Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

Kidney Diseases Classification Based on CT-Radiography Image Using Convolutional Neural Network ResNet-50

Icha Dwi Safira[1]✉, Dian Candra Rini Novitasari[2*]✉
Matematika, UIN Sunan Ampel Surabaya, Indonesia

Abstract
In 2022, kidney disease is impacting over 10% of the global population, which amounts over 800 million individuals. Kidney disease is projected to become the fifth most significant cause of mortality by 2040. Nephrolithiasis (kidney stones), kidney cysts, and Renal Cell Carcinoma (kidney tumor), are three kidney disease that can impair kidney function. Diagnosing kidney disease is crucial before any treatment can be given. Computer Tomography (CT) Radiography images can be used to determine the kind and severity of the disease. The obtained imaging data reveals normal findings and three specific kidney disease: cysts, tumors, and nephrolithiasis. The purpose of this research is to evaluate the efficacy of the Convolutional Neural Network (CNN) ResNet-50 with a hyperparameter tuning experiments in diagnose kidney disease based on the findings of kidney disease classified. The greatest outcomes in this research were achieved with a learning rate of 0.001, dropout probability of 0.1, and mini batch size of 32. The experiments yielded perfect sensitivity, accuracy, and specificity, all reaching 100%. The computation time required was 15 hours, 53 minutes, and 34 seconds. The CNN ResNet-50 ability to accurately diagnose kidney disease is demonstrated by these result.
Keywords: CNN; hyperparameter tuning; kidney disease; ResNet-50;

***Corresponding Author:**
Dian Candra Rini Novitasari, UIN Sunan Ampel Surabaya, Indonesia
Email: diancrini@uinsa.ac.id

1. INTRODUCTION

Kidney disease is a major health issue, impacting over 10% of the world's population, or more than 800 million individuals by 2022 [1]. Projections for 2040 suggest that kidney disease will become the fifth leading cause of death [2]. Research in this field indicates a concerning rise in global kidney disease fatalities, with the number reaching approximately 10 million by 2017 [3]. It captured global attention to the pressing global health concerns, compelling hospitals and health agencies to strengthen their programs for preventing and treating kidney disease [4]. In Indonesia, there has been a notable rise in the prevalence of kidney disease, with a significant increase observed from 2013 to 2018. It now affects 3.8% of the total population [5]. Renal cell carcinoma, kidney cysts, and nefrolithiasis are some of the most prevalent disorders kidney that greatly affect both the individual's health and the overall health system.

Diagnosis of kidney disease involves a range of techniques and examination methods, including Computer Tomography (CT), Magnetic Resonance Imaging (MRI), X-rays, and B-Ultrasound. Renal CT scans are particularly notable for their capacity to generate three-dimensional slices that are valuable for analysing and reconstructing the targeted region [6]. Identifying abnormalities like cysts, stones, and tumours is crucial in order to prevent severe complications like kidney failure. This highlights the significance of early diagnosis [7]. Nevertheless, the lack of expertise and research in the field of artificial intelligence technology poses a significant obstacle to the diagnosis and treatment of kidney disease [8]. Traditional approaches in machine learning still have limitations, particularly in the process of feature extraction which heavily depends on manual feature selection [9] [10].

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

Deep learning, specifically Convolutional Neural Networks (CNNs), has emerged as the most powerful technique for tasks such as object classification, medical image processing, and feature extraction [11]. According to recent studies, CNN has proven to be highly effective in acquiring detailed data characteristics and outperforming other methods like DNN and ANN in terms of detection capabilities [12] [13]. CNN's superiority is evident in its remarkable accuracy in rice classification and medical imaging, surpassing other methods like ANN and SVM [14] [15]. Various CNN architectures, including ResNet, GoogleNet, and AlexNet, have made remarkable advancements in deep learning by enhancing their ability to extract complex features [16]. These advancements, such as the implementation of shortcut connections in ResNet to tackle the issue of vanishing gradients, have led to significant enhancements in image classification and analysis [17].

Research demonstrating the effectiveness of ResNet includes studies that conduct COVID-19 classification by comparing various CNN architectures. These architectures consist of ResNet, GoogleNet, VGG-16, AlexNet, MobileNet, and Inception-V3. The research findings indicate that the ResNet method achieved a sensitivity value of 100%, while GoogleNet, VGG-16, AlexNet, MobileNet, and Inception-V3 achieved sensitivity values respectively 96.6%, 96.6%, 98.3%, 93.3%, and 88.3% [18]. In the medical field, ResNet is widely used for identification purposes. For example, a recent study on the classification of colon cancer found that ResNet achieved an impressive accuracy of 95% [19]. A study on glaucoma eye disease classification utilised ResNet and achieved an accuracy of 95% [20]. In a separate study, ResNet was used for pneumonia diagnosis and achieved an impressive accuracy rate of 88.88% [21].

There are multiple models of ResNet that vary in terms of how their layers are arranged. The models in ResNet's lineup consist of ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, ResNet 164, and ResNet-202 [22] [23]. ResNet-50 stands out among other ResNet models due to its exceptional performance in image classification tasks because it excels at extracting top-notch features [24]. This advantage has been demonstrated through research that utilised X-ray images to detect COVID by comparing multiple ResNet models. The research found that ResNet-50 had the highest sensitivity value among the different ResNet models [25]. A study that diagnosed cervical cancer also conducted a comparison between ResNet-18, ResNet-50, and ResNet-101. The obtained results showed that ResNet-50 and ResNet-101 achieved 100% accuracy. However, it is worth noting that ResNet-101 had a slower computing time compared to ResNet-50 [26]. These studies suggest that ResNet-50 is a relevant architecture for image classification tasks, particularly in extracting visual features from medical images. However, its classification performance may vary depending on the dataset characteristics and experimental hyperparameter tuning.

In the context of kidney disease classification, research has been conducted to examine various deep learning architectures using CT-radiography images to classify four categories: normal, cyst, stone, and tumor. Their study evaluated ResNet-50, VGG16, InceptionV3, and several Vision Transformer architectures. The results showed that ResNet-50 achieved an accuracy of 73.80%, whereas the Swin Transformer obtained the highest accuracy of 99.30%. The study also employed Grad-CAM to provide visual explanations of the classification results [27]. Similarly, one study proposed a hybrid approach combining DenseNet201 for feature extraction and Random Forest for classification, reporting an accuracy of 99.44% as stated in their study [28]. Further studies have explored different CNN architectures and learning strategies using the publicly available CT Kidney Dataset. One study proposed KidneyNet, a customized CNN architecture, and compared its performance with VGG19, EfficientNet-B1, EfficientNet-B2, and Xception. Using five-fold cross-validation, KidneyNet achieved an average accuracy of 99.88% [29]. Other related studies have investigated fine-tuned deep learning models, including VGG16, ResNet50, CNN AlexNet, and InceptionV3, by integrating transfer learning, image processing, and hyperparameter optimization. Their findings showed that InceptionV3 achieved the highest accuracy of 99.96% [30]. More recently, one study proposed a privacy-preserving federated learning framework involving five pre-trained architectures distributed across seven federated clients. Their results indicated that MobileNetV2 achieved the best performance, with an accuracy of 99.48%, while the average accuracy across client validation was 98.85%. These studies demonstrate the potential

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

of the dataset for kidney image classification using different architectures, optimization strategies, and learning frameworks.

Considering these previous studies that has explored diverse CNN architectures and learning strategies using the same CT Kidney Dataset employed here, the present research focuses specifically on the experimental evaluation of ResNet-50 under different combinations of learning rate, dropout probability, and mini-batch size. Unlike studies that emphasize the development of customized CNN architectures, comparisons among multiple pre-trained models, or federated learning frameworks, this study examines the selection of an appropriate ResNet-50 configuration by considering both classification performance and computational time. Therefore, the contribution of this study lies in providing an empirical evaluation of the investigated ResNet-50 hyperparameter tuning, particularly in terms of classification performance and computational requirements.

Based on this research focus, this study aims to evaluate the performance of the CNN ResNet-50 architecture in classifying kidney CT-radiography images into four categories: normal, cyst, stone, and tumor. The experiments examine different combinations of learning rate, dropout probability, and mini-batch size to identify an appropriate model configuration. Model performance is evaluated using accuracy, sensitivity, and specificity through five-fold cross-validation, while computational time is also considered in selecting the optimal configuration. As a result, the development of this kidney disease classification system is projected to serve as an alternative for early detection of kidney disease.

2. RESEARCH METHODOLOGY

This research uses CT-Radiography kidney disease image data uploaded in 2022 as input. CT-Radiography renal disease picture data supplied in 2022 is used as input in this investigation. The Picture Archiving and Communication System (PACS) was used to collect CT-radiography kidney disease image data from multiple hospitals in Dhaka, Bangladesh. The total number of data points was 12,466 and was partitioned into four classes: Normal, Cyst, Kidney Stone, and Tumor. The Normal class had 5,077 data, the Kidney Stone class had 1,350 data, the Cyst class had 3,709 data, and the Tumor class had 2,283 data [27].

