International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 6, December 2024, pp. 6466~6477
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i6.pp6466-6477

Bone fracture classification using convolutional neural network architecture for high-accuracy image classification

Solikhun[1], Agus Perdana Windarto[2], Putrama Alkhairi[2]
[1]Informatics Engineering Program, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
[2]Information Systems Program, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia

Article Info

Article history:

Received May 21, 2024
Revised Jul 13, 2024
Accepted Jul 17, 2024

Keywords:

Bone fractures
Convolutional neural network
Deep learning
Image classification
Medical imaging

ABSTRACT

This research introduces an innovative method for fracture classification using convolutional neural networks (CNN) for high-accuracy image classification. The study addresses the need to improve the subjectivity and limited accuracy of traditional methods. By harnessing the capability of CNNs to autonomously extract hierarchical features from medical images, this research surpasses the limitations of manual interpretation and existing automated systems. The goal is to create a robust CNN-based methodology for precise and reliable fracture classification, potentially revolutionizing current diagnostic practices. The dataset for this research is sourced from Kaggle's public medical image repository, ensuring a diverse range of fracture images. This study highlights CNNs' potential to significantly enhance diagnostic precision, leading to more effective treatments and improved patient care in orthopedics. The novelty lies in the unique application of CNN architecture for fracture classification, an area not extensively explored before. Testing results show a significant improvement in classification accuracy, with the proposed model achieving an accuracy rate of 0.9922 compared to ResNet50's 0.9844. The research suggests that adopting CNN-based systems in medical practice can enhance diagnostic accuracy, optimize treatment plans, and improve patient outcomes.

Corresponding Author:

Solikhun
Informatics Engineering Program, STIKOM Tunas Bangsa
Kartini street, Proklamasi, Siantar Bar., Pematangsiantar, North Sumatra 21143, Indonesia
Email: solikhun@amiktunasbangsa.ac.id

1. INTRODUCTION

The technological landscape has witnessed unprecedented progress, with a significant focus on artificial intelligence (AI) as a transformative force in various industries [1]–[3]. One of the main driving factors behind this surge is the exponential growth in computing power, which has enabled the execution of complex AI algorithms at scale [4]. Integrating AI into everyday life is becoming increasingly widespread, with applications ranging from virtual assistants and recommendation systems to autonomous vehicles and advanced healthcare diagnostics. In the healthcare sector, AI has become a breakthrough. The ability of AI algorithms to process large amounts of medical data, from electronic health records to medical imaging, has paved the way for more accurate diagnostics and personalized treatment plans [5], [6]. Machine learning models can now predict disease, analyze genetic data, and assist doctors in making decisions, ultimately improving patient outcomes [7], [8]. In the healthcare sector, AI has become a breakthrough. The ability of AI algorithms to process large amounts of medical data, from electronic health records to medical imaging, has paved the way for more accurate diagnostics and personalized treatment plans. Machine learning models can now predict disease, analyze genetic data, and help doctors make decisions, ultimately improving patient outcomes. At the forefront of AI, deep learning has sparked a revolution in machine learning methodologies [9]–[11]. This approach, inspired by the human brain's structure and function, involves using neural networks with many layers (deep neural networks). Deep learning excels at feature extraction and pattern recognition, allowing systems to automatically learn complex representations from data. This paradigm shift has resulted in breakthroughs in image and speech recognition and natural language processing and significantly enhanced the capabilities of AI systems. When analyzing disease using digital images, convolutional neural network (CNN) is currently highly recommended. Medical disease in fractures is closely related to digital images. There are many types of medical diseases, such as fractures, and CNN can take part in these fracture cases [12]–[14].

Fractures are a common injury and have significant implications for patient care and treatment planning [15]–[17]. Accurate fracture classification is important in guiding health professionals toward appropriate interventions. Traditionally, this classification process relies on manual interpretation of medical imaging, which can be time-consuming and subjective. In addition, existing automated methods often need help to achieve desired levels of accuracy, requiring more sophisticated approaches to improve diagnostic yield. In response to these challenges, the integration of AI and deep learning techniques, especially CNNs, has emerged as a promising solution to improve the accuracy of fracture classification from medical images [18], [19].

Conventional methods for fracture classification have limitations, including reliance on human expertise and potential interobserver variability. Manual interpretation is labor intensive and can lead to errors, impacting patient care. Automated systems, while offering a degree of efficiency, often face challenges in handling the complexity and variation inherent in medical images. As a result, there is an urgent need for advanced technologies capable of providing accurate, consistent, and rapid classification of fractures, thus laying the foundation for more effective treatment strategies [20]–[24].

The emergence of deep learning, and CNNs in particular, has revolutionized medical image analysis. CNNs excel at capturing hierarchical features from images, making them suitable for tasks such as image classification [25], [26]. In the context of fracture classification, using CNN architecture is promising to overcome the shortcomings of traditional methods. By automatically learning complex patterns and representations from medical images, CNNs have demonstrated superior performance in various medical fields, fostering optimism about their potential to improve fracture classification accuracy significantly.

Previous research conducted by Tanzi *et al.* [15] with the title "X-Ray bone fracture classification using deep learning: a baseline for designing a reliable approach" this research analyzed various studies that discussed bone fracture classification using a deep learning approach. I examined the studies based on certain criteria, such as data quality, image processing methods, type of neural network, and the final results. The research results show that the most effective approach is to use pre-training with a larger dataset, perform pre-processing correctly, use data augmentation techniques wisely, and implement class activation mapping (CAM) to understand the focus of the neural network. However, this study needs to be stronger in that it also does not fully utilize visualization techniques such as Grad-CAM to understand the focus of neural networks, which can help validate bone fracture classification results. Lastly, this article also needs to provide more information about evaluation by specialists, which may affect the validity of the classification results.

Meanwhile, the next research conducted by Yoon *et al.* [27] with the title "Automatic multi-class intertrochanteric femur fracture detection from CT images based on AO/OTA classification using faster R-CNN-BO method" study aims to develop a multi-class femur fracture detection model. -automatic class uses deep learning to classify fracture types and locations in a single flow. Based on the annotated expert data, the femur fracture image is transformed into three regions to train the data for the deep, faster R-CNN model. The proposed model performs better than previous studies when classifying and detecting region of interests (ROIs). This femur detection model can reduce differences in diagnosis due to the level of surgical experience and provide deeper insight when selecting or planning treatment of femur fractures. However, the accuracy of the proposed model needs to be improved when classifying femoral fractures into ten subgroups. The tendency for the number of false positives (FPs) to be higher than true positives (TPs) in the proposed model is in addition; this study also showed that the sensitivity and positive predictive value (PPV) for several classifications of femur fracture types tended to decrease as the number of classified classes increased.

