GAN-based Face Mask Removal using Facial Landmarks and Pixel Errors in Masked Region

Hitoshi Yoshihashi[1], Naoto Ienaga[2] and Maki Sugimoto[1]
[1]*Graduate School of Information and Computer Science, Keio University, Yokohama, Kaganawa, Japan*
[2]*Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki, Japan*

Keywords: Inpainting, Face Completion, Generative Adversarial Networks, Face Mask, COVID-19.

Abstract: In 2020 and beyond, the opportunities to communicate with others while wearing a face mask have increased. A mask hides the mouth and facial muscles, making it difficult to convey facial expressions to others. In this study, we propose to use generative adversarial networks (GAN) to complete the facial region hidden by the mask. We defined custom loss functions that focus on the errors of the feature point coordinates of the face and the pixels in the masked region. As a result, we were able to generate images with higher quality than existing methods.

1 INTRODUCTION

Since 2020, when COVID-19 became a global problem, there has been an increase in the number of conversations with others while wearing masks. When communicating with others, to read the other person's mind, humans have a habit of paying attention to cues that appear on the face, such as around the eyes, mouth, and facial muscles. When wearing a mask, these important cues are partially lost. The mask does not hide the area around the eyes, so emotions can be predicted by looking at his or her eyes, but when the mouth and facial muscles are obscured, it is difficult to read detailed changes in facial expressions. It has been reported that when a person wears a mask, it becomes 10–20 % harder to convey a smile than when a person does not wear a mask (Tsujimura et al., 2020). If it is possible to supplement the area hidden by masks in images, humans can expect smooth communication even in situations where masks are worn.

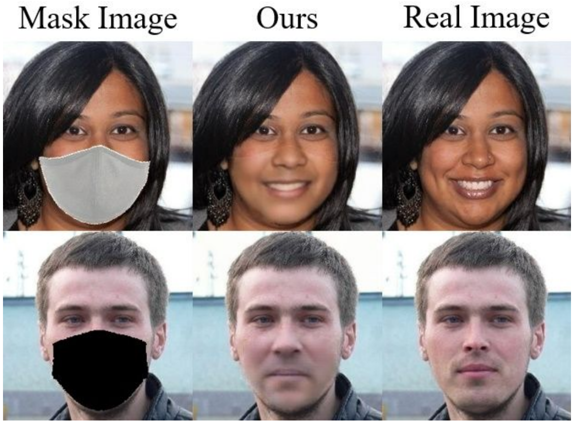

Figure 1: Examples of Mask Removal Results.

In this study, we propose a new method to complete the area hidden by a mask, which is an important cue for communication facilitation, in human face images.

2 RELATED WORKS

2.1 Inpainting

Inpainting is a technique used to repair scratches and holes in a part of an image. Inpainting attempts to recover the original image from the surrounded pixels of the missing area using a mask image that represents the missing area in black and white.

Inpainting has been used since the 1990s, but in 2012, a method using deep learning was first proposed (Xie et al., 2012). By using a denoising autoencoder, it outperformed conventional methods in removing white Gaussian noise in images and inpainting, but was unable to remove the noise of patterns that did not exist in the training data.

In DeepFill (Yu et al., 2018; Yu et al., 2019), generative adversarial networks (Goodfellow et al., 2014) were used. Generative adversarial networks

Yoshihashi, H., Ienaga, N. and Sugimoto, M.
GAN-based Face Mask Removal using Facial Landmarks and Pixel Errors in Masked Region.
DOI: 10.5220/0010827500003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 125-133
ISBN: 978-989-758-555-5; ISSN: 2184-4321

(GAN) consist of a generator that outputs a plausible image, and a discriminator that determines whether the input image is the correct image that comes from the training data (true) or the image created by the generator (false). Both networks compete with each other in the training process. The generator aims to create new data of the same quality as the training data. The discriminator aims to be able to perform binary classification based on the probability distribution of true and false images.

In the past, patch-based inpainting methods that cut and paste nearby pixels in the image were commonly used. However, DeepFill uses a learning-based inpainting method using GAN. The learning-based inpainting method learns a large number of pairs between the image that is correct (target) and the image that needs to be corrected (source), and it tries to predict a target image when a source image is given, using the trained model.

These methods are capable of repairing irregular scratches and holes in images. However, when there is a huge hole concentrated in one place in the image, such as in a face mask, the restoration may fail.