The undersampling technique was used in this study due to class imbalance in the dataset, specifically by reducing the number of image samples in the Normal, Cyst, and Tumor classes to the minimum class, which is Kidney Stones. The dataset's class imbalance was then resolved by randomly selecting 1350 images from every class except the Kidney Stone class [28]. The selection of undersampling in this study is supported by previous research using the same CT Kidney Dataset. In that prior studies randomly selected 1,300 images from each class to address class imbalance during model training [27]. Similarly, another studies reduced the number of images in the majority classes to match the minority class, resulting in a balanced dataset containing 1,350 images per class [28]. The effectiveness of undersampling has also been investigated in CNN-based medical image classification. That studies demonstrated that undersampling was effective in mitigating class imbalance in X-ray image classification, with a close-to-balanced training dataset achieving the best model performance [31]. Furthermore, other studies have reported that CNN models generally achieved better F-measure scores when trained using balanced image datasets compared to imbalanced datasets, particularly as the number of training epochs increased [32]. Based on these findings, the present study adopted undersampling by selecting 1,350 images from each class, resulting in a balanced dataset of 5,400 images for ResNet-50 classification. Figure 1 displays examples of images from each class.

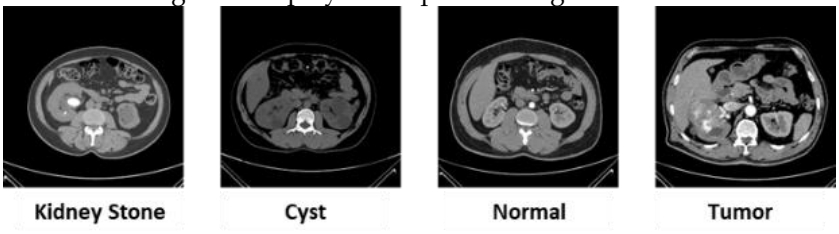


Figure 1. Research Data Sample

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

Figure 2 shows the stages performed for kidney disease classification using the Convolutional Neural Network (CNN) ResNet-50 method.

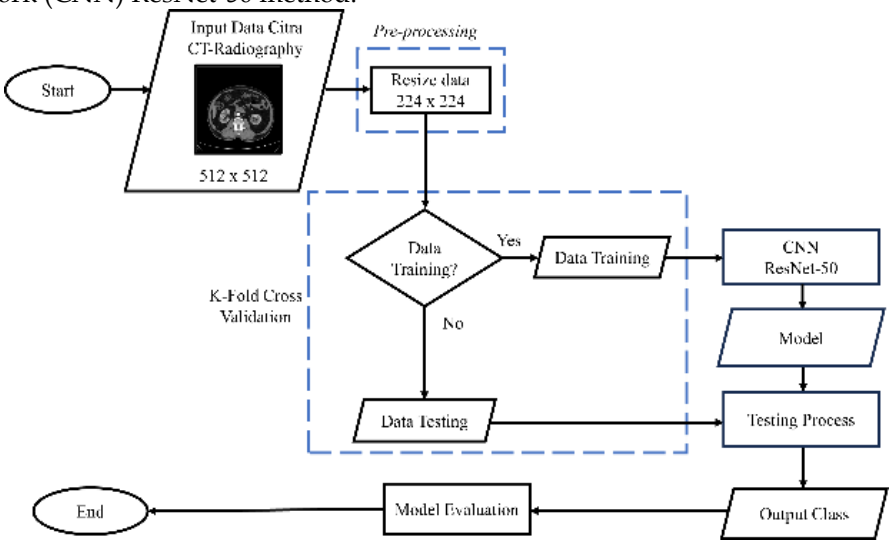


Figure 2. Research Methodology

Starting from the input of kidney disease image data obtained from various hospitals in Dhaka, Bangladesh through the Picture Archiving and Communication System (PACS). The contents of the data include images of Normal, Tumor, Kidney Stones, and Cysts. Then the preprocessing stage involves resizing the image data, which initially has a size of 512×512, to a size of 224×224 according to the ResNet method [33]. Then, the data splitting stage applies the K-Fold Cross Validation method, and the number of folds used is 5 folds. The training data is used for the training process conducted using the Convolutional Neural Network (CNN) architecture ResNet-50. During the training process, hyperparameter tuning experiments were conducted with learning rates of 0.1, 0.01, 0.001, 0.0001, dropout probabilities of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and mini batch size of 8, 16, 32, 64. The result of the training process is an optimal model that will be used in the testing process using the testing data, which then produces the classification output. Then, the model evaluation stage applies the multiclass confusion matrix method to obtain the parameter values of accuracy, sensitivity, and specificity.

According to the original dataset CT Kidney Dataset description, patient-related metadata were removed during the data preparation process, and the publicly distributed images were provided in JPG format [27]. Consequently, patient identifiers were unavailable in the dataset used in this study, making patient-level data partitioning infeasible. Therefore, the five-fold cross-validation procedure was performed at the image level rather than at the patient level.

2.1. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are the most common deep neural networks for visual image analysis [34]. CNN is a deep learning technique that uses picture input and improves Multi Layer Perceptron (MLP) for two-dimensional structured data [35] [36]. This approach can differentiate images and learn features by analysing them [37]. CNN architecture resembles brain nerve cells. CNN's design was inspired by the Visual Cortex, which helps it learn picture details. This is because the Visual Cortex processes visual information [38] [39]. CNN architecture is shown in Figure 3.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

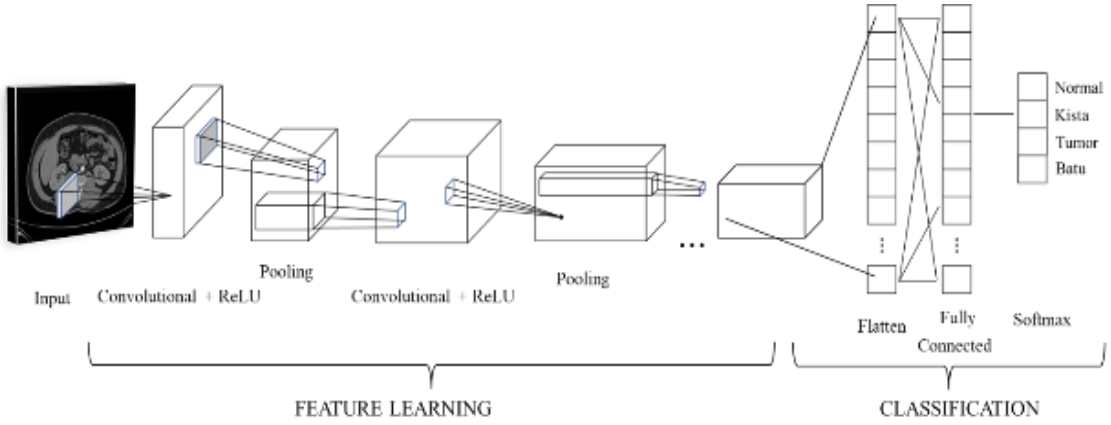


Figure 3. CNN Architecture

Layers in the CNN architecture allow the algorithm to work. The input layer, the first layer of the architecture, receives image data [40]. Next is the convolution layer, which contains the dot product multiplication between the input matrix obtained from the pixels in the image data and the filter or kernel in matrix form. Kernel size depends on architecture, and its contents are random values from -1 to 1. The CNN algorithm shifts the kernel over picture pixels by a number given by the stride parameter, which sets the number of horizontal or vertical steps. In addition to stride, padding adds 0 to each side of the image to keep the pixel matrix and kernel the same size when there are remainders [41]. Equation (1) calculates convolutional layer process [42].

$$Q_j = \sum_{i=1}^{N} (I_{i,j} \,.\, F_{i,j}) + B_j \tag{1}$$

Where Q_j is the result of the convolution operation, N is the number of channels, which typically in an RGB image consists of 3 channels. $I_{i,j}$ is the pixel value of the image represented as a matrix, $F_{i,j}$ is the kernel matrix, dan B_j is the bias.

The batch normalization layer is employed to improve system stability by normalizing the output of each layer and accelerating computation time [43] [44]. Batch normalization can be computed using Equations (2)-(4) [45] [46].

$$\mu = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i \tag{2}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (\bar{x}_i - \mu) \tag{3}$$

$$BN = (\gamma \frac{H - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta) \tag{4}$$

Where BN is the result of batch normalization, $\bar{x}_i$ is the average value of each channel, μ is the average of each mini batch size, σ^2 is the variance value of each mini batch size [46]. γ and β are learning parameters to measure the magnitude of the value produced by a channel within a layer, which is initialized randomly by taking samples from a normal distribution [47]. ϵ is a constant value to enhance numerical stability (generally 1× 10−3) [48].

Then the ReLU layer, which is one of the activation functions commonly applied in CNNs to speed up computation time, is mathematically shown in Equation (5).

$$f(x) = \max(0, x) \tag{5}$$

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

Down sampling reduces parameters in the pooling layer. The most common pooling algorithms are max and average. Max pooling uses the biggest feature map value [49]. Average pooling uses the feature map's average value [50].

The fully connected layer has numerous layers of neurones connected to all neurones in the previous layer [51]. A fully connected layer is necessary to determine the correlation between features and a specific class [52] [53]. Equation (6) calculates the fully connected layer [42].

$$y_j = b_j + \sum_{i=1}^{n} (w_{i,j} \, . \, x_i) \tag{6}$$

Where y_j is the output value from the fully connected layer j, j indicates the number of classes, b_j is the bias, $w_{i,j}$ is the network weight initialized randomly using the Glorot method that connects the input neuron i to the fully connected layer (output) neuron j, and x_i is the input from the feature learning process.

