

# Health Discernment System

**Pranav Jain, Rituraj Saha, Anish Kumar**

School of Computer Engineering,  
Kalinga Institute of Industrial Technology,  
Deemed to be University, Bhubaneswar, Odisha, India  
pranavjain594@gmail.com, saharituraj@gmail.com, 1706200@kiit.ac.in

**Abstract-** With broad data development in biomedical and healthcare sectors, detailed analyzes of medical data support early detection of illness, patient care and community services. However, the quality of the study is lowered when the content of the medical data is incomplete. Also, various regions exhibit unique features of certain regional diseases. This can hinder disease outbreak forecasting. In this project, we streamline deep learning algorithms to effectively predict chronic disease outbreaks in populations with recurrent diseases. The diagnosis of diseases is a critical and central aspect of medicinal science. Doctors breakdown side effects in the human body more often than not to foresee diseases. In recent times, numerous research strategies have been used with a specific goal to make it more accurate. In this system, we will provide a user-friendly interface that can be used by the users to detect whether their medical test results are positive or normal, i.e. it will detect the disease. This deep learning project is based on a user interface and its application of the Health Discernment System in real life. It will also describe how the system will perform and under what it must operate. In this document, the user interface will also be shown. Both the stakeholders(users) and the developers of the interface can benefit from this approach.

**Key Terms-** Convolutional Neural Network, Disease Detection, Deep Learning, Feature Selection, Tensorflow.

## I. INTRODUCTION

The days are long gone when data on health-care used to be small. The advancement level in devices for the acquisition of images is quite large and that is what makes image processing difficult and fascinating. This significant growth of medical images and techniques requires comprehensive and exhaustive efforts from a medical professional who is susceptible to human error and the result can also vary widely among various experts. The alternative to this approach is to use machine learning or deep learning strategies for automating the detection process of various diseases.

Machine Learning (ML) and Artificial Intelligence (AI) have made significant progress over the past few years. ML and AI techniques have influenced medical fields such as medical image processing, image recognition, computer-aided diagnosis, image segmentation, and image fusion to name a few. While automated disease detection based on conventional medical imaging methods demonstrated significant accuracies for decades, breakthroughs in machine learning approaches have sparked a growth in deep learning. Deep learning-based algorithms demonstrated remarkable outcomes in various fields such as computer-aided diagnosis, speech recognition etc.

For our project, we have used the above stated idea behind disease detection, to construct a system using Convolutional Neural Network that detects the diseases quickly and also guarantees it to be free of error. By doing so we meant to minimize the human efforts that are required to detect a medical test report. We have tried to make the system user-friendly with the help of GUI, so that it can be used not only by the medical professionals but also by the population at large.

## II. BASIC CONCEPTS/ TECHNOLOGY USED

**Deep Neural Network** - An Artificial Neural Network with several hidden layers between the input and output layers is a Deep Neural Network represented in Figure-1. A Deep Neural Network can model complicated non-linear interactions, similar to shallow ANNs. A neural network's main goal is to acquire a set of inputs, perform increasingly complex calculations on them and provide output to solve problems in the real world, such as classification. We limit ourselves to feeding on neural networks. In a deep network, we have an input, an output and a flow of sequential data.

**Convolutional Deep Neural Networks** - If we maximize the volume of layers in a neural network to make it more effective, it raises the network's complexity and helps one to model more complicated functions. The number of weights and biases, however, will rise exponentially. In reality, it can become hard for regular neural networks to think about such difficult problems. This leads to a solution, the neural networks of convolution.

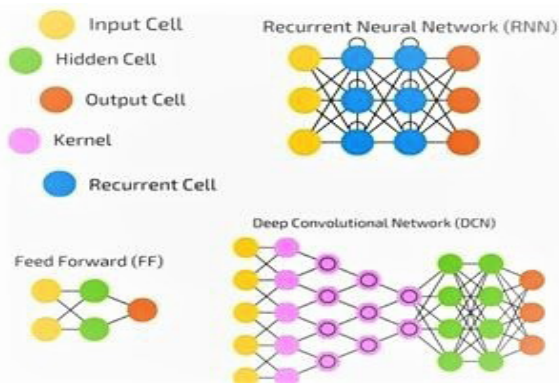


Figure 1: Deep Convolutional Neural Network.

