



Lung Nodule Diagnosis via Deep Learning and Swarm Intelligence

Cesar Affonso de Pinho Pinheiro¹, Nadia Nedjah¹(✉),
and Luiza de Macedo Mourelle²

¹ Department of Electronics Engineering and Telecommunications,
Faculty of Engineering, State University of Rio de Janeiro, Rio de Janeiro, Brazil
cesaraffonso05@hotmail.com, nadia@eng.uerj.br

² Department of Systems Engineering and Computation, Faculty of Engineering,
State University of Rio de Janeiro, Rio de Janeiro, Brazil
ldmm@eng.uerj.br

Abstract. In general, it is difficult to perform cancer diagnosis. In particular, pulmonary cancer is one of the most aggressive type of cancer and hard to be detected. When properly identified in its early stages, the chances of survival of the patient increase significantly. However, this detection is a hard problem, since it depends only on visual inspection of tomography images. Computer aided diagnosis methods can improve a great deal the detection and, thus, increasing the surviving rates. In this paper, we exploit computational intelligence techniques, such as deep learning, convolutional neural networks and swarm intelligence, in order to propose an efficient approach that allows identifying carcinogenic nodules in digital tomography scans. We use 7 different swarm intelligence techniques to approach the learning stage of a convolutional deep learning network. We are able to identify and classify cancerous pulmonary nodules successfully in the tomography scans of the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI). The proposed approach, which consists of training Convolutional Neural Networks using swarm intelligence techniques, proved to be more efficient than the classic training with Back-propagation and Gradient Descent. It improves the average accuracy from 93% to 94%, precision from 92% to 94%, sensitivity from 91% to 93% and specificity from 97% to 98%.

Keywords: Deep learning · Swarm intelligence ·
Lung cancer detection · Convolutional neural networks

1 Introduction

This work is proposed to contribute to the community of researchers who attempt to improve the identification techniques of this evil that carries such expressive numbers and ends a lot of victim's life every day. Methods like the one designed in this work are very useful to medical doctors, because they are able, in a short

period of time, to save human lives with greater efficiency. Therefore, computer aided diagnosis tools have turned out to be increasingly essential for medical's routines.

This work aims to explain its contribution by first presenting in Sect. 2 all of the main works that were used as knowledge basis and inspiration. Then, in Sects. 3 and 4 it gives through the conceptual background required for its development. It presents an introduction to the main concepts, such as Deep Learning and Swarm Intelligence. Later, in Sect. 5 we describe the proposed model of the nodules detection problem. Subsequently, in Sect. 6 we explain all the methodologies we used in this work. This includes the image processing, image classification and network training. Finally, Sect. 7 presents the results obtained in this work and discusses the relations between different swarm-training algorithms so that the final conclusion can be presented in Sect. 8.

2 Related Work

In [1], a Computer-Aided Diagnosis tool (CAD), which is based on convolutional neural network (CNN) and deep learning, is described to identify and classify lung nodules. Transfer Learning is used in this work. It consists of using a previously trained CNN model and it only adjusts some of the network last layer's parameters in order to make them fit the scope of the application. The authors used the ResNet network, designed by Microsoft. Among 1971 competitors, this model was ranked 41st in the competition organized by Microsoft. It thus confirmed that the method is really effective for this kind of application.

In [2] the authors used the same method as [1]. The difference between those works is the chosen networks for the use of Transfer Learning. In this work, the U-Net convolutional network was used, which significantly increased the performance of the nodule detection problem. The U-Net CNN specializes in pattern recognition tomography scans. The results reported in this work show a better performance than those shown in [1], allowing a higher accuracy, sensitivity and specificity.

In [3], the Particle Swarm Optimization algorithm was used by the authors to develop and train a CAD system based on an artificial neural network, which is able to identify and classify carcinogenic nodules in mammography digital images. The authors had to design a model dedicated to the extraction of features using image processing techniques, because of the lack of data. Once the attributes were extracted, the classification was performed according to two different approaches: the first one uses a neural network trained using back-propagation and the second one exploits a neural network trained using Particle Swarm Optimization (PSO). The former approach reached 88.6% of accuracy, 72.7% of sensitivity and 93.6% of specificity. The latter achieved 95.6% of accuracy, 87.2% of sensitivity and 97.3% of specificity. Thus, the results reported in this work justify the exploitation of swarm intelligence based techniques to discover the weights of the neural network, are able to enhance the classification performance in diagnosis tool.

