International Journal of Innovative Computing, Information and Control
Volume **21**, Number **6**, December **2025**

 ISSN 1349-4198
pp. **1691–1704**

IMPROVING THE U-NET ARCHITECTURE BY INCORPORATING A PRE-TRAINED VGG-16 MODEL FOR A FACE IMAGE RECONSTRUCTION

Indah Agustien Siradjuddin, Cucun Very Angkoso
and Kurniawan Eka Permana

Informatics Department
Faculty of Engineering
Universitas Trunojoyo Madura
Jl. Raya Telang, PO BOX 02 Kec. Kamal, Bangkalan, Madura 69162, Indonesia
{ indah.siradjuddin; cucunvery; kurniawan }@trunojoyo.ac.id

Received December 2024; revised May 2025

Abstract. *The paper presents a model for reconstructing face images by removing the appearance of a medical mask from a face image, effectively revealing the underlying facial features. The model is based on the U-Net architecture, which consists of four components. First is encoder blocks for feature extraction. Second is the decoder blocks to reconstruct the input image from the extracted features. Third is skip connection to retain the encoder's important features for the decoders. Finally, the bridge connects the last encoder block and the first decoder block. Instead of the original encoder blocks, this paper incorporates the convolutional blocks from VGG-16 to improve the model's performance. VGG-16 is used due to the similarity layers of its convolutional blocks with the encoder blocks of the original U-Net and the available pre-trained weights on large-scale datasets, making it a strong weight initialization. We trained the model with pairs of images, one is a face with a mask and the other is a face without a mask image. We evaluated the model's performance using the Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). The lowest MSE achieved was 0.0054, and the highest PSNR was 27.7625 db. The incorporating of pre-trained VGG-16 increases the PSNR up to 25.97%.*

Keywords: U-Net, Encoder blocks, Decoder blocks, Skip connection, VGG-16, Face image reconstruction

1. **Introduction.** Five years after the COVID-19 pandemic, we still adopted some beneficial new habits that have become part of our daily lives. This study will focus specifically on the habit of wearing masks in public spaces. Many of us continue to wear masks to protect ourselves from spreading any virus. However, the presence of masks on people's faces can make them less recognizable since facial parts below the nose are hidden. This creates a new issue; several people wear masks to hide their faces, making them unrecognizable if they are captured by security cameras while engaging in criminal activities. Therefore, to identify the faces captured by security cameras, it is essential first to eliminate the mask's appearance on the face. This research developed a model to reconstruct the face image by removing the mask on the face image, utilizing a deep learning approach.

Recent studies have explored deep learning approaches for a wide range of domains. In biomedical studies, deep learning is developed to identify and classify diseases, to predict the treatment, or others [1, 2, 3]. For the agriculture field, the deep learning approach can also be used for various tasks, such as crop disease detection [4, 5] and monitoring

DOI: 10.24507/ijicic.21.06.1691

crop health [6, 7]. In software engineering, deep learning is utilized to automate the selection of the software development life cycle (SDLC) [8], and many other research from other domains. These studies show the advantage of deep learning in addressing problems across various domains. One of the essential advantages of the deep learning approach is the ability to learn features; hence, the handmade features are not required.

Therefore, we developed a face image reconstruction based on the deep learning approach in the proposed study. However, we are focusing on approaches for pixel-wise transformation, receiving an image as input and resulting in an image as output. Generative Adversarial Networks (GANs) is a well-known approach to generating images [9]; however, the network requires large datasets for training. SAM C-GAN for removing a face mask from a face image required more than ten thousand images as a dataset [10], and large datasets with more than ten thousand images and seven thousand images are also required in GAN-based face reconstruction [11, 12]. Another approach for image inpainting is deep image prior based on ConvNets, which works well in filling in the missing part of the image [13]. However, the approach requires several minutes to compute per image using GPU.

U-Net is also a deep learning approach for pixel-wise transformation. It can capture the global context of the image and reconstruct the detailed context through the encoder and decoder components of U-Net. Further, U-Net can be developed and improved for various research domains. As a result, our work-study is based on this architecture.

To carry out the works in [10, 11, 12, 14], synthesized datasets are used to build the reconstruction model. A face mask is artificially added to the face image to create a pair of mask-face images and a face image without a mask. In our work, each subject was recorded both with and without a face mask in almost consecutive time to build the dataset. The following are the contributions of our work:

- Improving the original U-Net architecture with a pre-trained VGG-16 model for our face image reconstruction model;
- Creating a real dataset (not synthetic) consisting of a pair of face images and face images with a medical mask on them.

2. **Related Works.** U-Net architecture was first introduced in 2015 by Ronneberger et al. [15] for biomedical image segmentation. For this purpose, the proposed architecture surpassed other segmentation models, i.e., the network achieved a 92% average of IOU. The architecture's shape is like the shape "U", which is why it is called U-Net. This network has two main components: an encoder as the feature extractor with downsampling and a decoder component as the reconstructor with upsampling. One of the benefits of this network is that it has a skip connection, which links the encoder and decoder blocks of the U-Net architecture to retain important information by combining low-level and high-level features from both the encoder and decoder.

Studies have been conducted on biomedical image segmentation by modifying the U-Net architecture to enhance the accuracy of the segmentation. Oktay et al. [16] enhanced the U-Net architecture with attention mechanisms to improve the segmentation task, especially to detect small organs like the pancreas; the result outperformed the original U-Net and other segmentation algorithms. This attention mechanism allows the proposed model to focus on important areas of the image while still maintaining the global structure. Another study [17] by Yan et al. enhanced the U-Net by attention and hybrid dilated convolution to segment breast tumors in ultrasound images. This proposed method has significantly improved, with an accuracy of up to 95.81%. Attention mechanisms also can be found in [18]. Sulaiman et al. [18] incorporated four gates of attention in U-Net to localize the features for breast cancer segmentation, which achieved up to 0.98 accuracy.

