

Residual Network Deep Learning Model with Data Augmentation Effects in the Implementation of Iris Recognition

Farradila Ayu Damayanti¹, Rosa Andri Asmara^{1*}, and Gunawan Budi Prasetyo¹

¹*Department of Information Technology, Politeknik Negeri Malang, Malang, Indonesia*

Corresponding Author: Rosa Andri Asmara, rosa.andrie@polinema.ac.id

Received Date: 20-03-2024 Revised Date: 15-05-2024 Accepted Date: 21-06-2024

Abstract

Research in iris recognition using deep learning methods has gained significant traction in recent times, emerging as a robust biometric identification approach. Iris patterns encompass diverse elements such as color, vascular structure, and texture, which combine uniquely and pose challenges for forgery or replication. ResNet, a specialized Deep Learning model designed for object recognition, has demonstrated remarkable performance when trained on the ImageNet dataset. However, iris images pose a considerable challenge for recognition due to their intricate features, which are difficult to discern through conventional observation or content-based feature extraction methods. This study seeks to evaluate the efficacy of ResNet-34 and ResNet-50 in iris recognition. In addition, a data augmentation process will be implemented to accelerate the training process and improve accuracy. It is evident that ResNet-34 achieves a higher accuracy than ResNet-50. Specifically, ResNet-34 produces a test accuracy of 0.751 for the original dataset and 0.768 for the augmented dataset, while ResNet-50 achieves an accuracy of 0.73 for the original dataset and 0.747 for the augmented dataset.

Keywords : CNN, deep learning, residual network, data augmentation, UPOL dataset

1 Introduction

Research in iris recognition using deep learning methods has gained significant traction in recent times, emerging as a robust biometric identification approach. Iris patterns encompass diverse elements such as color, vascular structure, and texture, which combine uniquely and pose challenges for forgery or replication [1]. The efficacy of iris-based identification hinges on the system's capability to accurately capture and analyze iris images under various environmental conditions. Iris recognition systems identify individuals by discerning patterns within one or both of their irises [2].

The Convolutional Neural Network (CNN) stands out as a key architecture in deep learning algorithms, particularly adept at handling image and video data. Its notable strength lies in the autonomous extraction of intricate and layered features from images, eliminating the need for manual feature extraction. Implementing CNN in biometric technology enhances recognition precision and the system's adaptability across diverse conditions. Through extensive training with sizable datasets, CNNs can discern and interpret distinctive patterns, creating dependable and resilient models.

Azam et al. [3] introduced a streamlined method employing CNN and the support vector machine (SVM) for enhanced feature extraction and classification, with the aim of improving recognition efficiency. Daniel et al. [4] presented an iris recognition approach to detect eye irregularities, employing CNN. Their methodology incorporates two algorithms to mitigate dataset biases: an augmentation algorithm employing Gabor filters and a Naiver-Stokes-based algorithm for removing light spots. Meanwhile, using CNN, Jorge et al. [5] presented an iris recognition approach to detect eye irregularities. Their methodology incorporates two algorithms to mitigate dataset biases: an augmentation algorithm employing Gabor filters and a Naiver-Stokes-based algorithm for removing light spots -meanwhile, Jorge et al.

Mingyu et al. [6] proposed using the ResNet-34 architecture in conjunction with transfer learning to detect wood knot defects, employing ResNet-34 as a feature extractor. The ResNet-34 model, which incorporates transfer learning for identification and classification as suggested by Le et al. [7] proves adept at addressing challenges such as limited image availability, poor image quality, and complex feature extraction. Furthermore, ResNet-50 addresses performance concerns and facilitates deeper network structures, thereby helping to maintain model performance [8]. Another experiment carried out in our lab analyzed fingerprint image recognition, comparing the efficacy of Deep Learning ResNet-34 and ResNet-50 on the FVC2000 data set, with ResNet-34 achieving an accuracy close to 100%, significantly surpassing ResNet-50's accuracy of 77.86% [9].

