International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 3
ISSN: 2088-5334

An Efficient and Robust Ischemic Stroke Detection Using a Combination of Convolutional Neural Network (CNN) and Kernel K-Means Clustering

Zuherman Rustam [a*], Sri Hartini [a], Fevi Novkaniza [a], Jacob Pandelaki [b], Rahmat Hidayat [c], Mostafa Ezziyyani [d]

[a] *Department of Mathematics, University of Indonesia, Depok 16424, Indonesia*
[b] *Department of Radiology, Cipto Mangunkusumo Hospital, Jakarta 10430, Indonesia*
[c] *Department of Information Technology, Politeknik Negeri Padang, Padang 25163, Indonesia*
[d] *University Abdelmalek Essaadi, Faculty of Sciences and Techniques of Tangier, Morocco*
Corresponding author: [*]rustam@ui.ac.id

Abstract— **This study introduces a combined approach utilizing the widely-used Convolutional Neural Network (CNN) and Kernel K-Means clustering method for the detection of ischemic stroke from Magnetic Resonance Imaging (MRI) images. We propose an efficient and robust alternating classification scheme to overcome the challenges of extensive computation time and noisy ischemic stroke images obtained from Cipto Mangunkusumo Hospital in Indonesia. The method incorporates multiple convolutional layers from the CNN architecture and subsequently vectorizes the matrix output to serve as input for Kernel K-Means clustering. Through a series of experiments, our proposed method has demonstrated highly promising results. Employing 11-fold cross-validation and the RBF kernel function (sigma= 0.05), we achieved exceptional performance metrics, including 99% accuracy, 100% sensitivity, 98% precision, 98.04% specificity, and 98.99% F1-Score. These outcomes underscore the remarkable capabilities of the combined CNN and Kernel K-Means clustering approach in accurately identifying ischemic stroke cases. Furthermore, our method exhibits competitive performance when compared to several other state-of-the-art methods in the field of deep learning. By harnessing the power of CNN's convolutional layers and the clustering capability of Kernel K-Means, we have achieved significant advancements in the domain of ischemic stroke detection from MRI images. The implications of this research are substantial. By enhancing the accuracy and efficiency of ischemic stroke detection, our method has the potential to assist medical professionals in making timely and informed decisions for stroke patients. Early detection and intervention can greatly improve patient outcomes and contribute to more effective treatment strategies.**

Keywords—**Artificial neural network; deep learning; image classification; kernel function; k-means clustering; ischemic stroke detection.**

Manuscript received 27 Aug. 2022; revised 29 Oct. 2021; accepted 12 Dec. 2022. Date of publication 30 Jun. 2023.

I. INTRODUCTION

Ischemic stroke is a kind of stroke besides the hemorrhagic stroke. It is the most common type because it dominates 85 percent of stroke cases [1]. It is also ranked as the second leading cause of death worldwide, with an annual mortality rate of about 5.5 million [2]. This disease is usually caused by a blood clot that blocks or plugs a blood vessel in the brain that keeps blood from flowing to the brain. Therefore, the duration of cerebral ischemia is a critical factor in determining the severity of brain damage. Thus, accurate and timely observation of the ischemic process is highly critical to the course of action [3]. An ischemic stroke can be detected by analyzing its Computerized Tomographic (CT) scan and Magnetic Resonance Imaging (MRI) results. However, MRI is frequently used because it provides more detail than a CT scan result [4]. Several methods have been used in previous research to detect an ischemic stroke. Some applied support vector machines [5]–[9], clustering [10]–[14], but most used the various architecture of neural networks. As the frequently used method, deep learning outperforms other techniques for complex problems such as image classification if the data size is large. The Ischemic Stroke Lesion Segmentation 2017 provided a dataset of MRI images to detect an ischemic stroke. Many researchers almost always employed deep learning tools, predominately convolutional neural networks

