Text-guided Image Manipulation with Image Comparing & Sharp Region Enhancing

Yoga Fransiskus, Zhi Ye, Takumi Takada and Adnana Tudose
ETH Zurich, Switzerland

***Abstract*—Two main focuses in text-based image editing GAN is how we preserve regions in the image that are not related to the text description and how to generate higher quality images. In this project, we introduce two new methods: image comparing and sharp region enhancing (SRE). Image comparing improves the ability of the generator to preserve the image information by comparing the synthesized image with the original image. The comparison can indicate the likelihood of the text-affected region.We use this map to train the generator such that it can preserve information in unaffected regions. In addition, we also introduce the following techniques to improve the sharpness on the synthesized image. Naive Sharpness Loss (NSL) penalizes the blurriness on the image, gradient manipulation (GM) boosts the learning on the specific regions and sharpness-aware discriminator (SAD) feeds the discriminator with the blurred original images with labels as ”fake”, which encourages the discriminator to judge the blurred image as false and thus the generator is forced to generate sharper images to deceive the discriminator. We conducted experiments on the CUB bird dataset [1] and found out that our combined method outperforms the baseline models in terms of stability of text-irrelevant regions and the realism of generated images.**

I. Introduction

Since Reed et al. [2], Generative Adversarial Network (GAN) for image editing with text has achieved tremendous progress. Previous studies introduce ManiGAN [3], which includes a text-image affine combination module (ACM) that combines regional-level image features with sentence-level features and a detail correction module (DCM) which improves the quality of the edited image. Previous papers focus on developing the better attention mechanisms [4] that are able to indicate the affected region. In [5], the word level embedding is used instead of the full sentence. In [6], channel wise attention and cosine similarity are used to compare word level embedding and image features. DWC-GAN [7] includes a single channel attention mask to preserve the other parts of the image unaltered. Meanwhile, Li et al. [8] avoid pooling and cosine similarity to preserve dimension during attention calculation.

With all the advancement in text-based image editing, preserving information not contained in the text description remains an unsolved problem. The synthesized image in [3] [6] [8] can preserve the shape of unaffected regions. However, there is discoloration in the unaffected region [5]. This problem arises because of the lack of supervision in the unaffected region. The ability of the generator of synthesizing the image comes only from the gradient computed through the discriminator. Thus the capability of generating the image is dictated by how good the discriminator is [8].

The other goal of text-based image editing is improving the image quality. Recent papers already produce high-quality images that can capture both general shape and texture [9] [10]. However, there is still room for improvement. When we look deeper into texture-rich regions (e.g. bird’s feather), the synthesized image lacks sharpness. The sharpness detail is likely to be set aside by the discriminator. In most cases, the discriminator is only focused on finding text descriptions in the image which does not contain texture information.

II. Models and Methods

A. Baseline and Dataset

We examine our proposed model’s performance based on the Fréchet inception distance (FID) [11] and Inception Score (IS) [12], which evaluates the quality of synthetic images, and compare it with Lightweight-Manipulation GAN [8] and ManiGAN [3]. We recompute the FID and Inception Score of both Lightweight-GAN and ManiGAN and compare it with our method. We use the same CUB bird dataset [1] as the one used in the Lightweight-Manipulation GAN.

B. Image Comparing

We propose to compute a binary mask method to correct these undesired modifications as in Figure 1. The difference between the generated image $I_{\mathbf{g}}$ and the original image $I_{\mathbf{o}}$ are computed to get a mask $(M = I_{\mathbf{o}} - I_{\mathbf{g}})$. Then we smooth, soft-threshold and invert it to generate a binary mask $(M_{\mathbf{b}} = f(M))$, where f is smoothing and thresholding. In $M_{\mathbf{b}}$, 1 indicates text-irrelevant regions and 0 represents text-relevant regions.

