Journal of Information Technology and Cyber Security 1(2) July 2023: 68-75 EISSN 2987-386X | PISSN 2987-3878
DOI: 10.30996/jitcs.8716

Diabetic Retinopathy Blood Vessel Detection Using CNN and RNN Techniques

Adithya Kusuma Whardana [1,*], Parma Hadi Rentelinggi [2], and Hezkiel Dokta Timothy [3]

[1,3] Department of Informatics Engineering, Universitas Tanri Abeng, Indonesia
[2] Department of Information and Computer Science, Keio University, Japan
* Corresponding author: adithya@tau.ac.id

Received: 12 June 2023
Accepted: 10 January 2024
Revised: 1 January 2024
Available online: 17 January 2024

To cite this article: Whardana, A. K., Rentelinggi, P. H., & Timothy, H. D. (2023). Diabetic Retinopathy Blood Vessel Detection Using CNN and RNN Techniques. *Journal of Information Technology and Cyber Security, 1*(2), 68–75. https://doi.org/10.30996/jitcs.8716

Abstract
This research aims to detect diabetic retinopathy using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). The main objective is to compare these two methods in detecting the condition. Based on the study's result after training 10 times on each method, the accuracy results were 92% for the CNN method and 50% for the RNN method. These results show, this study with the dataset used, the CNN method is much more effective in detecting diabetic retinopathy than the RNN method. The CNN method is better due to its ability to extract spatial features from images, which is important in image classification tasks.
Keywords: Blood vessel detection, Convolutional Neural Network, deep learning, diabetic retinopathy, medical image analysis, Recurrent Neural Network.

1. Introduction
Diabetic retinopathy (DR) is a disorder that affects the tiny blood vessels in the retina of both type 1 and people with type 2 diabetes (Yang et al., 2022). One of the leading causes of blindness worldwide, DR results in abnormalities of the retina (Alyoubi et al., 2020; Asia et al., 2022; Reguant et al., 2021). Approximately one-third of diabetic people get DR, and almost everyone will at some point. An estimated 191 million individuals could have DR by 2030. Since DR cannot be cured, current therapies mostly aim to control eyesight. Early detection and treatment significantly reduce the risk of eyesight loss. Compared to computer-assisted diagnostic systems, manual diagnosis of DR by retinal fundus image evaluation by ophthalmologists is time-consuming and error-prone (Alyoubi et al., 2020).

This study explores the detection of diabetic retinopathy (DR) by observing various indications in the eye's retinal fundus. Specifically, it focuses on identifying blood vessel swelling, a significant indicator of DR. Utilizing the CNN method. The study aims to enhance images of the swollen areas, leveraging CNN's image processing capabilities well-suited for analyzing retinal fundus images (Bamber & Vishvakarma, 2023). CNNs excel at automatically learning and extracting pertinent features from structures like blood vessels and optic discs, enabling precise detection of abnormalities associated with DR.

Additionally, the study considers the RNN method as another technique for DR detection due to its proficiency in capturing temporal dependencies. RNNs are adept at analyzing evolving data, such as dynamic retinal images or sequential measurements obtained from patients during follow-up visits. This temporal aspect could be valuable in tracking changes or progression of DR over time in a patient's eye health (Haldar, 2019).

In this study, we aim to compare two methods for detecting diabetic retinopathy (DR), providing readers with insights to determine the most effective approach. We'll utilize Python programming language to implement both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) using retinal fundus photos. The dataset comprises 149 samples classified into three categories: train, validate, and test. Each category contains two data folders—'DR' and No DR.' These folders contain retinal fundus

images that either indicate the presence of DR (in the 'DR' folder) or do not show indications of DR (in the No DR folder). For both CNN and RNN methods, the data will undergo ten training iterations to ensure robustness. During training, the models will learn to distinguish between retinal fundus images showing signs of DR, particularly focusing on identifying swollen blood vessels, and those without such indications.

The performance metrics of the CNN and RNN techniques, including accuracy, sensitivity, specificity, and other pertinent metrics, will be evaluated to compare them. Through a comparative study, we will determine which technique, given the dataset and the particular features of retinal fundus images linked to the disease, shows higher efficacy in detecting DR.