(CNNs) to reach the optimal results [15]. However, even though deep learning, such as CNN, delivers impressive accuracy in detecting ischemic stroke, it takes much computational time to train the model. It depends on the number of epochs and batch size used so that the expected out-of-sample error can be reached without causing the model to become overfitting [16]. Meanwhile, machine learning methods such as Support Vector Machines (SVM) may usually become inefficient once it processes many feature vectors [17]. Therefore, in this research, we would like to explore combining deep learning architecture and machine learning methods to overcome those limitations. Specifically, we proposed an effective and robust method combining CNN and K-Means clustering based on kernel. The extensive input of images on CNN is reduced while passing the model layers. After that, using the flattened layer output of the last convolutional step in CNN, the data matrix of the different classes will be constructed as the input for Kernel K-Means clustering, which has a smaller size than the original input images. As a result, the centroid of each class was built to detect the class of the out-of-sample dataset.

II. MATERIAL AND METHOD

This section explains the dataset used and the proposed methods in detail.

A. The Dataset

The dataset used in this research was collected from Cipto Mangunkusumo (RSCM) Hospital, Indonesia. It has 334 images of 185 non-ischemic strokes and 149 ischemic stroke images. The phone camera with the 48-megapixel camera with an f/2.0 aperture took those images from an MRI scan. The average height of images is 3047 pixels, with the maximum and minimum heights being 4023 and 2225 pixels, respectively. Meanwhile, the average width of images is 2598 pixels, with the maximum and minimum widths being 3545 and 1902 pixels, respectively. However, the input of images that we used is 512 x 512 pixels, so the size of the input images is not huge. However, all of the regions of the brain are still included.

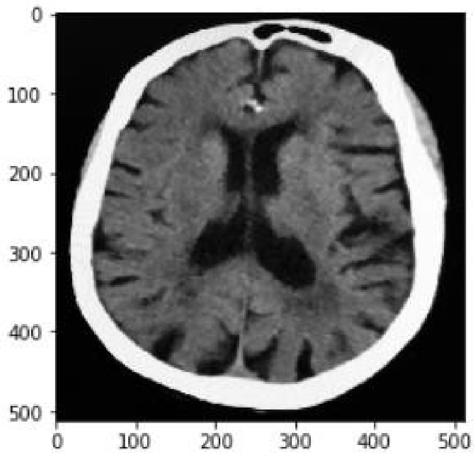

Fig. 1 The sample of the ischemic stroke MRI image

The sample of the ischemic stroke MRI image is shown in Figure 1. From the image, we can identify that the middle region of the brain is fueled by a dark area, which indicates the existence of the disease. Compared with Figure 1, the non-ischemic stroke MRI image shown in Figure 2 has a clearer region. However, the non-ischemic stroke dataset has not only a healthy brain image. Indeed, it also consists of the other brain diseases image that is not categorized as ischemic stroke.

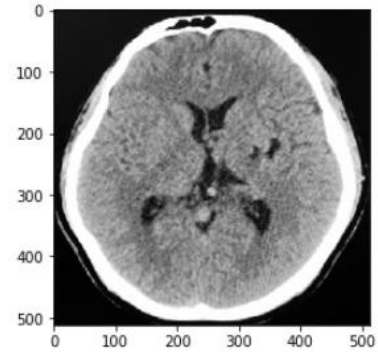

Fig. 2 The sample of the ischemic stroke MRI image

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep feed-forward artificial neural network architecture frequently used in computer vision problems such as image classification. The difference between CNN and multilayer perceptron (MLP) networks is its usage of convolutional layers, pooling, and non-linearities such as tanh, sigmoid, and ReLU [14]. In this research, Keras, as the Python deep learning library, is used to build the model architecture where the details were given in Figure 3 below.

