ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

THE IMPLEMENTATION OF BAYESIAN OPTIMIZATION FOR AUTOMATIC PARAMETER SELECTION IN CONVOLUTIONAL NEURAL NETWORK FOR LUNG NODULE CLASSIFICATION

Kadek Eka Sapta Wijaya[1], Gede Angga Pradipta[2], Dadang Hermawan[3]

[1] Magister Program, Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia
[2,3] Department of Magister Information System, Institut Teknologi dan Bisnis STIKOM Bali, Indonesia

email: 222012029@stikom-bali.ac.id[1], angga_pradipta@stikom-bali.ac.id[2], dadang@stikom-bali.ac.id[3]

Abstract

Lung cancer's high mortality rate makes early detection crucial. Machine learning techniques, especially convolutional neural networks (CNN), play a very important role in lung nodule detection. Traditional machine learning approaches often require manual feature extraction, while CNNs, as a specialized type of neural network, automatically learn features directly from the data. However, tuning CNN hyperparameters, such as network structure and training parameters, is computationally intensive. Bayesian Optimization offers a solution by efficiently selecting parameter values. This study develops a CNN classification model with hyperparameter tuning using Bayesian Optimization, achieving a 97.2% accuracy. Comparatively, Particle Swarm Optimization and Genetic Algorithm methods each resulted in 96.4% accuracy. The research concludes that Bayesian Optimization is an effective approach for CNN hyperparameter tuning in lung nodule classification.

Keywords : lung nodule, deep learning, convolutional neural network, hyperparameter, bayesian optimization

Received: 19-07-2024 | **Revised:** 30-09-2024 | **Accepted:** 12-11-2024
DOI: https://doi.org/10.23887/janapati.v13i3.82467

INTRODUCTION

Cancer ranks as the leading cause of death globally. It can be found in almost all areas of the human body, including the prostate, brain, lungs, breast and stomach. Notably, lung cancer is a common type of cancer with a very high mortality rate [1]. According to the Global Cancer Statistics 2020 report, lung cancer caused more than 10 million deaths, accounting for about 18% of total cancer deaths, and accounted for about 11.4% of the total nearly 19.3 million new cases diagnosed that year [2]. In the early stages, lung cancer symptoms are usually mild and often difficult to diagnose accurately. Because of this, many patients miss the best time for treatment after diagnosis. According to the literature, about 68% of lung nodules can be detected correctly by a single radiologist, and this accuracy rate can be increased to 82% with consultation from a second radiologist [3]. Radiologists use computed tomography (CT) to scan the patient's chest and identify whether the lung nodule is benign or malignant, as lung nodules are small enough to be difficult to see normally [4]. Nodules are generally small in structure and can be less than 30mm in diameter, if a lung nodule exceeds or grows larger than this diameter it can be classified as a malignant mass. Then, nodules that measure between 5 to 30mm are possible to be benign or malignant, this has the possibility that lung nodules can become malignant as the size of the lung nodule increases. The criteria that can be characterized as malignant are shaped like a spike or lobulation, and if benign, it usually has signs of general calcification [5].

There are two approaches used in medical image analysis to classify lung cancer. The first approach uses machine learning algorithms that can learn and classify lung cancer based on features extracted from images. Then the second approach, by designing a deep learning model for lung cancer analysis. Then, various types of deep learning models exist Convolutional Neural Network (CNN) is the most widely used image and video classification model due to its ability to understand data hierarchically. Thus, it can see complex information from existing data. CNN has the advantage of automatically learning and extracting important patterns from images without manual feature coding. Therefore, CNN can classify effectively if the images have the same features [6]. Machine learning researchers

are required to make adjustments to the external configuration of data on the Convolutional Neural Network (CNN) model. Where hyperparameters have network structure and training parameters on CNN that must be optimized to get parameters that produce more accurate and efficient models [7]. Determining the optimal hyperparameters is a time-consuming computational process that affects the performance of the model. Researchers in the last few decades have made tremendous efforts toward machine learning. Important tasks such as data preprocessing, model selection, and hyperparameter adjustment require highly professional expertise. Thus, the complexity of these tasks looks complicated to understand [8].

Loey et al [9] applied Bayesian Optimization as a hyperparameter selection in the Convolutional Neural Network (CNN) model, this study aims to identify COVID-19 patients using X-ray image data. The study's results obtained the accuracy rate of the method is 96%. Then, Azeez et al [10] combined the Convolutional Neural Network (CNN) method with Multi-Resolution Segmentation (MRS) to determine object boundaries in urban environments. This research also applies Bayesian Optimization techniques in finding optimal hyperparameters in segmentation and feature extraction. The results show that the model is computationally efficient, with a training time of less than 25 seconds per epoch. Thus, it can achieve a high level of accuracy and excel in geometric quality, as well as in the extraction and separation of adjacent objects. Furthermore, Zheng et al [11] developed a prediction model for Metabolic Syndrome (MetS) with indicators used from Traditional Chinese Medicine (TCM) and simple biological indicators. Medical record data were collected in this study from 2020 to 2021, which included 2040 patients who visited the outpatient clinic of a traditional Chinese medicine hospital in Guangdong. This study resulted in a new model derived from Bayesian Optimization and eXtreme Gradient Boosting (XGBoost) that allows the BO-XGBoost model to identify nineteen key features covering three categories, namely TCM habits, TCM indicators, and bio-information that have an influence on MetS prediction.

