

Multi Disease Diagnostic Platform

Sheetal S. Patil¹, Amit Kumar², Ankana Mukherjee³, Soubhagya Ajwani⁴, R.G.Khalkar⁵, Netra S.Patil⁶

^{1,2,3,4,5,6} Department of Computer Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune

E-mails: sspatil@bvucoep.edu.in, soubhagya.ajwani-coep@bvucoep.edu.in, ankana.mukherjee-coep@bvucoep.edu.in,

amit.kumar1-coep@bvucoep.edu.in, rgkhalkar@bvucoep.edu.in, nspatil@bvucoep.edu.in

Abstract— The future of AI in health care could include tasks that range from simple to complex. Everything from answering the phone to case history review, population health trending and analytics, therapeutic medicine, analyzing radiology images, predicting clinical diagnoses and treatment roadmaps, and even talking to patients.

The future of computer science in health care presents:

- A healthcare-oriented overview of AI (AI), tongue processing (NLP), and machine learning (ML)
- Current and future applications in health care and also the impact on patients, clinicians, and therefore the pharmaceutical industry
- A observe how the longer term of AI in health care might unfold as these technologies impact the practice of medication and health care over the subsequent decade

Keywords— **Disease Diagnosis, Neural Network, CNN, Regression, Deep Learning, Machine Learning**

1.Introduction

Technology innovations have produced radical medical advancements. Despite monumental improvements, society is simply experiencing the start phase of this process. While today could be a time of growth, it's also a time of growing pains. Duly, the medical field currently faces two prominent challenges: service integration, and repair quality.

For example: In today's world especially in India, there's an enormous shortage of proper diagnostic facilities. Without early detection of diseases, people don't seem to be ready to get proper medical facilities in time which ends up in death in many cases.

While innovations promise to alter healthcare service delivery forever, it'll take your time before the advancements reach their full potential.

1.1 Problem Definition

In this project, we aim to make a multi-disease diagnostic platform using various Machine Learning and Deep-learning models that accurately predict the probability of the occurrence of a disease. We aim to make a web-based application that will integrate these models. The users are going to be ready to interact with the online app, provide relevant information and also the web app will predict the end result supported by the knowledge provided by the user.

1.2 Motivation

In today's world especially in India, there's an unlimited shortage of proper diagnostic facilities. Among the foremost common health issue in India, diseases like COVID-19, dengue and malaria are at the highest of the list. In step with the WHO, the best number of dengue cases occurs within the monsoons in India.

Without early detection of diseases, people don't seem to be ready to get proper medical facilities in time which ends up in death in many cases. A better way of coping up with these problems is to harness the facility of computing.

2. Literature Survey

One of AI's biggest potential benefits is to help people stay healthy so they don't need a doctor, or at least not as often. The use of AI and the Internet of Medical Things (IoMT) in consumer health applications is already helping people. AI is already being used to detect diseases, such as cancer, more accurately and in their early stages. According to the American Cancer Society, a high proportion of mammograms yield false results, leading to 1 in 2 healthy women being told they have cancer. The use of AI is enabling review and translation of mammograms 30 times faster with 99% accuracy, reducing the need for unnecessary biopsies.

2.1 Review of Existing Models and Approaches:

DeepMind, a UK-based company, is teaming up with Google Health to tackle complex problems in healthcare. Alongside teams at Google, DeepMind will work to build products that support care teams and improve patient outcomes. DeepMind and Google Health will aim to develop tools that could potentially help prevent sepsis and acute kidney injury. By joining forces with Google Health, DeepMind will add to its many projects aimed at improving patient care. The company recently developed a clinical decision support tool that can accurately identify more than 50 eye diseases. Currently, eye care professionals use optical coherence tomography (OCT) scans to help diagnose eye conditions. These 3D images provide a detailed map of the back of the eye, but they are hard to read and need expert analysis to interpret. The company also partnered with the VA to use machine learning to develop predictive analytics tools. The project

Multi Disease Diagnostic Platform

aimed to identify and prevent patient deterioration and improve outcomes. With this new collaboration, both DeepMind and Google Health will continue to improve patient care.

3. Objective

Medical and technology together have produced such advancement in our society yet in today's time, there is vast shortage of proper service integration and service quality. India, for instance, dilemma for a proper diagnostic platform which is available and affordable to all people.