(1) INPUT: 152 x 152 x 1
(2) Conv2D: 3x3 size, 32 filters, 1 stride
(3) ReLU: max(0, hθ(x))
(4) MaxPooling: 2x2 size, 1 stride
(5) BatchNormalization
(6) Conv2D: 3x3 size, 64 filters, 1 stride
(7) ReLU: max(0, hθ(x))
(8) MaxPooling: 2x2 size, 1 stride
(9) BatchNormalization
(10) Conv2D: 3x3 size, 96 filters, 1 stride
(11) ReLU: max(0, hθ(x))
(12) MaxPooling: 2x2 size, 1 stride
(13) BatchNormalization
(14) Conv2D: 3x3 size, 64 filters, 1 stride
(15) ReLU: max(0, hθ(x))
(16) MaxPooling: 2x2 size, 1 stride
(17) BatchNormalization
(18) Conv2D: 3x3 size, 32 filters, 1 stride
(19) ReLU: max(0, hθ(x))
(20) MaxPooling: 2x2 size, 1 stride
(21) Dropout: rate=0.2
(22) Flatten output: 1 x 6272

Fig. 3 The CNN architecture used in this paper

Consider an input grayscale image of 152 x 152 x 1 pixel. This input image will be processed through a convolutional layer with the Rectified Linear Unit (ReLU) activation function, maximum pooling layer, and normalization layer, respectively, before finally being applied dropout and making the output flat.

In this research, the two-dimension convolutional layer (denoted by Conv2D) consists of a number of filters with size 3 × 3 × 1. This layer constructs a convolution kernel that is convoluted with the layer input to produce a tensor of outputs through the dot product computation. Convolution operation extracts useful features from locally correlated data points. The output of the convolutional kernels is assigned to the non-linear processing unit (activation function), which not only helps in learning abstractions but also embeds non-linearity in the feature space [15].

Consequently, an activation function is used for familiarizing non-linearities in the computation so that the model does not only learn linear mappings. The Rectified Linear Unit (ReLU) activation function is used in this research. This activation function is the most widely used activation function for deep learning because it delivers better performance and generalization in deep learning compared to tanh and sigmoid activation functions [16]. Simply thresholding matrix values implement ReLU at zero (see Eq. 1).

$$f\big(h\theta(x)\big) = max\big(0, h\theta(x)\big) \tag{1}$$

Then, the maximum pooling layer (denoted by MaxPooling) is used to reduce the size of input images from the convolutional layer using the max-pooling operation. Lastly, the normalization layer is used to normalize the previous layer's activations at each batch. For example, it applied a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

After the input image passes the bunch of convolutional layers, the information in every hidden neuron output will only keep 80 percent information because the dropout rate is 0.2. As the last step, the output obtained from this CNN is the vector with a length 6272 for every input image.

C. K-Means Clustering Based on Kernel

K-Means (KM) clustering based on kernel came from K-Means clustering, which was introduced by Lloyd [21] in 1982 as the non-probabilistic technique used to cluster the data. Consider dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ where **x** is the D-dimensional data point. The goal of KM clustering is grouping the dataset to C clusters, assumed C is given, where the identic data points are in the same cluster, and the data points with different characteristics are in a separate cluster.

For every cluster c , $c = 1,2, \dots C$, defined μ_c as the centroid of the cluster. Besides, for every data point **x**n, defined $r_{nc \in \{0,1\}}$ as the indicator of whether the data points are in the cluster (denoted as 1) or not (denoted as 0), KM clustering aims to minimize its objective function (see Eq. 2) so that every data point is in the cluster where the centroid is nearest [22].

$$J_{KM} = \sum_n \sum_c r_{nc} \, \|\boldsymbol{x}_n - \boldsymbol{\mu}_c\|^2 \tag{2}$$

As the real world cannot always be resolved using linear functions, the kernel offers an alternative solution by mapping the data point into a higher dimensional feature space so that it can make the data linearly separated [23]. Assumed the mapping function ϕ, which mapped the input data x to the higher dimensional feature space ϕ(x). Because now we work in this feature space, the objective function in Eq. 2 becomes as shown in Eq.3.

$$J_{KM} = \sum_n \sum_c r_{nc} \, \|\phi(\boldsymbol{x}_n) - \boldsymbol{\phi}(\boldsymbol{\mu}_c)\|^2 \tag{3}$$