2. Methodology

2.1. Convolutional Neural Network

Convolutional Neural Networks (CNN) are a particular form of the neural network specifically designed to process and analyze visual data, images. They have evolved from the Multilayer Perceptron (MLPs) concept but are structured differently to effectively handle 2D data, such as images (Putra et al., 2016). CNNs in image processing tasks due to their ability to retain spatial information in images. Unlike MLPs, which treat each pixel as an independent feature and struggle to maintain spatial relationships, CNNs use a convolution layer that captures local patterns through a sliding filter, preserving the spatial hierarchy and enabling effective feature extraction from images. The design of this research model is shown in Fig. 1.

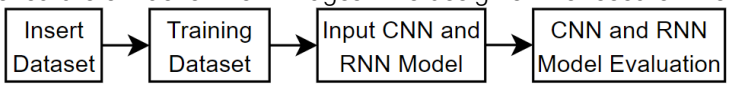

Fig. 1. Flowchart of proposed Method.

Flowcharts are handy for visualizing the sequence of processes in this research. Fig. 1 is a flow chart outlining the research steps: 1) Inserting the Dataset: The initial step involves inserting the dataset. This includes preparing the data, which may involve preprocessing and dividing it into training and validation/testing sets. 2) Training the Dataset: This stage involves putting the prepared dataset into the training process. During training, the models (CNN and RNN) learn from the dataset by adjusting their parameters to minimize errors and improve performance. 3) Inputting the CNN and RNN Models: Once the training is complete, the trained CNN and RNN models are evaluated. These models have learned patterns and features from the dataset during the training phase. 4) Model Evaluation: This step evaluates the performance of the CNN and RNN models. This may involve using separate validation or test datasets (not used during training) to assess how well each model generalizes to new unseen data. The evaluation process may include accuracy, precision, recall, or other metrics for detecting diabetic retinopathy.

CNN layers use filters or kernels to selectively capture features from the input image, such as edges, textures, and forms. CNN can automatically extract pertinent information from images to these layers, pooling and fully linking layers. This makes CNN ideal for object detection, segmentation, and image classification tasks. Due to their specific architecture, which considers the intrinsic spatial organization present in images, CNN has outperformed MLP in image-related tasks, resulting in more accurate and efficient image analysis and classification (Putra et al., 2016).

Kunihiko Fukushima indeed developed an early version of the Convolutional Neural Network (CNN) known as the Neo Cognitron during his time at NHK Broadcasting Science Research Laboratories in Tokyo, Japan. His work laid the foundational concepts for CNNs by focusing on hierarchical pattern recognition inspired by the human visual system (Fukushima, 1980; Putra et al., 2016). However, it was Alex Krizhevsky's implementation of the CNN, particularly the deep architecture known as AlexNet, that gained significant attention and acclaim. AlexNet's triumph in the ImageNet Large Scale Visual Recognition Challenge in 2012 demonstrated the effectiveness of CNNs in image classification tasks and signaled a significant turning point for deep learning. Deep CNNs' dominance in the ImageNet competition served as a spur for the broad acceptance and investigation of deep learning techniques in various fields. Deep neural networks can solve complex issues in computer vision, natural language processing, and other fields. Researchers and practitioners have recognized this.

It is true that the CNN approach, which is distinguished by its capacity to automatically extract features from unprocessed pixel data, has proven remarkably successful in object classification inside images. It has firmly established itself as a top technique in computer vision thanks to its capacity to extract hierarchical and spatial characteristics from images. It has outperformed many traditional machine learning approaches in numerous image-related tasks, such as Support Vector Machines (SVM). This triumph demonstrated deep CNNs' better performance over conventional machine learning techniques and greatly

aided the widespread acceptance of deep learning techniques (Gunawardena et al., 2017). When it comes to identifying objects in photos, the CNN approach has shown to be incredibly successful—it even outperforms other machine learning techniques like SVM (Putra et al., 2016).

2.2. MultiLayer Perceptron

Understanding the essential physics of a Multi-Layer Perceptron (MLP) can be greatly enhanced by creating one from scratch using NumPy. An MLP comprises an input layer, one or more hidden layers, and an output layer. Multiple neurons, or perceptrons, are connected to the next layer in each layer.

