International Conference on Innovative Research in Engineering, Management and Sciences
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 4 | Issue 9 | ISSN : 2456-3307

Carcinoma Detection using Convolution Neural Networks

Rajiv Gopal*[1], Naveen H[2], Karthik C V[3]

*[123]Asst. Professor, Dept. of ECE New Horizon College of Engineering, Bengaluru, India
rajivg@newhorizonindia.edu[1]

ABSTRACT

In this paper, we have proposed an optimized model which can predict the symptoms of breast cancer with an accuracy of 86%. The machine learned to predict test images at a more accurate rate than it could without optimization. Using Random forest, we got an accuracy of 83%. We have used Convolutional Neural Network to develop a model for breast cancer detection through a mammograph dataset. With the rapid development in deep learning, in the future, machine learning will surely bring much improvement in development of models for prediction, detection of several health issues even at an early stage and easier procedure. We have used Python language for the implementation of entire system.

Keywords : Convolution Neural Networks, Flattening, ReLU

I. INTRODUCTION

Breast Cancer is one of the leading cause of death in women all around the world. Cancer is a disease caused by the changes occurred in cells spreading uncontrollably. Lumps or masses are formed which is called tumour and it is named after the body part which it originates in. In breast cancer the pain is minimal to non-existent at early stage and can be treated easily, hence screening is important for early detection. 80% breast cancers are invasive and usually breast cancer is referred to single disease but there are up to 21 histological sub-categories. It has been observed that India has a much lower incidence of breast cancer than Western countries, even after adjusting for age structure of the population, about 1/3rd in urban areas and 1/9th in rural regions. The lack of population screening in India (and corresponding over diagnosis in Western populations) undoubtedly contributes to this statistic but more importantly, so do lifestyle, reproductive and dietary factors.

Mammography is currently one of the best methods to detect breast cancer early. The magnetic resonance imaging (MRI) is the most attractive alternative to mammogram. However, the MRI test is done when the radiologists want to confirm about the existence of the tumour. The cons of the MRI is that the patient might show an allergic reaction to the contrasting agent, or skin infection could develop at the place of injection.

The application of visual classification motivates the use of computer-aided diagnosis (CAD) systems to improve the diagnosis accuracy, reduce human error, increase the level of inter-observer agreement, and increased reproducibility. Deep learning based approaches have shown to perform better than conventional machine learning methods in many image analysis task, automating end-to-end processing. Convolutional neural networks (CNN) have been successfully used for diabetic retinopathy screening, bone disease prediction and age assessment, and other problems. Previous deep learning-based applications in histological

microscopic image analysis have demonstrated their potential to provide utility in diagnosing breast cancer. In this paper, we present a visual analysis approach for breast cancer classification. Here we make use of CNN (Convolutional Neural Networks) for feature extraction and classification.

II. MOTIVATION

The objective of using machine learning techniques is to develop a model which can be used for estimating, prediction, extraction of features, classification and so on. When a model is developed using machine learning techniques chances are there will be many training and generalizing errors. A good classification model should fit the training dataset and accurately classify all the instances. Any error on the training dataset will result to unreliable or inaccurate prediction of testing data. Also errors such as model over fitting can also occur even if the training error rate is reduced, which is achieved if the complexity of the model is increased.

Here we propose a classification model using Convolutional Neural Network (CNN), originally designed to mimic the neurons in the visual cortex, CNN are particularly good at classifying images. They work by sliding a filter around the input image, and multiplying the image by this filter at each step to create a new filtered image.

As mammography is a technique using X-rays to diagnose locate tumours in breast we considered CNN to be much ideal for developing a classification model for training, predicting, classification and other similar tasks.

III. IMPLEMENTATION

An overview of the approach towards the problem is given by the block diagram given below.

