Performance Analysis of Resampling Techniques for Overcoming Data Imbalance in Multiclass Classification

Anggit Larasati[1*], Sugiyarto Surono[2], Aris Thobirin[3], Deshinta Arrova Dewi[4]

[1,2,3] Department of Mathematics, Ahmad Dahlan University, Indonesia, ,[4] Faculty of Data Science and Information Technology, INTI International University, Malaysia

*corr-author: anggitlarasati31@gmail.com

Abstract - In the digital era, the development of modern technology has brought significant transformation to the medical world. The main objective of this research is to identify the performance of deep learning models in classifying kidney disease. By integrating the Convolutional Neural Network model, the performance of the classification process can be analyzed effectively and efficiently. However, data imbalance dramatically affects the performance evaluation of a model, requiring data resampling techniques. This research applies two resampling techniques, bootstrap-based random oversampling and random undersampling, to training data and adds data augmentation to increase image variations to prevent model overfitting. The architecture uses MobileNetV2, which compares hyperparameter fine-tuning in three optimizers. This research shows that the performance of MobileNetV2, which implements the bootstrap-based random oversampling technique, has the highest accuracy compared to random undersampling and no resampling methods. The oversampling technique with the RMSprop optimizer produced the highest accuracy, namely 95%. With precision, recall, and F-1 score, respectively, 0.93, 0.95, 0.94. The accuracy of oversampling with the Adam and Nadam optimizer is 94%. So, the contribution of this research is by applying bootstrap-based oversampling techniques and adding data augmentation to produce good model performance to be used to classify medical images.

Keywords: Convolutional Neural Network; data augmentation; optimizer comparison; resampling.

I. INTRODUCTION

In the digital era, advances in modern technology have brought significant transformation, including integrating big data into the medical world. This is due to the importance of efficiency and accuracy in health data analysis [1]. Image data such as CT scans, MRIs, and X-rays is commonly found in the world of health, especially image-based data [2] [3]. A deep learning approach is needed to process medical images accurately to overcome the disease image classification problem [4] [5]. One of the deep learning methods is convolutional neural network (CNN) [6]. CNN is a more sophisticated deep learning architecture because it focuses on image data processing [7] [8] [9]. In addition, CNN can identify more complex visual patterns in image pixels, where the image pixels are tiny and even invisible to human eyes [10].

However, one of the challenges in processing medical image data is class imbalance. This problem can cause bias in model evaluation and affect model performance [11]. The right step to handle data imbalance problems is to apply resampling techniques. The resampling process is only used for training data because imbalance problems in the training data will affect the overall model performance. Resampling techniques are divided into two, namely undersampling and oversampling. The undersampling technique reduces the majority dataset so that the number of datasets is balanced with the minority dataset. However, undersampling can reduce some vital information from missing data if not careful [12]. Meanwhile, the oversampling technique adds data to minority data by duplicating or creating new data. The oversampling process can cause overfitting because some images that appear only repeat the same previous data, so the model will better memorize the repeated data [13].

One technique used to overcome overfitting is applying data augmentation [14]. Data augmentation increases the variety of images in the training data [15]. After adding image variations, the number of datasets produced increases. This results in a sizeable computational process because it takes a long time during the training process if manual classification is used. Therefore, there is a need for transfer learning to speed up the CNN model training process [16].

Transfer learning is a process that previously trains a model with large amounts of data and then applies it to a different task. This is useful for the effectiveness of the model with limited resources. The aim of using transfer learning is to reduce the running time to be effective for classifying with CNN, considering that CNN uses large amounts of data [17]. With transfer learning, the model training process becomes faster. One transfer learning that can be used is MobileNetV2, where this model is designed using a unique architecture that combines inverted residual blocks and depthwise separable convolutions. This simple architecture model makes it suitable for handling large amounts of data with limited computing. Apart from that, this architecture also supports efficient memory use [18]. Although transfer learning can speed up the training process, the performance of the classification process also depends on selecting the best parameters. So, it is necessary to carry out hyperparameter tuning to adjust the critical parameters to improve CNN image classification performance.

Previous research by [19], has compared the pre-trained model architecture or transfer learning process with the Convolutional Neural Network method, namely MobileNetV2, EfficientNetB6, and NasNet. This comparison resulted in respective accuracies of 91.73%, 91.74%, and 91.72%. This reference journal uses hyperparameter tuning, including learning rate, batch size, number of epochs, and dropout, to avoid overfitting.

Based on research [19], this study emphasizes the importance of the preprocessing process in handling class imbalance in medical image classification. This research contributes to the preprocessing stage, where data resampling with bootstrap-based random oversampling and random undersampling techniques is used to overcome imbalances in training data distribution. Additionally, adding image variations to the training data with data augmentation techniques reduces the risk of overfitting the model. This research compares classification performance using the CNN model combined with three optimizers. This model was applied to classify kidney disease, and the accuracy of the model increased quite significantly. Thus, this research contributes by overcoming data imbalance and model overfitting at the preprocessing stage, which previous research has not explored in depth.

II. METHOD

This research focuses on resampling techniques, namely bootstrap-based random oversampling and random undersampling, to balance training data combined with augmented data. This is then implemented in three optimizers in the CNN model classification. This research stage is represented in Fig. 1.

