International Journal of Computer Applications Technology and Research
Volume 9–Issue 12, 311-322, 2020, ISSN:-2319–8656

Constructive Learning of Deep Neural Networks for Bigdata Analysis

Soha Abd El-Moamen Mohamed
Information System Department
Faculty of Computer and
Information Assiut University
Egypt

Marghany Hassan Mohamed
Computer Science Department
Faculty of Computer and
Information Assiut University
Egypt

Mohammed F. Farghally
Information System
Department
Faculty of Computer and
Information Assiut University
Egypt

Abstract. The need for tracking and evaluation of patients in real-time has contributed to an increase in knowing people's actions to enhance care facilities. Deep learning is good at both a rapid pace in collecting frameworks of big data healthcare and good predictions for detection the lung cancer early. In this paper, we proposed a constructive deep neural network with Apache Spark to classify images and levels of lung cancer. We developed a binary classification model using threshold technique classifying nodules to benign or malignant. At the proposed framework, the neural network models training, defined using the Keras API, is performed using BigDL in a distributed Spark clusters. The proposed algorithm has metrics AUC-0.9810, a misclassifying rate from which it has been shown that our suggested classifiers perform better than other classifiers.

Keywords: Health care, Deep learning, Constructive Deep learning, Diagnosis systems, Big data analysis

1 INTRODUCTION

Medical applications can predetermine the parts of lungs damaged in increasing levels and this is important at stages of lung cancer. We can collect images of damaged parts and diagnose the affected parts early. Medical data has three types: structured, semi-structured, and unstructured data. [4] explains an overview of unstructured data that considered as the foundation of predictive analysis. Here's the power and importance of unstructured data for predictive analysis. So, the organizations shouldn't neglect unstructured data. Structured data is the data which has a fixed and normalized shape or format. Structured or unstructured datasets can be handled effectively according to the perspective of a big data framework [18]. Machine learning and deep learning methodologies thus offer alternate

approaches to these problems and an explicit connection to big, high-dimensional datasets [12].
It might be reasonable to choose Neural Network (NN) architecture through manual design if there are qualified human experts with sufficient advanced knowledge of the problem to be solved. However, this is clearly not the case for some real-world situations where a number of advanced information is not available. NN deploys a fixed architecture in the back-propagation algorithm, while a constructive algorithm takes on dynamic NN architectures. Constructive algorithm is a supervised learning algorithm and its dynamic models are commonly used to solve real-world problems. Constructive algorithm starts a NN with a limited design, i.e. with a limited hid- den layers, neurons and links. First, start searching for a simple NN approach, then attach hidden nodes and weights gradually until an optimal methodology is identified. This paper analyzes a more proficient and massive data processing framework, Apache Spark. Apache-spark allows to solve iterative Machine Learning (ML) problems by using a distributed deep learning module called Spark BigDL framework which is a Apache Spark framework that depends on distributed deep learning [3]. Deep learning is useful in acceleration at data processing even if this data is structured or unstructured [18]. For managing high-dimensional datasets, we used a constructive deep-learning with Spark. Identifying in healthcare systems, [8] multi- disease simulations implemented.
The contribution of this work is to perform a distributed Convolution Neural Network (CNN) using constructive deep neural network algorithm called cascade-correlation growing Deep Learning Neural Network Algorithm (CCG-DLNN) by using Spark BigDl framework.
The paper is set out as following. Section 2 presents a related works. Section 3 includes a comprehensive overview of the proposed Algorithm. Section 4 is about results and experimental analysis and a conclusion is at Section 5.

2 RELATED WORKS

One of the most common cancer worldwide is the lung cancer. We generally apply different strategies for the detection of lung cancer. The detection of pulmonary nodules is using deep learning algorithms based on the CT (Computed Tomography) images exhibit. A deep neural network can be used to identify hidden details within morphological features and merging the learned representations with the initial morphological features as in [13] which used Stacked Denoising Auto Encoder (SDAE) for deep learning functionality. Extraction of CT images is by using ROI segmentation which saved using a compression technique [11]. [13] proposed a technique using the size of the nodule which has results better than im- age segmentation using ROI for detecting cells with cancers [11]. [21] performed the image preprocessing using mean and median filters, then it segmented the lung of CT image by Otsu’s threshold and a segmentation approach called marker-controlled Watershed. The physical dimensional measure and the gray-level co-occurrence matrix (GLCM) method are performed for the feature extraction step. Finally, it detected the cancer nodules from these resulted features. The best methods to classify images to detect cancer are GLCM and SVM. [19] make sputum color images using the CAD (computer-aided design) system, and this is different from NC ratio and circularity which are types of feature extraction derives properties of models and estimate cells as cancerous using a threshold.
[9] used deep learning applications for exploring non-small-cell lung cancer. extracting quantitative imaging features Prognostic signatures during deep learning networks, quantitative imaging features prognostic signatures and patient stratification scrutinized physiological imaging artifacts. [2] used nondelta features resulted from clinical tomography images for analyzing the features of delta radiomics to improve result from these images. Also, they used a ROC curve to improve the mechanism performance at lung cancer prediction with a multiple number of features. Moreover, the paper improved performance by utilizing the radiomics conventional features.