Based on previous literature studies and related research objects, the focus of this research proposal on CNN architecture is deliberate and strategic. CNNs are designed to mimic the human visual system [28], [29], utilizing convolutional layers to extract relevant features from input images. This architecture allows the model to distinguish complex patterns, textures, and spatial relationships important for accurate image classification. The incorporation of pooling layers and fully connected layers further enhances this process, making CNNs highly effective in handling medical imaging data. By adopting a CNN architecture, this study aims to harness the power of deep learning to achieve high accuracy in bone fracture classification, thereby overcoming the shortcomings of existing methods.

The emphasis on CNN architectures underscores the commitment to leverage state-of-the-art technologies to improve clinical outcomes with novel proposed CNN architectures [30], [31]. The main reason behind this research is to advance the field of fracture classification by introducing a state-of-the-art CNN-based approach. The ultimate goal is to develop a robust system that accurately categorizes fractures from medical images, ultimately facilitating timely and appropriate clinical decision-making. Through this research, we aim to contribute to the evolution of medical image analysis, paving the way for more efficient and accurate diagnostic processes in orthopedic practice.

This research introduces a new paradigm and architecture as a novelty in the field of fracture classification by strategically exploiting the transformative potential of CNN architecture [32]–[34]. Its novelty lies in the deliberate and specialized application of CNNs to significantly improve image classification accuracy, specifically overcoming the challenges inherent to traditional methods. By utilizing the hierarchical feature extraction capabilities of CNN, this research aims to overcome the limitations of manual interpretation and the shortcomings of existing automatic systems. Unlike conventional approaches, the proposed methodology goes beyond automation, delving into deep learning to empower systems with the capacity to distinguish complex patterns and subtle nuances in medical images autonomously. This innovation represents a paradigm shift in fracture classification. It underscores a commitment to advancing the field through cutting-edge technology, paving the way for more precise diagnostic capabilities in orthopedic practice.

2. METHOD

This research was carried out with the main aim of analyzing and evaluating comparative model optimization techniques on CNN in the process of classifying bone fractures using X-ray CT scan data. This study is very important because the classification results of this data serve to obtain in-depth information regarding the comparative analysis of the various CNN model optimization techniques applied. The research process began by collecting X-ray CT scan data from various bone fracture cases. This data is then processed and used as input for the CNN model. The CNN model used has been specifically designed to detect and classify types of bone fractures based on patterns and features identified in CT scan images. The results of this study show that applying appropriate optimization techniques can significantly improve the performance of CNN models in classifying bone fractures. The comparative analysis performed provides valuable insight into the advantages and disadvantages of each optimization technique. Thus, this research not only contributes to the development of technology in the health sector, especially in image-based medical diagnosis, but also offers practical guidance for researchers and practitioners who wish to optimize their CNN models.

2.1. Data collection methods

The data collection method used in research to obtain data and information is open-source data. We can explore these resources to create a research series with relevant data. The data was received from the Kaggle.com site.

2.2. Architecture CNN

Convolutional neural networks (or CNNs) are a special type of multi-layer neural network or deep learning architecture inspired by the visual systems of living things. CNN are deep learning algorithms designed to process two-dimensional data [35]–[37]. CNNs are usually used to learn and detect features in an image. Figure 1 is a sample of the CNN architecture with a deep learning approach. In Figure 1 model optimization in CNN refers to refining and improving a CNN model's performance. Various parameters and configurations are adjusted to improve the network's ability to accurately understand and represent complex patterns in image data [38].

Refinements are important for adapting existing CNN models to work well on a particular task or data set. Pretrained models provide a foundation for recognizing common features, while customization adapts the model to specific patterns and information relevant to a particular application. This process is important for achieving optimal performance in image classification, object detection, and segmentation tasks. These optimization techniques collectively contribute to CNNs' effectiveness and efficiency in various computer vision tasks. Researchers often consider combinations and variations of these techniques to find the optimal configuration for a particular application. Train a CNN using different optimization optimizers to compare with different batch sizes. The authors use popular deep learning frameworks like TensorFlow for this task. We present the CNN model optimization flowchart in Figure 2. The explanation of Figure 2 is as follows:

- Start: Flowcharts usually start with the "Start" symbol.

– Data collection: Use rectangular boxes to represent the steps when you load your data set. Label it "Load Data." Figures 3(a) and 3(b) are samples of the data used in this research.

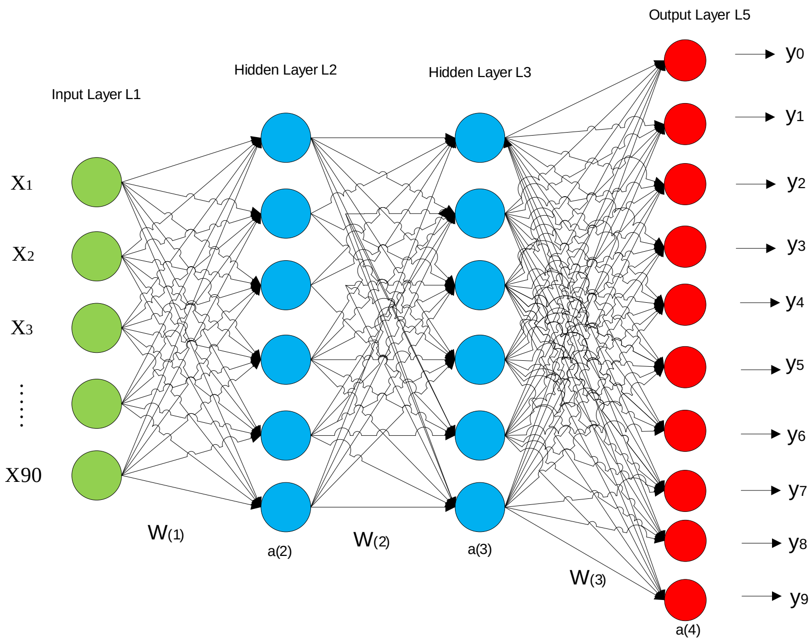


Figure 1. CNN algorithm architecture

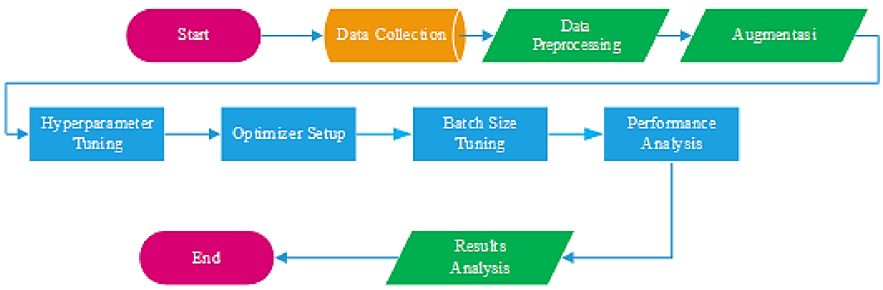
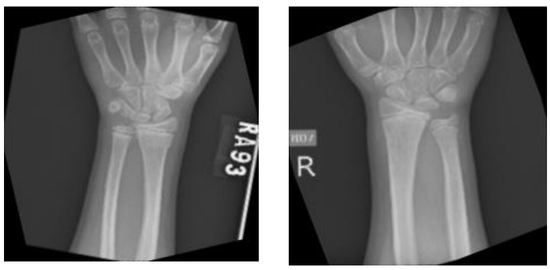


Figure 2. Model optimization flowchart



(a) (b)

Figure 3. Data sample (a) fracture and (b) not fracture

Figure 3(a) Fracture: The image shows a wrist bone fracture marked by irregular lines or clear breaks, indicating damage from trauma or pressure. Figure 3(b) No fracture: The image displays an intact wrist bone with smooth lines and no signs of cracks or breaks, indicating a healthy condition. The main difference is that Figure 3(a) shows a fracture, while Figure 3(b) depicts an undamaged bone.