Considering the computational cost of the distance of two points in the higher dimensional feature space, Vapnik [24] defined the kernel function (see Eq. 4) for every x ϵ $\mathbb{R}^n$ as the dot product of the mapped result of **x** in the feature space.

$$K(x, \boldsymbol{y}) = \big(\phi(\boldsymbol{x})\big)^T \phi(\boldsymbol{y}) \tag{4}$$

According to Eq. 4, the distance between $\phi(\boldsymbol{x}n)$ and $\phi(\boldsymbol{\mu}c)$ can be computed, as shown in Eq. 5

$$\|\phi(x_n) - \boldsymbol{\phi}(\boldsymbol{\mu}_c)\|^2 = K(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2K(\boldsymbol{x}_n, \boldsymbol{\mu}_c) + K(\boldsymbol{\mu}_c, \boldsymbol{\mu}_c) \tag{5}$$

Therefore, the objective function of KM clustering based on kernel becomes the Eq. 6 with the same goal as the KM clustering [25].

$$J_{KM} = \sum_n \sum_c r_{nc} \, K(\boldsymbol{x}_n, \boldsymbol{x}_n) - 2K(\boldsymbol{x}_n, \boldsymbol{\mu}_c) + K(\boldsymbol{\mu}_c, \boldsymbol{\mu}_c) \tag{6}$$

There are several kinds of kernel function. However, in this research, the Radial Basis Function (RBF) kernel function (see Eq. 7) is used due to its ability to perform well over the other kernel functions [26].

$$K(x, y) = \exp(\|x - y\|^2 / 2\sigma^2) \tag{7}$$

The algorithm of KM clustering based on the kernel is given in Figure 4 below.

Input:
$X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \boldsymbol{x_n}\}$, C the number of cluster , ε (epsilon), T the maximum number of iterations.
Output:
$\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \dots \boldsymbol{v_C}\}$ set of cluster center
$\boldsymbol{U} = [\boldsymbol{r_{nc}}]$ *where* n=1,2,…N , c=1,2,…,C

1) Initialization $\boldsymbol{V^0} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \dots \boldsymbol{v_C}\}$
2) Update membership of the data point $\mathbf{x_i}$ in j[th]-cluster, $\forall c = 1,2, \dots, C$
   $if\ c = \arg min \{K(\boldsymbol{xi}, \boldsymbol{xi}) - 2\,K(\boldsymbol{xi}, \boldsymbol{\mu c}) + K(\boldsymbol{\mu c}, \boldsymbol{\mu c})\}\ then\ r_{nc} = 1\ \ otherwise\ r_{nc} = 0$
3) Update cluster center V^t using
   $$\mu_C = \frac{\sum_n r_{nc} \boldsymbol{x}_n}{\sum_n r_{nc}}$$
4) Check the stop criteria :
5) $if\ \|V^{(t-1)} - V^t\| \le \varepsilon\ or\ T = t\ then$
   $the\ iteration\ stops,$
   $otherwise\ t = t + 1\ and\ go\ to\ step\ 2$
6) end

Fig. 4 The algorithm of K-Means Clustering based on kernel

D. The Proposed Method

Unlike previous research in image classification that combined CNN with other deep learning techniques, such as Recurrent Neural Network (RNN) by Yin et al. [27] and Long-Short Term Memory (LSTM) by Aditi et al. [28], we proposed the method that combined widely-used CNN with one of the well-known machine learning methods. However, it is not Support Vector Machines (SVM) like what Sugg [29] did in his thesis or Copur et al. [30] did in their paper, but we combine CNN with KM clustering based on RBF kernel instead.

