Multi disease Prediction using CNN and Logistic Regression

Shankar K
Assistant Professor, Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, India, shankar.k@eec.srmrmp.edu.in

Arunachalam S
Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, India, sararunachalam03112001@gmail.com

Dhanush G
Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, India, dhanushg222@gmail.com

Elangovan P
Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai, India, elangovan1918@gmail.com

Abstract

Preventable health problems can be avoided with the use of early detection methods, which are essential to improving the quality of treatment available to the public. Disease prediction and treatment response prediction are two areas where deep learning has shown to be particularly useful. Diseases including cancer, diabetes, and cardiovascular disorders have all seen increases in the Input, which is received through numerical data because to the rise of deep learning techniques and their impact on healthcare generally. An instantaneous, real-time output of a disease's likeliness, its likely causes, and its potential effects is obtained. The purpose of this study is to use real-time data, in the form of electronic health records, to forecast cancer and diabetes (EHR). The goal here is to use data from the CT scan and other chest x-rays to make an instantaneous diagnosis of pnuemonia. The top-tier application of machine learning and Deep Learning techniques. The goal is to develop an application-based server using the flask web framework.

Keywords: *CNN, Electronic health records, cancer, diabetes, Pneumonia, Chest X-rays, Flask web application*

I. INTRODUCTION

In Early disease prediction helps public health by allowing patients to adopt preventative measures. Researchers are becoming more interested in predictive modelling for illness progression and analysis as EHRs become more available. EHR data uses time to sequence patient visits as high-dimensional clinical events. Data mining for predictions is prevalent due to the noisy, irregular, and diverse nature of EHRs. To estimate individual illness risk, a global model is constructed using all available training data. With 77 million people already diagnosed with diabetes and a projected rise to 100 million by 2025, India is the second most populous country in the world

in terms of the diabetes epidemic, behind only China. Forecasts indicate that by 2025, there would be 29.8 million Indians battling the disease, up from 26.7 million in 2021. Pneumonia is the leading infectious killer of children across the world. A total of 740 180 children under the age of 5 lost their lives in 2019 due to pneumonia.

II. RELATED WORKS

There is great promise for employing deep learning algorithms in medical diagnosis [1. Whilst additional research into the practical use of such algorithms is required, the accuracy of deep learning algorithms is on par with that of healthcare experts. An overstated claim from a poorly planned or badly reported study may damage the credibility and road to effect of such diagnostic algorithms, which is the more relevant conclusion around methodology and reporting. In this summary, we have emphasised some of the more important design and reporting concerns that researchers should think about.

[2] Using both structured and unstructured data from the hospital, the authors present a novel convolutional neural network (CNN)-based multimodal illness risk prediction method. None of the previous research in the field of medical big data analytics seems to have combined the two kinds of data. Our proposed approach achieves a prediction accuracy of 94.8% and a convergence speed that is quicker than the CNN-based unimodal illness risk prediction algorithm, among other benchmarks.

[3] Machine-based diagnosis of disease. Decisions may be made more quickly and with less bias using machine learning methods like support vector machine (SVM), k-nearest neighbour (KNN), naive bayes (NB), and decision tree. Python, which may be used for actual implementation, is explored as a realistic means of implementing these algorithms. The likes of cancer, diabetes, epilepsy, heart attack, and other major disorders are all diagnosed with the use of these algorithms.

[4] From image analysis to natural language processing, deep neural networks are increasingly widely used in both academia and business as cutting-edge machine learning models. A lot of progress is being made, but the potential for these innovations to improve medical imaging, medical data analysis, medical diagnostics, and healthcare as a whole is enormous.

[5] Heart disease prognostication aids have been proposed, such as the Enhanced Deep learning aided Convolutional Neural Network (EDCNN). The EDCNN model incorporates regularisation learning strategies into a more in-depth design than that of the standard multi-layer perceptron. As a decision support system, EDCNN runs on the IoMT platform, which stores data about cardiac patients and makes it accessible from anywhere in the world through the cloud, allowing clinicians to make accurate diagnoses no matter where they are located.

[6] The ML algorithms K-Nearest Neighbor (KNN) and Convolutional neural network (CNN)are used to accurately forecast diseases. It's important to collect data on disease symptoms in order to forecast the spread of illness. It is important to take a person's lifestyle and medical history into account when making a general illness prognosis. CNN performs better than the KNN algorithm in predicting prevalent diseases by 8.5 percentage points. Both the time and storage needed are larger for KNN than for CNN.