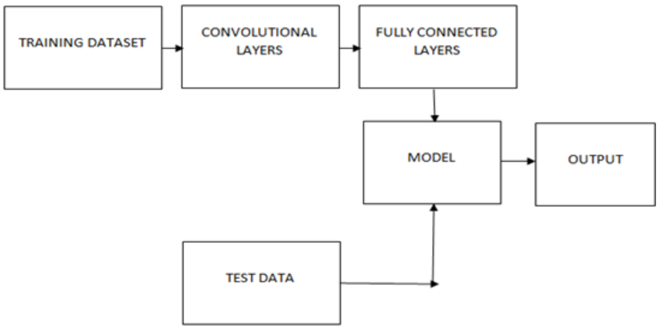

Fig. 1 Block Diagram of Breast Cancer Detection using Machine learning

A. Training Datasets

The dataset used here are a set of 9998 mammograph images which have been pre-labelled. Each images are of size: width-50, height-50, and depth-3. We split the datasets into test and training datasets in the ratio 1:4 i.e. 20% for test and remaining 80% for training. Even after splitting the dataset we make sure that the ratio of cancerous to non- cancerous data remains the same as the original dataset in the training and the test datasets.

B. Convolution Layer

It is the core building block of a CNN. Here weights, also known as filters, which is a small receptive field but extent to the full depth of the input volume. We take this filter and slide it over the image spatially and compute dot product at each spatial location. These filters learn different information present on the image like edges, bulge, corners etc. The result of the dot product is a 2-dimensional activation map of that filter. Hence a stack of such activation map for all the filters throughout the entire depth of the image forms the output volume of the convolutional layer. We can set up filters for low-level filter kernels for convolutional neural network, which means designing a filter for the input layer can be done but the hidden layer kernels are hard to engineer by human. “End-to-end learning” is where only the input and the output ends are what the human provide while all other parameters are learned or provided by the model itself.

C. Pooling Layer

Pooling layer is another building block in CNN. It helps make the representation smaller and more manageable. It works on each activation map independently. There are different methods of pooling – Mean pooling, Max pooling and Sum pooling. The most preferred being Max pooling.

D. ReLU

Rectifier is an activation function defined as the positive part of its argument. Mathematically, it is defined as

$$y = \max(0, x) \tag{1}$$

ReLU is the most commonly used activation function in neural networks, especially in CNNs. If you are unsure what activation function to use in your network, ReLU is usually a good first choice.

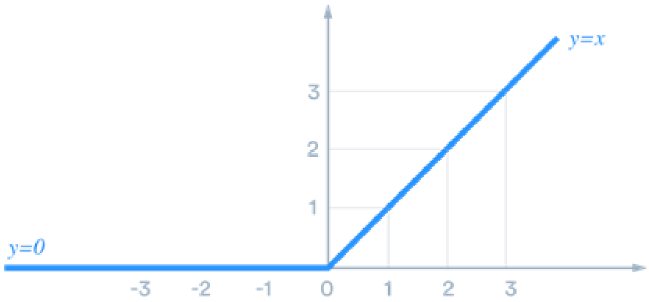

Fig. 2 ReLU activation function

E. Flattening

Flattening is the process of converting the resultant 2- dimensional array into a single continuous linear vector. This is required for feeding the collected features into the fully connected network.