3 Deep Learning with Convolutional Neural Networks

Convolutional neural networks are deep learning models that are structured as input layer, which receives the images; several layers, that execute image processing operation, aiming at feature mapping; and a classification neural network, which uses the obtained feature mapping and returns the classification result. The mapping layers implement three main operations, which are convolution, that is basically a matrix scan to detect feature similarities; pooling, that is an operation applied to extract the most relevant information of each feature map; and the activation layers, that are mostly used to reduce the linearity of these networks and increase its performance.

Besides the feature layers, CNNs include a fully connected layer. Based on the received feature maps, this classical multilayer perceptron analyzes the input and outputs the final classification result.

When Transfer Learning is used, a pre-trained CNN is selected and used to identify the application patterns after a new training. At this stage, the multi-layer perceptron network must be retrained to allow the correct classification of the application data. The chosen network for this work was the U-Net [4], as it is specialized in medical imaging pattern recognition.

4 Swarm Intelligence Algorithms

Swarm intelligence denotes a set of computational intelligence algorithms that simulate the behavior of colonies of animals to solve optimization problems. In order to perform neural networks training via swarm intelligence based algorithms, the following steps are implemented: (i) define the number of coefficients and that of the network layers; (ii) characterize the search space of the weights values; (iii) define a cost function to be used to qualify the solutions; (iv) select a swarm intelligence algorithm to be applied. Subsequently, a swarm of particles is initialized, wherein each particle is represented by the set of the weights required by the classification network. Afterwards, the quality of each particle of the swarm is computed and moved in the search space, depending on the rules of the selected optimization algorithm. This process is iterated until a good enough solution for the application is found. Algorithm 1 describes the thus explained steps.

In this application, 7 swarm intelligence algorithms were used. They are the Particle Swarm Optimization (PSO) [5], Artificial Bee Algorithm (ABC) [6], Bacterial Foraging Optimization (BFO) [7], Firefly Algorithm (FA) [8], Firework Optimization Algorithm (FOA) [9], Harmony Search Optimization (HSO) [10] and Gravitational Search Algorithm (GSO) [11].

5 Proposed Methodology

Given the swarm algorithms, the convolutional neural network is trained using each of the 7 methods and also using the traditional method with Back-Propagation and the Gradient Descent. Once the experiments have been executed, the performance of every selected algorithm is investigated regarding

Algorithm 1. Neural network training algorithm based on swarm intelligence approach

```

e = #epochs;
n = #particles;
Initialize the particle positions of the first swarm;
for i := 1 to e do
  for j = 1 to n do
    Select the j-th particle to be used as the set of weights for the classification
    network;
    Evaluate the idealized cost function of that particle;
  end for
  Apply the swarm rules to obtain a new swarm;
end for
Select the best particle as the final set of weights for the network;

```

accuracy, sensitivity, precision and specificity. During the experimentation phase, some of the network parameters were also tested to evaluate the impact and possibly result in performance improvements. Among the parameters chosen for these tests are the size of the Pooling matrix and the network’s activation function. After obtaining the results, a ordered Table was generated to choose the best algorithms for this application, according to each performance indicator.

6 Implementation Issues

In this work, the medical-image database LIDC-IDRI [12] was used to train the nodule classification model. It contains over 250,000 images of CT scans of lungs. All the images in this database were analyzed by up to four medics, which gave their diagnosis on each case.

The LUNA16 database [13], is a lung CT-image database which was originated from the LIDC-IRDI. This set was created by selecting only the images of the LIDC-IRDI that contains complete information. This database is also used in this work to provide a richer information.

6.1 Data Preprocessing

Training the network with all points in each image would highly increase the computational effort. To cope with this problem, square cuts of 50×50 pixels were selected around each nodule pointed by the specialists. Figure 1 shows an example of a square cut.

With this configuration, these images are ready to feed the model. However, the distribution of classes is still heavily unbalanced for the training. The current dataset has about 550,000 annotations, and about 1300 were classified as nodules. To prevent training problems, negative classifications are randomly reduced and data augmentation is applied on positive classifications. Data augmentation routines include rotating the images in 90° , horizontal and vertical inverting.