2.2 Face Completion

Studies of face completion, which attempt to repair missing regions in human face images, have also been conducted. For example, by using GAN, it is possible to repair a face image with randomly pasted squares (Cai et al., 2020) or to estimate the eye area of a person wearing a head-mounted display (Wang et al., 2019).

These methods are similar to learning-based inpainting, but they are designed to repair face images, and various parts of the human face are given as feature points. As a result, even if there is a large hole in one part of the image, the system can successfully repair it. However, this research did not aim at estimating face regions hidden by masks.

The state-of-the-art study to complete the masked region by using GAN (Yi et al., 2020) was conducted in 2020. The baseline of DeepFill with various custom loss functions was used, and it was able to inpaint the masked region. However, the quality of generated images could be improved, especially the skin color in the masked region.

3 METHOD

3.1 Inpainting using GAN

Pix2pix (Isola et al., 2017) is a GAN that learns the correspondence between a pair of images and, given one image, generates the other corresponding image. One of the features of pix2pix is that it uses conditional GAN (CGAN) (Mirza and Osindero, 2014). In CGAN, in addition to the noise vector, a condition vector containing information such as labels and text is given to the generator. In pix2pix, images are given as condition vectors. By referring to and comparing the vectorized values of the images, pix2pix brings the randomly given noise vector closer to the probability distribution of the correct image.

The pix2pix generator uses U-Net (Ronneberger et al., 2015) to perform precise pixel-by-pixel image transformation. The loss function of the pix2pix generator is the sum of D_{fake} (0–1 values returned by the discriminator for the true pairs) and the L_1 reconstruction error term.

$$G_{loss} = -\log D_{fake} + L_1 \times 100 \tag{1}$$

To minimize the value of G_{loss}, the generator aims to make D_{fake} close to 1 and L_1 close to 0. L_1 is assigned the average of the absolute error of the pixel values of the generated and real images.

The pix2pix discriminator, on the other hand, is similar to the discriminator used in general GAN, but it is designed to discriminate the authenticity of an image by using N × N data (patches), which is not the entire image but a portion of it. The loss function of the discriminator is calculated by using D_{fake} and D_{real} (0–1 values returned by the discriminator for true pairs).

$$D_{loss} = -(\log D_{real} + \log(1 - D_{fake})) \tag{2}$$

To minimize the value of D_{loss}, the discriminator aims to make D_{real} close to 1 and D_{fake} close to 0.

In this study, we propose a method based on pix2pix that completes pixels in the masked region for unknown mask images by learning pairs of mask and real (unmasked) images.

3.2 Custom Loss Function

In this study, two terms are added to the loss function of the generator (Eq. 1), and G_{loss} is redefined as follows, where λ_1 to λ_3 are hyperparameters.

$$G_{loss} = -\log D_{fake} + L_1 \times \lambda_1 + C_1 \times \lambda_2 + C_2 \times \lambda_3 \tag{3}$$

C_1 is a term to feed back the error of the face-feature point coordinates to the generator. For the generated

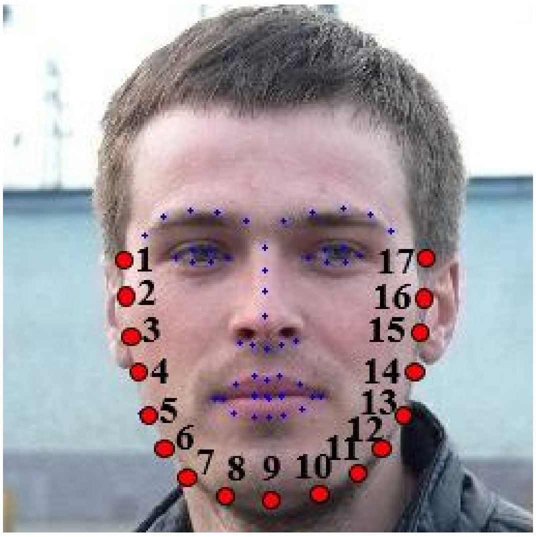

Figure 2: Example of Facial Feature Points.

image G and the real image R, the feature point coordinates of 17 points on the face contour (the points shown in red in Fig. 2) are obtained and their coordinates are fed back to the generator. C_1 is defined as follows, where G_n and R_n are the feature point coordinates of the generated image and the real image.