A. Dataset

This research uses a dataset of kidney disease images from Kaggle, namely https://www.kaggle.com/datasets/baalawi1/kidney-diseases-recognition. **The dataset consists of 4 classes: normal, cyst, stone, and tumor**, numbering 5077, 3709, 2283, and 1377 respectively. Fig. 2 is a sample dataset from each class.

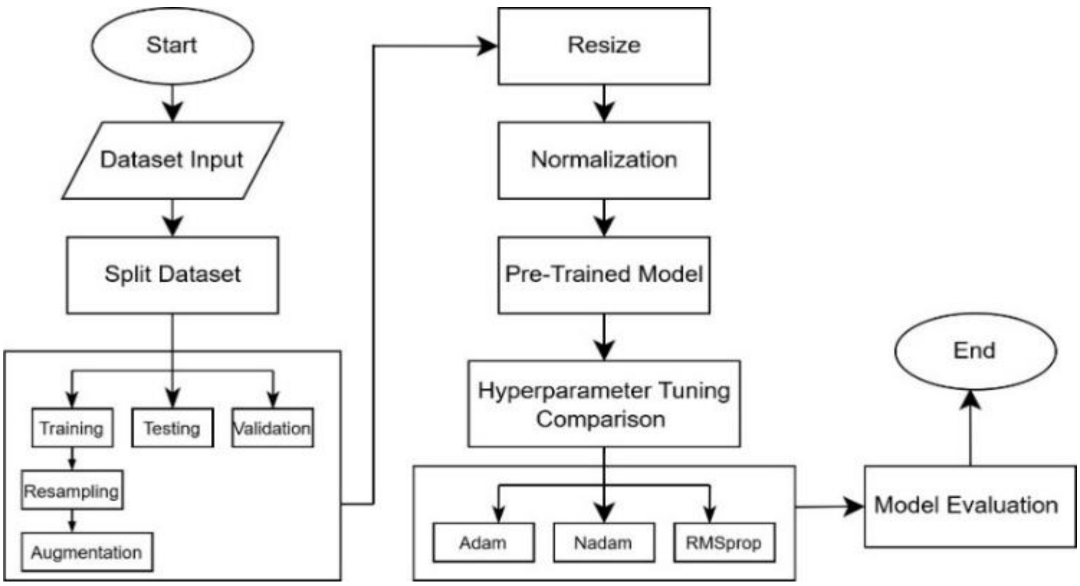

Fig. 1 Research flowchart

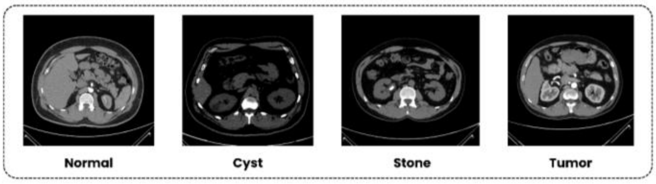

Fig. 2 Kidney disease dataset

B. Split Data

Split the dataset into three parts with a ratio of 80% training, 10% testing, and 10% validation, so each has 9956 training images, 1245 testing images, and 1245 validation images. Also, seed 42 should be implemented so that the experimental results are consistent and do not change.

C. Resampling Technique

The resampling technique is used to overcome class imbalance in the training data, and the aim is to ensure that the model can perform classification accurately [13]. This research applies two resampling techniques, including bootstrap-based random oversampling and random undersampling.

1) Bootstrap-based random oversampling: This research applies an oversampling technique using the bootstrap method (with replacement) to carry out random image sampling until the amount of minority data is balanced with the majority data so that there is a chance that data samples will appear more than once [20]. The random oversampling technique increases the number of minority class datasets consisting of cysts, tumours, and kidney stones to balance the results with the majority dataset (normal kidneys). This technique initializes the majority class, namely the regular kidney class, and then the minority class, which has fewer numbers than normal kidneys, will undergo a random sampling process with returns until the entire dataset for each class is balanced. This way, the minority class dataset is enlarged to reach the desired number or be equivalent to the majority class without modifying the original data.

2) Random undersampling: The random undersampling technique aims to reduce several datasets into 3 classes (normal, cyst, tumour) so that the number is the same as the minority class, namely kidney stones. This technique only deletes data randomly in the three majority classes. The first step is to initialize the minority class, namely kidney stones. Next, the regular, cyst, and tumour classes are randomly deleted until all classes are equal in number. Although effective, random undersampling can lead to the loss of important information because some data from the removed majority class may have helpful information for the model.

D. Preprocessing

The preprocessing stage determines the image quality to be modelled, so this step must be done well. This research integrates data augmentation techniques to modify the preprocessing stage to improve the accuracy performance of the CNN model on training data. The augmentation parameters used in this research include rotation, flipping, zoom, shear, fill mode, and image translation. Next, resize the image to size 224 x 224 to fit the input MobileNetV2 architecture and convert it into a 3-channel RGB image (red, green, blue). After resizing, normalize the data by dividing each pixel by 255 so, can change the pixel data to the range [0, 1] [9]. The rescaling formula at the preprocessing stage, namely (1):

$$Rescaling = \frac{i}{255} \tag{1}$$

where i is the number of each image pixel.

E. Pre-Trained Model

This research uses a pre-trained model to improve classification performance. The architecture used is MobileNetV2, which can be represented in Fig. 3.

