IMPLEMENTATION OF DEEP LEARNING WITH ARTIFICIAL NEURAL NETWORK ARCHITECTURE FOR IMAGE CLASSIFICATION USING AUTOENCODER TECHNIQUE

Elit Ahmad Zen [1], Dimas Fajar Maulana [2], [3], [4],
[1-4] JL. Anggrek No. 1 Village Pendem Junrejo District
Batu Department of Electrical Engineering , D4 Engineering Study Program ,
Elkasista Poltekad Kodiklatad
E- mail: elitahmadzen42@gmail.com [1], dimasf380@gmail.com [2], [3]

Implementation Of Deep Learning With Artificial Neural Network Architecture For Image Classification Using Autoencoder Technique

Abstract *- Digital image processing is rapidly evolving along with advances in artificial intelligence technology, particularly in the field of deep learning. In this context, the use of Artificial Neural Network (ANN) architecture has proven effective in improving image classification performance. The main objective of this study is to integrate autoencoder techniques into the ANN structure to improve accuracy in the image classification process. Autoencoders, which are unsupervised learning methods, function to extract important and representative features from a given image. These features are then used as input for the classification layer in a neural network. In this experiment, a carefully curated image dataset was used to train the model. After training, the model was tested and evaluated based on several performance metrics, including accuracy, precision, and recall. The test results significantly showed that the addition of autoencoders in the ANN architecture provided a significant increase in classification accuracy compared to conventional approaches that did not use this technique. These findings prove that autoencoders can play a significant role in improving the quality of deep learning-based classification systems, especially in applications that require more accurate and efficient image analysis.*

Keywords: *Deep Learning, Artificial Neural Network, Autoencoder, Image Classification, Feature Extraction.*

INTRODUCTION

One type of deep learning architecture called an autoencoder aims to learn data representations by reconstructing the original input. The two main parts of an autoencoder are the encoder and the decoder. The encoder transforms the data into a latent representation, or code, that is smaller than the original data. Conversely, the decoder reconstructs the original data from this latent representation. Autoencoders can identify and preserve important features of the data while ignoring irrelevant or extraneous information through this process (Zamachsari & Puspitasari, 2021a).

Autoencoders are crucial for their ability to simplify complex data representations by reducing their dimensionality. This facilitates better data processing and improves model performance for specific tasks such as classification, anomaly detection, and data visualization in low-dimensional spaces. Furthermore, autoencoders can be used to improve data quality by reducing noise (Hartono, 2020).

Some crucial factors to consider when designing an autoencoder include selecting the right network architecture, such as the number of layers and neuron units, as well as selecting an appropriate activation function and optimization method. The size of the latent representation is also crucial; if it is too small, important information may be lost, while if it is too large, it will reduce the autoencoder's ability to extract relevant features, thus affecting the overall model performance (Zamachsari & Puspitasari, 2021b).

Students will learn and apply image reconstruction methods using autoencoders in this chapter. In addition, autoencoder denoising, a very popular development of the autoencoder method, will be discussed. Autoencoder denoising is a technique for reconstructing original data that has been subjected to noise or interference. The main function of denoising is to increase the model's resilience to noise in the data and improve data quality by removing noise. Applications such as data preprocessing, image restoration, and pattern recognition in contaminated data greatly benefit from using this technique. (Alamsyah et al., 2022).

Basic theory

An autoencoder is an artificial neural network architecture designed to reduce data size, extract features, and learn representations. Autoencoders fall under the unsupervised learning paradigm, meaning they learn from unlabeled data. Autoencoders have two main functions. First, they compress the input data into a smaller representation, and second, they recode the data from the compressed representation. Figure 1 shows an example of image processing using an autoencoder architecture (Septia Nugraha & Ferdinandus Pardede, 2022).