First, using 5-fold cross-validation, we conducted experiments to find the optimal RBF parameter σ. After that, the value of σ is then used to several k in k-fold cross-validation. As the input of CNN, we used all 334 labeled images: 1 for ischemic stroke images and 0 for non-ischemic images. The images are resized to the same size 152 x 152 pixels. This input is then passed to the bunch of CNN layers, as described in Figure 3. As a result, every image 152 x 152 x 1 pixel became a vector with a length of 6272. Therefore, we now have matrix 334 x 6272, where the row index indicates the image we observed, and the column index indicates the feature map resulting from CNN.

The next step is then dividing the matrix data according to its label, whether it is included in class 1 (ischemic stroke) or class 0 (non-ischemic stroke). In this case, we have two matrices' data with the size 185 x 6272 for the non-ischemic stroke and 149 x 6272 for the ischemic stroke class. Those matrices were then used in k-fold cross-validation for evaluating KM clustering based on the kernel algorithm. For example, when we used 5-fold cross-validation, the data was divided into five folds for each class. Therefore, we get the number of points in every fold as shown in Table I.

TABLE I
THE NUMBER OF DATA IN EVERY 5 FOLDS OF ISCHEMIC AND NON-ISCHEMIC STROKE

Fold	The number of ischemic stroke data points	The number of dataset non-ischemic stroke data points
1	30	37
2	30	37
3	30	37
4	30	37
5	29	37
Total	149	185

The k-fold cross-validation using KM clustering based on kernel might be different from the usual used in the supervised learning method in machine learning. In this KM clustering based on kernel, a fold was used to obtain the centroids of the clusters according to the algorithm in Figure 4, while the rest $k-1$ folds were used to evaluate the method by determining the class of every data point according to its nearest centroid. If the data point was nearer to the centroid of class 1, then the predicted class for this data point is 1. Meanwhile, if the data point was nearer to the centroid of class 0, then the predicted class for this data point is 0.

E. Performance Measure

The confusion matrix (see Table II) is used to calculate the percentage of accuracy, sensitivity, precision, specificity, and F1-Score to measure the performance of combined CNN and KM clustering based on kernel method. There are four possible outcomes in the confusion matrix that we might obtain: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

TABLE II
CONFUSION MATRIX FOR ISCHEMIC STROKE DATA SET

Confusion Matrix		Predicted Class	
		Ischemic Stroke	Non-Ischemic Stroke
Actual Class	Ischemic Stroke	TP	FN
	Non-Ischemic Stroke	FP	TN

The condition where ischemic stroke data is correctly predicted as ischemic stroke counts as True Positive, while the condition where non-ischemic stroke data is correctly predicted as non-ischemic stroke counts as True Negative. Meanwhile, when ischemic stroke data is incorrectly diagnosed as non-ischemic stroke, it counts as False Negative. Lastly, it counts as False Positive when non-ischemic stroke data is incorrectly diagnosed as ischemic stroke.

Using the rule of the confusion matrix, we can examine the performance of combined CNN+KM clustering based on kernel using the formulas that were listed in Table III.

TABLE III
PERFORMANCE MEASURE

Performance measure	Formula
Accuracy	(TP + TN) / (TP + FP + TN + FN)
Sensitivity	TP / (TP + FN)
Precision	TP / (TP + FP)
Specificity	TN / (TN + FP)
F1-Score	(2 × Sensitivity × Precision) / (Sensitivity + Precision)

III. RESULTS AND DISCUSSION

As we have described in the previous section, the results of the last 2D-convolutional layer in CNN architecture were used as the new input for KM clustering based on the kernel in the vector form.

