Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057554

ARTICLE

Masked Face Restoration Model Based on Lightweight GAN

Yitong Zhou and Tianliang Lu*

College of Information and Cyber Security, People's Public Security University of China, Beijing, 100038, China

*Corresponding Author: Tianliang Lu. Email: lutianliang@ppsuc.edu.cn

Received: 21 August 2024 Accepted: 05 November 2024 Published: 17 February 2025

ABSTRACT

Facial recognition systems have become increasingly significant in public security efforts. They play a crucial role in apprehending criminals and locating missing children and elderly individuals. Nevertheless, in practical applications, around 30% to 50% of images are obstructed to varied extents, for as by the presence of masks, glasses, or hats. Repairing the masked facial images will enhance face recognition accuracy by 10% to 20%. Simultaneously, market research indicates a rising demand for efficient facial recognition technology within the security and surveillance sectors, with projections suggesting that the global facial recognition market would exceed US$3 billion by 2025. Therefore, finding a prompt and efficient solution to fix the masked face and enhance its accuracy has become a pressing issue that has to be resolved. Currently, the generative adversarial network has shown excellent performance in the field of image restoration, with high precision and good quality of restoration results, but it consumes a lot of computing resources. Based on this, this paper proposes a model architecture that uses the U-Net network to replace the generator in the generative adversarial network, and replaces all traditional convolutional layers with Depthwise Separable Convolutional (DWSC) to make the entire network lightweight. Ultimately, We utilise the Peak Signal-to-Noise Ratio (PSNR) value to assess the efficacy of the developed model. We select samples with occlusion levels ranging from 10%–15% and 20%–30%, yielding PSNR values of 35.51 and 30.33, respectively. In contrast, the PSNR values of the three predominant algorithms in image restoration—PM, ShiftNet, and PICNet—are all below 30, demonstrating the superiority of the model presented in this paper. However, the model presented in this work possesses certain drawbacks. This work employs solely black rectangles to replicate real-life occlusions. Future study should utilise tangible objects, like as sunglasses and masks, to directly imitate occlusions, so enhancing the accuracy of the restoration effect. The model presented in this study can be further expanded from image restoration to video restoration to investigate the potential for dynamic occlusion repair.

KEYWORDS

Deep learning; image restoration; GAN; U-Net network

1 Introduction

As computer networks continue to evolve, the need for automatic identity verification technology has grown. Numerous models have been developed utilizing advanced technologies in the domain of facial recognition. TV-CycleGAN [1] develops a multi-sensor facial detection model utilizing YOLO v3 and a common hot face database, grounded in a generative adversarial network, which markedly enhances recognition accuracy. LIPSNN (Light Intrusion-Proving Siamese Neural Network) [2] is an innovative lightweight neural architecture. The model, derived from the conventional Siamese network, incorporates two blocks: one for pre-training to extract facial features and another for refining false positive instances.

Nonetheless, current facial recognition methods concentrate exclusively on full facial photos. The issue of facial occlusion has impacted numerous domains. In security monitoring, obstruction of the face on the monitoring screen can lead to security vulnerabilities. The monitoring system is unable to effectively detect possible risks, may overlook suspects, or fail to respond to security incidents promptly. Facial recognition technology is progressively supplanting conventional passwords and fingerprint recognition in electronic payment systems. Should the user's face be obstructed throughout the payment process, the system will be unable to perform accurate verification, potentially resulting in payment failure or security vulnerabilities. Facial recognition in intelligent transportation systems can facilitate driver identity verification and infringement surveillance. Nevertheless, if the driver's face is obstructed within the vehicle, the system will be unable to accurately recognize it. Research indicates that when over 30% of the face is obscured, face recognition accuracy often declines by 20% to 30%. In certain instances, such as obstruction of the eyes or lips, the recognition rate may plummet to 40% or below. However, repairing masked facial images is challenging. The texture of the background, building, and other images is relatively simple, and there is a basis for repair, but the characteristics of the face are more complex; as long as there is a lack of a key point, the entire repair process will be hampered; the expression of the face, the angle of the moment of capture, the size and position of the occluded object, and so on, will all have a significant impact on the effect of the restoration [3].

Therefore, to solve the difficulties and problems mentioned above, this paper proposes a new model. The improvements in this paper are mainly carried out in the following three aspects:

1. The GAN (Generative Adversarial Networks) network suffers from the drawback of significant computing resource consumption, and the restored pictures produced by the U-Net structure are susceptible to blurring and structural deformation. The proposed model in this paper cleverly combines the discriminator in the GAN structure to address the issue of image inconsistency after processing by the U-Net network. The U-Net network excels at handling small targets and edge details, and it can achieve excellent training results with minimal training data, ensuring a lightweight model. The DWSC layer, a part of lightweight convolution, is particularly highlighted for its ability to significantly reduce the number of parameters and computational costs compared to ordinary convolution. This addresses the issue of excessive computation in network models and yields favourable outcomes in feature extraction.

2. This algorithm first preprocesses the clear face image for occlusion, and simulates the occluder with a black rectangle, which is random in size and position to better simulate the uncertainty of the occluded part in reality, and improves the likelihood that the algorithm will be put to realistic use.

3. In the final stage of experimental completion, the peak signal-to-noise ratio is introduced as an evaluation index, and the PSNR value of this algorithm is compared with the three mainstream restoration algorithms of PM, ShiftNet and PICNet, which proves the superiority of this paper's algorithm.

This section primarily references literature and data to substantiate the significance of this study and presents the innovative aspects of this model. The structure and content of the rest of this article are as follows. Section 2 combs through the existing work and relevant references in the field of image restoration in recent years and introduces the sources and limitations that inspired us to make algorithmic improvements. Section 3 is the basic methodology and implementation details of our approach, including the model construction for the algorithm. Section 4 provides the details of the parameter settings, environment configurations, dataset processing, and various results of our experiments, which validate the effectiveness of our model. Section 5 concludes with a summary of contributions and proposes future research directions to advance the field of image restoration.

