Efficient Tuning of Hyper-parameters in Convolutional Neural Network for Classification of Tuberculosis Images

Ruchika Arora*, Indu Saini, Neetu Sood

Department of Electronics and Communication Engineering, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, 144011, India

*Corresponding author
doi: https://doi.org/10.21467/proceedings.114.1

Abstract

Deep Learning (DL) algorithms, especially Convolutional Neural Network (CNN) have outperformed in medical image classification tasks and have achieved human-competitive performance. This has become possible because CNN learns image features through backpropagation. However, the strategy for designing a CNN model with the highest accuracy for a specific application is often unclear. Because finding an appropriate network structure with the best combination of hyperparameters for different datasets is always a challenging task. To address this, we propose an optimized CNN framework that automatically and efficiently tune its hyper-parameters using a hyperband search optimization approach. In this paper, an efficient CNN with optimized hyperparameters for the classification of tuberculosis disease in Chest X-Ray (CXR) images is trained and tested on a publicly available NLM-China dataset. The experimental results illustrate that the hyperparameters optimize the CNN framework and achieve 91.42% accuracy for the classification of tuberculosis disease in CXR images.

Keywords: Chest X-ray images; Deep learning; Image classification; CNN

1 Introduction

In the early 1980s, a supervised machine learning model was designed which is popularly known as CNN. The CNN directly operate on raw data and extract features from it. A CNN framework has an input layer, a hidden layer, and an output layer. In CNN features of input data are extracted and classified by hidden layers through a convolution operation. For this operation, a convolution filter with weight is applied to the input data. Then a pooling layer abstracts the input data. After that, a fully connected layer classifies the extracted features with a multilayer neural network architecture. During the training of the CNN framework, error backpropagation and gradient descent algorithms adjust the weights of the convolution kernel and the neurons in the fully connected layer [1]. However, the performance of a CNN framework depends on hyperparameters such as the size of the convolution kernel, number of channels, padding, and stride because they determine the structure of layers and the size of the feature maps. Structures and hyperparameters of the CNN framework are determined purely on a trial-and-error basis and mostly by the experience of the designer [2]. For example, the LeNet-5 network uses convolution kernels size of 5 for all layers. In the Alexnet and ZFNet architectures convolution kernels sizes are set to 11, 5, 3 and 7, 5, 3 respectively [3]. While the selection of all possible sets of hyperparameters for a good model is computationally expensive [4]. Therefore, finding mechanisms for automatically searching and optimizing hyperparameters for any network is crucially needed. To address this challenge, we design an efficient CNN model with a hyper band search approach, that automatically tuned and optimized hyperparameters for the classification of CXR images with tuberculosis.

The two main contributions in this paper are:

1. A small-size CNN model is proposed for the automatic classification of tuberculosis images.
2. We incorporate an efficient hyperparameters optimization approach for the same CNN network that obtains high performance for the classification task.

The rest of the work is organized as follows: A short introduction to the CNN model is given in Section 1. Section 2 briefly review related work. Then, Section 3 presents the methodology of the proposed approach.

Proceedings DOI: 10.21467/proceedings.114; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-8-8

The experimental setup and evaluation of the results are provided in Section 4. After that, a conclusion about the paper is given in Section 5.

2 Related Work

A broad range of existing hyperparameter optimization methods is shown in Figure 1. These methods are classified as black-box optimization and multi-fidelity optimization methods [5].