- Data preprocessing: add other rectangular boxes for data preprocessing steps, such as data normalization and augmentation, if applicable. Label it "data preprocessing."
- Data augmentation: data augmentation is very important to prevent overfitting and improve the model's generalization ability. Common augmentation techniques include rotation, flipping, scaling, and shifting. Augmentation implementations are often applied to training datasets when using libraries such as TensorFlow or PyTorch.
- Hyperparameter tuning: learning rate (η) controls the step size during optimization. Number of layers and convolutional filters architecture-related hyperparameters affect the network's depth and width. dropout rate is a regularization technique to prevent overfitting. Kernel size and stride: Parameters that determine the size of the convolutional kernel and its stride. Batch normalization can determine if and where a batch normalization layer is added. Activation function there are many options, such as ReLU, Sigmoid, or Tanh, for the activation function. Number of hidden units in dense layers architectural options for fully connected layers.
- Optimizer setup: common optimizers include Adam: adaptive moment estimation, popular for its adaptive learning rate and momentum. RMSProp: root mean square propagation, similar to Adam but with different update rules. Learning rate schedule: Adjusting the learning rate during training (for example, reducing it over time) can improve convergence. Weight decay: L2 regularization is applied to weights during optimization.
- Batch size: impact on training: larger batch sizes provide computational efficiency but may result in loss of generalization. Smaller batch sizes introduce more noise but can improve generalization. Selection: often depends on available computing resources and specific data set characteristics.
- Performance analysis: training and validation loss curves: plotting loss over time helps assess convergence and identify overfitting. Accuracy metrics: check accuracy, precision, recall, and F1-score on training and validation sets. Confusion matrix: provides insight into model performance across classes. Visual inspection: analyze misclassified samples and understand model issues.
- Refinement: Iterative process: hyperparameter refinements, architectural adjustments, and optimization choices may need to be iteratively refined. Regularization techniques: Apply dropout or batch normalization techniques to improve generalization. Early stop: training stops when the model's performance on the validation set no longer improves.
- Define CNN architecture: create a rectangular box to define the CNN architecture, including the number of layers, filters, and activation functions. Label it "Define CNN architecture". Determine batch sizes: use a diamond shape (decision symbol) to indicate that you will make decisions based on different batch sizes. Label it "Select batch size". Batch size = 128: Add an arrow pointing to the rectangular box with "Batch size = 128." In this box, you can indicate that model training was done with a batch size of 128. Batch size = 256: Add an arrow pointing to another rectangular box with "Batch size = 256." In this box, you can indicate that model training has been done with a batch size 256. In Figures 4 and 5, the following is the standard ResNet50 model with the proposed model.

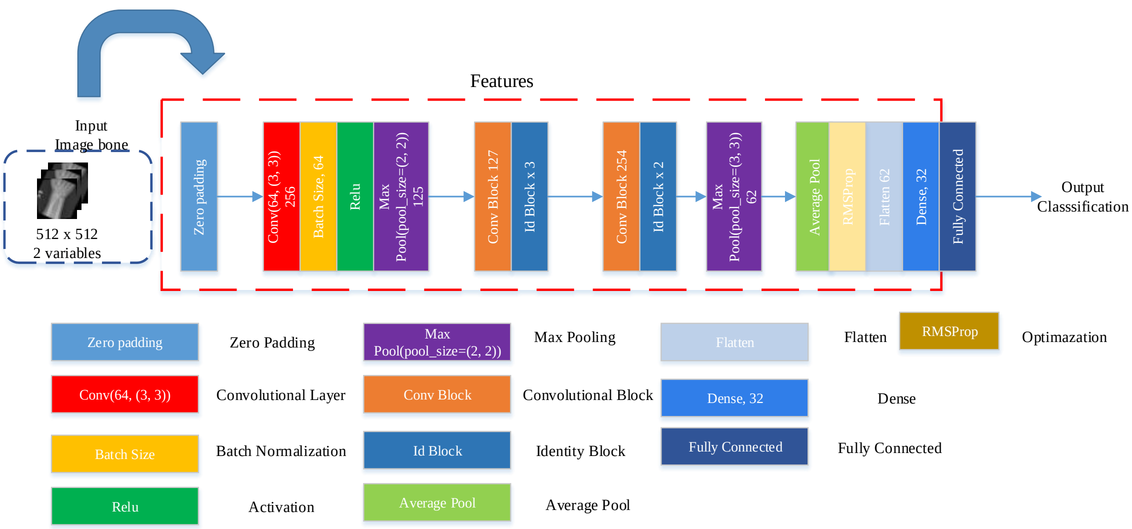


Figure 4. Standard ResNet50 model

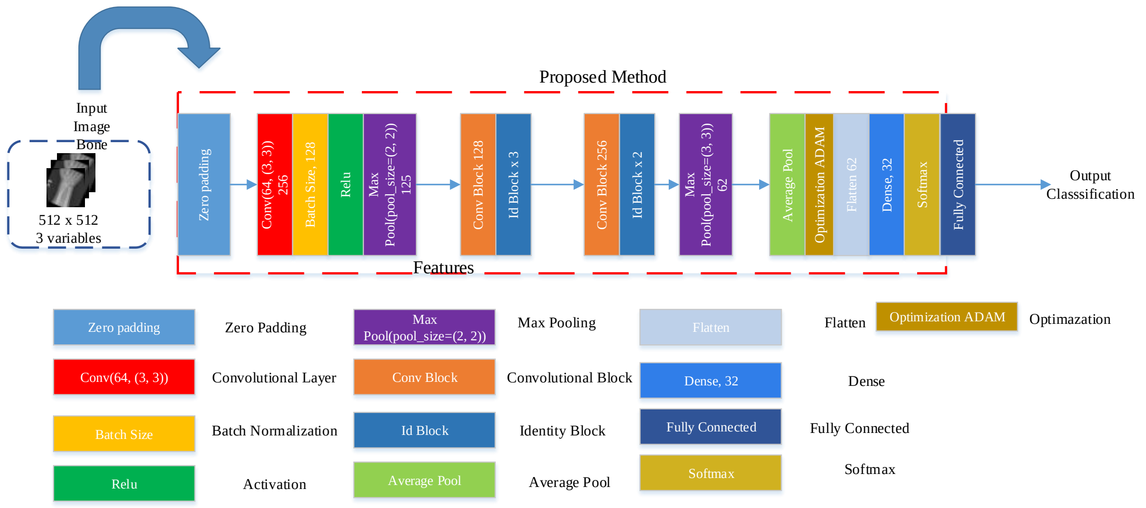


Figure 5. Proposed ResNet model

Train the model: i) Create a rectangular box to represent the model training steps; ii) Label it "Train model"; iii) This box shows the training process with epochs, loss, and accuracy; and iv) Use arrows to connect these boxes to the three batch size options, indicating that the model was trained with different sizes.