$$C_1 = \frac{\sum_{n=1}^{17}(|G_n - R_n|)}{17} \tag{4}$$

On the other hand, C_2 is a term to feed back the error of pixel values in the masked region to the generator. In the existing L_1 reconstruction error term, all pixels in the image are included in the calculation, but in C_2, only pixels in the masked region are included in the calculation. C_2 is defined as follows, where n_r, n_g, and n_b are the 256-level RGB pixel values at pixel n, and x is the number of pixels in the masked region.

$$C_2 = \frac{\sum_{n=1}^{x}(|G_{n_r} - R_{n_r}|^2 + |G_{n_g} - R_{n_g}|^2 + |G_{n_b} - R_{n_b}|^2)}{x} \tag{5}$$

4 IMPLEMENTATION

4.1 System Overview

In this study, we used pix2pix as a baseline and constructed a system as shown in Fig. 3. The light blue area is the part where new changes were made in this study.

First, the generated images are saved one by one as image files in png format so that we can process them using the OpenCV library and compare the generated images with the real images in the training data.

Next, we added two new custom terms to the loss function formula of the generator. One of the custom terms is the mean of the squared error of the pixel values limited to the masked region, and the other is the mean of the error of the feature point coordinates predicted from the face contour.

The flow of the system during training is as follows:

1. The generator reads a pair of a real image and a mask image of the same person.
2. The generator outputs an image with a probability distribution close to the real image from a randomly given noise vector.
3. The discriminator is given either a pair of real image and mask image (true pair), or a pair of generated image and mask image (false pair), and identifies whether the pair is true or false, and returns a value between 0 (false) and 1 (true).
4. The error between the value returned by the discriminator and the actual answer (1 for true, 0 for false) is fed back to the discriminator.
5. The value returned by the discriminator when a false pair is given is fed back to the generator.
6. The mean of the squared error of the pixel values of the generated image and the real image, and the mean of the error of the coordinates of the feature points of the contour, are fed back to the generator.

The above process is repeated for all images in the training data.

The flow of the system during testing is as follows:

1. The generator reads the mask image.
2. The generator generates a predicted image with a probability distribution close to the real image from a randomly given noise vector based on the pixel vector of the mask image.

The above process is repeated for all images in the testing data.

4.2 Predicting Feature Points on a Face

We used dlib (King, 2009) to obtain the feature point coordinates. Dlib is a library that can detect face regions and feature points for input images. By learning the correspondence between the face and the feature points, feature points of an unknown face image can be detected. However, when the dlib feature point detection program is executed on a face image in which a mask is worn, the coordinates of all feature points

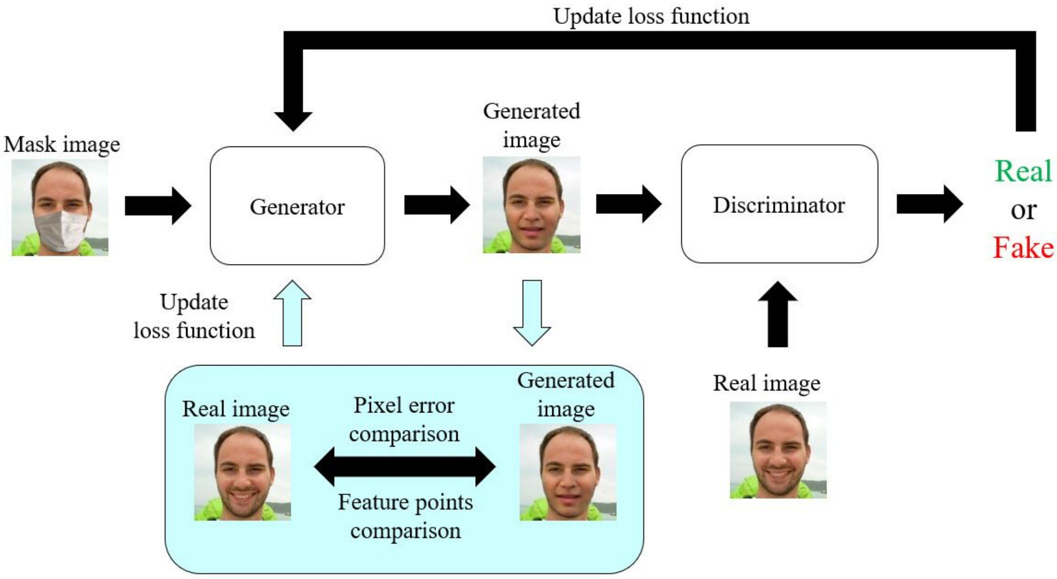

Figure 3: System Overview.