Based on Fig. 3, MobileNetV2 architecture used in this research as transfer learning. This research is suitable for using the MobileNetV2 architecture because it has a high-efficiency level in handling large numbers of images or limited computing power. After all, the design is light and efficient [21]. In implementing this model, MobileNetV2 uses initial weights obtained from the training process on the ImageNet dataset.

F. Hyperparameter Tuning

Hyperparameter tuning is very effective for the classification process, which aims to improve the performance of the CNN algorithm [22]. The parameters used in the CNN model classification include a learning rate of 0.0001, batch size of 32, number of epochs of 25, and dropout value of 0.3. Next, these parameters were tested with three different optimizers, namely Adam, Nadam, and RMSprop, which are explained as follows:

Adam: One of the algorithms for updating weights that combines root mean square propagation and adaptive gradient strength. The Adam optimizer is more efficient in exploring the parameter space and examining bias, and the computational resources used are reduced. Adam works by updating parameter weights using first and second-moment estimation corrections. The learning rate stage in parameter updating is (2) [23] [24]:

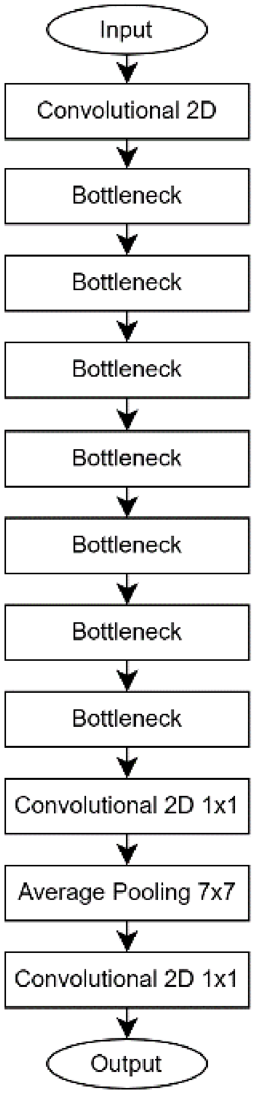

Fig. 3 MobileNetV2 architecture

$$\theta_t = \theta_{t-1} - \eta \frac{m'_t}{\sqrt{n'_t} + \varepsilon} \quad (2)$$

Where θ_t is the t-th learning rate, $\boldsymbol{\eta}$ is the initial learning rate, $m'_t \, dan \, n'_t$ are the first and second moment estimation corrections, ε is a numerical constant

1) NAdam: This optimizer is the same as Adam, but NAdam is a modification of Adam, carried out by estimating Nesterov's accelerated adaptive moments, thus speeding up the minimization process [24]. The stages of updating NAdam parameters are (3):

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_1} + \varepsilon} \left(\beta_1 \hat{m}_t + \frac{(1-\beta_1) g_1}{1-\beta_1^t}\right) \quad (3)$$

Where θ_t is the t-th learning rate, $\boldsymbol{\eta}$ is the initial learning rate, $\sqrt{\hat{v}_1} + \varepsilon$ is the correction between the mean squared gradient with the value ε, $\beta_1 \hat{m}_t$ is the average exponential momentum of the gradient, g_1 is the gradient of the loss function, $\frac{(1-\beta_1) g_1}{1-\beta_1^t}$ as bias correction in the gradient at the first iteration.

2) RMSprop: The RMSprop optimizer belongs to an optimization algorithm with an adaptive learning rate that adjusts each parameter individually. This optimizer effectively deals with noise and accelerates convergence by dynamically adjusting parameter scales [25]. The RMSprop parameter update stage is (4):

$$\theta_t = \theta_{t-1} - \alpha \frac{g_t}{\sqrt{v_1} + \varepsilon'} \quad (4)$$

Where θ_t is the t-th learning rate, α is the initial learning rate, g_t is the gradient of the loss function over the t-th iteration parameters, v_1 is the exponential average of the squared gradient, ε' is a small value to prevent division by zero

G. Confusion Matrix

The Confusion Matrix consists of 4 parts, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). True Positive (TP) is a model that correctly predicts positive values. False Positive (FP) is a model that incorrectly predicts positive values. True Negative (TN) is a model that accurately predicts negative values. False Negative (FN) is a model that incorrectly predicts negative values correctly [14]. Confusion matrix consists of accuracy, precision, recall, F-1 score.

III. RESULT AND DISCUSSION

This research applies a CNN model architecture, namely MobileNetV2 because the computing process is lighter even though the dataset is large and the model has been pre-trained using ImageNet. The total number of kidney disease datasets is 12,046 images divided into three: training, testing, and validation. Comparison of split data, namely 80% training, 10% testing, and 10% validation data. The dataset has four classes, namely normal, cysts, tumours, and kidney stones, and each dataset amounts to 3709, 2283, and 1377. The entire dataset is divided into training, testing, and validation data, each amounting to 9956 training images, 1245 testing images, and 1245 validation images.

Before classification, researchers used resampling techniques to balance classes and added several data augmentation techniques to the training data. Then, model classification is carried out using the MobileNetV2 architecture and hyperparameter tuning. The parameters in hyperparameter tuning are learning rate, batch size, epochs, and dropout and are implemented with different optimizers, namely Nadam, RMSprop, and Adam.

