Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/18/20230352

Autoencoders and their application in removing masks

Zixiang Liu

Maths department of Imperial College London, London, United Kingdom, SW7 2AZ

lzx556688@hotmail.com

Abstract. Images are frequently distorted by noises that have a negative impact on the quality of image data. In this study, the author focuses on coping with a specific type of noise that has arisen regularly in recent years as a result of the pandemic: masks covering portions of the photographs of human faces. The paper employs the autoencoder model, which offers unsupervised learning. It compresses or encodes original data input into a smaller latent vector, then decodes it back to its original size, learning and extracting relevant features from the data in the process. In a further phase, the author employs a combination of convolutional autoencoders and denoising autoencoders, treating masks as corruptions in order to get more accurate predictions regarding the image of a human face without any covering. After training on 2,500 image pairs with and without masks and validating on 200 such image pairs, the model presented in this research achieves an overall accuracy of 93%. The research demonstrates that the combination of convolutional and denoising autoencoders is an excellent method for removing masks from facial images, and the author believes it can also be used to effectively remove other types of noise. However, the study also reveals that the picture data generated in this manner are always inferior to the original, and that the autoencoder can only process data of the same or comparable type on which it has been trained. In the future, improved models will exist to address these shortcomings and be applied to more real-life situations.

Keywords: convolutional autoencoders, denoising autoencoders, denoising, image reconstruction, mask-removing

1. Introduction

Due to its numerous applications, image processing is currently quite popular. In the actual world, however, photos are frequently distorted by a variety of noises, limiting the application scenarios for our image data. Therefore, it is of utmost importance that we do picture denoising in an efficient manner, so that we can recreate the original images as faithfully as possible, give assistance for the analysis of our photos, and successfully expand the regions in which we may use our acquired image data [1].

There are now numerous study findings in the subject of image denoising. Among these is the use of Total Variation models (or TV for short), which eliminate sounds and recover fuzzy images. An alternative strategy is to construct a Generative Advisory Network (GAN) by training the network with original and noised images until it learns to produce good approximations of original images from images with noise. Both techniques are traditional approaches to denoising, and they achieve comparatively good outcomes in the majority of instances. To be more specific, the Total Variation models provide a higher working efficiency in terms of removing noises and recovering corrupted images; hence, they are more ideal for use with a large number of noised images. When utilizing the

DOI: 10.54254/2753-8818/18/20230352

Generative Advisory Network, however, we tend to get greater precision at the expense of denoising efficiency. In this paper, the author uses the autoencoder model, which is different from the two methods mentioned above, and combines denoising with convolutional autoencoders in order to remove noises and replicate the original images. The benefits of using a denoising autoencoder are that it permits unsupervised learning and is excellent at recognizing, predicting, and representing original visual data. In practice, the author also constructs a convolutional autoencoder to perform denoising, so enhancing the quality of reconstruction.

This publication has contributed to the study of image data denoising methods, and it also aids in the applications of denoising, particularly during the pandemic, when masks become a common "noise" in face identification. Using my autoencoder model, the author will demonstrate this with an example of attempting to predict the genuine human faces hidden behind masks. The author believes that this method may be put to widespread, regular use since it can be used to unlock our smart devices without having to remove masks and to identify criminals even if they are wearing face coverings based on photographs collected at crime scenes.

2. Autoencoder

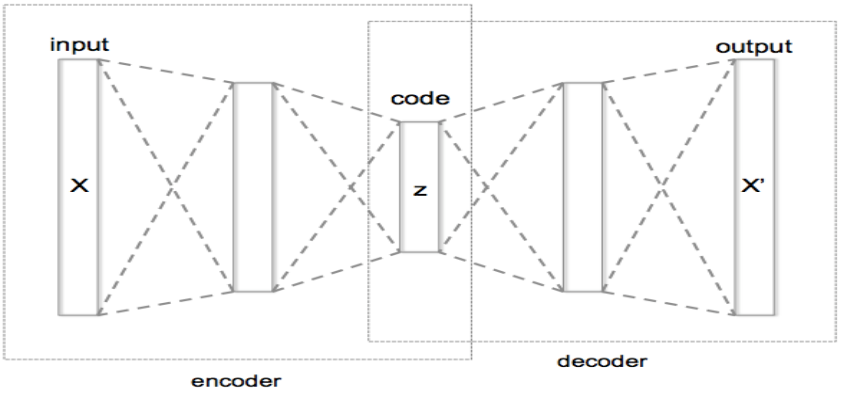


Figure 1. Schematic picture of an autoencoder architecture [2].