Recently, the use of Bayesian optimization has been widely used in various fields to improve model accuracy. Then, in the field of classification and diagnosis, machine learning models such as Support Vector Machine, k-nearest Neighbor, Naive Bayes, and Decision Tree have been utilized on leaf types [12] and the use of SVM in Rolling Bearing Fault diagnosis [13]. Furthermore, the use of Bayesian optimization as a predictor has been used in various applications including steering angle in the context of ADS [14] and predicted production from one oil well [15]. The previously mentioned research shows that the use of Bayesian Optimization as a method to search for optimal hyperparameters can effectively improve performance and accuracy across a wide range of applications.

METHOD

This study proposes a system that aims to find the optimal hyperparameter using the Bayesian Optimization on Convolutional Neural Network to classify lung nodules into benign, malignant, and normal classes. This research has three stages, namely dataset pre-processing, training phase, and testing phase. Then, this study uses data from The Lung Image Database Consortium (LIDC-IDRI) collection available on The Cancer Imaging Archive website. The dataset used in this study includes 1,018 scans from 1,010 patients about the lungs. Furthermore, the sequence of steps followed systematically in this study is shown in Figure 1.

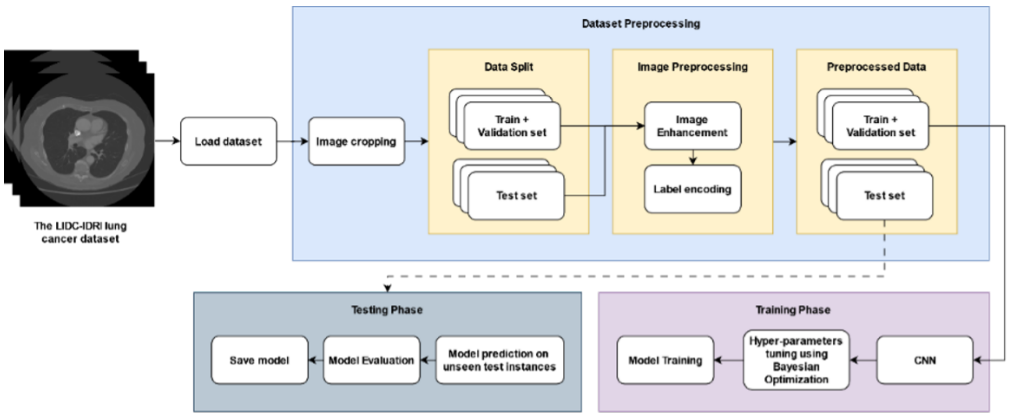

Figure 1. The Proposed Model's Flowchart

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Dataset Preprocessing

At this stage, pre-processing is performed before the data is used for model training, validation, and testing. Therefore, the image size is changed to 256 x 256 x 1 resolution according to the input tensor in the model. Next, the images were randomized to ensure an unbiased training process, and then divided in the proportion of 60:20:20 for training, validation, and testing data. Then, the use of Gaussian Filter and Gabor Filter is applied to image enhancement. The Gaussian Filter serves to reduce noise and produce a smoother image, while the Gabor Filter is used to detect edges and textures, which are essential in feature extraction.

Gaussian filter

Gaussian Filter is widely used for smoothing images in a different way than average filters. The Gaussian filter gives greater weight to pixels closer to the center compared to pixels further away, different from the average filter which gives equal weight to all surrounding values. The size of the Gaussian filter is controlled by σ and its standard deviation. The advantages of the Gaussian filter lie in its soft transitions and distance-based weighting, whereas the mean and median filters that do not consider distance can cause artifacts in the image, especially with larger filters [16]. The formula for the Gaussian Filter can be expressed by the equation (1).

The kernel value at pixels located at a distance of r rows and c columns from the center point are calculated as follows:

$$k_{r,c} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{r^2+c^2}{2\sigma^2}\right) \quad (1)$$

Gabor filter

Gabor Filter has similarities with itself. A Gabor filter $(G(x,y))$ is a highly complex combination of sinusoidal with Gaussian envelopes in the spatial domain. A Gabor filter bank is a collection of multi-channel filters derived from a single parent wavelet through various scales and rotations, thus using a wide range of frequencies in the spatial domain. The orientation of the sinusoidal and Gaussian scatterers is set against the x and (σx and σy). Gabor filters have a very similar system to human visuals and have proven to be very effective in representing and discriminating textures. The two-dimensional Gabor filter uses a Gaussian kernel as modulated by sinusoidal waves in the spatial domain, which can be defined as follows [17]:

$$G(x,y) = \frac{f^2}{\pi\gamma\eta}\exp\left(-(\alpha^2 x'^2 + \beta^2 y'^2)\right)\exp\left(-(2\pi f x')\right) \quad (2)$$

Where:

$$x' = x\,cos\theta + y\,sin\theta \quad (3)$$

$$y' = -x\,sin\theta + y\,cos\theta \quad (4)$$

Each filter usually has a plane shape with a specific frequency bounded by relative Gaussian envelope functions α and β. In an image, it is necessary to extract features through a set of Gabor filters of various orientations and frequencies [17]:

$$f_u = f_{max}/\sqrt{2}^{u-1},\ u = 0, 1, 2, \ldots, U\text{-}1 \quad (5)$$

$$\theta_v = \frac{v-1}{N}\pi,\ v = 0, 1, 2, \ldots, V\text{-}1 \quad (6)$$

f_{max} is the frequency that represents the highest value of γ and the ratio between the center frequency to the sharpness on the Gaussian major axis. The value of η represents the ratio of center frequency to sharpness on the Gaussian minor axis. U is the total number of scales, while V is the number of orientations [17].