A. Implementation of Resampling Techniques

This research applies undersampling and oversampling techniques to training data. The undersampling technique used is random undersampling (RUS), while the oversampling technique used is bootstrap-based random oversampling (ROS). The results of resampling the training data are in Table I.

Based on table I shows the results of the resampling technique. There are three comparisons: without resampling, with random undersampling, and with bootstrap-based random oversampling. Before resampling, the data distribution was different, namely, 4062 normal kidneys, 2944 kidney cysts, 1849 kidney tumours, and 1101 kidney stones. After applying the undersampling technique, the number of each class followed the lowest number (minority class), namely 1101. Meanwhile, the oversampling technique used applied Bootstrap, and the amount of data followed the most significant number (majority class), namely 4062. To get a clearer picture of the results of the class distribution that has been resampled, it can be represented in graphical form in Fig. 4.

Fig. 4 represents a graph depicting the distribution of each class before and after resampling. It can be seen that the distribution of each class using resampling is balanced towards the majority class (the most abundant class) in the normal kidney class and the minority class (the fewest class) in the kidney stone class.

B. Addition of Data Augmentation Techniques

Based on the parameters in data augmentation techniques such as rotation, flipping, zoom, shear, fill mode, and image translation, the resulting image after adding these parameters is shown in Fig. 5.

TABLE I
APPLICATION OF RESAMPLING TECHNIQUES

Class	**Before Resampling**	**After Undersampling**	**After Oversampling**
Normal	4062	1101	4062
Cyst	2944	1101	4062
Tumor	1849	1101	4062
Stone	1101	1101	4026

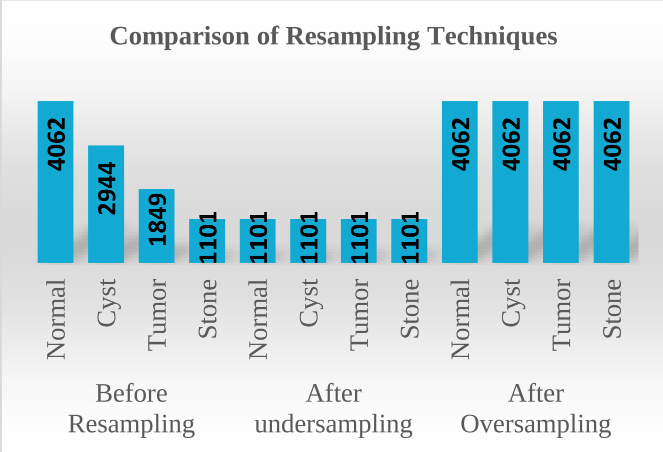

Fig. 4 Comparison graph of data resampling techniques

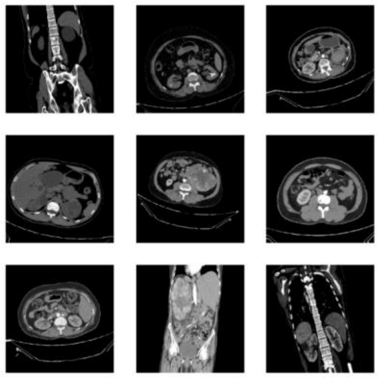

Fig. 5 Sample data augmentation

Based on Fig. 5, the dataset becomes more varied and does not only focus on one point. Using a rotation parameter of 10%, the resulting image looks skewed because there are not too many parameter sets. The image's brightness level is also slightly increased to give the image used for training a good brightness level. So, with data augmentation techniques, we succeed in expanding the variety of datasets to enrich visual information for training models. Additionally, even though the images undergo augmentation techniques, the critical information from the dataset remains valid. This technique can help the model create good data variations and overcome overfitting in the model being run.

C. CNN Model Classification

The results of the classification process include the values of training accuracy, validation accuracy, training loss, and validation loss, which are listed in Table II.

Based on Table II, the best validation accuracy was obtained using oversampling techniques in balancing classes in the training data. The CNN model that uses the RMSprop optimizer and is implemented in the

MobileNetV2 architecture has the highest accuracy, namely 0.9558 or 95.58%. Meanwhile, the accuracy validation results in the Adam optimizer got an accuracy of 95.5%, and Nadam got an accuracy of 94.7%. To find out the comparison between loss and validation accuracy, it can be represented in Fig. 6.

Based on Fig. 6, it can be seen that comparing validation data on the CNN model, which applies the resampling technique, namely bootstrap-based random oversampling (ROS), has better results than random undersampling (RUS) and without resampling (Normal). The graph shows that the lowest loss value is obtained using the oversampling technique. By balancing the amount of data between classes, the dataset for training becomes larger. This results in increasing model accuracy and decreasing loss values. The highest accuracy is also found in the oversampling technique. The highest accuracy of the CNN model uses the MobileNetV2 architecture with the RMSprop optimizer, which has an accuracy of 95.58%. Furthermore, the Adam optimizer has an accuracy of 95.5%, and the Nadam optimizer has an accuracy of 94.7%. Then, the confusion matrix images are presented in Fig. 7, Fig. 8, and Fig. 9, with each optimizer in sequence, namely Adam, Nadam, and RMSprop.