– Model evaluation: Add a rectangular box labeled "model evaluation" to show how the model performance was evaluated using validation or test data. In the evaluation stage, the performance of all 2 classifiers in classifying the bone data set into "bone fracture" and "non-fracture" was assessed. This process involves the utilization of confusion matrices, ROC curves, and Eq. The ROC curve and confusion matrix are built using the false positive rate and true positive rate obtained from calculations with the confusion matrix. Performance is determined by the AUC of the ROC curve, as shown in Algorithms 1 and 2. The larger the area under the curve indicates better performance.

$$\text{Accuracy}=\frac{TP+TN}{TP+FP+FN+TN} \tag{1}$$

$$\text{Precision}=\frac{TP}{TP+FP} \tag{2}$$

$$\text{Recall}=\frac{TP}{TP+FN} \tag{3}$$

– Performance analysis: Create a rectangular box labeled "performance analysis." In this box, you can indicate that you analyzed and compared performance metrics, such as accuracy, for different batch sizes. Accuracy is a metric that measures how well a model predicts overall, expressed as the percentage of correct predictions (true positives and true negatives) compared to the total number of forecasts (true positives, false positives, false negatives, and true negatives). In contrast, precision measures the proportion of true positive predictions out of all positive predictions made by the model, indicating its "precision" or accuracy in classifying positive labels. Recall, also known as sensitivity, measures the proportion of all positive cases a model successfully detects, reflecting its ability to "recall" or "detect" positive cases. Decision:
  a. Use the decision symbol to determine whether more batch sizes must be evaluated.
  b. If yes, loop back to the "Select Batch Size" decision symbol.
  c. If not, continue to the next step.
  End: Add an "End" symbol to indicate the end of the flowchart.

3. RESULTS AND DISCUSSION

Based on the results of training, validation, and testing, various hyperparameters and architectures are used to see the results of bone image classification. The training process involves the use of a bone image dataset that is divided into three subsets: training data, validation data, and testing data. Each subset serves a different purpose, namely training the model, tuning hyperparameters, and evaluating the final performance of the model. During training, the CNN model is tested with various combinations of hyperparameters such as batch size, number of epochs, learning rate, and network layer structure. The use of various CNN architectures, ranging from simple to complex, was also explored to determine their impact on bone image classification accuracy.

3.1. Result

The model training results presented in this article include some important information that explains the performance of the optimized CNN model for fracture classification. It uses the CNN architecture for High accuracy image classification with ResNet50 Plus model as the proposed model and ResNet50 Standard, as shown in Figure 5. The following is an explanation of the results for each epoch:

The ResNet50 plus model is optimized to improve the accuracy of bone image classification, taking into account factors such as network depth, number of filters, and regulation techniques used during training. In the training process, the model is tested over multiple epochs, where each epoch involves complete iteration through the entire training dataset. Figure 5 displays a graph of the model's performance during training, showing important metrics such as accuracy and loss on training and validation data. At each epoch, the model's accuracy is measured to assess how well the model can recognize patterns in bone images. Loss, which indicates how well the model fits the training data, is also monitored to ensure the model is not overfitting or underfitting.

The training process above, Figure 6 shows that after the 30th epoch, the training process was stopped because the loss value did not decrease for 5 consecutive epochs, which indicates that the model has reached convergence or is possibly too fit. Total training time is 30 minutes for ResNet50 plus (proposed method) and 25 minutes for ResNet50. This shows that the difference in training time is quite different. This will result in time inefficiencies in training, which is too time-consuming.

```
Epoch 25
Train_cost=0.0348 | Test_cost=0.0326 | Train_score=0.9899
| Test_score=0.9950 |
Train: 100%
 33/33 [00:30<00:00, 2.44it/s]
Test: 100%
 4/4 [00:04<00:00, 1.13it/s]

Epoch 26
Train_cost=0.0326 | Test_cost=0.0465 | Train_score=0.9879
| Test_score=0.9799 |
==> EarlyStop patience=1 | Best test_score: 0.9950
Train: 100%
 33/33 [00:32<00:00, 1.80it/s]
Test: 100%
 4/4 [00:03<00:00, 1.63it/s]

Epoch 27
Train_cost=0.0417 | Test_cost=0.0196 | Train_score=0.9865
| Test_score=0.9925 |
==> EarlyStop patience=2 | Best test_score: 0.9950
Train: 100%
 33/33 [00:35<00:00, 2.03it/s]
Test: 100%
 4/4 [00:03<00:00, 1.39it/s]

Epoch 28
Train_cost=0.0340 | Test_cost=0.0199 | Train_score=0.9891
| Test_score=0.9900 |
==> EarlyStop patience=3 | Best test_score: 0.9950
Train: 100%
 33/33 [00:31<00:00, 2.35it/s]
Test: 100%
 4/4 [00:03<00:00, 1.09s/it]

Epoch 29
Train_cost=0.0325 | Test_cost=0.0197 | Train_score=0.9908
| Test_score=0.9950 |
==> EarlyStop patience=4 | Best test_score: 0.9950
Train: 100%
 33/33 [00:50<00:00, 1.54it/s]
Test: 100%
 4/4 [00:03<00:00, 1.04it/s]

Epoch 30
Train_cost=0.0183 | Test_cost=0.0165 | Train_score=0.9947
| Test_score=0.9925 |
==> EarlyStop patience=5 | Best test_score: 0.9950
==> Execute Early Stopping at epoch: 30 | Best test_score:
0.9950
==> Best model is saved at model
```