Training Phase

The training phase is a critical stage in developing a robust CNN model for lung nodule classification. During this phase, the model learns to identify and distinguish between various features of lung nodules by iteratively adjusting its parameters based on the provided training data.

Convolution neural network

Convolution neural network is an artificial neural network architecture that is often used for classification, detection, and diagnosis of lung nodules using data from computed tomography (CT) in computer vision [1, 18].

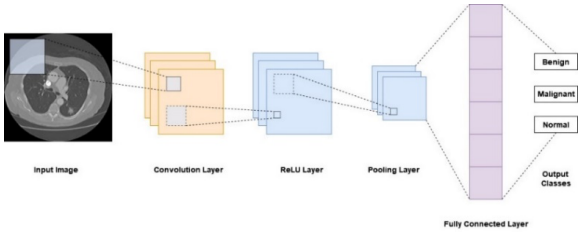

Figure 2. Architecture CNN

In the context of Convolutional Neural Network (CNN), there are several components that play a role in processing and making decisions on image data. These components will be explained in more detail as follows:
The convolution layer is the main element in the CNN architecture that oversees receiving input images and extracting features from them. The convolution process occurs within this layer, producing a new image that reflects the features of the input image [19]. The formula used in the convolution process can be seen in equation (7).

$$Q_{(i,j)} = \left(\sum_{i=1}^{N} I_{(i,j)} \odot F_{(i,j)}\right) + B_j \quad (7)$$

Activation function is a mechanism applied to modify data through the calculation of non-linear functions. In CNN architecture, one of the most used activation functions is Rectified Linear Unit (ReLU). ReLU serves to eliminate negative values in the image and can increase the efficiency of computation time [19]. The formula for the ReLU activation function can be expressed by the equation (8).

$$R(z) = max\ (0, z) \quad (8)$$

Pooling layer is one of the elements in the architecture that serves to reduce the dimension of the feature map and speed up the computation process is pooling. There are two types of pooling that are commonly applied, namely max pooling and average pooling. Max pooling takes the maximum value from each pixel area of the image, while average pooling calculates the average value from each pixel area [19].

Dropout is an effective strategy in neural network algorithms to prevent overfitting is to use a combination of various architectures. Increasing the weight values may cause the time required for the testing process to become longer, which in turn may affect CNN's performance in the evaluation [19].

Fully connected layer is the last stage after convolutional layer and pooling layer, which serves as the final stage to classify the data into classes in the training process. In this layer, the pixel values that were previously a matrix are converted into a one-dimensional format [19].

Softmax layer is a commonly used activation function for data classification by determining the highest probability for each class. The value of the softmax function or the resulting probability is in the range between 0 and 1 [19]. The formula used in the softmax operation can be found in equation (9).

$$p_j(x) = \frac{(e^{x_j})}{\sum_{i=1}^{k} e^{x_i}} \quad (9)$$

Hyperparameter tuning using bayesian optimization

Bayesian optimization is a method for optimizing hyperparameters that uses a stepwise approach [20]. This method is very effective in solving functions that require high computation. [21]. In addition, this method is suitable for functions that do not have a closed-form representation, do not provide a derived function, and can only be evaluated based on certain points [22]. The procedure to optimize the hyperparameters of Convolutional Neural Network (CNN) with Bayesian Optimization (BO) approach is shown in Algorithm 1.

Algorithm 1 Bayesian Optimization

Input: Objective function $f(x)$, Acquisition function $AF\ (x)$, initial set of observations $D = (x_1, y_q), \dots, (x_n, y_n)$
while: Stopping criterion not met **do**:
1: Fit GP to observations D
2: Find the next point for evaluation x_{next} by maximing the acquisition function:
3: $x_{next} = argmax\ AF(x \mid D, GP)$
4: Evaluate the objective function at x_{next}:
5: $y_{next} = f(x_{next})$
6: Update the set of observations:
7: $D = D\ (x_{next}, y_{next})$
8: If y_{next} is better than the current best solution y_{best} then:
9: $y_{best} = y_{next}$
10: $x^* = x_{next}$
end while:
Output: Return best solution x^*

Gaussian processing is a technique developed based on Gaussian stochastic processes and Bayesian learning theory. It is an extension of the Gaussian probability distribution. The model that occurs in this Gaussian process

is a finite subset of random variables that have a multivariate Gaussian distribution [21]. The formula used in this gaussian process can be seen in the equation (10)

$$f(x) \sim GP(m(x), k(x, x')) \quad (10)$$

Covariance function $k(x, x')$ also often referred to as the "kernel", describes the "smoothness" of the process. The assumption is that if two points x dan x' are adjacent, then the corresponding process results y dan y' will also approach each other. The two points are then highly dependent on each other, which is not the absolute location or direction of separation. Then, the covariance function used is generally a Squared Exponential (SE) function commonly known as the Radial Basis Function (RBF) [21].

$$k(x, x') = \exp\left(-\frac{1}{2\theta^2}\|x - x'\|^2\right) \quad (11)$$

Equation 11 shows that a decreasing relationship occurs in the square between the points. Then, the parameter θ is a scale that determines the decreasing length. Furthermore, kernel functions are often used in representing prior knowledge of the function [21]. In an experimental context, observations include normally distributed disturbances $\epsilon \sim N(0, \sigma 2)$. The structure of the observation model is as follows [21]:

$$y = f(x) + \epsilon \quad (12)$$

The regression contained in the Gaussian process is usually called kriging, which can estimate the value of the objective function $f(\cdot)$ concerning the time step $t + 1$ for each point x [21]. This estimation produces a normal distribution characterized by the mean $\mu_t\ (x)$ and the level of uncertainty $\sigma_t\ (x)$.