[20] extracted local features at video sequence by using the elastic solutions on the distributed environment based on the Spark paradigm and the bag of visual words (BoV) model state of the art. Then the paper computed feature maps of CNN by adopting the BigDL library.

Our focus in this paper is to design a distributed CNN model using a constructive deep neural network that effectively models the CT lung images. We used TensorFlow and Spark to force the parallelism of data and Spark scheduling, to enable tensor communication directly using parameter servers and TensorFlow executors. Direct Process-to-process communication enables TensorFlow program to scale without effort.

3 THE PROPOSED APPROACH

A detailed framework that is mentioned in this paper will be discussed in this section. The proposed approach combines the benefits of frameworks that used for big data processing (i.e. Spark) with the benefits of a constructive deep neural network algorithm called cascade-correlation growing Deep Learning Neural Network Algorithm (CCG-DLNN).

3.1 Cascade-Correlation Growing Deep Learning Neural Network Algorithm (CCG-DLNN)

Constructive neural networks (CoNN) are a set of algorithms that changes the design of the network, automatically generating a network with an acceptable dimension. The algorithms used for the CoNN learning method are called constructive algorithms. Constructive algorithm (As in Fig. 1) begins with a small network design and adding ayers, nodes, and connections as appropriate during training. The architecture adaptation process continues until the training

algorithm gets a near optimal solution for the problem.

The CCG-DLNN algorithm uses the same strategy like Cascade Correlation Neural Network algorithm (CCNN) [5] but by adding more than one hidden layer - that has more than one neuron - between the input and output layers. So, at each iteration we add a new hidden layer with one neuron or add a new neuron at the last added hidden layer (l = NHL). At the beginning, the learning algorithm begins with a simple network that has only the input and out- put layers and does not have any hidden layers. Due to the lack of hidden neurons, this network can be learned by a simple gradient descent algorithm ap- plied individually to each output neuron. New neurons are connected to the network one by one during the learning process. Each of them is put in a new hidden layer and linked to all of the previous hid- den neurons at the previous hidden layer. When the neurons are eventually connected to the network (activated), their input connections will freeze and no longer change.

Adding new neuron can be divided into two sections. First, we start with a candidate neuron and receiving input connections from all pre-existing hidden units at last hidden layer (l = NHL). The candidate unit's output isn't yet added to the current network. We do a sequence of passes over the examples set for training, updating the weights of the candidate unit for each pass at input side. Second, the candidate is linked to the output neurons (activated) and then all output connections are trained. The entire process is repeated multiple times until the desired accuracy of the network is achieved.

let C is introduced to measure changes in loss function when a new neuron is added to the current hid- den layer and τ is the threshold to add a new hidden layer. As shown in equation 1 and equation 2, If The loss function L (φ_{t-1}) at iteration (t − 1) is degraded compared to L(φˆt) at iteration (t) before adding the new neuron, then increment C by one. Otherwise, C will be zero. While C less than the threshold τ, add one new neuron to the last hidden layer. If C reaches the threshold value τ then adding a new hidden layer that has one neuron as in Fig. 2.

$$\begin{cases} C = 0 & if L(\varphi_{t-1}) - L(\hat{\varphi}_t) > \epsilon \\ C = C + 1 & otherwise \end{cases} \tag{1}$$

$$\begin{cases} l = l + 1 & if C \geq \tau \\ N_{ul} = N_{ul} + 1 & otherwise \end{cases} \tag{2}$$