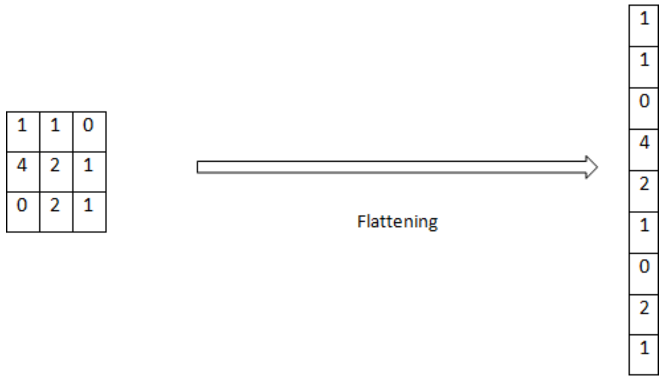

Fig. 3. Flattening process

IV. FLOWCHART

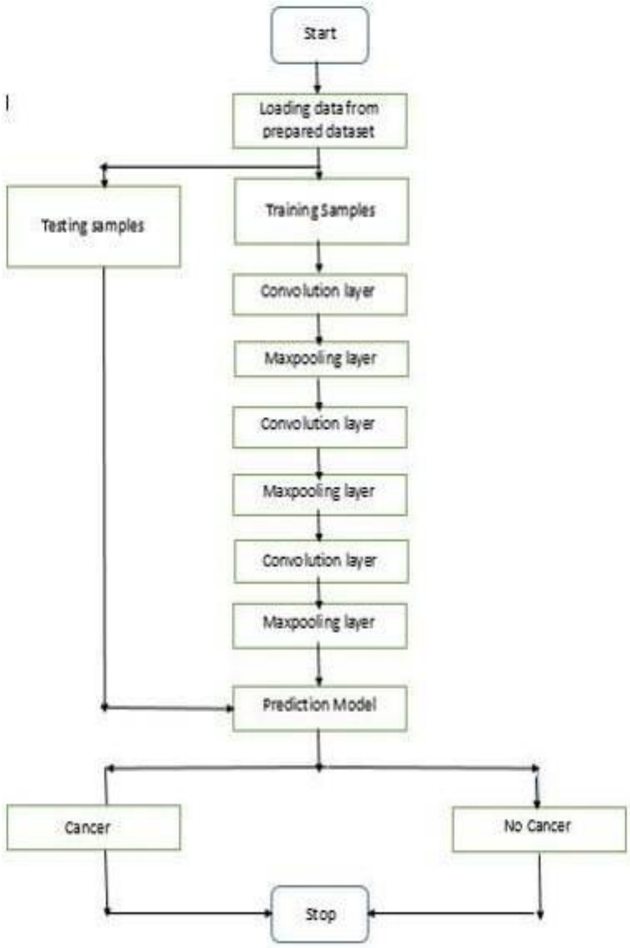

Fig. 4 Flowchart of the process

The above figure represents the flowchart of the code used in detection of cancer using CNN.

V. RESULTS

A. Classification using Random Forest

```
In [13]: v = val = val_features.reshape(val_features.shape[0], -1)

         clf.score(val, y_val)

Out[13]: 0.829
```

B. Classification using CNN

```
model.evaluate(val_features, alt_num_labs_val)

Epoch 1/15
8998/8998 [==============================] - 3s 364
Epoch 2/15
8998/8998 [==============================] - 3s 335
Epoch 3/15
8998/8998 [==============================] - 3s 335
Epoch 4/15
8998/8998 [==============================] - 3s 337
Epoch 5/15
8998/8998 [==============================] - 3s 339
Epoch 6/15
8998/8998 [==============================] - 3s 343
Epoch 7/15
8998/8998 [==============================] - 3s 352
Epoch 8/15
8998/8998 [==============================] - 3s 347
Epoch 9/15
8998/8998 [==============================] - 3s 338
Epoch 10/15
8998/8998 [==============================] - 3s 341
Epoch 11/15
8998/8998 [==============================] - 3s 339
Epoch 12/15
8998/8998 [==============================] - 3s 337
Epoch 13/15
8998/8998 [==============================] - 3s 339
Epoch 14/15
8998/8998 [==============================] - 3s 341
Epoch 15/15
8998/8998 [==============================] - 3s 342
1000/1000 [==============================] - 0s 106

Out[14]: [1.302553077191289, 0.844]
```