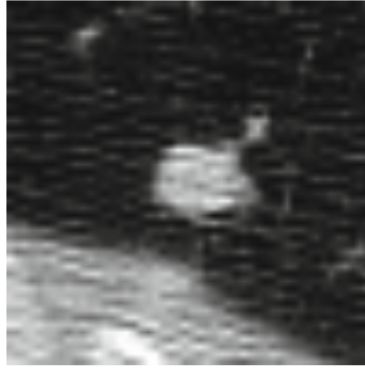


Fig. 1. Nodule cropped image

Once these routines are properly applied, the dataset is left as 80% negative and 20% positive regarding image labels, which still may not be the perfect balance, but over-increasing the positive classifications could lead into variance problems. After this class balancing, data preprocessing is thus completed.

6.2 The Model

The U-Net [4] convolutional neural network is the base of the model developed in this work. The architecture of the network is shown in Fig. 2.

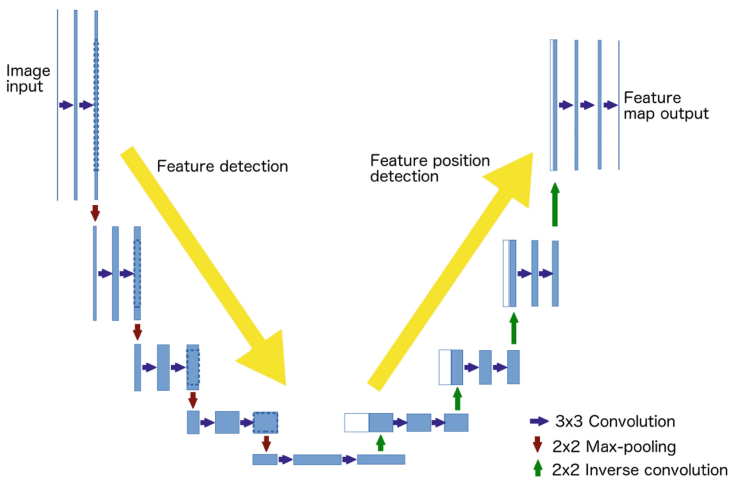


Fig. 2. U-Net architecture

The U-Net is composed of two stages. The first stage is focused on finding-out what features are present in the image. The second stage works on stating

where these features are located on the image. This network uses 3×3 filters to apply convolution operations. It also uses ReLU activation functions, 2×2 Max-pooling matrices, and 2×2 inverse convolution filters.

To fit the U-Net [4] network to the application images, its layers were remodeled to work with the $50 \times 50 \times 1$ size images generated during preprocessing. After the mapping operations, the network outputs a feature map, which is ready to be analyzed by classification neural network. The maps are used as inputs for a fully connected neural network built with 100 units. This classification network receives the feature map and outputs a two-dimensional Softmax vector, which is a vector with probabilistic values, ranging from 0 to 1 and that, all together, sum up to 1. The value of each position in this vector represents the chance of the following input belonging to a certain class. In this approaching, the highest value will be chosen as the classification result.

To this network a Dropout layer was also added, which adds a technique used to control the over-fitting problem, an increasing error due to over-training. Dropout is used to turn some activations layers to zero regarding a probabilistic decision. This operation is used make the network redundant, as it must classify correctly even without some of its activation layers. This technique is an ally against over-fitting.

6.3 Training and Testing

With the model and the dataset ready to be used, training could be started. The training dataset had about 6900 classified images. The validation set was then selected with about 1620 images. After separating these data sets, the first training was applied using just back-propagation and gradient descent techniques. To choose the best set of learning rate and training epochs, some possible values were tested in 100 experiments for each configuration.

In every experiment, 4 metrics were analyzed. Equations 1, 2, 3 and 4 explain these metrics, where TP is the number of true positives, TN true negatives, FP false positives FN false negatives:

- Accuracy: Rate of hits among all classifications (Eq. 1)

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision: Rate of positive hits among the classified as positive (Eq. 2)

$$\frac{TP}{TP + FP} \quad (2)$$

- Specificity: Rate of negative hits among the real negatives (Eq. 3)

$$\frac{TN}{TN + FP} \quad (3)$$

- Sensitivity: Rate of positive hits among the real positives (Eq. 4)

$$\frac{TP}{TP + FN} \quad (4)$$

Figure 3 shows the mean results obtained on each tested configuration of hyper-parameters. The number of epochs and the learning rate were tested in pairs, ranging from 4 different values for each hyper-parameters. Table 1 shows the results obtained on these experiments.

