualize the impact of applying different SISR methods. Ours (MixGE) uses $\lambda = 0.01$ and the Baseline is the pre-trained Y-channel ESPCN model.

with pretrained weights in the DIV2K dataset [28], and MSE loss function; (ii) **Baseline-RGB**, the ESPCN model with RGB images as input, trained in the DIV2k dataset, MSE loss function, applying the same network configuration of our methods; (iii) **Bilinear** and (iv) **Bicubic** interpolation methods were also evaluated to establish a lower bound.

V. Results

In this section, we discuss the results obtained from training the neural networks with the loss functions discussed through the work, presented in Table I. We also present a few handpicked samples (Figure 3) for discussion as part of our qualitative analysis. Due to space limitations, the results from the Bilinear, Baseline-RGB and Ours (MixGE with $\lambda_G = 0.10$ and $\lambda_G = 1.00$) were omitted, since these results were outperformed by their variants, Bicubic, Baseline and Ours (MixGE, $\lambda_G = 0.01$), respectively.

Two images outside from the datasets listed in Section IV-C were included in qualitative analysis in Figure 3: the image in the fourth row is an illustration frequently used as a test case across super-resolution works. The level of details in this image reduces the gap between the result obtained from the baseline network (Figure 3d) and our best result (Figure 3h).

Regarding the modification applied to the ESPCN architecture, we observe that the addition of the non-parametric normalization layer at the start of the network helped the training process making the model to learn faster.

From experimental observations, the MAE caused the training process to reach a plateau after the least number of iterations among the studied functions, followed by the DSSIM. The training process persisted for the MSE and the MixGE until the upper limit of 1500 epochs.

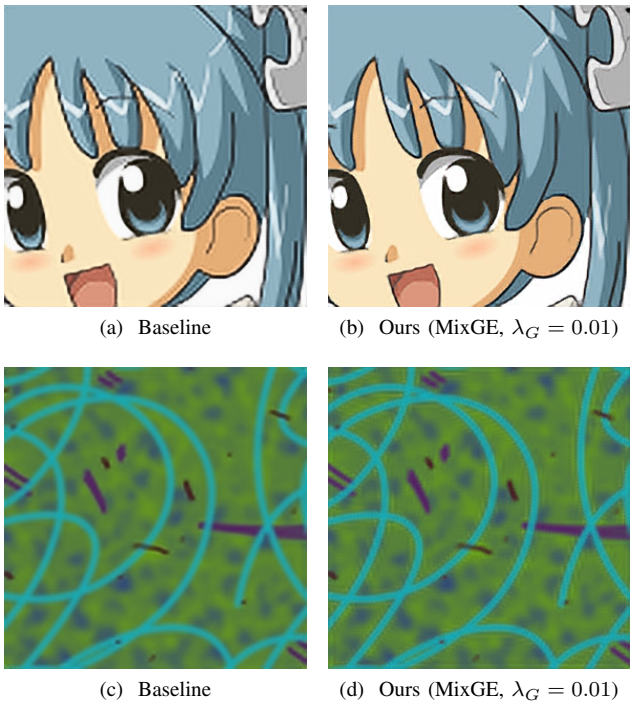
While the images produced by the network trained with the MAE loss have less noise than the one trained with the MSE, it has caused aliased edges in flat images, such as the second row in Figure 3e, motivating further exploration. It was observed that the DSSIM led the network to optimize for edge restoration at the expense of accuracy in color reproduction. In the second row of Figure 3g, the image has less noise than its counterparts, but the colorization of the character is visibly different. This can also be observed in the first row: the image has a slight red hue compared to its grayscale counterparts.

As seen in Table I, variations of the MixGE loss function have led to the best results in most recorded metrics. While it has been observed to reconstruct well in general scenarios, images restored from blurry low-resolution pictures, such as the image in the second row in Figure 3, have shown the highest incidence of noise among the trained networks, as seen in Figure 4, characterizing a pathological case.

