Recursive Journal of Informatics
Vol. 3, No. 1, March 2025

p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

Combination of Genetic Algorithm and Spiking Neural Network Leaky Integrate-And-Fire Model in Analyzing Brain Ct Scan Image for Stroke Disease Detection

Fabian Dominggus Eka Boro[1], Endang Sugiharti[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract. Stroke is a condition where there is impaired brain function due to lack of oxygen caused by blockage, breakdown, or blood clots inside brain. Diagnosis of stroke is usually based on symptoms, but symptoms are not always the correct measure. In examining a stroke, the most common way to examine a patient is to perform a CT scan of the brain.
Purpose: This study was conducted with the aim of predicting brain scan images consisting of normal brain, ischemic stroke brain, and hemorrhagic stroke brain. It is also to understand how an algorithm works to recognize and predict an image.
Methods/Study design/approach: The image data is trained using machine learning algorithm of neural network, specifically spiking neural network (SNN) using leaky Integrate-and-Fire (LIF) method, which practices the biological performance of human nerves. SNN offers an alternative way of a computational algorithm that mimics the workings of the human brain, especially the nerves in the brain at a low computational cost. In addition, this research optimizes SNN parameters using genetic algorithm (GA). GA is proven to be a successful optimization algorithm from many sources. GA is performed after going through the process in the SNN LIF algorithm, then the parameters in SNN are entered into the algorithm operations in GA until it reaches the most optimal parameter value. Although it requires a large amount of computational time and cost, combining it with SNN will be precise and less labor-intensive.
Result/Findings: Calculation of accuracy results in this study using confusion matrix, conducted on SNN test with LIF method resulted in 90.27%. While with parameter optimization with GA, the final result of the SNN LIF algorithm produces 96.3% accuracy.
Novelty/Originality/Value: This study was conducted to predict stroke disease with human brain images as training data, using the SNN LIF model to train the model and identify patterns that help in predicting stroke risk. For comparison, this research also uses optimization of the model using GA which is useful for determining the core parameters in the training process of the SNN LIF model.

Keywords: Spiking Neural Network, SNN, Leaky Integrate-and-Fire, LIF, Genetic Algorithm, stroke, image detection, deep learning
Received April 17, 2024 / **Revised** May 14, 2024 / **Accepted** March 27, 2025

INTRODUCTION

The field of modern computer science has gone through many stages of development. All aspects of life have come to rely on computing systems, from work, transportation, education, development, health, and more. This is related to the development of machine learning and deep learning algorithms that have created an intelligence object that has been increasingly able to help human life in the last 100 years[1]. These rapid developments have also managed to completely change the way people learn, work and live. Jobs have also changed following the advancement of computer development, especially in the medical field.

Stroke is a condition of impaired brain function due to reduced blood flow to the brain. A stroke occurs when a blood vessel in the brain becomes blocked or ruptured. As a result, part of the brain does not get the blood supply that carries the necessary oxygen, resulting in brain cell damage [2]. The high incidence of stroke increases the need for information on the causes and possible ways of early detection, prediction, and prevention. The development of the field of computer science, especially in machine learning and deep

[1]*Corresponding author.
Email addresses: fabian.boro.fb@students.unnes.ac.id (Boro)
DOI: 10.15294/rji.v3i1.3492

learning to perform these tasks has become very important in relation to the needs of the times. Machine learning can be developed to solve various problems in various fields[3].

In a study [4], stroke is the leading cause of death in China and Japan, with high systolic blood pressure being the main factor causing stroke. In another study, in [5]'s research on people with stroke disease showed that the career age range (18-65) experienced many physical, emotional, and psychosocial challenges. In his report, 44% faced stress and depression, 43% were unable to return to work, 28% rated their quality of life as poor.

The high incidence of stroke increases the need for information on the causes and possible ways of early detection, prediction, and prevention. The development of the field of computer science, especially in machine learning and deep learning to perform these tasks, has become very important due to the needs of the times. based on research belonging to [3] that machine learning can be developed to overcome various problems in various fields, especially in this medical case.

In other research in its development, [6] uses Naïve Bayes as the main method in identifying stroke disease symptoms, by determining the Naïve Bayes Classifier value for each class created, and calculating the probability value with a predetermined formula. Proven to get results with an accuracy value of 80%. The disadvantages of using this artificial method cannot be separated from the certainty of the accuracy of CT scan or MRI results. Other research conducted by [7] using K-Nearest Neighbor (K-NN) for stroke disease classification by classifying objects based on learning data closest to the object with the euclidean distance equation. K-NN evaluation was found with the highest accuracy value of 93.54%. This is a supporting factor where the importance of computer science is one of the fields of science that can be useful both in the health sector.

In the use of deep learning, especially in the health sector, data is needed to process a data model algorithm. Data that is commonly used in processing is called Big Data. Big Data is often related to large-scale data, real-time streaming, and complex data generated by various sources such as environmental sensors, GPS signals, satellites, social media, or smart phones [8]. Scanned image data from the lab is also part of Big Data itself. Image data is also data that can be used to train with machines. In modern industry, the level of intelligence, precision and integration is getting higher and higher [9]. Therefore, image data can be processed through certain processing. One of the uses of images as Big Data for training is corn kernel detection in the field using image processing research [10]. The image data is trained with certain Machine Learning algorithms to evaluate the loss of corn kernels during harvest in the field.

One of the Machine Learning that is often used is Neural Network. Neural Network (NN) is a machine learning algorithm that applies the performance functions of human nerves, and is applied to many sectors that rely on computational methods [11]. One of them is in research on predicting drug-disease interactions by relying on deep learning with the Similarity Network Fusion-Neural network (SFN-NN) [12].Another study, [13] using 2 Artificial Neural Network models to predict mud loss in natural fractures and induced fractures.