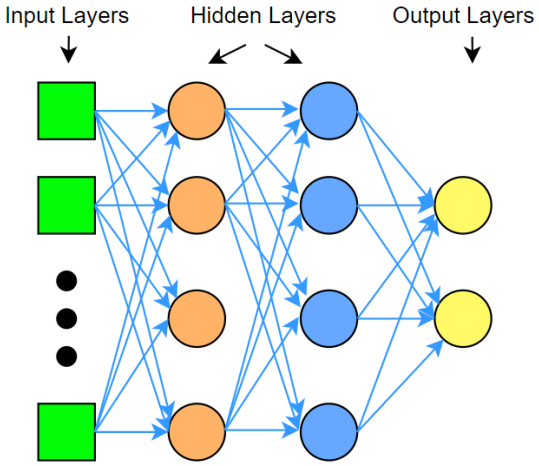

Fig. 2. Representation of multilayer perceptron.

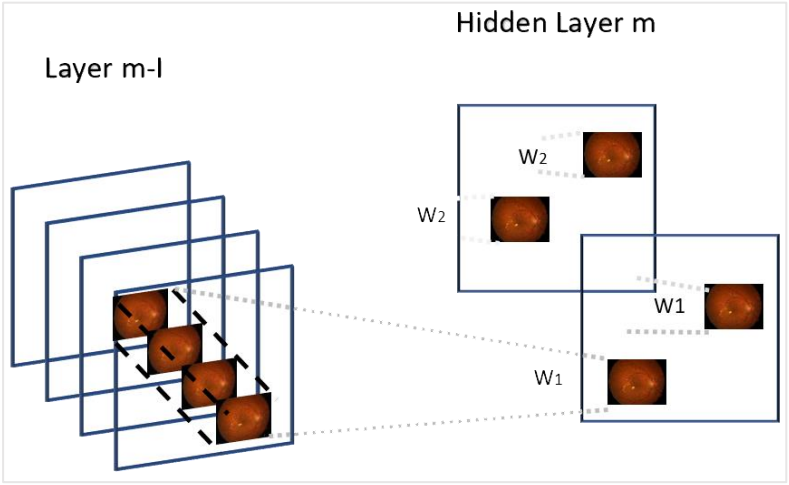

Fig. 3. Convolution process.

An input layer, one or more hidden layers, and an output layer make up the Multi-Layer Perceptron (MLP), a feedforward artificial neural network, as shown in Fig. 2. Each layer comprises networked nodes, or neurons, that use weighted connections to transfer signals from the input to the output layer. The fundamental algorithm used to train JSTs is backpropagation. For a given set of input data, it entails modifying the weights of the network's connections to reduce the discrepancy between the expected output and what is produced. This is accomplished by repeatedly propagating the error backward through the network and utilizing gradient descent and other approaches to update the weights accordingly.

MLP is considered a form of deep learning due to its capacity to learn and model complex patterns in data through multiple layers and non-linear activation functions. Learning hierarchical representations of data allows MLPs to handle various tasks, including classification, regression, and pattern recognition. The backpropagation algorithm allows MLPs to adjust their internal parameters during the training phase, allowing them to generalize and make accurate predictions on unseen data. The weights in CNNs are shared across the input, allowing the network to detect the same feature regardless of its location in the input image. This shared weight scheme significantly reduces the number of parameters compared to fully connected layers in MLPs, making CNNs more efficient for processing large-scale data like images. Furthermore, CNNs often include additional layers like pooling layers (to reduce spatial dimensions) and fully connected layers at the end for classification tasks, combining the extracted features into a final

prediction. The structure and operation of CNNs are specifically tailored to handle 2D data, making them highly effective for tasks like image recognition, object detection, and other applications involving spatially structured data (Zhang et al., 2018).

There is a fundamental difference between Convolutional Neural Networks (CNN) and Multi-Layer Perceptions (MLP). Every neuron in a layer is connected to every neuron in the next layer. The weights in MLPs are represented as a matrix for each layer, usually a 2D matrix for the connections between layers (such as between input and hidden layers and between hidden layers and output layers). Fig. 3 shows the Convolution process in CNN. CNN utilizes convolution operation. The main difference lies in the division of weights and the concept of filters/kernels. Full connectivity: CNN exploits the local connectivity patterns present in the image.