The softmax layer for normalizing the network [54]. The softmax layer generates probability values between 0 and 1 for each data point, facilitating classification based on relative class probabilities [55]. The softmax computation is presented in Equation (7) [54] [56].

$$p_j(\hat{y}) = \frac{e^{\hat{y}_j}}{\sum_{i=1}^{k} e^{\hat{y}_j}} \tag{7}$$

Where $p_j(\hat{y})$ is the probability value of class j, $\hat{y}_j$ is the value of class j, $e^{\hat{y}_j}$ is the exponential (base of the natural logarithm, approximately 2.718) of the value of class j, and k is the number of classes.

2.2. Residual Net (ResNet-50)

ResNet-50 is a variant of the Residual Network architecture of 50 layers, trained using at least one million image from the ImageNet database [57]. From 2010 to 2015, the ResNet model has the lowest error rate compared to other designs. In 2015, the ResNet architecture with the concept of skip connection or shortcut was established to solve the constraints of deep learning at specific layers [58].

ResNet employs shortcut connections to bypass one or more layers, effectively mitigating the vanishing gradient issue [59]. The employed structure is typically known as a Residual Block. The ResNet architecture has two varieties of Residual Blocks: the Identity Block and the Convolutional Block [45]. Identity Block is the standard Residual Block of ResNet that requires the input and output data to have the same dimensions [60]. The Identity Block employs the residual learning technique, as outlined in Equation (8).

$$y = F(x, Wi) + x \tag{8}$$

The Convolution Block is a variant of the Residual Block that closely resembles the Identity Block [61]. The Convolution Block employs the residual learning technique. However, in the shortcut connection it has, the Convolution Block adds a weight layer. The Convolution Block formula is generally represented in Equation (9).

$$y = F(x, Wi) + W_s x \tag{9}$$

Figure 2 illustrates the structure of the Identity Block on the left and the Convolution Block on the right. Both blocks have weight layers, ReLU activations, and addition layers. The weight layer is a convolution layer combined with batch normalization [62]. The addition layer is a mathematical operation that adds the output from the main connection (input that has passed through three weight layers) and the shortcut connection that occurs within the residual block [63].

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

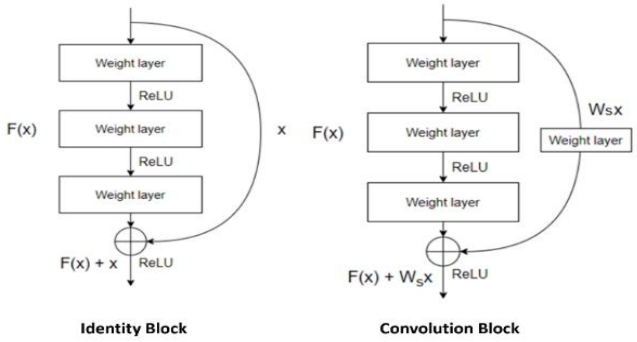


Figure 4. Residual Block

ResNet architecture resembles GoogleNet because the first two layers perform convolution processes with a kernel size of 7×7 and max pooling of 3×3 with a stride of 2 [64]. Figure 5 shows the ResNet-50 architecture.

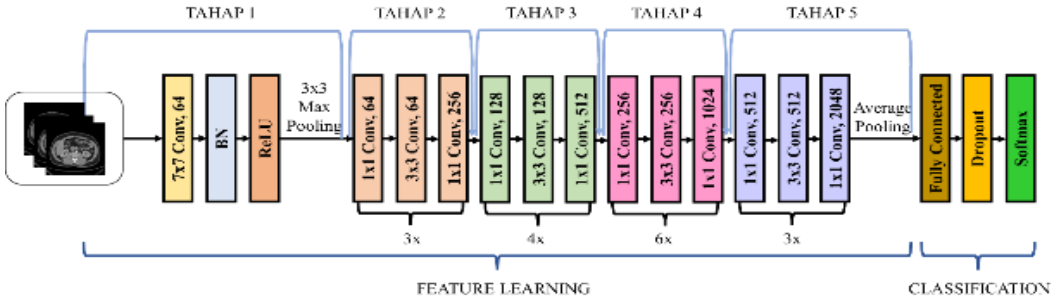


Figure 5. ResNet-50 Architecture

Feature learning in the ResNet-50 architecture commences at the initial stage, which comprises an input layer, convolution layer, batch normalization, ReLU activation function, and max-pooling layer, succeeded by two types of residual blocks: convolution blocks and identity blocks, amounting to a total of 16 blocks. Each block comprises three convolutional layers with shortcut connections, distributed across stages 2 to 5 [26]. The second stage comprises one convolution block and two identity blocks with designated kernel sizes. The third stage comprises one convolution block and three identity blocks with the designated kernel size. The fourth stage comprises one convolution block and five identity blocks, each with the designated kernel size. The fifth stage comprises one convolution block and two identity blocks with designated kernel sizes, thereafter succeeded by global average pooling to diminish the feature size at the conclusion of the fifth stage. A fully connected layer is employed for classification, followed by a dropout mechanism to mitigate overfitting, and ultimately, the output is processed through a softmax layer to get classification results [65] [66].

2.3. Confusion Matrix

The confusion matrix is a technique utilized in classification systems for the model evaluation [67]. The confusion matrix is structured as a N × N table, where N represents the number of classes [68]. The model's accuracy can be determined by utilizing the confusion matrix to compare actual values with predicted values [69]. Table 1 represents a multiclass confusion matrix.

Table 1. Multiclass Confusion Matrix

		Prediction Class			
	Class	**Normal**	**Cysts**	**Tumor**	**Kidney Stones**
Actual Class	**Normal**	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$
	Cysts	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$
	Tumor	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$
	Kidney Stones	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

The confusion matrix contains numerous terms, including True Positive (TP), which is the number of class i data properly predicted using Equation (10). Equation (11) calculates True Negative (TN), which is the number of class j data properly projected as class j. False Positive (FP) is the number of class j data mistakenly forecasted as class i data using Equation (12). False Negative (FN) is the number of class i data mistakenly forecasted as class j data using Equation (13).

$$TP_{all} = \sum_{j=1}^{n} x_{jj} \tag{10}$$

$$TN_i = \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq i}^{n} x_{jk} \tag{11}$$

$$FP_i = \sum_{j=1, j\neq i}^{n} x_{ji} \tag{12}$$

$$FN_i = \sum_{j=1, j\neq i}^{n} x_{ij} \tag{13}$$

The three evaluation parameters are calculated from TP, TN, FP, and FN. The confusion matrix evaluation test uses sensitivity, specificity, and accuracy, which may be calculated using Equations (14), (15), and (16). Sensitivity is the result of classifying kidney disease patients who are diagnosed as having kidney disease, specificity is the result of classifying someone who is not a kidney disease patient as not having kidney disease, and accuracy to the classification system's success.

$$Accuracy = \frac{TP_{all}}{n_{all}} \times 100\% \tag{14}$$

$$Sensitivity = \sum \frac{\frac{TP_{all}}{TP_{all} + FN_i}}{n} \times 100\% \tag{15}$$

$$Specificity = \sum \frac{\frac{TN_i}{TN_i + FN_i}}{n} \times 100\% \tag{16}$$

3. RESULT AND DISCUSSIONS

The kidney disease picture data used in this research is 512×512×3, while the ResNet-50 model requires 224×224×3 input images. In this study, image data must be resized from 512×512×3 to 224×224×3. Example of 224x224x3 preprocessing results is shown in Figure 6.

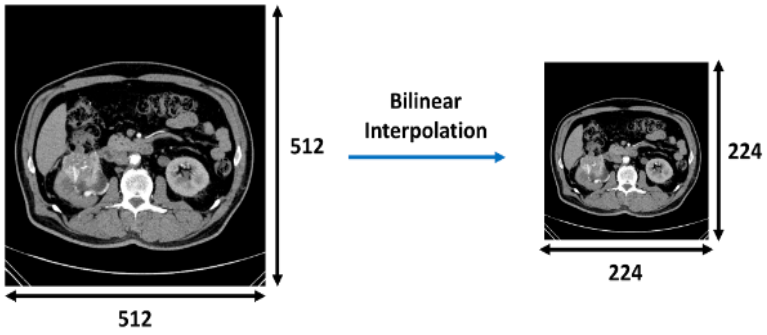


Figure 6. Visualization of Resize Results

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

K-Fold Cross Validation with K=5 divides 224x224x3-pixel picture data into training and testing data. The data partitioning was performed across five folds, in which each fold was used as testing data in a different iteration, while the remaining folds were used as training data [70] [71]. The training process was conducted using the Convolutional Neural Network (CNN) architecture ResNet-50. During training, several hyperparameter tuning were performed to obtain the optimal model configuration. The hyperparameters evaluated in this study consisted of learning rate, dropout probability, and mini-batch size. Learning rate is a hyperparameter that determines the step size taken when updating model weights during the training optimization process [72]. It also plays an important role in determining the convergence speed of the model [73]. Therefore, different learning rate values were evaluated to examine their effects on the model's classification performance and computational time. The learning rates examined in this study were 0.1, 0.01, 0.001, and 0.0001. Dropout probability is another hyperparameter examined to reduce overfitting during CNN training. Dropout is a regularization technique that randomly deactivates a number of neurons based on a predetermined probability ranging from 0 to 1 [74] [75]. This technique helps reduce network complexity and overfitting [76] [77]. In this study, dropout probabilities of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, and 0.7 were evaluated to investigate their effects on the classification performance of ResNet-50. Mini-batch size determines the number of data samples processed in each training iteration [78]. The use of mini-batch size allows the model to process a smaller group of samples during each iteration. Furthermore, selecting mini-batch sizes that are powers of two is related to GPU memory optimization [79]. Therefore, this study evaluated mini-batch sizes of 8, 16, 32, and 64. The number of epochs was set to 10 for all experiments. The resulting models from the training process were subsequently evaluated using the corresponding testing data in each fold. The classification outputs were then assessed using a multiclass confusion matrix to calculate accuracy, sensitivity, and specificity. The experimental results for each hyperparameter tuning are presented in the following tables.