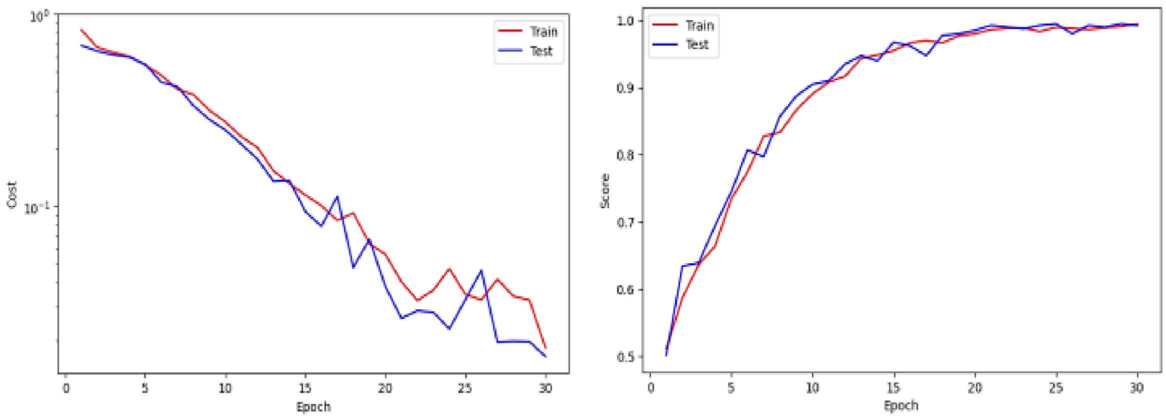
(a)

```
Epoch 25
Train_cost=0.1060 | Test_cost=0.0541 | Train_score=0.9645
| Test_score=0.9724 |
==> EarlyStop patience=1 | Best test_score: 0.9875
Train: 100%
 33/33 [00:36<00:00, 2.63it/s]
Test: 100%
 4/4 [00:03<00:00, 1.26it/s]

Epoch 26
Train_cost = 0.0286 | Test_cost = 0.0313 | Train_score =
0.9918 | Test_score = 0.9875 |
==> EarlyStop patience = 2 | Best test_score: 0.9875
Train: 100%
 33/33 [00:31<00:00, 1.85it/s]
Test: 100%
 4/4 [00:04<00:00, 1.25it/s]

Epoch 27
Train_cost = 0.0246 | Test_cost = 0.0587 | Train_score =
0.9918 | Test_score = 0.9875 |
==> EarlyStop patience = 3 | Best test_score: 0.9875
Train: 100%
 33/33 [00:30<00:00, 1.72it/s]
Test: 100%
 4/4 [00:03<00:00, 1.24it/s]

Epoch 28
Train_cost = 0.1039 | Test_cost = 0.0333 | Train_score =
0.9642 | Test_score = 0.9850 |
==> EarlyStop patience = 4 | Best test_score: 0.9875
Train: 100%
 33/33 [00:30<00:00, 2.61it/s]
Test: 100%
 4/4 [00:03<00:00, 1.08s/it]

Epoch 29
Train_cost = 0.0230 | Test_cost = 0.0368 | Train_score =
0.9942 | Test_score = 0.9875 |
==> EarlyStop patience = 5 | Best test_score: 0.9875
==> Execute Early Stopping at epoch: 29 | Best test_score:
0.9875
==> Best model is saved at model
```

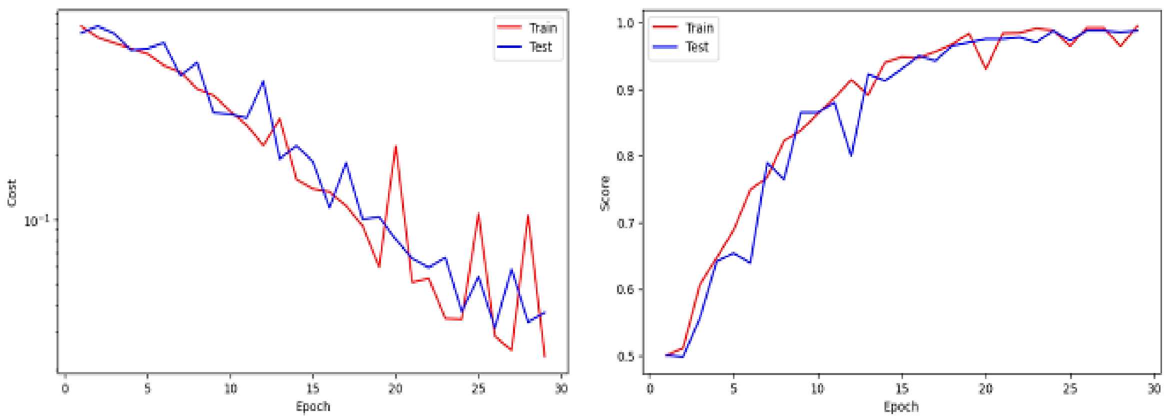
(b)

Figure 6. Training process comparison between (a) ResNet50 plus (proposed method) and (b) ResNet50 standard

Figure 6 is a graph that compares the performance of two algorithms, in Figure 6(a) ResNet50 plus (proposed method) and Figure 6(b) ResNet50, with 30 and 29 epochs, respectively. On the cost (loss) graph on a logarithmic scale, ResNet50 plus shows a decrease from around 1 (epoch 0) to around 0.02 (epoch 30) for training and testing data with early stop patience = 5, Best test score: 0.9950, Execute early stopping at epoch: 30, Best test score: 0.9950, with relatively small fluctuations. ResNet50 also shows a decrease in cost from around 1 to around 0.02 with early stop patience = 5, Best test score: 0.9875, Execute early stopping at epoch: 29, Best test score: 0.9875, but with larger fluctuations. On the score (accuracy) graph, ResNet50 plus improves from around 0.5 (epoch 0) to around 1 (epoch 30) for both training and test data, with smoother and more consistent lines. Meanwhile, ResNet50 increases from around 0.5 to 1 with larger fluctuations, especially in the early epoch. ResNet50 plus (proposed method) shows a more stable and consistent performance than ResNet50. Figures 7(a) and 7(b) shows the confusion matrix of the two architectures tested: ResNet50 plus (proposed method) and ResNet50 standard.



(a)



(b)

Figure 7. Loss learning graph and score learning curve graph fracture image classification training process using architecture: (a) ResNet50 plus (proposed method) and (b) ResNet50 standard

Figure 8 shows the confusion matrix for two algorithms, Figure 8(a) ResNet50 plus (proposed method) and Figure 8(b) standard ResNet50, which classify data into two categories: fractured and not fractured. In the proposed method, the algorithm correctly classifies 57 out of 57 fractured images (True Positive) and incorrectly classifies 1 not fractured image as fractured (False Positive), with 70 out of 71 not fractured images being correctly classified (True Negative). In contrast, the standard method correctly classifies 64 out of 64 fractured images (True Positive) and incorrectly classifies 2 not fractured images as fractured (False Positive), with 62 out of 64 not fractured images being correctly classified (True Negative). The proposed method shows higher accuracy with fewer misclassifications.

