Super Resolution Losses for Text Reconstruction

Simone Dutto
Politecnico di Torino
Turin, Italy
s257348@studenti.polito.it

Davide Fiorino
Politecnico di Torino
Turin, Italy
s256843@studenti.polito.it

Abstract

This study analyzes the effectiveness of various loss functions in the performance of Single Image Super Resolution (SISR) task, where the goal is to generate an high resolution image from a single low resolution one. The proposed method builds upon the NTIRE winning EDSR architecture, where we added the new loss functions built for generic SISR or specifically for text recognition. First the reconstruction capabilities of the model have been evaluated with generic images from the DIV2K dataset. Then we focused our attention on the recovery of images with text, where the sharpness of characters is the most important thing. To evaluate those performances we used the photos of civic numbers from Google's Street View SVHN dataset. Generated images have been evaluated both quantitatively with PSNR score and qualitatively through visual examination. In total 9 different losses, plus some combinations of them have been tested. The results we obtained prove the importance of task-tailored loss functions in the context of SISR.

Figure 1. Super Resolution x2 with Fair Loss

1. Introduction

Image super-resolution (SR) problem, particularly single image super-resolution (SISR), has gained increasing research attention for decades. SISR aims to reconstruct a high-resolution image (HR) from a single low-resolution image (LR). Generally, the relationship between LR and the original high-resolution image HR can vary depending on the situation, the degradation factors can be blur, noise or decimation. Many studies assume that LR is a bicubic downsampled version of HR. The attention, in these years focused on finding the best architecture to achieve higher PSNR score on images. It has been demonstrated that Deep Neural Networks, and in particular Convolution Neural Networks tends to perform really well for SR tasks [8, 9]. However, we saw a great work on choosing, optimizing or create new architectures. From VDSR[8] to SRGAN [11] there has been a lot of interest in peaking state-of-the-art performance to recover photo-realistic textures, working on heavily downsampled images or proposing different metrics. What we try to do in this paper is to propose an alternative strategy of facing Super Resolution, we would like to demonstrate the strength of choosing task-specific losses to push the model into maximizing super-resolution's effect on what we desire. We concentrate our experiments using the EDSR architecture [12], which is a enhanced deep super-resolution network and winner of the NTIRE2017 Super-Resolution Challenge [21]. Then, we try different losses and intuitions to achieve the best PNSR score on Text Images.

We decided to focus our attention on Text images because nowadays there are a lot of content which can stand loss to details/textures of the image but if the text is not readable the image loses its significance. For example, if we take into consideration subtitled video content, according to an Ofcom survey [14] in 2006 7.5 million people in

the UK (18% of the population) used closed captions: of that 7.5 million, only 1.5 million were deaf or hard of hearing. Another example are memes, they are captioned images, usually the image is recurrent and the real content is in the caption. These examples give the perspective on how important can be to retrieve text quality and how significant is to work on this particular application.

Finding the right and open dataset was a difficult task by itself, since our work is original in this particular declination; at the end we decided to use The Street View House Numbers (SVHN) dataset. This dataset fits our task because it is composed by real photos with characters in it. We prepared those datasets to be used properly for our scope by down-sampling (with bicubic interpolation) the images and defining train and test splits.

2. Related works

In recent years a wide variety of methods have been proposed for the task of image super resolution. Interpolation-based methods try to estimate the value of missing pixels directly from the value of their surrounding pixels. Those interpolation algorithms have low complexity and run very fast but their accuracy is also very low, with respect to other methods. Remaining in this category, very common is the usage of bicubic interpolation [7] or Lanczos re-sampling [3]. Given the simplicity of these algorithms they are often used as a comparison baselines to evaluate more advanced techniques. A second category of methods are those that tries to reconstruct the original image using prior knowledge to restrict the possible solution space. This provides an improvement on the overall image quality, especially on the details. In terms of disadvantages, these methods are very time-consuming and does not scale well when trying to reconstruct images at an higher scale factor. Two popular methods that base reconstruction on some kind of prior knowledge are: [19] and [25]. Some example-based approaches have also been proposed, like [4] which is based on nearest-neighbor and [22] which uses sparse dictionary learning.