Early detection of any disease can increase the life expenditure of many people but because of its storage, there are millions of deaths across the nation every year.

The goal of the project is to introduce a multi-disease diagnostic platform that provides affordable diagnosis. The platform uses various Machine Learning and Deep learning models that accurately predicts the probability of a disease occurrence. The user at the comfort of their home, can interact with the application, provide all the relevant information and get the result of their probability of the respective disease occurrence.

4. Approach

Through Artificial Intelligence, we are able to simulate human intelligence in a machine. It can perform human intelligent task when provided with adequate data for its decision making.

Artificial Intelligent systems work through algorithms. Techniques like machine learning and deep learning are used. Machine learning is a subbranch of artificial intelligence that perform tasks without being directly programmed.

It takes the input and output for a problem and tries to build a logic for its output. To accomplish this efficiently and effectively, machine learning uses statistics and computer science.

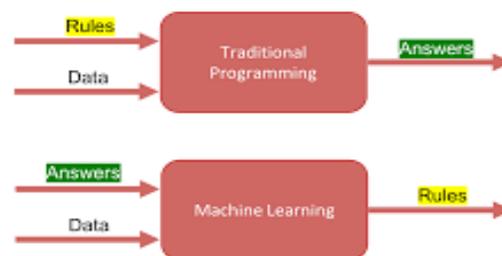


Figure 1 Traditional Programming V/S Machine Learning

Deep learning is the subset of machine learning which uses multi-layered structure of algorithms called neural networks. Just like the brain, it processes data and create patterns that is used for decision making.

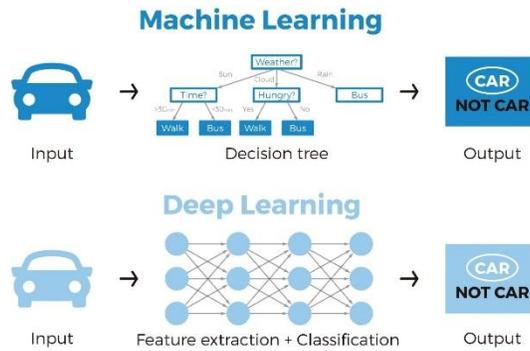


Figure 2 Machine Learning V/S Deep Learning

One of the deep learning algorithm that takes image as an input, assigns importance (learnable weights and biases) and differentiates the image, is Convolutional Neural Network. This is a classification algorithm that works on discrete class label data or categorical data.

Computers represents an image as an array of pixels. Some of the common terminologies of CNNs are:

- Convolution – the most building block of CNN is that the convolution layer which may be a mathematical process to merge two sets of data. The convolution is applied on the input file employing a convolution filter to supply a feature map. The first layer to extract features from an input image is convolution.
- Striding – the amount of shifts over the input image is named stride. Example- When stride is 1 then we move the filters to 1 pixel at a time.
- Non-Linearity – The Activation function generally used is ReLU stands for Rectified Linear Unit for a non-linear operation. The output is:

$$\{F(x) = \max(O, x)\}$$

- Pooling Layer – After a convolution operation we usually perform pooling to scale back the dimensionality. This enables us to scale back the amount of parameters, which successively shortens the training time and reduces chances of over fitting. Pooling layers down sample each feature map independently, reducing the peak and width, keeping the depth intact. Contrary to the convolution operation, pooling has no parameters. It slides a window over its input, and easily takes the utmost value within the window. The largest element from the feature map is chosen by using this operation.

Multi Disease Diagnostic Platform

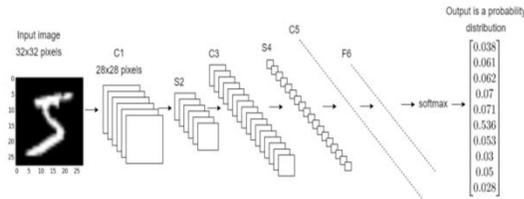


Figure 3 CNN MODEL ARCHITECTURE

Another supervised classification algorithm used is logistic regression. Logistic regression assumes that the data follows the sigmoid function.

It becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect and is dependent on the classification problem itself.