$$P(f_{t+1}|\mathcal{D}_{1:t}, x) = \mathcal{N}(\mu_t\ (x), \sigma_t^2(x)) \quad (13)$$

Hyperparameter tuning using genetic algorithm

Genetic algorithm is a method for hyperparameter search and optimization that uses the principle of natural selection. GA works on a population of individuals with chromosomes consisting of genes. Individuals are evaluated based on their fitness function. GA uses three main operators: selection to choose the best individual, crossover to combine two parents into offspring, and mutation to make small changes to genes. The main objective of GA is to find the optimal solution by evolving the chromosomes based on a predefined fitness function [23]. The procedure for optimizing the Convolutional Neural Network (CNN) hyperparameters with the Genetic Algorithm (GA) approach is shown in Algorithm 2.

Algorithm 2 Genetic Algorithm

Data: population of individuals;
Result: best individuals;
begin:
1: t = 0;
2: Create the starting population – Pop(0);
3: Perform individual evaluation – calculate the fitness function value for each individual in the population P_0;
4: **do**
5: select individuals to form a new population $Pop(t)$ – selection;
6: carry out crossover operation;
7: Implement mutations in individuals;
8: Evaluate individuals;
9: Replacing the old population with the new population;
10: $t = t + 1$;
11: **while** stop condition reached;
end:

Hyperparameter tuning using particle swarm optimization

Particle Swarm Optimization, commonly referred to as PSO, is a technique for stochastic hyperparameter optimization inspired by birds searching for food. In PSO, birds (or particles) do not know the exact location of the food, but through an iterative process, they approach the food source. Each bird follows the most successful strategy in the group and considers the best position it has ever achieved. PSO updates the position and velocity of each particle to find the optimal solution with equations (14) and (15) [24]. Algorithm 3 shows the procedure for optimizing CNN hyperparameters using the PSO approach.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Algorithm 3 Particle Swarm Optimization

Input: ***M*** → population size,
ub, lb → domain (upper bound and lower bound)
$\boldsymbol{D}$ → Dimension
c1, c2 → Personal and Global Learning Coefficient
MaxIt → number of maximum iterations
Output: The solution that has the best fitness value result
begin:
Initially initialize particle positions and velocities randomly (X)
Determine personal best (p_{best}) and global best (g_{best})
for $t = 1$ *to* ***MaxIt do***
  for $i = 1$ to **M do**
    set r_1, r_2, $w \in [0, 1]$
      for $j = 1$ to $\boldsymbol{D}$ do
        $V_{i,j}(t+1) = wV_{i,j}(t) + \boldsymbol{c1}r_1\left(p_{Best_{i,j}} - X_{i,j}(t)\right) + \boldsymbol{c2}r_2\left(g_{Best_j} - X_{i,j}(t)\right)$
        $X_{i,j}\ (t+1) = X_{i,j}\ (t) + V_{i,j}(t+1)$
        if $X_{i,j}(t+1) > \boldsymbol{ub\ then}$
          $X_{i,j}(t+1) = \boldsymbol{ub}$
        if $X_{i,j}(t+1) < \boldsymbol{lb\ then}$
          $X_{i,j}(t+1) = \boldsymbol{lb}$
      if $f(X_i)\ \leq f\left(p_{Best_i}\right)\ \boldsymbol{then}$
        $p_{Best_i} = X_i$
  argmin$(p_{best}) \rightarrow g_{best}$
end:

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * (pBest_{id} - x_{id}^{t}) + c_2 * r_2 * \left(gBest_{gd} - x_{id}^{t}\right) \quad (14)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (15)$$

At iteration t and dimension d in the optimization search space, $c1$ and $c2$ are acceleration constants that give weight to the stochastic acceleration component. This component drives each particle towards its individual best position $(pBestid)$ and global best position $(gBestgd)$. Value r_1 dan r_2 is a randomly distributed number in the range [0, 1].

$$W = w_{max} + (w_{min} + w_{max}) * \log_{10}(a + \frac{10\mathrm{t}}{\mathrm{T}_{max}}) \quad (16)$$

Where a is a constant for evolutionary speed adjustment, with a value of a = 1. w_{min} dan w_{max} are the minimum and maximum weights, while t is the iteration.

RESULT AND DISCUSSION

This chapter of results and discussion will highlight three main aspects: discussion on the results of preprocessing, hyperparameter experiments, and performance analysis. This research utilized RunPod's platform for a rental fee of $1.14 per hour to access higher-level resources for training and testing. Therefore, the resources used in training and testing must be adequate so that the performance and efficiency in testing the model can be optimized properly.

Result Preprocessing

The split data results have been presented in Table 1, it shows the distribution of lung nodule data used in this study, Then the data distribution is divided into three parts, namely training, validation, and testing. Data that is part of training, validation, and testing is classified into three categories, namely benign, malignant, and normal. The training subset was divided into 60% of the total images. Thus, there are 1,964 images categorized as benign, 3,149 images with malignant class, and 4,131 images labeled as normal class. Therefore, the total number of images that have been divided in the training subset is 9,244. Then, in the validation subset

with an image division of 20%. Thus, there are 655 images in the benign class, 1,050 images in the malignant class, and 1,377 images in the normal class. Therefore, the total number of images in the validation subset is 3,082 images. While the testing subset is divided by 20% of the total data. Thus, there are 655 images in the benign class, 1,050 images in the malignant class, and 1,377 images in the normal class. Thus, this test subset has a total of 3,082 images. The image distribution of each subset plays an important role in the development and evaluation of the lung nodule detection model.