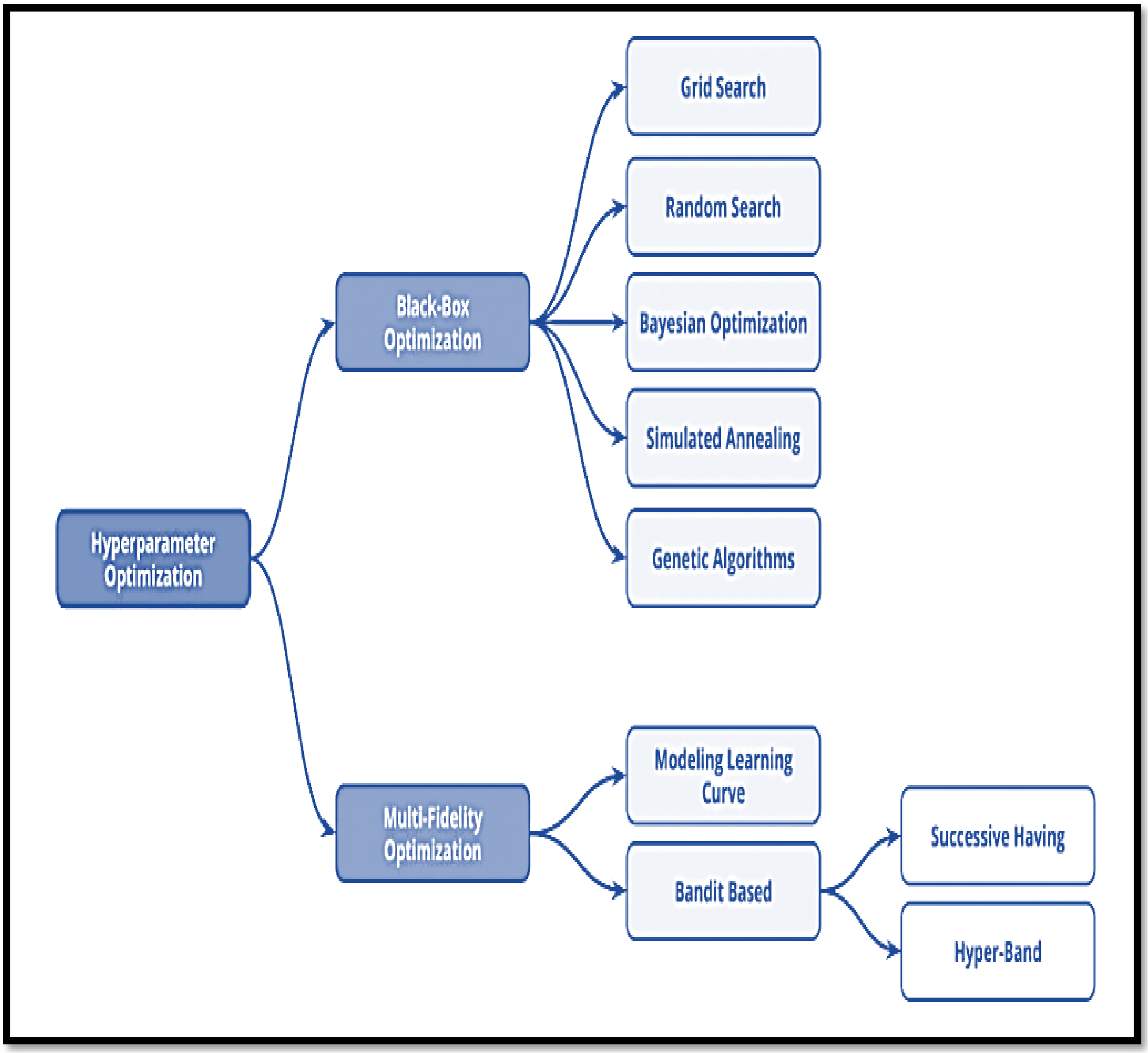


Fig. 1 Categorization of hyperparameter optimization methods [5]

The black-box optimization methods such as grid search, random search, Bayesian optimization, work on a combination of hyperparameters of the model, and find the best solutions from all different choices. While multi-fidelity optimization methods such as successive halving and hyper band allocate weights to the most feasible algorithms and discard half of the algorithms and continue this until the best algorithm is selected. This makes multi-fidelity methods perform faster than black-box methods as shown in Figure 2 [6].

Prior work on hyperparameter optimization has been designed and applied to different research problems. For example, Snoek *et al.* [7] designed the Bayesian optimization method with an acquisition function for exploring algorithms in the search space. This method used the Gaussian process for improving the algorithm in the search space. The problem with this method is that it is computationally expensive. Fu *et al.* [8] proposed a gradient descent method that automatically discovered the CNN framework. Baker *et al.* [9] developed a mechanism that explored layer- by-layer CNN network. A Q-learning agent is built that optimizes all hyperparameters. Zoph *et al.* [10] applied reinforcement learning with a recurrent neural network that discovers structures of the CNN model.