Most of the above-mentioned methods fit in the category of classical computer vision (CV) approaches but the recent advent of deep learning and convolutional neural networks (CNN) has improved many CV tasks by huge margins. Super-resolution also has benefited from deep learning. In the following subsections we mention some famous CNN-based SISR methods and some studies on the usage of various loss functions to improve the capabilities of SISR in more specific tasks.

2.1. CNN-based approaches

The first work that proved the superiority of CNN-based approaches in SR was SRCNN [2], published in 2014 by Dong et al. The method uses a neural network with only 3 convolutional layers and an appropriate dataset to learn a mapping between LR and HR images. The model was also fast and outperformed traditional methods in terms of SSIM and PSNR. Since then, most of the research in SR has been centered on CNN.

After 2012 CNN architectures started to grow in depth: on this wave is inspired the VDSR approach by Kim et al. [8] which is based on the VGG-net architecture [18]. In VDSR the authors proposed a 20-layer architecture and a training procedure based on residuals to work with such a deep network. The approach also leverages the idea of using a single network to handle multiple scale factors efficiently.

In SRGAN [11] Ledig et al. focused on recovering finer texture details at larger scale factors. The authors noticed that an high PSNR in the reconstructed image, does not always imply a better perceptual quality, so they proposed a new architecture based on a generative adversarial network (GAN). The loss used is composed of an adversarial loss that tries to make the image closer to the natural images manifold and a content loss that pushes the solution towards a better perceptual quality. To evaluate perceptual quality the authors performed a mean opinion score test on 26 people.

EDSR [12] (Lim et al) is the approach upon which we based our study. Published in 2017 the work proposes a new optimized architecture for SISR. The network is based on SRResNet from [11], which is basically the SRGAN method without the adversarial loss component. This architecture is in his turn based on ResNet [5], but the authors removed unnecessary layers, like batch-normalization, that turned out worsening accuracy in this task. The authors also proposed a version for training multiple scales at once (MDSR). The model won the 2017 NTIRE Super-Resolution Challenge.

2.2. Task-specific losses

The content loss used in SRGAN was proposed by Johnson et al. [6] with the idea of increasing visual quality, not necessary PSRN or SSIM. In this last paper, the authors claims that using a per-pixel loss does not capture best the perceptual differences between output and ground-truth images. The proposed solution is to measure differences in a high level feature representation from a pretrained network (VGG-16).

In [15] the authors proposed a perceptual loss focused on reproducing realistic textures rather than reducing pixel-wise differences alone. The texture matching loss was extracted from the layers of a network pre-trained on the target texture. The authors tried different combinations of MSE pixel-loss, adversarial loss (from SRGAN) and texture loss.

In 2018 Anagun et al. [1] made an extensive study on the usage of different loss functions for SISR in their work named SRLibrary. The authors fixed a CNN-based architec-

ture and tested reconstruction accuracy with eight different losses. Some of these losses were parametric and some had no parameters. Some of the tried loss reported big improvements and robustness to noise in the low resolution image.

In a paper on multi-scale video frame prediction [13], Mathieu et al., proposed a strategy to sharpen the image prediction: to directly penalize the differences of image gradient predictions in the generative loss function. Following this lead they designed the Gradient Difference Loss (GDL), which was then combined with other two losses to achieve their goal.

3. Proposed method

In this section, we describe the proposed method. Since the main goal of this work is to analyze the performance of SISR for a specific task, with different loss functions, we decided to use a base architecture that performs well and is flexible in the usage of losses. The EDSR model [12] obtains state of the art performances on SISR, but it was only tested with two loss functions (L1 and L2). First we will illustrate the peculiarities of the EDSR architecture, then we will describe the various loss functions that we implemented.

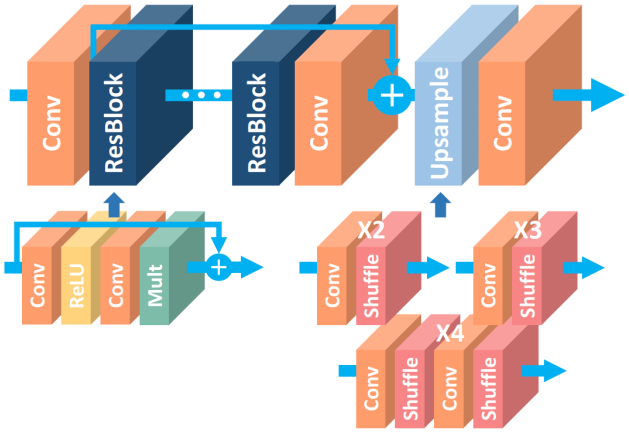

Figure 2. The EDSR architecture upon which we based our study.