Table 1. Distribution class train, validation, and test

Classes	Split	Image	Total
Benign	Train	1964	9244
Malignant	(60%)	3149	
Normal		4131	
Benign	Validation	655	3082
Malignant	(20%)	1050	
Normal		1377	
Benign	Test	655	3082
Malignant	(20%)	1050	
Normal		1377	

Based on the image enhancement shown in Figure 3, showing a comparison of CT scan images before and after image enhancement. Part (a) shows a benign lesion in the lung with clear boundaries and no signs of aggressive growth, as well as a reduction in the size of the lesion. Section (b) shows a malignant tumor in the lung with irregular borders and signs of infiltration into the surrounding tissue. Section (c) shows normal lung tissue with no visible lesions or abnormalities before observation and remains normal with no pathological changes after the observation period.

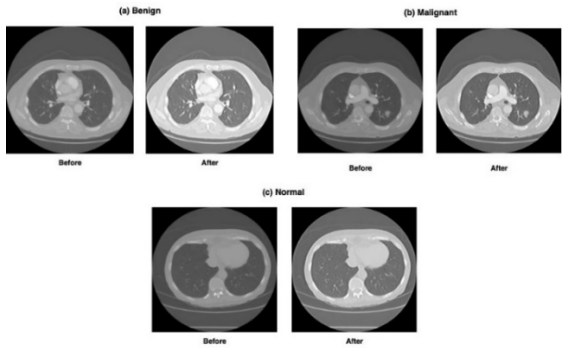

Figure 3. CT-Scan lung images before and after image enhancement

Experiment Hyperparameter

It is necessary to select the right hyperparameters in optimizing the performance of the machine learning model. Various methods have been developed to find the best hyperparameter combination, including Bayesian Optimization, Genetic Algorithms, and Particle Swarm Optimization. This section will describe each of these methods in detail, along with Table 2, which lists the hyperparameters to look for and their value ranges for CNN, and presents the computational time results, the best-selected hyperparameters, and visualization architecture for each method.

Table 2. Hyperparameters range of CNN

Hyperparameter	Value
Learning rate	[0.000001, 0.001]
Num of Layers	[2-5]
Num of Neurons	[32, 64, 128, 256]
Dropout Rate	[0.0-0.5]
Optimizer	(Adam, SGD)

Bayesian optimization

In this experiment, Bayesian Optimization with Expected Improvement (EI) acquisition function and ten iterations were performed in a total time of two hours and twenty-one minutes and thirty-seven seconds. This method found the optimal parameters with learning_rate 0.0000158, num_layers 2, num_neurons 256, dropout_rate 0.1164, and optimizer Adam. As can be seen in Figure 4, The result of the model architecture includes three, namely, three convolution layers, three max-pooling layers, a dropout layer, and two fully connected layers with a softmax activation function in classifying it into three classes: benign, malignant, and normal. The convolution and max pooling process starts from a 256 × 256 pixel grayscale image, generating features with increasingly smaller sizes until it reaches 60 × 60 pixels in the last convolution layer, followed by dropout and fully connected layers for final classification.

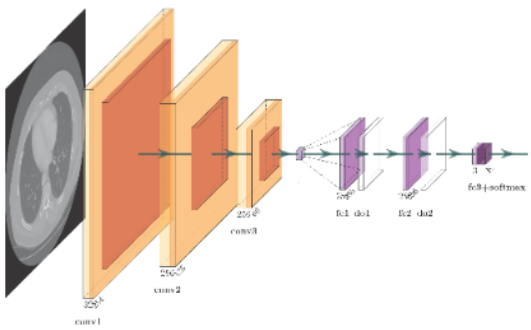

Figure 4. Visualization architecture BO-CNN

Genetic algorithm

In this experiment, the GA parameters were population_size 3, mutation_rate 0.1, and

five iterations were completed in three hours and fifty-two seconds. The method identified the optimal parameters as learning_rate 0.0002752, num_layers 5, num_neurons 128, dropout_rate 0.3614, and optimizer Adam. Figure 5 is the result of the model architecture consisting of six convolution layers, six max-pooling layers, five dropout layers, and five connected layers with softmax activation function to classify three classes: benign, malignant, and normal. The convolution and max pooling process reduces a 256 × 256 pixel grayscale image to 4 × 4 pixels, followed by dropout and fully connected layers for final classification.

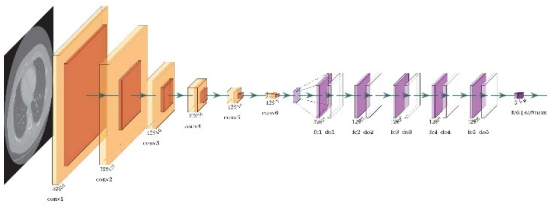

Figure 5. Visualization architecture GA-CNN

Particle swarm optimization

In this experiment, the PSO parameters were set to c1 0.5, c2 0.3, w 0.9, and five iterations were performed over two hours, fifteen minutes, and twenty-seven seconds. The method found the optimal parameters: learning_rate 0.0007628, num_layers 3, num_neurons 128, dropout_rate 0.3576, and optimizer Adam. Figure 6 shows that the model architecture includes four convolution layers, four max-pooling layers, three dropout layers, and fully connected three layers with a softmax activation function for classifying three classes: benign, malignant, and normal. The convolution and max pooling process reduces a 256 × 256 pixel grayscale image to 14 × 14 pixels, followed by dropout and fully connected layers for final classification.