C. Classifiaction using Optimised version of CNN

```
Train on 8998 samples, validate on 1000 samples
Epoch 1/50
8998/8998 [==============================] - 3s 294us/step - loss: 1.0950 - acc: 0.8200 - val_loss:
Epoch 2/50
8998/8998 [==============================] - 2s 275us/step - loss: 0.6724 - acc: 0.8554 - val_loss:
Epoch 3/50
8998/8998 [==============================] - 2s 275us/step - loss: 0.5484 - acc: 0.8693 - val_loss:
Epoch 4/50
8998/8998 [==============================] - 2s 273us/step - loss: 0.4574 - acc: 0.8803 - val_loss:
Epoch 5/50
8998/8998 [==============================] - 2s 277us/step - loss: 0.4099 - acc: 0.8853 - val_loss:

Epoch 00005: ReduceLROnPlateau reducing learning rate to 3.9999998989515007e-05.
Epoch 6/50
8998/8998 [==============================] - 3s 283us/step - loss: 0.2137 - acc: 0.9292 - val_loss:
Epoch 7/50
8998/8998 [==============================] - 3s 279us/step - loss: 0.1864 - acc: 0.9363 - val_loss:
Epoch 8/50
8998/8998 [==============================] - 2s 278us/step - loss: 0.1729 - acc: 0.9402 - val_loss:
Epoch 9/50
8998/8998 [==============================] - 3s 278us/step - loss: 0.1657 - acc: 0.9447 - val_loss:

Epoch 00009: ReduceLROnPlateau reducing learning rate to 7.999999797903002e-06.
Epoch 10/50
8998/8998 [==============================] - 2s 277us/step - loss: 0.1412 - acc: 0.9540 - val_loss:
Epoch 11/50
8998/8998 [==============================] - 3s 278us/step - loss: 0.1382 - acc: 0.9540 - val_loss:
Epoch 12/50
8998/8998 [==============================] - 3s 279us/step - loss: 0.1365 - acc: 0.9568 - val_loss:
Epoch 13/50
8998/8998 [==============================] - 3s 281us/step - loss: 0.1342 - acc: 0.9574 - val_loss:

Epoch 00013: ReduceLROnPlateau reducing learning rate to 1.5999999596038884e-06.
Epoch 14/50
8998/8998 [==============================] - 3s 279us/step - loss: 0.1301 - acc: 0.9583 - val_loss:
Epoch 15/50
8998/8998 [==============================] - 3s 279us/step - loss: 0.1297 - acc: 0.9583 - val_loss:
Epoch 16/50
8998/8998 [==============================] - 3s 280us/step - loss: 0.1294 - acc: 0.9583 - val_loss:

Epoch 00016: ReduceLROnPlateau reducing learning rate to 3.200000037395512e-07.
Epoch 17/50
8998/8998 [==============================] - 3s 284us/step - loss: 0.1284 - acc: 0.9583 - val_loss:
Epoch 18/50
8998/8998 [==============================] - 3s 282us/step - loss: 0.1284 - acc: 0.9587 - val_loss:
Epoch 00018: early stopping
```

D. Graph

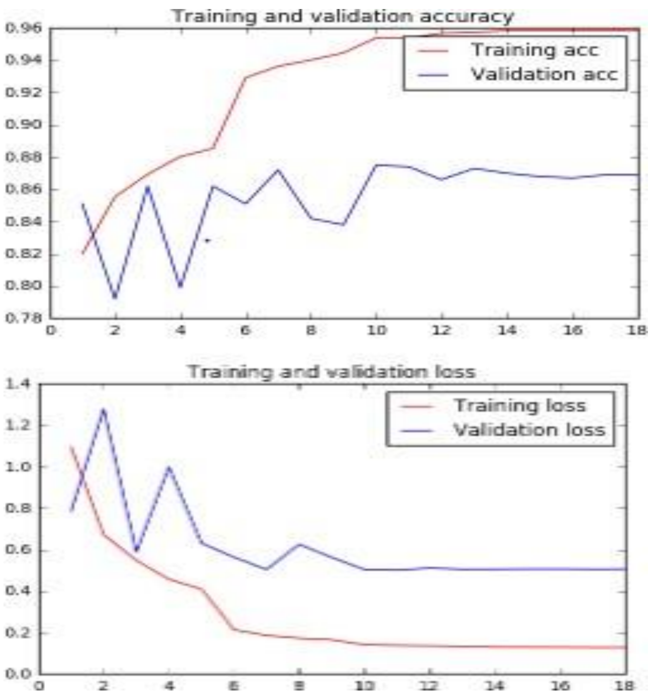

Fig. 5. Graph showing Training and Validation Loss and Accuracy

E. Prediction of Cancerous and Non-cancerous Images

```
In [20]: labs_pred = model.predict(tr_flat[:10])
         pred = [np.where(i == max(i))[0][0] for i in labs_pred]
         pred

Out[20]: [0, 1, 0, 0, 0, 0, 0, 0, 1, 0]

In [21]:
         path = r'E:/New folder/10302/1/10302_idx5_x501_y1601_class1.png'

         import cv2

         image = cv2.imread(path)
         image = image.reshape(1, image.shape[0], image.shape[1], image.shape[2])

         _sp_features = conv_base.predict(np.array(image), batch_size=BATCH_SIZE, verbose=1)

         class_lab = ["No Cancer", "Cancer"]

         sp_features = _sp_features.reshape(-1, 512)

         lab = model.predict(sp_features)
         pred = [np.where(i == max(i))[0][0] for i in lab]

         print("Diagnosis (predicted):", class_lab[pred[0]])

         1/1 [==============================] - 0s 78ms/step
         Diagnosis (predicted): Cancer
```