Table 1. Pairs of hyper-parameters experimented

Case	Epochs	Learning rate	Case	Epochs	Learning rate
S_1	30	0.0001	S_9	300	0.0001
S_2	30	0.001	S_{10}	300	0.001
S_3	30	0.01	S_{11}	300	0.01
S_4	30	0.1	S_{12}	300	0.1
S_5	100	0.0001	S_{13}	1000	0.0001
S_6	100	0.001	S_{14}	1000	0.001
S_7	100	0.01	S_{15}	1000	0.01
S_8	100	0.1	S_{16}	1000	0.1

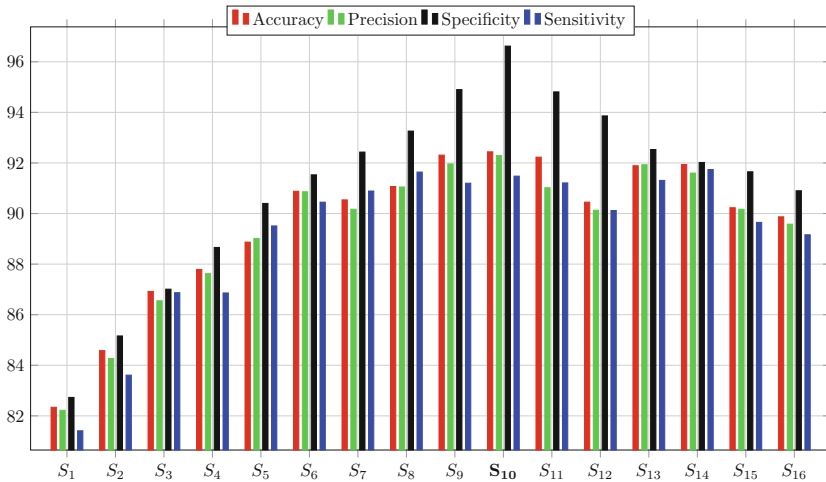


Fig. 3. Impact of the number of epochs and the learning rate

The model that used 300 epochs and a 0.001 learning rate presented a slightly better performance regarding accuracy and precision. This configuration also performed better in specificity. With these results, this configuration was chosen for this application. After testing the model with Back-propagation and gradient descent, it was ready to be tested with swarm intelligence.

6.4 Training with Swarming

The swarm algorithms used in this work are composed of 100-coordinate vector particles. Each swarm had 300 particles and was trained over 300 iterations per experiment. Figure 4 explains the training of a neural network via swarm intelligence, where each particle is a full configuration of weights for the classifier network.

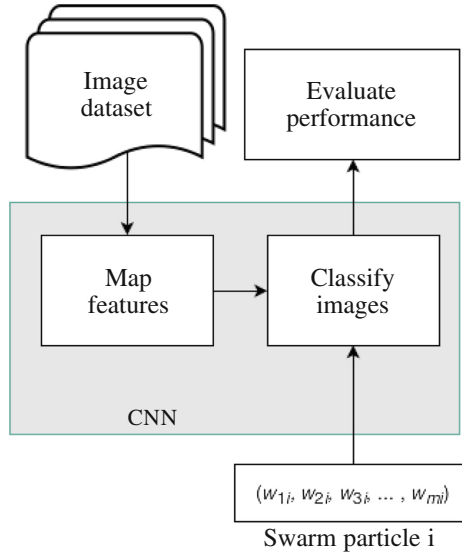


Fig. 4. Network training using the swarm of particles

Besides the number of particles and the number of iterations, the swarm algorithms have their own hyper-parameters that were set after some simulations conducted for every algorithm. The Bacterial Foraging Optimization sets the chemotaxis steps, which mimics the movement of bacteria, to 5 steps, the max distance of navigation to 2 units, the size of each step to 1 unit, and the probability of elimination to 5%. The Firework Optimization sets both the number of normal and Gaussian fireworks to 50. The Firefly Algorithm sets the mutual attraction index to 1, light absorption index to 1 while the two randomization parameters denominated α_1 and α_2 are set to 1 and 0.1, respectively and two parameters to adjust the Gaussian curve to 0 and 0.1. The Harmony Search Algorithm uses the pitch adjustment rate of 0.5, the harmony consideration rate to 0.5, and the bandwidth to 0.5. The Particle Swarm Optimization sets the inertia coefficient to 0.49, the cognitive coefficient to 0.72 and the social coefficient to 0.72. The Gravitational Search Algorithm uses an initial gravitational constant of 30 and an initial acceleration α of 10. The Artificial Bee Algorithm uses only two parameters: the number of particles and iterations parameters.