NN has many models that are a development of the NN model itself. One of the models of NN is Spiking Neural Network (SNN). This algorithm has a personal modeling methodology that is known to be superior when modeling temporal data [14]. Another approach related to the use of SNN is using the Leaky Integrate-and-Fire (LIF) model. According to [15], the LIF model uses a method whose concept is simple, because it mimics the way biological neuron cell membranes work. Based on [15]'s research, it also provides a statement that this model design can simplify circuits, reduce production costs, and increase integration density. But it still has some component space that can be optimized with other algorithm models.

The uses of the LIF model SNN algorithm still has the disadvantage that it is difficult to determine the parameter value (the value that determines the performance of the model) so that it requires optimization that improves accuracy and processing speed [16]. One of the machine learning algorithms that can perform an optimization includes Genetic Algorithm (GA). GA is proven to increase the efficiency of Neural Network performance [17]. GA is also used by [18] as feature selection in music genre classification using the Random Forest method. Feature selection using GA is done to reduce data dimensions and irrelevant features. In this study, the highest accuracy result was 67%, after several tests were carried out, namely classifying with all features (62%), classifying with normalized data (59%), classifying with original data using the AG feature selection stage (64%), and classifying on normalized data with feature selection using GA (67%).

With that note, it is stated that GA can have an effect on improving the performance of an algorithm so that it can be combined with any algorithm. Therefore, this research will predict stroke disease using SNN LIF model and to be combined with AG as well in order to optimize the algorithm in this study.

METHODS

The method used in this research is a combination of GA which is used as optimization in combination with the SNN LIF algorithm. The purpose of optimization using GA is to obtain the most relevant training model parameters for the algorithm model to be run so as to improve accuracy and efficiency, in this case, SNN with LIF model. The development of this proposed method is based on a combination of several previous studies with slight modifications, resulting in a method that is appropriate for research.

The SNN method used is based on previous studies [14][15][19][20][21][22][23], then combined with GA with some considerations. The method is described with a flowchart so that it can be clearly understood step by step, listed in figure 1.

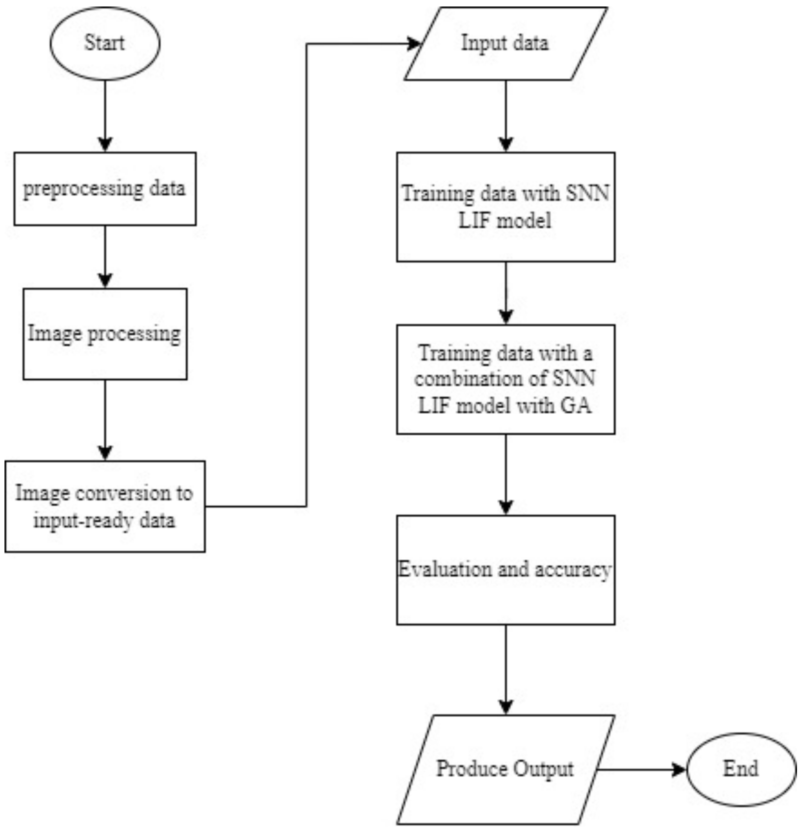

Start
preprocessing data
Image processing
Image conversion to input-ready data
Input data
Training data with SNN LIF model
Training data with a combination of SNN LIF model with GA
Evaluation and accuracy
Produce Output
End

