Pneumonia and Eye Disease Detection using Convolutional Neural Networks

Parnasree Chakraborty
Electronics & Communication Engineering
BSA Crescent Institute of Science & Technology
Chennai, India

C. Tharini
Electronics & Communication Engineering
BSA Crescent Institute of Science & Technology
Chennai, India

Abstract—**Automatic disease detection systems based on Convolutional Neural Networks (CNNs) are proposed in this paper for helping the medical professionals in the detection of diseases from scan and X-ray images. CNN based classification helps decision making in a prompt manner with high precision. CNNs are a subset of deep learning which is a branch of Artificial Intelligence. The main advantage of CNNs compared to other deep learning algorithms is that they require minimal pre-processing. In the proposed disease detection system, two medical image datasets consisting of Optical Coherence Tomography (OCT) and chest X-ray images of 1-5 year-old children are considered and used as inputs. The medical images are processed and classified using CNN and various performance measuring parameters such as accuracy, loss, and training time are measured. The system is then implemented in hardware, where the testing is done using the trained models. The result shows that the validation accuracy obtained in the case of the eye dataset is around 90% whereas in the case of lung dataset it is around 63%. The proposed system aims to help medical professionals to provide a diagnosis with better accuracy thus helping in reducing infant mortality due to pneumonia and allowing finding the severity of eye disease at an earlier stage.**

Keywords-convolutional neural network; artificial intelligence; X-rays; pneumonia

I. Introduction

A medical image based disease detection system using CNN is proposed in this paper. The suggested system has the ability of detecting pneumonia and eye disease from X-rays and scan images respectively. The novel feature of the proposed system is that it has been implemented using low cost hardware. In [1], a diagnostic system is proposed for detecting retinal diseases. The result shows that the performance of the proposed method is comparable to that of human experts. However, the implementation of the system using hardware is not suggested. A computationally efficient algorithm is introduced in [2]. Adam stochastic optimization method is used to train the neural network. Empirical results demonstrate that Adam works well in practice and compares favourably to other stochastic optimization methods. In [3], the effect of the convolutional network depth on its accuracy is investigated and changes in architectural configuration which improve the accuracy of the algorithm are proposed. A deep-learning-based approach to detect diseases and pests in tomato plants using images is presented in [4]. The images are captured in-place by camera devices with various resolutions and are processed. The experimental results show that the proposed system can effectively recognize nine different types of diseases and pests in tomato plants. In [5], the Face Detection and Face Recognition pipeline framework (FDREnet) is proposed which involves face detection through histograms of oriented gradients and uses Siamese technique and contrastive loss to train a deep learning architecture. However, disease detection is not investigated in this paper. On the other hand, a review of the applications of AI in soil management, crop management, weed management and disease management can be seen in [6], but disease management and disease detection in humans using AI are not investigated.

II. Datasets used

In order to test the proposed idea, two datasets were considered. The Lung dataset consisted from images from [7] and the eye dataset from images from [8]. Data are essential to train any neural network. The neural network, apart from other parameters, is only as good as the data it is trained on. For training the CNN, medical image data are used. Two different kinds of publicly available medical image datasets are considered for training two convolutional neural networks. OCT images in the iris region of the eye are considered for eye disease detection. OCT is a non-invasive method of capturing biological tissues using low-coherence light. It can capture two dimensional and three dimensional images of micro meter level. The images of the OCT scan are classified under 4 categories: i) choroidal neovascularization, ii) diabetic macular edema, iii) multiple drusen, and iv) normal. Choroidal neovascularization is the creation of new blood vessels in the choroid region of the eye. This problem is a major cause of vision loss. Macular edema is build-up of fluid in an area in the center of the retina. This build up causes the macular to thicken, distorting vision. Drusen consists of multiple deposits under retina. Drusen is a fatty protein made up of lipids. Having drusen may increase the possibility of age-related macular degeneration. The dataset contains normal/healthy iris scan images too. The images are collected from [8] dataset which contains more than 5GB of 84438 images from [9, 10] which are classified on the above mentioned categories. Chest x-ray images of children belonging to 3 classifications: i) viral pneumonia, ii) bacterial pneumonia, and iii) normal were taken from [7] and are considered in this study. Pneumonia is an

Corresponding author: P. Chakraborty (prernasree@crescent.education) infection that accumulates in the lung's air sacs causing hindrance for breathing. The lung image dataset contains 1GB of 5238 images belonging to the 3 above mentioned categories. Both datasets were split into three sets: train, test and validation.