**TensorFlow** - It is a well -documented python library which contains various algorithms for machine learning and deep learning. This python library is widely used to train and develop deep learning and machine learning models.

**Tkinter Programming** - The basic GUI library for Python is Tkinter. Python offers a quick and simple way to build GUI applications when imported with Tkinter. Tkinter provides the Tk GUI toolkit with a versatile object-orientated GUI.

### III. STUDY OF SIMILAR PROJECTS OR TECHNOLOGY / LITERATURE REVIEW

**Breast cancer** - There are various ways of detecting breast cancer including mammography, MRI scans, computed tomography (CT) scans, ultrasound, and nuclear imaging. Although, none of these approaches provides a perfectly accurate cancer prediction. Artificial neural networks rely, for example, on MLE (Maximum Likelihood Estimation) [1]. RBF Neural Networks on their work [2], the GRU-SVM model which is an ML algorithm coupled with a type of recurrent neural network (RNN) and gated recurrent unit (GRU) with support vector machine (SVM) [3]. Another method is the Fuzzy-artificial immune system and the K-NN algorithm listed in Ref. [4]. Descriptors like CLBP, GLCM, LBP, LPQ, ORB, PFTAS are defined in paper [5] with breast cancer classification up to 85.1% accuracy.

**Pneumonia** - Recent advances in deep learning models and the access to large datasets have enabled algorithms to outshine medical workers in various diagnostic imaging tasks, such as detection of skin cancer, haemorrhage identification, arrhythmia detection [6], and diabetic retinopathy detection [16]. Automated diagnosis enabled by chest X-rays has taken on huge interest. These algorithms are progressively used to detect pulmonary lung nodule [7] and pulmonary tuberculosis classification [8]. The performance of many convolutional models on various abnormalities relying on the OpenI database available to the public [9] discovered that the same deep convolutional network architecture doesn't well work across all abnormalities [10], ensemble models dramatically improved classification accuracy compared to a single model, and finally, the deep learning method improved accuracy compared to rule-based approaches.

**Malaria** - Malaria is commonly diagnosed by microscopic blood cell analysis using blood films[2]. Nearly 167 million blood films had been tested for malaria using microscopy during 2010 which was less expensive and less complex than the diagnosis based on polymerase chain reaction [11]. Liang et al. have proposed an approach based on deep learning for classifying cells infected with malaria from red blood smears. Their proposed method is based on a 16-layer convolutional neural network that uses the AlexNet

architecture[17] pre-trained on the CIFAR-100 [12] dataset to outshine their transfer learning-based model.

**Skin cancer** - Deep learning algorithms have recently gained huge success in various computer vision issues. Krizhevsky et al.[13] in 2012 inbuilt a novel technique (AlexNet) using convolutional neural networks for classifying a large data (1,2 million images) containing 1000 categories of objects in the 2010 ImageNet Large Scale Visual Recognition Challenge (ILSVRC2010) and delivers the highest result and, therefore, tremendous interest among academics in the field of computer vision. Dorj et al. [15] developed ECOC SVM with such a deep, convolutional neural network approach to classify 4 diagnostic categories of images of clinical skin cancer. The clinical images of 12 skin diseases were classified by a deep convolutional neural network by Han et al.[14].

#### IV. PROPOSED MODEL / TOOL

The proposed idea was implemented using Python in the form of a desktop application having a GUI and made using the *Model-View-Controller (MVC)* design pattern. The application is ".exe" installable. The Model and the View can run independently using the MVC pattern. The Model module can be used separately in other programs.

##### The MVC model

Model-view-controller (MVC) is an architectural pattern widely used to design user interfaces, which divides an application into three interrelated components. This is done to distinguish internal information representations from how information is communicated to and accepted from the user. The design pattern of MVC decouples these major components to allow efficient reuse of code and also allows parallel development of each of the component. This architecture is mostly used for desktop graphical user interfaces (GUIs) and is popular in designing web applications.