Figure 1. Flowchart of research

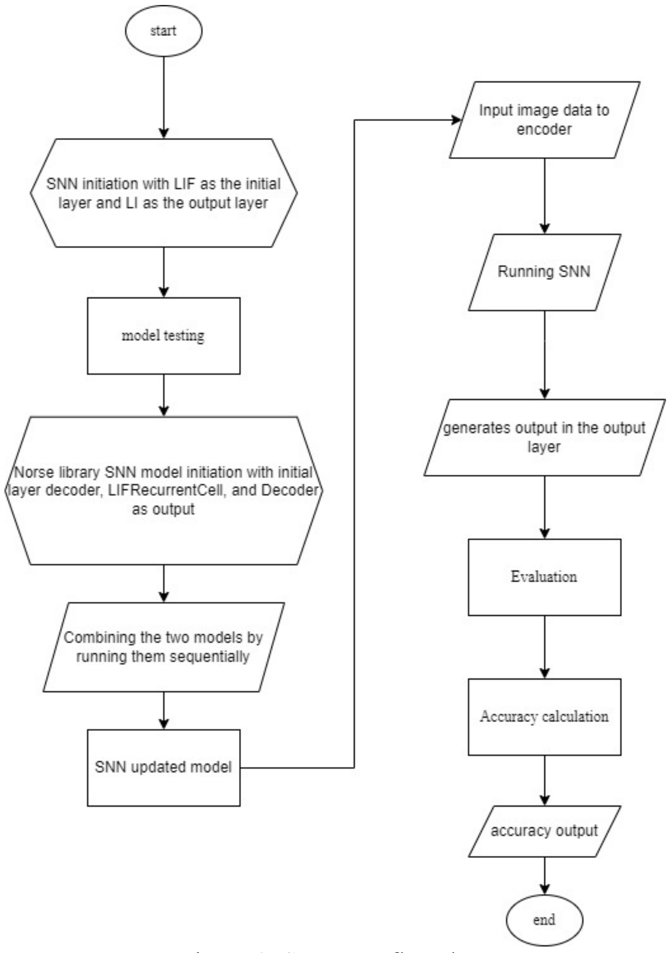

start
SNN initiation with LIF as the initial layer and LI as the output layer
model testing
Norse library SNN model initiation with initial layer decoder, LIFRecurrentCell, and Decoder as output
Combining the two models by running them sequentially
SNN updated model
Input image data to encoder
Running SNN
generates output in the output layer
Evaluation
Accuracy calculation
accuracy output
end

Figure 2. SNN LIF flowchart

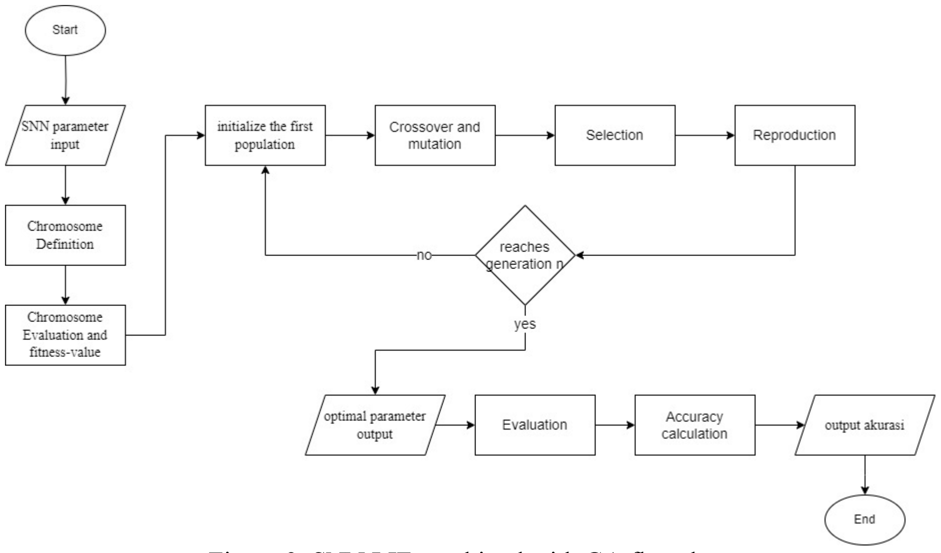


Figure 3. SNN LIF combined with GA flowchart

Dataset

Data is sourced from the internet, namely from the [24]. The datasets used in this study are CT scan images of the brain consisting of hemorrhagic stroke patients, ischemic stroke patients, and patients with normal brains. The dataset is collected from various authors who are credited with uploading certain data as material for training. The dataset detail can be seen in Table 1.

Tabel 1. Dataset Description

Type	Description	Source
Normal	consists of normal brain CT scan image data	[25], [26], [27]
ischemic stroke patients	consists of CT scan image data of ischemic stroke brain	[28], [29], [30]
hemorrhagic stroke patients	consists of CT scan image data of hemorrhagic stroke brain	[31], [32], [33]

Data Preprocessing

Starting from data preprocessing based on Figure 1 that sorts patient data because it is still random and unbalanced from the source by oversampling and undersampling the data [34][35]. Each patient's CT scan is collected into a total of 30 images, so that each patient's data has the same amount and makes it easier to process image data in the training model. Then image processing is given such as rotation, dimensional change, and removal of blemishes in the image. The addition of augmentation as a data multiplication method [36][37].

Image Processing

After loading the image data, the image is resized to ensure consistency. This step involves image resize and conversion to grayscale format. Image resizing uses a Python function that utilizes the image module of the Python Imaging Library (PIL). Image is checked whether it is square or not. To avoid the image shape becoming sprawling, cropping is done to make it square by cutting the top and bottom. Image is then resized according to the specified target size (128 x 128 pixels) using the Lanczos method for optimal interpolation in the context of the image, can be seen in Figure 4.

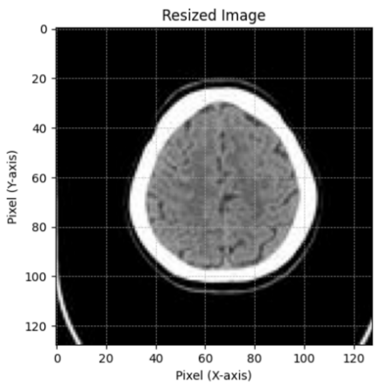


Figure 4. Image after Resize

After resizing, the image is converted to grayscale and saved in JPG format in the corresponding output folder. The choice of grayscale representation aims to simplify binarization, which has an output of 0 or 1 [38]. This is in line with the approach required by SNN. After the repair, the images are then repaired one by one. As can be seen in Figure 5, the brain image has blotches that can affect the model training process.