Enhancing U-Net architecture with inception studies can be found in [19, 20]. The inception module captures local and global features by convoluting the input with different kernel sizes and concatenating the result for the subsequent layers. It also employs the 1×1 convolution to reduce the depth of the features without eliminating the model's ability to capture important information. In [19], the architecture was proposed for semantic segmentation in Microscopy cell images. The result showed that inception has a better IOU than the original U-Net. In [20], the U-Net with inception was proposed for another research domain, i.e., building detection. The proposed architecture achieved higher accuracy in the study than the original U-Net.

Besides adding attention or an inception module to the original U-Net architecture, improving U-Net by changing the encoder blocks with other architectures, such as VGG-16, is also studied in [21, 22]. The experiments that were conducted resulted in the VGG-16 increasing the accuracy of the original U-Net. Encoder blocks play a feature extractor role in the U-Net architecture. Meanwhile, VGG-16 is known for its feature learning layers and has been well-tested to classify images in many research studies. Thus, using VGG-16 as the encoder in U-Net will result in better feature extraction.

In this study, we developed a face image reconstruction based on the U-Net architecture while using VGG-16 as the encoder block for the model. However, we also utilize the pre-trained weights of the feature learning layers, which have been trained on over one million images, to classify with 1000 categories.

3. Mask Face Reconstruction.

We present a mask-face image reconstruction that removes the mask's appearance from the face image. Therefore, the reconstruction model reveals the facial features hidden under the mask. We developed the reconstruction model based on the U-Net architecture and replaced certain components of the architecture with VGG-16 convolutional blocks. We trained the model with a pair of face images, one with a mask and one without a mask.

3.1. U-Net architecture.

The mask face reconstruction in this research is developed based on the main architecture, U-Net. The U-Net comprised of encoder and decoder components, incorporating a skip connection from encoder to decoder. This architecture is well-suited to our research since we built a model for pixel-wise transformation tasks. The U-Net will learn to map each pixel in the input image to the corresponding output image to reveal the facial features inside the mask from the input image.

The encoder (contracting path) component is the feature extractor in the U-Net architecture. It learns the best representation by increasing the depth of the feature maps and reducing the input's spatial dimension. Therefore, convolution and pooling layers are involved in this component. The model has four encoder blocks in this encoder component. Each block consists of two convolution layers with 3×3 kernels to extract the features, followed by max-pooling with 2×2 stride to reduce the spatial dimension. The feature maps from the convolution layers are obtained using Equation (1).

$$f_l(x, y) = k * f_{l-1}(x, y) = \sum_{i=-m}^{m} \sum_{j=-m}^{m} k(i, j) f_{l-1}(x + i, y + j) \tag{1}$$

Here, m is the size of the kernel matrix, k, divided by two, such that the index of the center position of a kernel matrix is zero. If the kernel size is 3×3, then i and j will start from -1 up to 1. The f_l is the current feature map, and f_{l-1} is the previous one.

At two consecutive encoder blocks from the contracting path, the depth of the feature maps is doubled. Figure 1 shows the four encoder blocks along with size of the feature maps in each layer. The depth of feature map in the first encoder block is 64, and the

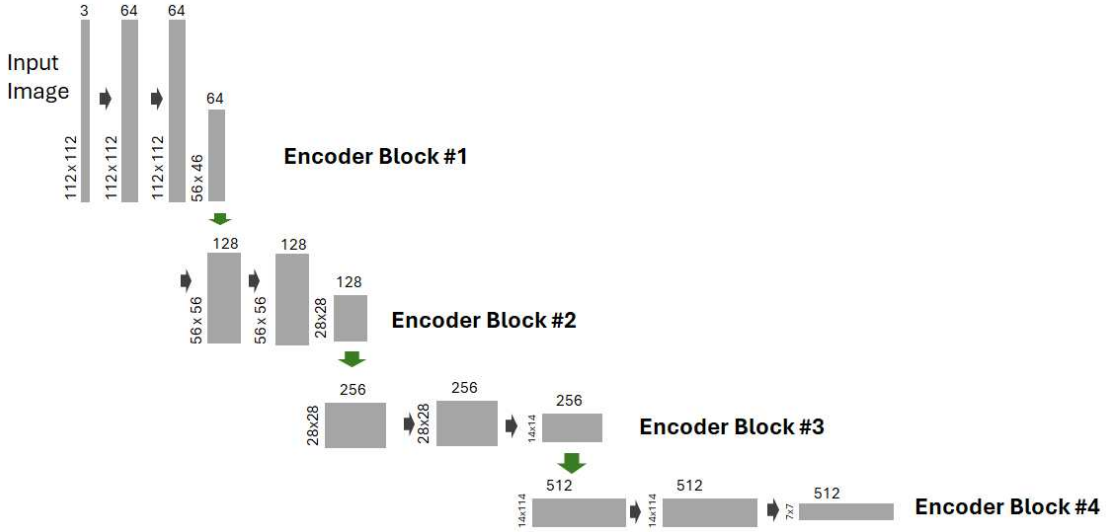


FIGURE 1. Four encoder blocks in encoder (contracting) component

depth in the second encoder block is 128. The depth of the feature map is increasing up to 512 in the output of the last encoder block. Therefore, architecture learns the best features from convolution and pooling processes to represent the input images in this stage.

The decoder (expanding path) component reconstructs the input images by preserving the extracted features from the encoder component; therefore, in this component, there are three parts involved. First, it includes the transposed convolutional layers for upsampling the feature map; hence, the output size is equal to the input size at the last layer of the U-Net. Second, the concatenation of skip connection from the encoder helps retain the encoder's important contextual information by combining low-level spatial features with high-level semantic features. The last part is the two convolutions by 3×3 kernels to refine the reconstruction process.