3.1. The base architecture

The EDSR architecture is based on a modified version of the ResNet model. Compared to the original residual blocks, here batch normalization layers have been removed since they reduce the range flexibility of the network and this has been proven to worsen the performance substantially. For similar reasons, the ReLu activation layer after each residual block has also been removed, resulting in a lighter architecture.

Figure 2 shows the complete architecture: in the first part of the network a series of modified residual blocks are chained. Their function is to extract meaningful features to then increase the image resolution in the last upsampling layers. In this study, since we had a limited computational capacity, we decide to use the EDSR baseline version, which essentially has 1.5 million parameters, compared to the 43 million of full EDSR, by having fewer residual blocks and convolutional filters. In particular we are using 16 residual blocks and 64 convolutional filters. Figure 2 also shows a layer called "Mult" which performs residual scaling that has also been removed in the baseline version we used.

Upsampling is achieved through convolutional filters that increments the size of the output image, also known as de-convolution. Shi et al. [16] proposed a computationally efficient version of this operation called sub-pixel convolution. This is implemented by using the Shuffle layer that essentially rearranges the elements in a tensor in order to implement sub-pixel convolution. The number of layers in the upsampling part depends on the reconstruction scale, which we kept at x2 throughout all our experiments.

3.2. Loss functions

Having fixed a base architecture, the goal of our work is to evaluate the impact of loss functions in generic SISR performance and in the specific task of text reconstruction from an LR image. Other than classic L1 and L2 loss functions, we tried the perceptual loss from SRGAN [11], four losses from the SRLibrary paper [1], the GDL loss from [13], the WMSE loss by Zhang et al. [26] and a linear combination of them.

The L2 loss is the euclidean distance between the original high-res image and the reconstructed one in the pixel space, also known as mean squared error (MSE). While L2 is the most widely used loss function in image restoration, both Zao et al. [27] and the authors of EDSR reported that the L1 distance performed better in the SISR task.

The perceptual loss from SRGAN is a weighted sum of two components: an adversarial loss and a content loss. The adversarial component is the BCE loss between original and generated images computed by a trained GAN discriminator. As content loss we used the euclidean distance between the features extracted from a pretrained VGG-54 on the original and the generated images.

The losses proposed in SRLibrary have been studied with the goal of making the SISR task more robust to outliers, observations that do not comply with the general trend of sub-images. Below are reported the expressions of these four losses, where r is the residual between estimated and actual data and c is a tuning parameter of the function.
Cauchy loss:

$$\frac{c^2}{2} log(1 + (r/c)^2)$$

Charbonnier loss:

$$\sqrt{r^2 + c^2}$$

Huber loss:

$$\begin{cases} r^2/2 & \text{if } |n| \leq c \\ c(|r| - c/2) & \text{if } |n| > c \end{cases}$$

Fair loss:

$$c^2(|r|/c - log(1 + |r|/c))$$

Another loss that we implemented is the Image Gradient Difference (GDL) loss from [13]. GDL loss has been proposed in order to sharpen the image prediction by directly penalizing gradient differences between the prediction and the true output. The loss has also been studied to be computationally efficient by choosing the simplest possible image gradient that considers the neighbor pixel intensities differences. The GDL function between the ground truth image Y, and the prediction G(X) = $\hat{Y}$ is given by:

$$\sum_{i,j}(||Y_{i,j} - Y_{i-1,j}| - |\hat{Y}_{i,j} - \hat{Y}_{i-1,j}||^{\alpha} +$$

$$+||Y_{i,j-1} - Y_{i,j}| - |\hat{Y}_{i,j-1} - \hat{Y}_{i,j}||^{\alpha})$$

The last loss that we implemented is the Weighted Mean Squared Error (WMSE) from [26]. This loss has been studied specifically for the SISR task tailored for text recognition (OCR). The intuition is that high-frequency image details like high contrast edges play a more important role in OCR, and thus they should weight more. The WMSE loss between ground truth image I and predicted image $\hat{I}$ of size m by n is given by:

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}||I(i,j) - \hat{I}(i,j)||^2 \times f[grad(i,j)]}{mn}$$

where $grad(\cdot,\cdot)$ is the gradient magnitude map of the original image, which is obtained by using Sobel operator. $f[\cdot]$ is a certain function to convert gradient magnitude into weight; in our implementation we used the identity function.