2 Related Work

2.1 Image Restoration Based on Conventional Algorithms

The conventional face image restoration model utilizes concepts of image sample similarity and structural texture consistency, along with mathematical and physical principles, to develop an algorithmic model for repairing small damaged areas in images. This model is primarily categorized into two types of repair methods based on the type of partial differential equations and the type of samples [4]. The VFGL system, introduced by Bertalmio et al. [5], utilizes partial differential equations, grey levels, joint interpolation of vector fields, and fluid dynamics concepts to restore basic pictures. However, it is not suitable for repairing complicated textures. Criminisi et al. [6] proposed the Criminisi model based on sample classes to replicate the structural texture for small-area restoration, but calculating the similarity function is unstable and slow. Cao et al. [7] designed an image restoration model based on weighted priority and classification matching, which improves the restoration order and the selection problem of the best matching block, but the computation is large.

2.2 Image Restoration Based on Deep Learning

Conventional restoration techniques are ineffective for face photos that have intricate textures and require more sophisticated semantic processing. Researchers have utilized deep learning models to successfully fill in missing information and accomplish successful picture restoration, thanks to the ongoing advancements in deep learning. Wang et al. [8] proposed GFP-GAN that leverages rich and diverse priors encapsulated in a pretrained face GAN for blind face restoration. Pan et al. [9] utilized deep generative prior (DGP) based on GAN to learn from lots of natural pictures and fix missing parts in pictures, like colors or details. However, GAN network model is prone to issues such as unstable image restoration, image artefacts, and high computational resource consumption. Restormer, proposed by Syed et al. [10], and SwinIR, proposed by Liang et al. [11] both based on Transformer network model. This model can address the issue of limited convolutional layers that only capture local features by incorporating a self-attention mechanism, enabling the restoration of larger missing regions in an image. However, this method requires a large number of parameters and struggles to repair smaller visual regions in the image. U-Net is a fully convolutional network with a classical encoder-decoder structure [12,13]. Liu et al. [14] proposed a new coherent semantic attention layer (CSA) that preserves background information as well as accurately predicts missing regions, but its repaired images show artefacts and the model repeatedly calculates inter-sample similarity with a large number of parameters. Zeng et al. [15] proposed a context encoder, PEN-Net, based on pyramid networks. The network converges quickly and shows excellent vertical performance, but the restored image shows texture artefacts. Luo et al. [16] integrated face feature point prediction with the U-Net structure to enhance the knowledge of facial key points. However, they did not successfully utilize the diverse scale feature information, resulting in picture blurring. The MSA-Net model, presented by Qin et al. [17], incorporates multi-scale attention units into the U-Net structure to improve the capability of capturing profound characteristics from various receptive fields. However, it is afflicted by semantic ambiguity. Wang et al. [18] used a dynamic selection mechanism and relocatable convolution to dynamically select spatial convolution positions, but the restored image suffered from structural distortions. Liao et al. [19] employed U-Net architecture with a semantic attention propagation module to capture long-range semantic correlations in images. However, the lack of consistency in boundary information led to the blurring of the restored picture. The field of image restoration has developed rapidly in recent years, and many new and effective methods have been proposed. OKNet [20] enhances detail recovery through multi-level feature extraction and context fusion, making it effective for high-resolution images. However, it can yield unnatural results in severely missing areas and requires significant computational resources. DiffBIR [21], which could achieve realistic restoration results by leveraging the prior knowledge of pre-trained Stable Diffusion. Extensive experiments have validated the superiority of DiffBIR over existing methods, but it requires 50 sampling stepsto restore one low-quality image, which is computationally expensive. TextIR [22] leverages the recent CLIP model and designs a simple and effective framework that allows users to use text input to obtain the desired image restoration results. Because text input is easier to obtain and provides information with higher flexibility, the entire image restoration system shows superior performance.

2.3 Image Restoration Based on Other New Technologies

Recent advancements in technology have led to the emergence of many ways for rectifying partial photos across different domains. Yang et al. [23] proposed a new method to synthesize realistic image restoration training pairs using the emerging denoising diffusion probabilistic model (DDPM). DDPM transforms noisy input into a desired low-quality image and uses it to define the target data distribution, which is then denoised. The final low-quality image is synthesized with a given high-quality image and used to train a robust model for real-world image restoration tasks. Jaiswal et al. [24] proposed a physically integrated restoration network (PiRN), which introduced a physics-based simulator into the training process to address image distortion caused by atmospheric turbulence, and further introduced a PiRN with stochastic refinement (PiRN-SR) to improve its perceptual quality. The model provides state-of-the-art restoration in terms of pixel accuracy and perceptual quality.

Considering our technology and a comprehensive analysis of future deep learning advancements, this paper suggests an enhanced lightweight model grounded in the GAN network, despite the numerous emerging technologies in image restoration.

3 Methods

3.1 Overall Model Architecture

Fig. 1 displays the whole network architecture of the newly introduced model as described in this research. The network architecture is founded on the generative adversarial network as the fundamental framework, comprising two primary components: the generator and the discriminator. One of the components is the generator, which is built using the U-Net network architecture. The U-Net architecture has three main parts: the encoder, decoder, and skip connection. The encoder consists of an 8-layer Depthwise Separable Convolutional (DWSC) network, with a dropout layer introduced simultaneously in the final 5 levels. The decoder consists of a 7-layer DWSC network, with a dropout layer introduced simultaneously in the first 5 layers. Integrating a dropout layer into the DWSC layer can enhance the model's performance by mitigating overfitting, enhancing the model's resilience, and promoting feature sharing. Subsequently, the outcome produced by the generator is inputted into the discriminator with the initial picture, which comprises of four convolutional layers. Following repeated comparisons, the discriminator ultimately generates a feature map that determines the authenticity, marking the completion of the training process.

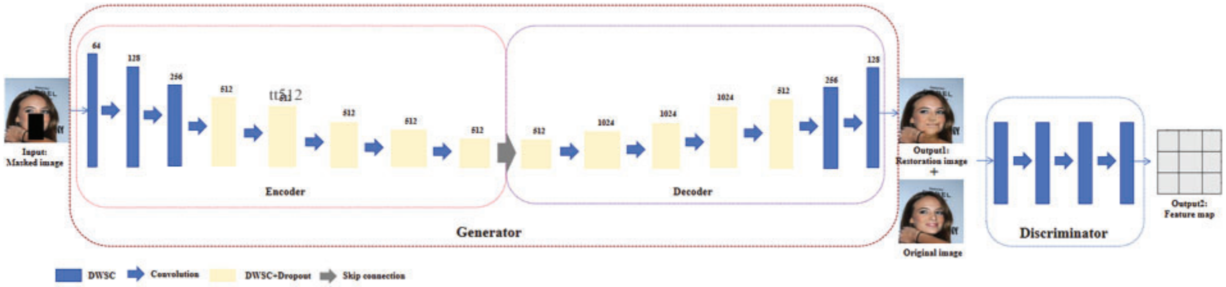

Figure 1: Overall model architecture

This image illustrates the structural diagram of the algorithmic model presented in this paper. The GAN network comprises two components: the generator and the discriminator. The generator component is structured as a U-Net, comprising an encoder with 8 depthwise separable convolution (DWSC) layers and a decoder with 7 DWSC layers, along with interconnecting skip connections. This image constitutes the foundation of the entire article and embodies the principal work.