The swarm optimization algorithms are used to train and test the classification model over 100 experiments. In these experiments, accuracy, precision, sensitivity and specificity are used to evaluate the classification performance. To look for possible improvements, some of the U-Net parameters were also modified. These parameters are the size of the max-pooling matrices (3×3 , 4×4 and 5×5) and the activation functions. Regarding the activation functions, the hyperbolic tangent was also tested to substitute the ReLU function.

7 Performance Evaluation

In order to find out the best swarm strategy for training the CNNs for applications such as the one under consideration here, the performance evaluation of each algorithm was computed. The results obtained proved that swarm algorithms provide high efficiency.

Another included test is the use of the hyperbolic tangent instead of the ReLU as activation function. This testing is conducted to compare the efficiency of these functions in this application. Although the ReLU function is the most used function in convolutional neural networks, the hyperbolic tangent function was also tested with a swarm-trained model.

Figures 6, 7, 8 and 9 present the performance averages (in percentage) of each algorithm in the four metrics at the end of 100 experiments. Each figure presents all the results obtained in one metric with models tested at the following conditions: ReLU with pooling matrix sizes of 2×2 (C_1), 3×3 (C_2), 4×4 (C_3) and 5×5 (C_4), and Hyperbolic Tangent functions with pooling matrix sizes of 2×2 (C_5), 3×3 (C_6), 4×4 (C_7) and 5×5 (C_8).

After these experiments, a decrease in the performance metrics was observed as the max-pooling matrices size increased. Thus, it is possible to conclude that, for this application, the bigger the max-pooling matrix, the worse the model performs. This behavior may be caused by the loss of information in the feature map as these matrices grow bigger.

Based on these results, we can say that the TanH function performed a little worse than the ReLU. It is known that the latter decreases the linearity of data processing in CNNs, providing a better performance. Considering different configurations of the activation function and max-pooling matrix size, Fig. 5 shows the average results for accuracy, precision, sensitivity and specificity. Besides showing the better performance of the ReLU models, we can see that the smaller size of the max-pooling matrix also contributes to the enhancement of the performance.

Table 2 shows the best results obtained in each performance metric, stating which algorithm achieved that result and comparing it to the Back-propagation model. Taking these results into account, one can state the real effectiveness of using swarm intelligence algorithms to train convolutional neural networks for detecting pulmonary nodules. The experiments showed that at least 5 out of 7 swarm-trained models were superior when compared to the back-propagation

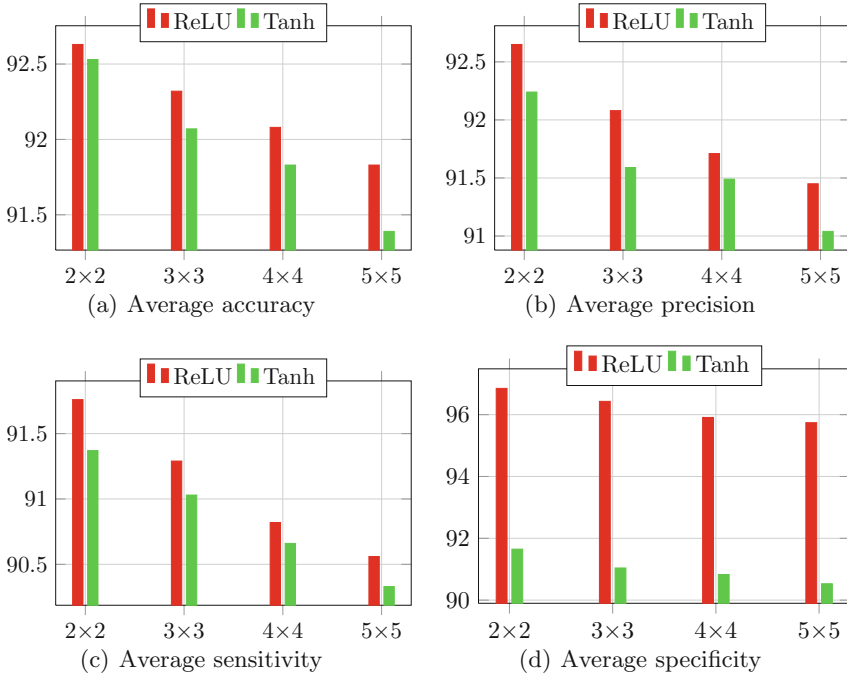


Fig. 5. Average performance comparison when using ReLU *vs.* TanH regarding all considering techniques