An autoencoder is a sort of artificial neural network used to learn effective encodings of unlabelled input [3]. Figure 1 depicts the three parts of an autoencoder: the encoder, which compresses the data; the code, which deals with the compressed data, and the decoder, which restores the original size of the picture data that was compressed. When a piece of image data x is fed into an autoencoder, it first passes through the encoder, where it is compressed down into a latent-space representation by the transformation $s = E(x)$. We usually call this latent representation the “code”. After that, the compressed data message passes through the decoder and is reconstructed there to get an output $o = D(x)$, where the function D denotes the decoding process. Our goal is to coerce the network into identifying the fundamental features of the data we provide it. It is possible to provide output that is comparable to our input data x by creating a model to minimize the dissimilarity between the two after training the network, or to have our learnt representations capture more valuable attributes. The encoder function can be used to add a level of mathematical precision to this procedure:

$$E_{\varphi}(x) = \sigma(Wx + b)$$

where σ{\displaystyle \sigma }σ is an element-wise activation function such as a sigmoid function or a rectified linear unit, W{\displaystyle W} is a matrix called "weight", and b{\displaystyle b}b is a vector called "bias", and the decoder function:

$$D_{\theta}(z) = \sigma'(W'z + b')$$

DOI: 10.54254/2753-8818/18/20230352

We need to make our reconstructed data sample the same as our original input x, that is,

$$x \approx \sigma'[W'\sigma(Wx + b) + b']$$

or

$$x \approx D_\theta(E_\varphi(x))$$

To put it another way, we have to learn an identity function. There are various matrices that can be applied to quantify the differences between the input and output, such as cross entropy where the activation function is sigmoid, or as simple as MSE loss:

$$\mathcal{L}(\theta, \varphi) = \frac{1}{n}\sum_{i=1}^{n} ||D_\theta(E_\varphi(x^{(i)})) - x^{(i)}||^2$$

which is required to be minimised.

Sparse autoencoders, contractive autoencoders, variational autoencoders, and convolutional and denoising autoencoders are the variants on the traditional autoencoder paradigm [4]. In my experiment, the author employed a hybrid of the last two types of autoencoders to make an accurate prediction of unmasked human faces.