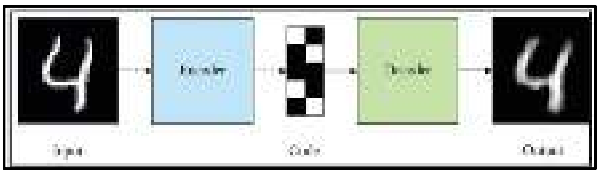


Figure 1. Image processing using an autoencoder. The encoder converts data into a more compact representation. The decoder converts the representation into the original data.

2.1 Components in Autoencoder

An autoencoder generally consists of three main components:

Encoder:

- It consists of multiple layers of neural networks that gradually compress high-dimensional data into a lower-dimensional representation (latent space) (Septia Nugraha & Ferdinandus Pardede, 2022).
- Each layer uses a linear transformation followed by a non-linear activation function such as ReLU, sigmoid, or tanh, which allows for the extraction of increasingly abstract and concise features from the input data (Fadlullah et al., 2025).
- The encoder reduces redundancy by removing unimportant or uninformative features (Kemendikbud et al., 2021).

Latent Representation (Code):

- It is the core and most critical part of the autoencoder, located between the encoder and decoder (Kemendikbud et al., 2021).
- In general, the input layer has a much smaller number of neurons, forcing the network to identify and store the most important data attributes.
- By using this bottleneck, concise and effective abstract features are created from the original data (Judijanto, 2025).

Decoder:

- It has a structure that is a mirror image of the encoder, which is responsible for recovering the original data from the latent representation.
- To ensure accurate reconstruction of the original data, it uses a network architecture similar to the encoder, usually symmetric.
- The decoder attempts to reduce the reconstruction error, known as reconstruction error, so that it can produce the least complex data with the original input (Maulana et al., 2020).

2.2 Types of Autoencoders

Vanilla autoencoders are the simplest type of autoencoder, aiming to learn a well-defined data representation. They work by using an encoder to compress the input data into a smaller space, then attempting to reconstruct the original data using a decoder (Li et al., 2024). By training the autoencoder to minimize the difference between the reconstructed data and the original data, the model is able to identify the most important patterns or features in the data. In a military context, vanilla autoencoders have several advantages, particularly in terms of effective data processing, analysis, and management. Here are some specific advantages:

1. Dimensionality reduction or data compression: Autoencoders help the military save storage space and speed up data transmission over limited or encrypted communication networks. This is especially useful when working in remote locations with limited bandwidth.
2. Autoencoders can easily reconstruct normal data. However, they struggle to reconstruct unusual or anomalous data. Finding anomalies is easier by finding differences between the original and reconstructed data. This is known as reconstruction error.

Denoising autoencoders are designed to remove interference, also known as noise, from distorted or contaminated data. Denoising autoencoders intentionally introduce noise into the input data. The autoencoder is then trained to reconstruct the original, clean data from the perturbed version. This differs from a traditional autoencoder that uses clean data as input. This method allows the model to learn robust features and distinguish important signals from irrelevant noise. In addition to data preprocessing, denoising autoencoders can also be used to improve the quality of degraded images and clean audio signals from background noise (Alamsyah et al., 2022). Figure 2 shows the data processing process for denoising autoencoders.

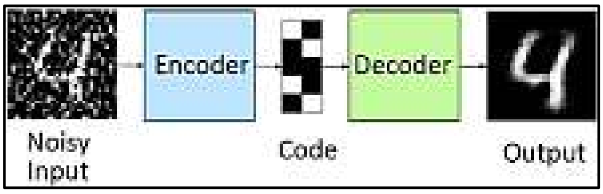


Figure 2. Image processing process with denoising autoencoder, where dirty images are processed to become clean images.

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1$$
$$H_{out} = \frac{H_{in} - F + 2P}{S} + 1$$

- W_{out}AndH_{out} is the width and height of the feature map
- W_{in}AndH_{in} is the width and height size of the input image
- $F$is the size of the filter
- $S$is stride
- $P$is the amount of padding

RESEARCH METHODS

In this lab, we will explore how to process image-based data using two different Replication Neural Network architectures: MLP and CNN. Two different datasets will be used in this lab: the first is MNIST, and the second is CIFAR-10.