Weight Representation in CNN: weights are represented as a 4D array (tensor). This tensor consists of the following dimensions, Height and Width: Corresponds to the spatial coverage of the filter. Depth/Channels: Represents the depth of the input volume or the number of input channels. Numberof Filters: Indicates the number of filters/kernels in the layer. These 4D weights are convolved with the input volume (which also has multiple channels) to produce a feature map. The convolution operation, coupled with weight sharing and the use of multiple filters, allows CNN to learn the hierarchical representation of features in an image (Reguant et al., 2021).

CNN significantly reduces the number of parameters compared to fully connected layers in an MLP, making CNN more suitable for processing high-dimensional inputs such as image 2 while maintaining spatial relationships in the data (Jafari & Karami, 2023). The dimensions of the weights in CNN are structured as on Eq. (1).

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega ab y_{(i+a)(j+b)}^{l-1} \quad (1)$$

Assume that it possesses some $N \times N$ our convolutional layer comes after the square neuron layer. Using an $m \times m$ filter ω, the output of our convolutional layer will be size $(N - m + 1) \times (N - m + 1)$. To calculate the pre-nonlinearity input for a given unit x_{ij}^{l} in the layer, one must total the contributions from the preceding layer cells, weighted by the filter components.

To compute the pre-nonlinearity input for multiple units x_{ij}^{l} in the convolutional layer:

1) Each unit x_{ij}^{l} in the convolutional layer will be associated with a different set of indices (i, j) in the output of the convolutional layer.
2) For each unit x_{ij}^{l}, the process involves summing up the contributions from the cells in the previous layers, weighted by the filter components.
3) Since the convolutional layer is derived from multiple layers of $N \times N$ square neurons, the contributions from these multiple layers need to be considered for the computation.
4) For each unit x_{ij}^{l}, the weighted sum of contributions from the cells in the previous layers ($N \times N$ square neurons) is calculated by applying the filter ω to the corresponding spatial regions of these multiple layers.
5) The filter ω is convolved with the respective regions in the preceding layers, and the element-wise multiplication of the filter components and the neuron values in these regions is performed. These products are then summed up to compute the pre-nonlinearity input for the unit x_{ij}^{l}.

CNNs are designed to utilize the spatial correlations seen in 2D data, such as spectrograms (representations of sound) and images. The convolution function of CNN is built to function well with structured data. Convolution operates very well with 2D data due to its intrinsic assumptions about spatial locality and hierarchy, especially in terms of the technique of sliding window filters over the input data. Convolution layers use spatially applied weights, or kernels, to capture elements in image and sound representations such as edges, textures, shapes, and more intricate patterns.

2.3. MultiLayer Perceptron

Recurrent neural networks (RNN) have a lot of promise for processing sequential data for various applications, including sentiment analysis (Nasser & Yusof, 2023). This is especially true for Long Short-Term Memory (LSTM) models. This feature works with various data kinds, including text, audio, and video. In Araque's 2017 work, the LSTM model was especially used to examine sentiment inside a set of Spanish-language tweets.

The study examined word embedding and sentiment lexicon values as two different methodologies. The study found that combining and utilizing these variables improved the overall performance of sentiment

analysis. This suggests that integrating several data or characteristics might frequently result in better sentiment categorization task outcomes.

Models based on LSTMs for sentiment analysis and related applications. RNN, particularly LSTM, are useful for sentiment analysis in texts and other sequential data types because of their capacity to retain information over sequences, which makes them ideal for identifying contextual dependencies and patterns within sequential data (Balaji et al., 2023). The efficacy of LSTM-based RNN in sentiment analysis is demonstrated by this research, which also illustrates how integrating several characteristics or methodologies can significantly improve task performance in sentiment classification, resulting in more precise and nuanced assessments of textual data, such as tweets (Nguyen et al., 2020).