[7] The core components of every system are the tools and algorithms used to operate it. In deep learning, software is used to transform raw data into useful insights. When it comes to

deep learning models, you may find a lot of distinctions between supervised and unsupervised approaches. Given the significance of these algorithms, it is imperative that medical instruments, such as the microscope, phonendoscope, and EKG, be developed gradually based on these models in order to accommodate the limitations of physicians' perceptual capabilities. The usage of deep learning in the analysis and prediction of data from big datasets is becoming increasingly important.

[8] Many remote and non-invasive imaging technologies have been outlined, each of which can shed light on the diagnostic value of various symptom kinds in relation to certain medical diseases. In this study, we used computer vision techniques to automatically recognise symptoms of over 30 different diseases to make preliminary diagnoses.

[9] To combat this, we offer a unique ML00 based health CPS framework that can efficiently analyse data from wearable IoT sensors to forecast the likelihood of developing noncommunicable diseases like diabetes at an early stage. An authentic diabetes dataset was used for the experiment's training phase, whereas sensor data created in a lab were used for testing.

[10] Novel deep learning algorithms and methods are discussed. The clinical judgements of future radiologists can be aided by DLA. DLA can automate the job of radiologists and help even the most novice among them make decisions. DLA's goal is to help doctors by automatically detecting and categorising lesions so that a more accurate diagnosis may be made. Doctors can benefit from DLA since it speeds up the interpretation of medical images and reduces the likelihood of making mistakes.

TABLE I. COMPARISON OF LITERATURE SURVEY

Author	Methodology	Features	Challenges
Liva Faes; Xiaoxuan Liu; Pearse A Kean, The Lancet Digital Health, October 2020	"A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis	Critically appraise the current state of diagnostic performance by deep learning algorithms for medical imaging.	Limited amount of dataset.
Pushpa Singh; Narendra Singh; Krishna Kant Singh; Akansha Singh, Machine Learning and the Internet of Medical Things in Healthcare, 2021	Diagnosing of disease using machine learning	Helps to verify this data and foretell that everything from sickness outbreaks to severe infectious diseases.	Dependency on real data leads to data scarcity. .
Yuanyuan Pan; Minghuan Fu; Biao Cheng; Xuefei Tao; Jing Guo, IEEE Access ,May 2020	Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on th	The predictions are very fast.	It is Computationally expensive

M. Chen, Y. Hao, K. Hwang, L. Wang and L. Wang, IEEE Access, 2020	Disease prediction by machine learning over big data from healthcare communities	Prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed which is faster than that of the CNN-based unimodal disease risk prediction (CNN-UDRP) algorithm.	If the images inovlves some degree of changes or angle changes, then CNNs find it difficult in classifying the image.
Alexander Selvikvag; Lundervold; Arvid Lundervold; Katarzyna Węgrzyn-Wolska, Zeitschrift für Medizinische Physik, May 2020	An overview of deep learning in medical imaging focusing on MRI	Results in close to state-of-the-art performance on 2D object detection.	3D CNN's are a comparatively new concept and yet to be like 2D's.

III. PROPOSED PLAN

This is a fully functional application that uses ML and Deep Learning Algorithms for disease prediction across the board, including cancer, diabetes, and pnuemonia. Early illness prediction, patient treatment, and community services all gain from the suggested app's meticulous analysis. The suggested prediction model achieves this goal by systematically using techniques including data cleansing, feature extraction, and classification. To ensure the neural network only receives useful information, feature selection is used to filter out extraneous data. The suggested application's simplicity and efficient performance are two of its strongest selling points.

Module 1: Data Acquisition And Pre-Processing

a. Data Gathering

We have gathered the data for our project for the following diseases in the CSV and Image format:

Cancer : CSV dataset

Diabetes: CSV dataset

Pnuemonia:IMAGE dataset

b. Data Preprocessing

In data mining, preprocessing is a process0 that converts. raw data into a more usable and efficient format. Data preprocessing converts unprocessed0. data into a more manageable format for later use in data mining, machine learning, and other data science processes. Findings may be trusted when the methods are implemented early in the machine learning and AI development pipeline. We have focused on the following data preparation methods notwithstanding the wide availability of other options.

For dataset in CSV file Format :

Removing Unwanted Columns:

Our dataset contains redundant data. We consider it a good idea to eliminate these features from the dataset because their presence has no impact on the target. To ensure that we only work with relevant data, this method of deleting unnecessary features and maintaining only the required features in the dataset is quite

helpful. Therefore, we drop a few columns from our dataset using the drop command. Python's "df.drop" function can be used to remove a single column or a number of columns from a pandas data frame. We further define the title of the column, index, axis and labels.