No	Latent Size	MSE 0.3
1	4	0.0234
2	8	0.0143
3	16	0.0125
4	32	0.0060

MNIST is a dataset containing thousands of handwritten dark and white drawings ranging from 0-9. Each painting has dimensions of 28x28 pixels. On the other hand, CIFAR-10 contains thousands of patterned paintings which are classified into 10 types (planes , cars, birds, cats, deer, dogs, frogs, horses, ships and trucks). Each painting has dimensions of 32x32 pixels (Raharjo, 2022).

Understanding Notebook Structure

1. The initial notebook contains the models used to cluster the MNIST number images using MLP and CNN methods. MNIST is a dataset containing thousands of images with the numbers 0-9 and black and white. Each image has dimensions of 28x28 pixels and one channel (Sely Wita & Yanti Liliana, 2022).

2. The second notebook contains the model used to perform the denoising or image removal process on the MNIST dataset using the CNN method. The image is created by adding noise or disturbances derived from Gaussian noise to the original image.

No	Latent Size	MSE (Noise factor = 0.9)
1	2	0.0477
2	4	0.0390
3	8	0.0341
4	32	0.0284

3.1.2 Understanding Tunable Parameters.

In both notebooks, there are several parameters that can be modified to improve model performance. Some of the main parameters we will explore include:

- **The size of the latent representation (code)** – Sets how much information compression we want.
- **Noise Factor** – Noise aspect to determine how much Gaussian

noise will be added to the original painting.

3.1.3 Experiment and Evaluation

- We will retrain the model and analyze the results after changing certain parameters.
- The main goal of this experiment is to find out how the size of the latent representation (code) and the noise factor affect the image reconstruction and cleaning process (Lee et al., 2021).

3.1.4 Comparing Results

- After conducting several experiments with different latent representation parameters (code) and noise factors, we will compare the quality of the decoder output results.
- This analysis will help understand how certain parameters affect the model performance as well as how to improve the quality of the image reconstruction process (Arijona et al., 2020).

RESEARCH RESULT

3.1` Comparing Results

After experimenting with various noise factors and latent representation (code) parameters, we will compare the quality of the results from the decoder. This analysis will help us understand how certain parameters affect model performance and how to improve the image reconstruction process (Raharjo, 2022).

3.2 Running the program Experiment Data

Conduct an experiment by running the first notebook **(cnn_autoencoder_ latent.ipynb).** After conducting the experiment, record the Mean Squared Errors (MSE) results from the image reconstruction results and explain why such results can be obtained.

Analysis:

- The larger the latent size, the more data can be retained throughout the painting reconstruction.
- The MSE decreases because with more parameters, the autoencoder can include more details from the original painting, resulting in a closer re-creation of the original.
- However, overfitting can occur due to too large a latent size, while too small a latent size can result in the loss of important details.(Guillem Boquet, 2021)

Conclusion:

The reconstruction quality is affected by the latent size; the larger the latent size, the lower the reconstruction error; however, to keep the model efficient and avoid overfitting, the latent size must be well adjusted.

We found in our experiments that the larger the given latent value, the smaller the Errors (MSE) value is affected.