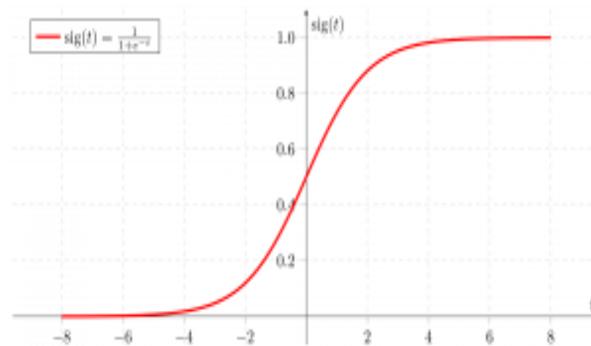


Figure 4 Sigmoid Equation Paper

The decided value of the threshold is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but it happens rarely. The following are the ways threshold is decided:

- **Low Precision/High Recall:** In places where we want to reduce the number of false negatives without necessarily reducing the number false positives, we choose a decision value which has a low value of Precision or high value of Recall.

For example, in a cancer diagnosis application, we do not want any affected patient to be classified as not affected without giving much heed to if the patient is being wrongfully diagnosed with cancer. This is because, the absence of cancer can be detected by further medical diseases, but the presence of the disease cannot be detected in an already rejected candidate. (1)

- **High Precision/Low Recall:** In places where we want to reduce the number of false positives without necessarily reducing the number false negatives, we choose a decision value which has a high value of Precision or low value of Recall.

For example, if we are classifying customers whether they will react positively or negatively to a personalised advertisement, we want to be absolutely sure that the customer will react positively to the advertisement (2) because otherwise, a negative reaction can cause a loss potential sales from the customer. (1)

Based on the number of categories, Logistic regression can be classified as:

binomial: target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc. (1)

multinomial: target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”. (1)

ordinal: it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3. (1)

SMOTE stands for Synthetic Minority Oversampling Technique. This is a statistical technique for increasing the number of cases in your dataset in a balanced way. The module works by generating new instances from existing minority cases that you supply as input. This implementation of SMOTE does not change the number of majority cases. (2)

The new instances are not just copies of existing minority cases; instead, the algorithm takes samples of the feature space for each target class and its nearest neighbours, and generates new examples that combine features of the target case with features of its neighbours (3). This approach increases the features available to each class and makes the samples more general. (2)

SMOTE takes the entire dataset as an input, but it increases the percentage of only the minority cases. For example, suppose you have an imbalanced dataset where just 1% of the cases have the target value A (the minority class), and 99% of the cases have the value B. To increase the percentage of minority cases to twice the previous percentage, you would enter 200 for SMOTE percentage in the module's properties. (2)

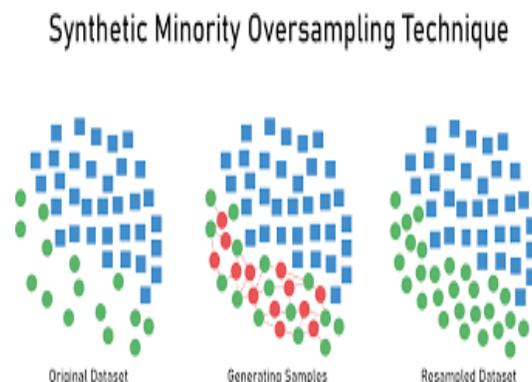


Figure 5SMOTE

Multi Disease Diagnostic Platform

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. (3)

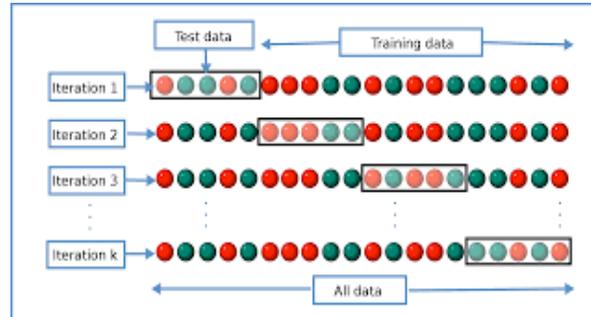


Figure 6 Cross Validation

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. (3)

The general procedure is as follows:

- Shuffle the dataset randomly.
- Split the dataset into k groups
- For each unique group:
 - Take the group as a hold out or test data set
 - Take the remaining groups as a training data set
 - Fit a model on the training set and evaluate it on the test set
 - Retain the evaluation score and discard the model
- Summarize the skill of the model using the sample of model evaluation scores (3)

5. System Design

The project design is divided into two parts:

5.1 USER INTERFACE:

Using flask framework, the web pages and the disease models are integrated.