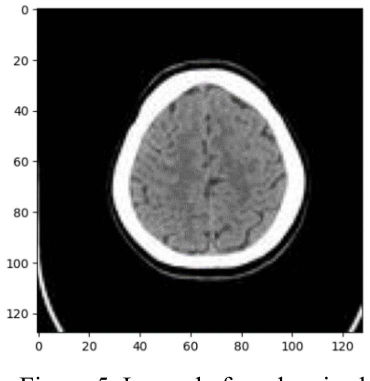


Figure 5. Image before denoised

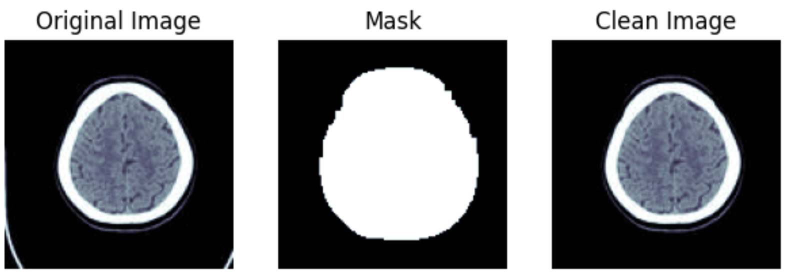
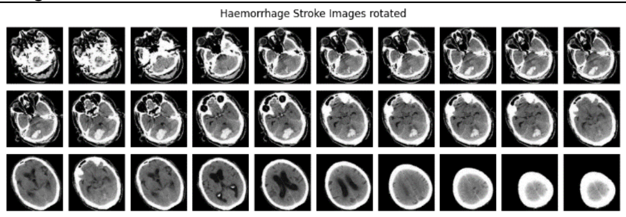
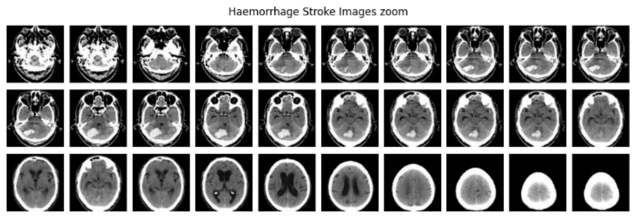
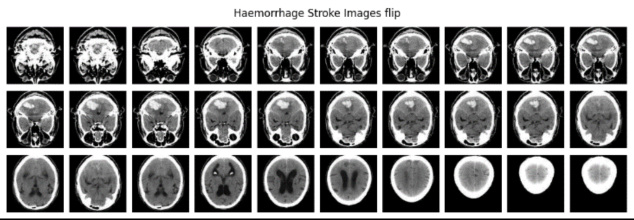


Figure 6. Image after denoised

It is known that the stain is at the edge of the image, so the morphological dilation creates an image segmentation where if a pixel is between the origin and the edge of a 3x3 square, it belongs to the same class. The original idea of this function was to first separate the background, the edges of the brain image, and the brain image itself. By ignoring the background, and clarifying the image by creating a mask of the edges, we can perform the morphological dilation required to remove blotches in the image. In this case, the dilation is done in the black part and ignores the image in the mask, as can be seen in Figure 6.

At this stage, the image is given additional processing to improve the training level of machine learning. Image augmentation has been confirmed as an effective and efficient way to provide machine learning and deep learning training performance especially on limited datasets[37]. Augmentations added are rotation, magnification, and mirroring. Using the PIL library, it can perform all three augmentations. The augmentation results are listed in Table 2.

Tabel 2. Image Augmentation

Augmentation	Images
rotation	
magnification	
mirrorization	

Data Splitting
After going through some data processing, the next step is to divide the data into training data and testing data. Training data is used as data used to train the model, while test data is used to test the performance of the model that has been trained. The division is done with a ratio of 70% to 30%.

Image Conversion

In preparing the input format of the SNN model, the data needs to be prepared into a spike-time representation, which is temporal data that considers time or sequence information for decision making. SNN enables temporal data representation through the use of spike-time representation (spatial and temporal data), which means that the time of spike appearance has meaning about neuron activity and time sequence. This enables the processing of important temporal information in brain images, especially for the analysis of functional data such as Functional Magnetic Resonance Imaging (fMRI). By taking into account spatial and temporal interactions in complex brain image data, it enables detection of more subtle and detailed patterns in brain structure and activity.

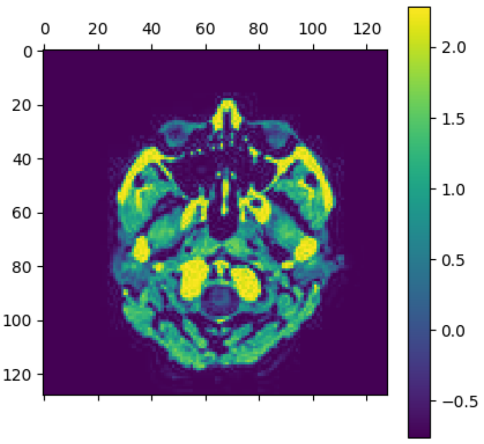


Figure 7. Spike-time Representation

In Figure 7, the graph on the right is a measure of the spike-time representation of a CT scan image of a brain categorized as "Normal". Each pixel in the image is represented by a neuron, and the color of each pixel on the graph indicates the time of spike appearance (neuron activity) within a certain time interval.

To use the SNN model, it is necessary to process data represented by spikes or peaks that occur at specific times. However, common data sets for machine learning. Using the Norse library to decode the data into a spike-time representation, the results of the image conversion are shown in Figure 8.

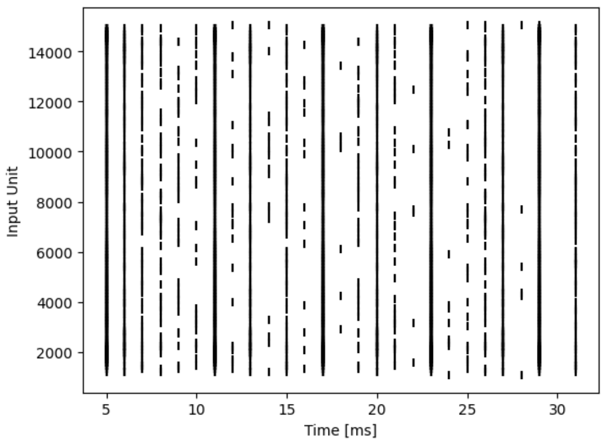


Figure 8. Spike-time Representation after Encoding

Training with SNN LIF

It begins with the construction of an SNN model defined with one recurrent reconnected layer and one output readout layer. The model used has two types of neuron cells: LIF Recurrent Cell as the initial layer and with LI as the output layer. The construction of this model as a model that tests the neuron voltage when it reaches a certain threshold. By initiating the initial layers, here is the description of the first layer with LIF as the initial layer and LI as the output layer, seen in Table 3.