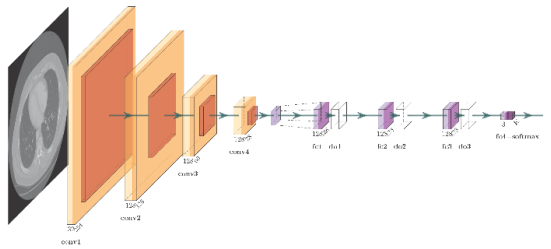

Figure 6. Visualization architecture PSO-CNN

Performance Analysis

The results in Table 3 show that our method outperforms other classification methods in terms of precision, recall, F1-Score, accuracy, and ROC. It achieved 97% precision, 97% recall, 97% F1-Score, 97.2% accuracy, and 99% ROC on the LIDC-IDRI dataset. Furthermore, the best method in the accuracy section is the SCCNN method with an accuracy of 95.45%, recall of 96.66%, and ROC of 97.26%. The SCCNN method only uses recall, ROC, and accuracy for evaluation.

However, based on the results presented, the proposed method outperforms SCCNN in terms of recall, and accuracy. Among other methods, the ResNet50-Ensemble Voting Method has a fairly high accuracy value but the accuracy value is still lower than our method. The Attention-based 3D-CNN method performs quite well, with an accuracy of 81.6%, but this is still lower than the method developed here. Overall, our method performs well for lung nodule classification compared to existing methods. The method proposed here can provide a comprehensive and effective solution for the task under consideration. In general, having high values in precision, recall, and ROC can be used as a tool for a reliable and trustworthy benchmark.

In addition, popular CNN architectures in comparison such as VGG16, AlexNet, and ResNet-152 have more parameter structures with trainable values of about sixty million parameters. Then, the model that we have optimized with Bayesian optimization has fewer parameter values when compared to previous architectures of 59,713,603. The fact that our method still works despite using fewer parameters demonstrates its superiority in reducing memory requirements during training and computational costs incurred to reduce the overall computation time. Having fewer parameters makes the model size more compact, which can be easily implemented on devices that have a limited amount of resources. In comparison, our method contains only 10 layers compared to the standard architecture size that includes 13 layers, yet still offers better computational efficiency.

In Figure 7, there is confusion matrix that illustrate the prediction results. The model successfully predicted 2,989 data. To be precise, 612 benign class (class 0) images were correctly predicted with 43 errors, 1002 malignant class (class 1) images were correctly predicted with 48 errors, and 1375 normal class (class 2) images were correctly predicted with only 2 errors.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 3, December 2024

Table 3. Comparison of classification results with previous state-of-the-art models

No	Author	Method	Object	Accuracy %	Precision %	Recall %	F1-Score %	ROC %
1	Shao et al. [25]	Deep Diffractive Neural Network (D2NN)	Lung Nodule	76.77	–	–	–	–
2	Liu et al. [26]	Attention-based 3D-CNN	Lung Nodule	81.6	–	–	–	–
3	Li et al. [27]	ResNet50-Ensemble Voting	Lung Nodule	94.3	–	–	–	–
4	Dong et al. [28]	Semantic Characteristics Convolutional Neural Network (SCCNN)	Lung Nodule	95.45	–	96.66	–	97.26
5	**Proposed Method**	**BO-CNN**	**Lung Nodule**	**97.2**	**97.00**	**97.00**	**97.00**	**99.67**

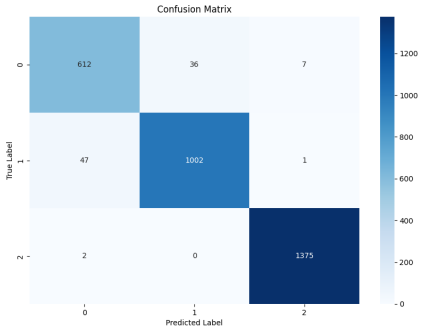

Figure 7. Confusion matrix BO-CNN

Then, Figure 8 shows that the model successfully predicted 2,972 data points. To be precise, 606 images of the benign class (class 0) were predicted correctly with 49 errors, 998 images of the malignant class (class 1) were predicted correctly with 52 errors, and 1,368 images of the normal class (class 2) were predicted correctly with 9 errors. The results of this model achieved the highest accuracy across all classes, demonstrating its effectiveness in distinguishing between benign, malignant, and normal lung nodules.

Furthermore, in Figure 9, the model successfully predicted 2,954 data. To be precise, 603 benign class (class 0) images are correctly predicted and there are 52 errors, 988 malignant class (class 1) images are correctly predicted and there are 62 errors, and 1363 normal class (class 2) images are correctly predicted and there are 14 errors.

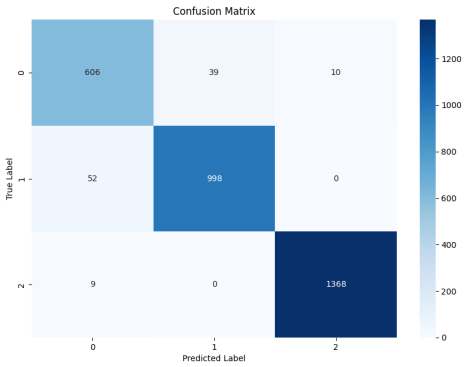

Figure 8. Confusion matrix GA-CNN

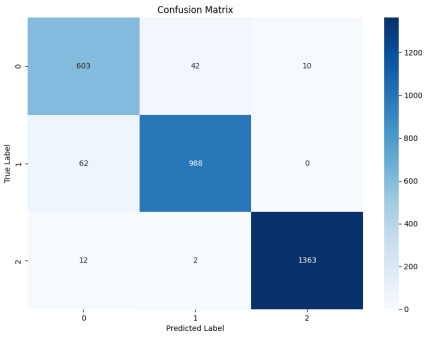

Figure 9. Confusion matrix PSO-CNN

In addition, we evaluated the performance of three different CNN models, namely: BO-CNN, GA-CNN, and PSO-CNN. The model used can classify lung nodules into three classes namely