3. Convolutional autoencoder

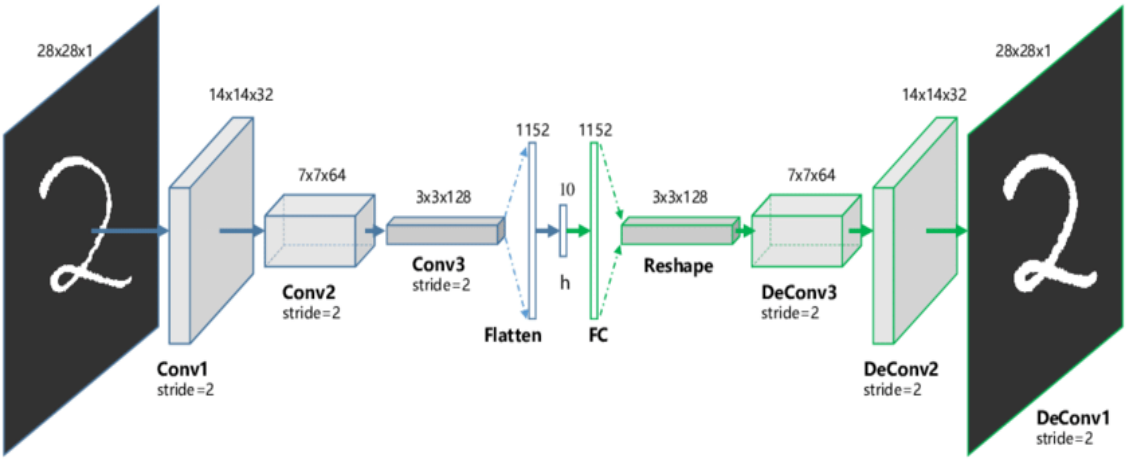


Figure 2. The structure of proposed Convolutional AutoEncoders (CAE) for MNIST [5].

For input, a convolutional autoencoder uses subsegmentation and convolution layers to reduce complex signals to their simplest forms, as shown in Figure 2. A new representation of the data is then derived by adding the signals together. A convolutional autoencoder, like a convolution neural network, may efficiently learn new images by applying a filter that is shifted across the entire image section by section. An image's geometry can be transformed, reflected, or reconstructed using the encodings produced by the encoding layer. As soon as the network has learned the filters, they may be used to extract image attributes from any input that is similar enough [4].

DOI: 10.54254/2753-8818/18/20230352

4. Denoising autoencoder

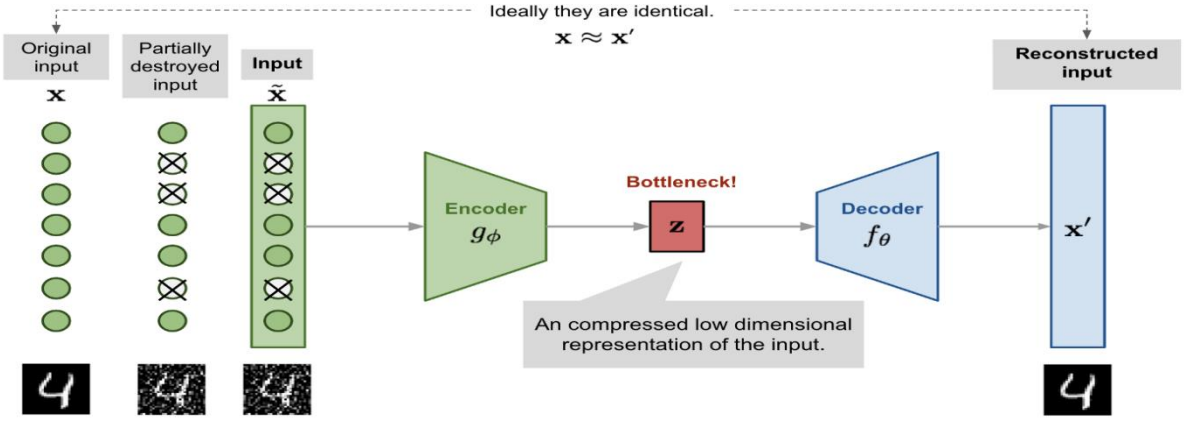


Figure 3. Illustration of denoising autoencoder model architecture [6].

In order to process noisy images as input, a denoising autoencoder must be used. By adjusting the reconstruction criterion, it strives to produce reasonably close representations of the original images [7]. The fundamental difference between this and a regular autoencoder is the addition of a new "adding the noise" technique, wherein we systematically corrupt our photos by introducing noise into the input, hence compelling our model to extract key features from the images. After training the network on noisy photos, we can use the resulting model to clean up noisy input images and produce results that are visually identical to the originals but without the noise.

We first add noise to the original data samples:

$$\tilde{x}^{(i)} \sim M_D(\tilde{x}^{(i)}|x^{(i)})$$

Where M_D stands for the mapping from the true data samples to the corresponding noisy ones.