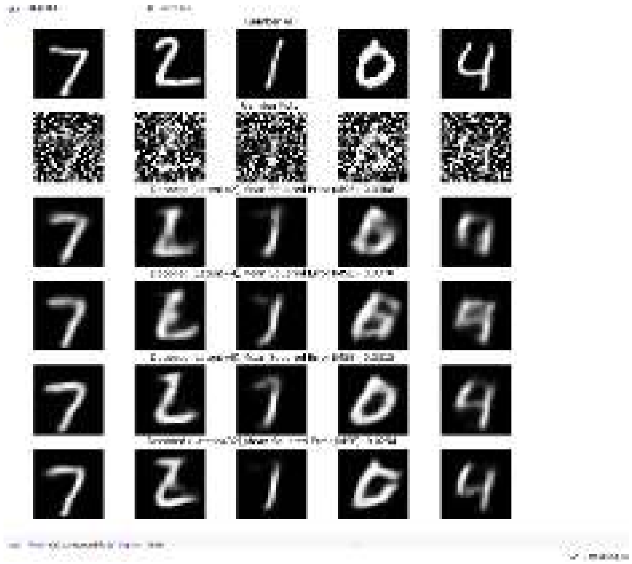
The experiment was carried out by running the second notebook **(cnn_denoising_ autoencoder.ipynb)** then recording the MSE value with the test data.

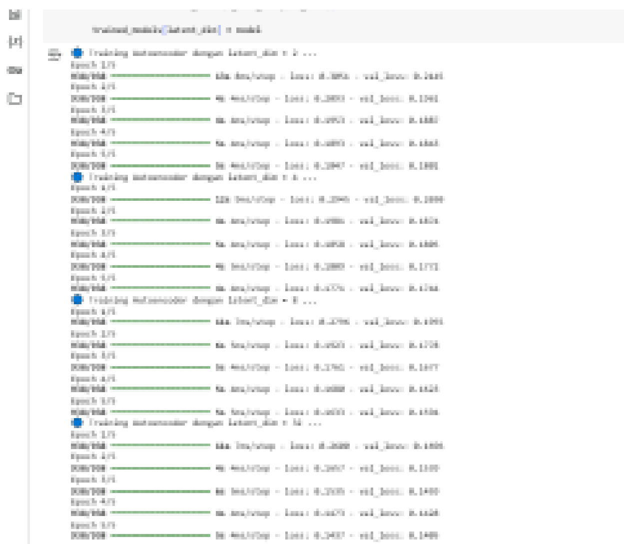
Analysis:

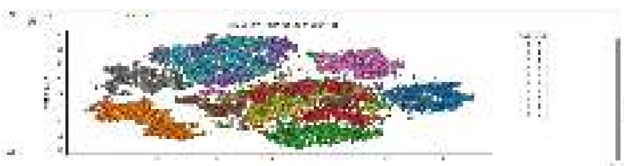
As the latent size increases, the MSE decreases because the model can retain more information to reconstruct a more noise-free image. Poorer reconstructions are achieved with smaller latent sizes because too little information is retained, but with larger latent sizes, models can capture more features from the original image, allowing them to more effectively remove noise (Cohen et al., 2025).

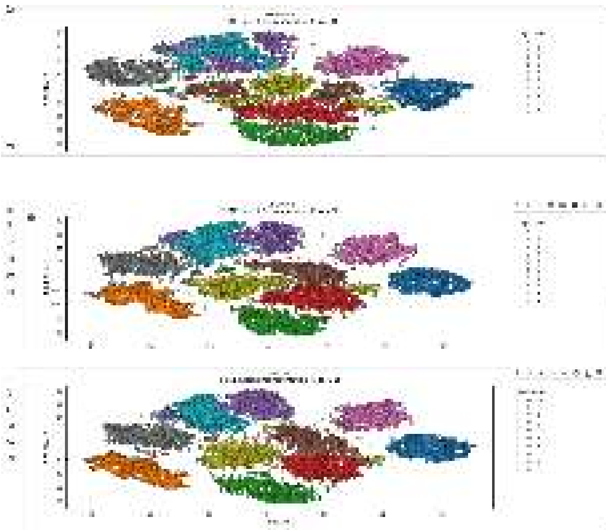
Conclusion:

With a larger latent size, the autoencoder can better remove noise from the image. Furthermore, it produces a lower error rate. However, a balance is important to maintain model efficiency and prevent overfitting. T-SNE plots of the most effective and least effective experimental models are shown. The results are demonstrated (Faysal et al., 2025).