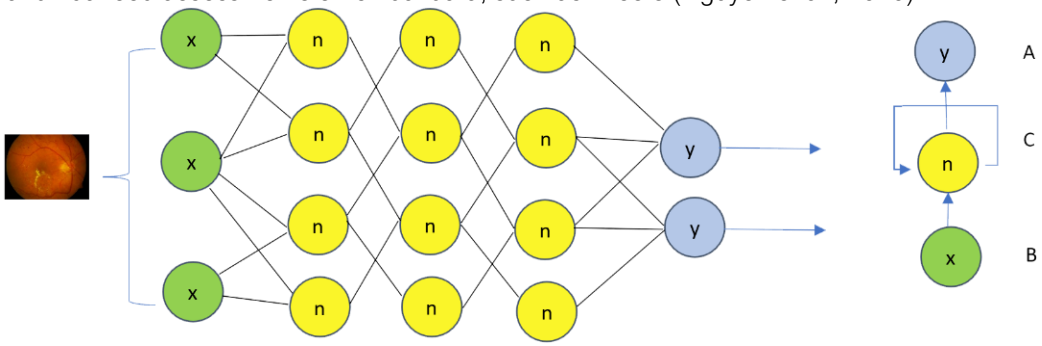

Fig. 4. Recurrent neural network process.

Table 1
Training dataset.

Test	Epoch	Verbose	Loss	Accuracy	Val. Loss	Val. Accuracy
1	10	2	0.64	0.6	0.46	0.8
2	10	2	0.42	0.8	0.53	0.8
3	10	2	0.63	0.7	0.34	0.87
4	10	2	0.32	0.87	0.26	0.95
5	10	2	0.29	0.95	0.35	0.92
6	10	2	0.19	1.0	0.29	0.92
7	10	2	0.28	0.92	0.29	0.92
8	10	2	0.25	0.9	0.25	0.85
9	10	2	0.25	0.87	0.14	0.97
10	10	2	0.27	0.85	0.17	0.9

Two different methodologies: word embedding and sentiment lexicon values. The results showed that combining these two characteristics improved sentiment analysis's performance (Hayum et al., 2023). Similarly, Hassan carried out the study in 2017 on the IMDB website, analyzing sentiment using the LSTM approach. A word2vec-based LSTM model was used in Hassan's study, which produced sentiment analysis results that were more accurate (Cahyadi et al., 2020).

Fig. 4 illustrates that recurrent neural networks have signals that move in both directions using feedback loops in the network. Features derived from previous inputs are returned to the network, allowing it to be remembered. This interactive network is dynamic as its state changes until it reaches equilibrium. These networks are commonly used in autocorrelative sequence data such as time series (Haldar, 2019).

Training a CNN involves feeding data (such as images) multiple times (epochs) to learn patterns and features. Once trained, it can use the model to predict or classify new data. DR can refer to various things usually associated with Deep Learning, which may mean Dropout Regularization, a technique used during training to prevent overfitting by randomly dropping neurons at each epoch. The accuracy of CNN methods on models and datasets can be assessed using test or validation data not used during training. This makes it possible to see how well the model generalizes to unseen data (Altwijri et al., 2023). The accuracy of the CNN model after training, usually, Train the Model: Use your dataset to train the CNN. Monitor loss and accuracy metrics during training to ensure the model learns correctly. Validation: Using a separate validation dataset to evaluate the model's performance after training. Calculate accuracy by comparing the model's predictions with the actual labels in the validation set. Output Accuracy: The program will output the CNN accuracy on the validation dataset, indicating its performance on new data it has never seen.

3. Results and Discussion

The dataset that has been collected will be entered into the program data frame and divided into 3 training data, namely test, train, and valid. Each training data folder will be divided based on 2 labels: DR

and no DR. The DR label contains a collection of retinal fundus photo objects of eyes infected with DR disease with swollen blood vessels. The dr data object is implemented with 89 photo datasets. Then the nodr label contains a collection of eye retina fundus photo objects that are healthy or not infected with DR disease. The No DR data object is implemented with 60 photo datasets. After the dataset is labeled into 2 classes, namely dr and nodr. Then, the next process is scanning using the vgg16 model. The vgg16 model stands for "Visual Geometry Group 16". After that, training is carried out on the model by conducting 10 experiments, as shown in Fig. 5. Training a CNN involves feeding data (such as images) multiple times (epochs) to learn patterns and features. Once trained, it can use the model to predict or classify new data. DR can refer to various things usually associated with Deep Learning, where it may mean Dropout Regularization, a technique used during training to prevent overfitting by randomly dropping neurons at each epoch. The accuracy of CNN methods on models and datasets can be assessed using test or validation data not used during training. This makes it possible to see how well the model generalizes to unseen data. The accuracy of the CNN model after training, usually, Train the Model: Use your dataset to train the CNN. Monitor loss and accuracy metrics during training to ensure the model learns correctly.