Our experiments showed little to no impact in assigning three different weights (0.01, 0.10, 1.00) to the λ_G gradient component of the MixGE loss — as seen in Table I, metric results were close to each other and no discernible differences were observed in an analysis of the reconstructed images.

VI. Conclusions

Through the analysis of the experimental evaluation, we observed significant improvements in the super-resolution task by applying the domain knowledge in the loss function of a neural network. Within the context of this work, the knowledge



(a) Baseline (b) Ours (MixGE, $\lambda_G = 0.01$)

(c) Baseline (d) Ours (MixGE, $\lambda_G = 0.01$)

Fig. 4. Comparison of the photography-centric baseline against the MixGE loss, depicting a case that (b) the model produces smoother edges, and (d) a pathological case due to excessive production of noise.

that illustrations generally emphasize edges was used in order to search for a loss function more apt to consider this factor. However, the best network showed pathological behavior over certain types of images, *e.g.*, an originally blurred image.

Motivated by the fact that some loss functions perform better on some types of illustrations, which encompasses line art to drawings with rich textures, future works may benefit a stricter segregation by image types of the test dataset. There is an assumption that the quality characteristics and relationships of the loss functions are maintained if used on a larger network architecture. Verifying such assumption is planned as future work. Moreover, our work did not explore complex loss functions, such hybrid losses other than the MixGE, losses based on feature descriptors (*i.e.*, perceptual losses [22]), nor GANs, leaving room for improvement.

Acknowledgments. The authors would like to thank CAPES, CNPq and FAPEMIG agencies for supporting this project.

REFERENCES

[1] Statista, "Global 4k uhd tv unit sales from 2014 to 2019," 2019. [Online]. Available: https://www.statista.com/statistics/540680/global-4k-tv-unit-sales/

[2] R. E. W. Rafael C. Gonzalez, Digital Image Processing, 4th Edition. Pearson, 2018.

[3] bloc97, "SYNLA Dataset — image super-resolution for anime-style art." [Online]. Available: https://github.com/bloc97/SYNLA-Dataset

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," 2015.

[5] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," 2016.

[6] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," IEEE Transactions on Multimedia, vol. 21, no. 12, p. 3106–3121, Dec 2019. [Online]. Available: http://dx.doi.org/10.1109/TMM.2019.2919431

[7] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in 2008 IEEE conference on computer vision and pattern recognition. IEEE, 2008, pp. 1–8.

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

[9] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[10] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," 2017.

[11] Anonymous, Danbooru community, and G. Branwen, "Danbooru2020: A large-scale crowdsourced and tagged anime illustration dataset," https://www.gwern.net/Danbooru2020, January 2021. [Online]. Available: https://www.gwern.net/Danbooru2020

[12] Tyler, "Dandere2x — fast waifu2x video upscaling." [Online]. Available: https://github.com/akai-katto/dandere2x

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

[14] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2472–2481.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[16] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[17] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1664–1673.

[18] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," IEEE Transactions on computational imaging, vol. 3, no. 1, pp. 47–57, 2016.

[19] Z. Lu and Y. Chen, "Single image super resolution based on a modified u-net with mixed gradient loss," 2019.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[22] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, 2014.

[24] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," IEEE Journal of solid-state circuits, vol. 23, no. 2, pp. 358–367, 1988.

[25] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," Multimedia Tools and Applications, vol. 76, no. 20, pp. 21 811–21 838, 2017.

[26] K. Aizawa, A. Fujimoto, A. Otsubo, T. Ogawa, Y. Matsui, K. Tsubota, and H. Ikuta, "Building a manga dataset "manga109" with annotations for multimedia applications," IEEE MultiMedia, vol. 27, no. 2, pp. 8–18, 2020.

[27] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in International conference on curves and surfaces. Springer, 2010, pp. 711–730.

[28] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, July 2017.