ResNet a specialized Deep Learning model [10] designed for object recognition, has demonstrated remarkable performance when trained on the ImageNet dataset. However, iris images pose a considerable challenge for recognition due to their intricate features, which are difficult to discern through conventional observation or content-based feature extraction methods. This study seeks to evaluate the efficacy of ResNet-34 and ResNet-50 in iris recognition. In addition, a data augmentation process will be implemented to accelerate the training process and improve accuracy.

2 Method

2.1 Dataset

The University of Palackeho and Olomouc (UPOL) data set [11] provided the secondary data comprising the iris images utilized in this study. The database contained iris images acquired through various methods, offering various images suitable for system training purposes. In total, there were 384 original images obtained from 64 individuals, with each person contributing three images captured from the left eye and three from the right eye [12, 13, 14].

Table 1: Dataset

Dataset	Original Data	Augmented Data
Train	256	1024
Test	128	128
Total	384	1216

2.2 Data Augmentation

Data augmentation involves generating additional samples to complement the existing ones in the dataset by modifying current samples. This process aims to improve the accuracy and consistency of the classification results in the data [15]. One of the primary purposes of data augmentation is to prevent over-fitting. Without enhancement or regularization, deep neural networks often learn irrelevant correlations and memorize intricate patterns that are difficult for humans to discern [16]. Various augmentation techniques used in this study include:

2.2.1 Rotation

Rotation involves turning the image or data by a specific angle to introduce more diversity into the training data. Mathematically, rotation can be expressed as follows [17]:

$$x'(R) = \{Rx_1, \dots, Rx_t, \dots, Rx_T\} \quad (1)$$

R represents the rotation matrix applied to rotate the image or data. Employing this rotation method ensures that the model can identify objects even when they are positioned at various angles within the image.

2.2.2 Random Erasing and Random Cropping

Random erasing involves randomly deleting regions within an image as an enhancement technique. In contrast, random cropping involves selecting a small section of the original image and resizing it to match the dimensions of the original image [18].

2.2.3 Random Perspective

Perspective transformation typically involves altering the angle of an image, introducing random fluctuations. The equation for random perspective transformation is expressed as follows [19]:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (2)$$

2.2.4 Gaussian Blur

Gaussian blur adjusts the pixel weight in an image based on the Gaussian distribution function, incorporating the average value with neighboring pixels. The following equation represents the formula for Gaussian blur [20]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

Where σ represents the standard deviation, while x and y denote the distances from the origin along the horizontal and vertical axes. These values from the distribution are utilized to generate a convolution matrix, which is then employed on the initial image. This process blurs the original image and softens the pixels surrounding the edges of the object. Consequently, it facilitates edge detection for object recognition algorithms and enhances the precision of bounding box presentations [21].

2.2.5 Horizontal Flip

Horizontal flipping involves randomly flipping the input image along the vertical axis (from left to right) with a certain probability. The formula for horizontal flipping is elucidated as follows [22]:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

$$x' = -x, \quad y' = y \quad (5)$$

2.2.6 Vertical Flip

Vertical flipping entails randomly flipping the input image along the horizontal axis (from top to bottom) with a certain probability. The formula for vertical flipping is elucidated as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

$$x' = x, \quad y' = -y \quad (7)$$

2.2.7 Residual Network

Residual Network (ResNet) is among the models introduced in 2016 [23], specifically engineered to address the challenges encountered during deep learning training, which often requires lengthy durations and is limited by a maximum number of layers. ResNet is advantageous over alternative architectural models in that its performance remains consistent despite its increased depth. In addition, it simplifies computations and improves network training capabilities [24].

In this research, the proposed approach is depicted in Figure 1. The block diagram illustrates that both the training and testing phases follow identical procedures throughout. In the training phase, features are stored in the knowledge base, while during the testing phase, extracted features are compared with those from the training phase using a classifier.