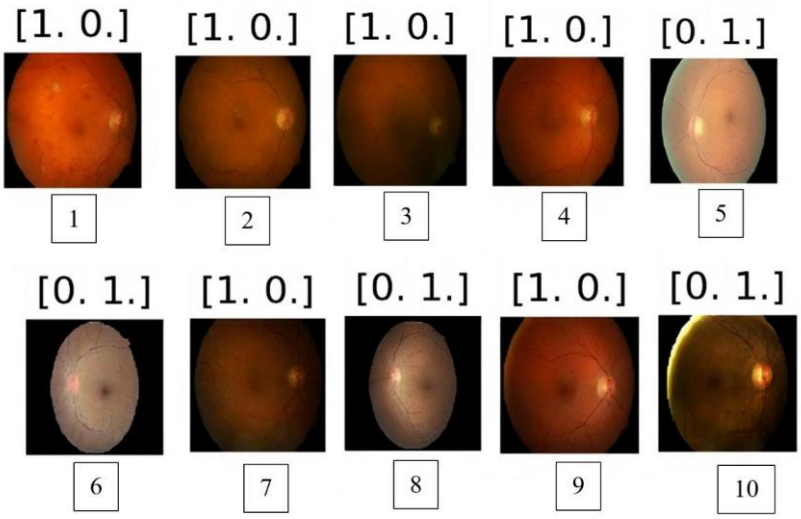

Fig 5. Train result image.

Table 2
Comparison of method accuracy.

Method	Amount of Dataset	Amount of Training	Accuracy (%)
CNN	149	10	92
RNN	149	10	50

Table 1 shows the plot that occurs when the training process takes place, which results in the data table above by doing 10 trials. This figure also shows an indication of DR in the training image. After the training is complete, the program will output the level of accuracy of the CNN method on the model and dataset entered. The model improves over time with reduced loss and accuracy, tending to increase for both the training and validation datasets. However, it is important to watch out for signs of overfitting, where the model performs better on training data but worse on unseen data (validation/testing set). Epoch indicates the epoch number; Loss represents the loss (often a measure of error) calculated during training. Accuracy represents the accuracy achieved on the training dataset for that epoch. Val_Loss denotes the loss calculated on the validation dataset. Val_Accuracy represents the accuracy achieved on the validation dataset for that epoch. In Table 2 is explained that the level of accuracy using the CNN method in clarifying the 2 classes of image objects dr and nodr is 92%, and RNN results is 50%.

4. Conclusions

Research on blood vessel detection using CNN and RNN yields quite different findings; CNN achieves 92% accuracy in blood vessel detection, whereas RNN only reaches 50%. Because its structure consists of cells (units) connected in the form of a chain or sequence, RNN needs to produce better results. Each cell receives input from the previous step and provides output to the next step. RNN uses shared parameters for each time step, meaning the same parameters are used at each sequence step. This allows the model to retain knowledge of the same learning so that when testing data on blood vessels in a specific area, such

as in the optical disk area, the area is considered not to be a blood vessel. RNN. RNNs can suffer from long-time detection problems, where the gradient calculated during training can become very small, causing difficulties in transferring information over long distances.

5. CRediT Authorship Contribution Statement

Adithya Kusuma Whardana: Conceptualization, Data Curation, Investigation, Methodology, Resources, Software, Supervision, and Visualization. **Parma Hadi Rentelinggi**: Data Curation, Formal Analysis, Methodology, and Supervision. **Hezkiel Dokta Timothy**: Methodology, Project administration, Resources, Writing – original draft, and Writing – review & editing.

6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments and suggestions.

8. Data Availability

This study uses an open-source diabetic retinopathy dataset that can be accessed via https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-2015-data-colored-resized, an open-source online data repository hosted at Kaggle (www.kaggle.com).

9. Funding

No funding was received for this study.

10. Ethical Approval

Ethical approval No patient-identifying parts in this paper were used or known to the authors. Therefore, no ethical approval was requested.