Table 2. Dropout 0.1 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
0.1	8	99.53	99.55	99.52	18h 54m 34s
	16	99.54	99.53	99.53	17h 9m 13s
	32	98.24	98.27	98.27	19h 5m 53s
	64	88.14	88.15	88.82	19h 45m 40s
	Average	96.36	96.38	96.54	18h 43m 50s
0.01	8	100	100	100	18h 16m 46s
	16	100	100	100	15h 54m 35s
	32	99.98	99.97	99.97	16h 12m 9s
	64	98.74	98.75	98.93	24h 23m 36s
	Average	99.68	99.68	99.73	18h 41m 47s
0.001	8	100	100	100	18h 52m 52s
	16	100	100	100	17h 32m 19s
	32	100	100	100	15h 53m 34s
	64	100	100	100	16h 28m
	Average	100	100	100	17h 11m 41s
0.0001	8	100	100	100	18h 12s
	16	100	100	100	15h 51m 9s
	32	100	100	100	16h 58m 39s
	64	99.98	99.97	99.97	15h 44m 25s
	Average	99.99	99.99	99.99	16h 13m 6s

Table 3. Dropout 0.2 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	99.53	99.55	99.52	18h 3m 28s

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

	16	99.72	99.72	99.72	19h 4m 20s
0.1	32	98.52	98.52	98.50	15h 3m 56s
	64	90.72	90.73	90.72	25h 25m 54s
	Average	97.12	97.13	97.13	19h 24m 25s
	8	99.96	99.95	99.95	18h 17m 43s
	16	100	100	100	16h 22m 50s
0.01	32	99.98	99.97	99.97	15h 4m 57s
	64	99.40	99.40	99.40	28h 2m 41s
	Average	99,84	99,83	99,83	19h 6m 13s
	8	100	100	100	17h 58m 23s
	16	100	100	100	16h 7m 20s
0.001	32	100	100	100	14h 51m 18s
	64	100	100	100	23h 29m 54s
	Average	100	100	100	18h 6m 44s
	8	100	100	100	17h 32m 20s
	16	100	100	100	15h 53m 43s
0.0001	32	100	100	100	14h 49m 51s
	64	99.98	99.97	99.97	20h 3m 21s
	Average	99.99	99.99	99.99	17h 4m 49s

Table 4. Dropout 0.3 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	98.26	98.25	98.27	18h 17m 13s
	16	98.77	98.77	98.80	16h 6m 55s
0.1	32	98.94	98.95	98.95	15h 16m 32s
	64	97.74	97.72	97.75	21h 43m 14s
	Average	96.36	96.38	96.54	17h 50m 59s
	8	99.96	99.95	99.95	18h 46m 12s
	16	99.96	99.95	99.95	15h 13m 45s
0.01	32	99.98	99.97	99.97	15h 21m 12s
	64	98.27	98.27	98.27	37h 1m 35s
	Average	99.79	99.78	99.78	21h 35m 41s
	8	100	100	100	20h 38m 2s
	16	100	100	100	16h 25s
0.001	32	100	100	100	15h 10m 39s
	64	100	100	100	19h 30m 51s
	Average	100	100	100	17h 49m 59s
	8	100	100	100	18h 16m 50s
	16	100	100	100	16h 4m 10s
0.0001	32	100	100	100	17h 45m 45s
	64	99.88	99.87	99.87	21h 48m 30s
	Average	99.97	99.96	99.96	18h 28m 49s

Table 5. Dropout 0.4 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	90.31	90.32	91.67	19h 5m 55s
	16	99.63	99.62	99.62	19h 7m 40s
0.1	32	98.92	98.92	98.92	14h 56m 21s
	64	97.94	97.95	97.95	21h 28m 26s
	Average	96.70	96.38	97.04	18h 39m 36s

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

	8	99.96	99.95	99.95	17h 26m 37s
	16	100	100	100	16h 43m 52s
0.01	32	99.96	99.95	99.95	15h 11m 55s
	64	99.98	99.97	99.97	19h 59m 10s
	Average	99.97	99.97	99.97	17h 20m 24s
	8	100	100	100	18h 49m 18s
	16	100	100	100	15h 52m 30s
0.001	32	100	100	100	16h 31m 4s
	64	100	100	100	19h 43m 5s
	Average	100	100	100	17h 44m
	8	100	100	100	18h 10m 6s
	16	100	100	100	16h 46m 3s
0.0001	32	100	100	100	15h 59m 43s
	64	99.98	99.97	99.97	18h 25m 22s
	Average	99.99	99.99	99.99	16h 38m 39s

Table 6. Dropout 0.5 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	98.24	98.25	98.27	18h 22m 7s
	16	98.77	98.80	98.80	15h 56m 54s
0.1	32	99.37	99.40	99.37	14h 58m 2s
	64	90.72	90.73	90.78	37h 53m 20s
	Average	96.77	96.79	96.80	21h 47m 36s
	8	99.98	99.97	99.97	17h 49m 26s
	16	99.98	99.97	99.97	24h 24m 10s
0.01	32	99.98	99.97	99.97	15h 33m 43s
	64	99.96	99.95	99.95	44h 12m 49s
	Average	99.97	99.96	99.96	25h 30m 2s
	8	100	100	100	17h 43m 37s
	16	100	100	100	16h 10m 45s
0.001	32	100	100	100	14h 59m 14s
	64	100	100	100	23h 5m 10s
	Average	100	100	100	17h 59m 42s
	8	100	100	100	18h 40m 26s
	16	100	100	100	15h 37m 53s
0.0001	32	100	100	100	16h 35m 15s
	64	99.98	99.97	99.97	22h 30m 7s
	Average	99.99	99.99	99.99	18h 20m 55s

Table 7. Dropout 0.6 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	98.24	98.25	98.27	17h 28m 49s
	16	98.87	98. 87	98. 85	16h 17m 42s
0.1	32	99.37	99.40	99. 37	14h 53m 15s
	64	97.74	97.72	97.75	23h 57m 51s
	Average	98.55	98.56	98.56	18h 9m 24s
	8	99.98	99.97	99.97	18h 12m 43s
	16	99.98	99.97	99.97	15h 33m 8s
0.01	32	100	100	100	15h 18m 25s
	64	99.70	99.72	99.70	25h 43m 42s

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

	Average	99.91	99.91	99.91	18h 42m
	8	100	100	100	17h 56m 39s
	16	100	100	100	16h 4m 47s
0.001	32	100	100	100	16h 53m 6s
	64	100	100	100	19h 56m 44s
	Average	100	100	100	17h 42m 49s
	8	100	100	100	20h 38m 17s
	16	100	100	100	15h 47m 6s
0.0001	32	100	100	100	15h 56m 33s
	64	99.98	99.97	99.97	18h 53m 3s
	Average	99.99	99.99	99.99	17h 48m 45s

Table 8. Dropout 0.7 Hyperparameter Tuning Result

Learning Rate	Mini Batch Size	Accuracy (%)	Sensitivity (%)	Specificity (%)	Computational Duration
	8	99.33	99.35	99.35	18h 45m 18s
	16	98.63	98.60	98.62	16h 12m 36s
0.1	32	97.40	97.40	97.55	15h 28m 55s
	64	97.94	97.95	97.95	22h 27m 56s
	Average	98.33	98.33	98.37	18h 55m 21s
	8	99.96	99.95	99.95	23h 21m 53s
	16	99.98	99.97	99.97	17h 2m 19s
0.01	32	99.98	99.97	99.97	15h 11m 17s
	64	99.94	99.95	99.95	23h 26m 27s
	Average	99.97	99.96	99.96	19h 45m 29s
	8	100	100	100	19h 50m 21s
	16	100	100	100	15h 38m 27s
0.001	32	100	100	100	17h 27m 58s
	64	100	100	100	17h 36m 59s
	Average	100	100	100	17h 38m 26s
	8	100	100	100	18h 37m 18s
	16	100	100	100	15h 34m 9s
0.0001	32	100	100	100	15h 9m 48s
	64	99.98	99.97	99.97	20h 15m 50s
	Average	99.99	99.99	99.99	17h 24m 16s

Based on Tables 2–8, the best average classification performance was achieved at a learning rate of 0.001 across all dropout probabilities, from 0.1 to 0.7, with accuracy, specificity, and sensitivity values of 100% for each configuration. However, the lowest computational time was obtained at a dropout probability of 0.1, with an average computational time of 17 hours, 11 minutes, and 41 seconds. Table 2 presents the experimental results for a dropout probability of 0.1 and a learning rate of 0.001. The results show that mini-batch sizes of 8, 16, 32, and 64 produced identical accuracy, sensitivity, and specificity values, each reaching 100%. However, a mini-batch size of 32 required the shortest computational time of 15 hours, 53 minutes, and 34 seconds, representing the total computational time across all five folds. Table 9 presents the experimental results using a learning rate of 0.001, a dropout probability of 0.1, and a mini-batch size of 32 for each fold.