ResNet50 plus (proposed model):

$$Akurasi = \frac{57+70}{57+70+1+0} = \frac{127}{128} \approx 0.9922$$

$$Presisi = \frac{57}{57+1} = \frac{57}{58} \approx 0.9828$$

$$Recall = \frac{57}{57+0} = \frac{57}{57} \approx 1.0$$

$$F1\ Score = \frac{2.(0.9828\ .1.0)}{0.9828\ .1.0} = \frac{1.9656}{1.9828} \approx 0.9922$$

ResNet50 Standard:

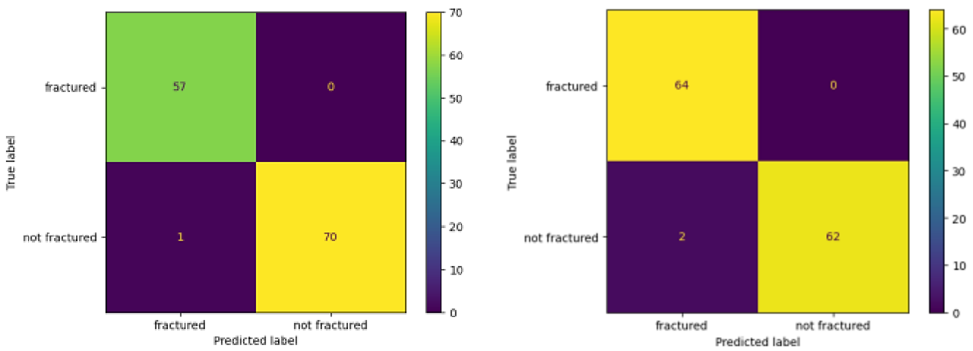
$$Akurasi = \frac{64+62}{64+62+2+0} = \frac{126}{128} \approx 0.9844$$

$$Presisi = \frac{64}{64+2} = \frac{64}{68} \approx 0.9697$$

$$Recall = \frac{64}{64+0} = \frac{64}{64} \approx 1.0$$

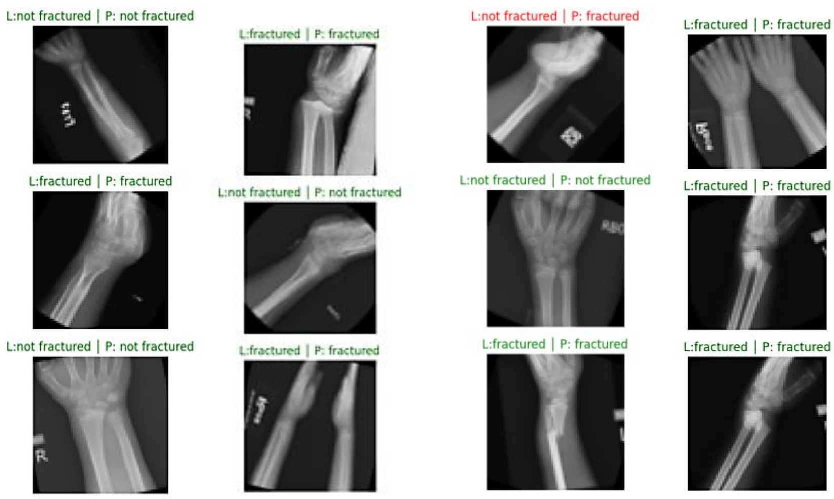
$$F1\ Score = \frac{2.(0.9697\ .1.0)}{0.9697\ .1.0} = \frac{1.9394}{1.9697} \approx 0.9847$$

In ResNet50 plus, of the 58 fractured cases, 57 were correctly classified (true positive), and 1 was incorrectly classified as fractured (false positive), while of the 70 not fractured cases, all were correctly classified (true negative). This algorithm produces an accuracy of 0.9922, precision of 0.9828, recall of 1.0, and F1-score of 0.9922. In ResNet50, of the 66 fractured cases, 64 were correctly classified (true positive), and 2 were incorrectly classified as fractured (false positive), while of the 62 not fractured cases, all were correctly classified (true negative). This algorithm produces an accuracy of 0.9844, precision of 0.9697, recall of 1.0, and F1-score of 0.9847. These results show that ResNet50 plus (proposed method) performs slightly better accuracy and F1-score than ResNet50. The classification results can be seen in the Figure 9. Figure 9(a) ResNet50 plus and Figure 9(b) ResNet50 Standard illustrate the classification results of image parameter predictions of fractured and non-fractured bones. The green marks at the bottom of the font are correct predictions; if they are red, they are incorrect or inaccurate.



(a) (b)

Figure 8. Confusion matrix (a) ResNet50 plus dan (b) ResNet50 standard



(a) (b)

Figure 9. Classification results (a) ResNet50 plus and (b) ResNet50 standard

3.2. Discussion

This study aims to compare two CNN architectural models for fracture classification with high accuracy. The two models tested are ResNet50 plus (proposed method) and ResNet50. In Table 1, the classification comparison results of the two CNN architectures are presented. The ResNet50 plus model was developed with several improvements and optimizations to improve performance in classifying bone fracture images. Meanwhile, the ResNet50 model is used as a baseline or standard model for evaluation. Both models were tested using the same dataset, and their performance was compared based on various metrics such as accuracy, precision, recall, and F1-score.

The comparison results shown in Table 1 show the differences in performance between the two models. The ResNet50 plus model, which is the proposed method, shows significant improvement in terms of classification accuracy compared to the standard ResNet50 model. Additionally, improvements in other metrics such as precision, recall, and F1-score are also visible in the ResNet50 plus model.

Table 1. Comparison results of the classification of two CNN architectures

Optimize	Batch size	Optimization	Epoch	Precision	Recall	F1-score	Accuracy
ResNet50 plus (proposed model)	128	Adam	30	0.9828	1.0000	0.9922	0.9922
ResNet50	64	RMSProp	29	0.9697	1.0000	0.9847	0.9844

The research results show that ResNet50 Plus uses a batch size of 128, Adam optimizer, and is trained for 30 epochs. This model achieved a precision of 0.9828, a recall of 1.0000, an F1-score of 0.9922, and an accuracy of 0.9922. On the other hand, ResNet50 uses a batch size of 64, RMSProp optimizer, and is trained for 29 epochs. This model achieved a precision of 0.9697, a recall of 1.0000, an F1-score of 0.9847, and an accuracy of 0.9844. The results show that ResNet50 plus (proposed method) performs slightly better than ResNet50, with a higher precision value of 0.9828 compared to 0.9697 in ResNet50. Even though both models achieved a perfect recall value (1.0000), the F1-score and accuracy of ResNet50 Plus (0.9922) were also superior to ResNet50 (F1-score 0.9847 and accuracy 0.9844).

The success of ResNet50 plus (proposed method) in achieving better results can be attributed to using larger batch sizes and the Adam optimizer, which is known to have better adaptive capabilities in updating parameters. This shows that selecting appropriate parameters and optimization methods can improve model performance in complex image classification tasks such as bone fracture identification. This study's results confirm that ResNet50 plus (proposed method) is more effective in performing bone fracture classification with high accuracy compared with ResNet50, so it can be a better choice for medical applications that require a high level of precision.