Figure 1 describes the training of a neural network. The given data is split into training, validation and testing data with each utilizing 70%, 20% and 10% of the data respectively. After each iteration of training, the neural network is tested with the validation data to see its performance at that instant. After completing the whole training process, the performance is evaluated using the testing data. This proposed method is heavily inspired from [11] and a similar neural network with the lung data which was presented in [12]. The image datasets [7-10] are first collected and annotated or labeled in order to distinguish the normal images from images with diseases. To generate the training dataset the existing labeled data are further used to generate a new dataset using a technique called augmentation. Annotated and augmented data are used for training the proposed neural network.

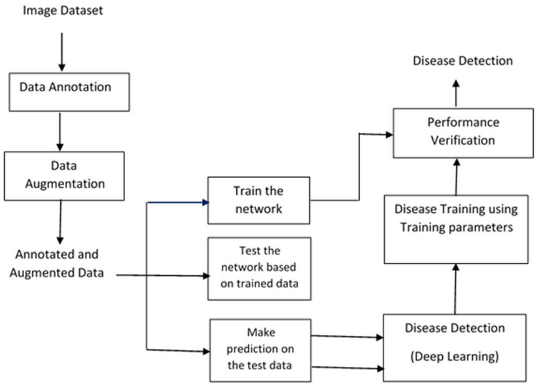


Image Dataset
Data Annotation
Data Augmentation
Annotated and Augmented Data
Train the network
Test the network based on trained data
Make prediction on the test data
Disease Detection (Deep Learning)
Disease Training using Training parameters
Performance Verification
Disease Detection

Fig. 1. Block diagram of the proposed system

III. CONVOLUTIONAL NEURAL NETWORKS

CNNs [3] are a type of deep artificial neural networks, used mainly to identify and cluster images, and perform object recognition. A CNN consists of image processing layers and neural network layers namely: (a) convolutional layer, (b) pooling layer, (c) flattening layer, (d) ReLU layer, and (e) Softmax layer. These layers are described briefly below.

A. Convolutional Layer

The convolutional layer is the main building block of a CNN. The layer's parameters consist of a set of user-defined learnable filters (or kernels), which is generally a 3×3 matrix, and iterates through each submatrix of the input. The number of input filters used is generally of the order of 2N. During a forward pass, each filter is convolved across the dimensions of the input image matrix, the mathematical function carried out being dot product and thus producing a 2-dimensional feature-extracted matrix of that filter. This reveals various details like vertical or horizontal edges of the images which are extracted and fed into the next layer. The weights that are used are generated randomly using the Glorot uniform distribution function. Figure 2 shows the filters. Figure 3(c) demonstrates the output image when an input image, shown in Figure 3(a) is convolved with the one of the above displayed filters.

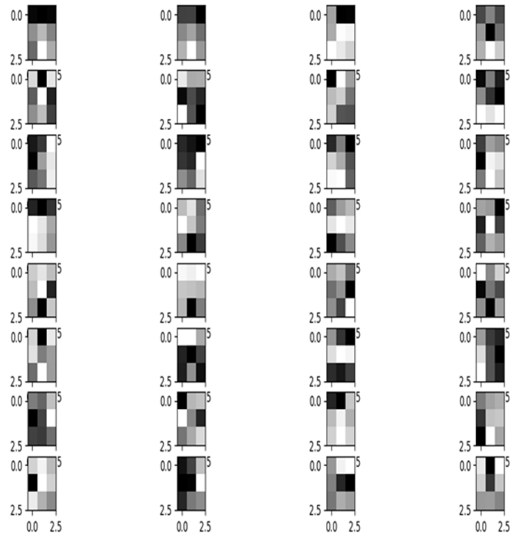
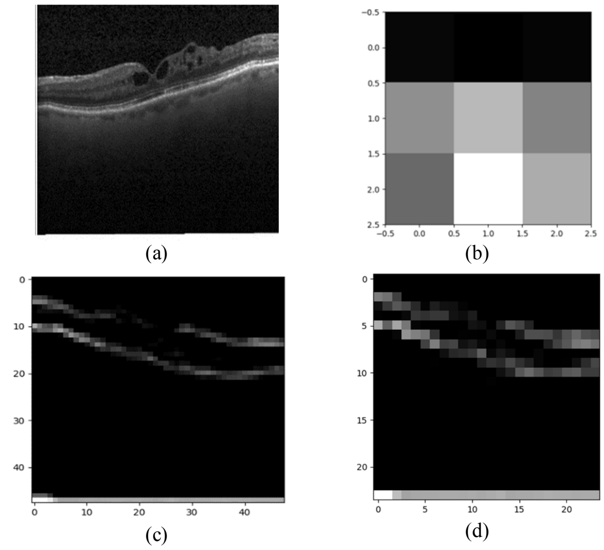