VI. CONCLUSION

From a total of 9998 mammograph data, with 90% data as training data and 10 % as test data we developed a optimized model which can predict with an accuracy of 86%. The machine learned to predict test images at a more accurate rate than it could without optimization. Using Random forest, we got an accuracy of 83%. This level of accuracy is decent but when it can be seen that there is almost 17% chance for the machine's prediction to go wrong. This may cost the patient to suffer more due to late detection and in worst case scenario even death. Overall, we conclude that using Convolutional Neural Network to develop a model for breast cancer detection through a mammograph dataset is way better than Random forest Classifier or any other classifiers. With the rapid development in deep learning, in the future, machine learning will surely bring much improvement in development of models for prediction, detection of several health issues even at an early stage and easier procedure.

VII. REFERENCES

[1] A. Alarabeyyat, A.M., "Breast Cancer Detection Using K-Nearest Neighbor Machine Learning Algorithm" in 9th International Conference on, IEEE, v.i.e.E.(DeSE), pp. 35-39, 2016.

[2] A. Francillon, P.R., "Smart Card Research and Advanced Applications" in 12th International Conference CARDIS 2013, Berlin, Germany:Springer International Publishing.

[3] S.K. Prabhakar, H. Rajaguru, N. Maglaveras, I. Chouvarda, P. de Carvalho, "Performance Analysis of Breast Cancer Classification with Softmax Discriminant Classifier and Linear Discriminant Analysis" in Precision Medicine Powered by pHealth and Connected Health. IFMBE Proceedings, Singapore, Springer, vol. 66, 2018.

[4] L.A. Altonen, R. Saalovra, P. Kristo, F. Canzian, A. Hemminki, P Peltomaki, R. Chadwik, A. De La Chapelle, "Incidence of hereditary nonpolyposis colorectal cancer and the feasibility of molecular screening for the disease", N Engl J Med, vol. 337, pp. 1481-1487, 1998.

[5] S. Chakraborty, "Bayesian kernel probit model for microarray based cancer classification", Computational Statistics and Data Analysis, vol. 12, pp. 4198-4209, 2009.

[6] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, "Gene selection for cancer classification using support vector machines", Machine Learning, vol. 46, pp. 389-422, 2002.

[7] S. Gokhale, "Ultrasound characterization of breast masses", The Indian journal of radiology &imaging, vol. 19, pp. 242-249, 2009.

[8] T. Jinshan, R.R., X. Jun, I. El Naqa, Y. Yongyi, "Computer-Aided Detection and Diagnosis of Breast Cancer With Mammography: Recent Advances" in Information Technology in Biomedicine, IEEE, vol. 13, pp. 236-251, 2009.

[9]

[10] A. Jemal, R.S., E. Ward, Y. Hao, J. Xu, T. Murray, M.J. Thun, "Cancer statistics", A Cancer Journal for Clinicians, vol. 58, pp. 71- 96, 2008.

[11] L. Adi Tarca, V.J.C., X. Chen, R. Romero, S. Drăghici, "Machine Learning and Its Applications to Biology", PLoS Comput Biol, vol. 3, pp. 116-122, 2007.

[12] JF McCarthy, M.K., PE Hoffman, "Applications of machine learning and high-dimensional visualization in cancer detection diagnosis and management", Ann N Y Acad Sci, vol. 62, pp. 10201259, 2004.

[13] AC. Tan, D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification", Appl. Bioinform, vol. 2, pp. 75-83, 2003.

[14] S. Kanta Sarkar, A.N., "Identifying patients at risk of breast cancer through decision trees", International

Journal of Advanced Research in Computer Science, vol. 08, pp. 88-96, 2017.

[15] JA. Cruz, W.D, "Applications of Machine Learning in Cancer Prediction and Prognosis", Cancer Inform, vol. 2, pp. 56-77, 2006.

[16] M. Sugiyama, Introduction to Statistical Machine Learning, T. Green: Morgan Kaufmann, 2006.

[17] A. Lavecchia, "machine-learning approaches in the context of ligand- based virtual screening for addressing complex compound classification problems and predicting new active molecules", D. Montesano, vol. 49, 2005.

[18] P. Baldi, S.R.B., S.r.B. Pierre Baldi, Bioinformatics: The machine learning approach, 2001.

[19] N. Bhatia, "Survey of Nearest Neighbor Techniques", International Journal of Computer Science and Information Security, vol. 8, no. 2, 2010.