4. CONCLUSION

Based on the research results comparing two convolutional neural networks architecture models in bone fracture classification, it can be concluded that ResNet50 plus (proposed method) shows superior performance compared to ResNet50. ResNet50 Plus, which uses a batch size of 128 and the Adam optimizer for 30 epochs, achieves a precision of 0.9828, recall of 1.0000, F1-score of 0.9922, and accuracy of 0.9922. On the other hand, ResNet50, which uses a batch size of 64 and the RMSProp optimizer for 29 epochs, achieves a precision of 0.9697, recall of 1.0000, F1-score of 0.9847, and accuracy of 0.9844. The success of ResNet50 Plus in attaining better results can be attributed to using a larger batch size and the Adam optimizer, which has better adaptive capabilities in updating parameters. These results demonstrate that appropriate parameter selection and optimization methods can improve model performance in complex image classification tasks, such as bone fracture identification. Overall, this study confirms that ResNet50 plus (proposed method) is more effective in performing bone fracture classification with high accuracy than ResNet50. Thus, ResNet50 Plus can be a better choice for medical applications requiring high precision and accuracy.

REFERENCES

[1] P. Alkhairi and A. P. Windarto, "Classification analysis of back propagation-optimized CNN performance in image processing," *Journal of Systems Engineering and Information Technology (JOSEIT)*, vol. 2, no. 1, Mar. 2023, doi: 10.29207/joseit.v2i1.5015.

[2] A. P. Windarto, T. Herawan, and P. Alkhairi, "Early detection of breast cancer based on patient symptom data using naive Bayes algorithm on genomic data," in *The International Conference on Artificial Intelligence and Smart Environment*, 2024, pp. 478–484.

[3] A. P. Windarto, T. Herawan, and P. Alkhairi, "Prediction of kidney disease progression using k-means algorithm approach on histopathology data," in *The International Conference on Artificial Intelligence and Smart Environment*, 2024, pp. 492–497.

[4] S. Defit, A. P. Windarto, and P. Alkhairi, "Comparative analysis of classification methods in sentiment analysis: the impact of

feature selection and ensemble techniques optimization," *Telematika*, vol. 17, no. 1, pp. 52–67, 2024.
[5] A. Hedström *et al.*, "Quantus: an explainable ai toolkit for responsible evaluation of neural network explanations and beyond," *Journal of Machine Learning Research*, vol. 24, no. 34, pp. 1–11, 2023.
[6] C. Qing, N. Yang, S. Tang, J. Chen, and J. Wang, "CNN-aided timing synchronization in OFDM systems by exploiting lightweight cascaded mode," *ICT Express*, Mar. 2023, doi: 10.1016/j.icte.2023.03.003.
[7] R. Thupae, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "Machine learning techniques for traffic identification and classifiacation in SDWSN: a survey," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2018, pp. 4645–4650, doi: 10.1109/IECON.2018.8591178.
[8] G. Nguyen *et al.*, "Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 77–124, Jun. 2019, doi: 10.1007/s10462-018-09679-z.
[9] J. Xie, R. Liu, J. Luttrell, and C. Zhang, "Deep learning based analysis of histopathological images of breast cancer," *Frontiers in Genetics*, vol. 10, Feb. 2019, doi: 10.3389/fgene.2019.00080.
[10] N. Khan, A. V. Singh, and R. Agrawal, "Enhanced deep learning hybrid model of CNN based on spatial transformer network for facial expression recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 14, Nov. 2022, doi: 10.1142/S0218001422520280.
[11] M. Turuk, R. Sreemathy, S. Kadiyala, S. Kotecha, and V. Kulkarni, "CNN based deep learning approach for automatic malaria parasite detection," *IAENG International Journal of Computer Science*, vol. 49, no. 3, 2022.
[12] M. Ahmad *et al.*, "A lightweight convolutional neural network model for liver segmentation in medical diagnosis," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–16, Mar. 2022, doi: 10.1155/2022/7954333.
[13] W. Baccouch, S. Oueslati, B. Solaiman, and S. Labidi, "A comparative study of CNN and U-Net performance for automatic segmentation of medical images: application to cardiac MRI," *Procedia Computer Science*, vol. 219, pp. 1089–1096, 2023, doi: 10.1016/j.procs.2023.01.388.
[14] S. A. Narod, "Personalised medicine and population health: breast and ovarian cancer," *Human Genetics*, vol. 137, no. 10, pp. 769–778, Oct. 2018, doi: 10.1007/s00439-018-1944-6.
[15] L. Tanzi, E. Vezzetti, R. Moreno, and S. Moos, "X-Ray bone fracture classification using deep learning: a baseline for designing a reliable approach," *Applied Sciences*, vol. 10, no. 4, Feb. 2020, doi: 10.3390/app10041507.
[16] A. Hazra, S. Kumar, and A. Gupta, "Study and analysis of breast cancer cell detection using naïve Bayes, SVM and ensemble algorithms," *International Journal of Computer Applications*, vol. 145, no. 2, pp. 39–45, Jul. 2016, doi: 10.5120/ijca2016910595.
[17] H.-C. Breit *et al.*, "CNN-based evaluation of bone density improves diagnostic performance to detect osteopenia and osteoporosis in patients with non-contrast chest CT examinations," *European Journal of Radiology*, vol. 161, Apr. 2023, doi: 10.1016/j.ejrad.2023.110728.
[18] Q. Hou, R. Xia, J. Zhang, Y. Feng, Z. Zhan, and X. Wang, "Learning visual overlapping image pairs for SfM via CNN fine-tuning with photogrammetric geometry information," *International Journal of Applied Earth Observation and Geoinformation*, vol. 116, Feb. 2023, doi: 10.1016/j.jag.2022.103162.
[19] F. Alzubaidi, P. Makuluni, S. R. Clark, J. E. Lie, P. Mostaghimi, and R. T. Armstrong, "Automatic fracture detection and characterization from unwrapped drill-core images using mask R–CNN," *Journal of Petroleum Science and Engineering*, vol. 208, Jan. 2022, doi: 10.1016/j.petrol.2021.109471.
[20] W. Cheng *et al.*, "Prediction of hospitalization and mortality in patients with chronic obstructive pulmonary disease with the new global initiative for chronic obstructive lung disease 2023 group classification: a prospective cohort and a retrospective analysis," *International Journal of Chronic Obstructive Pulmonary Disease*, vol. 18, pp. 2341–2352, Oct. 2023, doi: 10.2147/COPD.S429104.
[21] M. Shanid and A. Anitha, "Adaptive optimisation driven deep belief networks for lung cancer detection and severity level classification," *International Journal of Bio-Inspired Computation*, vol. 18, no. 2, 2021, doi: 10.1504/IJBIC.2021.118101.
[22] H. J. Kim, "Classification of healthy and affected lungs by pneumonia disease from x-ray images of lungs and gene sequencing using inception model," *Journal of medical pharmaceutical and allied sciences*, vol. 11, pp. 4114–4118, 2022.
[23] L. Vinet and A. Zhedanov, "A 'missing' family of classical orthogonal polynomials," *Journal of Physics A: Mathematical and Theoretical*, vol. 44, no. 8, Feb. 2011, doi: 10.1088/1751-8113/44/8/085201.
[24] F. C. Detterbeck *et al.*, "The international association for the study of lung cancer staging project: methods and guiding principles for the development of the ninth edition tnm classification," *Journal of Thoracic Oncology*, vol. 17, no. 6, pp. 806–815, Jun. 2022, doi: 10.1016/j.jtho.2022.02.008.
[25] A. Harshavardhan, S. Cheerla, A. Parkavi, S. A. Latha Mary, K. Qureshi, and H. R. Mhaske, "Deep learning modified neural networks with chicken swarm optimization-based lungs disease detection and severity classification," *Journal of Electronic Imaging*, vol. 32, no. 06, May 2023, doi: 10.1117/1.JEI.32.6.062603.
[26] N. Hong Son and N. Thai-Nghe, "Deep learning for rice quality classification," in *2019 International Conference on Advanced Computing and Applications (ACOMP)*, Nov. 2019, pp. 92–96, doi: 10.1109/ACOMP.2019.00021.
[27] S.-J. Yoon, T. Hyong Kim, S.-B. Joo, and S. Eel Oh, "Automatic multi-class intertrochanteric femur fracture detection from CT images based on AO/OTA classification using faster R-CNN-BO method," *Journal of Applied Biomedicine*, vol. 18, no. 4, pp. 97–105, Dec. 2020, doi: 10.32725/jab.2020.013.
[28] T. Yuan, W. Liu, J. Han, and F. Lombardi, "High performance CNN accelerators based on hardware and algorithm co-optimization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 250–263, Jan. 2021, doi: 10.1109/TCSI.2020.3030663.
[29] D. Ruan, J. Wang, J. Yan, and C. Gühmann, "CNN parameter design based on fault signal analysis and its application in bearing fault diagnosis," *Advanced Engineering Informatics*, vol. 55, Jan. 2023, doi: 10.1016/j.aei.2023.101877.
[30] M. M. Emon, T. R. Ornob, and M. Rahman, "Classifications of skull fractures using CT scan images via cnn with lazy learning approach," *Journal of Computer Science*, vol. 18, no. 3, pp. 116–129, Mar. 2022, doi: 10.3844/jcssp.2022.116.129.
[31] A. Jamali, S. K. Roy, and P. Ghamisi, "WetMapFormer: A unified deep CNN and vision transformer for complex wetland mapping," *International Journal of Applied Earth Observation and Geoinformation*, vol. 120, Jun. 2023, doi: 10.1016/j.jag.2023.103333.
[32] Y. Zhu, H. Sekiya, T. Okatani, M. Tai, and S. Morichika, "B-CNN: a deep learning method for accelerometer-based fatigue cracks monitoring system," *Journal of Civil Structural Health Monitoring*, vol. 13, no. 4–5, pp. 947–959, Jun. 2023, doi: 10.1007/s13349-023-00690-9.
[33] J. D. Rosita P and W. S. Jacob, "Multi-objective genetic algorithm and CNN-based deep learning architectural scheme for effective spam detection," *International Journal of Intelligent Networks*, vol. 3, pp. 9–15, 2022, doi: 10.1016/j.ijin.2022.01.001.
[34] H. S. Hoang, C. Phuong Pham, D. Franklin, T. van Walsum, and M. Ha Luu, "An evaluation of CNN-based liver segmentation methods using multi-types of CT abdominal images from multiple medical centers," in *2019 19th International Symposium on*