Tabel 3. First Type Model Layer

SNN Layer Model	Type	Layer contents
Input Layer	Input	Input_features
(L1)	LIFRecurrentCell (hidden_layer)	Alpha Value Threshold
(fc_out)	Linear	Output_features
(out)	LICell	tau_syn_inv tau_mem_inv v_leak dt
Output_Layer	Output	Output_features

Then the model is tested with one of the image data so that it can assess whether the model can work or not. However, the model will be combined again with the encoder and decoder functions from the norse library so that the layer arrangement will be changed again. The details are shown in Tabel 4.

Tabel 4. Second Type Model Layer

SNN Layer Model	Type	Parameter
Layer Encoder	Encoder	SpikeLatencyLIFEncoder timesteps
SNN	L1(LIFRecurrentCell)	Input features Hidden features tau_syn_inv tau_mem_inv v_leak dt Alpha Value Threshold
	Fc_out (Linear)	Input hidden features Output features
	Out (LICell)	tau_syn_inv tau_mem_inv v_leak dt
Layer Decoder	Decoder	torch.nn.functional.log_softmax

Training SNN LIF Combined with GA

The weakness of the model used is the difficulty in determining the optimal parameters. By using the AG approach as optimization, the parameters used in the model are changed randomly and training is done again. Training is done as many individuals in 1 population so that individuals are formed with each parameter as a gene. The parameters chosen are only those that are considered to be the distinguishing factor in training. The description of genes in an individual is shown in Table 5.

Tabel 5. Individual Genes in GA architecture

Individual Genes
Hidden features
Output features
Learning Rate
Epoch
Timesteps
Alpha
Batch_Size
Encoder

Then the individual initialization of the above parameters is carried out, by selecting values randomly but determined from a certain range of values. The fitness value of an individual is evaluated from the previous SNN model, which is using the accuracy calculation of the model work. The details can be seen in Tabel 6.

Tabel 6. Range of Random Values of Parameters on Individual Genes

Parameter Name	Random Value Range
T	$20 \le T \le 50$
LR	$0.001 \le LR \le 0.01$
alpha	$40 \le alpha \le 150$
v_th	$0.3 \le v\_th \le 0.7$ (*tensor PyTorch*)
HIDDEN_FEATURES	$50 \le HIDDEN\_FEATURES \le 200$
OUTPUT_FEATURES	3 (constant)
BATCH_SIZE	$8 \le BATCH\_SIZE \le 64$
EPOCHS	$3 \le EPOCHS \le 20$
Encoder	*encode.SpikeLatencyLIFEncoder* *encode.ConstantCurrentLIFEncoder* *encode.PoissonEncoder*

From the initialized individuals, next create a population from the set of individuals. Next, crossover is performed. The crossover used is two-point crossover and produces 2 new individuals. Individuals are then mutated by permutating the values from Table 6 with the specified mutation rate. After that, elitism selection is carried out by selecting individuals with the best fitness value. The work continues to the generation iteration with each generation running a population update. Parameter values for both crossover, mutation, and elitism.

Evaluation

After the training ends, the evaluation process with confusion matrix is performed. This process takes the model that has been trained on the device where the model is evaluated, and the test loader data (test_loader). Then evaluate each test data, then calculate the prediction of the model. The return value is 2 arrays, which contain the true value (label) and the prediction result. Tabel 7 represents the confusion matrix.

Tabel 7. Confusion Matrix

	Positive	Negative
Positive	TP	FP
Negative	FN	TN

RESULT AND DISCUSSION

This study was conducted to predict stroke disease with human brain images as training data, using the SNN LIF model to train the model and identify patterns that help in predicting stroke risk. For comparison, this research also uses optimization of the model using GA which is useful for determining the core parameters in the training process of the SNN LIF model. In determining the accuracy value, confusion matrix is used to identify the accuracy of the model accuracy after the data is divided into 70% training data and 30% test data.

By defining an SNN model with one hidden_layer and recurrently connected with LIF hidden_features neurons and a readout layer with output_features and LI, this model can freely combine the primitive SNN network with the layers of an ordinary torchNN module, making it good for combination with other ANN-based algorithms. Figure 9 is a visualization of the voltage output generated by the recurrent SNN on the example input as a test model, which displays the voltage output at each time in the sequence.

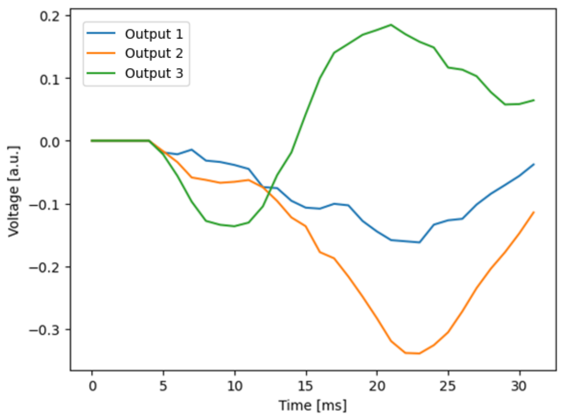


Figure 9. Visualization of First Type Model Layer Test Voltage Output

Figure 9 illustrates the graph of the neuron potential membrane voltage in the model that changes over time. The 3 lines on the graph represent the number of features in the output layer and can change each time the training is re-run. The lines represent the change in voltage over time in those cells. The curved shape of the voltage graph symbolizes the reset process after reaching the threshold. The positive and negative values represent that there are inactive neuron outputs. When it reaches a negative value, the neuron shows an inhibitory or suppressive condition so that the neuron is not active. Conversely, positive voltage values indicate that there is activation in the neuron.