Fig. 2. The 32 filters used in the proposed CNN

B. Pooling Layer

Another important concept used in CNNs is pooling, which is a form of non-linear down-sampling. Out of the several pooling functions analyzed in [13], max-pooling is the most effective. Max-pooling partitions the input image into a set of (n×n) (generally 2×2) sub matrices and the output is the maximum value. The convolved image is first converted into arrays and then maxpooling is performed. Figure 3 displays the convolution and maxpooling steps. In maxpooling the dimensions of the image are reduced from a 50×50 matrix to a 24×24 matrix.



(a) (b)

(c) (d)

Fig. 3. (a) Input image, (b) filter function, (c) resultant convoluted image, and (d) the pooling layer's output shrinking the 50×50 image to a 24×24 image

C. Flattening Layer

The output from the pooling layer will be in a matrix form which can't be fed into the neural network. The flattening layer converts the n×n matrix from the pooling layer into a $n^2 \times 1$ matrix which is a compatible format to be fed into the neural network.

D. RELU Layer

ReLU is the abbreviation of Rectified Linear Unit, which applies a non-saturating activation function. These functions remove negative values of weights by replacing them with zero. It increases the nonlinear properties of the decision function. This activation function is used in input and hidden layers of the neural network. The type of ReLU used is leaky ReLU. ReLU as explained in [14] is used in the neural network layers. Figure 4 shows the leaky ReLU activation function. Mathematically, the Leaky ReLU can be defined as:

$$y = 0.01\ x \quad \text{when } x < 0 \qquad (1)$$

$$y = x \quad \text{when } x > 0 \qquad (2)$$

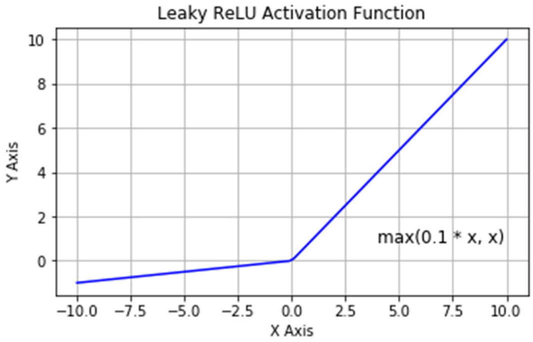


Fig. 4. Graphical representation of the ReLU

E. Softmax Layer

This layer is predominantly used when the neural network solves multiclass-classification problems. It usually consists of a number of output nodes with Softmax as activation function. Softmax function assigns probability to each node in the output layer. These probability values are normalized to one. The node with highest value is the prediction of the neural network. The ReLu layer and the Softmax layer both use backpropagation [15] and forward propagation to train the CNN. Figure 5 shows the softmax function.

Mathematically, softmax function can be defined as:

$$\sigma(z)_i = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \qquad (3)$$

where i= 1, 2,….k and z= z_1, z_2,…..z_k. Equation (3) shows the standard exponential function to each element z_i of the input vector **Z** and normalizes these values by dividing by their sum. This normalization ensures that the sum of the components of the output vector σ(z) is 1.

1) Loss Function

Loss function or cost function generally is the difference between the actual output and the predicted output. The main aim of the loss function is to reduce error. i.e. to minimize the difference between the predicted value and the actual value. The loss function predominantly used in both datasets is mean squared error. In this method, the difference between the predicted and the actual output is squared. It is better than the gradient descent methods for decreasing loss [16]. The sum of all these squares is divided by their total number. Mathematically this can be represented as

$$\frac{1}{n} \sum_{i=1}^{n} (Yi - \bar{Y}i) \qquad (4)$$

where n is the number of inputs, Y_i is the actual output and $\hat{Y}_i$ is the predicted output.