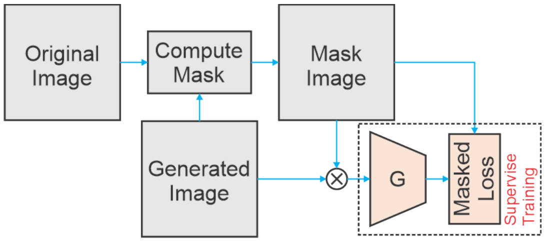


Figure 1. The work flow of image comparison method.

Using the binary mask, we can synthesize a new image that both preserves the modified region from $I_{\mathbf{g}}$ and the text-irrelevant region from $I_{\mathbf{o}}$ [13].

The mask we computed can be used to train the generator in a supervised manner to improve the performance of our model by preserving information in text-irrelevant regions from original images.

We use two different methods to create a binary mask. The two approaches which generated the binary mask are:

Histogram matching approach: We used a histogram matching (HM) approach to match the original and generated images color space. The mask will contain a lot of text-irrelevant regions, if we compute the difference between the generated image $I_{\mathbf{g}}$ and the original image $I_{\mathbf{o}}$ directly ($M = I_{\mathbf{o}} - I_{\mathbf{g}}$). This is because the original Lightweight GAN will slightly shift the color space in the text-irrelevant regions. Histogram matching will fix this problem by matching the histogram of the original and generated image. After histogram matching, the text-irrelevant regions, such as the background of birds, will be more similar and easier to distinguish. Then we smooth the mask by applying a Gaussian filter. Finally, we create the binary mask using thresholding. In practice, we set the threshold to 32.5 (in the scale of 255).

Pre-trained model approach: The model used for this approach is a Fully-Convolutional Network (FCN) with a ResNet-101 backbone [14] [15]. This network has been trained using a subset of COCO train2017 dataset [16] on 20 class labels, including the *bird* class (labelled as class 3). This helped to easily extract the background when predicting masks using FCN.

First, a number of pre-processing steps were applied before using the FCN model to perform semantic segmentation. Namely, pixel values of the real images used in the Lightweight GAN ranged from (-1, 1). The images were shifted to (0, 1) instead and further normalized with the ImageNet mean and standard deviation. The images were also resized to [224x224] in order to comply with the pre-trained network.

The FCN model outputs the probabilities of each pixel belonging to one of the 20 classes. For each pixel, the class with the highest probability was selected. The binary mask, $M_{\mathbf{b}}$, is therefore computed by assigning 1 to pixels whose class is 3 (denoting the bird class) and 0 otherwise. We finally inverted the mask in order to only capture the background of the image by computing $M_{\mathbf{FCN}} = 1 - M_{\mathbf{b}}$.

We integrate either of the two approaches presented above into the original Lightweight GAN architecture by including the mask into an MAE loss multiplied by the mask, either $M_{\mathbf{HM}}$ or $M_{\mathbf{FCN}}$, to enforce that the background is retained. This loss was subsequently added to the generator loss. The **MAE-Mask loss** $\mathcal{L}_{\mathbf{MAE-M}}$ was computed as follows:

$$\mathcal{L}_{\mathbf{MAE-M}} = \alpha * \mathcal{L}_{\mathbf{MAE}}(I_{\mathbf{o}} * M_{\mathbf{HM/FCN}}, I_{\mathbf{g}} * M_{\mathbf{HM/FCN}})$$

where we set α = 25 and $I_{\mathbf{o}}$ and $I_{\mathbf{g}}$ are the original images and the generated images, respectively. We increased the weight of the loss in order to be on the same scale as the other losses used in the generator.