In the literature, various researchers have designed Evolutionary Algorithms (EA) for optimizing hyperparameters of a network. These are metaheuristic algorithms that evolve genes in large search space and produce as well as pass high-quality solutions to the next generations. Real *et al.* [11] used different mutation operations for evolving CNN architectures but have a high computational cost. Suaganuma *et al.* [12] applied a Cartesian genetic programming encoding approach into the model architecture for tuning hyperparameters. Rojas *et al.* [13] proposed Genetic algorithms that parallelized several processors to achieve good results and saves from trapping in local minima or local maxima.

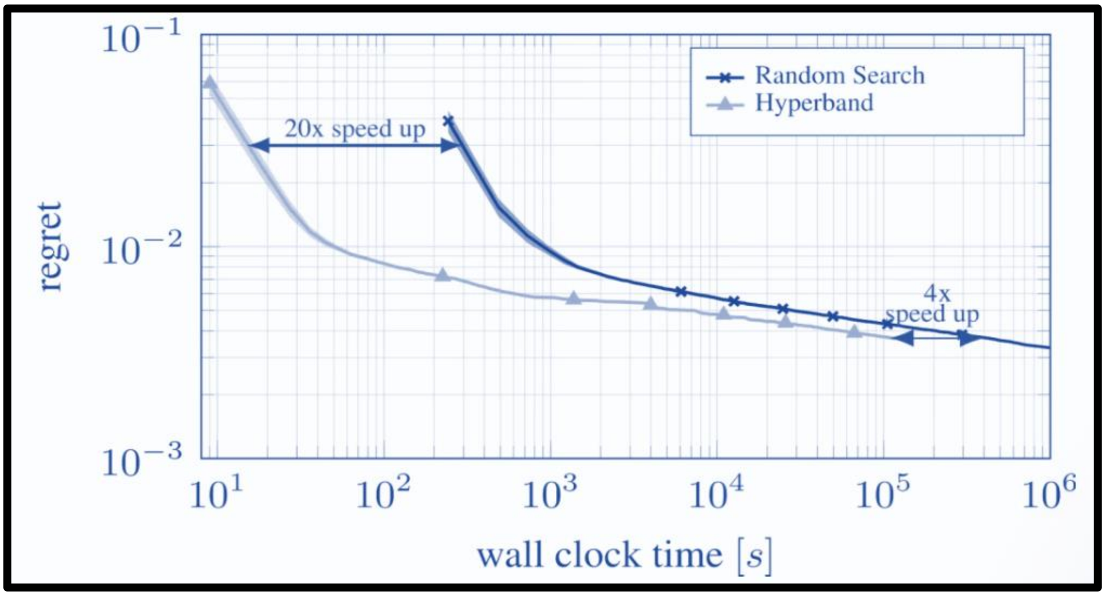


Fig. 2 Comparison of the speed of Random search and Hyperband methods [6]

3 Methodology

This section describes an efficient hyperparameter tuned small-sized CNN framework for the classification of tuberculosis CXR images. The proposed CNN model shown in Figure 3 has three convolution layers with three dropout layers. Each convolution layer has Rectified Linear Unit (ReLU) and max-pooling layer. ReLU activation function maintains non-linearity and adaptability in pixel data. And, Max Pooling passes the highest values in the next section. This model has hyperparameters such as the number of neurons, batch size, kernel constraints, dropout rate, learning rate, momentum, and the number of epochs that should be optimized and set to best feasible values for accurate classification of tuberculosis cases from normal cases. We used the hyperband optimization method for tuning these hyperparameters, that extract features in a 3-layer small-sized CNN model. Unlike successive halving algorithms, hyperband does not evaluate the performance of all hyperparameter configurations but addresses grid search on only the best feasible values of hyperparameters. Hyperband applies aggressive early-stopping by exploiting adaptive budget allocations to hyperparameter congratulations instead of fixed budget allocations. It considers a predefined space for exploring hyperparameter configuration and budget allocation that results in minimum validation loss.

The proposed hyperband tuned CNN model is applied on the NLM-China CXR dataset, which classifies the presence or absence of tuberculosis disease in CXR images.