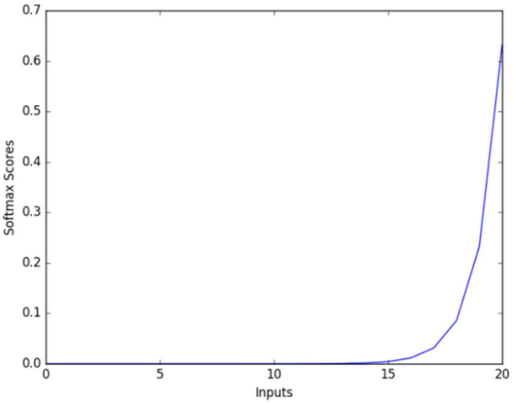


Fig. 5. Graphical representation of the softmax function

2) Optimizer

The optimizer is a function which is guided by the loss function to update the weights so that the loss is minimized. It does so by changing the learning rate after every iteration in accordance with the calculated loss function. The weights of each node change based on the learning rate. If the learning rate is too fast, the neural network may not learn enough to generalize. If the learning rate is too low, the neural network may learn very slowly. The neural network needs to learn in an optimum speed and optimum manner and that is helped by the optimizer function. The optimizers used were the Adam optimizer and the Root Mean Square Propagation optimizer.

3) Adam Optimizer

It is one of the best optimizers available. It is computationally efficient, it augments optimized learning and has very little memory requirements. Adam [2] stands for adaptive moment estimation. Instead of changing the weights based on the first moment (mean) alone or based on the second moment (variance) alone, this uses both first moment and second moment to update the learning parameters:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\tilde{v}t} + \varepsilon}\ \hat{m}_t \qquad (5)$$

where m_t and v_t are first moments and second moments respectively, and η is the learning rate.

IV. RESULTS AND DISCUSSION

Neural networks with different architectures have been considered. The architecture of a neural network mainly depends on parameters such as the optimizer type, the number of nodes in each layers etc. The results are discussed below.

A. Lung Dataset

Table I shows the results for the lung dataset [7]. The architecture comprises of an input layer, multiple hidden layers and an output layer. The training accuracy, training losses, validation accuracy and validation losses with respect to the number of iterations used for simulation are listed. The simulation is performed using Python Integrated Development Environment (IDE) Spyder. The maximum validation accuracy obtained in the case of the lung dataset is only around 63% with 10 epochs/iterations and 5215 steps per epoch. This result can be further improved with larger size dataset. In Table II, the complete architecture consisting of two pairs of convolution layers (named as conv2d_13 and conv2d_14), maxpooling layers (named as max_pooling2d_13 and max_pooling2d_14), and a flattening layer are shown. The optimum artificial neural network consists of an input layer consisting of 7 nodes and an output layer consisting of 3 nodes for each classification.

TABLE I. OBSERVATIONS OF THE LUNG DATASET

Iterations	Optimizer	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
10	RMSprop	0.4848	0.2122	0.4706	0.2131
15	Adam	0.7549	0.1215	0.4706	0.2901
10	Adamax	0.4849	0.2108	0.4118	0.2217
15	RMSprop	0.4779	0.3009	0.4290	0.2308
12	RMSprop	0.4851	0.2108	0.5294	0.2040
15	Adamax	0.7682	0.2286	0.5294	0.2286
10	Adam	0.7698	0.2201	0.6309	0.2252
15	Adam	0.5396	0.3547	0.3509	0.3688

TABLE II. SUMMARY OF THE NN FOR LUNG DATASET WHICH YIELDED THE BEST PARAMETERS DURING TRAINING AND TESTING

Layer (type)	Output shape	Param #
conv2d_13 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_13	Maxpooling (None, 15, 15, 32)	0
conv2d_14 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_14	(Maxpooling (None, 6, 6, 64)	0
flatten_7 (Flatten)	(None, 2304)	0
dense_13 (Dense)	(None, 7)	16135
dense_14 (Dense)	(None, 3)	24

Total params: 35,551, trainable params: 35,551, non-trainable params: 0

Param# in Tables II and IV indicates the number of input weights that is processed through that one given layer. Total params is the sum of all the input weights in the total architecture of the neural network. The output shape denotes the number of inputs at a time (given by none) followed by the expected shape of the input.

B. Eye Dataset

Table III shows the observations for eye dataset [8-10]. The maximum validation accuracy obtained in the case of the eye dataset is around 90% which can be further improved with a larger size dataset. Table IV shows the summary of the neural network model which yielded the best parameters during training and validation for eye disease detection. The maximum number of epochs used for the eye dataset ranged from 5 to 64, and the maximum validation accuracy was obtained for 15 epochs.