Table 9. Optimal model results: Learning rate 0.001, Dropout Probability 0.1, Mini Batch Size 32 on K-Fold Cross Validation

Fold	Accuracy	Sensitivity	Specificity	Time
1	100%	100%	100%	187m 20s
2	100%	100%	100%	189m 11s

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

3	100%	100%	100%	191m 44s
4	100%	100%	100%	191m 43s
5	100%	100%	100%	193m 36s

Table 9 shows that accuracy, sensitivity, and specificity reached 100% across all folds. However, the first fold required the shortest computational time compared to the other folds, at 187 minutes and 20 seconds. The model training process for this fold is illustrated in Figure 7.

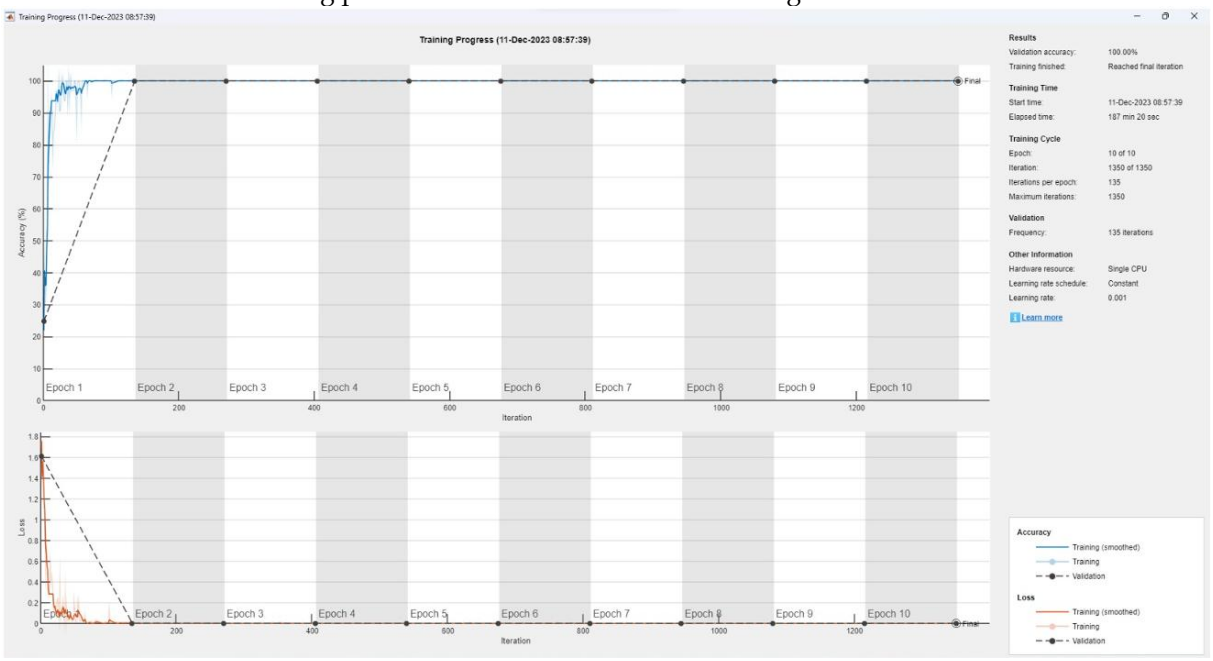


Figure 7. Optimal Model Training Process Graph

The hyperparameter tuning experiments achieved an accuracy of 100%, sensitivity of 100%, and specificity of 100%, with these values obtained from the calculation of the confusion matrix using Equations (14)-(16) from the multiclass confusion matrix table shown in Figure 8.

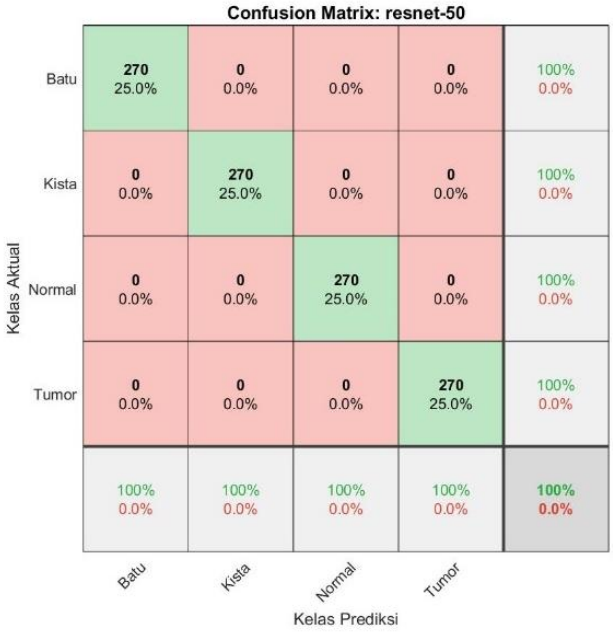


Figure 8. Optimal Model Confusion Matrix

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

Table 10 presents a comparison of the system evaluation results with prior research.

Table 10. Comparison of System Evaluation Results with Previous Research

	Method	Accuracy
This research	ResNet-50	100%
Islam et al., 2022 [27]	ResNet-50 (without hyperparameter tuning)	73,80%
	VGG-16	98,20%
	Inception v3	61,60%
Qadir and Abd, 2023 [28]	DenseNet-201-Random Forest	99,44%
Almuayqil et al., 2024 [29]	KidneyNet	99.88%
Pimpalkar et al., 2025 [30]	InceptionV3	99.96%
Konala and Koppu, 2026 [80]	MobileNetV2	99.48%

The kidney disease diagnostic system is developed using a Graphical User Interface to facilitate the automatic classification of kidney disease kinds. The classification system use the optimal model derived from the model training process. This program is designed to classify kidney disease by kind, including normal, tumor, kidney stones, and cysts. This diagnostic system comprises two pages: the main page and the classification display page. Figure 9 shows the main page of the classification system.



Figure 9. Main Page of The Kidney Disease Diagnosis System

On the main page, there is a "Mulai" button that when clicked will leads to the second page shown in Figure 10.

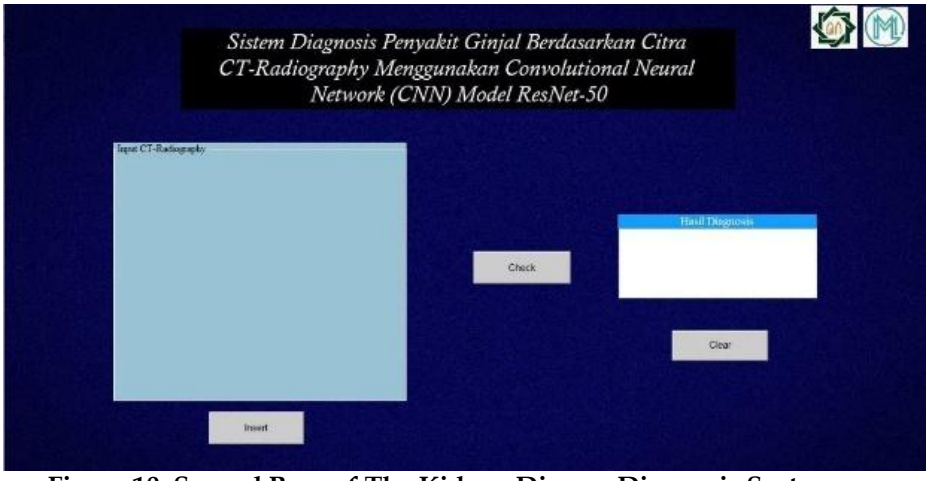


Figure 10. Second Page of The Kidney Disease Diagnosis System

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

On the second page, there are buttons labeled "Insert," "Check," and "Clear." The "Insert" button opens the selected CT-Radiography image data file, the "Check" button initiates the classification process and displays the diagnosis results for the selected image, and the "Clear" button removes the diagnosis results and image data. Figure 11 shows the display of the file containing image data upon clicking the "Insert" button.



Figure 11. Second Page of The Kidney Disease Diagnosis System When The "Insert" Button is Clicked

Figure 12 shows that the selected image data is from the Tumor class.