3.2 *Depthwise Separable Convolution*

The U-Net network in this paper's model primarily consists of multiple convolutional layers. However, unlike traditional convolutional layers, the convolutional kernel in U-Net is static. This static kernel limits the model's ability to extract features effectively during feature extraction due to the absence of interaction between spatial and channel feature information. Consequently, this limitation leads to subpar image restoration results. Therefore, this article suggests employing depthwise separable convolution (DWSC) as a substitute for the original convolution in U-Net. In depth-separable convolution, dot convolution is used to improve how information flows between channels, which makes it easier for the model to express features. Furthermore, the use of DWSC creates a lightweight structure and reduces the model's computational burden.

Feature Map 1 employs M feature channels, utilizing M convolution kernels of size 3 × 3 to concentrate on the feature information within each channel. The M 3 × 3 convolution kernels will traverse each feature channel to generate M single-channel feature maps, as illustrated in Feature Map 2. These feature maps preserve the local information regarding the spatial position within the input feature map. Subsequently, the 1 × 1 convolution will utilize these M single-channel feature maps as input and execute point convolutions independently. The point convolutions employ a 1 × 1 convolution kernel at each spatial location, which assigns weights to the features of the M channels and produces a new feature map, referred to as Feature Map 3. Thus, the attributes at each position amalgamate the information from all channels, achieving efficient feature fusion. The proposed depth-separable convolution module is shown in Fig. 2. The proposed depth-separable convolution module is shown in Fig. 2. Afterwards, the optimised image features are passed backward for convolution, which provides a more effective feature representation and enhances the accuracy of image feature extraction by the backbone network [25].

For the Feature Map 1 with M feature channels, M 3 × 3 convolution kernels are used to focus on the feature information in each channel to obtain the single-channel Feature Map 2 with rich feature information, and then the Feature Map 3 with the information fusion of different channels is obtained by using the feature information at the same spatial position in different channels through the pointwise convolution of 1 × 1 × M.

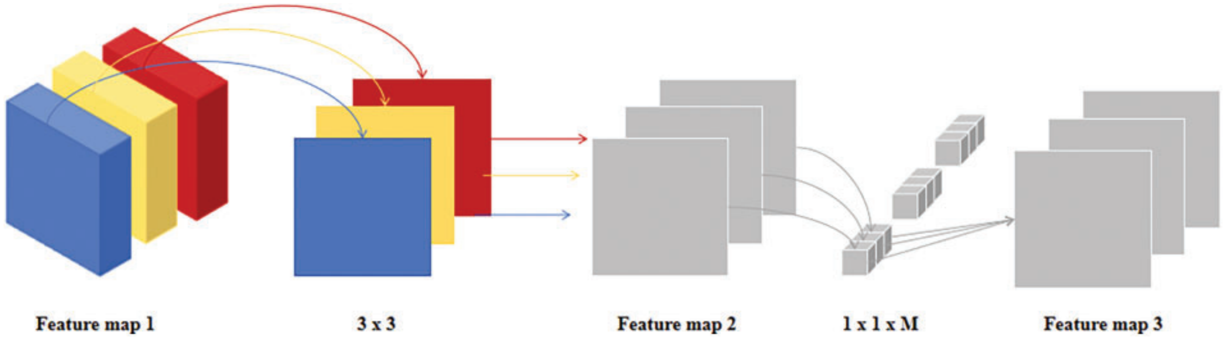

Figure 2: Depthwise separable convolution

3.3 Face Image Pre-processing

The majority of current public face image datasets consist of clear frontal photographs of faces. However, the image we need to fix is a face image with occlusions. As a result, we need to pre-process the face image before repairing it. As a first step, we generate a collection of randomly occluded face images to use as samples. This algorithm generates a randomly sized rectangle that is filled with black color. This rectangle's purpose is to act as an occluder, obstructing parts of the face. This is done to simulate the uncertainty of occlusion in real-life scenarios, such as when a hat, sunglasses, or a mask partially cover the face. The position at which the rectangle appears is also randomly determined. In the dataset introduction, all images are expected to have a treatment size of 256 × 256 pixels. To prevent overflow, the size of the randomly generated mask is adjusted. Two points, p1 and p2, are randomly generated with horizontal and vertical coordinates ranging from 0 to 156 pixels. The rectangle's width is randomly chosen between 50 and 100 pixels, and it is drawn using the thickness constructor.

After applying random black rectangles to the imported dataset, we duplicate the dataset and fill the rectangles with white color. This creates a pure white image of the same size as the original. We then duplicate the randomly generated small rectangles to generate the mask image. We designate the imported clear frontal photo of a human face as A and the masked image as B. The subsequent phase involves building the model in such a way that A is generated as output to B and subsequently sent to C.

3.4 Occluded Face Repair Model Construction

The occluded face restoration model employs a GAN architecture, with the U-Net network structure serving as the generator. The initial weights and biases of the neural network are set using a normal distribution with a mean of 1 and a standard deviation of 0.02 for both the convolutional layer and the batch normalization layer. The mean of 1 accelerates the activation function's entry into its nonlinear area, but the standard deviation of 0.02 is comparatively minimal, mitigating the issue of gradient vanishing and fostering a more stable training process. By making these selections, the model can more rapidly converge to an optimal state, thus enhancing the quality of generation.