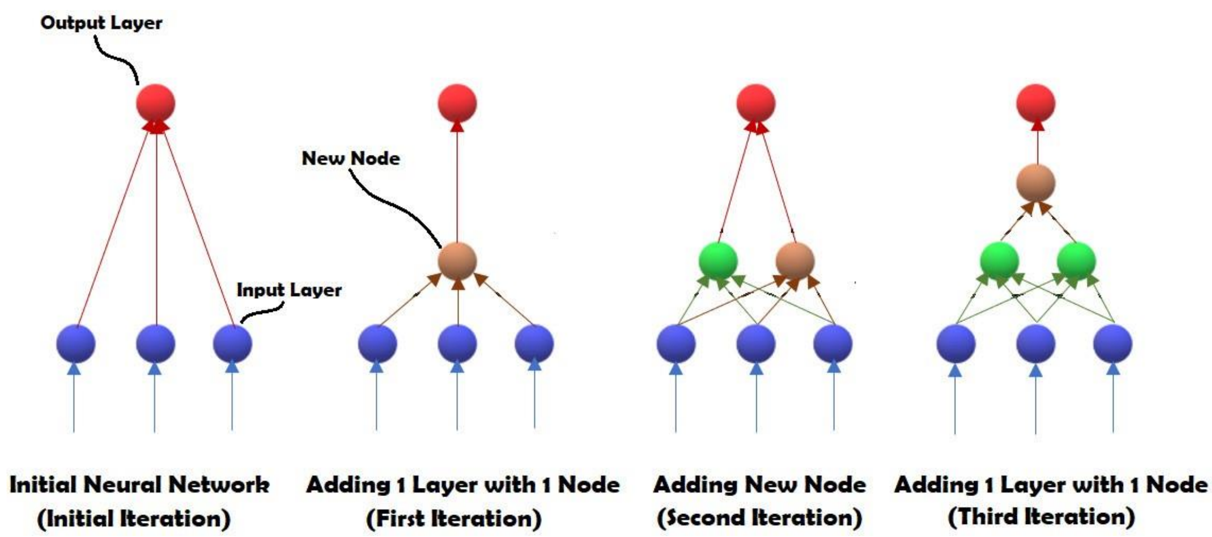


Figure 1: Constructive Deep Neural Network

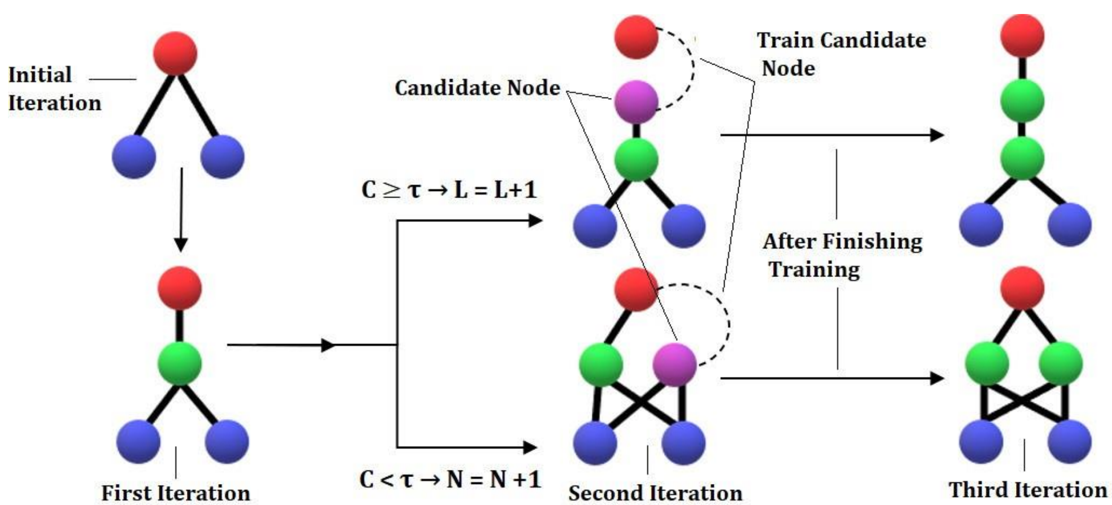


Figure 2: CCG-DLNN Algorithm

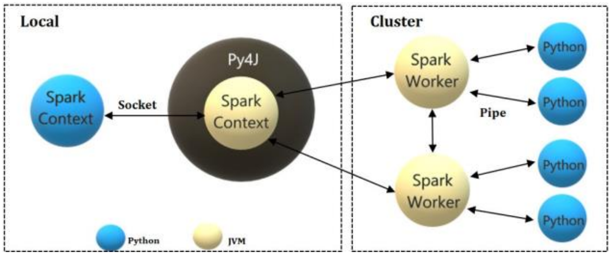


Figure 3: PySpark Framework

3.1 Framework

To solve real-world data problems like the classification of lung cancer images, we propose a big data analytic framework with constructive deep neural net- work. Addressing such problems is difficult because of time and space constraints. Because of the huge data and the lack of devices with high computing power, supportive learning frameworks are needed that can handle this data using only the stipulated sources.