C. Sharp Region Enhancing

Naive Sharpness Loss (NSL): The intuitive method to improve the sharpness is by applying sharpness loss. The sharpness loss will penalize the region with low sharpness. However, there are some regions that are supposed to be in low sharpness. For instance, the background may be blurred and this could be caused by the natural effect from the camera focus range. Therefore, we divide the region into two regions, a high sharpness region and a low sharpness region. The process to separate the region starts by calculating the sharpness using the Laplace filter. Then, we feed the sharpness map to k-means clustering with k (number of clusters) equal to 2. Finally, we create a sharpness label image ($I_{\mathbf{SL}}$) with value 1 in the high sharpness region and 0 in the low sharpness region. We mathematically define the sharpness loss as follows:

$$\mathcal{L}_{\mathbf{NSL}} = \alpha * BCE(I_{\mathbf{SL}}, tanh(\gamma * Laplace(I_{\mathbf{g}})))$$

where BCE is the binary cross entropy loss, $Laplace$ is the Laplace filter, and γ is a hyperparameter to tune the sensitivity of the $tanh$ function. The low γ moves the low sharpness region closer to its target value (0) and the high sharpness region farther from its target value (1), hence the loss function will neglect the low sharpness region and focus on training the high sharpness region.

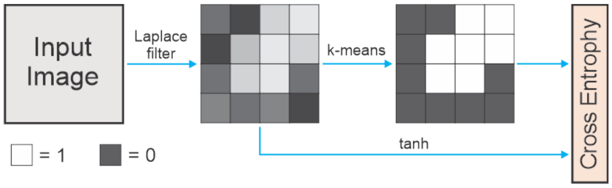


Figure 2. The calculation of naive sharpness loss method

Gradient Manipulation (GM): The alternative method of sharpness enhancing is by boosting the learning on a certain region of image. Because we want to speed up the learning in the high sharpness regions, we can multiply the gradient around this region by a scalar value. This is analogous with increasing the learning rate of certain sections of the generated image. The more robust proof of this method is presented in the appendix.

We make the gradient manipulator layer that directly passes the input to the output but modify the incoming gradient. The algorithm is described in Figure 3. We only use the gradient manipulator right after the image generated by the generator network.

Sharpness Aware Discriminator (SAD): While GM focuses on accelerating the change (gradient) of the sharp

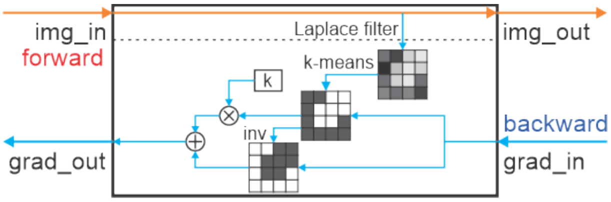


Figure 3. The architecture of the gradient manipulation method.

region and NSL focuses on enhancing the sharpness using an additional loss, SAD focuses on inferring the sharpness awareness of the discriminator. We modify the existing discriminator to discriminate the blurred and the sharp images. We apply the Gaussian filter to the original images from the given dataset and ask the discriminator to label it as ”FAKE”, as presented in Figure 4. This method infuses the sharpness loss to the discriminator, and therefore the additional loss function is not necessary. The discriminator will then give low output for the blurred generated image, and the generator will learn to enhance the image sharpness.

We use different sizes of blurring for the image. The blur method is box blurring with kernel varying from 5 to 9.

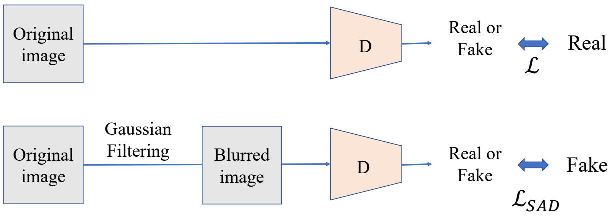


Figure 4. The architecture of the SAD method. The discriminator has to judge the original image as real, while it has to judge the blurred real image as fake.

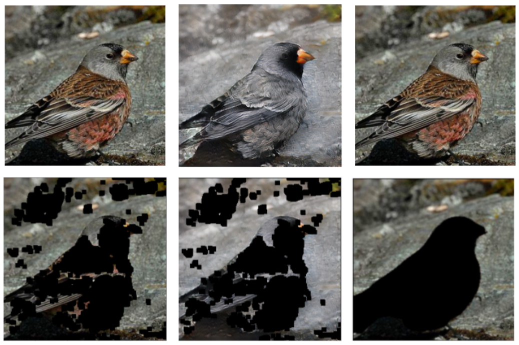
III. Results

A. Image Comparing

*1) **Histogram Matching**:* In HM, we compute the difference between the generated and original images. As shown in Figure 5, two masks were computed via the HM method, the black region in the mask indicating the text-affected regions. It can be seen that this method can distinguish between the text-irrelevant regions both in the bird and the background. However, at times, the mask failed to extract the text-irrelevant regions in background and misclassified the text-affected regions into mask.