The decoder component consists of four decoder blocks, with the depth of feature maps halved in each successive block. Figure 2 shows the four decoder blocks with their feature map size. There are two colors in specific rectangles in the figure. It represents

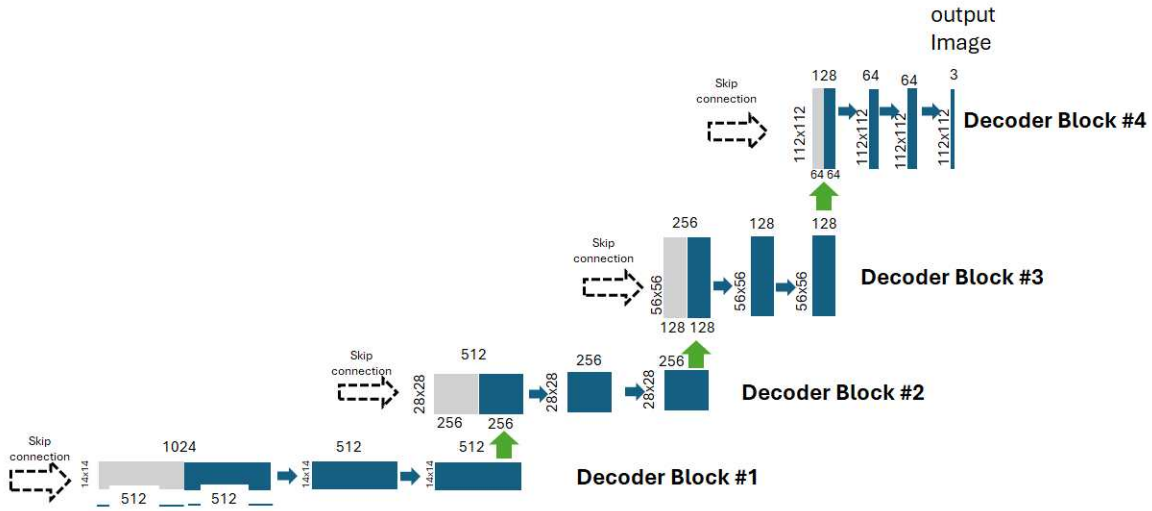


FIGURE 2. Four decoder blocks in decoder (expanding) component

the concatenation process. The grey is the feature map from the encoder blocks, while the blue rectangle is the upsampled feature map from the previous layer.

Figure 3 illustrates the upsampling using the transpose convolution process in the decoder component. The figure shows feature map transformation in the upsampling process. Initially, the input feature maps are modified based on the stride value, and then, to ensure the desired spatial resolution, the feature map is padded before the convolution process.

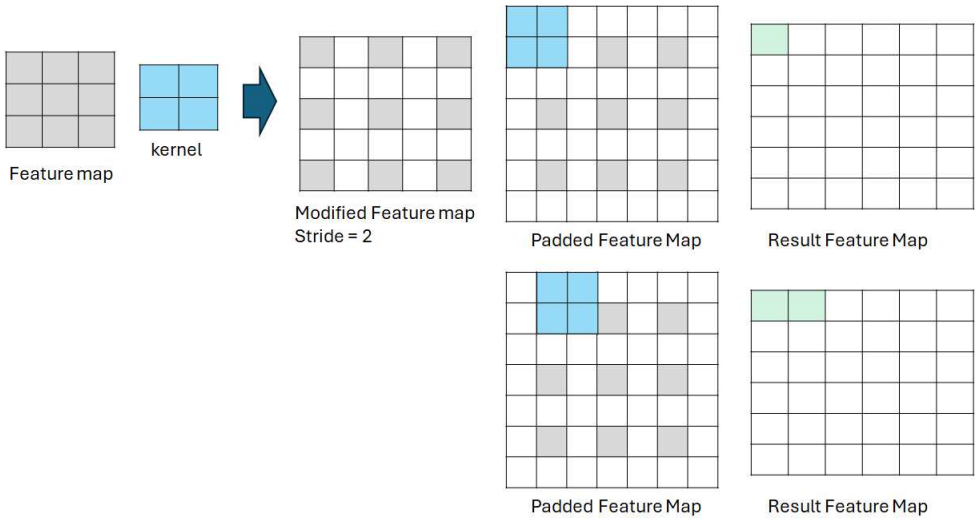


FIGURE 3. Transpose convolution for upsampling in decoder blocks

The last component in the U-Net architecture is the bridge component. This component connects the encoder and the decoder component. There are two convolution layers in this component as seen in Figure 4.

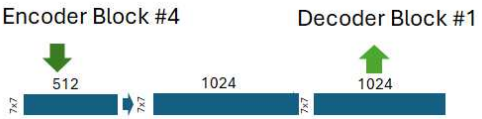


FIGURE 4. Bridge component in U-Net architecture

3.2. **VGG-16 architecture.** The VGG-16 architecture was developed by the Visual Geometry Group (VGG) and became a milestone in deep learning, especially in the domain of classification and feature extraction. The VGG-16 was first introduced in the proceeding article [23]. The architecture consists of 16 layers of trainable weights, as in Figure 5. The figure shows there are five convolutional blocks in the feature learning layers; each block

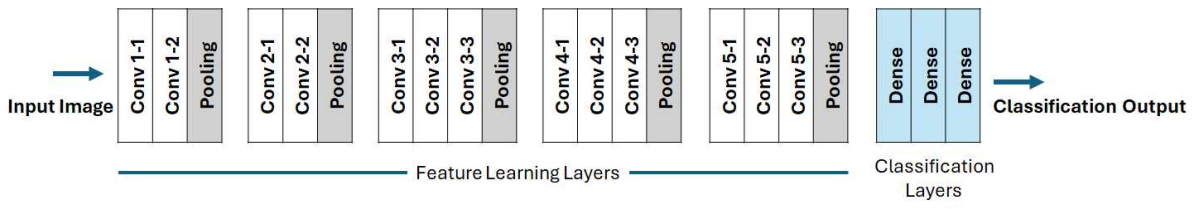


FIGURE 5. VGG-16 architecture

comprises convolution and pooling layers with different numbers of convolution layers and a number of kernels in each block.