To overcome these challenges, this paper combines big data analysis, machine learning and constructive deep learning. The basic structural framework presented in this section forms the core of our research and experience.

3.1.1 Apache Spark

Apache Spark can analyze faster than Apache Hadoop. It is considered as a general-purpose computing system and provides an open-source, distributed cluster system. Scala programming is what Spark is based on. Like java, Scala compiles firstly at bytecode using JVM to process the big data using Spark. PySpark is released by the Apache Spark com- munity for supporting python at Spark. For Spark programming, Pyspark API is developed in python and for developing spark applications at Python as shown at Fig. 3.

In this paper, we combine TensorFlow and Spark [1] to perform the parallelism for the data processing and Spark scheduling to enable tensor communication directly using executors of TensorFlow and parameter servers.

3.1.2 Distributed Convolutional Deep Neu- ral Network in big data analytics

There are many hierarchical layers at the Convolutional Neural Network (CNN). These layers divided into feature maps and classification layers. CNN accepts data from the input layer and sends it to the convolutional layer as shown in Fig. 4. The convolutional layer does convolution operations by having the same size filter maps. Then, the output of the convolutional layer is directed to the sampling layer to reduce layer size. There are many numbers of deep learning methods which are locally connected to the CNN [10]. The limited shared memory is considered as a challenge in big data analytics. Researchers com- bine the layers of convolution and sampling at only one step [17]. So, the activities and error values are stored at a single step when applying backpropagation.

At the proposed framework, a distributed convolution deep learning is used in big data analysis as shown in Fig. 4 to learn features and classify nodules in CT Lung images into malignant or benign nodules.

3.1.3 The Steps of a proposed framework

1. Image pre-processing:
   The first step at the proposed framework is the image pre-processing which has two steps:

   (a) The first step at the image pre-processing is image smoothing. Median filters are applied to input image which helps in reducing image noises. Median filters remove all elements that have a high frequency from input images for providing smoothed and accurate intensity surface image as an output [16].

   (b) Then the dual-tree complex wavelet trans- form (DTCWT) algorithm that was pro- posed by [14] is performed. The DTCWT is important in solving shift variance problems and lowness at directional selectivity in two or more dimensions pictures. It is based on a discrete wavelet transform (DWT). It calculates the mean energies of the real and imaginary parts of separated

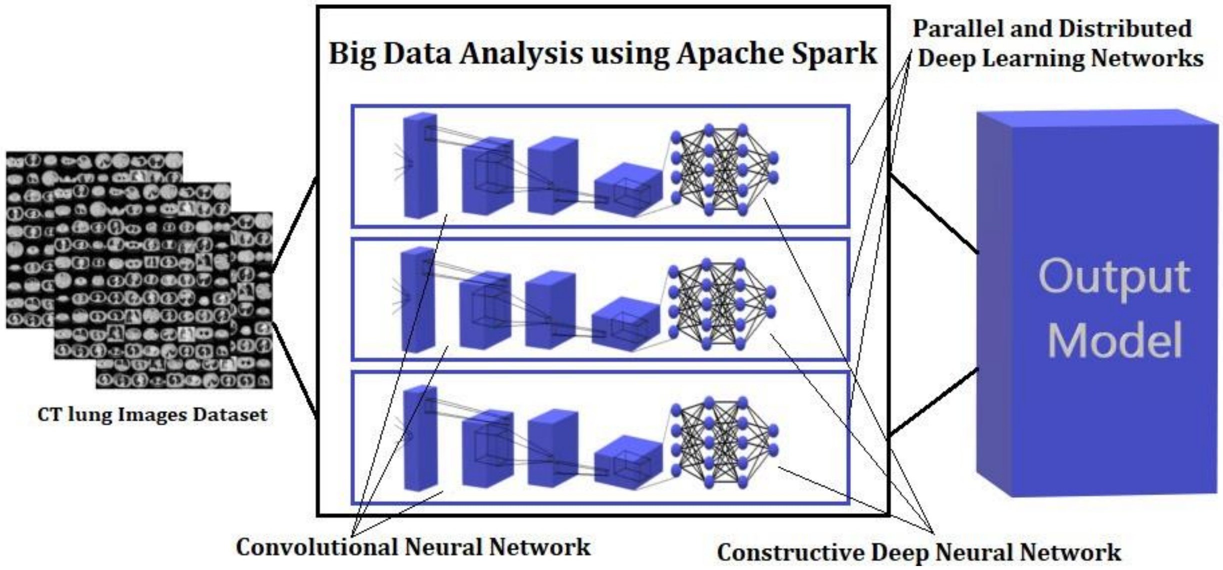


Figure 4: The proposed Model

complex wavelet coefficients. Then, these energies identify the image effective features for defect detection. DTCWT includes dual-tree shift-invariance and selective orientation for surveying the wavelets which are 2D or 3D. DTCWT helps in denoising at volume and image.