TABLE II
RESULTS TRAINING

Optimizer	***Resampling Technique***	***Comparison of Training Results***			
		Tr-Acc	**Val-Acc**	**Tr-Loss**	**Val-Loss**
Adam	RUS	0.9364	0.9020	0.1768	0.2491
	ROS	0.9836	0.9550	0.0552	0.1118
	Normal	0.9669	0.9250	0.0986	0.1907
RMSprop	RUS	0.9203	0.8627	0.2129	0.3344
	ROS	0.98	0.9558	0.0586	0.1144
	Normal	0.9626	0.9285	0.1023	0.1727
Nadam	RUS	0.9371	0.8916	0.1795	0.2683
	ROS	0.9831	0.9470	0.0515	0.1382
	Normal	0.9696	0.9357	0.0966	0.1637

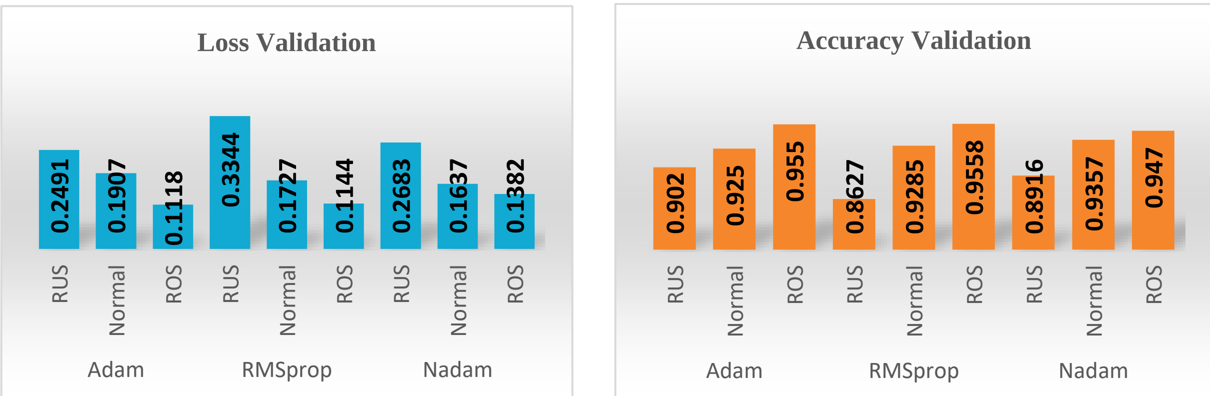

Fig. 6 Graphs of loss and accuracy values on validation data

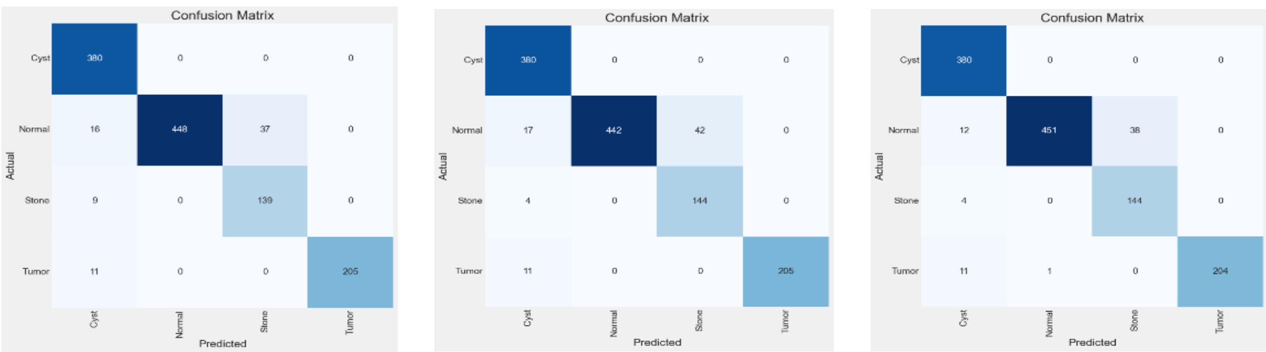

Fig. 7 Confusion matrix with bootstrap-based random oversampling

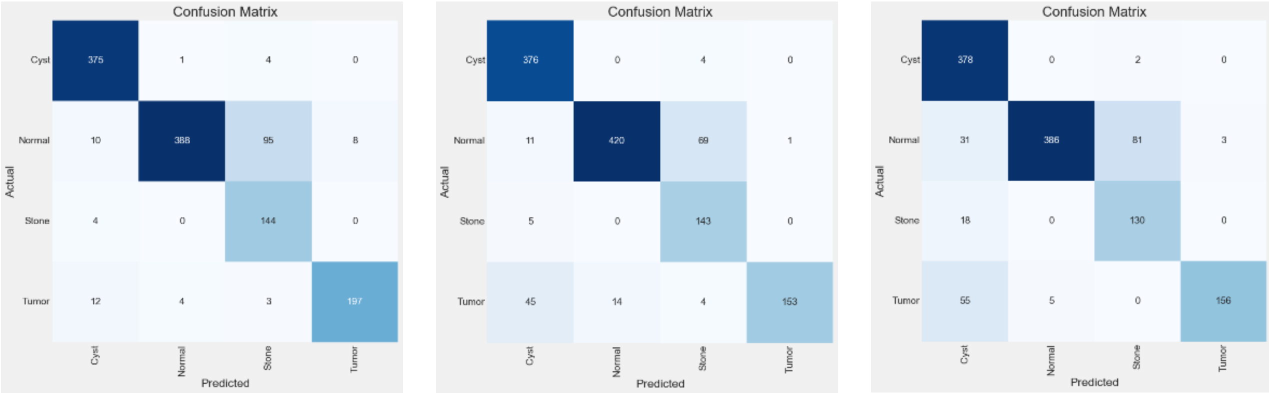

Fig. 8 Confusion matrices with random undersampling

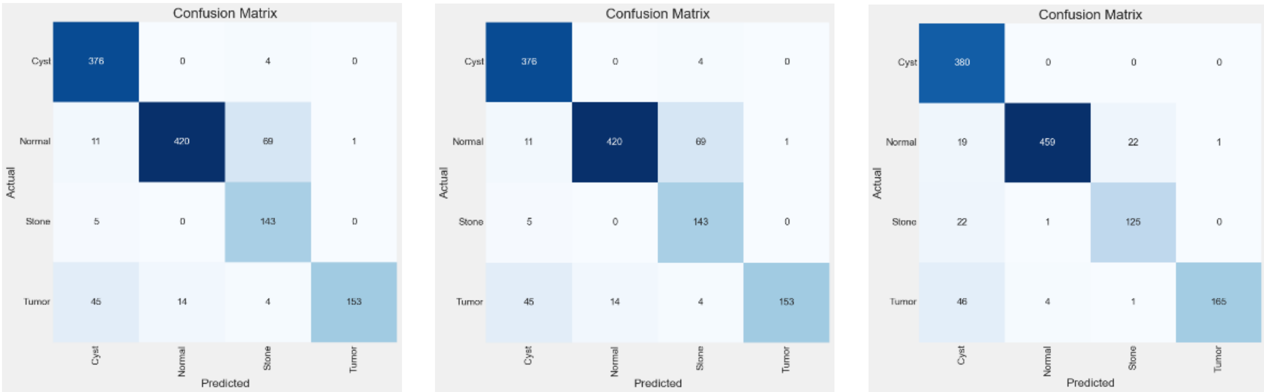

Fig. 9 Confusion matrices without resampling

Model performance evaluation is carried out by calculating evaluation metrics from the data in the confusion matrix. **The confusion matrix calculation namely accuracy, precision, recall, and F-1 Score**. This metric helps assess model performance based on classification accuracy for certain classes. Table III presents the confusion matrix calculation results to facilitate the model performance analysis process.

Based on Table III, the application of resampling techniques combined with several optimization methods succeeded in improving the performance of the CNN model in classifying kidney disease datasets. The results show that using random oversampling with Bootstrap produces the best performance, especially when combined with the RMSprop optimizer. The model with this combination achieved the highest accuracy of 95%, with precision, recall, and F1 scores of 0.93, 0.95, and 0.94, respectively.

TABLE III
CALCULATION OF THE CONFUSION MATRIX

Optimizer	***Resampling Technique***	**Accuracy (%)**	**Precision**	**Recall**	**F-1 Score**
Adam	RUS	89%	0.87	0.91	0.88
	ROS	94%	0.93	0.95	0.93
	Normal	88%	0.87	0.88	0.86
RMSprop	RUS	84%	0.84	0.84	0.83
	ROS	95%	0.93	0.95	0.94