benign, malignant and normal. The evaluation methods used include precision, recall, F1-Score, and number of samples (support) for each class. Based on the test results that have been carried out on the BO-CNN model performance test dataset contained in Table 4, the precision value reaches 92%, 96%, and 100% for benign, malignant, and normal classes. The recall values for the same classes were 93%, 96%, and 99%, resulting in F1-Score of 93%, 96%, and 99%.

The support for each class was 656, 1049, and 1377. Then, for the GA-CNN model in the benign nodule class category, the precision value obtained was 91%, the recall value was 93%, and the F1-Score value was 92%. Then, the malignant class has a precision of 96%, a recall value of 95%, and an F1-Score of 96%. Furthermore, the normal class achieved an almost perfect metric, with a precision value of 99%, a recall value of 99%, and an F1-Score value of 99%. The number of samples for benign, malignant, and normal classes had consistent values at 655 for the benign category, 1050 for the malignant category, and 1377 for the normal category. Furthermore, the PSO-CNN model has a precision value of 89%, a recall value of 92%, and an F1-Score value of 91% for the benign class. Furthermore, in the malignant class, the model recorded a precision value of 96%, a recall value of 94%, and an F1-Score value of 95%. Then, in the normal class, the precision and recall values obtained were 99%, and the resulting F1-Score was 99%. This model has a total number of samples in the benign class of 655, malignant class of 1050, and normal class of 1377.

The three models described in detail in the previous paragraphs were found to perform very well in classifying normal lung nodules. Then, the GA-CNN model had the highest accuracy value for malignant nodules, and the BO-CNN model showed a balanced performance across all classes. Furthermore, the PSO-CNN model in detecting benign nodules was slightly less precise but it still gave a fairly strong performance in terms of malignant and normal nodules. Therefore, these findings further emphasize the potential of CNN-based approaches in classifying lung nodules accurately. Thus, this finding is very important in early treatment planning to avoid lung nodules from getting bigger.

CONCLUSION

From this study, it can be concluded that the use of CNN models with the application of Bayesian Optimization as the best parameter selection is a very effective approach in classifying lung nodules with benign, malignant, or normal classes. Furthermore, the model evaluation results show an accuracy value of 97.2% with a total search time for the best parameters of 8497 seconds. When compared to CNN models that apply other hyperparameter methods such as Particle Swarm Optimization (PSO), it produces an accuracy value of 96.4% with a total time of 8127 seconds, and the Genetic Algorithm (GA) gets an accuracy value of 96.4% with a total time of 10852 seconds. Thus, the Bayesian Optimization (BO) approach shows better performance in terms of accuracy when compared to the PSO and GA hyperparameters, which can help radiologists predict and make decisions to diagnose lung cancer.

Table 4. Model performance metrics evaluation results

Method	Classes	Precision	Recall	F1-Score	Support
BO-CNN	Benign	92%	93%	93%	656
	Malignant	96%	96%	96%	1049
	Normal	100%	99%	99%	1377
	Average	**97%**	**97%**	**97%**	**–**
GA-CNN	Benign	91%	93%	92%	655
	Malignant	96%	95%	96%	1050
	Normal	99%	99%	99%	1377
	Average	**96%**	**96%**	**96%**	**–**
PSO-CNN	Benign	89%	92%	91%	655
	Malignant	96%	94%	95%	1050
	Normal	99%	99%	99%	1377
	Average	**96%**	**96%**	**96%**	**–**

REFERENCES

[1] S. Tomassini, N. Falcionelli, P. Sernani, L. Burattini, and A. F. Dragoni, "Lung nodule diagnosis and cancer histology classification from computed tomography data by convolutional neural networks: A survey," Jul. 01, 2022, *Elsevier Ltd*. doi: 10.1016/j.compbiomed.2022.105691.

[2] R. Wu, C. Liang, Y. Li, X. Shi, J. Zhang, and H. Huang, "Self-supervised transfer learning framework driven by visual attention for benign–malignant lung nodule classification on chest CT," *Expert Syst Appl*, vol. 215, Apr. 2023, doi: 10.1016/j.eswa.2022.119339.

[3] I. Ali, M. Muzammil, I. U. Haq, A. A. Khaliq, and S. Abdullah, "Efficient lung nodule classification using transferable texture convolutional neural network," *IEEE Access*, vol. 8, pp. 175859–175870, 2020, doi: 10.1109/ACCESS.2020.3026080.

[4] P. Zhai, Y. Tao, H. Chen, T. Cai, and J. Li, "Multi-Task Learning for Lung Nodule Classification on Chest CT," *IEEE Access*, vol. 8, pp. 180317–180327, 2020, doi: 10.1109/ACCESS.2020.3027812.

[5] A. E. Sebastian and D. Dua, "Lung Nodule Detection via Optimized Convolutional Neural Network: Impact of Improved Moth Flame Algorithm," *Sens Imaging*, vol. 24, no. 1, Dec. 2023, doi: 10.1007/s11220-022-00406-1.