2. Feature Extraction:

In our framework, we use Gray Level Co-occurrence Matrix (GLCM) that proposed by [7] for extraction the features from the MRI lungs images. The extraction of the feature is a dimensional reduction process. It transforms data into feature sets.

A Gray Level Co-Occurrence Matrix (GLCM) has information of pixels' positions at the image which have the same values of gray levels. The GLCM entropy feature is used in this paper. Entropy is used for measuring the missing information or message in the transmission process and also measures the information into image (complexity of an image).

The entropy equation is shown at Equation 3 [7].

if the probability values P (i, j) is allocated uniformly along the matrix of GLCM, then the highest Entropy value is found. This occurs when the image hasn't pairs of grey level, with particular preference over others.

$$f = -\sum_i \sum_j p(i,j) log(p(i,j)) \quad (3)$$

where i, j is the cells number of GLCM matrix.

3. Data parallelism:

In data parallelism, the data is partitioned into small subsets. Each data partition enters as an input to each executor.

At each executor, there is a copy of CNN and a constructive deep neural network which take the subset data as input and parallelize the processing of gradient descent.

The server receives gradient delta from each executor and then synchronizes the model parameters between executors. Then, the output of each executor is combined as shown at Fig. 4.

For task scheduling and partitioning data, we used Spark at the proposed framework. Each executor has a Spark wrapper of the Tensorflow application.

Each executor has one node that handles the synchronization of the parameter and the remaining nodes run independently the application of Tensorflow. After batch elements processing and receive the last parameter from the server, the executor sends the delta of its gradient to the parameter server.

At the same time, Spark core sends the partition of the data to each executor.

The count of data partitions is based on the count of epochs the size of the dataset. Spark driver handles tasks and replicates the Tensor-Flow model to each cluster.

Tasks are created and sent to all executors for each stage and partition. The driver generates new tasks for the new stage and sends these tasks to executors, after complete all tasks of one stage. This task repeats up to the last stage and sends the results to the driver [15].

4. CNN and Constructive deep neural network:
As shown at Fig. 4, each executor has a Convolutional neural network (CNN) model which based on a constructive deep neural network.
As shown at Fig. 5, CNN consists of two main layers which are the layer of feature extraction and the fully-connected layer. The layer of the feature extraction consists of (a convolution, a nonlinear, a pooling, and an overlay layer). The fully connected layer is responsible for taking the output of the pooling layer and classifying the image into a label [22]. At our proposed framework, we replace the traditional fully connected layer at CNN model with a constructive deep neural network algorithm called CCG-DLNN algorithm (see 3.1). The CCG-DLNN algorithm has two benefits: Firstly, the cascade architecture where adding the hidden node to the neural network each time and doesn't modify after inserting it. Secondly, it is a learning algorithm that creates and adds a new hidden neuron.

4 EXPERIMENTAL RESULTS

4.1 Dataset

We used the The Lung Image Database Consortium image collection (LIDC-IDRI) [6].
The LIDC-IDRI dataset is an available set which has 1018 lung CT scans from many different organizations and universities. There are 3 structures types at this dataset:

1. The largest diameter of lung nodules which is wider than 3 mm.
2. The largest diameter of lung nodules which is narrower than 3 mm.
3. The largest diameter of Nonnodule structures which is wider than 3 mm.

We divided the Motor The LIDC-IDRI dataset into two subsets of 734 training and the remaining 284 cases for validating.

For performing our framework, the preparation steps include pre-processing (Median filters for image smoothing Fig. 6 and, The dual-tree complex wavelet transform for solving shift variance problems and lowness at directional selectivity) is performed firstly. Feature extraction is performed after pre- processing step by using e Gray Level Co-occurrence Matrix (GLCM) for extraction the features from the MRI lungs image.