Table 2: ResNet Architecture [25]

Layer Name	Output Size	18-Layer	34-Layer	50-Layer	101-Layer	152-Layer
conv1	112x112	7x7, 64, stride 2				
		3x3 max pool, stride 2				
conv2_x	56x56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	Average pool, 1000-d fc, softmax				
FLOPs		1.8x10 ⁹	3.6x10 ⁹	3.8x10 ⁹	7.6x10 ⁹	11.3x10 ⁹

3 Results and discussion

This research conducted experiments with 80% of the data allocated for training and 20% for testing. Pre-processing was initially performed on RGB iris images, followed by using various techniques to augment training data samples. Figure 2 illustrates the results of the augmentation process in the training data. The data augmentation techniques used in this study are Resize, Random Crop, Random Rotation, Random Perspective, Random Horizontal and Vertical, and Gaussian Blur. These techniques are used not only to increase the dataset on the training data but also to improve the performance of the model by training the model on more diverse data so that the model can handle variations that occur in the data, such as changes in scale, rotation, size, image sharpness. In addition to reducing overfitting, the model recognizes the training data and learns to generalize on data that has never been studied before.

After going through the data augmentation stage, the next step is to apply learning with the ResNet model. The layers in the ResNet architecture have their requirements; generally, feature learning models use convolutional layers. In addition to the convolutional layer, there are batch normalization, activation, and pooling layers in the ResNet architecture. The description of the preparation of ResNet-34 and ResNet-50 can be seen in Table 2.

Figure 3 illustrates the accuracy contrast among training, testing, and validation data for two architectures utilizing both original and augmented data. Figure 3(a) shows that the ResNet-34 Aug and ResNet-50 Aug architecture models exhibit higher and more consistent accuracy values compared to ResNet-34 and ResNet-50 with original data, particularly noticeable at epoch 60. Notably, at epoch 100, the training accuracy for ResNet-34 is 0.9995, and for ResNet-50 is 0.9992, whereas, for ResNet-34 Aug and ResNet-50 Aug, it stands at 1 and 0.9991, respectively. The training accuracy outcomes for ResNet-34 and ResNet-34 Aug appear superior to those of ResNet-50 and ResNet-50 Aug. In Figure 3(b), ResNet-50 exhibits higher test accuracy than ResNet-34, with values at epoch 100 of 0.760 and 0.751, respectively. However, ResNet-34 Aug achieves a higher accuracy value of 0.768 at epoch 100 than ResNet-50 Aug, which records 0.666 at the same epoch. This shows that ResNet-34 is better at generalizing on varied augmentation data. Figure 3(c) demonstrates the validation accuracy results, all indicating a value of 1, with ResNet-34 at epoch 70, ResNet-50 at epoch 77, ResNet-34 Aug at epoch 92, and ResNet-50 Aug at epoch 89.

Table 3: Hasil Model pada Dataset Asli dan Dataset Augmentasi

Model	Train	Test	Val
Original Dataset			
ResNet-34	0.99951	0.751	1
ResNet-50	0.98921	0.730	1
Augmented Dataset			
ResNet-34	1	0.768	1
ResNet-50	0.991	0.747	1

The results of ResNet-34 augmentation show that data augmentation provides more significant benefits for simpler models with fewer parameters. The model can utilize the additional image variations from the