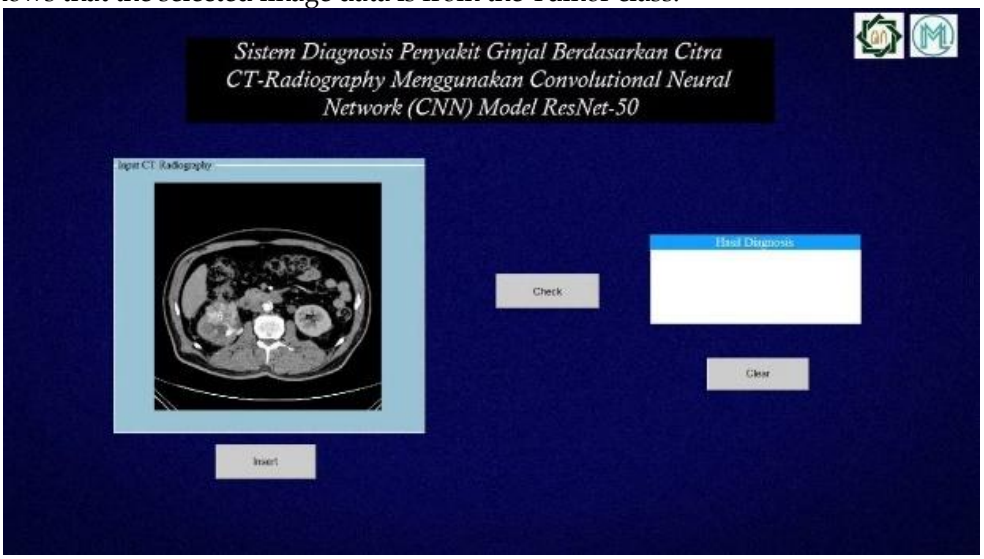


Figure 12. Second Page of The Kidney Disease Diagnosis System when The Tumor Image Data is Selected

Figure 13 shows that when the "Check" button is clicked, the diagnosis result obtained is Tumor, which means the system diagnoses the image accurately.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

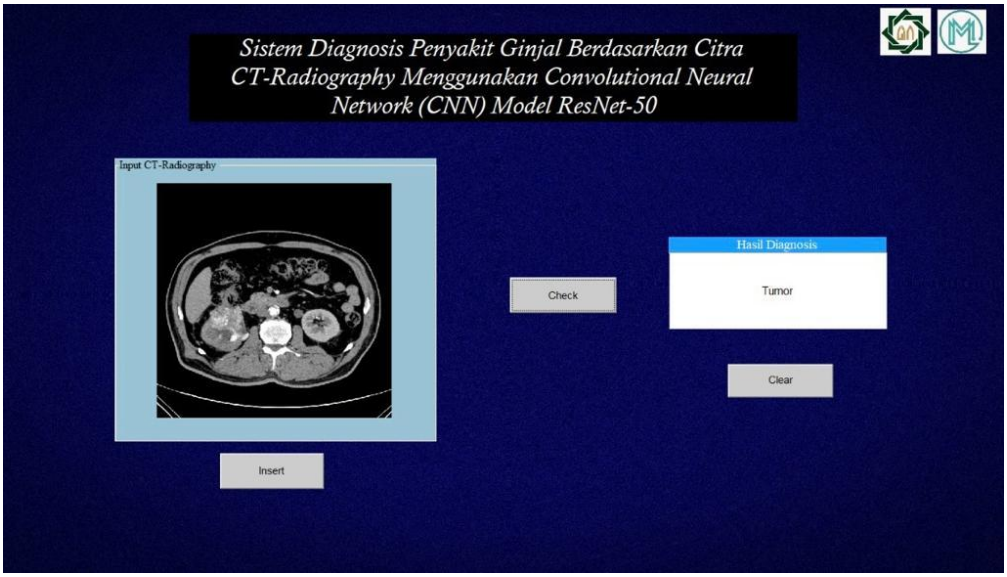


Figure 13. Second Page of The Kidney Disease Diagnosis System when The "Check" Button is Clicked

If the "Clear" button is clicked, the image input display and the automatic diagnosis result will be reset, and the display will return to the state shown in Figure 10.

4. CONCLUSION

ResNet-50, a convolutional neural network architecture, was applied to classify kidney diseases based on CT-radiography images. This study demonstrated that the model could identify visual patterns associated with normal kidneys, cysts, kidney stones, and tumors. Optimal results were achieved with hyperparameter tuning experiments of a learning rate of 0.001, a dropout rate of 0.1, and a mini-batch size of 32. In this combination, the model achieved perfect classification metrics with accuracy, sensitivity, and specificity each reaching 100% on the testing dataset, signifying that it classified the full test data flawlessly without any errors. The model training process necessitated a computation duration of 15 hours, 53 minutes, and 34 seconds, indicative of the network's complexity and the data volume processed. However, these results should be interpreted with caution because the publicly available dataset did not provide patient identifiers. Consequently, data partitioning was performed at the image level rather than the patient level, and potential information leakage between training and testing data cannot be entirely ruled out. Therefore, the reported performance does not yet establish the model's generalization capability to independent patients or its clinical diagnostic reliability. Nevertheless, ResNet-50 demonstrates potential as an image-based classification support system for preliminary kidney disease screening. Further validation using independent patient-level datasets and clinical assessment is necessary before its application in clinical practice can be established.

CRediT AUTHORSHIP CONTRIBUTION STATEMENT

Icha Dwi Safira: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Conceptualization. **Dian Candra Rini Novitasari**: Writing – original draft, Formal analysis, Data curation, Conceptualization.

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.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Journal of Artificial Intelligence and Technology Information
Website : https://ejournal.techcart-press.com/index.php/jaiti
E-ISSN 2985-6396
P-ISSN 2985-5306
Volume 4, Number 4, December 2026
https://doi.org/10.58602/jaiti.v4i4.458
Page 863-882

FUNDING STATEMENTS

The author(s) received no financial support for the research, authorship, and/or publication of this article.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

The authors used ChatGPT (GPT-6, OpenAI) to improve the writing clarity and language of this paper. The authors reviewed and edited the AI-assisted content and take full responsibility for the final publication.

REFERENCES

[1] C. P. Kovesdy, "Epidemiology of Chronic Kidney Disease: An Update 2022," *Kidney Int. Suppl. Vol. 12, Issue 1, April 2022, Pages 7-11*, vol. 12, no. 1, pp. 7–11, 2022.

[2] K. J. Foreman *et al.*, "Forecasting Life Expectancy, Years of Life Lost, and All-Cause and Cause-Specific Mortality for 250 Causes of Death: Reference and Alternative Scenarios for 2016–40 for 195 Countries and Territories," *Lancet*, vol. 392, no. 10159, pp. 2052–2090, 2018, doi: 10.1016/S0140-6736(18)31694-5.

[3] S. Safiri *et al.*, "The burden of kidney cancer and its attributable risk factors in 195 countries and territories, 1990–2017," *Sci. Rep.*, vol. 10, no. 1, pp. 1–20, 2020, doi: 10.1038/s41598-020-70840-2.

[4] P. F. Wiliyanarti and A. Muhith, "Life Experience of Chronic Kidney Diseases Undergoing Hemodialysis Therapy," *J. NurseLine*, vol. 4, no. 1, pp. 54–60, 2019.

[5] I. F. Jaya, "Edukasi Pengetahuan Pembatasan Cairan pada Pasien Gagal Ginjal Kronik yang Menjalani Hemodialisis," *Indones. J. Community Serv.*, vol. 3, no. 2, pp. 61–68, 2023, [Online]. Available: http://ijocs.rcipublisher.org/index.php/ijocs/article/view/254/195

[6] A. A. Salimi, "Simulasi Finite Element Analysis (FEA) dan Fabrikasi Implan untuk Perbandingan Kinerja Implan Kovensional dan Custom Cruciate Retaining (CR) untuk penderita Osteoarthritis Berdasarkan Rekonstruksi Data MRI/CT Scan Tulang Lutut," Universitas Islam Indonesia, 2023.

[7] Y. Widjaja *et al.*, "Peningkatan Kewaspadaan Masyarakat Terhadap Penyakit Ginjal Kronis dengan Edukasi Gaya Hidup dan Skrining Fungsi Ginjal," *Communnity Dev. J.*, vol. 4, no. 6, pp. 12147–12153, 2023.

[8] J. Sun *et al.*, "Performance of a Chest Radiograph AI Diagnostic Tool for COVID-19: A Prospective Observational Study," *Radiol. Artif. Intell.*, vol. 4, no. 4, 2022, doi: 10.1148/ryai.210217.

[9] M. M. Ahsan, S. A. Luna, and Z. Siddique, "Machine-Learning-Based Disease Diagnosis : A Comprehensive Review," *Healthcare*, pp. 1–30, 2022.

[10] S. K. Wildah, S. Agustiani, A. Mustopa, N. Wuryani, H. M. Nawawi, and R. A. Safitri, "Pengenalan Wajah Menggunakan Pembelajaran Mesin Berdasarkan Ekstraksi Fitur Pada Gambar Wajah Berkualitas Rendah," *INFOTECH J. Inform. dan Teknol.*, vol. 2, no. 2, pp. 95–103, 2021, doi: 10.37373/infotech.v2i2.189.

[11] R. Tekade and K. Rajeswari, "Lung Cancer Detection and Classification using Deep Learning," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–5.