Proceedings DOI: 10.21467/proceedings.114
ISBN: 978-81-947843-8-8

Series: AIJR Proceedings
ISSN: 2582-3922

4 Experiment and Results

4.1 Dataset setup

We considered the NLM-China CXR dataset as the experimental dataset for the proposed model. This dataset has 336 CXR images with tuberculosis cases and 326 as normal. We split training, validation, and testing images into the ratio of 70%, 10%, and 20% respectively. Input images are resized to 224*224*1. The CXR images are low contrast images, so preprocessing is done by applying the Contrast-Limited Adaptive Histogram Equalization (CLAHE) method.

4.2 Evaluation Metrics

We use performance metrics such as accuracy, and cross-entropy loss to evaluate the classification of tuberculosis disease in CXR images.

a) Accuracy: Accuracy calculates the ratio of true predictions to the total number of predictions.

$$\text{Accuracy} = (TP+TN)\ /(TP+TN+FP+FN) \qquad (1)$$

b) Cross entropy loss: It is a loss function that measures relative entropies between probability distributions A and B.

$$H(A, B) = -\text{sum x in X } A(x) * \log(B(x)) \qquad (2)$$

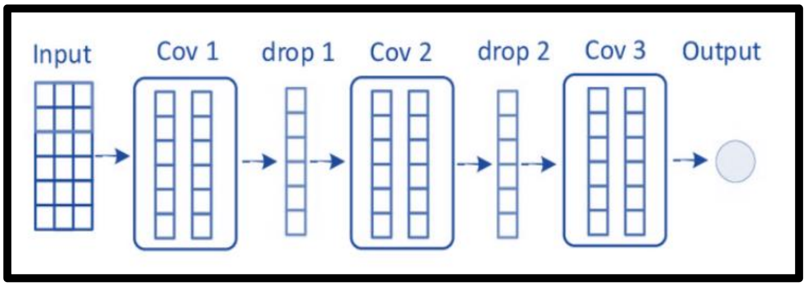


Fig. 3 Architecture of small size CNN model

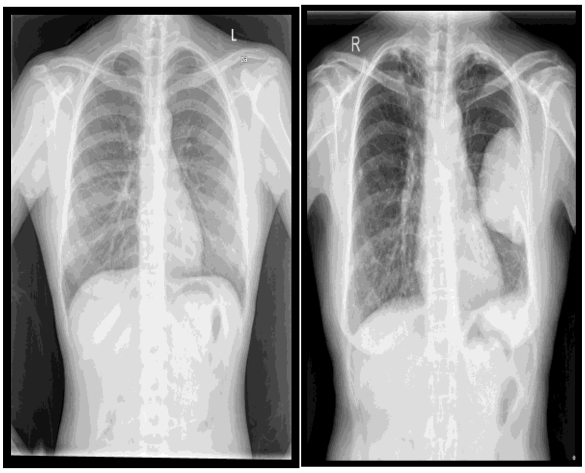
TABLE1: HYPERPARAMETERS OF THE PROPOSED SMALL SIZE HYPERBAND TUNED CNN MODEL

Hyperparameter	**Abbreviation**	**Range**
Dropout rate	dropout_rate	[0.0, 0.2, 0.4]
Epochs	epochs	[100, 150, 200]
Neurons	Neurons	[10, 32, 64]
Kernel Constraint	weight_constraint	[1,2,3]
Batch size	batch_size	[]10,20,40]
Learning Rate	learning_rate	[0.001,0.01,0.03]
Momentum	momentum	[0.9,0.95,0.99]

5 Results

All experiments are implemented in python and conducted on a computer that has specifications such as Intel Corei5, 1.19 GHz CPU, 8G RAM, and Windows 10 operating system. The model achieves a test accuracy of 91.42% and the time taken is 167 minutes. Figure 4 shows the classification results of the proposed model, where 4 (a) represents normal CXR mage and 4(b) represents CXR image with tuberculosis disease. The best hyperparameter configuration achieved has batch size=10, epochs=150, dropout rate=0.0, kernel constraint=2, neurons=32, learning rate=0.03 and momentum= 0.99.

We compare test error results of the proposed small size CNN model with state-of-the-art methods. Table 2 shows that the test error result of the proposed model for the classification task is very less in comparison to other models. Figure 5 graphically represents the distribution of test errors for all models.