Then, in a manner similar to but distinct from the ordinary autoencoder situation, we minimize the difference between our outputs and the original images rather than the corrupted input:

$$min\big(\mathcal{L}(\theta,\varphi)\big) = min\,(\frac{1}{n}\sum_{i=1}^{n}||D_\theta(E_\varphi(\tilde{x}^{(i)})) - x^{(i)}||^2)$$

We need to note that the use of denoising autoencoder depends on two assumptions:

●There exist representations to the messages that are relatively stable and robust to the type of noise we add;

●The said representations capture useful structures in the input distribution [8]

In principle, a noise process is defined by a probability distribution μ_T {\displaystyle \mu _{T}} over functions T: X → X {\displaystyle T:{\mathcal {X}}\to {\mathcal {X}}}, that is, the function T{\displaystyle T} takes a message x ∈ X{\displaystyle x\in {\mathcal {X}}}, and corrupts it to a noisy version T(x){\displaystyle T(x)}. The function T{\displaystyle T} is randomly selected with a probability distribution μ_T {\displaystyle \mu _{T}} [9]. In my “removing masks” experiment, though, the added noise is the image of masks covering part of the human faces in the pictures collected, instead of some random noise process.

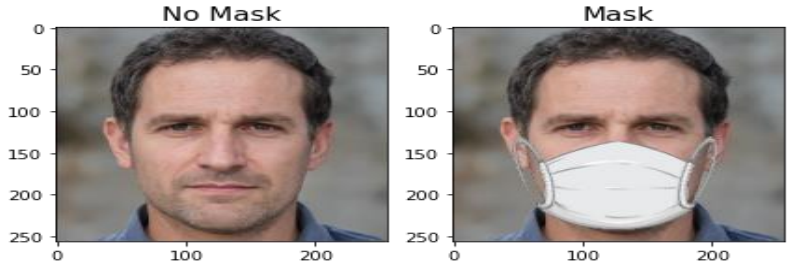
5. Practice of the autoencoder model

5.1. Procedures

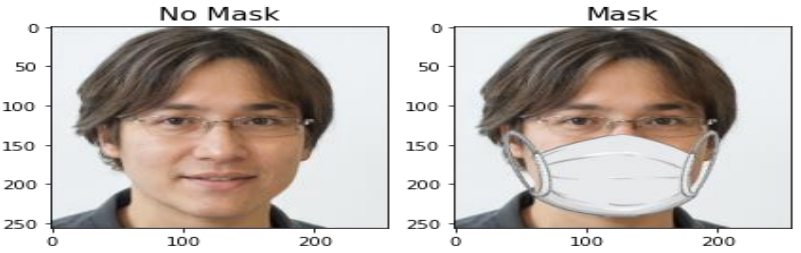
Following this theoretical discussion, the author will implement the combination of convolutional and denoising autoencoders in a practical setting, specifically, the removal of a face mask.

DOI: 10.54254/2753-8818/18/20230352

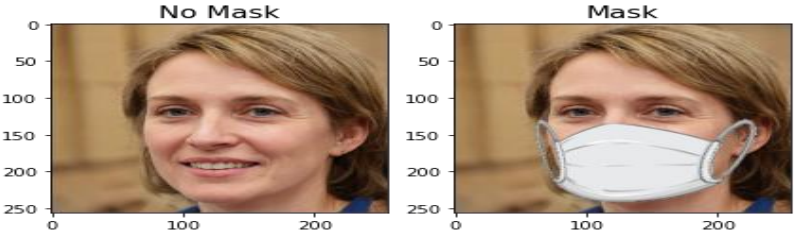
It must first acquire an adequate dataset of faces and photos of those faces covered up with masks. Since it is difficult to obtain a significant number of image pairs in which all the faces are in the same state in both the mask-on and mask-off cases, this presents the biggest problem. To save time and effort inviting and photographing a large number of applicants, the author relied on a Kaggle dataset containing 7,219 photographs of human faces, each with a resolution of 256 x 256. Our model is more generalizable and unbiased because the dataset contains photographs from people of diverse religions, ethnicities, ages, and demographic profiles, as well as some GAN-generated images. Similar to the "adding the noise" technique, the author picked the first 2,600 photographs and added masks to them. The pictures are transformed into arrays and added to a previously empty array. Some of the image pairs the author has created are shown here.