Identifying Null Values

We have a Kaggle dataset with some missing data that we need to fill up. It's possible that this is because no such information exists or because it was not obtained effectively. None or NaN is used to indicate these data points in the dataset. Whatever the cause, it slows down and messes up our computations. So, we identify the missing information and supply it with relevant pieces.

When working with missing or null data, Pandas treat None and NaN in the same way. Pandas DataFrame has a number of tools to assist in locating, removing, and replacing null values, making it much simpler to adhere to this practise. To identify null values, we utilise the isnull() and notnull() methods built into Pandas DataFrame (). Both help establish if a value is NaN or not. These methods may also be applied to Pandas Series in order to locate missing values in a dataset.

Removing Null Values

After locating the blanks, the corresponding rows and columns are removed. This dataset has null values eliminated using the dropna() method. Columns and rows containing null values are purged by this operation. If we don't want to get rid of the column of null values altogether, we may use the fillna() method to iteratively replace them with anything we provide.

Splitting The Dataset

ML algorithms suitable for prediction-based algorithms/applications can have their performance predicted and evaluated with the train-test split. Our own machine learning model's output may be compared to that of an automated system using this straightforward way. The Test set must consist of at least 70% simulated data in order to be valid, whereas the Training set must consist of at least 30% real data. It is necessary to split a dataset into "train" and "test" sets so that we may evaluate the accuracy of our machine learning model. The model is adjusted using the train set, and its components are recognised. The second group is used only for making projections; it is referred to as the test data set. We utilise this method to split our dataset into a train and test set. This includes bringing in the pandas and sklearn packages. If you're looking for a powerful and flexible machine learning package, go no further than Sklearn in Python. Scikit-splitter learn's functiontrain test split is part of the library's model selection module (). A CSV file is read in using the read csv() function. We have assigned df to hold the data frame. Next we split the data in half, with 70% going towards training and 30% going towards testing. For data to be arbitrarily divided between the two sets, we additionally set random state=0.

For Pneumonia Dataset In Images Format

Removing Unwanted Images

Artifacts (irregularities) or undesirable portions in a picture can sometimes present. After these areas have been located, classifying them into their respective categories will help in developing an effective strategy for eradication. To begin, we brought in the PIL image library and the Numpy module, which

would allow us to keep similar pixel values in separate arrays for more rapid manipulation.

After that, the Image.open() method was used to build an image object from the image and change the colour mode from RGBA to RGB. After that, we loaded the picture data into a Numpy array with the help of the np.array() method. The region's pixel values were then modified via indexed slicing. At the end, we used Image.fromarray to show a picture built from the revised pixel data ().

Resizing The Images

In order for the Python interpreter to be able to alter pictures, the Python Imaging Library (PIL) must be installed.

The Image module provides a class with the same name that may be used to represent PIL images. There are a variety of manufacturing features available in this module as well, such as picture creation and file loading tools. The image.resize function alters an image's dimensions and returns the resultant size (). All of these factors are considered:

The desired size, expressed as a 2-tuple, in pixels. (Size, Ranging)

This filter, known as "Resample," is used to resample the input signal. This may be used with PIL.Image.NEAREST (use closest neighbour), PIL.Image.BILINEAR (linear interpolation), PIL.Image.BICUBIC (cubic spline interpolation), and PIL.Image.LANCZOS (a high-quality downsampling filter). If not specified, or if mode == 1, PIL.Image.NEAREST is used.

Function that returns an Image object.

Change The Color Channel

In order to create a grayscale image, the RGB, CMYK, HSV, etc. colour spaces must be converted to grayscale. Totally dark or totally light is possible. To do this, import OpenCV, then load the original picture using imread(), and last, use cv2.cvtcolor() to convert the image to grayscale. At any point in time, users can shut all open windows when they leave a script by calling the destroyAllWindows() method.

Removing Noisy Images

We can remove the noisy parts of any image in the dataset using the command cv2.fastNImeansDenoisingColored(). The following parameters have been defined in our model for the same:

- h: the filter strength determining parameter. A higher h value removes noise more

effectively, but it also removes image details.