11. References

Altwijri, O., Alanazi, R., Aleid, A., Alhussaini, K., Aloqalaa, Z., Almijalli, M., & Saad, A. (2023). Novel Deep-Learning Approach for Automatic Diagnosis of Alzheimer's Disease from MRI. *Applied Sciences*, *13*(24), 13051. https://doi.org/10.3390/app132413051

Alyoubi, W. L., Shalash, W. M., & Abulkhair, M. F. (2020). Diabetic retinopathy detection through deep learning techniques: A review. *Informatics in Medicine Unlocked*, *20*, 100377. https://doi.org/10.1016/j.imu.2020.100377

Asia, A.-O., Zhu, C.-Z., Althubiti, S. A., Al-Alimi, D., Xiao, Y.-L., Ouyang, P.-B., & Al-Qaness, M. A. A. (2022). Detection of Diabetic Retinopathy in Retinal Fundus Images Using CNN Classification Models. *Electronics*, *11*(17), 2740. https://doi.org/https://doi.org/10.3390/electronics11172740

Balaji, P., Chaurasia, M. A., Bilfaqih, S. M., Muniasamy, A., & Alsid, L. E. G. (2023). Hybridized Deep Learning Approach for Detecting Alzheimer's Disease. *Biomedicines*, *11*(1), 149. https://doi.org/10.3390/biomedicines11010149

Bamber, S. S., & Vishvakarma, T. (2023). Medical image classification for Alzheimer's using a deep learning approach. *Journal of Engineering and Applied Science*, *70*, 1–18. https://doi.org/10.1186/s44147-023-00211-x

Cahyadi, R., Damayanti, A., & Aryadani, D. (2020). Recurrent Neural Network (RNN) dengan Long Short Term Memory (LSTM) untuk Analisis Sentimen Data Instagram. *JIKO (Jurnal Informatika Dan Komputer)*, *5*(1), 1–9.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*, 193–202. https://doi.org/10.1007/BF00344251

Gunawardena, K. A. N. N. P., Rajapakse, R. N., & Kodikara, N. D. (2017). Applying convolutional neural networks for pre-detection of alzheimer's disease from structural MRI data. *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. https://doi.org/10.1109/M2VIP.2017.8211486

Haldar, S. (2019). Design and Implementation of an Image Classifier using CNN. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*.

Hayum, A. A., Jaya, J., Paulchamy, B., & Sivakumar, R. (2023). A modified recurrent neural network (MRNN) model for and breast cancer classification system. *Automatika*, *64*(4), 1193–1203. https://doi.org/10.1080/00051144.2023.2253064

Jafari, Z., & Karami, E. (2023). *Breast Cancer Detection in Mammography Images: A CNN-Based Approach with Feature Selection*. Preprints. https://doi.org/10.20944/preprints202305.2209.v1

Nasser, M., & Yusof, U. K. (2023). Deep Learning Based Methods for Breast Cancer Diagnosis: A Systematic Review and Future Direction. *Diagnostics*, *13*(1), 161. https://doi.org/10.3390/diagnostics13010161

Nguyen, M., He, T., An, L., Alexander, D. C., Feng, J., Yeo, B. T. T., & the Alzheimer's Disease Neuroimaging Initiative. (2020). Predicting Alzheimer's disease progression using deep recurrent neural networks. *NeuroImage*, *222*, 117203. https://doi.org/10.1016/j.neuroimage.2020.117203

Putra, I. W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. *Jurnal Teknik ITS*, *5*(1), A65–A69. https://doi.org/10.12962/j23373539.v5i1.15696

Reguant, R., Brunak, S., & Saha, S. (2021). Understanding inherent image features in CNN-based assessment of diabetic retinopathy. *Scientific Reports*, *11*. https://doi.org/10.1038/s41598-021-89225-0

Yang, Z., Tan, T.-E., Shao, Y., Wong, T. Y., & Li, X. (2022). Classification of diabetic retinopathy: Past, present and future. *Frontiers in Endocrinology*, *13*. https://doi.org/10.3389/fendo.2022.1079217

Zhang, C., Sargent, I., Pan, X., Gardiner, A., Hare, J., & Atkinson, P. M. (2018). VPRS-Based Regional Decision Fusion of CNN and MRF Classifications for Very Fine Resolution Remotely Sensed Images. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(8), 4507–4521. https://doi.org/10.1109/TGRS.2018.2822783