The advantage of the VGG-16 architecture lies in its simplicity and depth. Therefore, it is well-suited to a wide range of applications, such as in the medical image classification [24], agriculture [25], and autonomous car [26].

3.3. **U-Net and VGG-16 for face image reconstruction.** We developed our face image reconstruction model by combining the U-Net and VGG-16 architectures (V-Net). The adoption of VGG-16 into the U-Net is due to two main advantages. First, layers of the encoder blocks in the U-Net architecture (Figure 1) and the convolutional blocks in VGG-16 (feature learning layers) (Figure 5) are quite similar, so changing the original encoder blocks with convolution blocks of VGG-16 will not affect the whole U-Net architecture. Second, VGG-16 provides ImageNet pre-trained weights trained on the ImageNet visual database, i.e., a large-scale dataset of more than one million images and 1000 categories. Thus, these pre-trained weights will be a strong initialization in the U-Net architecture.

Therefore, in this study, we replaced only the original encoder blocks with the convolutional blocks from VGG-16, making four modifications. First, two blocks from the convolutional blocks in VGG-16 replace U-Net's first two encoder blocks. Second, the third convolutional block from VGG-16 is replacing the third encoder block. Third, the fourth convolutional block from VGG-16 replaces the fourth encoder block. Fourth, the last layer (pooling layer) of the fourth convolutional block from VGG-16 is removed to meet the output size requirement from the encoder component. The bridge and the decoder component of U-Net architecture are still maintained.

The detailed layers of the encoder component from the original U-Net and encoder component from a combination of U-Net and VGG-16 are summarized in Table 1, and all the layers from both models are visualized in Figure 6. The figure shows that the encoder component of V-Net has deeper layers than in the original U-Net. The bridge and the decoder component of our proposed model are the same as the original U-Net. The skip connection in V-Net links the result of the encoder block with the decoder block, and then both feature maps are concatenated for further convolution process.

In the combination of VGG-16 and U-Net (V-Net), we utilize the available pre-trained weights that were trained on over one million images from the ImageNet Visual database. These weights are trained to classify images into 1000 categories. We built the mask face reconstruction model from these pre-trained weights by setting up the pre-trained weights whether to freeze them or keep them not frozen during the training process. Frozen pre-trained weights remain unchanged throughout the training of the face reconstruction model; on the contrary, the unfrozen pre-trained weights are updated during the training process.

4. **Main Results.**

4.1. **Dataset.** We built a dataset for the experiments with the help of 152 volunteers. Images were taken from three distinct points of view, and captured when the volunteers were wearing a mask and without one within a short time interval. Therefore, the process assumed that facial expressions in both conditions are almost equivalent. Since there are 152 volunteers, the dataset should consist of 456 images of a face without a mask and 456 images of a face with a mask. However, problems capturing the images resulted in only 330 pairs of images, a face without a mask, and a face with a mask. The dataset is available in [27].

We preprocessed the images before using them as a dataset for our analysis, ensuring they met the required format in subsequent tasks. The preprocessing involved face

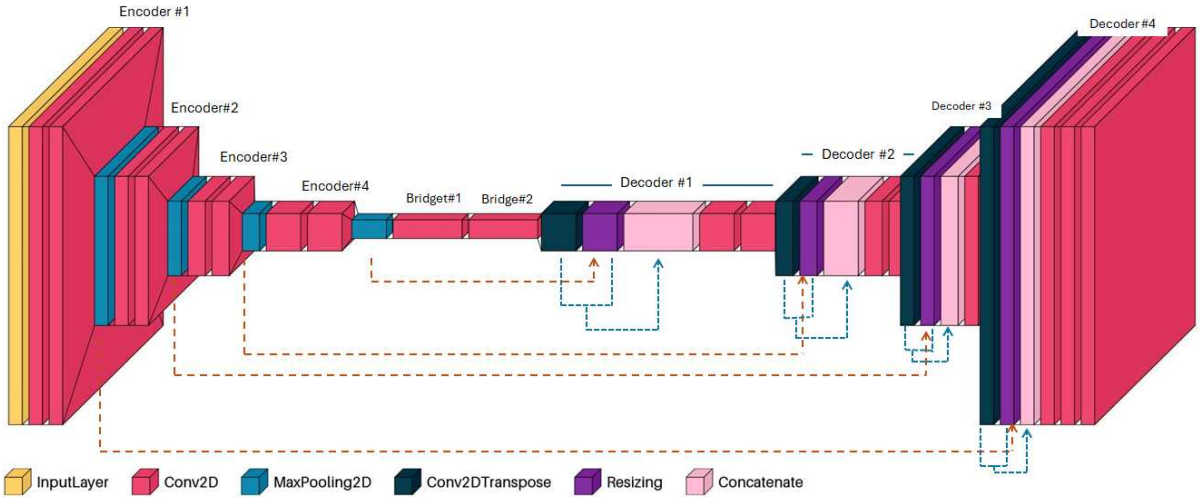
TABLE 1. Encoder component from original U-Net and combination of U-Net and VGG-16 (V-Net)