TABLE III. OBSERVATIONS OF THE EYE DATASET

Iterations	Optimizer	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
15	Adam	0.7060	0.1016	0.9062	0.039
5	Adam	0.6823	0.1079	0.7812	0.078
20	Adam	0.6886	0.1062	0.8438	0.072
32	Adam	0.7031	0.1031	0.0982	0.718
34	RMSprop	0.4702	0.1608	0.5625	0.143
13	Adam	0.6625	0.1120	0.7812	0.085
64	Adam	0.7379	0.0942	0.7241	0.098
10	Adam	0.7969	0.5274	0.7325	0.688

TABLE IV. SUMMARY OF THE NN FOR EYE DATASET WHICH YIELDED THE BEST PARAMETERS DURING TRAINING AND TESTING

Layer (type)	Output shape	Param #
conv2d (Conv2D)	(None, 48, 48, 32)	320
max_pooling2d (MaxPooling2D)	(None, 24, 24, 32)	0
conv2d_1 (Conv2D)	(None, 22, 22, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 11, 11, 32)	0
flatten (Flatten)	(None, 3872)	0
dense (Dense)	(None, 4)	15492

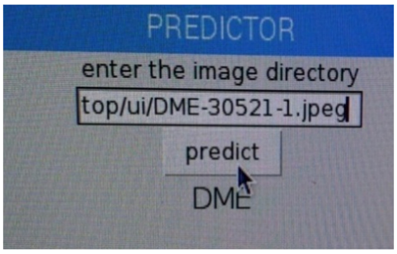
Total params: 25,060, trainable params: 25,060, non-trainable params: 0

It can be seen from Table III that for the eye dataset the optimizer predominantly used was the Adam optimizer. In the 5th trial of Table III, the RMS Prop optimizer was used. The loss was the Mean Squared Error loss function with the exception of the 8th trial where categorical cross entropy loss function was used. The complete architecture consists of two pairs of convolution layers (named as conv2d and conv2d_1) and Maxpooling layers (named as max_pooling2d and max_pooling2d_1) and a flattening layer named as Flatten.

The ANN has an output layer consisting of four nodes, one for each kind of classification. The output shape consists of a 4 dimensional array for the 2 pairs of convolutional and maxpooling layer. The 1st dimension (denoted by None in all the given pairs) is the number of inputs that will be fed into that given layer at that particular given time. It is mentioned as None because these observations were taken after training, when there was not any input to be fed into at that instant of time. The rest of the 3 dimensions mention the dimensions of a single input unit. The same holds true for remaining flattening and neural network layers.

V. DEPLOYMENT

The neural networks which yielded the best parameters were saved in h5 format and were deployed in a Raspberry Pi which uses Raspbian with features to program in Python 3.5.3. A simple Graphic User Interface (GUI) was made where the user was asked to enter the directory of the image and the neural network would make the prediction and display the result. Snapsots of the results and of the GUI output for both datasets can be seen in Figures 6-11. The complete setup used to implement the proposed system using hardware is shown in Figure 12. The hardware part includes the Raspberry Pi board for interface with the GUI using Tkinter library in Python IDE. Further, there are two ways to connect the LCD to the Raspberry Pi board: 4 bit mode and 8 bit mode. In this work, 4 bit mode was used in which the byte to be sent is split into two sets that (upper bits and lower bits) of 4 bits each which are sent one by one over 4 data wires.

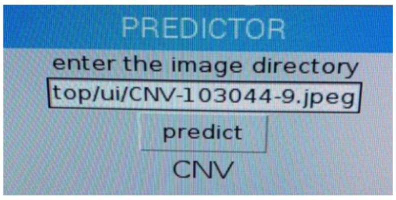


enter the image directory

top/ui/DME-30521-1.jpeg

DME

Fig. 6. GUI output of the neural network predicting the given iris image has diabetic macular edema

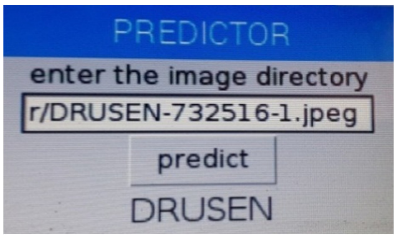


enter the image directory

top/ui/CNV-103044-9.jpeg

CNV

Fig. 7. GUI output of the neural network predicting the given iris image has choroidal neovascularisation