Table 2. Performance comparison regarding swarm *vs.* back-propagation experiments

Metric	Back-propagation	Swarm-based training	Algorithm
Accuracy	92.80	93.71	PSO
Precision	92.29	93.53	GSA
Sensitivity	91.48	92.96	PSO
Specificity	96.62	98.52	HA

models. The PSO algorithm reached the best performance in accuracy and sensitivity, the Harmony Search in specificity and the Gravitational Search in precision. When comparing to [3], the same behavior was observed, where the swarm-trained methods were vastly superior against back-propagation ones. From these results, problem’s nature must be analyzed to choose the best algorithm.

Figures 6, 7, 8 and 9 demonstrate the behaviors of the top three algorithms used in this work under all tested conditions regarding accuracy, precision, sensitivity and specificity. From these behaviors, its possible to observe the tendency of decreasing performance as the pooling matrix size increases as well as a slight superiority between ReLU models over Hyperbolic Tangent cases.

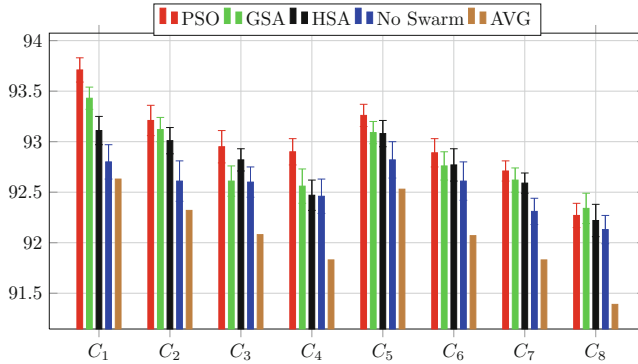


Fig. 6. Accuracy performance of top 3 algorithms, no swarm model and average of the results obtained by all algorithms

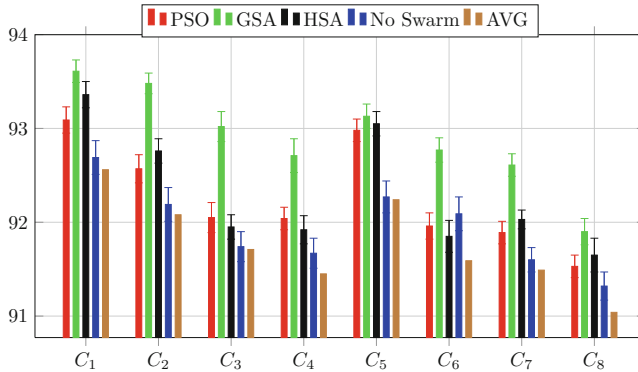


Fig. 7. Precision performance of the top 3 algorithms, no swarm model and average of the results obtained by all algorithms

Figures 6, 7, 8 and 9 show that the best performance for specificity is obtained by Harmony Search algorithm. Therefore, we can conclude that this algorithm is well suited for assuring that lung nodules classified as non-cancerous are in fact non-cancerous. However, when precision is concerned, the best algorithm is Gravitational Search, which means that it is suited for assuring that lung nodules classified as cancerous are in fact cancerous. Moreover, concerning accuracy and sensitivity, the best algorithm is PSO, which means that it is well suited for assuring that cancerous lung nodules are positively classified.

Classifying a cancerous patient as healthy is worse than classifying a non-cancerous patient as healthy. So, the false negative rate is the most important factor. Accuracy is the second most important, which outputs the overall model performance. Based on this observation, PSO can be considered the best algorithm for training the lung nodule classifier.