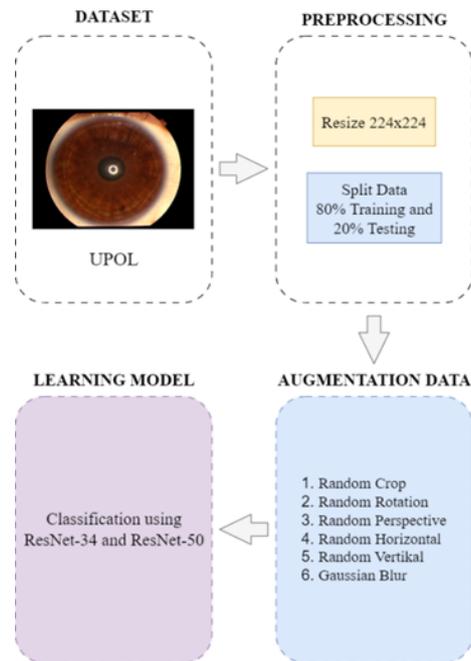


Figure 1: The block diagram

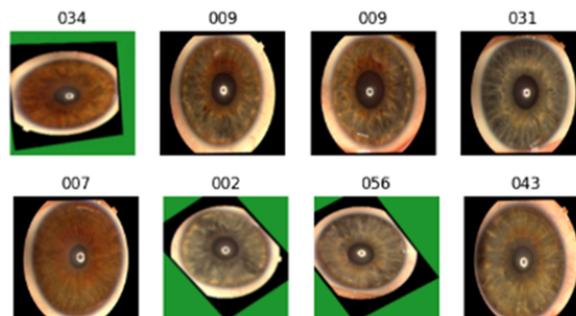


Figure 2: Image samples of Data Augmentation

augmentation more efficiently. Although ResNet-50 has improved the performance of the model with data enhancement, the results cannot still outperform ResNet-34. This is because the complexity of the model is higher, making it more difficult for ResNet-50 to generalize effectively on the augmented dataset.

ResNet-34's superiority in this study is due to its efficient handling of both the original and augmented datasets. The smaller and simpler model size of ResNet-34 allows it to avoid overfitting and generalize patterns on the dataset effectively. In contrast, the larger and more complex ResNet-50 struggles to fully exploit the advantages of a deeper layer architecture, especially when dealing with a relatively small dataset, highlighting the challenges it faces. Further comparison details of the two architectures using original and augmented data are presented in Table 3.

The test results reveal that the training duration for ResNet-34 and ResNet-34 Augmentation is shorter compared to ResNet-50 and ResNet-50 Augmentation. Specifically, training ResNet-34 with the original data set requires 909 seconds, while using the augmented data set takes 924 seconds. In contrast, training ResNet-50 with the original dataset requires 3303 seconds, while utilizing the augmented dataset extends the training time to 4004 seconds. This discrepancy is attributable to the greater complexity of ResNet-50, which features more convolution layers than ResNet-34. The increased number of layers in the architecture can prolong the duration of training and testing.