(i) **THE MODEL** - A *Model* is the application's principal central component. It represents the behaviour of the application, independent of the user interface, in terms of the problem domain. It manages the applications data, logic and rules directly. It is responsible for administering the application's logic and data. It receives user inputs and commands via the View component and uses the logic to generate outputs, which is shown again via the View component. Our Model comprises of various sub-models which represent each disease. In our project, we have used four different diseases, each having its own model.

1. **Malaria model** - For designing the malaria disease model, we have taken a dataset comprising of cell images of 50x50 pixels in .png format.

The dataset is then divided into the train set and test set of 22046 and 5512 images respectively. Then Convolutional Neural Network (CNN) has been applied, having 3 convolutional layers and one fully-connected layer. 'relu' and 'softmax' activation functions are applied to the hidden layers and the output layer respectively. While compiling 'categorical\_crossentropy' has been used as the loss function and 'adam' as an optimization function.

2. **Pneumonia model** - For pneumonia model, a dataset comprising of chest x-rays has been taken. The images are of 224 X 224 pixels. We then divided the dataset into the train set and test set and as the input images of both the sets are not balanced, we have used class-weight technique to balance it. To the dataset, CNN was applied, having 10 convolutional layers and one fully-connected layer. For activation purpose, 'relu' and 'sigmoid' functions are used in the hidden and output layer respectively. While compiling 'binary\_crossentropy' has been used as the loss function and 'adam' as an optimization function.

3. **Breast Cancer model** - For this, we have taken a dataset of cell images of micro-anatomy that contains RGB images of 50x50 pixels. To the dataset, we applied a CNN model of 6 convolutional layers and one fully-connected layer. To the output layer and the hidden layers, 'sigmoid' and 'relu' activation functions are applied.

4. **Skin Cancer model** - We have taken a dataset comprising of RGB images of skin samples. The images were of 224x224 pixels. The dataset is then divided into the train set and test set of 2637 and 660 images respectively. We have then applied a CNN model of 4 convolutional layers and one fully-connected layer. For activation, 'softmax' and 'relu' are applied to the output layer and hidden layers respectively. While compiling, 'categorical\_crossentropy' has been used as loss function and 'adam' as an optimization function.

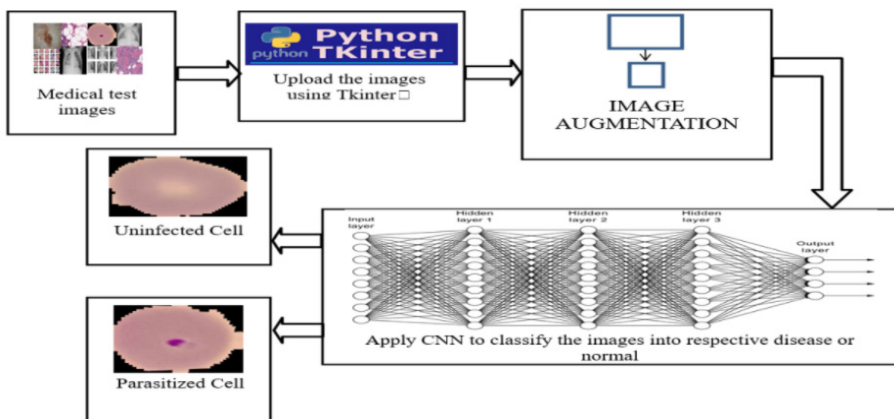


Figure 2: Flowchart of the proposed method.

(ii) **THE VIEW** - A *View* is something available to the user. It reflects the user interface with which the user is communicating while using an application. While the View has buttons, it, itself remains unaware of the fundamental interaction that exists with the back-end. It helps UI / UX people to operate in parallel with the people at the back-end of the user interface.

(iii) **THE CONTROLLER** - A *Controller* is a master that synchronizes the Model along with View. It obtains the user's interaction with the View, transmits them on to the Model that then processes the input information for output production. Through the View, the outputs (results) are then shown to the user.