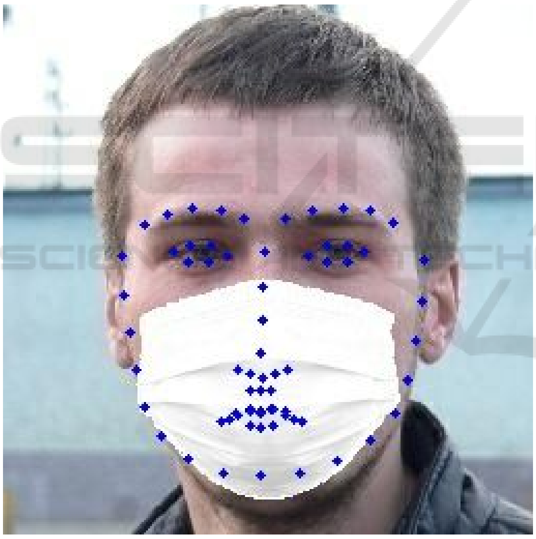

Figure 4: Example of Predicting Feature Points on a Mask Image.

cannot be correctly obtained, because the nose and mouth are hidden. The reason for this is that the model loaded by dlib is trained only on general face images without masks. Therefore, we retrained the model using face images with and without masks in equal proportions. The retrained model did not fail to predict feature points around the nose and mouth for mask images, as shown in Fig. 4.

On some of the images generated during the training process, some parts of the face were still hidden by the mask, so it was necessary to retrain the model to avoid failing to predict feature points.

5 EXPERIMENT

5.1 Procedure

We experimented to investigate whether the implemented system can accurately complete the masked region. The experiment was performed using the following procedure.

1. 7931 pairs of face images of a human wearing a mask and a human without a mask were prepared.
2. 5600 of them were given to the system for 300 epochs of training.
3. After the training was completed, the remaining 2331 images were given to the system for testing.
4. The patch-based inpainting method (OpenCV's inpaint function (Telea, 2004)), the original pix2pix method (Isola et al., 2017), and the state-of-the-art mask removal method (Yi et al., 2020) were tested with the same test data.
5. Quantitative evaluation metrics were calculated for the images generated by each method and compared.

The dataset was created by pasting face masks on real images using MaskTheFace (Anwar and Raychowdhury, 2020). Test dataset consists of 1400 images from Flicker-Faces-HQ (FFHQ) Dataset (Karras et al., 2019), 588 from the Karolinska Directed Emotional Faces (KDEF) Dataset (Lundqvist et al., 1998), and 343 from UTKFace Dataset (Zhang et al., 2017). The alignment of Stylegan2encoder (Karras et al., 2020) was applied to all the images in the dataset to correct the face positions and feature point coordinates. Each image file was resized to 256x256.

We also designed facial expression identification and quality evaluation experiments using some of the generated images and calculated qualitative evaluation metrics. Participants (8 men and 2 women, 10 total) were asked to see 10 images per method, including real images, and judge facial expression (neutral, happy, angry, sad, or surprise) and quality (7-point scale, 1 is the worst and 7 is the best).

5.2 Evaluation Metrics

For quantitative metrics, we used mean squared error (MSE), peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and learned perceptual image patch similarity (LPIPS) to evaluate the generated images. MSE was derived by taking the squared error of the pixel values at each pixel of the real image and the generated image in the masked region, adding them together, and taking the average.

These quantitative metrics were also used for determining how many epochs to train the model of our proposed method. As shown in Table 1, we decided to do training for 300 epochs, since both SSIM and LPIPS had the best values.

For qualitative metrics, we used human-rated quality score (HQS), accuracy, and duration, since correctly and quickly understanding facial expressions is important for communication. The HQS was derived by taking the average of the scores given by participants to 10 images of each method. The accuracy was derived by taking the average of the percentage of correct expressions selected by participants for 10 images of each method. The duration was derived by taking the average of the total time taken by participants to select the correct expression for 10 images of each method.

5.3 Results

The output results of the test data for each method are shown in Figs. 5, 6, and 7. In the case of our method, no mask pixels remained and no noise was generated, resulting in an image of high quality.