[12] W. Wang *et al.*, "Detection of SARS-CoV-2 in Different Types of Clinical Specimens," *JAMA - J. Am. Med. Assoc.*, 2020, doi: 10.1001/jama.2020.3786.

[13] B. Feng, H. Zhou, G. Li, Y. Zhang, K. Sood, and S. Yu, "Enabling Machine Learning with Service Function Chaining for Security Enhancement at 5G Edges," *IEEE Netw.*, vol. 35, pp. 196–201, 2021.

[14] E. Suherman, D. Hindarto, A. Makmur, and H. Santoso, "Comparison of Convolutional Neural Network and Artificial Neural Network for Rice Detection," *Sinkron*, vol. 8, no. 1, pp. 247–255, 2023, doi: 10.33395/sinkron.v8i1.11944.

[15] T. Maruyama *et al.*, "Comparison Of Medical Image Classification Accuracy Among Three Machine Learning Methods," *J. Xray. Sci. Technol.*, vol. 26, no. 6, pp. 885–893, 2018.

[16] S. Naz, K. Bibi, and R. Ahmad, "DeepSignature: Fine-Tuned Transfer Learning Based Signature Verification System," *Multimed. Tools Appl.*, vol. 81, pp. 38113–38122, 2022, doi: 10.1007/s11042-022-12782-7.

[17] C. Jianfei and Z. Changming, "Research on Image Recognition Based on Improved ResNet," in *2020 IEEE 6th International Conference on Computer and Communications*, 2021, pp. 1422–1426. doi: 10.1088/1742-6596/1982/1/012011.

[18] U. Maheen, K. I. Malik, and G. Ali, "Comparative Analysis of Deep Learning Algorithms for Classification of COVID-19 X-Ray Images," *Int. J. Comput. Sci. Trends Technol.*, vol. 9, no. 4, pp. 73–81, 2021, [Online].

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Available: www.ijcstjournal.org
[19] V. G. Krishnan, P. A. A. Saleem, K. Sathyamoorthy, K. H. Priya, and T. K. S. Kumar, "Colorectal Cancer Prediction using ResNet-CNN Classification Method," *Int. J. Adv. Soft Comput. its Appl.*, vol. 15, no. 2, pp. 54–66, 2023, doi: 10.15849/IJASCA.230720.04.
[20] S. Sumitha and S. Gokila, "Experimental Approach to Identify the Optimal Deep CNN Models to Early Detection of Glaucoma from Fundus CT-Scan Images," *2023 7th Int. Conf. Intell. Comput. Control Syst.*, pp. 511–515, 2023.
[21] S. A. Rachman, D. C. Bagaskara, R. Magdalena, and S. Sa'idah, "Classification of Pneumonia Based on X-Ray Images with ResNet-50 Architecture," 2023.
[22] M. Chhabra and R. Kumar, "An Efficient ResNet-50 based Intelligent Deep Learning Model to Predict Pneumonia from Medical Images," *2022 Int. Conf. Sustain. Comput. Data Commun. Syst.*, pp. 1714–1721, 2022, doi: 10.1109/ICSCDS53736.2022.9760995.
[23] S. Sukegawa *et al.*, "Multi-Task Deep Learning Model for Classification of Dental Implant Brand and Treatment Stage using Dental Panoramic Radiograph Images," *Biomolecules*, vol. 11, no. 6, 2021, doi: 10.3390/biom11060815.
[24] L. Wen, X. Li, and L. Gao, "A Transfer Convolutional Neural Network for Fault Diagnosis Based on ResNet-50," *Neural Comput. Appl.*, vol. 32, pp. 6111–6124, 2020, doi: 10.1007/s00521-019-04097-w.
[25] S. Showkat and S. Qureshi, "Efficacy of Transfer Learning-based ResNet Models in Chest X-ray Image Classification for Detecting COVID-19 Pneumonia," *Chemom. Intell. Lab. Syst.*, vol. 224, p. 104534, 2022, doi: 10.1016/j.chemolab.2022.104534.
[26] D. C. R. Novitasari, P. Wulandari, and D. Z. Haq, "Cervical Cancer Diagnosis System using Convolutional Neural Network ResidualNet," *Int. J. Comput.*, vol. 21, no. 1, pp. 61–68, 2022, doi: 10.47839/ijc.21.1.2518.
[27] M. N. Islam, M. Hasan, M. K. Hossain, M. G. R. Alam, M. Z. Uddin, and A. Soylu, "Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography," *Sci. Rep.*, vol. 12, no. 11440, pp. 1–14, 2022, doi: 10.1038/s41598-022-15634-4.
[28] A. M. Qadir and D. F. Abd, "Kidney Diseases Classification using Hybrid Transfer-Learning DenseNet201-Based and Random Forest Classifier," *Kurdistan J. Appl. Res.*, no. March, pp. 131–144, 2023, doi: 10.24017/science.2022.2.11.
[29] S. N. Almuayqil, S. A. El-Ghany, A. A. A. El-Aziz, and M. Elmogy, "KidneyNet: A Novel CNN-Based Technique for the Automated Diagnosis of Chronic Kidney Diseases from CT Scans," *Electronics*, vol. 13, no. 24, 2024, doi: 10.3390/electronics13244981.
[30] A. Pimpalkar, D. K. J. B. Saini, N. Shelke, A. Balodi, G. Rapate, and M. Tolani, "Fine-tuned deep learning models for early detection and classification of kidney conditions in CT imaging," *Sci. Rep.*, vol. 15, no. 10741, 2025, doi: 10.1038/s41598-025-94905-2.
[31] W. Qu, I. Balki, M. Mendez, J. Valen, J. Levman, and P. N. Tyrrell, "Assessing and mitigating the effects of class imbalance in machine learning with application to X-ray imaging," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, pp. 2041–2048, 2020, doi: 10.1007/s11548-020-02260-6.
[32] W. Albattah and R. U. Khan, "Impact of imbalanced features on large datasets," *Front. Big Data*, vol. 8, no. 1455442, 2025, doi: 10.3389/fdata.2025.1455442.
[33] S. A. A. Ismael, A. Mohammed, and H. Hefny, "An Enhanced Deep Learning Approach for Brain Cancer MRI Images Classification using Residual Networks," *Artif. Intell. Med.*, vol. 102, pp. 1–14, 2019, doi: 10.1016/j.artmed.2019.101779.
[34] M. S. Khan, N. Salsabil, M. G. R. Alam, M. A. A. Dewan, and M. Z. Uddin, "CNN-XGBoost Fusion-Based Affective State Recognition using EEG Spectrogram Image Analysis," *Sci. Rep.*, vol. 12, no. 1, pp. 1–19, 2022, doi: 10.1038/s41598-022-18257-x.
[35] B. W. Zan, Z. H. Han, C. Z. Xu, M. Q. Liu, and W. Z. Wang, "High-Dimensional Aerodynamic Data Modeling Using a Machine Learning Method Based on a Convolutional Neural Network," *Adv. Aerodyn.*, vol. 4, no. 39, pp. 1–31, 2022, doi: 10.1186/s42774-022-00128-8.
[36] Y. Kaya and E. Gürsoy, "A Novel Multi-Head CNN Design to Identify Plant Diseases using The Fusion of RGB Images," *Ecol. Inform.*, vol. 75, p. 101998, 2023.
[37] S. Das, K. Kharbanda, S. M, R. Raman, and D. E. D, "Deep Learning Architecture Based On Segmented Fundus Image Features for Classification of Diabetic Retinopathy," *Biomed. Signal Process. Control*, vol. 68, p. 102600, 2021.
[38] T. D. Tran, M. Kimura, and Y. Nakashima, "Primary Visual Cortex Inspired Feature Extraction Hardware Model," in *2020 4th International Conference on Recent Advances in Signal Processing, Telecommunications and Computing, SigTelCom 2020*, 2020, pp. 20–24. doi: 10.1109/SigTelCom49868.2020.9199057.
[39] S. Monaco, G. Malfatti, J. C. Culham, L. Cattaneo, and L. Turella, "Decoding Motor Imagery and Action Planning in The Early Visual Cortex: Overlapping but Distinct Neural Mechanisms," *Neuroimage*, vol. 218,