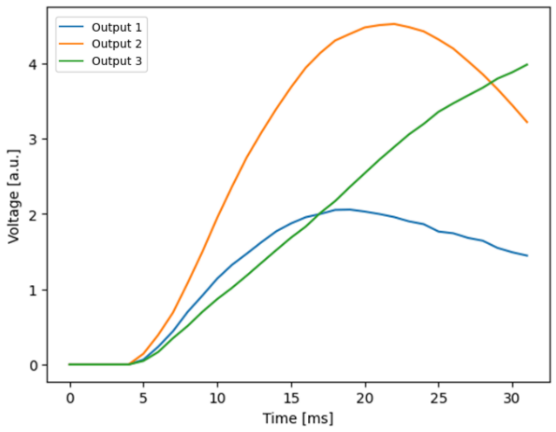


Figure 10. Visualization of Voltage Output of Second Type Model Layer Experiment

Figure 10 is slightly different from Figure 9, which is found in the shape of the line or voltage value. The curved line shape symbolizes that LI works well in this model by responding to the stimulus parameters on each input. The positive voltage values all indicate that the neurons perform the activation process all and none are extinguished / inactive.

Next is to run the model training with train and test data, using the traditional PyTorch library commonly used in training Neural Network algorithms. The training was run by iterating 5 times with the training data iterated using the progress bar from the tqdm library. For each batch of training data, the gradient optimizer was reset, and model predictions were obtained by passing the model input data. Backpropagation on the model and optimizer was used to update the model parameters, and the loss value of each training for each batch was recorded. A graph of the Loss value and accuracy measured per batch and epoch is shown in Figure 11.

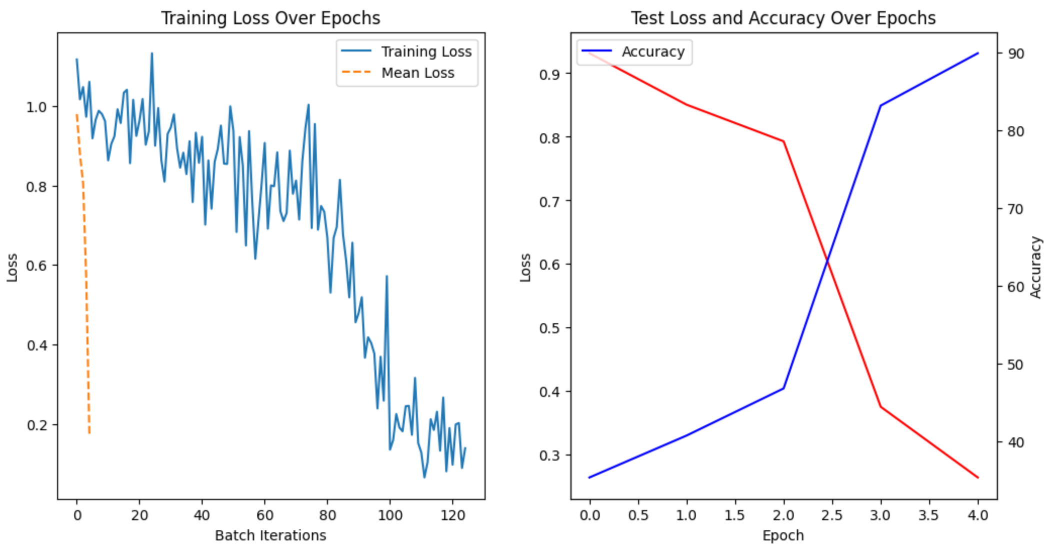


Figure 11. Training Graph, Loss Value, and Accuracy of each Batch Iteration

The total accuracy obtained from the model with the given parameters is 90.27%. The calculation of accuracy in the model uses a common approach to classification tasks, namely by calculating the percentage of correct predictions, True Positive (TP) and True Negative (TN) from the total samples in the test set.

$$\frac{number\ of\ correct\ predictions}{total\ test\ data} \times 100\% \tag{1}$$

This accuracy measurement does not involve the confusion matrix, but relies on comparing the predictions with the actual truth labels.

Then the SNN LIF parameter optimization training is carried out using GA. The training starts with population and generation iterations. In this function, if there is no improvement and gets the same result, then the generation iteration is stopped. This is presented in Figure 12. Table 8 shows the optimal parameters resulting from GA iteration.

Tabel 8. Optimal Parameter

Optimal Parameters	
T	25
LR	0.0026
alpha	52
v_th	tensor(0.6800)
HIDDEN_FEATURES	147
OUTPUT_FEATURES	3 (constant)
BATCH_SIZE	45
encoder	*SpikeLatencyLIFEncoder*

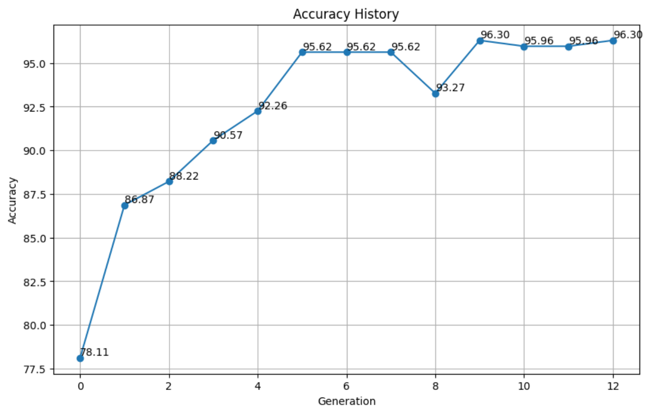


Figure 12. Accuracy Results for each Generation

Figure 12 shows the highest accuracy result of 96.3%. The evaluation that runs in each generation is done directly with the confusion matrix. The use of the Seaborn library helps visualize the confusion matrix generated by the model. The resulting visualization is a 3×3 matrix heatmap with an output of 3 classes, where the diagonal elements show the number of correct predictions for each class. Figure 13 shows the visualization of the confusion matrix.