*2) **Pre-trained model approach**:* In the pre-trained model (FCN) approach, we only use the original image to generate the binary mask, where the black area represents the bird class predicted by the model. In this case, the bird is clearly identified and separated from the background.

To show the HM and FCN masks and for ease of comparison between backgrounds, the second row masks the item of interest shown in the top row.



Original HM
Generated HM
Original FCN

Figure 5. Comparison between the HM mask applied to the original image and to the generated image by the Lightweight GAN model and the FCN mask applied to the original image only.

We then used a mask score method to evaluate our two approaches to compute the mask. In this score method, we apply the mask to the same pair of real and fake images to extract the text-irrelevant region or the background of them. Then we compute the MSE between them to get the $Mask_{score}$, we want to minimize it because it indicates the change of text-irrelevant regions.

$$Mask_{score} = \mathcal{L}_{\mathbf{MSE}}(Mask * \mathbf{I_o}, Mask * \mathbf{I_g})$$

We sampled 1000 images from our dataset to compute the mask scores and compared them in Table I.

Method	Score	Improve
HM with Lightweight	389.94	
HM with ours	243.72	37.50%
Pre-trained model with Lightweight	518.21	
Pre-trained model with ours	267.11	48.50%

Table I
Mask score [1]for the two mask methods applied in our model and the Lightweight GAN baseline.

We generated images from Lightweight GAN without our mask method as a baseline, then we generated images from our model with HM and the pre-trained model, respectively. The mask of the generated image was computed again using the same mask method used in training, and was applied to both the original image and the generated image to compute the $Mask_{score}$. We compute the difference between the mask score of Lightweight GAN and our method using the average mask score. As indicated in Table I, although the score of HM is lower than the pretrained model in both Lightweight GAN and ours, they are not comparable because

[1] Computed and averaged on 1000 generated images.

the mask used to compute the score is the same one used in training, which will cause bias in their own score. The improvement measures the percent decrease in mask score after adding HM and pre-trained model.

B. *Sharp Region Enhancing*

In this subsection, we show the result of how SRE method affect the quality of generated images. Figure 6 shows the comparison between generated image from original model, NSL method, GM method, and SAD method. All method consistently show an improvement over the original model. The image in the first row of Figure 6 shows that the original method looks dimmer because of the lack of sharpness. The sharpness of the feather is more pronounced in the GM, NSL, and SAD methods. In the second row, the sharpness of the GM method focuses only on the body, the sharpness of original method can reach the body and head, while the SAD method can create images with whole bird sharpness. Although the changes are not significant, from these three rows of images, it can be said that some areas, such as th ones around the bird's feathers, are sharper.

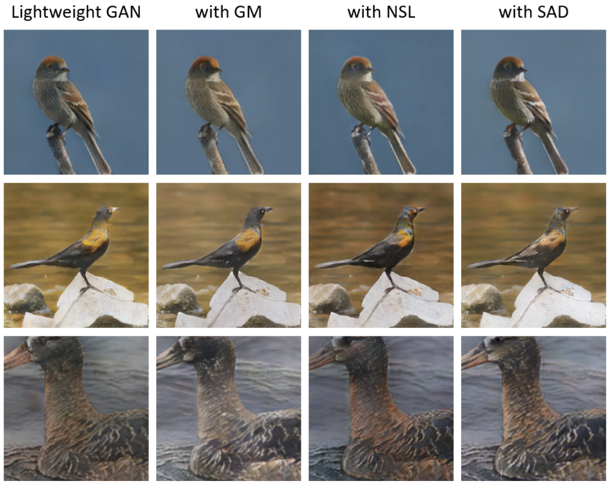


Figure 6. Comparison between generated images of the Lightweight-GAN, GM, NSL and SAD methods.