Next, we construct the whole U-Net network architecture in the second phase to be the generator. The generator adopts the U-Net network architecture, which takes advantage of the small size of U-Net and the ability to achieve good training results with very little data, while offsetting the disadvantage of the GAN network that requires large computing resources. The encoder and decoder of this network are constructed independently. The first layer of the encoder is the downsampling layer, which applies DWSC normalisation individually to each channel instead of using the standard convolutional layer. Additionally, the LeakyReLU [26] activation function and dropout layer are incorporated. A sequential model is employed to link all layers together in a sequential manner, enabling forward propagation for input. The upsampling layer is constructed using the same method. Following this, a forward technique is implemented to propagate forward, and connections and character diagrams are piled vertically. In order to achieve this goal, a codec and a skip connection structure are generated successfully. Upon observing the U-Net network structure, it becomes apparent that it is comprised of multiple modules. Therefore, it is necessary to define and utilize multiple modules in a forward serial execution in order to construct a generator for a high-quality U-Net architecture. The process of building the downsampling layer is shown in Algorithm 1, and the same for building the upsampling layer.

Algorithm 1: UNetDown

```
Require: in_size, out_size, normalize, dropout
# Defining Depthwise  Separable Convolution layers
layers ← [
    # Depthwise  Convolution
    nn.Conv2d(in_size, in_size, 4, stride = 2, padding = 1, groups = in_size, bias = False),
    nn.BatchNorm2d(in_size),
    nn.LeakyReLU(0.2, inplace  = True),

    # Pointwise  Convolution
    nn.Conv2d(in_size, out_size, 1, stride = 1, padding = 0, bias = False)
]

if normalize then
    layers.append(nn.InstanceNorm2d(out_size))
end if

# Introduce dropout
if dropout then
    layers.append(nn.Dropout(dropout))
end if

# Sequential threading
model ← nn.Sequential(∗layers)
Ensure: model
```

The third phase involves constructing a distinct element throughout the network. The discriminator's goal is to train the institute to distinguish between the differences in the image generated by the generator and the actual image. Despite the U-Net architecture employed in the generator being effective for lightweight applications, it is unavoidable that the resultant training images may exhibit blurriness or inaccuracies. Consequently, the discriminator is employed to filter out suboptimal training outcomes, thereby enhancing the stability of the model's repair results. It utilizes the discriminator_block function to define the detectors' details and constructs a hierarchical model with four differential blocks. The number of channels gradually increases from 64 to 512. A zero fill layer and a volume layer are applied. The input image img_A and the conditional image img_B are stacked on the channel dimension using the same forward method, resulting in a new input image. This new input image is then passed forward to obtain the final output of the differentiator. The details of this process can be found in Algorithm 2.

Algorithm 2: Discriminator_block

```
Require in_filters, out_filters, normalization
# Depthwise Convolution
layers ← [
    nn.Conv2d(in_filters, in_filters, 4, stride = 2, padding = 1, groups = in_filters, bias = False),
    nn.BatchNorm2d(in_filters),
    nn.LeakyReLU(0.2, inplace = True)
]

# Pointwise Convolution
layers.append(nn.Conv2d(in_filters, out_filters, 1, stride = 1, padding = 0, bias = False))

# Optional normalization
if normalization then
    layers.append(nn.InstanceNorm2d(out_filters))
end if

Ensure: layers
```

Once the generator and discriminator models are constructed, the next step is to address the issue of how to initiate the training process. The primary concern at this stage is to determine the method for loading the data once the experiment is initiated. The getitem function is utilized to retrieve all the anticipated input datasets and iterate through them once. Subsequently, we need to output a subset of these datasets based on their index. The reading process commences from the top left corner of the image, as we aim to horizontally connect the A, B, and C pictures mentioned earlier, starting from the rightmost side. As a result, we measure the width of the picture for each iteration. Finally, we employ the formarray and transform functions to convert the outcome into a PyTorch-compatible image. Convert the output into a tensor format that can be utilized by PyTorch. Develop two distinct data loaders to be utilized for training and validation purposes. It is important to mention that because of the hardware setup and network architecture, ongoing adjustments are required during the learning process. This means that not all the data can be imported simultaneously. Within the training data loader, considering the computers used for training and the overall model performance, we utilize the batch-size parameter to choose a batch size of 64. Additionally, we employ the epoch parameter to determine the total number of training iterations. This allows for flexible modification of the method to accommodate various computer setups and optimize performance in a variety of environments. Within the validation data loader, we establish a batch size of 3. Consequently, during each validation iteration, we evaluate 3 samples. Subsequently, the Adam algorithm is initially employed to establish four optimizers for optimizing the generators and discriminators in the from-A-B and from-A-C processes, respectively.

Following this, the official training of the model commences. During each epoch, the new loop starts by going through the batch of training data. The model inputs are then extracted from the batch and converted into tensors that can be used in the PyTorch environment. The valid and fakeFace adversarial labels are initialized, and the gradient of the face generator optimizer is reset. Upon finishing the aforementioned task, the face generator generates a counterfeit image called fake_B. This image is then inputted into the face discriminator to calculate the anticipated output of the counterfeit image (pred_fakeF). The GAN loss is computed by subtracting the predicted output of the fake image from the valid labels. The pixel-level loss is computed by subtracting the generated fake image from the real image. The total loss of the facial generator is obtained by adding these two losses together, and it is used to update the weights of the generator. Afterwards, the aforementioned process is repeated multiple times, during which the weights of the generator are optimized to minimize the loss. As a result, the generator will produce images that closely resemble the actual data distribution, specifically more realistically restored images of obscured faces. This iterative training process is carried out repeatedly. After performing the update, we evaluate each individual's ultimate error and schedule an assessment of the learning impact for every 20 batches. The detailed procedure is illustrated in Algorithm 3.

Algorithm 3: Training generators

```
#Create a new loop and initialise it
for epoch = opt.epoch to opt.nepochs do
    for i,batch in dataloader do
        real_A ← Variable(batch["A"].type(Tensor))
        real_B ← Variable(batch["B"].type(Tensor))
        real_C ← Variable(batch["C"].type(Tensor))
#Generate samples that can be trained against
valid ← Variable(Tensor(np.ones((real A.size(0),∗patch))),requires grad =
False)
fakeFace ← Variable(Tensor(np.zeros((real A.size(0),∗patch))),requires grad =
False)
fakeMask ← Variable(Tensor(np.zeros((real A.size(0),∗patch))),requires grad =
False)
                                                  ▷ Train Generator for Face
        OPTIMIZER FACE_ G.ZERO_ grad()
        fake_B ← OPTIMIZER FACE(real_A)
        pred_fakeF ← DISCRIMINATIONFACE(fake_B,real_A)
#Calculation of losses
        loss_GAN_F ← criterion_GAN(pred_fakeF,valid)
        loss_pixel ← criterion_pixelwise(fake_B,real_B)
        lossFace_G ← loss_GAN_F + λ_pixel ∗ loss_pixel
        lossFace_G.backward()
        OPTIMIZER FACE_ G.step()
    end for
end for
```

3.5 Loss Function

3.5.1 MSE Loss Function

Mean Squared Error (MSE) is a commonly used loss function to measure the difference between the predicted and true values of a model. In deep learning, we often use MSE as the loss function for regression problems because it is simple and easy to compute, while the penalty for outliers is large, which can effectively reduce the impact of outliers on the model.MSE is calculated as shown in Eq. (1) as follows:

$$MSE = \frac{I}{n} \sum\nolimits_{i=1}^{n} (x_i - y_i) \tag{1}$$

Of these, x represents the original image, y represents a repaired image, i is a pixel index, and n is the number of pixels.