The results of quantitative metrics for each method are shown in Table 2. The unit for PSNR is decibel (dB). MSE, SSIM, and LPIPS have no units. In all four metrics, our method outperformed the others (For MSE and LPIPS, smaller is better). Our method improved MSE by 22.61 %, PSNR by 5.70 %, SSIM by 1.94 %, and LPIPS by 10.87 % compared with Yi's method.

The results of qualitative metrics for each method are shown in Table 3. For the HQS and the accuracy, our method had the next highest values after the real image. On the other hand, the duration was relatively uniform for all the methods except the real image.

Wilcoxon rank-sum test was conducted for HQS, and Tukey's multiple comparison test was conducted for accuracy and duration to investigate which methods had significant differences (5 % significant level). The results of p-values are shown in Table 4. For the HQS, there were significant differences between all method pairs except for Yi's-Ours. HQS of our method was significantly higher than that of Isola's original pix2pix, but was significantly lower than that of the real image. For the accuracy, there were significant differences only between Yi's and real image. For the duration, there was no significant difference among the methods.

6 DISCUSSION

6.1 Quality of Generated Images

Our method generated high-quality face images as shown in Tables 2 and 3 using the test dataset, which consists of synthesized mask images.

In addition to this, our method can be applied to real-world cases. To test the robustness of our method, we compared the results of each method using the dataset which consists of real-world images wearing face masks. As shown in Fig. 8, our method performed better than other methods in most cases. The results looked more blurred than the generated images by the synthesized mask image dataset (Fig. 5), however, our method showed a capability to be applied to real-world images. When the image was showing the side view of the face, our method was still able to complete the masked region. However, when the masked region was too large, our method could partially fail to complete the masked region.

In the generated images in Fig. 8, our method showed less error in facial color and geometry since our network was trained with custom loss functions which considered errors in the pixel level and the facial landmark level (focused on the masked region). On the other hand, the networks of other methods were trained with only whole image-based loss functions, such as L_1 reconstruction error and structural similarity (not focused on the masked region). Whether loss functions focused on the errors in the masked region may have made the difference between our method and other methods.

Table 1: Transition in Quantitative Metrics by Number of Epochs.

Metric	10	30	60	100	300
MSE	2594.464	**2230.605**	2249.282	2242.827	2319.373
PSNR (dB)	27.596	28.700	28.823	**28.843**	28.789
SSIM	0.901	0.916	0.922	0.924	**0.926**
LPIPS	0.0907	0.0626	0.0556	0.0522	**0.0459**