DISCUSSION

This lab aims to gain an understanding of how autoencoders, a type of artificial neural network (ANN), can be used to reconstruct images and how different parameters, such as the size of the latent representation and the noise factor, affect the model's performance.

An autoencoder consists of two main parts: an encoder and a decoder. The encoder converts the input image data into a smaller representation (called a latent code), and the decoder recovers the image from the latent code. Reconstruction performance can be measured using the Mean Squared Error (MSE), which indicates the difference between the original and reconstructed images.

The first experiment involved varying the size of the silent latent representation. The results showed that the latent size used was negatively correlated with the resulting MSE values.

This shows that larger latent dimensions have the ability to retain more data from the original image, resulting in more accurate reconstructions. For example, the MSE at a latent size of 4 was 0.0177, but the MSE dropped to 0.0064. This indicates that with increased representation capacity, the model can extract more complex features from the input image.

To test the robustness of the model, a second experiment involved adding noise to the input data. The MSE values for each latent size increased as a result of the addition of noise; however, the decreasing trend of MSE with increasing latent size remained consistent. For example, the MSE reached 0.0388 at a latent size of 2 with noise, but dropped to 0.0254 at a latent size of 32. This indicates that the autoencoder can still reconstruct images well despite the presence of noise in the input, especially in cases with sufficient latent size.

From these two experiments, it can be concluded that:

1. The quality of the reconstruction results is significantly affected by the size of the latent representation; the larger the latent size, the better the reconstruction quality, as indicated by the decrease in the MSE value.

2. The addition of noise degrades the reconstruction quality, but the model still exhibits stable performance as the latent size increases. This demonstrates the autoencoder's ability to be a robust preprocessing technique against noise.
3. Consider the difference between model accuracy and complexity. While larger latent sizes yield better reconstruction results, they also increase computational complexity.

Overall, this study shows that understanding the parameters of autoencoder architecture is crucial for creating efficient models in image reconstruction tasks.

Discussion of the results of the CNN autoencoder practicum.

In this experiment, we tested and applied a Convolutional Autoencoder model with various latent dimensions to assess its impact on the quality of the painting reconstruction. Based on observations of the Mean Squared Error (MSE) values, we found that the larger the latent dimensions used, the smaller the MSE value. This indicates that the closer the reconstruction is to the original painting.

Technically, a larger potential dimension provides a greater representational capacity for the autoencoder to store meaningful data from the input view. Ultimately, the image can preserve more spatial and compositional details of the image during the encoding and decoding processes, resulting in a more accurate reconstruction of the image.

Conversely, when the potential dimensions are too small, the resulting representation becomes too simple and unable to fully represent the complexity of the painting's features. This situation results in the loss of significant details, which results in an increase in the MSE score, indicating that the reconstruction deviates from the original painting.

However, it's important to note that excessively increasing the potential dimension isn't always beneficial. While the error rate decreases, using a very large potential dimension can lead to overfitting, where the model adapts too much to the training data and fails to abstract from the experimental data. Furthermore, a large potential dimension also increases computational requirements and training time.

Conclusion of Discussion:

From the results of this lab, it can be concluded that determining the appropriate potential dimension is a crucial aspect of the autoencoder architecture concept. Optimal potential dimension will create a balance between reconstruction accuracy (low MSE) and shape performance, thus optimizing the autoencoder's performance without overfitting or excessive computational overhead.

Discussion of Practical Results – Denoising Autoencoder

In this experiment, we tested and applied a CNN-based Denoising Autoencoder to see how changes in latent space dimensions affect the model's ability to reconstruct noisy images. The Mean Squared Error (MSE) was used as a metric to assess the quality of the image reconstruction results from noisy experimental data.

Observations show that the larger the potential dimension used, the lower the MSE value obtained. This indicates that shapes with larger potential dimensions can retain more accurate and accurate data from the original image, even if it is distorted by noise. The autoencoder is more efficient at identifying important patterns in the image and ignoring random or meaningless data (noise), resulting in a cleaner reconstruction that is closer to the original image.