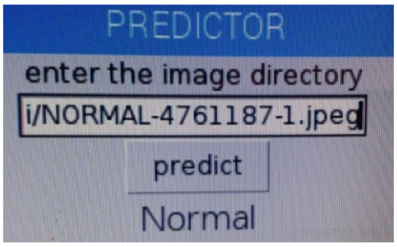


enter the image directory

r/DRUSEN-732516-1.jpeg

DRUSEN

Fig. 8. GUI output predicting the given iris image has multiple drusen

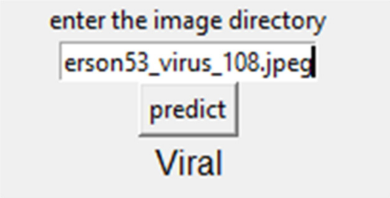


enter the image directory

i/NORMAL-4761187-1.jpeg

Normal

Fig. 9. GUI output predicting the given iris image doesn't have any disease

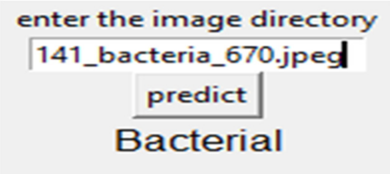


enter the image directory

erson53_virus_108.jpeg

Viral

Fig. 10. GUI output predicting the given X-ray image has viral pneumonia



enter the image directory

141_bacteria_670.jpeg

Bacterial

Fig. 11. GUI output of the neural network predicting the given X-ray image has bacterial pneumonia

Figures 13 and 14 show the eye disease detection system and the pneumonia detection system implemented in hardware.

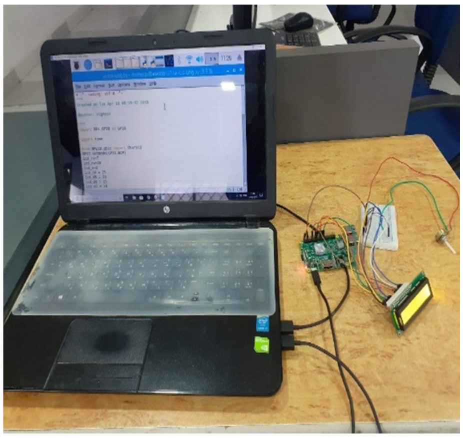


Fig. 12. Experimental setup

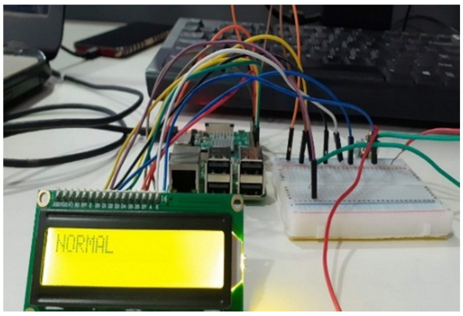


Fig. 13. Result obtained for a normal human iris scan

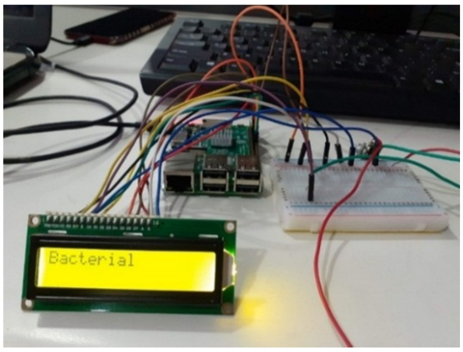


Fig. 14. Result obtained for human chest X-ray of bacterial pneumonia

VI. Conclusions and Future Work

A medical image based disease detection system using Convolutional Neural Networks is proposed and developed. The eye disease detection system effectively classifies the normal eye images and eye images with diseases like choroidal neovascularization, diabetic macular edema, and multiple drusen. Lung image dataset [7] consisted of bacterial pneumonia, viral pneumonia, and normal lung X-ray images of children in the age group of 1-5 years. The training model was simulated using Python libraries like Tensorflow, Keras, Skimage, etc. to improve training speed. The enhanced speed of the training model yielded the real time implementation of the systems more suitable. The proposed system has the potential to be used in generalized high-end applications in biomedical imaging and provides a cost effective solution at a single board computer (Raspberry Pi). Regarding future work, focus will be given in improving the current results. Another promising application will be to extend the idea for identification of various diseases not only in humans but also in plants and crops.