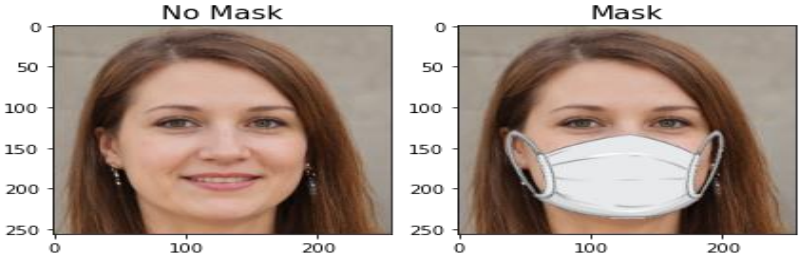
(a)



(b)

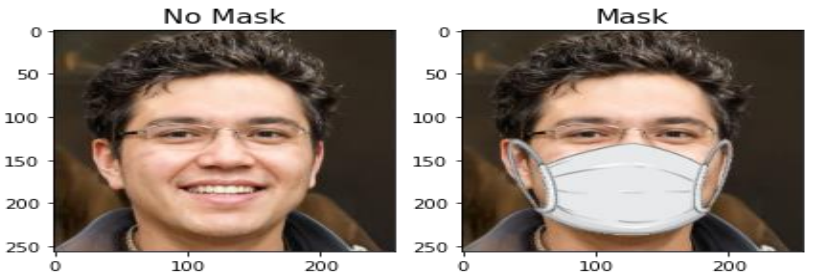


(c)



(d)

DOI: 10.54254/2753-8818/18/20230352



(e)

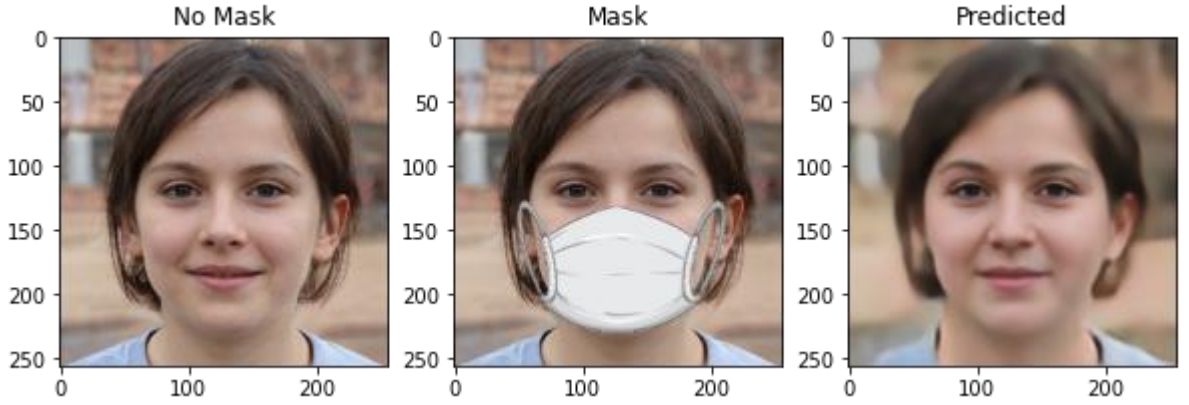
Figure 4. Image pairs of human faces with and without masks.

The first 2,400 of my image pairs were used for training, while the last 200 were used for validation. Afterward, the author began constructing my autoencoder model. Since the convolutional autoencoder performs much better than other methods when it comes to processing images, the author has chosen to employ it here. Convolution2D was employed, with a 3 * 3 kernel, 2 * 2 strides, and "ReLU" as the activation function. The author further utilized MaxPooling2D for the encoder network, setting the pool size to 2 * 2. The encoder network then compressed image of shape (256, 256, 3) to shape (16, 16, 64). The author ultimately obtained the code, or the latent vector, from this condensed form. When the code was fed into the decoder network, the author utilized Convolution2D and UpSampling2D to resample the latent vector in an effort to rebuild the images and cut down on the reconstruction loss. The author employed the identical parameters and activation function in the decoder network to those that were employed in the encoder network to ensure the two processes exactly opposite. The decoding process uncompressed the photos to their original size but removed masks as corruptions.