3.5.2 L1 Loss Function

The $L1$ loss function is a commonly used loss function for regression problems, also known as Absolute Error. The $L1$ loss function is calculated by summing and averaging the absolute values of the differences between the predicted and true values of the model, and the $L1$ is calculated as shown in Eq. (2):

$$L1(x, y) = \frac{1}{N_y} \|x - y\|_1 \tag{2}$$

In this case, x represents input, y represents real value, N_y represents the number of elements. Unlike MSE, the $L1$ loss function uses an absolute value rather than a square. This means that the $L1$ loss function is relatively less penalized for big errors, so it is more applicable when dealing with data that contains big error problems. In addition, the $L1$ loss function is less penalised for abnormal values, which reduces the effect of the anomaly on the model.

4 Experimental Results and Analysis

4.1 Experiment Environment

The experimental platform in this paper is an NVDIA GeForce GTX 1070 graphics card with 16 GB of video memory, an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz. The experimental code is implemented in the PyTorch deep learning framework with version 1.10.1, and Python version 3.7.

4.2 Introductions of Datasets

A face image dataset is a collection of numerous face photographs used to train and test algorithms and models pertaining to face identification, face detection, facial expression recognition, and other related tasks. Table 1 provides a summary of the frequently utilized public face picture databases.

Deep learning often requires high-resolution, square picture samples. This not only improves the quality of the generated results but also allows for easier resizing of face images without significant blurring, especially when CPU/GPU computational capacity is limited. This experiment utilizes the CelebA dataset, which is extensively employed in research and experimentation pertaining to face attribute analysis, face recognition, face creation, and other related domains.

Table 1: Commonly used public face image datasets

Datasets	Number of images	Feature
Labeled faces in the wild (LFW)	Over 13,000	Face image diversity is high, but there is some noise and labelling errors.
CelebA	Over 200,000	Higher quality face images, but less variation in face posture and expression.
Face recognition technology evaluation dataset (FRVT)	177,000 (The number of images is constantly updated and expanded)	Covers multiple sub-datasets, allowing for the selection of appropriate datasets for evaluation based on specific scenarios.
MegaFace	Over 1 million	Face images are sourced from the Internet with high diversity and complexity, but with some noise and labelling errors.

4.3 Experimental Details

4.3.1 Experimental Results

After choosing CelebA as the dataset, we randomly picked 3000 photos from it. These images were numbered from 0001 to 3000 and were then separated into three folders based on their numbering sequence: train (0001–1000), test (1001–2000), and val (2001–3000). The val folder represents the validation set. Load the constructed model and import the training dataset, the model automatically loads each face image and then converts it into a tensor, firstly, random black rectangle masking is performed on each face image according to the operations of face image preprocessing, and a three-image splice of A, B, and C presented from right to left is generated. As shown in Fig. 3, there are the mask image, original image, and masked image, respectively, which are output to the established folder.

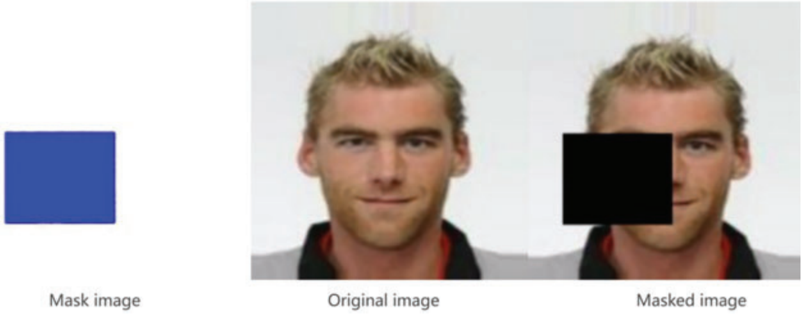

Figure 3: Face image preprocessing results

The mask image in the figure is juxtaposed with the rectangular occluder referenced in the third section of the methods. The masked image is produced by applying the rectangular occluder to the original image during the experiment. This is the initial phase of the experiment, the picture preparation procedure, shown with an image for the reader’s comprehension.

Next the generator automatically predicts the unmasked face, and after many confrontations with the discriminator, 3 photos are output as a batch for the final training results. The initial restoration results are shown in Fig. 4, and the restoration results after several training sessions are shown in

Fig. 5, where each line represents the occluded image, the restored image, the masked image and the real image, respectively.