	Normal	91%	0.91	0.88	0.89
Nadam	RUS	88%	0.87	0.88	0.86
	ROS	94%	0.92	0.95	0.93
	Normal	88%	0.87	0.88	0.86

The superiority of RMSprop in this scenario can be attributed to its parameter update mechanism, as stated in equation (4), where the learning rate value is adjusted adaptively based on the exponential average of the squared gradients. Thus, RMSprop can effectively handle oscillating gradients and maintain stability during the learning process. This capability is very useful in kidney disease datasets that have an unbalanced class distribution and may contain noise.

In comparison, Adam and Nadam have different characteristics in optimizing the model. Adam (Adaptive Moment Estimation) uses first and second momentum estimates as stated in equation (2), which allows adjusting the learning rate for each parameter. However, in some cases, Adam can have difficulty finding the optimal solution because it tends to maintain greater momentum, which can lead to overshooting or overfitting.

Nadam, as a modification of Adam with Nesterov momentum, implements a more stable approach through momentum correction in the gradient as formulated in equation (3). Although it is expected to provide faster convergence than Adam, in these experiments, Nadam does not show significant advantages over RMSprop. This may be due to the characteristics of the dataset that are more favorable to optimization methods with quadratic gradient-based learning rate adjustments compared to momentum-based approaches.

The main advantage of RMSprop compared to Adam and Nadam in this experiment can be seen from the final results which are more stable and have a higher level of accuracy. This shows that gradient variability-based learning rate control in RMSprop is more suitable for kidney disease datasets, which have complex distribution patterns. Figure 6 shows the confusion matrix of the combination of the Bootstrap resampling technique with the RMSprop optimizer, which shows the best performance in classifying all classes of kidney diseases.