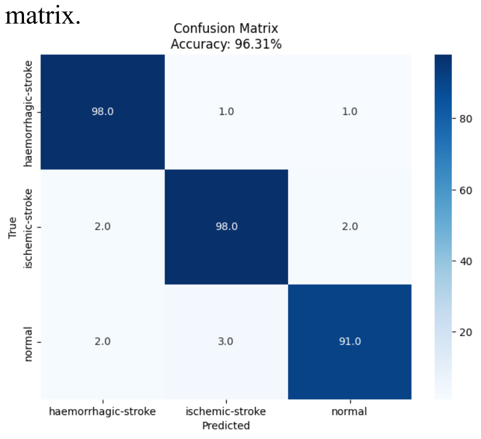


Figure 13. Confusion Matrix Visualization

Table 9 shows the description of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) based on the matrix above.

Tabel 9. Confusion Matrix Description

	Predicted 0	Predicted 1	Predicted 2
Actual 0	TN (91)	FP (1)	FP (1)
Actual 1	FN (2)	TP (98)	FP (2)
Actual 2	FN (2)	FP (3)	TP (91)

CONCLUSION

The application of SNN algorithm with LIF method and optimization using GA to predict stroke disease from brain CT scan image can be done. SNN successfully trains image data with the help of encoders from the Norse library to convert image data that has been transformed into tensor data into input data by providing information on spikes or peaks that occur at certain times. Based on the results of the research conducted, an increase in the results of each model was obtained. The model used from SNN LIF method only, which gets 90.27% accuracy. The next model after optimized using GA to get the best parameters. The combination results in an accuracy of 96.3% and an increase of 6.03% from the SNN algorithm before optimization.

REFERENCES

[1] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. Peter Campbell, "Introduction to machine learning, neural networks, and deep learning," *Transl Vis Sci Technol*, vol. 9, no. 2, 2020, doi: https://doi.org/10.1167/tvst.9.2.14.

[2] Q. Wang *et al.*, "Normobaric hyperoxia therapy in acute ischemic stroke: A literature review," *Heliyon*, vol. 10, no. 1, p. e23744, Jan. 2023, doi: https://doi.org/https://doi.org/10.1016/J.HELIYON.2023.E23744.

[3] K. Uchida *et al.*, "Development of Machine Learning Models to Predict Probabilities and Types of Stroke at Prehospital Stage: the Japan Urgent Stroke Triage Score Using Machine Learning (JUST-ML)," *Transl Stroke Res*, vol. 13, no. 3, pp. 370–381, Jun. 2022, doi: https://doi.org/10.1007/s12975-021-00937-x.

[4] J. Cao, E. S. Eshak, K. Liu, A. Arafa, H. A. Sheerah, and C. Yu, "Age-period-cohort analysis of stroke mortality attributable to high systolic blood pressure in China and Japan," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: https://doi.org/10.1038/s41598-021-98072-y.

[5] U. Gopaul *et al.*, "Age-specific information resources to address the needs of young people with stroke: a scoping review protocol," *Syst Rev*, vol. 11, no. 1, Dec. 2022, doi: https://doi.org/10.1186/s13643-022-02147-4.

[6] F. Karim, G. W. Nurcahyo, and S. Sumijan, "Sistem Pakar dalam Mengidentifikasi Gejala Stroke Menggunakan Metode Naive Bayes," *Jurnal Sistim Informasi dan Teknologi*, pp. 221–226, Aug. 2021, doi: https://doi.org/10.37034/jsisfotek.v3i4.69.

[7] Zuriati Z and Qomariyah N, "Klasifikasi Penyakit Stroke Menggunakan Algoritma K-Nearest Neighbor (K-NN)," *Jurnal Sistem dan Teknologi Informasi*, vol. 1, no. 1, pp. 1–8, 2023, doi: https://doi.org/10.25181/rt.v1i1.2665.

[8] M. T. Huynh, M. Nippa, and T. Aichner, "Big data analytics capabilities: Patchwork or progress? A systematic review of the status quo and implications for future research," *Technol Forecast Soc Change*, vol. 197, p. 122884, Dec. 2023, doi: https://doi.org/10.1016/J.TECHFORE.2023.122884.

[9] W. Yu, "Image processing methods based on physical models," *Results Phys*, vol. 56, p. 107199, Jan. 2024, doi: https://doi.org/10.1016/J.RINP.2023.107199.

[10] X. Liu *et al.*, "Infield corn kernel detection using image processing, machine learning, and deep learning methodologies under natural lighting," *Expert Syst Appl*, vol. 238, p. 122278, Mar. 2024, doi: https://doi.org/10.1016/J.ESWA.2023.122278.

[11] Yu Zhang, J. Zeng, Y. Li, and D. Chen, "Convolutional Neural Network-Gated Recurrent Unit Neural Network with Feature Fusion for Environmental Sound Classification," *Automatic Control*

and Computer Sciences, vol. 55, no. 4, pp. 311–318, Jul. 2021, doi: https://doi.org/10.3103/S0146411621040106.

[12] T. N. Jarada, J. G. Rokne, and R. Alhajj, "SNF-NN: computational method to predict drug-disease interactions using similarity network fusion and neural networks," *BMC Bioinformatics*, vol. 22, no. 1, Dec. 2021, doi: https://doi.org/10.1186/s12859-020-03950-3.

[13] H. H. Alkinani, A. T. T. Al-Hameedi, and S. Dunn-Norman, "Artificial neural network models to predict lost circulation in natural and induced fractures," *SN Appl Sci*, vol. 2, no. 12, Dec. 2020, doi: https://doi.org/10.1007/s42452-020-03827-3.

[14] M. Doborjeh *et al.*, "Personalized Spiking Neural Network Models of Clinical and Environmental Factors to Predict Stroke," *Cognit Comput*, vol. 14, no. 6, pp. 2187–2202, Nov. 2022, doi: https://doi.org/10.1007/s12559-021-09975-x.