[6] R. Raza *et al.*, "Lung-EffNet: Lung cancer classification using EfficientNet from CT-scan images," *Eng Appl Artif Intell*, vol. 126, Nov. 2023, doi: 10.1016/j.engappai.2023.106902.

[7] N. M. Aszemi and P. D. D. Dominic, "Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms," 2019. [Online]. Available: www.ijacsa.thesai.org

[8] A. F. Hamdani, D. Swanjaya, and R. Helilintar, "Facebook Prophet Model with Bayesian Optimization for USD Index Prediction," *JUITA: Jurnal Informatika*, vol. 11, no. 2, pp. 293–300, 2023.

[9] M. Loey, S. El-Sappagh, and S. Mirjalili, "Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data," *Comput Biol Med*, vol. 142, Mar. 2022, doi: 10.1016/j.compbiomed.2022.105213.

[10] O. S. Azeez, H. Z. M. Shafri, A. H. Alias, and N. A. Haron, "A Joint Bayesian Optimization for the Classification of Fine Spatial Resolution Remotely Sensed Imagery Using Object-Based Convolutional Neural Networks," *Land (Basel)*, vol. 11, no. 11, Nov. 2022, doi: 10.3390/land11111905.

[11] J. Zheng *et al.*, "Metabolic syndrome prediction model using Bayesian optimization and XGBoost based on traditional Chinese medicine features," *Heliyon*, vol. 9, no. 12, Dec. 2023, doi: 10.1016/j.heliyon.2023.e22727.

[12] M. F. ASLAN, "Comparative Analysis of CNN Models and Bayesian Optimization-Based Machine Learning Algorithms in Leaf Type Classification," *Balkan Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 13–24, Jan. 2023, doi: 10.17694/bajece.1174242.

[13] X. Song *et al.*, "Bayesian-Optimized Hybrid Kernel SVM for Rolling Bearing Fault Diagnosis," *Sensors*, vol. 23, no. 11, Jun. 2023, doi: 10.3390/s23115137.

[14] A. Riboni, N. Ghioldi, A. Candelieri, and M. Borrotti, "Bayesian optimization and deep learning for steering wheel angle prediction," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi: 10.1038/s41598-022-12509-6.

[15] L. Zhang *et al.*, "CNN-LSTM Model Optimized by Bayesian Optimization for Predicting Single-Well Production in Water Flooding Reservoir," *Geofluids*, vol. 2023, 2023, doi: 10.1155/2023/5467956.

[16] K. V. Ranjitha and T. P. Pushphavathi, "Analysis on Improved Gaussian-Wiener filtering technique and GLCM based Feature Extraction for Breast Cancer Diagnosis," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2857–2866. doi: 10.1016/j.procs.2024.04.270.

[17] Y. Heningtyas, A. Syarif, and A. Gieniung Pratidina, "Implementation of Gabor Filter for Carassius Auratus's Identification," vol. 11, no. 2, 2021.

[18] A. A. Abd Al-Ameer, G. Abdulameer Hussien, and Hajer. A. Al Ameri, "Lung cancer detection using image processing and deep learning," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, p. 987, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp987-993.

[19] S. N. Fadilah, D. C. R. Novitasari, and L. Hakim, "Pengaruh Reduksi Fitur Pada Klasifikasi Kanker Paru Menggunakan CNN Dengan Arsitektur GoogLeNet," *Jurnal Fourier*, vol. 12, no. 1, pp. 20–32, Apr. 2023, doi: 10.14421/fourier.2023.121.20-32.

[20] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, "Basic Enhancement

Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks," *IEEE Access*, vol. 8, pp. 52588–52608, 2020, doi: 10.1109/ACCESS.2020.2981072.

[21] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.

[22] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian Optimization for Adaptive Experimental Design: A Review," *IEEE Access*, vol. 8, pp. 13937–13948, 2020, doi: 10.1109/ACCESS.2020.2966228.

[23] P. R. Togatorop, M. Sianturi, D. Simamora, and D. Silaen, "Optimizing Random Forest using Genetic Algorithm for Heart Disease Classification," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 13, no. 1, p. 60, Aug. 2022, doi: 10.24843/lkjiti.2022.v13.i01.p06.

[24] M. Rosyda, "Logarithm Decreasing Inertia Weight Particle Swarm Optimization Algorithms for Convolutional Neural Network," *JUITA: Jurnal Informatika*, vol. 10, no. 1, pp. 99–105, 2022.

[25] J. Shao, L. Zhou, S. Y. F. Yeung, T. Lei, W. Zhang, and X. Yuan, "Pulmonary Nodule Detection and Classification Using All-Optical Deep Diffractive Neural Network," *Life*, vol. 13, no. 5, May 2023, doi: 10.3390/life13051148.

[26] G. Liu, F. Liu, J. Gu, X. Mao, X. T. Xie, and J. Sang, "An attention-based deep learning network for lung nodule malignancy discrimination," *Front Neurosci*, vol. 16, Jan. 2023, doi: 10.3389/fnins.2022.1106937.

[27] W. Li *et al.*, "Machine Learning Model of ResNet50-Ensemble Voting for Malignant–Benign Small Pulmonary Nodule Classification on Computed Tomography Images," *Cancers (Basel)*, vol. 15, no. 22, Nov. 2023, doi: 10.3390/cancers15225417.

[28] Y. Dong, X. Li, Y. Yang, M. Wang, and B. Gao, "A Synthesizing Semantic Characteristics Lung Nodules Classification Method Based on 3D Convolutional Neural Network," *Bioengineering*, vol. 10, no. 11, Nov. 2023, doi: 10.3390/bioengineering10111245.