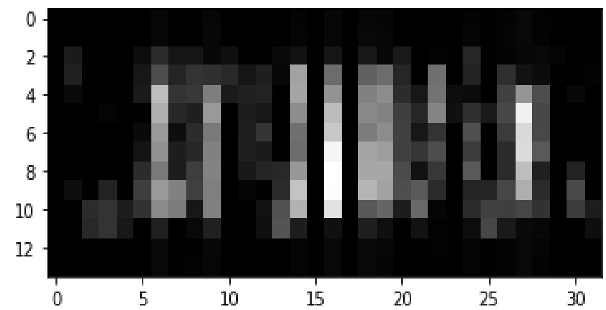

Fig. 5 The result of the last convolutional layer for an image in Figure 1

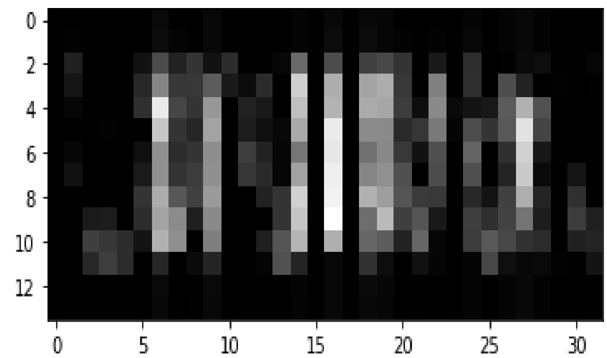

Fig. 6 The result of the last convolutional layer for image in Figure 2

However, the convolutional result of Figure 1 in image form can be seen in Figure 5. As a comparison, we also provided the convolutional result of Figure 2 in image form, which can be seen in Figure 6. It might look similar, but they are different because they come from different images.

Because every image 152 x 152 x 1 pixel became a vector with length 6272, we obtained two matrices, each from class ischemic and non-ischemic strokes. Table IV below provided the matrix input from a class ischemic stroke, while Table V provided the matrix input from a class non-ischemic stroke. Each column represents the image's feature map after passing the convolutional operations.

TABLE IV
MATRIX INPUT FOR ISCHEMIC STROKE (CLASS 1)

Data	f1	f2	…	f6271	f6272
x1	0.139936	0	…	0.045084	0.051006
x2	0.153615	0	…	0.258974	0.633877
…	…	…	…	…	…
x148	0.211017	0	…	0.143986	0.172800
x149	0.439847	0	…	0.980306	5.199656

TABLE V
MATRIX INPUT FOR NON-ISCHEMIC STROKE (CLASS 0)

Data	f1	f2	…	f6271	f6272
x1	0.176003	0	…	0.107532	0.073519
x2	0.158362	0	…	0.089604	0.091670
…	…	…	…	…	…
x184	0.187925	0.013181	…	0.272290	0.502731
x185	0.478123	0	…	1.752787	9.493455

Using both of those matrices in KM clustering based on RBF kernel on 5-fold cross-validation, the performance of this combined method is evaluated, which is shown in Table. VI. Given several RBF kernel parameters σ, it was obtained that $\sigma = 5\times10^{-2}$ delivered the highest performance measure.

TABLE VI
THE PERFORMANCE OF 5-FOLD CROSS-VALIDATION OF CNN

σ	Accuracy	Sensitivity	Precision	Specificity	F1- Score
10-8	97.00	97.96	96.00	96.08	96.97
10-4	96.50	97.94	95.00	95.15	96.45
10-3	96.50	96.97	96.00	96.04	96.48
5×10^{-2}	**97.50**	**97.98**	**97.00**	**97.03**	**97.49**
10-1	95.50	95.96	95.00	95.05	95.48
1	95.50	95.05	96.00	95.96	95.52
10	95.50	94.17	97.00	96.91	95.57
10^2	96.50	96.97	96.00	96.04	96.48
10^3	95.00	95.00	95.00	95.00	95.00
10^4	95.50	95.05	96.00	95.96	95.52

Therefore, we evaluated several k values in k-fold cross-validation such as k = 3, 5, 7, 9, and 11 to see its behavior. As the results, we can see in Table VII that the performance of our proposed method was increasing in line with the increase in the value of k.