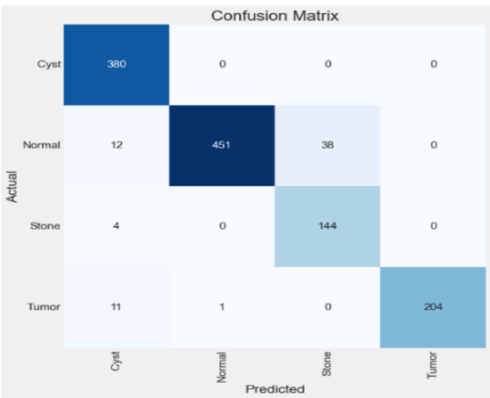

Fig. 10 Confusion matrices with random oversampling technique – RMSprop

Fig. 10 shows the model evaluation results by applying resampling techniques using Bootstrap and the RMSprop optimizer. The results show that the cyst class has 380 correct predictions. Twelve wrong predictions were classified in the normal class. Four wrong predictions were classified in the stone class. Eleven incorrect predictions were classified into tumour classes. The normal class has 451 correct predictions and one incorrect prediction. Thirty-eight incorrect predictions were classified into the stone class, and 12 incorrect predictions were classified into the cyst class. Then, the stone class had 144 correct predictions and 38 incorrect predictions. There were four incorrect predictions in the cyst class. Meanwhile, the tumour class had 204 correct predictions. One incorrect prediction was in the normal class, and 11 were in the cyst class. So, using the resampling model and optimizer, RMSprop has the best value for evaluating the model.

IV. CONCLUSION

Kidney disease is a disease that is of particular concern in the world of health. Therefore, accurate early detection is needed by utilizing deep learning models for disease image classification. This study used a kidney disease dataset of 12,046 images of 4 classes: normal kidneys, kidney cysts, kidney tumours, and kidney stones. The author uses resampling techniques with undersampling and oversampling on training data and adding data augmentation to overcome overfitting and increase variations in the resampled images. The architecture used, MobileNetV2 was then classified using three different optimizers, namely Nadam, RMSprop, and Adam, and then the accuracy results were compared. The results of this research are that the performance of MobileNetV2 implemented with a bootstrap-based oversampling technique has the highest accuracy compared to undersampling and without resampling techniques. The oversampling technique with the RMSprop optimizer produced the highest accuracy, namely 95%. Then, the accuracy of oversampling with the Adam and Nadam optimizer is 94%. The performance of MobileNetV2, which implements random undersampling techniques with the RMSprop, Adam, and Nadam optimizers, is 84%, 89%, and 88%. Meanwhile, accuracy without resampling techniques using the optimizer is 91%, 88%, 88%. So, it can be concluded that applying the resampling technique with bootstrap produces a good model performance that can be used to classify medical images. This research could be further explored with different architecture performances and added optimizations such as Grey Wolf optimizer. Apart from that, it can be continued by implementing the results in the form of an application or website that can be used directly in the world of health.

ACKNOWLEDGEMENT

The author would like to thank all parties for their guidance and financial support, which provided various facilities during the research process, especially for the mathematics laboratory at Ahmad Dahlan University, which helped me in this research.

REFERENCES

[1] X. Bai, J. Duan, Bo Li, S. Fu, W. Yin and Z. Qu, "Global quantitative analysis andBai, X., Duan, J., Li, B., Fu, S., Yin, W., Yang, Z., & Qu, Z. (2024). Global quantitative analysis and visualization of big data and medical devices based on bibliometrics. Expert Systems with Applications, 254(June). http," *Expert Syst. Appl.*, vol. 254, no. June, 2024, doi: 10.1016/j.eswa.2024.124398.

[2] V. Muthukrishnan, S. Jaipurkar, and N. Damodaran, "Continuum topological derivative - A novel application tool for segmentation of CT and MRI images," *NeuroImage: Reports*, vol. 4, no. 3, 2024, doi: 10.1016/j.ynirp.2024.100215.

[3] M. Momeny, A. A. Neshat, M. A. Hussain, S. Kia, M. Marhamati, A. Jahanbakhshi, "Learning-to-augment strategy using noisy and denoised data: Improving generalizability of deep CNN for the detection of COVID-19 in X-ray images," *Comput. Biol. Med.*, vol. 136, no. July, 2021, doi: 10.1016/j.compbiomed.2021.104704.

[4] Y. Kim, S. Bu, C. Tao, K. T. Bae, P. Kidney, and D. Study, "Deep Learning – Based Automated Imaging Classi fi cation of ADPKD," pp. 1802–1809, 2024, doi: 10.1016/j.ekir.2024.04.002

[5] J. Achatz, M. Lukovic, S. Hilt, T. Lädrach, and M. Schubert, "Convolutional neural networks for quality and species sorting of roundwood with image and numerical data," *Expert Syst. Appl.*, vol. 246, no. October 2023, 2024, doi: 10.1016/j.eswa.2023.123117.

[6] S. Surono, M. Y. F. Afitian, A. Setyawan, D. K. E. Arofah, and A. Thobirin, "Comparison of CNN Classification Model using Machine Learning with Bayesian Optimizer," *HighTech Innov. J.*, vol. 4, no. 3, pp. 531–542, 2023, doi: 10.28991/HIJ-2023-04-03-05.

[7] C. W. O. Khang Wen Goh, Sugiyarto Surono, M. Y. Firza Afiatin, K. Robiatul Mahmudah, Nursyiva Irsalinda, Mesith Chaimanee, "Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification," *Emerging Science Journal*, vol. 8, no. 2. pp. 592–602, 2024. doi: 10.28991/ESJ-2024-08-02-014.