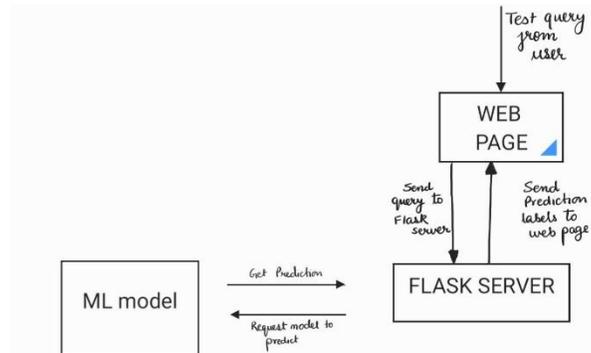


Figure 7 ML WEB APP ARCHIRECTURE

5.2 ML/Deep Learning implementation

The models are either implemented through machine learning approach or deep learning.

The models are trained using different datasets, which have to go through the following process:-

Multi Disease Diagnostic Platform

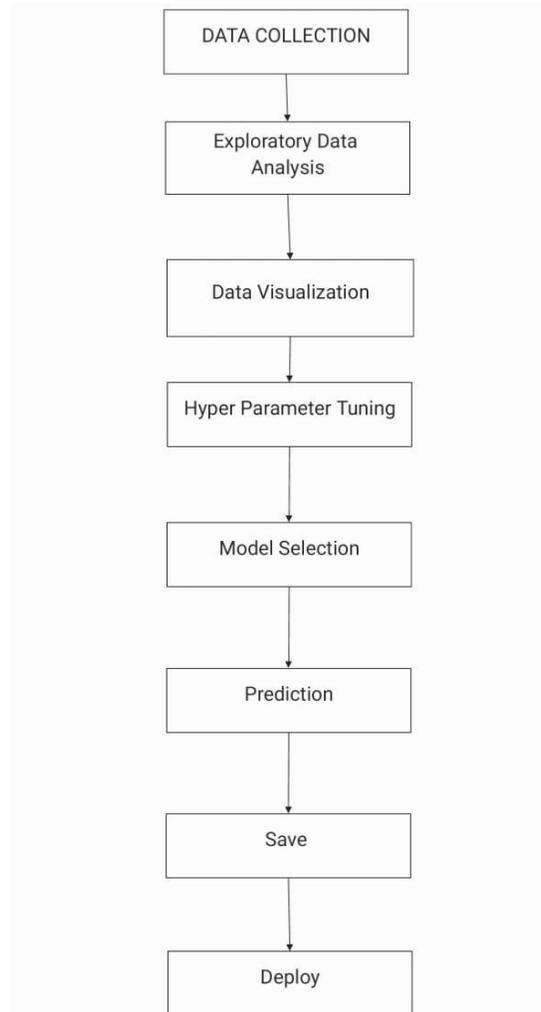


Figure 8 ML Model System Architecture

The project consists of different disease models which use either machine learning or deep learning approaches. We will classify them based on their dataset type. We will use two types of datasets which will consist of

- **csv files**

For datasets of type csv files , We have Planned to use Machine Learning Algorithms. For this we will first start with Exploratory Data Analysis using Numpy and Pandas Library. It is one of the first approaches to our Data Modelling Process. It helps us to analyze our data sets and summarize their main characteristics. Then another step that we will take is Data Visualization. Data visualization helps to tell stories by curating data into a form easier to understand, highlighting the trends and outliers. We will do data visualization through Matplotlib ,seaborn and plotly and also look forward to using some auto data visualization library such as lux, etc for better understanding of our data. Then We will do HyperParameter Tuning. Sometimes, we might get confused with creating the model and

selecting the model. When we create the model, we have to select the appropriate algorithm, sometimes this step(where we select the algorithm) can also be referred to as selecting the model. So, hyper parameter helps in selecting the best model, therefore we will be using hyper parameter tuning. Then in the next case we will evaluate performance based on many Machine Learning Algorithms such as KNN, Naïve Bayes , SVM, Decision Tree , Random Forest, Light GBM, XGBoost,e.t.c. and select the best model based and do prediction.