REFERENCES

[1] D. S. Kermany, M. Goldbaum, W. Kai et al., “Identifying medical diagnoses and treatable diseases by image-based deep learning”, Cell, Vol. 172, No. 5, pp. 1122-1131, 2018

[2] D. P. Kingma, J. Ba, “Adam: A method for stochastic optimization”, International Conference on Learning Representations, San Diego, USA, May 7-9, 2015

[3] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, International Conference on Learning Representations, San Diego, USA, May 7-9, 2015

[4] A. Fuentes, S. Yoon, S. Cheol Kim, D. Sun Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition”, Sensors, Vol. 17, No. 19, Article ID 2022, 2017

[5] D. Virmani, P. Girdhar, P. Jain, P. Bamdev, “FDREnet: Face detection and recognition pipeline”, Engineering, Technology & Applied Science Research ,Vol. 9, No. 2, pp. 3933-3938, 2019

[6] N. C. Eli-Chukwu, “Applications of artificial intelligence in agriculture: A review”, Engineering, Technology & Applied Science Research, Vol. 9, No. 4, pp. 4377-4383, 2019

[7] D. Kermany, K. Zhang, M. Goldbaum, “Labeled optical coherence tomography (OCT) and chest X-ray images for classification”, Mendeley Data, Vol. 2, 2018

[8] K. S. Mader, “Eye OCT datasets: Retina OCT datasets with accompanying fundus images from published studies”, available at: https://www.kaggle.com/kmader/eye-oct-datasets

[9] T. Mahmudi, R. Kafieh, H. Rabbani, “Comparison of macular OCTs in right and left eyes of normal people”, in: Proceedings SPIE 9038, Medical Imaging 2014: Biomedical Applications in Molecular, Structural, and Functional Imaging, 90381K, San Diego, California, USA Feb. 15-20, 2014

[10] M. K. Jahromi, R. Kafieh, H. Rabbani, A. M. Dehnavi, A. Peyman, F. Hajizadeh, Mohammadreza Ommani, “An automatic algorithm for segmentation of the boundaries of corneal layers in optical coherence tomography images using Gaussian mixture model”, Journal of Medical Signals and Sensors, Vol. 4, No. 3, pp. 171-180, 2014

[11] P. Rajpurkar, J. Irvin, R. L. Ball, et. al, “Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists”, PLOS Medicine, Vol. 15, No. 11, Article ID e1002686, 2018

[12] X. Gu, L. Pan, H. Y. Liang, R. Yan, “Classification of bacterial and viral childhood pneumonia using deep learning in chest radiography”, 3rd International Conference on Multimedia and Image Processing, Guiyang, China, March 16-18, 2018

[13] D. Scherer, A. Muller, S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition”, 20th International Conference on Artificial Neural Networks, Thessaloniki, Greece, September 15-18, 2010

[14] K. He, X. Zhang, S. Ren, J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on image net classification”, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1026-1034, IEEE, 2015

[15] Y. LeCun, L. Buttou, G. B. Orr, K. R. Muller, ”Efficient backprop”, in: Neural Networks: Tricks of the Trade, 2nd edition, Springer-Verlag, 1998

[16] Y. LeCun, L. Bottou, Y. Bengio, P. Haner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278-2324, 1998

AUTHORS PROFILE

Parnasree Chakraborty is an Assistant Professor (Sr.Grade) in the Department of Electronics and Communication Engineering at B. S. Abdur Rahman Crescent Institute of Science & Technology. Her research interests include Digital Signal Processing, AI & Robotics, Wireless Sensor Networks, and Digital Communication. She is a life member of the ISTE. She has published many papers in journals and conferences in the area of signal processing and wireless sensor networks.

C. Tharini is a Professor and the Head of the Department of Electronics and Communication Engineering at B. S. Abdur Rahman Crescent Institute of Science & Technology. She received her PhD in Information and Communication Engineering from Anna University in 2011. Her research interests include Wireless Communication, Wireless Sensor Networks and signal processing algorithms for Wireless Sensor Networks. She is an active member of the Computer Society of India. She has a teaching experience of more than 15 years. Her students are presently working in Wireless Sensor Networks, Signal Processing and in the Wireless Communication domain. She has published many papers in international journals and in conferences in the area of signal processing and wireless sensor networks.