Then the author compiled the model using optimizer Adam and set loss equal to mean absolute error, training the model on 2,400 paired photos of human faces with and without masks (epoch is set to 100). After that, the author conducted an evaluation of the model using validation image pairs, and the author was pleased to see that the model achieved an accuracy of 93%.

5.2. Results

Finally, the author tested the autoencoder model on another set of 100 photos following the 2,600 images that dad already been used, to ensure the model was robust. Parts of the results are presented in the plots below after the author applied masks to these face photos and fed them into the autoencoder model.



(a)

DOI: 10.54254/2753-8818/18/20230352

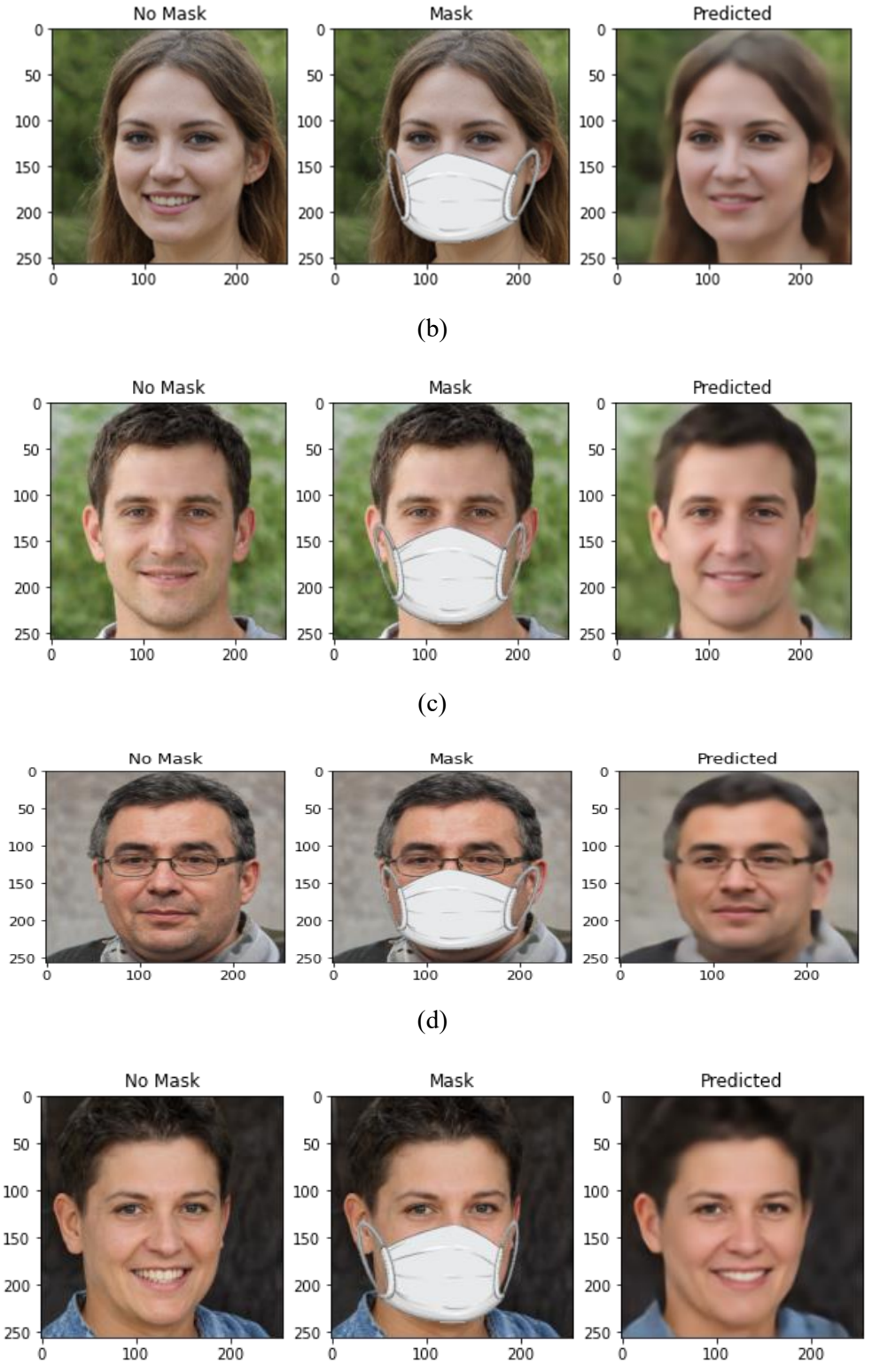


Figure 5. results of prediction using my autoencoder model.

DOI: 10.54254/2753-8818/18/20230352

The predicted images show a high degree of similarity to the genuine, authentic human faces; this is compelling evidence that the combination of convolutional and denoising autoencoder is useful in ideal settings. We do find, however, that some details from the source photos are lost in the predictions, and that the predicted images are often blurrier, as seen in figure 5. This finding shows one drawback of using autoencoders is that the model's outputs will be deteriorated compared to the input data [4].

6. Conclusion

Ultimately, using an autoencoder, this research aims to de-mask real human faces hiding behind masks and make accurate predictions about them. In order to accomplish the objective of eliminating masks from facial photos captured under ideal conditions, the article introduced the convolutional autoencoder and the denoising autoencoder, using a combination of these two types of autoencoders to construct an effective model. This model adds a new method to our toolbox of denoising image data which includes the application of total variation model or building a generative advisory network. The benefits and drawbacks of utilizing an autoencoder were also investigated during the course of this study. Autoencoders' ability to facilitate unsupervised learning demonstrates their value. They save a ton of time and