Conversely, at small potential dimensions, the model has limitations in capturing significant features of the image, resulting in the resulting reconstruction being incomplete or lacking significant details. This is because the resulting representation is too simple and unable to efficiently distinguish between significant features and noise.

However, as with conventional autoencoders, using too large a potential dimension can also lead to overfitting, where the model remembers too much of the training data and cannot perform well on new data. Therefore, a compromise is needed between the potential dimension, performance, and performance of the model.

Conclusion of Discussion:

From these experimental results, it can be concluded that the potential dimension significantly influences the autoencoder's denoising ability. A larger potential dimension generally produces cleaner and more accurate reconstructions, with a lower MSE. However, it is necessary to determine the optimal potential dimension so that the model is not too complex and still performs a good abstraction of the experimental data.

t-SNE Visualization Analysis:

To strengthen the analysis, visualization of the latent representation results was carried out using the **t-distributed Stochastic Neighbor Embedding (t-SNE) method** on two models with the best and worst performance based on the MSE value.

1. Best Model(large potential dimension): The t-SNE flow shows a clear clustering along the image categories, which indicates that the obtained potential representation has a good and dense feature division, thus facilitating the reconstruction and noise removal process.

2. Worst Model (small potential dimension): The t-SNE plot appears more scattered and unorganized, indicating that the potential representation is not informative enough or is too overlapping between categories, resulting in reduced reconstruction quality and difficult to remove noise efficiently.

CLOSING

Experiments conducted using Convolutional Autoencoder and Denoising Autoencoder models show that understanding architectural parameters, particularly the latent size, has a significant impact on how well digital image reconstruction works. Empirically, these experiments show that the larger the latent size, the more information can be stored in the encoding process, and the larger the latent size, the more information can be stored in the encoding process.

Using a standard autoencoder without noise, it was shown that a large latent dimension helps the model capture important features such as contours, textures, and pixel distributions of the input image. This allows the reconstruction results to retain relevant visual details. However, it is important to note that increasing the latent dimension also directly impacts model complexity and computational burden. Furthermore, if not balanced with a robust regularization or validation mechanism, overfitting may occur.

Furthermore, in the autoencoder denoising experiments, the model was tested under more challenging conditions, namely adding noise to the input image. Here, a larger latent size indicates that the model is better able to distinguish between the original image features and irrelevant noise information. Therefore, autoencoders can not only be used as compression and reconstruction tools but also have outstanding capabilities for image purification (image denoising), which is very beneficial for various applications such as medical image processing, surveillance systems, and facial recognition.

To improve the results of numerical analysis, visualization of latent representations was performed using the t-distributed Stochastic Neighbor Embedding (t-SNE) method. The t-SNE results for the best performing model (large latent size) show dense, structured, and separable clustering of latent data between classes, indicating that the model is able to filter and group features efficiently. In contrast, the t-SNE results for the poor performing model (small latent size) show

Thus, it can be concluded that:

1. Better reconstruction quality with smaller MSE is obtained from standard

autoencoder and denoising because larger latent size increases the risk of overfitting and requires larger resources.

2. It is important to note the difference between accuracy and computational efficiency because larger latent sizes can increase the risk of overfitting.

3. Autoencoders can be used for compression and reconstruction as well as to overcome noise disturbances in digital images.

4. Visualization of t-SNE latent representations provides a clear picture of the effectiveness of encoding and can be used as a tool for qualitative analysis and understanding of model performance.

Overall, this practicum enhances participants' understanding of the basic concepts of deep learning, convolution-based ANN modeling, and the importance of parameter selection in designing an effective image reconstruction system.