4.2 Cloud platform

The proposed model is deployed on Google Cloud Dataproc which is a managed service for big data processing. Our model is based on BigDL library. BigDL is a library which based on distributed deep learning and developed for bringing deep learning that is supported to Apache Spark. The pre-processed CT-lung image data was stored in the cloud through Google Storage Bucket. We use Google Dataproc Cluster with the following versions (Debian=9,

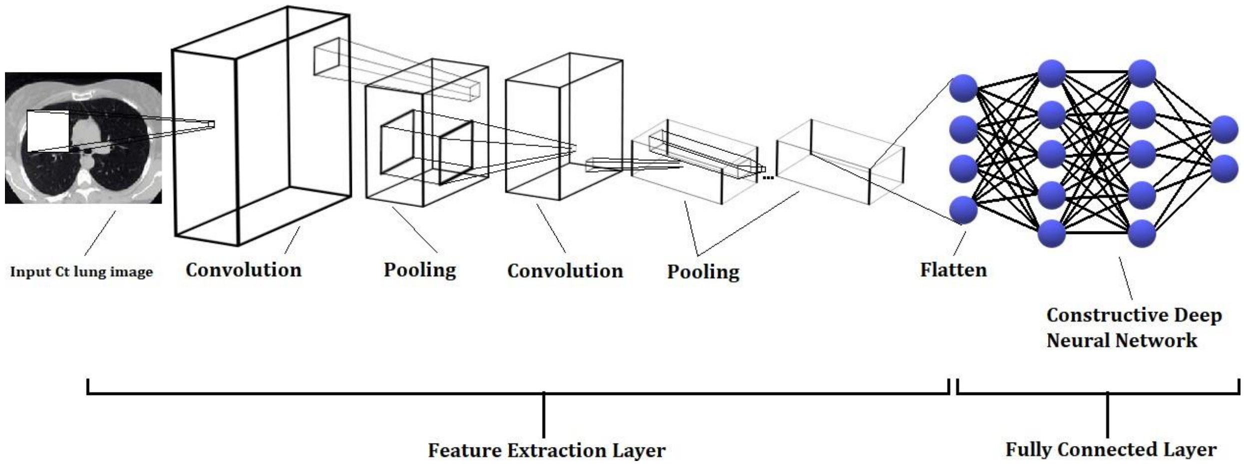


Figure 5: CNN and Constructive deep neural network

Hadoop=2.9, Apache Spark=2.3.4, Cloud Dataproc image version=1.3, TensorFlow=2.2.0, Keras=2.4.3 and python=3). The cluster has Master node and many Worker nodes. Master node contains the YARN Resource Manager, HDFS Name Node, and The pre-processed CT-lung image data was stored in the cloud through Google Storage Bucket. We use Google Dataproc Cluster with the following versions (Debian=9, Hadoop=2.9, Apache Spark=2.3.4, Cloud Dataproc image version=1.3, TensorFlow=2.2.0, Keras=2.4.3 and python=3). The cluster has Master node and many Worker nodes. Master node contains the YARN Resource Manager, HDFS Name Node, and all job drivers. Master node has 4 cores CPU, 15 Gb memory and, 500 Gb disk size. Each worker node contains a YARN Node Manager, a HDFS Data Node and The HDFS replication factor is 2. And also, each worker node has 4 cores CPU, 15 Gb memory and, 500 Gb disk size.

To measure the performance, we used the Area under ROC curve (AUC) for comparing the proposed classifier which uses CNN with constructive deep neural network algorithm (CCG-DLNN) with the traditional CNN model. Fig.7 shows the ROC curve for the proposed architecture that used a CCG-DLNN algorithm with the CNN model is AUC=0.9810. How- ever, the ROC curve for the normal CNN model has AUC=0.9666.

Fig.8 shows the training speed for the proposed classifier. The highest throughput is when executing 10 executors and with 3 or 5 cores per executor with 9300 and 28030 respectively.

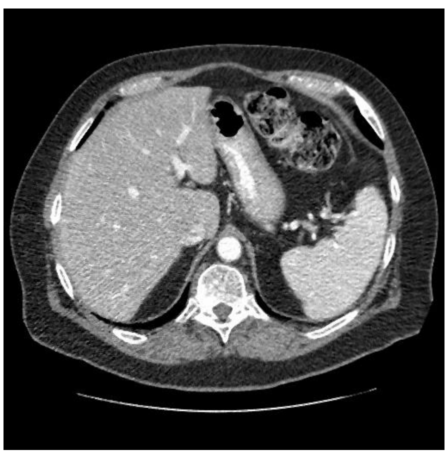
Fig.9 and Table 1 show that the training loss of the proposed framework which add the constructive algorithm CCG-DLNN to the CNN model, less than the traditional CNN model at all iterations.

Fig.10 shows the results of the training speedup of the proposed model for a different nodes number and cores at CPU. The results show that when the number of cores increased (from 10 to 100), the training time decreased.

5 CONCLUSION AND FUTURE WORK

This paper uses Apache Spark processing framework to analyze CT-lung image data. Apache-spark allows to solve iterative Machine Learning problems by using a distributed constructive deep learning module called Spark BigDL. The proposed approach com- bines the benefits of frameworks that used for big data processing (i.e. Spark) with the benefits of a constructive deep neural network algorithm called cascade-correlation growing Deep Learning Neural

(a)



(b)

Figure 6: (a) The Original CT Lung Image (b) CT Lung Image after performing median filter× with 3 3 window size.