We also evaluated our methods by calculating the FID score. The FID score measures the similarity between the original dataset and the generated image dataset with the proposed method. The results are shown in Table II. These FID values in Table II are calculated after training 100 epochs. We use the same dataset as the one used for training the model to compute the FID score, while the authors of the original Lightweight GAN model use the evaluation dataset from the same domain, which causes the differences in the FID values.
From Table II, we can see that all proposed methods decrease the FID values. Notice that the lower the FID score is, the more realistic the generated images are. From this result, we can see that the SAD method contributes to improving the FID score.

The FID score is not enough to measure the sharpness improvement. We also implement a simple sharpness score to compare the sharpness result of the images. The sharpness score is defined as the absolute mean of the gradient image. The gradient image is the image passed through the Laplace filter. The result shows that all the proposed methods are able to improve the image sharpness. The highest sharpness result is produce by the SAD method with kernel 9.

From this experiment, we can see that the proposed method improves the sharpness without losing the realness.

Method	FID score	Sharpness score
Original Image	-	13.719
Lightweight GAN	4.075	10.710
w NSL α=1 γ=0.1	3.490	11.424
w GM k=1.5	3.539	11.475
w SAD, Kernel=5	4.068	10.965
w SAD, Kernel=9	4.068	11.934

Table II
Comparison of the FID score between models with and without the SAD method.

C. *Combined method*

Finally, we show the result of our method, which combines the image comparing method with pre-trained model and SAD approach (the size of blurring filter kernel is 9). Figure 7 shows the comparison between original images and generated images from our model and Lightweight GAN. From this figure, we can see that images from our method became sharper than those from Lightweight GAN, while the areas irrelevant to modification instruction, such as background, are remained.

We further present the Fréchet inception distance (FID) and Inception Score in Table III by comparing some baseline models with our method, where for both evaluation metrics lower is better.

Method	FID score	Inception Score
Lightweight GAN	4.07	5.076
ManiGAN	7.12	5.745
Ours	3.45	5.037

Table III
Comparison with baseline models

From this result, we can see that our method has better performance than ManiGAN and Lightweight GAN in terms of the FID score and the Inception score.

IV. Discussion

Image Comparing. In this binary mask method, we applied two different methods to compute the masks. Firstly,

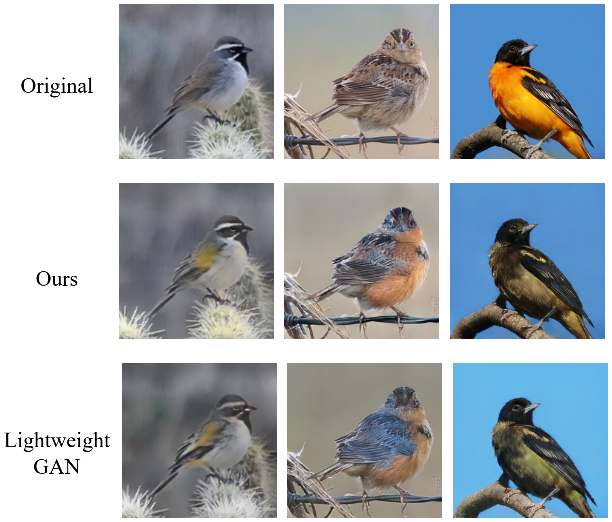


Figure 7. Image comparison between original and generated images from our method and Lightweight GAN.

we use a histogram matching to match the histograms of the generated and the original image before computing their difference. The performance of this approach is unstable. It is hard to tune a threshold, given the high number of images, to get a good mask. Sometimes the mask either extracts too many parts from the background or fails to extract the bird.

Then we considered using a deep learning model pre-trained on a large dataset. In practice, using a deep learning model pre-trained on a large dataset containing our object of interest will outperform other methods such as the pixel-wise comparison we considered. This approach has a further advantage, namely that it does not involve computing the difference between two images. With these methods, we did not need to include the synthetic image, but only created the mask using the real one. Thus, we could create good quality masks independent of the capacity of the generator to produce accurate synthetic images.