- **image files**

Now for images datasets, We will use Deep Learning Algorithms and techniques. First we will import our dataset then we will implement Exploratory Data Analysis and Data Visualization to view images contained in our dataset. Then , We will do Data Augmentation if there is need. It is a strategy that enables practitioners to significantly increase the diversity of data available for training models, without collecting new data. Data augmentation techniques such as cropping, padding, and horizontal flipping are commonly used to train large neural networks. Then as we will be working on an image dataset then we have to do image conversion, resize and Normalization. Then we will create a Sequential Model where we will use various activation Functions according to the need in different layers. Then , We will Compile our model using Optimizer and then we will go for model training. We will then check the accuracy of our model and do predictions.

After this, we will Save our Model and go for Flask Deployment

6. Implementation

Various diseases implemented in our project are:

6.1 Cancer

So in Cancer classification, We have first imported basic python libraries such as NumPy and Pandas. Then we have imported our dataset. Then we performed Exploratory Data Analysis removing some columns such as unnamed: 32, id. We also created a single data frame named 'a' that contain diagnosis. Then we have concat 'data' and 'a' dataframe to make a single dataframe cancer. Then we have renamed this cancer. Then we have removed columns named diagnosis and B. Then, we have renamed "M" to Malignant/Benign. Then we divide our dataset into X and Y for training and testing. In X, we have stored all the independent variables, and in Y, we have stored the dependent variables. Then we have used the train test and split it with random 30% test size data. Then we get a total of 398 X-train and Y-train and 198 X-test and Y-test. Then we

Multi Disease Diagnostic Platform

have logistic regression for fitting our data for classification then we have saved our model.

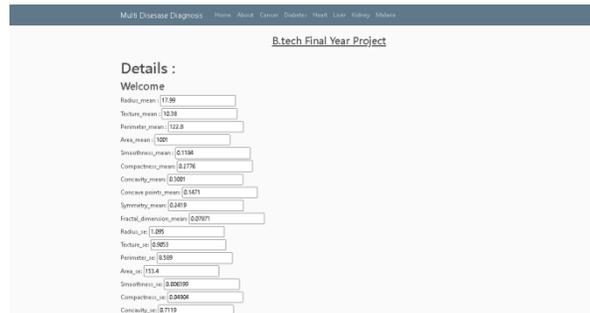


Figure 9 Cancer Prediction

6.2 Kidney Health

First of all, basic Library packages such as NumPy and Pandas are imported. Then we have imported our dataset 'kidney_disease.csv'. Then we go for exploratory data analysis. In the exploratory data analysis first, we have done data encoding of 'htn', 'dm', 'cad', 'pe', 'ane' columns. Here we convert 'Yes':1 and 'no':0. We do another hot encoding of columns 'rbc' and 'pc' columns 'abnormal': 1, 'normal': 0. Then we also replace the values of 'pcc' and 'ba' to 'present':1 and 'notpresent':0 and we also do hot encoding of 'dm' values '\tno': 0, '\tyes': 1, 'yes': 1, 'np.nan'. Then we have at last do the final hot encoding of classification column 'ckd': 1.0, 'ckd\t': 1, 'notckd': 0 and 'no': 0. Then we rename the column Classification into class. Then we have dropped columns like "su", "rbc", "rc", "wc", "plot", "sod", "be", "pa", "cad", "ane". For missing values, we have used the forward fill() method. So now after completing all this now we are left with 13 columns which are 'age', 'bp', 'al', 'pcc', 'bgr', 'bu', 'sc', 'hemo', 'pcv', 'htn', 'dm', 'appet', 'class'. Then we divide the dataset into dependent and independent Variables for training and testing in source and target. We have chosen 90% of our dataset for training and 10% of random values for training and testing. Then we have used SMOTE in X-train and Y-train for handling imbalanced data. Then we use the Logistic Regression algorithm for classification. Then we have used cross-validation with cv=10 to estimate our model.