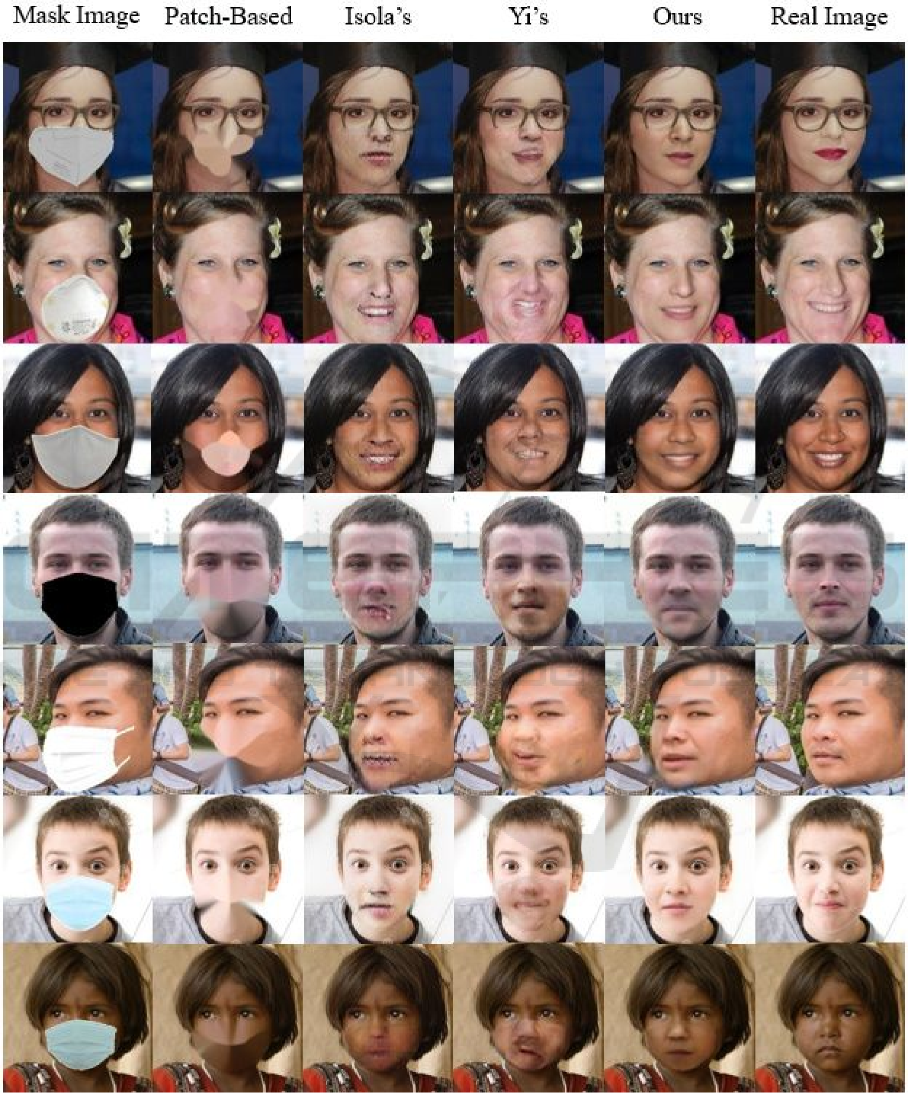

Figure 5: Examples of Mask Removal Results on FFHQ Dataset and UTKFace Dataset.

Table 2: Results of the Quantitative Evaluation Experiment.

Metric	Patch-based	Isola's	Yi's	Ours
MSE	6745.976	2685.080	2997.073	**2319.373**
PSNR (dB)	23.999	27.633	27.148	**28.789**
SSIM	0.907	0.896	0.908	**0.926**
LPIPS	0.0916	0.0547	0.0515	**0.0459**

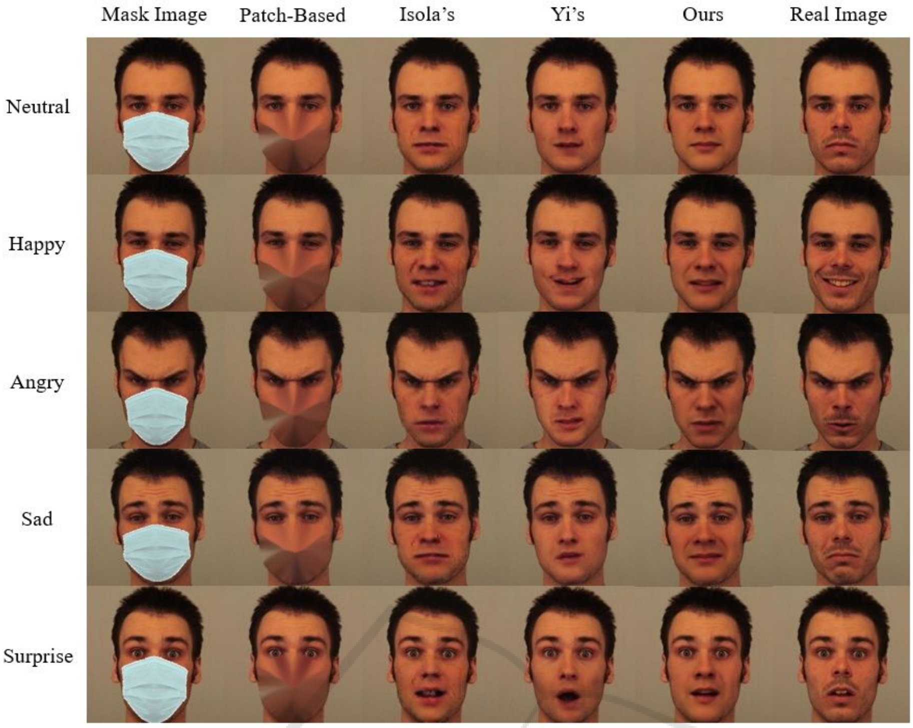

Figure 6: Comparison Among Each Facial Expression of the Same Man on KDEF Dataset.