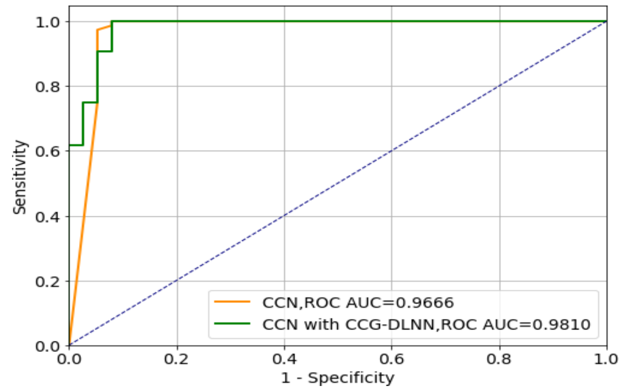


Figure 7: ROC curve for proposed architecture

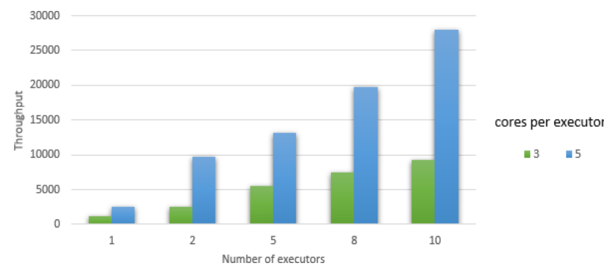


Figure 8: Throughput when changing the executor's number and number of cores at each executor for the proposed architecture

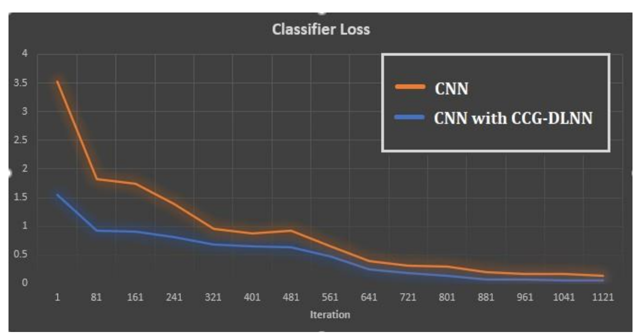


Figure 9: The comparison at loss rate between the traditional CNN model and the proposed framework

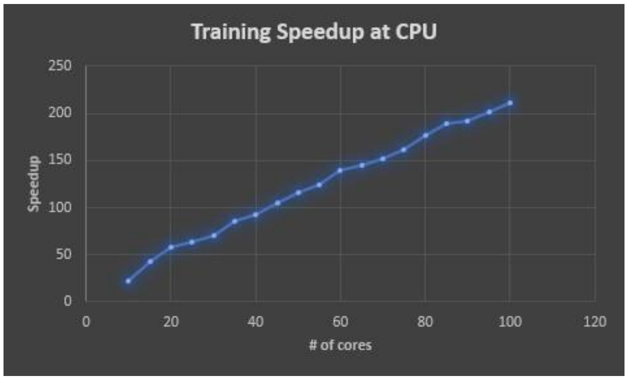


Figure 10: Training speed-up versus the numbers of CPU cores

Table 1: The comparison at loss rate between the traditional CNN model and the proposed framework

Iterations	CNN	CNN with CCG-DLNN
1	3.5	1.6
81	1.8	0.8
161	1.6	0.6
241	0.9	0.2
321	0.7	0.04
401	0.9	0.02
481	0.4	0.01
561	0.3	0.01
641	0.3	0.01
721	0.2	0.01
801	0.1	0.01
881	0.04	0.01
961	0.02	0.01
1041	0.01	0.01
1121	0.01	0.01

Network Algorithm (CCG-DLNN). The data is partitioned into small subsets. Each data partition enters as an input to each executor. At each executor, there is a copy of CNN and a constructive deep neural net- work which take the subset data as input and parallelize the processing of gradient descent. The server receives gradient delta from each executor and then synchronizes the model parameters between executors. Then, the output of each executor is combined. Our results showed that our model can increase to many of CPU cores.

The results show that the proposed architecture that used a CCG-DLNN algorithm with the CNN model is better than the traditional CNN model as the ROC curve for our proposed framework is AUC=0.9810. However, the ROC curve for the normal CNN has AUC=0.9666.