Communications and Information Technologies (ISCIT), Sep. 2019, pp. 20–25, doi: 10.1109/ISCIT.2019.8905166.
[35] N. N. Prakash, V. Rajesh, D. L. Namakhwa, S. Dwarkanath Pande, and S. H. Ahammad, "A DenseNet CNN-based liver lesion prediction and classification for future medical diagnosis," *Scientific African*, vol. 20, Jul. 2023, doi: 10.1016/j.sciaf.2023.e01629.
[36] A. Naik and D. R. Edla, "Lung nodule classification using combination of CNN, second and higher order texture features," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 5, pp. 5243–5251, Nov. 2021, doi: 10.3233/JIFS-189847.
[37] T. H. Chun, U. R. Hashim, S. Ahmad, L. Salahuddin, N. H. Choon, and K. Kanchymalay, "Efficacy of the image augmentation method using CNN transfer learning in identification of timber defect," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, doi: 10.14569/IJACSA.2022.0130514.
[38] Z. A. Sejuti and M. S. Islam, "A hybrid CNN–KNN approach for identification of COVID-19 with 5-fold cross validation," *Sensors International*, vol. 4, 2023, doi: 10.1016/j.sintl.2023.100229.

BIOGRAPHIES OF AUTHORS

Solikhun is an associate professor in the STIKOM Tunas Bangsa Informatics Engineering Undergraduate Study Program. In 2022 he will also become Chair of the STIKOM Tunas Bangsa Informatics Engineering Undergraduate Study Program. He is a writer who completed his doctoral education at the Faculty of Computer Science and Information Technology, University of North Sumatra. where he is the author/co-author of more than 2013 research publications. He can be contacted at email: solikhun@amiktunasbangsa.ac.id.

Agus Perdana Windarto the authors were born in Pematangsiantar on August 30 1986. They completed their masters in computer science and doctoral (Ph.D.) degrees at Putra Indonesia University 'YPTK' Padang in 2014 and 2024. The author has been an active lecturer at STIKOM Tunas Bangsa since 2012, teaching in the information systems program. Their research focuses on artificial intelligence (decision support systems, expert systems, data mining, neural networks, fuzzy logic, deep learning, and genetic algorithms). Additionally, the author has served as a reviewer for various nationally accredited journals (SINTA 2 - SINTA 6) and manages a community called "Pemburu Jurnal" at STIKOM Tunas Bangsa. They have won multiple research grant proposals from DIKTI (twice in 2018-2019), DIKTI Community Service Grant (once in 2019), PKM-P Grant (as a student advisor in 2018), and PKM-AI Grant (as a student advisor in 2019). The author is also part of the Relawan Jurnal Indonesia (RJI) community in North Sumatra, the Data Science Indonesia Researchers Association (PDSI), the Forum of Higher Education Communities (FKPT), and is a co-founder of the Yayasan Adwitiya Basurata Inovasi (Yayasan Abivasi) foundation with fellow professors. He can be contacted at email: agus.perdana@amiktunasbangsa.ac.id or agus.perdana@abivasi.id.

Putrama Alkhairi the writer Putrama Alkhairi was born in Sinaksak on April 17 1997. They completed a master's degree in computer science in 2023 at the Main Potential University (UPU) Medan. The author has been an active lecturer at STIKOM Tunas Bangsa since 2021, teaching in the information systems program. Their research focuses on artificial intelligence (decision support systems, data mining, artificial neural networks, deep learning, and genetic algorithms). Apart from that, the author has also been active as a member of a community called "Journal Hunters" at STIKOM Tunas Bangsa as a student who took part in joint research with lecturers and has won several awards as the best article writer at the national seminar held by SENARIS 2018. He can be contacted at email: putrama@amiktunasbangsa.ac.id.