[8] N. A. Mohammed, M. H. Abed, and A. T. Albu-Salih, "Convolutional neural network for color images classification," *Bull. Electr. Eng. Informatics*, vol. 11, no. 3, pp. 1343–1349, 2022, doi: 10.11591/eei.v11i3.3730.

[9] V. Sharma, S. Kannan, S. Tanya, and N. Panda, "Detecting Plant Diseases at Scale: A Distributed CNN Approach with PySpark and Hadoop," *Procedia Comput. Sci.*, vol. 235, no. 2023, pp. 1044–1057, 2024, doi: 10.1016/j.procs.2024.04.099.

[10] F. S. Ishengoma and N. N. Lyimo, "Ensemble model for grape leaf disease detection using CNN feature extractors and random forest classifier," *Heliyon*, vol. 10, no. 12, 2024, doi: 10.1016/j.heliyon.2024.e33377.

[11] M. Elena, V. Barbu, and V. Barbu, "ScienceDirect Advanced methods for dealing with high data imbalance for Embryo Advanced dealing high Advanced methods methods for for Fertility dealing with with high data data imbalance imbalance for for Embryo Embryo Classification Fertility Fertility C," 2024, doi: 10.1016/j.procs.2024.09.230

[12] B. Zhu, X. Jing, L. Qiu, and R. Li, "An Imbalanced Data Classification Method Based on Hybrid Resampling and Fine Cost Sensitive Support Vector Machine," *Comput. Mater. Contin.*, vol. 79, no. 3, pp. 3977–3999, 2024, doi: 10.32604/cmc.2024.048062.

[13] P. Wibowo and C. Fatichah, "Pruning-based oversampling technique with smoothed bootstrap resampling for imbalanced clinical dataset of Covid-19," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7830–7839, 2022, doi: 10.1016/j.jksuci.2021.09.021.

[14] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artif. Intell. Agric.*, vol. 6, pp. 23–33, 2022, doi: 10.1016/j.aiia.2021.12.002.

[15] R. Ye, A. Boukerche, X. S. Yu, C. Zhang, B. Yan, and X. J. Zhou, "Data augmentation method for insulators based on Cycle GAN," *J. Electron. Sci. Technol.*, vol. 22, no. 2, 2024, doi: 10.1016/j.jnlest.2024.100250.

[16] K. Alhada-Lahbabi, D. Deleruyelle, and B. Gautier, "Transfer learning for accelerating phase-field modeling of ferroelectric domain formation in large-scale 3D systems," *Comput. Methods Appl. Mech. Eng.*, vol. 429, no. June, pp. 1–16, 2024, doi: 10.1016/j.cma.2024.117167.

[17] S. G. Paul, A. A. Biswas, A. Saha, Md. S. Zulfiker, N. A. Ritu, I. Zahan, M. Rahman, "A real-time application-based convolutional neural network approach for tomato leaf disease classification," *Array*, vol. 19, no. March, 2023, doi: 10.1016/j.array.2023.100313.

[18] G. Zhou, Q. He, X. Liu, X. Kai, W. Cao, J. Ding, B. Zhuang, S. Xu, "Optimizing MobileNetV2 for improved accuracy in early gastric cancer detection based on dynamic pelican optimizer," *Heliyon*, vol. 10, no. 16, pp. 1–17, 2024, doi: 10.1016/j.heliyon.2024.e35854.

[19] V. Singh, A. Chug, and A. P. Singh, "Classification of Beans Leaf Diseases using Fine Tuned CNN Model,"

Procedia Comput. Sci., vol. 218, no. 2022, pp. 348–356, 2022, doi: 10.1016/j.procs.2023.01.017.

[20] R. Bertolini, S. J. Finch, and R. H. Nehm, "Quantifying variability in predictions of student performance: Examining the impact of bootstrap resampling in data pipelines," *Comput. Educ. Artif. Intell.*, vol. 3, no. February, p. 100067, 2022, doi: 10.1016/j.caeai.2022.100067.

[21] K. Baran, "Smartphone thermal imaging for stressed people classification using CNN+MobileNetV2," *Procedia Comput. Sci.*, vol. 225, pp. 2507–2515, 2023, doi: 10.1016/j.procs.2023.10.242.

[22] G. S. Manivannan, H. Rajaguru, R. S, and S. V. Talawar, "Cardiovascular disease detection from cardiac arrhythmia ECG signals using artificial intelligence models with hyperparameters tuning methodologies," *Heliyon*, vol. 10, no. 17, 2024, doi: 10.1016/j.heliyon.2024.e36751.

[23] J. Kang, X. Zhu, L. Shen, and M. Li, "Fault diagnosis of a wave energy converter gearbox based on an Adam optimized CNN-LSTM algorithm," *Renew. Energy*, vol. 231, no. July, 2024, doi: 10.1016/j.renene.2024.121022.

[24] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, "Survey of Optimization Algorithms in Modern Neural Networks," *Mathematics*, vol. 11, no. 11, 2023, doi: 10.3390/math11112466.

[25] S. A. Hasib, M. M. Gulzar, A. Shakoor, S. Habib, and A. F. Murtaza, "Optimizing electric vehicle driving range prediction using deep learning: A deep neural network (DNN) approach," *Results Eng.*, vol. 24, no. October, 2024, doi: 10.1016/j.rineng.2024.103630.