(a) (b)

Fig. 4. (a) Normal CXR image 4 (b) CXR image with tuberculosis disease

Table 2: Classification Results of the Proposed Small CNN Model and Other State-of-the-Art Models

Method	**Test Error %**
LeNet-5 [14]	0.95
Recurrent CNN [15]	0.31
Gated Pooling CNN [16]	0.29
Proposed small CNN	0.17

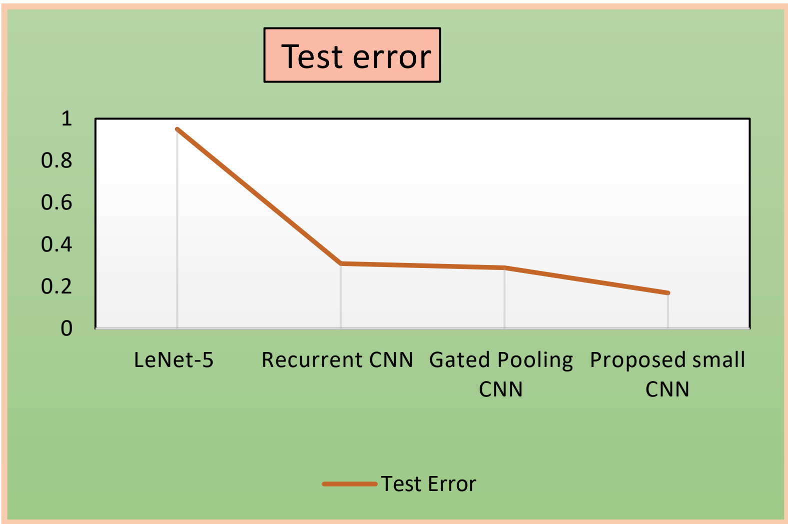


Fig. 5. Distribution of test errors for all models

6 Conclusion

In this paper, we introduced a hyperband tuned small size CNN model that classifies CXR images for tuberculosis disease. The proposed model used a fast hyperband optimization method for efficient tuning of hyperparameters. This optimization method uses aggressive early-stopping with adaptive budget allocation for finding the best feasible hyperparameter configuration. A small-sized CNN model with three convolution layers is designed for this classification problem. Experiments performed on the NLM-China CXR dataset demonstrates that the proposed small size CNN model achieved the highest accuracy score of 91.42% with a test error of 0.17.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, DOI: 10.1109/5.726791.

[2] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015). https://doi.org/10.1038/nature14539

[3] M.D. Seiler, R.Fergus, Visualizing and understanding convolutional networks, Springer, European Conference on Computer Vision, pp. 818-833, 2014.

[4] J. Bergstra, J. B. Ca, and Y. B. Ca, “Random Search for Hyper-Parameter Optimization Yoshua Bengio,” 2012.

[5] Elshawi, R., Maher, M., & Sakr, S. (2019). Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*.

[6] https://www.automl.org/blog-2nd-automl-challenge/

[7] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, pages 2951–2959, Lake Tahoe, Nevada, USA, 2012.

[8] Jie Fu, Hongyin Luo, Jiashi Feng, Kian Hsiang Low, and Tat-Seng Chua. DrMAD: Distilling Reverse-Mode Automatic Differentiation for Optimizing Hyperparameters of Deep Neural Networks. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pages 1469–1475, 1 2016.

[9] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing Neural Network Architectures using Reinforcement Learning. arXiv preprint arXiv:1611.02167, 11 2016.

[10]] Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. arXiv preprint arXiv:1611.01578, 11 2017.

[11] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. Large-Scale Evolution of Image Classifiers. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, pages 2902–2911, Sydney, NSW, Australia, 2017.

[12] Masanori Suganuma, Shinichi Shirakawa, and Tomoharu Nagao. A genetic programming approach to designing convolutional neural network architectures. In Proceedings of the Genetic and Evolutionary Computation Conference on - GECCO ’17, pages 497–504, Berlin, Germany, 4 2017. ACM Press.

[13] R. Rojas, Neural Networks: A Systematic Introduction. Springer-Verlag, 1996.

[14] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick ´ Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] Ming Liang and Xiaolin Hu, “Recurrent convolutional neural network for object recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.

[16] Chen-Yu Lee, Patrick W. Gallagher, and Zhuowen Tu, “Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree,” in International Conference on Artificial Intelligence and Statistics, 2016, pp. 464–472.