The modified version of EDSR where we implemented these new losses and the Jupyter notebook to run the experiments can be found in our GitHub repository [17].

4. Experiments

4.1. Datasets

Firstly we used DIV2K to evaluate general purpose performance of our work, and then to test the effectiveness of text-specific losses we used SVHN. The DIV2K dataset is a high-quality (2K resolution) image dataset for image restoration tasks and consists of 800 training images (10 are used for validation) and 100 test images. SVHN is a dataset composed of real photos of Houses Numbers taken from Google Street View. We took 400 images for training (10 for validation) and 100 photos for test set.

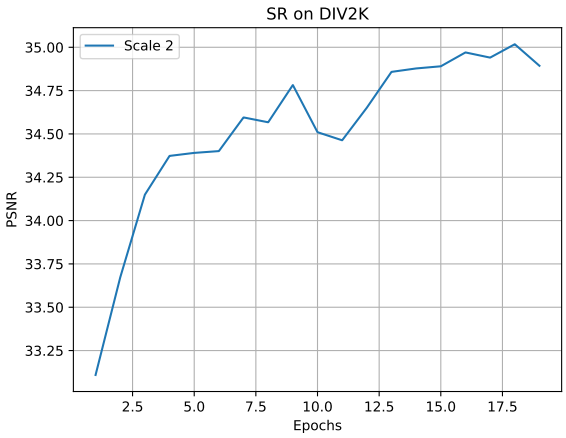

Figure 3. PSNR on Validation Set DIV2K Loss=L1

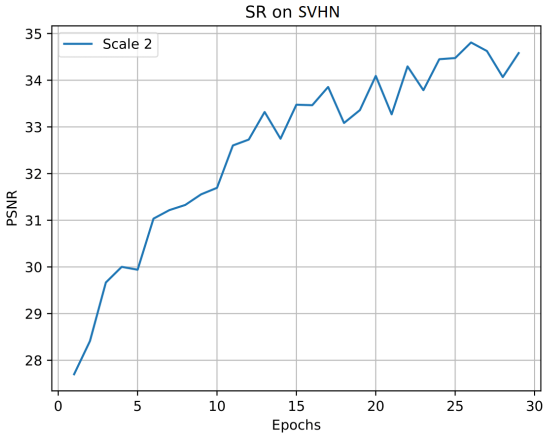

Figure 4. PSNR on Validation Set SVHN Loss=L1

4.2. Training Method

The training phase was lead by the EDSR training method [12] that we are using in single-scale mode. The architecture is trained from scratch, with ADAM optimizer [10], mini-batch sets to 16 and learning rate of 10^{-4}. Since we had limited hardware resources (Google Colaboraty) we had to keep the number of epochs relatively low (20 Epochs for DIV2K, 30 Epochs for SVHN). We noticed that the training phase is really stable on both datasets and the higher the number of epochs the higher is the PSNR achieved (see Figure 3, Figure 4).

This suggest that we can make assumptions on the performance of the various losses keeping reasonable training times. We can try different configurations, change optimizer parameters and mini-batch size. What we notice is that the default configuration of EDSR training is, by far, the most stable configuration.

To give an example, changing the Learning Rate to 10^{-3}

	L1	MSE (L2)	Huber	Charbonnier	Fair	WMSE	L1+0.1GDL	L1+0.01GDL
DIV2K	34.015 dB	33.821 dB	34.022 dB	34.015 dB	34.046 dB	33.581 dB	11.265 dB	34.022 dB
SVHN	36.560 dB	36.966 dB	36.536 dB	36.432 dB	36.952 dB	36.278 dB	37.119 dB	/

Figure 5. Performance on test sets with different losses

speeds up the convergence of the model in the first epochs but then the loss diverges to 10^7 in just 14 epochs.
Another example is that changing the STEP_SIZE for the scheduler has not an impact on our training phase because we are dealing with very few epochs compared to the standard for competitions (650 epochs is the number declared by the author of the GitHub repository [20]).