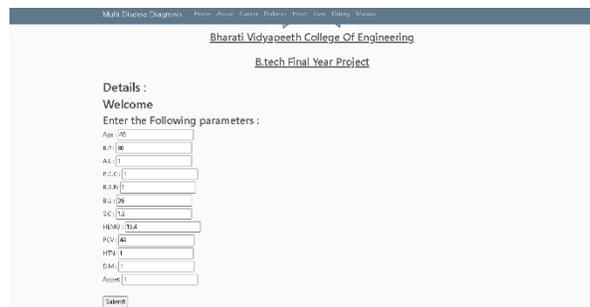


Figure 10 Kidney Prediction

6.3 LiverClassification

First of all, we have imported libraries such as NumPy and Pandas. Then we have loaded our dataset 'Indian_liver_patient.csv', downloaded from Kaggle. After loading the dataset, we use the 'fillna()' method to fill the missing values present in the dataset. Then we have done data encoding of the Gender column. After that, we have used a standard scaler for scaling the dataset. Then we divide the dataset into dependent and independent variables for training and testing in source and target variables. We have chosen 90% of our dataset for training and 10% of random values for training and testing. Then we have used SMOTE in X-train and Y-train for handling imbalanced data. Then we use the Logistic Regression algorithm for classification. Then we have used cross-validation with cv=10 to estimate our model on unseen data. Then we have saved our model.



Figure 11 Liver Classification

6.4 Heart Disease

In Heart Disease Prediction, we first basic library packages NumPy and Pandas. Then we imported our dataset, 'Cleveland.csv'. Then we checked missing values using "dataFrame.isnull().sum()". Then we found that our dataset contains some missing values. Then we replaced the missing values column with their column meaning. Then we have to do Data Encoding of the 'target' and 'thal' columns. Then we did some data visualization using Matplotlib and Seaborn.

Then we also visualized the distribution of age vs Sex with the target class.

From this, we analysed the amount of heart disease patients from different male and female categories. Then we have divided our DataFrame into dependent and Independent Variables. Then we have done a train test split with test size = 20% random data. Then we have used standard scaling to scale the data. Then we have used several Machine Learning algorithms such as SVM, Naïve Bayes, Logistic Regression, Decision Trees, Random Forest, LightGBM, XgBoost.

According to this, we analyzed that Random Forest gives good training and testing accuracy. So, we will save the Random Forest model.

Multi Disease Diagnostic Platform



Figure 12 Heart Disease Prediction

6.5 Diabetes

First of all, we have imported basic python libraries such as NumPy and Pandas. Then we imported our dataset and calculated all the null values. Then we find the correlation among the column using the corr() method.

Then we visualized the outcome class using Seaborn. Our dataset contains approximately 500 Healthy and approx. 250 Diabetic patients.

Then we proceed further to Data Cleaning. As we all know, the value of Glucose, Blood Pressure, Skin Thickness, Insulin, BMI cannot be zero, so we change the value where 0 is present to 'Nan'. Then, we replace the 'Nan' value with their column's mean value. Then we normalize our data with a min-max scaler. Then we go for allocating Predictor and Target Variables. Then we train-test and split our data.

Then we applied various ml algorithms to it, such as Logistic Regression, KNN, SVM, etc. Based on this, we selected KNN and we save the KNN model.



Figure 13 Diabetes Prediction

6.6 Malaria

First of all, we have imported all the necessary python libraries. Then we will upload our malaria dataset in the form of images containing Parasitized and Uninfected. Then we give the directories for Parasitized as PARA_DIR and UNIF_DIR for uninfected. Then we will plot some random Parasitized and Uninfected.

After then we will convert all the Parasitized and Uninfected images to RGB and resize them into 50X50. Then we converted it into a NumPy array. Then we randomly shuffled them. Then we normalized our data by dividing it by 255. Then we did one-hot-encoding as the classifier had multiple classes. Then we created our CNN Model.

CNN Architecture

Here, we first specified that our model is sequential. After initialising it, we added

- 2 X convolution layer of 32 channel of 2X2 kernel and same padding.
- 1 X maxpool layer of 2 X 2 pool size.
- 1 X Dropout 0.2
- 2 X convolution layer of 32 channel of 2X2 kernel and same padding.
- 1 X maxpool layer of 2 X 2 pool size.
- 1 X Dropout 0.2
- 2 X convolution layer of 32 channel of 2X2 kernel and same padding.
- 1 X maxpool layer of 2 X 2 pool size.
- 1 X Dropout 0.2

We also added Relu (Rectified Linear) to each layer for not passing negative values to the next layer.