effort because they learn on their own and do not require labelling, which means they can be used right away. Furthermore, under ideal circumstances, they produce rather accurate predictions given sufficient data of a specific type of input. However, autoencoders are lossy because the quality of the prediction images produced by them is always worse than that of the original input before noise is introduced.

The author conducted an experiment in which he restricted his images of human faces to those taken in ideal situations. The author believes that it will be of tremendous benefit to apply the autoencoder model in more realistic and complex scenarios in the future. One such scenario is the identification of criminals who may be hiding their faces behind masks or other types of face coverings. To construct such a model, we need to enhance our current model of autoencoders so that they can make accurate predictions even when there are many confounding variables and much fewer training samples available. But once we get through the hurdles and make this improved autoencoder model a reality, it will be used in more contexts and improve our lives.

References

[1] He Chuan, Hu Changhua, Qi Naixin, et al. Fast proximal splitting algorithm for constrained TGV-regularised image restoration and reconstruction[J]. IET Image Processing, 2019, 13(4): 576-582

[2] Chervinskii. Own work. CC BY-SA 4.0. 2015.

[3] Kramer, Mark A. "Nonlinear principal component analysis using autoassociative neural networks" (PDF). AIChE Journal, 1991, 37 (2): 233–243.

[4] Daniel Nelson. What is an Autoencoder. 2020. www.unite.ai.

[5] Guo, Xifeng & Liu, Xinwang & Zhu, En & Yin, Jianping. Deep Clustering with Convolutional Autoencoders. 2017: 373-382.

[6] Lilian Weng. From Autoencoder to Beta-VAE. lilianweng.github.io, 2018

[7] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. Deep Learning. MIT Press, 2016.

[8] Vincent, Pascal; Larochelle, Hugo. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". Journal of Machine Learning Research, 2010, 11: 3371–3408.

[9] Wikipedia contributors, "Autoencoder," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Autoencoder&oldid=1109263923 (accessed September 28, 2022).