### How MVC fits in this project ?

There are mainly three files in the project: main.py, detector and upload which contains the GUI, Detector and Controller components, respectively. The GUI component is concerned for the View, the Detector component represents the Model and the Controller brings these together by receiving the communication from the GUI component and carrying them on to the Detector component. With the help of these, we can alter and run the GUI component separately without compromising the functionality of the other components. Similarly, by importing separately from the GUI we can modify the Detector component and use it as a module as well.

## V. IMPLEMENTATION AND RESULTS

### WORKING

On opening "*HDS.exe*" a GUI based menu displays four buttons (for each disease). When the user clicks on the button of a particular disease, function call immediately executes the "*application\_[disease name].py*" file to initialize a new Tkinter window. This new window consists of the '*Upload*' button, the '*Detect*' button along with an *ImageView* and *TextView*. This '*Upload*' button calls upon the *open\_img()* function which reads the path of the image that is to be uploaded using *filedialog.askopenfilename()*, and stores it in the file named '*Upload*' by using the *wrt()* function. The uploaded image is then resized into 325 x 200 pixels automatically before displaying it to the user at the *ImageView* with the help of the '*PIL*' module.

When the user clicks on the '*Detect*' button, the button calls upon the *callback()* function which then retrieves the path of the uploaded image from the '*Upload*' file using *red()* function. The image is then transformed into the target size before loading. The corresponding saved CNN model is then used to classify the image. The result is displayed to the user on the screen at the *TextView*. In this way, the user can classify different images of a particular

disease in the same window or can go back to the menu to detect different diseases.

## SOFTWARE REQUIREMENT

- Python 3.6 or 3.7
- Pip version 19 or 20
- Python modules include tensorflow, numpy, pandas, PIL, tkinter.

## INSTALLATION

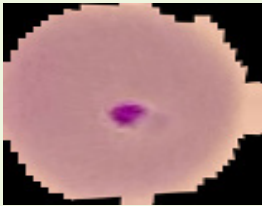

The application is “*exe*” installable and can be installed in the following way-

- Click on the package named “*setup.exe*”, choose the path where you want to install the application and then press the Install button.
- You will see the application package installing and after it is done press Close.
- Go to the selected path and open the “Health Discernment System” folder.
- Open “*requirements.exe*” to check and install the required libraries. This is only a one-time execution. This may take some time if the required libraries are not installed.
- Open the “*HDS.exe*” file to run the application.




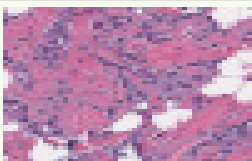


**Note:** Python compiler is the primary requirement.

## TEST CASES AND TEST RESULTS

Table 1. Different test cases and their results.

Test ID	Test Case Title	Test Condition	System Behaviour	Expected Result
T01	Malaria Detection I		Parasitized Cell Detected	Parasitized Cell Detected
T02	Malaria Detection II		Uninfected Cell Detected	Uninfected Cell Detected



T03	Pneumonia Detection I		Normal Detected	Normal Detected
T04	Pneumonia Detection II		Pneumonia Detected	Pneumonia Detected
T05	Breast Cancer Detection I		Benign Tissue Detected	Benign Tissue Detected
T06	Breast Cancer Detection II		Malignant Tissue Detected	Malignant Tissue Detected
T07	Skin Cancer Detection I		Benign Skin Detected	Benign Skin Detected
T08	Skin Cancer Detection II		Malignant Skin Detected	Malignant Skin Detected
T09	Non-image file uploaded	Any video (.mp4, .mkv etc.), audio (.mp3, .wav etc.), and other file formats (.exe, .py, .txt etc.)	Image File Not Uploaded	Image File Not Uploaded
T10	None uploaded	System idle	Image File Not Uploaded	Image File Not Uploaded