After creating above all the convolution layer, data is passed to the dense layer where we flatten the vector which comes out for convolutions. We added the Dense Softmax activation layer in the end to predict 2 classes, Parasitized and Uninfected. Softmax activation layer will output the value between 0 and 1 based on the confidence of the model that which class the images belongs i.e. Parasitized and Uninfected.

After Preparing the model we compiled the model with loss = "binary_crossentropy", optimizer="adam" and metrics = "accuracy". After compiling the model we imported EarlyStopping. It helps us to stop the model training early by monitoring val_acc. Then we used model.fit() to pass data into the model. We have passed x_train and x_test to model.fit() and we train our model for 20 epochs. But the model stopped training at the 13th epoch due to EarlyStopping. Then we saved our model generating an accuracy of 95.86%.

Multi Disease Diagnostic Platform

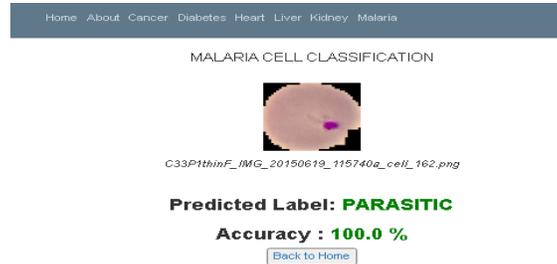


Figure 14 Parasitic Malaria Classification

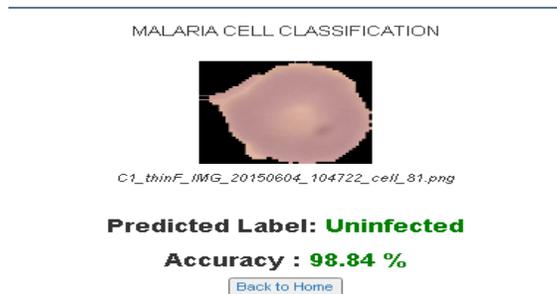


Figure 15 Uninfected Malaria Classification

Conclusion

Our Project Multi Disease Diagnose System will be a research-based flask web application which will be capable of diagnosing multi diseases such as Malaria, Diabetes, Chronic Kidney, Heart Disease, etc. Our platform will use powerful Machine Learning and Deep Learning Algorithms to predict the infection of various diseases. Our models rely on the analysis and interpretation of huge amounts of data in order to help both doctors and people to make better diagnosis and also, saving people's fortune. We will check and take care of everything because we know our project is related to medical healthcare. We will use the Flask web framework for our model Deployment. With the help of Flask and other web development technologies we will create a beautiful, simple and easy to interact GUI where any user has to just fill in the required information and That's it! Our platform will predict the chances of being infected by the disease.

References

1. Understanding Logistic Regression. Retrieved from <https://www.geeksforgeeks.org/understanding-logistic-regression/>
2. SMOTE. Retrieved from Microsoft: <https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/smote>
3. A Gentle Introduction to k-fold Cross Validation. Retrieved from <https://machinelearningmastery.com/k-fold-cross-validation/#:~:text=Cross%2Dvalidation%20is%20primarily%20used,the%20training%20of%20the%20model.>
4. R. Gupta, N. Koli, N. Mahor and N. Tejashri, "Performance Analysis of Machine Learning Classifier