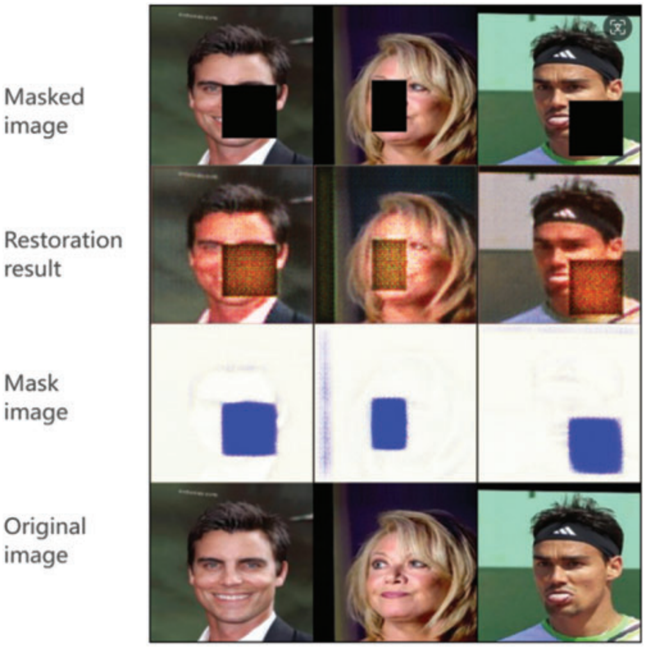

Figure 4: A clear picture of the initial restoration results

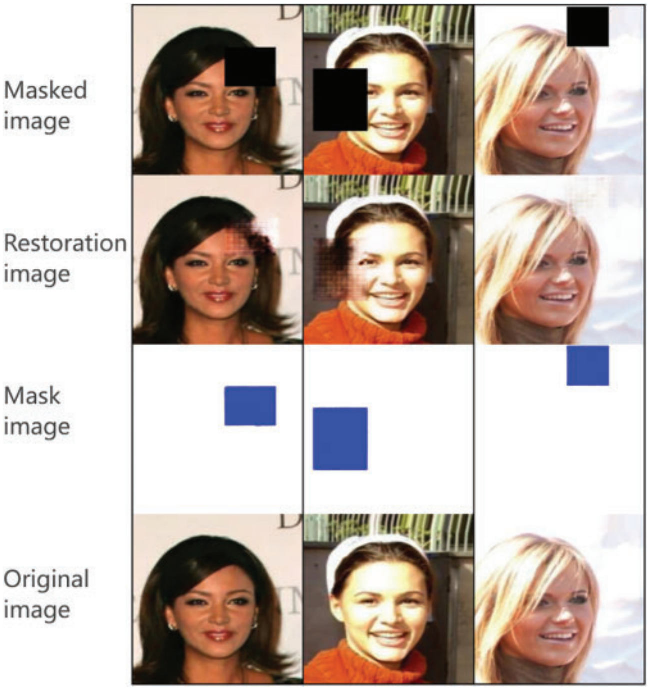

Figure 5: A clear image of the final restoration results

The illustration presents three groups of clear facial photographs: from top to bottom, the masked images, the restoration results, the mask images, and their original forms. As these represent the preliminary training outcomes, it is evident that the image restoration quality is subpar, and they are presented here for comparison with results following extensive training.

The particular structure is identical to Fig. 4. This image represents the outcome of the final model restoration following numerous training sessions, provided here for comparison with the original restoration result, demonstrating the model's superiority.

4.3.2 Comparative Analyses

The studies utilized rectangular masks of varying sizes to imitate occlusion caused by glasses, hats, masks, etc., in order to obtain a clear image of the face. Fig. 6 illustrates the image of the same face shot with varied proportions of rectangular masks. Fig. 6a depicts the original face picture, Fig. 6b shows the mask image with an occlusion ratio of 10%–15%, Fig. 6c displays the face image with the same occlusion ratio, and Fig. 6d exhibits the restored face image after occlusion. Similarly, Fig. 6e,f,g represents the corresponding images with masking ratios in the range of 20%–30%.

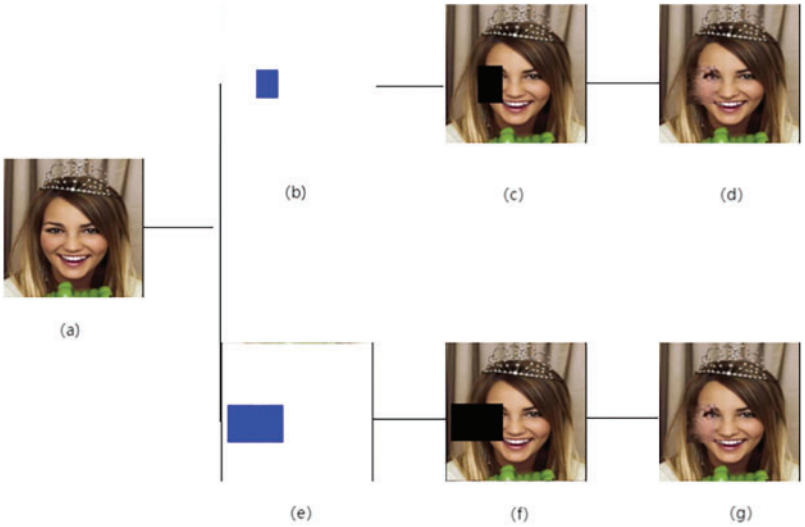

Figure 6: Restoration effect of the same photo with different occlusion ratios

The masked photos with occlusion ratios of 10%–15% and 20%–30%, together with their restoration effects, are presented, demonstrating that regardless of the occlusion ratio for clear facial images in real-world scenarios, the model provided in this research exhibits effective restoration capabilities.

Directly perceiving the disparity in restoration effect is challenging for the human eye. Therefore, the peak signal-to-noise ratio (PSNR) is employed as an evaluation metric in this paper to compare the performance of the occluded face restoration algorithm. The algorithm is tested on the same face image with varying occlusion ratios. The PSNR is a significant measure used to quantify picture quality. It is derived from the concept of MSE, which calculates the squared difference between pixel values of two images at corresponding positions. PSNR represents the average MSE of the two images. Therefore, the higher the value of MSE, the lower the similarity between the two images. For a given original image I of size $m \times n$ and a noisy image K to which noise has been added, the MSE is defined as shown in Eq. (3):

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{3}$$

The double summation equation of MSE utilized here is differentiated from the single summation equation in Eq. (1). When associated with the PSNR of the measured picture samples, the data to be processed transforms into multi-dimensional data, and the double summation method effectively captures the errors of each sample across several feature dimensions, thus facilitating a more thorough error assessment. Then PSNR is defined as shown in Eq. (4):

$$PSNR = 10 \cdot log_{10}\left(\frac{MAX1^2}{MSE}\right) = 20 \cdot log_{10}\left(\frac{MAX_1}{\sqrt{MSE}}\right) \tag{4}$$

Here, $MAX1$ is the maximum possible pixel value of the image. From the equation, it can be seen that the value of $MAX1$ is proportional to the $PSNR$ value and inversely proportional to the MSE. So the higher the $PSNR$ value, the higher the similarity between the two images and the smaller the difference value. The $PSNR$ values are compared for different masking ratios respectively and the results are shown in Table 2.