TABLE VII
THE PERFORMANCE OF K-FOLD CROSS-VALIDATION OF CNN

k	Accuracy	Sensitivity	Precision	Specificity	F1-Score
3	96.00	96.00	96.00	96.00	96.00
5	97.50	97.98	97.00	97.03	97.49
7	97.00	97.00	97.00	97.00	97.00
9	98.50	98.99	98.00	98.02	98.49
11	**99.00**	**100.00**	**98.00**	**98.04**	**98.99**

We obtained the best results when 11-fold cross-validation and RBF kernel parameter $\sigma = 5\times10^{-2}$ is used. For the computational time that our proposed method needs, it takes less than 10 seconds to pass the dataset to the CNN model and 71 ± 0.098 seconds on average for running the k-fold cross-validation on KM clustering based on RBF kernel.

Our proposed method is more efficient in time compared to the only used deep learning CNN model to test the model after training it with the number of epochs and batch sizes. The performance measures, especially accuracy, also compete with several state-of-art deep learning methods.

IV. CONCLUSION

Detecting the ischemic stroke from the MRI image is not an easy task. Considering the deep learning method, especially CNN, as the reliable method in image classification, we proposed an efficient and robust method that combined CNN with KM clustering based on kernel. Our experiments used the ischemic dataset from Cipto Mangunkusumo (RSCM) Hospital, Indonesia. Thus, this research proposes a method that is more efficient in time computation and delivers competing accuracy with the other deep learning methods.

ACKNOWLEDGMENT

This research was financially supported by University of Indonesia, with a PUTI 2022 grant scheme. The authors are thankful for the kindness of the medical octor at the Department of Radiology, Cipto Mangunkusumo Hospital, in providing the ischemic stroke MRI image datasets.

REFERENCES

[1] T. D. Musuka, S. B. Wilton, M. Traboulsi, and M. D. Hill, "Diagnosis and management of acute ischemic stroke: speed is critical," *CMAJ: Canadian Medical Association journal (journal de l'Association medicale Canadienne)*, vol. 187(12), pp. 887–893, September 2015.

[2] E. S. Donkor, "Stroke in the 21st century: A snapshot of the burden, epidemiology, and quality of life," *Stroke research and treatment*, vol. 2018(3238165), pp. 1–10, November 2018.

[3] T. Savić, G. Gambino, V. S. Bokharaie, H. R. Noori, N. K. Logothetis, and G. Angelovski, "Early detection and monitoring of cerebral ischemia using calcium-responsive MRI probes," *Proceedings of the National Academy of Sciences*, vol. 116(41), pp. 20666–20671, October 2019.

[4] H. J. Audebert and J. B. Fiebach, "Brain Imaging in Acute Ischemic Stroke—MRI or CT?" *Current Neurology and Neuroscience Reports*, vol. 15(6), pp. 1–6, February 2015.

[5] O. Maier, M. Wilms, J. v. d. Gablentz, U. Krämer, and H. Handels, "Ischemic stroke lesion segmentation in multi-spectral MR images with support vector machine classifiers," *Proceedings of SPIE-The International Society for Optical Engineering*, vol. 9035(903504), March 2014.

[6] S. Sabut, K. S. Asit, P. K. Biswal, and S. Sahoo, "Segmentation and classification of ischemic stroke using optimized features in brain MRI," *Biomedical Engineering Applications Basis and Communications*, January 2018.

[7] N. D. Forkert, T. Verleger, B. Cheng, G. Thomalla, C. C. Hilgetag, and J. Fiehler, "Multiclass support vector machine-based lesion mapping predicts functional outcome in ischemic stroke patients," *PLoS ONE*, vol. 10(6), June 2015.

[8] Z.Rustam , D.A.Utami, R. Hidayat, J.Pandelaki, A.W.Nugroho, "Hybrid preprocessing method for support vector machine for classification of imbalanced cerebral infarction datasets", *International Journal on Advanced Science, Engineering and Information Technology,*Volume 9, Issue 2, Pages 685 – 691, 2019.

[9] G.Saragih, Z. Rustam,"Comparison support vector machines and K-nearest neighbors in classifying Ischemic stroke by using

convolutional neural networks as a feature extraction”, *ACM International Conference Proceeding Series*, 2021.