No	Layers		Hyperparameter				Output	Size
	U-Net	V-Net	K	F	S	P	U-Net	V-Net
1	Input	Input					$112 \times 112 \times 3$	$112 \times 112 \times 3$
Encoder Block #1								
2	Conv2D	Conv2D	64	3	1	1	$112 \times 112 \times 64$	$112 \times 112 \times 64$
3	Conv2D	Conv2D	64	3	1	1	$112 \times 112 \times 64$	$112 \times 112 \times 64$
4	MaxPool	MaxPool	–	2	2	–	$56 \times 56 \times 64$	$56 \times 56 \times 64$
Encoder Block #2								
5	Conv2D	Conv2D	128	3	1	1	$56 \times 56 \times 128$	$56 \times 56 \times 128$
6	Conv2D	Conv2D	128	3	1	1	$56 \times 56 \times 128$	$56 \times 56 \times 128$
7	MaxPool	MaxPool	–	2	2	–	$28 \times 28 \times 128$	$28 \times 28 \times 128$
Encoder Block #3								
8	Conv2D	Conv2D	256	3	1	1	$28 \times 28 \times 256$	$28 \times 28 \times 256$
9	Conv2D	Conv2D	256	3	1	1	$28 \times 28 \times 256$	$28 \times 28 \times 256$
10		Conv2D	256	3	1	1	$28 \times 28 \times 256$	$28 \times 28 \times 256$
11	MaxPool	MaxPool	–	2	2	–	$14 \times 14 \times 256$	$14 \times 14 \times 256$
Encoder Block #4								
12	Conv2D	Conv2D	512	3	1	1	$14 \times 14 \times 512$	$14 \times 14 \times 512$
13	Conv2D	Conv2D	512	3	1	1	$14 \times 14 \times 512$	$14 \times 14 \times 512$
14		Conv2D	512	3	1	1	$14 \times 14 \times 512$	$14 \times 14 \times 512$
15	MaxPool	MaxPool	–	2	2	–	$7 \times 7 \times 512$	$7 \times 7 \times 512$
16		Conv2D	512	3	1	1	$7 \times 7 \times 512$	$7 \times 7 \times 512$
17		Conv2D	512	3	1	1	$7 \times 7 \times 512$	$7 \times 7 \times 512$
18		Conv2D	512	3	1	1	$7 \times 7 \times 512$	$7 \times 7 \times 512$

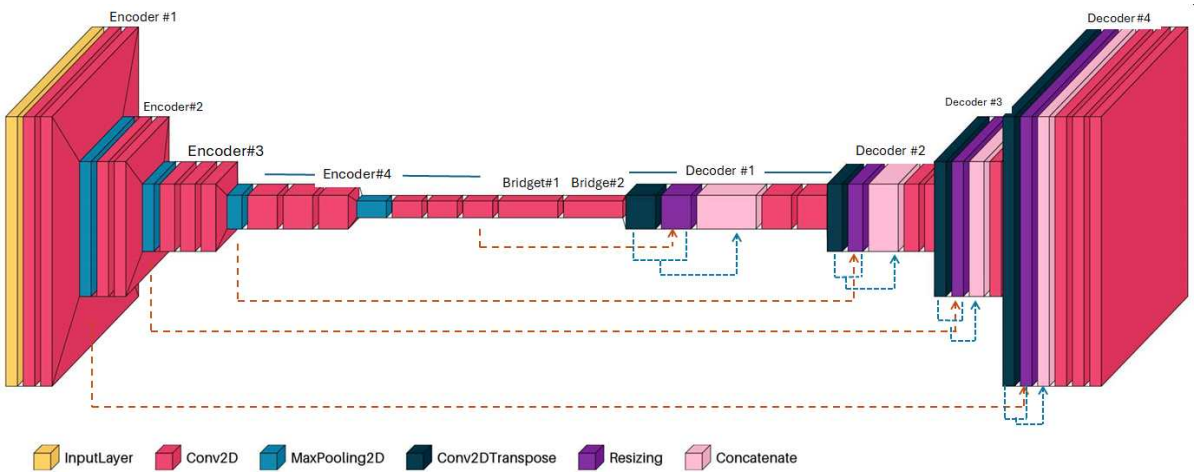
detection and image resizing. Since the captured images included the face down to the shoulders and visible background, a face detection preprocess was required to focus on the face area. We cropped the images based on the detected face location. We use a face recognition library taken from pypi.org to retrieve the area of the face. The last preprocessing is image resizing. The cropped images are then resized into 112×112.

We also involve the augmentation process in the dataset to increase the variety and number of images. Therefore, the ability of the reconstruction model is enhanced. We employed three kinds of augmentation: the first is darkening the original image, and the second is brightening the original image, where both of these processes use exposure adjustment. The last augmentation is flipping the original image. The examples of the original and the augmented images are shown in Figure 7. Because of the augmentation process, the number of images in the dataset is increased, i.e., 1320 images of a face with a mask, and its pairs are 1320 images of a face without a mask. The original and augmented datasets are then split into 80% training data and 20% testing data. The splitting was chosen randomly.

4.2. **Evaluations.** We measure the performance of our reconstruction face image using two evaluation metrics, i.e., MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio). MSE calculates the average difference between the reconstructed and the original image, as written in Equation (2). The lower value of MSE represents that the reconstructed image is almost similar the the original one. Additionally, the PSNR calculates



(a) U-Net



(b) V-Net on encoder blocks

FIGURE 6. (color online) A masked face reconstruction

FIGURE 7. The original image (left) and augmented images (three images on the right)

the quality of the reconstructed image relative to the original image as written in Equation (3). The higher PSNR value indicates that the result of the reconstruction image has a good image quality.

$$MSE = \frac{1}{mn}\sum_{j=0}^{m-1}\sum_{j=0}^{n-1}(O(i,j) - R(i,j))^2 \tag{2}$$

where m and n are the dimensions ($height \times width$) of the original image (O) and the reconstructed image (R).

$$PSNR = 10\log_{10}\frac{(L-1)^2}{MSE} \tag{3}$$

Here L is the maximum possible intensity value, which is equal to 256 if the image is in 8-bit image representation.