Table 2: PSNR values for different masking ratios

Different masking ratios	PSNR
10%–15%	35.51
20%–30%	30.33

After that, the occluded face repair algorithm designed in this paper is compared with PM [27], ShiftNet [28] and PICNet [29] repair algorithms. Firstly, it can be intuitively seen with the naked eye in Figs. 7 and 8 that the algorithm in this paper has made significant progress in terms of repair accuracy, colour coordination and contour clarity.

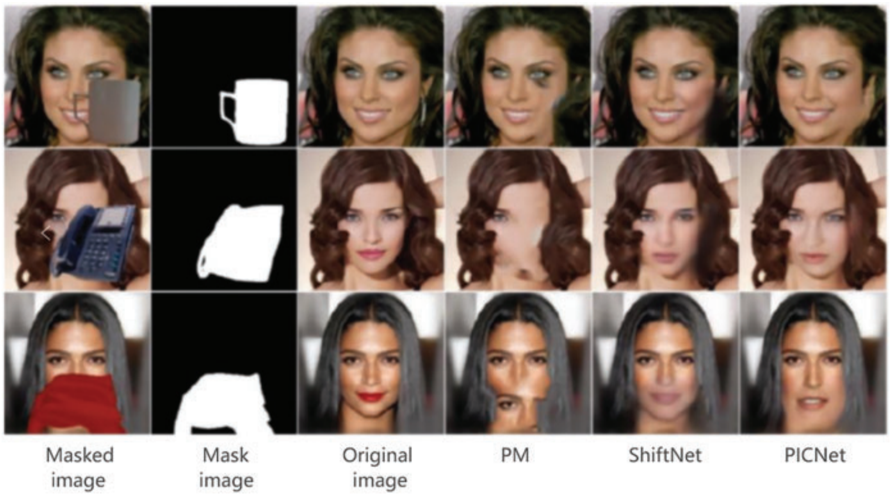

Figure 7: Effects of other algorithmic fixes

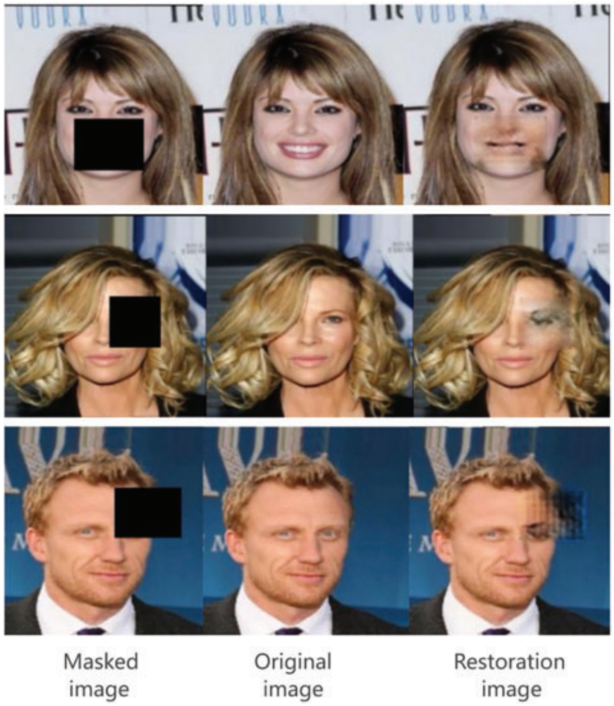

Figure 8: Repair effect of the algorithm in this paper

The picture enumerates the various restoration impacts of PM, ShiftNet, and PICNet under identical occlusion and occlusion position conditions. The face features are evidently disordered, and the boundaries of the restoration appear indistinct. The comparison with Fig. 8 below demonstrates the algorithm's superiority presented in this paper.

The restoration outcomes for the mouth, eyes, and forehead are chosen for longitudinal comparison, demonstrating that the proposed method effectively repairs facial pictures with varying occlusion positions. The images depicting the repair outcomes are juxtaposed horizontally with the repair results of alternative algorithms in Fig. 7, demonstrating the superiority of the novel algorithm suggested in this paper.

Subsequently, by contrasting the introduced PSNR indicators, due to the principle of reducing the error, selecting the data of 20%–30% of the lower blocking ratio of PSNR, the contrast results as shown in Table 3 can be seen that the PSNR value of the blocking face repair algorithm proposed in this paper is higher than the other three algorithms, proving its superiority.

Table 3: Comparison of PSNR values of different algorithms

Different algorithms	PSNR
The algorithms in this paper	30.33
PM	28.98
ShiftNet	29.36
PICNet	28.93

5 Conclusions

In this paper, in the beginning of the design network first firmly the principles of robustness, efficiency, scalability principles, after introducing the current mainstream image repair algorithm, discovered its limitations, successfully introduced a new form of deep learning-based disguised face repair network. We developed a hybrid network combining GAN and U-Net architectures. The U-Net architecture serves as the generator in the GAN framework and is regulated by the discriminator. The training begins with an obstructed facial image featuring a black rectangle that simulates the occluder. Following extensive training, a precise and accurate reconstructed image is produced, while maintaining a lightweight overall model. We selected samples with varying occlusion levels to evaluate the PSNR value. This result indicates that the algorithm presented in this research surpasses the three predominant algorithms in image restoration: PM, ShiftNet, and PICNet, hence demonstrating the model's supremacy.

However, the network still exhibits some deficiencies. During the experiment, I observed a significant disparity in the workload between the two tasks: repairing the covered face image and extracting the mask image. The task of creating masks required little information and could be accomplished with minimal training. However, repairing the image was time-consuming and required substantial effort. Additionally, the overall efficiency of connecting the two networks was not scientifically sound. In future research, this issue could be addressed by separating the networks in a more logical manner.

At the same time, due to time and resource constraints, the network was only able to achieve a basic level of repair for occluded faces. Although the model was trained, the system implementation was not fully completed. In the future, I plan to continue refining the network by optimizing its parameters to improve the repairs accuracy. Additionally, I aim to replace the current occluder, which is a black rectangle, with more realistic occluders such as sunglasses and masks. This will enhance the model's ability to repair occluded faces and make it more robust. Ultimately, I hope to extend the application of occluded face repair to videos.

Acknowledgement: All the face images used in this paper are from publicly available datasets CelebA.

Funding Statement: This work was supported by the Double First-Class Innovation Research Project for People's Public Security University of China (No. 2023SYL07).