BIBLIOGRAPHY

Alamsyah, N., Fauzan, MN, Putrada, AG, & Pane, SF (2022). Autoencoder Image Denoising to Increase Optical Character Recognition Performance in Text Conversion. *ICACNIS 2022 - 2022 International Conference on Advanced Creative Networks and Intelligent Systems: Blockchain Technology, Intelligent Systems, and the Applications for Human Life, Proceedings* . https://doi.org/10.1109/ICACNIS57039.2022.10054885

Arijona, M., Yohanes Stefanus, L., Arijona Sinaga, M., & Yohanes Stefanus, L. (2020). *Least Square Adversarial Autoencoder* . https://www.researchgate.net/publication/344516945

Cohen, E., Achituve, I., Diamant, I., Netzer, A., & Habi, H. V. (2025). *Efficient Image Restoration via Latent Consistency Flow Matching* . http://arxiv.org/abs/2502.03500

Fadlullah, R., Winarno, S., & Naufal, M. (2025). Integration of Convolutional Autoencoder with Support Vector Machine for Almond Variety Classification. *Journal of Informatics Engineering and Information Systems* , *11* (1), 63–77. https://doi.org/10.28932/jutisi.v11i1.9738

Faysal, A., Rostami, M., Boushine, T., Roshan, R. Gh., Wang, H., & Muralidhar, N. (2025). *DenoMAE2.0: Improving Denoising Masked Autoencoders by Classifying Local Patches* . http://arxiv.org/abs/2502.18202

Guillem Boquet, EMAMJS and JLV (2021). *Theoretical Tuning of the Autoencoder Bottleneck Layer Dimension: A Mutual Information-based Algorithm* . IEEE.

Hartono, P. (2020). *Mixing autoencoders with classifier: conceptual data visualization* . https://doi.org/10.1109/ACCESS.2020.2999155

Judijanto, L. (2025). *ARTIFICIAL INTELLIGENCE* . https://www.researchgate.net/publication/391369694

Kemendikbud, T., Kurniawan, W., Kristian, Y., & Santoso, J. (2021). *J-INTECH (Journal of Information and Technology) Utilization of Deep Convolutional Auto-encoder to Mitigate Adversarial Attacks on Digital Images* .

Lee, W.-H., Ozger, M., Challita, U., & Sung, K. W. (2021). *Noise Learning Based Denoising Autoencoder* . https://doi.org/10.1109/LCOMM.2021.3091800

Li, Z., Liu, Y., Li, J., Ye, G., Chai, H., Lu, Z., & Wu, J. (2024). VAEFL: Integrating variational autoencoders for privacy preservation and performance retention in federated learning. *Security and Safety* , *3* , 2024005. https://doi.org/10.1051/sands/2024005

Maulana, A., Fatichah, C., & Suciati, N. (2020). Facial inpainting in unaligned face images using a generative adversarial network with feature reconstruction loss. *JUTI: Jurnal Ilmiah Teknologi Informasi* , 171–178. https://doi.org/10.12962/j24068535.v18i2.a1004

Raharjo, B. (2022). *PY FAYASAN PRIMA AGUS TEKNIK* .

Sely Wita, D., & Yanti Liliana, D. (2022). Identity Classification with Palm Images Using Convolutional Neural Network (CNN). *JURTI* , *6* (1). https://www.kaggle.com/mahdieizadpanah/birjand-university-

Septia Nugraha, F., & Ferdinandus Pardede, H. (2022). *AUTOENCODER FOR BIRTH WEIGHT PREDICTION SYSTEM* . https://doi.org/10.25126/jtiik.202293868

Zamachsari, F., & Puspitasari, N. (2021a). Application of Deep Learning in Detecting Electronic Financial Transaction Fraud. *RESTI Journal* , *5* (2), 203–212. https://doi.org/10.29207/resti.v5i2.2952

Zamachsari, F., & Puspitasari, N. (2021b). Application of Deep Learning in Detecting Electronic Financial Transaction Fraud. *RESTI Journal* , *5* (2), 203–212. https://doi.org/10.29207/resti.v5i2.2952