4.3. **Result and analysis.** In the conducted experiments, the performance of the combination of the VGG-16 and U-Net (V-Net) model was measured by comparing it with four different models: the original U-Net [14], U-Net with Inception Blocks [20], V-Net with frozen pre-trained weights, and V-Net with unfrozen pre-trained weights.

In the U-Net with Inception Blocks, the original architecture of the U-Net is added with the Inception Blocks [20], i.e., a variety of filter sizes are added in the convolution layers: 1×1, 3×3, and 5×5. The feature map from the convolution is then concatenated and fed to the subsequent layers. The Inception Blocks are meant to capture features with various kernel sizes; therefore, more important features are obtained from these blocks.

In the V-Net model, we employed the pre-trained weights of VGG-16 for the ImageNet classification. These pre-trained weights are not updated during the training to build the reconstruction model or freeze the pre-trained weights. We called this model V-Net with frozen pre-trained weights. For the last model, the pre-trained weights are updated during the training process to build the model, or it is called in this study as, unfrozen pre-trained weights. Table 2 shows the experimental setup of this study. The setup is set for all models in the experiments to observe the performance focus on the models without any modification to the setup environment.

Table 2. Experimental setup

Name	**Information**
Image size	$112 \times 112 \times 3$
Learning rate	10^{-5}
Batch size	8
Epochs	[100, 200, 300, 400]
Optimizer	Adam
Training and testing set size on original dataset	264 and 66
Training and testing set size on augmented dataset	1056 and 264

In the first experiment, we trained and tested the models on the original dataset, which consists of 330 pairs of images of a face with and without a mask. The results are summarized in Table 3. The result shows that, in general, increasing the number of epochs in all models decreases the MSE value and increases the PSNR value. However, V-Net with unfrozen pre-trained weights obtained the lowest MSE value and the highest PSNR value. These values are plotted in Figure 8.

TABLE 3. Reconstruction performance of each approach on original dataset

Epoch	U-Net		U-Net inception		Frozen V-Net		Unfrozen V-Net	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	0.0221	17.0147	0.0229	16.9087	0.0236	16.7507	**0.0220**	**17.1068**
200	0.0236	16.7833	0.0243	16.6690	0.0261	16.2516	**0.0219**	**17.1263**
300	0.0251	16.4690	0.0234	16.8531	0.0266	16.1585	**0.0219**	**17.1246**
400	0.0254	16.4112	0.0227	16.9765	0.0260	16.2583	**0.0219**	**17.1409**

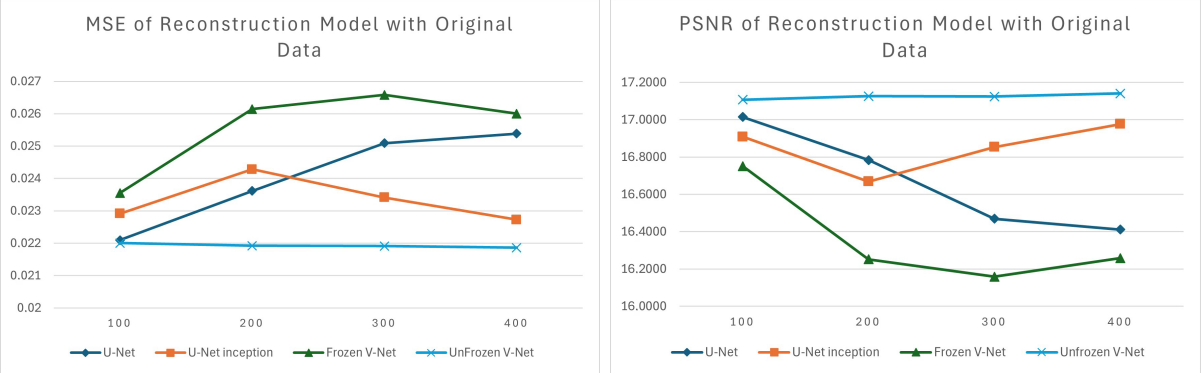


FIGURE 8. Performance on original dataset, MSE curve (left) and PSNR curve (right)

TABLE 4. Reconstruction performance of each approach on augmented dataset

Epoch	U-Net		U-Net inception		Frozen V-Net		Unfrozen V-Net	
	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
100	0.0142	19.1275	0.0151	18.8850	0.0121	20.0596	**0.0081**	**22.2585**
200	0.0097	22.0022	0.0124	20.1473	0.0102	21.3392	**0.0065**	**24.6839**
300	0.0082	23.7165	0.0106	21.2839	0.0098	21.7795	**0.0057**	**26.3980**
400	0.0075	25.02801	0.0096	22.1970	0.0093	22.2620	**0.0054**	**27.7625**

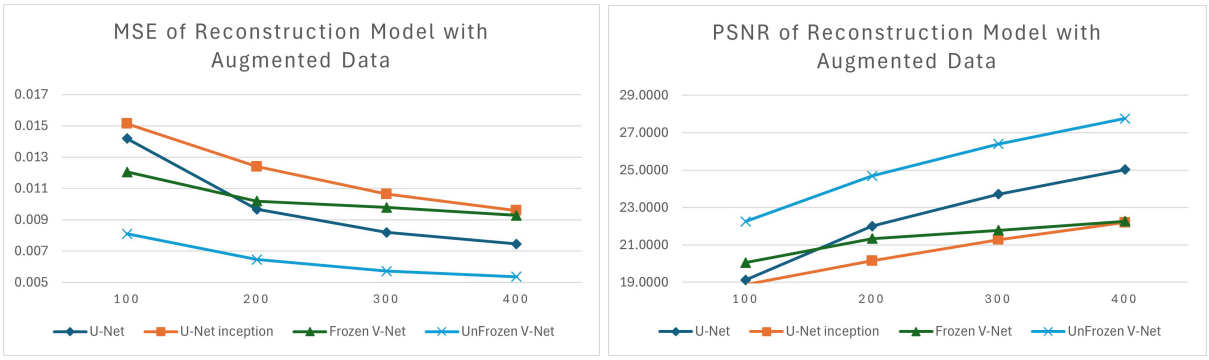


FIGURE 9. Performance on augmented dataset, MSE curve (left) and PSNR curve (right)