[15] T. Guo *et al.*, "Adjustable Leaky-Integrate-and-fire neurons based on memristor-coupled capacitors," *Mater Today Adv*, vol. 12, p. 100192, Dec. 2021, doi: https://doi.org/10.1016/J.MTADV.2021.100192.

[16] P. Arsi and O. Somantri, "Deteksi Dini Penyakit Diabetes Menggunakan Algoritma Neural Network Berbasiskan Algoritma Genetika," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 3, pp. 290–294, Oct. 2018, doi: https://doi.org/10.30591/jpit.v3i3.1008.

[17] P. Abdolghader, F. Haghighat, and A. Bahloul, "Predicting Fibrous Filter's Efficiency by Two Methods: Artificial Neural Network (ANN) and Integration of Genetic Algorithm and Artificial Neural Network (GAINN)," *Aerosol Science and Engineering*, vol. 2, no. 4, pp. 197–205, Dec. 2018, doi: https://doi.org/10.1007/s41810-018-0036-2.

[18] N. Amini *et al.*, "Implementasi Algoritma Genetika untuk Seleksi Fitur pada Klasifikasi Genre Musik Menggunakan Metode Random Forest," *Jurnal Informatika Polinema (JIP)*, no. Vol 9 No 1 (2022), pp. 75–82, 2022, doi: https://doi.org/10.33795/jip.v9i1.1028

[19] H. Zhang, J. Cheng, J. Zhang, H. Liu, and Z. Wei, "A regularization perspective based theoretical analysis for adversarial robustness of deep spiking neural networks," *Neural Networks*, vol. 165, pp. 164–174, Aug. 2023, doi: https://doi.org/10.1016/J.NEUNET.2023.05.038.

[20] L. Y. Niu and Y. Wei, "CIRM-SNN: Certainty Interval Reset Mechanism Spiking Neuron for Enabling High Accuracy Spiking Neural Network," *Neural Process Lett*, 2023, doi: https://doi.org/10.1007/s11063-023-11274-5.

[21] J. B. Aimone, A. J. Hill, W. M. Severa, and C. M. Vineyard, "Spiking Neural Streaming Binary Arithmetic," *Sandia National Laboratories*, 2021, doi: https://doi.org/10.48550/arXiv.2203.12662.

[22] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. S. Maida, "Deep Learning in Spiking Neural Networks," Apr. 2018, doi: https://doi.org/10.1016/j.neunet.2018.12.002.

[23] D. Goodman, T. Fiers, R. Gao, M. Ghosh, and N. Perez, "Spiking Neural Network Models in Neuroscience - Cosyne Tutorial 2022," Sep. 2022, doi: https://doi.org/10.5281/ZENODO.7044500.

[24] radiopaedia, "https://radiopaedia.org/," https://radiopaedia.org/ (accessed January 2nd, 2024).

[25] B. Di Muzio, "Normal brain CT - 60-year-old," Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/normal-brain-ct-60-year-old-2?lang=gb, (accessed January 2nd, 2024).

[26] M. R. Abidin, "Normal CT brain plain," Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/normal-brain-non-contrast-ct?lang=gb, (accessed January 2nd, 2024).

[27] J. Peacock, "Normal Brain Amyvid and FDG PET/CT scans for dementia," Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/normal-brain-amyvid-and-fdg-petct-scans-for-dementia?lang=gb, (accessed January 2nd, 2024).

[28] Y. Sheikh and D. Cuete, "Ischemic stroke CT Scan," Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/ischemic-stroke-2?lang=gb, (accessed January 2nd, 2024).

[29] A. Abdrabou and D. Asadov, "Left middle cerebral artery territory infarct," Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/left-middle-cerebral-artery-territory-infarct-1?lang=gb, (accessed January 2nd, 2024).

[30] A. Haouimi and D. Cuete, “PCA territory infarct,” Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/pca-territory-infarct-1?lang=gb, (accessed January 2nd, 2024).
[31] D. Smith, “Haemorrhagic stroke,” Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/haemorrhagic-stroke-1?lang=gb, (accessed January 2nd, 2024).
[32] Y. Sheikh and A. Goel, “Cerebellar haemorrhage,” Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/cerebellar-haemorrhage-5?lang=gb, (accessed January 2nd, 2024).
[33] M. Rodrigues, “Intracerebral haemorrhage,” Radiopaedia.org. [Dataset]. Available: https://radiopaedia.org/cases/intracerebral-haemorrhage-8?lang=gb, (accessed January 2nd, 2024).
[34] N. Rodríguez, D. López, A. Fernández, S. García, and F. Herrera, “SOUL: Scala Oversampling and Undersampling Library for imbalance classification,” *SoftwareX*, vol. 15, p. 100767, Jul. 2021, doi: https://doi.org/10.1016/J.SOFTX.2021.100767.
[35] C. Vairetti, J. L. Assadi, and S. Maldonado, “Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification,” *Expert Syst Appl*, vol. 246, p. 123149, Jul. 2024, doi: https://doi.org/10.1016/J.ESWA.2024.123149.
[36] J. Desternes, “Data augmentation for medical image analysis in deep learning.” Accessed: Nov. 27, 2023. [Online]. Available: https://www.imaios.com/en/resources/blog/ai-for-medical-imaging-data-augmentation
[37] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, “A Comprehensive Survey of Image Augmentation Techniques for Deep Learning,” *Pattern Recognit*, vol. 137, p. 109347, May 2023, doi: https://doi.org/10.1016/J.PATCOG.2023.109347.
[38] X. Zhou, W. Hong, G. Yang, T. S. Chen, and J. Chen, “An unsolvable pixel reduced authentication method for color images with grayscale invariance,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 9, p. 101726, Oct. 2023, doi: https://doi.org/10.1016/J.JKSUCI.2023.101726.