- for Predicting Chronic Kidney Disease," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154147.
5. M. Bozorgi, K. Taghva and A. Singh, "Cancer survivability with logistic regression," 2017 Computing Conference, 2017, pp. 416-420, doi: 10.1109/SAI.2017.8252133.
 6. S. H. Adil, M. Ebrahim, K. Raza, S. S. Azhar Ali and M. Ahmed Hashmani, "Liver Patient Classification using Logistic Regression," 2018 4th International Conference on Computer and Information Sciences (ICCOINS), 2018, pp. 1-5, doi: 10.1109/ICCOINS.2018.8510581.
 7. J. Ma, "Machine Learning in Predicting Diabetes in the Early Stage," 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 167-172, doi: 10.1109/MLBDBI51377.2020.00037.
 8. P. Sonar and K. Jaya Malini, "Diabetes Prediction Using Different Machine Learning Approaches," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 367-371, doi: 10.1109/ICCMC.2019.8819841.
 9. R. Katarya and P. Srinivas, "Predicting Heart Disease at Early Stages using Machine Learning: A Survey," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 302-305, doi: 10.1109/ICESC48915.2020.9155586.
 10. O. Iradukunda, H. Che, J. Uwineza, J. Y. Bayingana, M. S. Bin-Imam and I. Niyonzima, "Malaria Disease Prediction Based on Machine Learning," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1-7, doi: 10.1109/ICSIDP47821.2019.9173011.
 11. S. S. Prakash, B. C. Koor and K. Visakha, "Convolutional Neural Network Based Malaria Parasite Infection Detection Using Thin Microscopic Blood Smear Samples," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 308-313, doi: 10.1109/ICIRCA48905.2020.9182944.
 12. G. Kalipe, V. Gautham and R. K. Behera, "Predicting Malarial Outbreak using Machine Learning and Deep Learning Approach: A Review and Analysis," 2018 International Conference on Information Technology (ICIT), 2018, pp. 33-38, doi: 10.1109/ICIT.2018.00019.
 13. K. Shailaja, B. Seetharamulu and M. A. Jabbar, "Machine Learning in Healthcare: A Review," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 910-914, doi: 10.1109/ICECA.2018.8474918.
 14. N. G. Maity and S. Das, "Machine learning for improved diagnosis and prognosis in healthcare," 2017 IEEE Aerospace Conference, 2017, pp. 1-9, doi: 10.1109/AERO.2017.7943950.
 15. D. Ravi et al., "Deep Learning for Health Informatics," in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 4-21, Jan. 2017, doi: 10.1109/JBHI.2016.2636665.
 16. M. Irfan and I. A. Hameed, "Deep learning based classification for healthcare data analysis system," 2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC), 2017, pp. 1-6, doi: 10.1109/BESC.2017.8256396.
 17. T. Le Nguyen and T. T. H. Do, "Artificial Intelligence in Healthcare: A New Technology Benefit for Both Patients and Doctors," 2019 Portland International Conference on Management of Engineering and Technology (PICMET), 2019, pp. 1-15, doi: 10.23919/PICMET.2019.8893884.
 18. <https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/>
 19. © 2019 JETIR June 2019, Volume 6, Issue 6 www.jetir.org (ISSN-2349-5162)

Multi Disease Diagnostic Platform

20. R. S. Suryawanshi, A. Kadam, and D. R. Anekar, "Software defect prediction: A survey with machine learning approach," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 330–335, 2020.
21. A. Kurhade, J. Naveenkumar, and A. K. Kadam, "An experimental on top-k high utility itemset mining by efficient algorithm Tkowithtku," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8 Special Issue 3, pp. 519–522, 2019.
22. Dr.S. D. Joshi, Dr. A. K. Kadam, Pritee Hulule, "A Survey Novel Approach for Efficient Selection of Test Case Prioritization Techniques," vol. 3085, no. 12, pp. 999–1001, 2018.
23. Dr. A. K. Kadam, Amruta Magdum, prof. Dr. S. D Joshi, "A Survey on Test Case Prioritization with Rate of Fault Detection," *Int. J. Res. Electron. Comput. Eng.*, vol. 6, no. 4, 2018.
24. A. N. Patil, A. Kadam, S. B. Wakurdekar, and S. D. Joshi, "Hybrid Approach of Code Analysis and Efforts Calculation for Software Reliability Growth Measurement and Cost Estimation," *Iioab J.*, vol. 9, no. 2, SI, pp. 116–120, 2018.
25. A. Magdum, S. D. Joshi, A. K. Kadam, and A. Sarda, "Test case ranking with rate of fault finding," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 8 Special Issue 3, pp. 462–464, 2019.
26. A. K. Kadam, S. D. Joshi, D. Bhattacharyya, and H.-J. Kim, "Software Superiority Achievement through Functional Point and Test Point Analysis," *Int. J. Softw. Eng. Its Appl.*, vol. 10, no. 11, pp. 181–192, 2016.
27. V. E. Pawar, A. K. Kadam, and S. D. Joshi, "Analysis of software reliability using testing time and testing coverage," *Int. J. Adv. Res. Comp. Sci. Manag. Stud.*, vol. 3, no. 5, pp. 143–148, 2015