For the second experiment, we used the augmented dataset to train and test the models. Table 4 and Figure 9 show the performance of all the models with the augmented dataset. As seen in the results from the first and second experiments, the augmentation of the dataset increases the performance of each approach, i.e., 52.51%, 30.75%, 36.93%, and

61.97% for the PSNR value in U-Net, U-Net with inception, V-Net with frozen pre-trained weights, and V-Net with unfrozen pre-trained weights, respectively.

Training and evaluation using the original and augmented datasets also indicate consistent results, i.e., increasing the number of epochs increases the performance of each model, and the V-Net with unfrozen pre-trained weights outperforms all other models since it achieved the highest PSNR value and the lowest MSE value. Table 4 shows that the unfrozen V-Net increases the PSNR value by 10.93%, 25.97%, and 24.71% compared with the PSNR value of U-Net, U-Net with inception, and frozen V-Net, respectively. The unfrozen V-Net outperforms all other models due to the use of pre-trained weights obtained from the classification model of the ImageNet dataset as the initial weights of the proposed model. Additionally, the pre-trained weights are also updated during the training process according to the dataset of the face image reconstruction model. As a result, the model is more aligned to the specific dataset.

However, in the conducted experiments in the study, adding the inception module in the original U-Net architecture did not improve the performance, whether in the original dataset or the augmented dataset. The lower performance is achieved since the images in the dataset only focus on the face area; hence, extracted features from the original U-Net are sufficient for reconstructing face images.

Figure 10 shows the result of the face reconstruction of all models in this study. As seen in the figure, the V-Net with unfrozen weights reconstructed the best face image than other models. Meanwhile, the U-Net with inception and V-Net with frozen weights are showing less performance compared to others.

FIGURE 10. Result of the reconstruction model. The first row is the input images (a face with a mask), the second row is the reconstruction from the original U-Net, the third row is the reconstruction images from the U-Net with inception, the fourth row is the reconstruction from the V-Net with frozen pre-trained weights, the fifth row is the reconstruction from V-Net with unfrozen pre-trained weights, and the final row is the target images.

5. **Conclusions.** This study presents a face image reconstruction by incorporating VGG-16 in U-Net architecture. In this approach, the feature learning layers of the VGG-16 replace the layers in encoder blocks of the original U-Net architecture. Therefore, the proposed model has deeper layers than the original U-Net. Additionally, in this study,

we also utilize the pre-trained weights of VGG-16 derived from ImageNet classification and update the weights during the training process. The proposed model outperforms all other models in the conducted experiments using the original and augmented dataset. It increases the PSNR value by 10.93%, 25.97%, and 24.71% compared with the PSNR value of U-Net, U-Net with inception, and frozen V-Net, respectively. However, it is important to note that the selection of dataset for training and testing is randomly chosen. As a result, the same subject with different poses or light conditions is possible to appear in the training and also in the testing data. This situation makes the model actually seen the subject in the training process, although with different poses or lights. Therefore, for future research, the face reconstruction model will be developed to reconstruct the unseen data.

Acknowledgment. The authors would like to thank the Indonesian Directorate General of Higher Education (DIKTI) for supporting this research under the Research Grant Number 101/E5/PG.02.00.PL/2024.

REFERENCES

[1] F. Yousaf, S. Iqbal, N. Fatima, T. Kousar and M. S. M. Rahim, Multi-class disease detection using deep learning and human brain medical imaging, *Biomedical Signal Processing and Control*, vol.85, 2023.

[2] K. Thakur, M. Kaur and Y. Kumar, A comprehensive analysis of deep learning-based approaches for prediction and prognosis of infectious diseases, *Archives of Computat Methods in Engineering*, vol.30, pp.4477-4497, 2023.

[3] I. Slimene, I. Messaoudi, A. E. Oueslati and Z. Lachiri, Human disease prediction based on deep and machine learning classification of genes with miRNA binding sites, *Multimedia Tools and Applications*, vol.83, pp.49243-49260, 2023.

[4] M. Jung, J. S. Song, A. Y. Shin, B. Choi, S. Go, S. Y. Kwon, J. Park, S. G. Park and Y. M. Kim, Construction of deep learning-based disease detection model in plants, *Scientific Reports*, vol.13, 2023.

[5] Md. M. Islam, Md. A. A. Adil, Md. A. Talukder, Md. K. U. Ahamed, Md. A. Uddin, Md. K. Hasan, S. Sharmin, Md. M. Rahman and S. K. Debnath, DeepCrop: Deep learning-based crop disease prediction with web application, *Journal of Agriculture and Food Research*, vol.14, 2023.

[6] S. Kumar B P, Ekant, P. Mogallapu, V. Kalmat and Y. Nalla, Crop health monitoring system using deep learning, *International Journal for Research in Applied Science and Engineering Technology*, vol.11, no.5, pp.943-950, 2023.

[7] J. Logeshwaran, D. Srivastava, K. S. Kumar, M. J. Rex, A. Al-Rasheed, M. Getahun and B. O. Soufiene, Improving crop production using an agro-deep learning framework in precision agriculture, *BMC Bioinformatics*, vol.25, 2024.