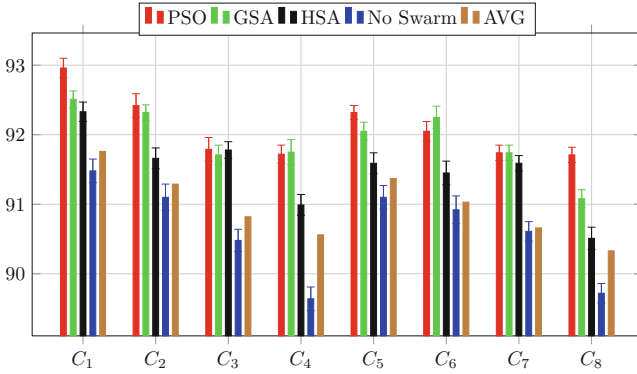


Fig. 8. Sensitivity performance of the top 3 algorithms, no swarm model and average of the results obtained by all algorithms

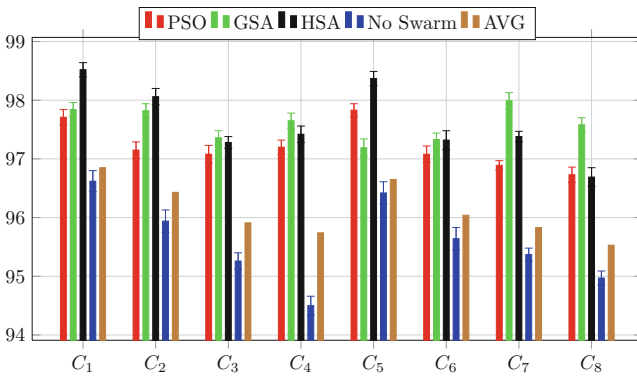


Fig. 9. Specificity performance of the top 3 algorithms, no swarm model and average of the results obtained by all algorithms

8 Conclusion

With the results obtained, it is possible to confirm the efficiency of adopted training strategy, based on the usage of swarm intelligence techniques. It achieved an improvement of the average accuracy from 92.80% to 93.71%, precision from 92.29% to 93.53%, sensitivity from 91.48% to 92.96% and specificity from 96.62% to 98.52%. During the performed simulation, we investigated the impact of the activation function. We verified the performance of the Rectified Linear Unit (ReLU) function as well as the Hyperbolic Tangent function. We also investigated the impact of different max-pooling functions on the performance of the network. We concluded that the ReLU models achieved a better performance than hyperbolic tangent based models. With respect to the max-pooling matrix size, we proved that the larger the matrix is, the worse the performance obtained.

Regarding the performance of the investigated swarm intelligence algorithms, three out of the seven exploited methods provided the best performances. PSO, HSA and GSA allowed the achievement of the best performances, regarding the four considered metrics. It is noteworthy to point out that labeling a non-cancerous nodule as cancerous is a bad decision. However, identifying a cancerous nodule as non-cancerous is worse as it affects the following treatment of the corresponding patient. Based this observation, one would elect the PSO technique as the best one simply because it achieved higher accuracy yet it provided the lowest rate of false negatives.

As a future work, we intend to apply our approach to images of other kind of tumors so as to generalize the obtained results.

References

1. Kuan, K., et al.: Deep learning for lung cancer detection: tackling the kaggle data science bowl 2017 challenge. arXiv preprint [arXiv:1705.09435](https://arxiv.org/abs/1705.09435) (2017)
2. Chon, A., Balachandra, N., Lu, P.: Deep convolutional neural networks for lung cancer detection. Stanford University (2017)
3. Jayanthi, S., Preetha, K.: Breast cancer detection and classification using artificial neural network with particle swarm optimization. *Int. J. Adv. Res. Basic Eng. Sci. Technol.* **2** (2016)
4. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
5. Kennedy, J., Eberhart, R.: Particle swarm optimization (1995)
6. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Global Optim.* **39**(3), 459–471 (2007)
7. Passino, K.M.: Bacterial foraging optimization. *Int. J. Swarm. Intell. Res.* **1**(1), 1–16 (2010)
8. Ritthipakdee, T., Premasathian, J.: Firefly mating algorithm for continuous optimization problems, July 2017
9. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13495-1_44
10. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *SIMULATION* **76**(2), 60–68 (2001)
11. Khajezadeh, M., Eslami, M.: Gravitational search algorithm for optimization of retaining structures. *Indian J. Sci. Technol.* **5**(1) (2012)
12. Cancer Imaging Archive: The lung image database consortium image collection (LIDC-IDRI). <https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI>. Accessed 2 Feb 2018
13. Jacobs, C., Setio, A.A.A., Traverso, A., van Ginneken, B.: Lung nodule analysis 2016 (LUNA16). <https://luna16.grand-challenge.org/>. Accessed 2 Feb 2018