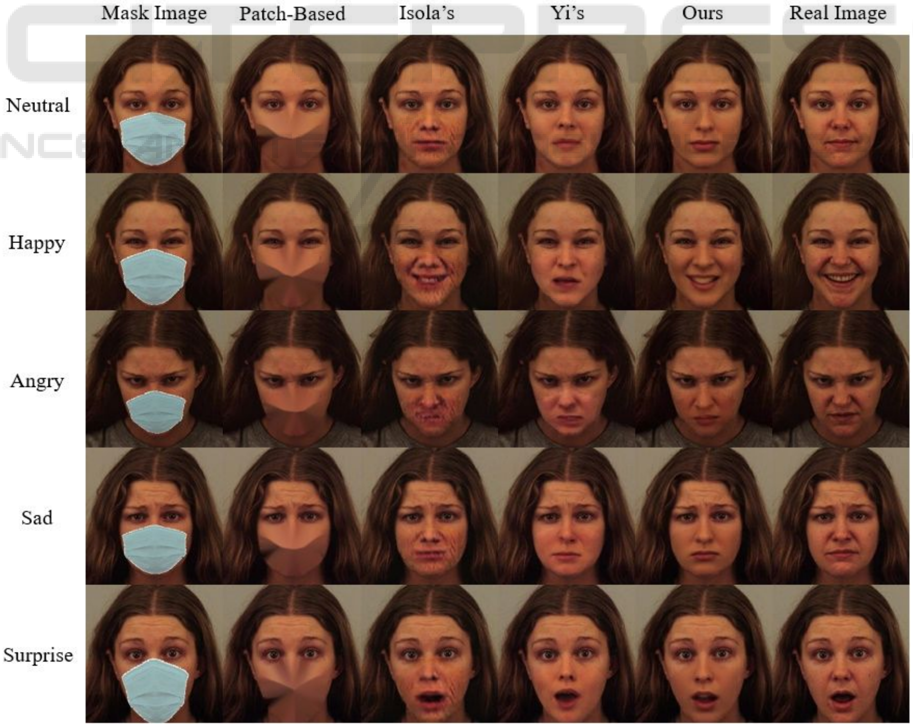

Figure 7: Comparison Among Each Facial Expression of the Same Woman on KDEF Dataset.

Table 3: Results of the Qualitative Evaluation Experiment.

Metric	Patch-Based	Isola's	Yi's	Ours	Real Image
HQS	1.41±0.49	2.70±0.79	4.54±0.96	**4.78±1.08**	6.04±1.04
Accuracy (%)	**96.00±5.16**	95.00±7.07	89.00±5.68	94.00±5.16	97.00±4.83
Duration (sec)	60.20±24.22	60.10±18.36	56.90±10.67	**54.10±7.58**	47.00±5.96

Table 4: P-values of Each Method Pair.

Method Pair	HQS	Accuracy	Duration
Patch-based and Isola's	$\mathbf{9.74 \times 10^{-5}}$	9.95×10^{-1}	1.00
Patch-based and Yi's	$\mathbf{1.08 \times 10^{-5}}$	5.80×10^{-2}	9.88×10^{-1}
Patch-based and Ours	$\mathbf{1.08 \times 10^{-5}}$	9.31×10^{-1}	8.93×10^{-1}
Patch-based and Real Image	$\mathbf{1.08 \times 10^{-5}}$	9.95×10^{-1}	3.00×10^{-1}
Isola's and Yi's	$\mathbf{5.30 \times 10^{-4}}$	1.40×10^{-1}	9.89×10^{-1}
Isola's and Ours	$\mathbf{1.95 \times 10^{-4}}$	9.95×10^{-1}	8.98×10^{-1}
Isola's and Real Image	$\mathbf{2.17 \times 10^{-5}}$	9.31×10^{-1}	3.08×10^{-1}
Yi's and Ours	6.43×10^{-1}	2.91×10^{-1}	9.93×10^{-1}
Yi's and Real Image	$\mathbf{7.22 \times 10^{-3}}$	$\mathbf{2.20 \times 10^{-2}}$	5.85×10^{-1}
Ours and Real Image	$\mathbf{1.90 \times 10^{-2}}$	7.57×10^{-1}	8.28×10^{-1}