[8] J. Dhami, N. Dave, O. Bagwe, A. Joshi and P. Tawde, Deep learning approach to predict software development life cycle model, *Proc. of the International Conference on Advances in Computing, Communication, and Control (ICAC3)*, Mumbai, India, 2021.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *Proc. of the 27th International Conference on Neural Information Processing Systems*, 2014.

[10] A. Kumar, M. Kaushal and A. Sharma, SAM C-GAN: A method for removal of face masks from masked faces, *Signal, Image and Video Processing*, pp.1-9, 2023.

[11] F. Farahanipad, M. Rezaei, M. Nasr, F. Kamangar and V. Athitsos, GAN-based face reconstruction for masked-face, *Proc. of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, Corfu, Greece, 2022.

[12] H. Yoshihashi, N. Ienaga and M. Sugimoto, GAN-based face mask removal using facial landmarks and pixel errors in masked region, *Proc. of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2022.

[13] D. Ulyanov, A. Vedaldi and V. Lempitsky, Deep image prior, *International Journal of Computer Vision*, vol.128, pp.1867-1888, 2020.

[14] I. A. Siradjuddin, K. E. Permana and C. V. Angkoso, Face image reconstruction: Removing the mask of the face image using U-Net architecture, *Proc. of the 2023 8th International Conference on Information Technology and Digital Applications (ICITDA)*, Yogyakarta, Indonesia, pp.1-5, 2023.

[15] O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional networks for biomedical image segmentation, *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*, pp.234-241, 2015.

[16] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker and D. Rueckert, Attention U-Net: Learning where to look for the pancreas, *Proc. of the 1st Conference on Medical Imaging with Deep Learning (MIDL 2018)*, Amsterdam, the Netherlands, 2018.

[17] Y. Yan, Y. Liu, Y. Wu, H. Zhang, Y. Zhang and L. Meng, Accurate segmentation of breast tumors using AE U-net with HDC model in ultrasound images, *Biomedical Signal Processing and Control*, vol.72, 2022.

[18] A. Sulaiman, V. Anand, S. Gupta, A. Rajab, H. Alshahrani, M. S. Al Reshan, A. Shaikh and M. Hamdi, Attention based UNet model for breast cancer segmentation using BUSI dataset, *Scientific Reports*, vol.14, 2024.

[19] N. S. Punn and S. Agarwal, Inception U-Net architecture for semantic segmentation to identify nuclei in microscopy cell images, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol.16, no.1, pp.1-15, 2020.

[20] I. Delibasoglu and M. Cetin, Improved U-Nets with inception blocks for building detection, *Journal of Applied Remote Sensing*, vol.14, no.4, 2020.

[21] A. Huang, Q. Wang, L. Jiang and J. Zhang, Automatic segmentation of median nerve in ultrasound image by a combined use of U-Net and VGG16, *Proc. of the IEEE International Ultrasonics Symposium (IUS)*, Xi'an, China, 2021.

[22] S. Ghosh, A. Chaki and K. C. Santosh, Improved U-Net architecture with VGG-16 for brain tumor segmentation, *Physical and Engineering Sciences in Medicine*, vol.44, pp.703-712, 2021.

[23] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *Proc. of the 3rd International Conference on Learning Representations (ICLR 2015)*, pp.1-14, 2015.

[24] Q. Guan, Y. Wang, B. Ping, D. Li, J. Du, Y. Qin, H. Lu, X. Wan and J. Xiang, Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: A pilot study, *Journal of Cancer*, vol.10, no.20, pp.4876-4882, 2019.

[25] L. Yang, S. Xu, X. Y. Yu, H. Long, H. Zhang and Y. W. Zhu, A new model based on improved VGG16 for corn weed identification, *Frontiers in Plant Science*, vol.14, 2023.

[26] U. Sumanth, N. S. Punn, S. K. Sonbhadra and S. Agarwal, Enhanced behavioral cloning-based self-driving car using transfer learning, in *Data Management, Analytics and Innovation, Lecture Notes on Data Engineering and Communications Technologies*, vol.71, Springer, Singapore, 2021.

[27] I. A. Siradjuddin, M. I. Zakaria and M. H. Akhyar, A masked and unmasked face dataset, *Mendeley Data*, Version 2, DOI: 10.17632/xyc9h3wjxf.2, 2025.

Author Biography

Indah Agustien Siradjuddin received the Bachelor degree in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 2002, and Magister degree and Doctoral degree in Computer Science from the Faculty of Computer Science, University of Indonesia, Indonesia, in 2006 and 2010. She is currently a lecturer and researcher at the Informatics Department, Universitas Trunojoyo Madura, Indonesia. Her research interests include artificial intelligence, machine and deep learning for image and video processing. She received research projects funded by the Indonesian Directorate General of Higher Education.

Cucun Very Angkoso received the B.Eng. degree in Electrical Engineering from Brawijaya University, Indonesia, in 2001, and the M.Eng. and Ph.D. degrees in Electrical Engineering from Institut Teknologi Sepuluh Nopember (ITS), Indonesia, in 2011 and 2022, respectively. He is currently a faculty member and researcher at the Informatics Department, Universitas Trunojoyo Madura, Indonesia. His research interests include intelligent systems, pattern recognition, and computational image analysis.

Kurniawan Eka Permana received the B.Sc. degree in Informatics Engineering from Institut Teknologi Sepuluh Nopember, Indonesia, in 2003, and the M.Sc. degree in Computer Science from the University of Technology Malaysia, Malaysia, in 2010. He is currently serving as a lecturer in the Informatics Department at the Universitas Trunojoyo Madura, Indonesia. His research interests include biomedical image and signal processing, machine learning, embedded systems, and their applications in health and agriculture.