Author Contributions: Study conception and design: Yitong Zhou, Tianliang Lu; data collection: Yitong Zhou; analysis and interpretation of results: Yitong Zhou; draft manuscript preparation: Yitong Zhou; supervision: Tianliang Lu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this paper can be requested from the corresponding author upon request.

Ethics Approval: The study involves human subjects, but uses occluded face images from the open source dataset CelebA, which complies with ethical requirements. The committee which approved the study is People's Public Security University of China.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] N. Benamara, E. Zigh, T. Stambouli, and M. Keche, “Towards a robust thermal-visible heterogeneous face recognition approach based on a cycle generative adversarial network,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 7, no. 4, pp. 132–145, 2022. doi: 10.9781/ijimai.2021.12.003.

[2] A. Alcaide, M. Patricio, A. Berlanga, A. Arroyo, and J. Gallego, “LIPSNN: A light intrusion-proving siamese neural network model for facial verification,” *Int. J. Interact. Multimed. Arti. Intell.*, vol. 7, no. 4, pp. 121–131, 2022. doi: 10.9781/ijimai.2021.11.003.

[3] X. Yang, W. Wang, Z. Wang, J. Wu, and W. Li, “Attention-based generative adversarial network for face inpainting,” *IEEE Transact. Image Process.*, vol. 29, pp. 5435–5448, 2020.

[4] X. Zhao, Z. Zhao, and C. Yang, “Lightweight face image restoration algorithm based on multi-scale feature fusion,” (in Chinese), *Telecommun. Sci.*, vol. 40, no. 8, pp. 42–51. doi: 10.11959/j.issn.1000-0801.2024183.

[5] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “Filling-in by joint interpolation of vector fields and gray levels,” *IEEE Transact. Image Process.*, vol. 10, no. 8, pp. 1200–1211, 2001. doi: 10.1109/83.935036.

[6] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004. doi: 10.1109/TIP.2004.833105.

[7] Y. Cao, W. Jin, and R. Fu, “An image inpainting method based on weighted priority and classification matching,” *Telecommun. Sci.*, vol. 33, no. 4, pp. 94–100, 2017.

[8] X. Wang, Y. Li, H. Zhang, and Y. Shan, “Towards real-world blind face restoration with generative facial,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 9168–9178.

[9] X. Pan, X. Zhan, B. Dai, D. Lin, C. Chen and P. Luo, “Exploiting deep generative prior for versatile image restoration and manipulation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7474–7489, 2022 Nov. doi: 10.1109/TPAMI.2021.3115428.

[10] W. Syed, A. Aditya, K. Salman, H. Munawar, S. Fahad and Y. Ming-Hsuan, “Restormer: Efficient transformer for high-resolution image restoration,” 2022, *arXiv:2111.09881v2*.

[11] J. Liang, J. Cao, G. Sun, K. Zhang, V. Luc and T. Radu, “SwinIR: Image restoration using swin transformer,” 2021, *arXiv:2108.10257v1*.

[12] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, Boston, MA, USA, 2015, pp. 3431–3440.

[13] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *LNCS 9351: Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interven.*, Munich, Germany, Cham, Springer, 2015, pp. 234–241.

[14] H. Liu, B. Jiang, and C. Yang, “Coherent semantic attention for image inpainting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, Long Beach, CA, USA, 2019, pp. 4170–4179.

[15] Y. Zeng, J. Fu, H. Chao, and B. Guo, “Learning pyramid context encoder network for high-quality image inpainting,” in *Proc. 2019 IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Piscataway, IEEE, 2019, pp. 1486–1494.

[16] S. Luo, M. Chen, and L. Chen, “Face image restoration network based on facial feature point,” (in Chinese), *Chin. Sci. Tech. Paper*, vol. 16, no. 7, pp. 729–734+742, 2021.

[17] J. Qin, H. Bai, and Y. Zhao, “Multi-scale attention network for image inpainting,” *Comput. Vis. Image Underst.*, vol. 204, 2021, Art. no. 103155. doi: 10.1016/j.cviu.2020.103155.

[18] N. Wang, Y. Zhang, and L. Zhang, “Dynamic selection network for image inpainting,” *IEEE Trans. Image Process.*, vol. 30, pp. 1784–1798, 2021. doi: 10.1109/TIP.2020.3048629.

[19] L. Liao, J. Xiao, Z. Wang, L. Chia-Wen, and S. Shinichi, “Image inpainting guided by coherence priors of semantics and textures,” in *Proc. 2021 IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, USA, IEEE Computer Society, 2021, pp. 6539–6548.

[20] Y. Cui, W. Ren, and A. Knoll, “Omni-kernel network for image restoration,” *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 2, pp. 1426–1434, 2024. doi: 10.1609/aaai.v38i2.27907.

[21] X. Lin *et al.*, “DiffBIR: Towards blind image restoration with generative diffusion prior,” 2023, *arXiv:2308.15070*. doi: 10.48550/arXiv.2308.15070.

[22] Y. Bai, C. Wang, S. Xie, C. Dong, C. Yuan and Z. Wang, "TextIR: A simple framework for text-based editable image restoration," 2023, *arXiv:2302.14736*. doi: 10.48550/arXiv.2302.14736.

[23] T. Yang, P. Ren, and L. Zhang, "Synthesizing realistic image restoration training pairs: A diffusion approach," 2023, *arXiv:2303.06994*. doi: 10.48550/arXiv.2303.06994.

[24] A. Jaiswal, X. Zhang, S. Chan, and Z. Wang, "Physics-driven turbulence image restoration with stochastic refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.(CVPR)*, Vancouver, BC, Canada, 2023, pp. 12170–12181.

[25] Y. Guo, Y. Li, L. Wang, R. Feris, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 8368–8375. doi: 10.1609/aaai.v33i01.33018368.

[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, USA, 2016.

[27] Y. Zhang and J. Wu, "PM: A patch matching algorithm for image inpainting," *IEEE Transact. Image Process.*, vol. 28, no. 9, pp. 4572–4584, 2019.

[28] J. Liu, Y. Zhang, and S. Wang, "ShiftNet: Image inpainting via deep shift convolutional networks," *IEEE Transact. Image Process.*, vol. 27, no. 6, pp. 2923–2934, 2018.

[29] Y. Liu and Y. Zhang, "PICNet: A novel image inpainting network based on patchwise interaction and contextual information," *IEEE Transact. Image Process.*, vol. 29, pp. 3718–3731, 2020.