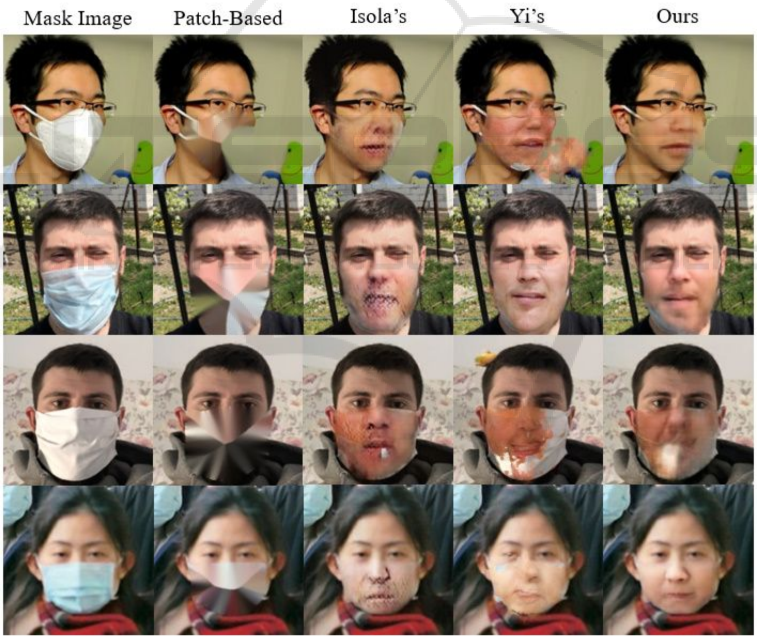

Figure 8: Examples of Mask Removal Results on Real-World Dataset.

6.2 Difficulty of Discriminating Facial Expressions

As shown in Table 5, the accuracy in the qualitative evaluation of our method was high for all facial expressions except for happiness. Yi's method also had difficulty in happy images. Many participants misidentified the happy images as neutral because the mouth was not smiling, as shown in Figs. 6 and 7.

Considering the accuracy of the patch-based inpainting method, which cannot use the mouth as a cue, it can be inferred that surprise is difficult to recognize based on the eyes alone. For other facial expressions, the accuracy was high with the patch-based inpainting method, suggesting that the cues around the eyes, especially the angle of the eyebrows, may have contributed to discriminating facial expressions.

Table 5: The Accuracy (%) of Each Facial Expression in Qualitative Evaluation.

Facial Expression	Patch-Based	Isola's	Yi's	Ours	Real Image	Average
Neutral	100	100	100	100	90	98
Happy	95	85	55	75	100	82
Angry	100	95	100	100	100	99
Sad	95	95	90	95	95	94
Surprise	90	100	100	100	100	98
Average	96	95	89	94	97	

7 LIMITATIONS

The limitations of this study are that the result may get worse if the given face image is wearing a mask different from the one used in the training dataset and that the method does not support video yet. Therefore, as future research, we need to develop a more robust program that can handle any mask input and do real-time face completion using video input.

8 CONCLUSION

In this study, we proposed a machine learning based approach to complete hidden parts by face masks in consideration of facial landmarks and pixel errors. We utilized a GAN-based model as a baseline and modified the loss function formula of the generator to calculate and update the errors in the coordinates of facial feature points and pixel values in the masked region, which enabled us to generate images with higher quality than existing methods.

REFERENCES

Anwar, A. and Raychowdhury, A. (2020). Masked face recognition for secure authentication. *ArXiv*, abs/2008.11104.

Cai, J., Han, H., Shan, S., and Chen, X. (2020). Fcsr-gan: Joint face completion and super-resolution via multi-task learning. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(2):109–121.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc.

Isola, P., Zhu, J., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4396–4405.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8107–8116.

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(60):1755–1758.

Lundqvist, D., Flykt, A., and Öhman, A. (1998). *The Karolinska Directed Emotional Faces*. Karolinska Institutet.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *ArXiv*, abs/1411.1784.

Ronneberger, O., P.Fischer, and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, volume 9351 of *LNCS*, pages 234–241. Springer.

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9.

Tsujimura, Y., Nishimura, S., Iijima, A., Kobayashi, R., and Miyajima, N. (2020). Comparing different levels of smiling with and without a surgical mask. *Journal of Comprehensive Nursing Research*, 19(2):3–9.

Wang, M., Wen, X., and Hu, S. (2019). Faithful face image completion for hmd occlusion removal. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct*, pages 251–256.

Xie, J., Xu, L., and Chen, E. (2012). Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 341–349.

Yi, J., Ud Din, N., Javed, K., and Bae, S. (2020). A novel gan-based network for unmasking of masked face. *IEEE Access*, 8:44276–44287.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. (2019). Free-form image inpainting with gated convolution. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 4470–4479.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). Generative image inpainting with contextual attention. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5505–5514.

Zhang, Z., Song, Y., and Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4352–4360.