## VI. CONCLUSION

In this work, a health discernment system has been proposed for image classification that will work in real-life scenarios. The proposed method is based on an MVC architecture and the model is responsible for the behavioral aspect of the application. Different sub-models pertaining to the four diseases (malaria, pneumonia, breast cancer, and skin cancer) have been designed using a convolutional neural network (CNN). Each of the sub-models has its own specific set of activation and loss function with a single optimization function across all the models. Each of the models has been trained separately and the performance of the proposed model has been validated using a completely separate test set. Among all the four diseases, the model for skin cancer recorded the lowest accuracy at 84.6%. While classification accuracy to detect the presence of parasitized cells of malaria is seen to be highest at 95.21%. Pneumonia and breast cancer models showed performance accuracy of 90.47% and 86.88% respectively. Although the computational time of training each of the models is high, once the trained model is deployed, testing time is minimal. A GUI based web application has been designed with the help of python Tkinter such that real-life input can be given which will be passed on to the trained model for prediction. Therefore, the proposed work is suitable to be used in real-life situations as it is user friendly and cost-effective in nature.

## REFERENCES

1. A. Alias, B. Paulchamy, Detection of breast cancer using artificial neural network, International Journal of Innovative Research in Science 3 (3).
2. M.G. Kanojia, S. Abraham; Breast cancer detection using RBF neural network. In Contemporary computing and informatics (IC3I), 2016 2nd international conference on, IEEE (2016), pp. 363-368
3. A.F. Agarap; On breast cancer detection: an application of machine learning algorithms on the Wisconsin diagnostic dataset, CoRR abs/1711.07831
4. S. Sahan, K. Polat, H. Kodaz, S. Günes; A new hybrid method based on the fuzzy-artificial immune system and K-NN algorithm for breast cancer diagnosis Comput Biol Med, 37 (3) (2007), pp. 415-423
5. F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte; A dataset for breast cancer histopathological image classification. IEEE Trans Biomed Eng, 63 (7) (2016), pp. 1455-1462.
6. R. Pranav, Y. H. Awni, H. Masoumeh, B. Codie, and Y. N. Andrew, "Cardiologist-level arrhythmia detection with convolutional neural networks," 2017.

7. P. Huang, S. Park, R. Yan et al., “Added value of computer-aided CT image features for early lung cancer diagnosis with small pulmonary nodules: a matched case-control study,” *Radiology*, vol. 286, no. 1, pp. 286–295, 2017.
8. P. Lakhani and B. Sundaram, “Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks,” *Radiology*, vol. 284, no. 2, pp. 574–582, 2017.
9. F. D. Demner, M. D. Kohli, M. B. Rosenman et al., “Preparing a collection of radiology examinations for distribution and retrieval,” *Journal of the American Medical Informatics Association*, vol. 23, no. 2, pp. 304–310, 2015.
10. T. I. Mohammad, A. A. Md, T. M. Ahmed, and A. Khalid, “Abnormality detection and localization in chest x-rays using deep convolutional neural networks,” 2017.
11. Nadjm, Behzad, and Ron H. Behrens. “Malaria: An update for physicians.” *Infectious Disease Clinics* 26, no. 2 (2012): 243-259.
12. Krizhevsky, Alex, and Geoff Hinton. “Convolutional deep belief networks on cifar- 10.” Unpublished manuscript 40, no. 7 (2010).
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
14. S. S. Han, M. S. Kim, W. Lim, G. H. Park, I. Park, and S. E. J. J. o. I. D. Chang, “Classification of the Clinical Images for Benign and Malignant Cutaneous Tumors Using a Deep Learning Algorithm,” 2018.
15. U.-O. Dorj, K.-K. Lee, J.-Y. Choi, M. J. M. T. Lee, and Applications, “The skin cancer classification using deep convolutional neural network,” pp. 1-16, 2018.
16. G. Varun, P. Lily, C. Marc et al., “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2017.
17. Liang, Zhaohui, Andrew Powell, Ilker Ersoy, Mahdieh Poostchi, Kamolrat Silamut, Kannappan Palaniappan, Peng Guo et al. “CNN-based image analysis for malaria diagnosis.” In 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 493-496. IEEE, 2016.