Sharpness Region Enhancing. As for sharpness enhancing, we introduce and compare three sharpness enhancing methods. As presented in the section before, all three methods are able to produce sharper images. The best method which improves the sharpness is the SAD method. On the other side, the realness presented by the FID score is better in both the GM and NSL methods. This may happen due to the saturation of the discriminator. With a bigger discriminator size, the discriminator will be able to learn an additional realness parameter (in this case, sharpness). Meanwhile, the GM and NSL are separate functions that do not rely on the complexity of the discriminator network. With these methods, the discriminator can focus on only improving realness.

Overall, as presented in Table II, the impact on the sharpness score is not significant. Also, it is not self-evident if applying the blurring filter on the entire image is the best way for the SAD approach since there are some images which are blurry from the beginning and being blurry does not necessarily mean it is a fake one, though we confirm through the experiments that even if we apply the blurring filter on the entire regions, the sharpness would improve to some extent.

We combined both methods and integrated them into the Lightweight GAN. As shown in Figure 7, our method improves the performance of Lightweight GAN by preserving more information from text-irrelevant regions such as the background and generating sharper images.

Limitations and future steps. There are a few limitations we observed during the development process. As mentioned before, the histogram matching approach is very sensitive to the threshold value that we assign in the beginning. We can make the threshold value more adaptive, so it is more robust to different contrast levels. Although this problem can be solved using the pretrained segmentation model, the segmentation model is not applicable for general text input. The pretrained model works well with the CUB birds dataset because the text only describes the appearance of the bird. In a more complex dataset such as Imagenet, the dataset has multiple classes, hence our method would require further modifications, and we would need to ensure it is pretrained on the objects of interest. Moreover, the pretrained model will, in most cases, create a mask that is as good as the model is. Therefore, in some cases, we could not correctly identify the bird in the CUB dataset even with the FCN-ResNet model.

The other limitation we observe is the incoherence between text input and the image result. This text-image incoherence may be caused by the weak image and text embeddings. The embedding networks used may weakly understand the relevance between the text and image parts. More powerful image-text embeddings are required to fix the problem.

V. Summary

In this paper, we propose two new methods to improve the performance of text-guided image manipulation tasks. One is image comparing, which aims to preserve regions in the image that are not related to the text description and the other one is sharp region enhancing, which tries to encourage the generator to generate sharper images. Each method can improve the performance of the model on its own, and find out that by combining those two methods, we can better preserve the regions unrelated to the text instructions and improve the sharpness of the generated images.

References

[1] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[2] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," 2016.

[3] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7880–7889.

[4] H. Tang, D. Xu, N. Sebe, and Y. Yan, "Attention-guided generative adversarial networks for unsupervised image-to-image translation," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[5] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5706–5714.

[6] B. Li, X. Qi, T. Lukasiewicz, and P. H. S. Torr, "Controllable text-to-image generation," 2019.

[7] Y. Liu, M. D. Nadai, D. Cai, H. Li, X. Alameda-Pineda, N. Sebe, and B. Lepri, "Describe what to change: A text-guided unsupervised image-to-image translation approach," 2020.

[8] B. Li, X. Qi, P. H. Torr, and T. Lukasiewicz, "Lightweight generative adversarial networks for text-guided image manipulation," *arXiv preprint arXiv:2010.12136*, 2020.

[9] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: Manipulating images with natural language," 2018.

[10] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. W. Taylor, "Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 304–10 312.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.