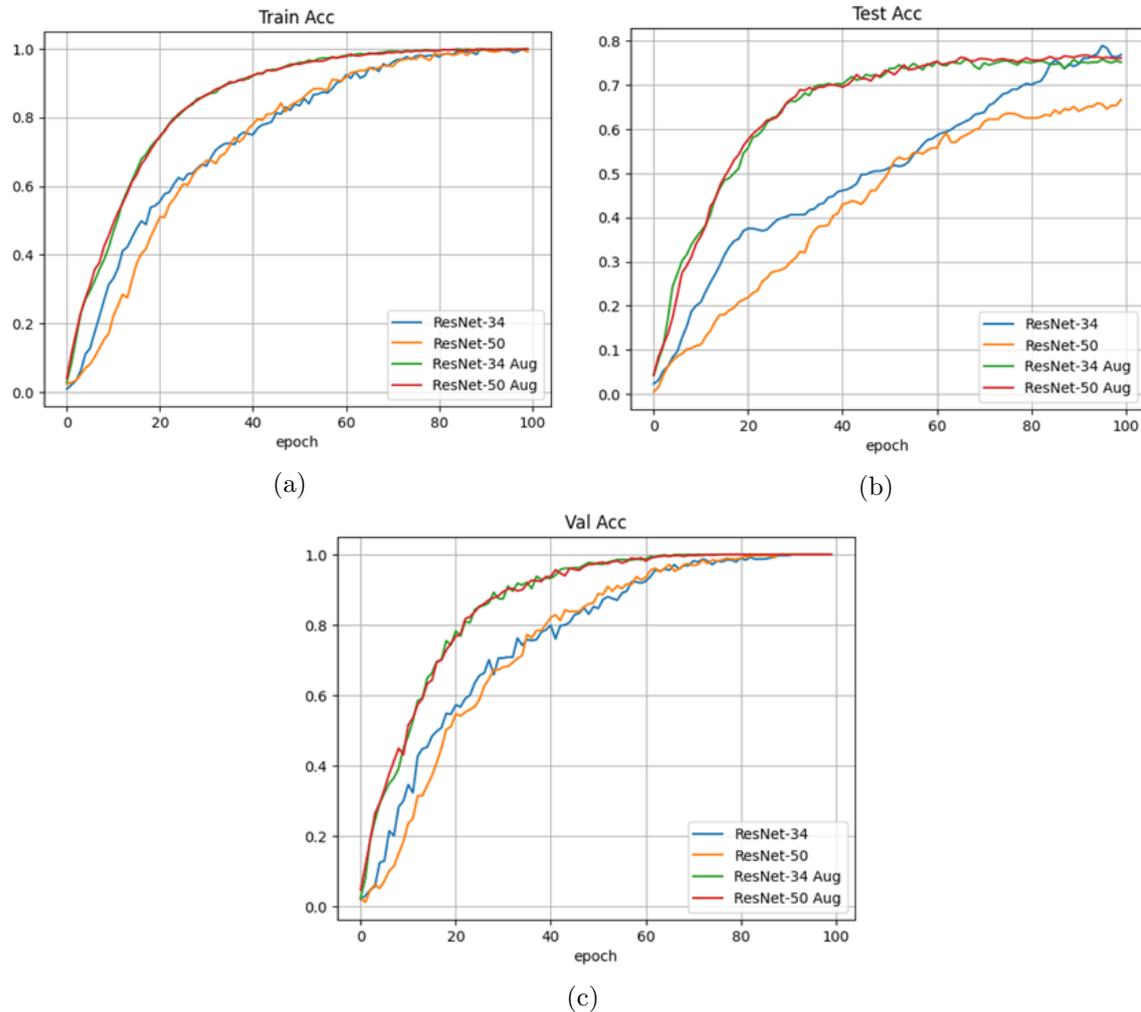


Figure 3: (a) Train Accuracy; (b) Test Accuracy; and (c) Validation Accuracy

4 Conclusion

According to the findings of the use of the CNN architecture for iris recognition utilizing two ResNet models, namely ResNet-34 and ResNet-50, it is evident that ResNet-34 achieves higher accuracy than ResNet-50. Specifically, ResNet-34 produces a test accuracy of 0.751 for the original dataset and 0.768 for the augmented dataset, while ResNet-50 achieves an accuracy of 0.760 for the original dataset and 0.666 for the augmented dataset. Based on the comparison of the computation time obtained by ResNet-34 on the original and augmented datasets (909 and 3303 seconds), it is more efficient compared to ResNet-50 (924 and 4004 seconds), which takes longer because more layers and parameters need to be calculated, coupled with data augmentation because each data transformation extends the overall training process. Our analysis shows that ResNet-34 adeptly captures the dataset's structure or pattern, potentially generating a superior feature representation compared to ResNet-50. In contrast, ResNet-50 encompasses numerous parameters, making it more susceptible to overfitting. ResNet-50 has about 25.6 million parameters, and ResNet-34 has about 21.8 million. This shows that ResNet-50's parameter overhead is large and may not be fully utilized on simpler datasets. If the data set is not large enough or complex enough, adding more parameters to ResNet-50 will not provide much benefit and may even become a training bottleneck. In this case, models with fewer parameters, such as ResNet-34, can learn important patterns more efficiently without further complicating the training process.

Acknowledgments

Author would like to thank Politeknik Negeri Malang for Master Program Research Grant 2024.