4.3. Difference between training on DIV2K and SVHN

Since there are only slight differences between the two testing procedures we decided to resume the difference in a separate paragraph.
The two datasets are different in the size of each image, in fact the image size typical for the DIV2K is 2000x1000 pixels, for SVHN is 150x150 pixels. For this reason: first, we set the patch size with DIV2K is 96x96 and with SVHN is 10x10. If we don't change the patches size we can't retrieve enough information from an image small like the ones in SVHN, therefore the model can't perform good. Second, we increase a little the number of epochs with SVHN to match the performance with the DIV2K.

4.4. Testing Method

The best performing model on validation set in the training phase is selected to be used in the test phase. Testing time is applied a data augmentation technique, called Geometric Self Ensemble [23]. This permits to apply 8 transformations to each entry(flip and rotation), generate an output from the 8 generated images, apply the inverse transformations and average them to give a result which is more resilient to noise and achieves best PSNR score.

4.5. Quantitative Evaluation on DIV2K/SVHN

For evaluation we trained our models with the different losses on DIV2K and SVHN.
DIV2K is used to have a base to make assumptions on how much a task-specific loss can impact the ability of a certain model to adapt to a task by only changing the loss.
What we can get from the Table 5 is that L1+0.1GDL is the best candidate for Text Super Resolution, in fact if we are considering general purpose test (on DIV2K) this loss is increasing irrelevantly the PSNR score with respect to L1, on the contrary with SVHN, where the database is composed by photo of characters, the L1+0.1GDL loss peaks at 37.119 dB (+1.52% better than L1).

Figure 6. Qualitative comparison between different losses.

This is a good result considering our limitation on train time and available hardware.

4.6. Qualitative Evaluation on DIV2K/SVHN

We understand evaluating a model with PSNR is very easy but it can be the cause of some issues regarding the recovery of fine-textures or high level of details. In fact from this paper [11] we notice higher PSNR does not always translate in better perceived quality.
From Figure 6 we notice that the more the GLD factor in the loss the more the text is highlighted. This is important because it gives an idea of what the GDL factor is doing to our model.
Obviously the photo in the right-bottom corner is not good with respect to HR version, but this photo could be pipelined into a text recognition model [24], and this pre-transformation can be very effective in increasing the accuracy of the text recognition model.

5. Conclusion

In this paper, we demonstrated the value of choosing carefully a loss for a task after having chosen the architecture. A lot of our considerations have to be further inspected since they are based on few epochs and we are sure task-

specific losses are an interesting field to work on.
Finally, it could be interesting to see Super Resolution in combination with other models (Text Recognition, Face Recognition etc.) and inspect how much SR can improve performances of others model and maybe one day it will become the de-facto standard in pre-processing images for computer vision.

References

[1] Yildiray Anagun, Sahin Isik, and Erol Seke. Srlibrary: Comparing different loss functions for super-resolution over various convolutional architectures. *Journal of Visual Communication and Image Representation*, 61:178–187, 2019.

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[3] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of applied meteorology*, 18(8):1016–1022, 1979.

[4] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Computer graphics and Applications*, (2):56–65, 2002.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[7] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

[8] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[9] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 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, 2017.

[12] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[13] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[14] Office of Communication UK Ofcom. Television access services review of the code and guidance. 2006.

[15] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4491–4500, 2017.

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[17] Davide Fiorino Simone Dutto. Edsr with additional loss functions. `https://github.com/SimoneDutto/EDSR`, 2020.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[20] thstkdgus35. EDSR-PyTorch. `https://github.com/thstkdgus35/EDSR-PyTorch/`, 2017.

[21] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 114–125, 2017.

[22] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 1920–1927, 2013.

[23] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016.

[24] Wenjia Wang, Enze Xie, Peize Sun, Wenhai Wang, Lixun Tian, Chunhua Shen, and Ping Luo. Textsr: Content-aware text super-resolution guided by recognition. *arXiv preprint arXiv:1909.07113*, 2019.

[25] Qing Yan, Yi Xu, Xiaokang Yang, and Truong Q Nguyen. Single image superresolution based on gradient profile sharpness. *IEEE Transactions on Image Processing*, 24(10):3187–3202, 2015.

[26] Haochen Zhang, Dong Liu, and Zhiwei Xiong. Cnn-based text image super-resolution tailored for ocr. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[27] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.