[13] A. You, C. Zhou, Q. Zhang, and L. Xu, "Towards controllable and photorealistic region-wise image manipulation," *Proceedings of the 29th ACM International Conference on Multimedia*, Oct 2021. [Online]. Available: http://dx.doi.org/10.1145/3474085.3475206

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[16] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

APPENDIX

A. *Gradient Manipulation Mathematical Proof*

The gradient manipulation method comes from scientific question on how to modify certain region in the generated image while preserving the other region. Similar concept has been extensively studied in image segmentation [[14]. In segmentation, the segment with low area tend to be ignored by the model because it contribute less to the total loss value. In [14], they modify the cross entropy loss to enhance the loss influence on low-area region using scalar multiplication. This technique will prevent the model from ignoring small area region. Similar technique can be applied in GAN. We can interpret a GAN as an segmentation network with different loss function. Instead of cross entropy loss we use discriminator network as a loss. However, modifying discriminator loss require mathematical trick. Unlike cross entropy, there is no intuitive understanding to increase the loss influence of certain region.

Using first order Taylor expansion, we can formulate the loss function as follow:

$$\begin{aligned}\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, k) &= \mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)\\ &\quad + f(k) * \partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}} * \Delta I_{\mathbf{ER}}\\ &\qquad + \partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{PR}} * \Delta I_{\mathbf{PR}}\end{aligned}$$

with $I_{\mathbf{ER}}$ is the enhanced region, $I_{\mathbf{PR}}$ is the preserved region, k is constant scaling factor of enhanced region influence ($k > 1$), $I^{\star}_{\mathbf{ER}}$ is the enhanced region with minimum loss value, $I^{\star}_{\mathbf{PR}}$ is the preserved region with minimum loss value, $\Delta I_{\mathbf{ER}}$ is $I_{\mathbf{ER}} - I^{\star}_{\mathbf{ER}}$, and $\Delta I_{\mathbf{PR}}$ is $I_{\mathbf{PR}} - I^{\star}_{\mathbf{PR}}$. We need to define function $f(k)$ to make sure the increase of loss function because of the enhance region. We defined $f(k)$ as follows:

$$f(k) = \begin{cases} k, & \text{if } \partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}} * \Delta I_{\mathbf{ER}} \geq 0\\ -k, & \text{otherwise}\end{cases}$$

To get the sign of k we need to know the sign of both $\partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}}$ and $\Delta I_{\mathbf{ER}}$. Because we do not have the exact value of $I^{\star}_{\mathbf{ER}}$ and $I^{\star}_{\mathbf{PR}}$, we will estimate it with the next iteration of gradient descent. We also assume that the change of gradient is not significant.

$$\partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1) \approx \partial\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, 1)$$

$$\begin{aligned}I^{\star}_{\mathbf{ER}} &= I_{\mathbf{ER}} - \eta\partial\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, 1)/\partial I_{\mathbf{ER}}\\ \Delta I^{\star}_{\mathbf{ER}} &= \eta\partial\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, 1)/\partial I_{\mathbf{ER}}\end{aligned}$$

with

$$\eta$$

is the learning rate. Using these equations, we get that

$$\begin{aligned}\partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}} * \Delta I_{\mathbf{ER}} &\approx\\ \eta(\partial\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, 1)/\partial I_{\mathbf{ER}})^2 &> 0\end{aligned}$$

The square term is always positive and learning rate is also positive. It implies that $\partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}} * \Delta I_{\mathbf{ER}} \geq 0$. Therefore, the final equation is equal:

$$\begin{aligned}\mathcal{L}_{\mathbf{GM}}(I_{\mathbf{ER}}, I_{\mathbf{PR}}, k) = \mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1) \\ + k * \partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{ER}} * \Delta I_{\mathbf{ER}} \\ + \partial\mathcal{L}_{\mathbf{GM}}(I^{\star}_{\mathbf{ER}}, I^{\star}_{\mathbf{PR}}, 1)/\partial I^{\star}_{\mathbf{PR}} * \Delta I_{\mathbf{PR}}\end{aligned}$$

This equation is equal to standard Taylor expansion with gradient boosting in the enhanced region. Applying this technique means that we do not have to change the loss function explicitly. We only need to multiply the gradient passing through the enhance region with some constant k. The result is also consistent with scalar multiplication in cross entropy loss [14]. The derivation of loss function in [14] will produce the same result.