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

p. 116981, 2020, doi: 10.1016/j.neuroimage.2020.116981.
[40] K. Liu, G. Kang, N. Zhang, and B. Hou, "Breast Cancer Classification Based on Fully-Connected Layer First Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 23722–23732, 2018, doi: 10.1109/ACCESS.2018.2817593.
[41] J. V. N. Rakamawati, "Klasifikasi Diabetic Retinopathy Berdasarkan Foto Fundus Menggunakan Convolutional Neural Network (CNN) Jenis Densenet," UIN Sunan Ampel Surabaya, 2021.
[42] I. Wulandari, H. Yasin, and T. Widiharih, "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (CNN)," *J. Gaussian*, vol. 9, no. 3, pp. 273–282, 2020, doi: 10.14710/j.gauss.v9i3.27416.
[43] M. A. Bilal, Y. Ji, Y. Wang, M. P. Akhter, and M. Yaqub, "Early Earthquake Detection Using Batch Normalization Graph Convolutional Neural Network (BNGCNN)," *Appl. Sci.*, vol. 12, no. 7548, 2022, doi: 10.3390/app12157548.
[44] Y. Seo and K. shik Shin, "Hierarchical Convolutional Neural Networks for Fashion Image Classification," *Expert Syst. Appl.*, vol. 116, pp. 328–339, 2019, doi: 10.1016/j.eswa.2018.09.022.
[45] S. Pang, S. Qiao, T. Song, J. Zhao, and P. Zheng, "An Improved Convolutional Network Architecture Based on Residual Modeling for Person Re-Identification in Edge Computing," *IEEE Access*, vol. 7, pp. 106749–106760, 2019, doi: 10.1109/ACCESS.2019.2933364.
[46] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Int. Conf. Mach. Learn.*, pp. 448–456, 2015.
[47] H. Zebua, "Pembuatan Model Deteksi Gejala Awal Penyakit Mulut dan Kuku pada Sapi Berbasis Citra Menggunakan Metode Convolutional Neural Network," Universitas Jambi, 2023.
[48] P. M. Allport, *Applications of Fault-Tolerant Software Architecture Principles in The Detection of Adversarial Attacks*. 2023.
[49] M. Yani, B. Irawan, and C. Setiningsih, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail," in *Journal of Physics: Conference Series*, 2019, vol. 1201, no. 1, p. 012052. doi: 10.1088/1742-6596/1201/1/012052.
[50] T. Y. Hsiao, Y. C. Chang, H. H. Chou, and C. Te Chiu, "Filter-Based Deep-Compression with Global Average Pooling for Convolutional Networks," *J. Syst. Archit.*, vol. 95, pp. 9–18, 2019, doi: 10.1016/j.sysarc.2019.02.008.
[51] D. C. R. Novitasari *et al.*, "Detection of COVID-19 Chest X-Ray Using Support Vector Machine and Convolutional Neural Network," *Commun. Math. Biol. Neurosci.*, vol. 42, pp. 1–19, 2020, doi: 10.28919/cmbn/4765.
[52] I. K. Trisiawan and Y. Yuliza, "Penerapan Multi-Label Image Classification Menggunakan Metode Convolutional Neural Network (CNN) Untuk Sortir Botol Minuman," *J. Teknol. Elektro*, vol. 13, no. 1, p. 48, 2022, doi: 10.22441/jte.2022.v13i1.009.
[53] A. Alqumboz and M. Naser, "Classification of Avocado Using Deep Learning," *Int. J. Acad. Eng. Res.*, vol. 3, no. 12, pp. 30–34, 2019, [Online]. Available: http://dspace.alazhar.edu.ps/xmlui/handle/123456789/563
[54] I. Kouretas and V. Paliouras, "Simplified Hardware Implementation of the Softmax Activation Function," in *8th International Conference on Modern Circuits and Systems Technologies, MOCAST 2019*, 2019, pp. 1–4. doi: 10.1109/MOCAST.2019.8741677.
[55] M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach, "Beyond Temperature Scaling: Obtaining Well-Calibrated Multiclass Probabilities with Dirichlet Calibration," 2019.
[56] R. Fachmi, A. Hidayatno, and Y. A. A. Soetrisno, "Sistem Identifikasi Ukuran Tubuh Menggunakan Metode Convolutional Neural Network (CNN)," *Transient J. Ilm. Tek. Elektro*, vol. 9, no. 1, pp. 1–7, 2020, doi: 10.14710/transient.v9i1.1-7.
[57] A. V. Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 375–381, 2021, doi: 10.1016/j.gltp.2021.08.027.
[58] S. Raman, R. Maskeliūnas, and R. Damaševičius, "Markerless Dog Pose Recognition in The Wild using Resnet Deep Learning Model," *Computers*, vol. 11, no. 2, pp. 1–12, 2021, doi: 10.3390/computers11010002.
[59] A. Negi, K. Kumar, N. S. Chaudhari, N. Singh, and P. Chauhan, "Predictive Analytics for Recognizing Human Activities Using Residual Network and Fine-Tuning," *Big Data Anal.*, vol. 13147, 2021, doi: 10.1007/978-3-030-93620-4_21.
[60] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," *Procedia Comput. Sci.*, vol. 179, pp. 423–431, 2021, doi: 10.1016/j.procs.2021.01.025.
[61] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and M. S. El-Mahallawy, "An Efficient Brain Tumor Image

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/

Segmentation Based on Deep Residual Networks (ResNets)," *J. King Saud Univ. - Eng. Sci.*, vol. 33, pp. 404–412, 2021, doi: 10.1016/j.jksues.2020.06.001.

[62] B. Yang *et al.*, "Identification of Species by Combining Molecular and Morphological Data Using Convolutional Neural Networks," *Syst. Biol.*, vol. 71, no. 3, pp. 690–705, 2022, doi: 10.1093/sysbio/syab076.

[63] M. H. A. Jihaannuriy, "Pengenalan Pelat Nomor Otomatis Menggunakan Metode Inception-ResNet v2," UIN Syarif Hidayatullah Jakarta, 2023.

[64] S. Wan, Y. Liang, and Y. Zhang, "Deep Convolutional Neural Networks for Diabetic Retinopathy Detection by Image Classification," *Comput. Electr. Eng.*, vol. 72, pp. 274–282, 2018, doi: 10.1016/j.compeleceng.2018.07.042.

[65] L. Da Quach, N. P. Quoc, N. H. Thi, D. C. Tran, and M. F. Hassan, "Using SURF to Improve ResNet-50 Model for Poultry Disease Recognition Algorithm," *2020 Int. Conf. Comput. Intell. ICCI 2020*, no. October, pp. 317–321, 2020, doi: 10.1109/ICCI51257.2020.9247698.

[66] M. R. Hossain, S. Afroze, N. Siddique, and M. M. Hoque, "Automatic Detection of Eye Cataract using Deep Convolution Neural Networks (DCNNs)," *2020 IEEE Reg. 10 Symp. TENSYMP 2020*, no. June, pp. 1333–1338, 2020, doi: 10.1109/TENSYMP50017.2020.9231045.

[67] J. Xu, Y. Zhang, and D. Miao, "Three-Way Confusion Matrix for Classification: A Measure Driven View," *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, 2020, doi: 10.1016/j.ins.2019.06.064.

[68] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass Skin Cancer Classification Using EfficientNets – A First Step Towards Preventing Skin Cancer," *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, 2022, doi: 10.1016/j.neuri.2021.100034.

[69] G. Zeng, "On The Confusion Matrix in Credit Scoring and Its Analytical Properties," *Commun. Stat. - Theory Methods*, vol. 47, no. 24, pp. 1–15, 2020, doi: 10.1080/03610926.2019.1568485.

[70] H. Tabrizchi, M. M. Javidi, and V. Amirzadeh, "Estimates of Residential Building Energy Consumption using a Multi-Verse Optimizer-Based Support Vector Machine with K-Fold Cross-Validation," *Evol. Syst.*, vol. 12, pp. 755–767, 2021, doi: 10.1007/s12530-019-09283-8.

[71] Y. N. Fuadah, I. D. Ubaidullah, N. Ibrahim, F. F. Taliningsing, N. K. Sy, and M. A. Pramuditho, "Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, pp. 728–741, 2022, doi: 10.26760/elkomika.v10i3.728.

[72] Y. Wu *et al.*, "Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks," *2019 IEEE Int. Conf. Big Data (Big Data)*, pp. 1971–1980, 2019, doi: 10.1109/BigData47090.2019.9006104.

[73] K. You, M. Long, J. Wang, and M. I. Jordan, "How Does Learning Rate Decay Help Modern Neural Networks?," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1908.01878

[74] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[75] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020, doi: 10.30871/jaic.v4i1.2017.

[76] S. Cai, Y. Shu, G. Chen, B. C. Ooi, W. Wang, and M. Zhang, "Effective and Efficient Dropout for Deep Convolutional Neural Networks," pp. 1–12, 2019, [Online]. Available: http://arxiv.org/abs/1904.03392

[77] A. Ashiquzzaman *et al.*, "Reduction of Overfitting in Diabetes Prediction Using Deep Learning Neural Network," in *IT Convergence and Security 2017*, 2017, pp. 35–43. doi: 10.1007/978-981-10-6451-7.

[78] S. Lee *et al.*, "Improving Scalability of Parallel CNN Training by Adjusting Mini-Batch Size at Run-Time," in *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, 2019, pp. 830–839. doi: 10.1109/BigData47090.2019.9006550.

[79] T. I. Z. M. Putra, Suprapto, and A. F. Bukhori, "Model Klasifikasi Berbasis Multiclass Classification dengan Kombinasi Indobert Embedding dan Long Short-Term Memory untuk Tweet Berbahasa Indonesia (Classification Model Based on Multiclass Classification with a Combination of Indobert Embedding and Long ," *J. Ilmu Siber dan Teknol. Digit.*, vol. 1, no. 1, pp. 1–28, 2022, [Online]. Available: https://doi.org/10.35912/jisted.v1i1.1509

[80] S. S. H. Konala and S. Koppu, "A robust privacy-preserving federated framework for kidney CT image classification using transfer learning models," *Front. Artif. Intell.*, vol. 9, 2026, doi: 10.3389/frai.2026.1840721.

Copyright © 2026, The Author(s).
This is an open access article under the CC BY-SA license
https://creativecommons.org/licenses/by-sa/4.0/