References

- [1] H. M. Therar, L. D. E. A. Mohammed, and A. P. D. A. J. Ali, "Multibiometric system for iris recognition based convolutional neural network and transfer learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1105, no. 1, p. 012032, jun 2021. [Online]. Available: <https://dx.doi.org/10.1088/1757-899X/1105/1/012032>
- [2] S. Punnoose and J. S. J. Kumar, "Iris recognition for security & safety of automobiles," *International Journal of Innovative Science, Engineering and Technology*, vol. 2, no. 4, pp. 961–966, 2015.
- [3] M. S. Azam and H. K. Rana, "Iris recognition using convolutional neural network," *International Journal of Computer Applications*, vol. 175, no. 12, pp. 24–28, 2020.
- [4] D. F. Santos-Bustos, B. M. Nguyen, and H. E. Espitia, "Towards automated eye cancer classification via vgg and resnet networks using transfer learning," *Engineering Science and Technology, an International Journal*, vol. 35, p. 101214, 2022.
- [5] J. E. Zambrano, D. P. Benalcazar, C. A. Perez, and K. W. Bowyer, "Iris recognition using low-level cnn layers without training and single matching," *IEEE Access*, vol. 10, pp. 41 276–41 286, 2022.
- [6] M. Gao, J. Chen, H. Mu, and D. Qi, "A transfer residual neural network based on resnet-34 for detection of wood knot defects," *Forests*, vol. 12, no. 2, pp. 1–16, 2021.
- [7] L. Gao, X. Zhang, T. Yang, B. Wang, and J. Li, "The application of resnet-34 model integrating transfer learning in the recognition and classification of overseas chinese frescoes," *Electronics*, vol. 12, no. 17, 2023.
- [8] Z. Zahisham, C. P. Lee, and K. M. Lim, "Food recognition with resnet-50," in *2020 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAJET)*, 2020, pp. 1–5.
- [9] R. A. Asmara, C. Rahmad, J. Rahimatullah, and A. N. Handayani, "Analysis of fingerprint image recognition using deep residual convolutional neural network," in *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 2024, pp. 422–427.
- [10] S. A. Hasanah, A. A. Pravitasari, A. S. Abdullah, I. N. Yulita, and M. H. Asnawi, "A deep learning review of resnet architecture for lung disease identification in cxr image," *Applied Sciences*, vol. 13, no. 24, 2023.
- [11] "Palacký University Olomouc (UPOL) Iris Image Dataset," <https://phoenix.inf.upol.cz/iris/>, accessed: 2023-08-23.
- [12] S. Shirke, R. Sonkamble, G. Machhale, and S. Makubhai, "Identification of a person using a different database by iris," *European Chemical Bulletin*, vol. 12, no. 8, pp. 4155–4166, 2023.
- [13] L. Omelina, J. Goga, J. Pavlovicova, M. Oravec, and B. Jansen, "A survey of iris datasets," *Image and Vision Computing*, vol. 108, p. 104109, 2021.
- [14] H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L. A. Alexandre, "The ubiris.v2: A database of visible wavelength iris images captured on-the-move and at-a-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1529–1535, 2010.
- [15] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, 2020.
- [16] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of Big Data*, vol. 8, no. 101, pp. 1–34, 2021.
- [17] G. Iglesias, E. Talavera, González-Prieto, A. Mozo, and S. Gómez-Canaval, "Data augmentation techniques in time series domain: a survey and taxonomy," *Neural Computing and Applications*, vol. 35, no. 14, pp. 10 123–10 145, 2023.
- [18] E. Goceri, "Medical image data augmentation: techniques, comparisons and interpretations," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12 561–12 605, 2023.
- [19] H. Huang, H. Zhou, X. Yang, L. Zhang, L. Qi, and A. Y. Zang, "Faster r-cnn for marine organisms detection and recognition using data augmentation," *Neurocomputing*, vol. 337, pp. 372–384, 2019.

- [20] C. Song, W. Xu, Z. Wang, S. Yu, P. Zeng, and Z. Ju, "Analysis on the impact of data augmentation on target recognition for uav-based transmission line inspection," *Complexity*, vol. 2020, 2020.
- [21] M. R. Prusty, V. Tripathi, and A. Dubey, "A novel data augmentation approach for mask detection using deep transfer learning," *Intelligent Medicine*, vol. 5, p. 100037, 2021.
- [22] R. Hao, K. Namdar, L. Liu, M. A. Haider, and F. Khalvati, "A comprehensive study of data augmentation strategies for prostate cancer detection in diffusion-weighted mri using convolutional neural networks," *Journal of Digital Imaging*, vol. 34, no. 4, pp. 862–876, 2021.
- [23] D. McNeely-White, J. R. Beveridge, and B. A. Draper, "Inception and resnet features are (almost) equivalent," *Cognitive Systems Research*, vol. 59, pp. 312–318, 2020.
- [24] D. Sarwinda, R. Hilya, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer," *Procedia Computer Science*, vol. 179, no. 2019, pp. 423–431, 2021.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.