In the future, we expect that many of the tools will develop gradually. We can regard that the development will make improvements and will gain from the knowledge learned by a many number of developers.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensor-flow: Large-scale machine learning on hetero- geneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)

[2] Alahmari, S.S., Cherezov, D., Goldgof, D.B., Hall, L.O., Gillies, R.J., Schabath, M.B.: Delta radiomics improves pulmonary nodule malig-nancy prediction in lung cancer screening. IEEE Access **6**, 77,796–77,806 (2018)

[3] Dai, J.J., Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., Jia, X., Zhang, C.L., Wan, Y., Li, Z., et al.: Bigdl: A distributed deep learning

framework for big data. In: Proceedings of the ACM Symposium on Cloud Computing, pp. 50– 60 (2019)

[4] Eberendu, A.C.: Unstructured data: an overview of the data of big data. International Journal of Computer Trends and Technology **38**(1), 46–50 (2016)

[5] Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Advances in neural information processing systems, pp. 524–532 (1990)

[6] Hancock, M.C., Magnan, J.F.: Lung nodule malignancy classification using only radiologist- quantified image features as inputs to statisti- cal learning algorithms: probing the lung image database consortium dataset with two statistical learning methods. Journal of Medical Imaging **3**(4), 044,504 (2016)

[7] Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. IEEE Transactions on systems, man, and cybernetics (6), 610–621 (1973)

[8] Harimoorthy, K., Thangavelu, M.: Multi-disease prediction model using improved svm-radial bias technique in healthcare monitoring system. Journal of Ambient Intelligence and Humanized Computing pp. 1–9 (2020)

[9] Hosny, A., Parmar, C., Coroller, T.P., Grossmann, P., Zeleznik, R., Kumar, A., Bussink, J., Gillies, R.J., Mak, R.H., Aerts, H.J.: Deep learning for lung cancer prognostication: A ret- rospective multi-cohort radiomics study. PLoS medicine **15**(11), e1002,711 (2018)

[10] Jan, B., Farman, H., Khan, M., Imran, M., Islam, I.U., Ahmad, A., Ali, S., Jeon, G.: Deep learning in big data analytics: A comparative study. Computers & Electrical Engineering **75**, 275–287 (2019)

[11] Kaucha, D.P., Prasad, P., Alsadoon, A., Elchouemi, A., Sreedharan, S.: Early detection of lung cancer using svm classifier in biomedical image processing. In: 2017 IEEE International Conference on Power, Control, Signals and In- strumentation Engineering (ICPCSI), pp. 3143– 3148. IEEE (2017)

[12] Khan, M.A., Karim, M., Kim, Y., et al.: A two-stage big data analytics framework with real world applications using spark machine learning and long short-term memory network. Symme- try **10**(10), 485 (2018)

[13] Kim, B.C., Sung, Y.S., Suk, H.I.: Deep feature learning for pulmonary nodule classification in a lung ct. In: 2016 4th International Winter Conference on Brain-Computer Interface (BCI), pp. 1–3. IEEE (2016)

[14] Kingsbury, N.G.: The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters. In: IEEE digital signal processing workshop, vol. 86, pp. 120–131. Cite- seer (1998)

[15] Makkie, M., Huang, H., Zhao, Y., Vasilakos, A.V., Liu, T.: Fast and scalable distributed deep convolutional autoencoder for fmri big data analytics. Neurocomputing **325**, 20–30 (2019)

[16] Sazzad, T.S., Ahmmed, K.T., Hoque, M.U., Rahman, M.: Development of automated brain tumor identification using mri images. In: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–4. IEEE (2019)

[17] Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architec- tures for object recognition. In: International conference on artificial neural networks, pp. 92–101. Springer (2010)

[18] Sujitha, R., Seenivasagam, V.: Classification of lung cancer stages with machine learning over big data healthcare framework (2020)

[19] Taher, F., Werghi, N., Al-Ahmad, H.: Rule based classification of sputum images for early

lung cancer detection. In: 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 29–32. IEEE (2015)

[20] Tu, N.A., Huynh-The, T., Wong, K.S., Bui, D.M., Lee, Y.K.: Distributed feature extraction on apache spark for human action recognition. In: 2020 14th International Conference on Ubiq- uitous Information Management and Communi- cation (IMCOM), pp. 1–6. IEEE (2020)

[21] Tun, K.M.M., Khaing, A.S.: Implementation of lung cancer nodule feature extraction using dig- ital image processing (2014)

[22] Zhang, Y., Wang, S., Zhao, H., Guo, Z., Sun, D.: Ct image classification based on convolutional neural network. NEURAL COMPUTING & APPLICATIONS (2020)