[10] N. Rajini and R. Bhavani, “Automatic detection and classification of ischemic stroke using k-means clustering and texture features,” *In book: Emerging Technologies in Intelligent Applications for Image and Video Processing*, January 2016.

[11] F. Aboudi, C. Drissi, and T. Kraiem, “Brain ischemic stroke segmentation from brain MRI between clustering methods and region-based methods,” *In book: Heart Failure*, pp. 144–154, January 2019.

[12] A. Vupputuri, S. Ashwal, B. Tsao, and N. Ghosh, “Ischemic stroke segmentation in multi-sequence MRI by symmetry determined superpixel based hierarchical clustering,” *Computers in Biology and Medicine*, vol. 116, January 2020.

[13] Q.S.Setiawan, Z.Rustam, A.A. Sa'id, W.Sadewo,FNovkaniza,”Fuzzy C-Means-Grey Wolf Optimization for Classification of Stroke”,*International Conference on Decision Aid Sciences and Application*, DASA 2021, 2021, pp. 971–975.

[14] G.S.Saragih,Z.Rustam,D.Aldila,R.E.Yunus,J.Pandelaki,”Ischemic Stroke Classification using Random Forests Based on Feature Extraction of Convolutional Neural Networks”, *International Journal on Advanced Science, Engineering and Information Technology*, 2020, 10(5), pp. 2177–2182.

[15] S. Winzeck, A. Hakim, R. McKinley, J. Pinto, V. Alves, C. Silva, M. Reyes, “ISLES 2016 and 2017-benchmarking ischemic stroke lesion outcome prediction based on multispectral MRI,” *Frontiers in neurology*, vol. 9(679), pp. 1–20, September 2018.

[16] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Sahl, "Measuring the effects of data parallelism on neural network training," *Journal of Machine Learning Research*, vol. 20, pp. 1–49, July 2019.

[17] Z. Xiao, R. Huang, Y. Ding, T. Lan, R-F. Dong, Z. Qin, X. Zhang, W. Wang, “A deep learning-based segmentation method for brain tumor in MR images,” *IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*, pp. 1–6, October 2016.

[18] A. F. M. Agarap, “An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification,” *arXiv preprint arXiv.* 1712.03541v2, February 2019.

[19] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *arXiv preprint arXiv*. 1901.06032, 2019.

[20] C. E. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv.* 1811.03378v1, November 2018.

[21] S. P. Lloyd, “Least squares quantization in PCM,” IEEE Transactions on Information Theory, vol. 28(2), pp. 129–137, March 1982.

[22] C. M. Bishop, “*Pattern recognition and machine learning,”* New York: Springer, 2006.

[23] N. Cristianini, J. Shawe-Taylor, *“An introduction to support vector machines and other kernel-based learning methods,”* Cambridge: Cambridge University Press, 2014.

[24] V. N. Vapnik, *“Statistical Learning Theory,”* New York: Wiley, 1998.

[25] M. Welling, *“Kernel k-means and spectral clustering,”* 2013.

[26] L. Liu, B. Shen, and X. Wang, “Research on kernel function of support vector machine,” *In: Huang YM., Chao HC., Deng DJ., Park J. (eds) Advanced Technologies, Embedded and Multimedia for Human-centric Computing. Lecture Notes in Electrical Engineering*, vol. 260, pp. 827–834. Springer, Dordrecht, 2014.

[27] Q. Yin, R. Zhang, and X. Shao, “CNN and RNN mixed model for image classification,” *MATEC Web of Conferences*, vol. 277(4), January 2019.

[28] Aditi, M. K. Nagda, and E. Poovammal, “Image classification using a hybrid LSTM-CNN deep neural network,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8(6), pp. 1342–1348, August 2019.

[29] B. Sugg, “Convolutional support vector machines for image classification,” M.S. thesis, Dept. Computer Science, the University of Exeter, Exeter, England, 2018.

[30] M. Copur, B. M. Ozyildirim, and T. Ibrikci, “Image Classification of Aerial Images Using CNN-SVM,” *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Adana, 2018, pp. 1–6.