- ForColorComponents: the same as h, but only for colour images. (usually the same as h)

- templateWindowSize: This should be unusual. (Suggested 7)

- searchWindowSize: this should be unusual. (recommended number 21)

Module 2: Model Training and Validation

a. Model Creation Using Logistic Regression

In our approach, we combine binary classification and logistic regression methods. Classifier development often use logistic regression. Logistic regression is more complex than its simpler counterpart, linear regression. This occurs because data that are geographically distributed are difficult for linear regression to classify. Using linear regression, where the line may divide the input data into two primary groups, the data can be categorised (or classes). When there is overlap

in the data, the line is unable to reliably divide it in half. Logistic regression is able to avoid this limitation.

LR is one among the most used techniques used to make predictions about the binary values (1/0, Yes/No, True/False) of a collection of independent variables. Dummy variables are used to stand in for boolean or categorical values. Forecasting the likelihood of an occurrence by fitting data to a logistic function, the log of chances is used as the dependent variable when the outcome variable is categorical. Logistic regression may be seen as a subset of linear regression, as demonstrated by the following:

O = (1 + e)(I0*x)/(1 + e)(I1*x)).

In this formula, O represents the expected result, I0 represents the bias or intercept term, and I1 represents the coefficient for the independent variable (x). First, we must bring in Numpy. Next, we will use the LogisticRegression() method in the sklearn package to create a logistic regression model. Parameters describing the relationship between the independent and dependent variables are added to the regression object through the Fit() method on this object.

b. Model Training And Validation

We utilise a machine learning technique to assess the efficacy of the training dataset once we've separated it from the test data. scikit is used to do data fitting in this investigation. When it comes to Python machine learning libraries, Scikit-learn is your best bet. Clustering, classification, regression, and dimensionality reduction are just few of the many helpful machine learning and statistical modelling algorithms available in the sklearn toolkit.

The 'fit' function in scikit-learn is one such instrument. The 'fit' technique is used to train the algorithm based on the training data once the model has been setup. It merely serves that purpose. Hence, the sklearn fit approach use the training data to educate the ML model. Machine learning may then proceed with the help of other scikit learn algorithms like predict and score.

Here's how we put the fit approach to use on a data set meant for practise. Following the instance definition, the fit technique is applied, as seen in the following figure. Lastly, then, the features and label vector for the training dataset are included after the closing parenthesis. Both X train and Y train are names for these datasets. The X-input to fit() is a two-dimensional format called X train, and it is used to ensure a good fit of the model. If this isn't the case, our model is flawed.

After further testing on the validation dataset, we'll keep the most promising findings. Thereafter, we optimise our model by making adjustments based on the data we have. We repeat this procedure till we reach our goal. Results are validated by using a test dataset once the model has provided the optimal version.

c. Model Prediction

The final step is to create a prediction function that, given a location and a time, returns a set of potential criminal acts at that time and place. The labels of the data values may be predicted using Python's predict() function, which is based on the trained model.

The predict() method normally only takes one input, the test data. It takes an input dataset and generates labels for it based on what the model has learned about it (the "trained data"). So, the predict() method builds on top of the trained

model and employs the acquired label to map and forecast the labels for the test data.

The process begins with the loading of the dataset into the ecosystem. By utilising the pandas.read csv() function, the dataset may be read in from the system.

Afterwards, we use the accuracy score command, which we imported from the metrics metrics accuracy score, to make predictions about our model's performance.

Module 3: Web App Deployment

a. Web Development

We're going to develop the code to take care of the processing on the server side. The requesting party will contact our code. The results will specify the type and scope of the inquiries. In addition, it will determine the appropriate response to the user's query. Flask will serve as our primary tool here. It streamlines the method of creating applications for the web. Because to Flask, we can zero in on what our customers need and how to best meet their demands.

In the folder where the project files will be stored, we will create a virtual environment. First, get virtualenv set up with pip. Follow the instructions to install virtualenv, then build and activate a virtual environment, and then run the command pip install Flask. Now that the project's groundwork is in place, we can start implementing its features. A new file is created in the same directory with an overview of our site to begin things moving. Here, the file is called main.py. Now that the terminal knows what programme to run, we can launch it by typing export FLASK APP=main.py and set FLASK APP=main.py.

Once the script has been executed and the data has been written to a file, we can examine the results. The notice will be shown on your first local server-hosted webpage if you click the Link supplied there. A closer look reveals that the route() decorator in Flask is used to link a URL to a function. So that you can get the most out of our basic web app, we've added a new feature called "add url rule" (). This method, which is a function of an application object, may be used in the same way as route() to associate a URL with a certain function.

Next, we use Flask's built-in HTTP methods to construct the form. Data transport on the Internet is predicated on the Hypertext Transfer Protocol, or HTTP. This protocol specifies a number of methods that may be used to obtain information from a specified URL.

Flask's route takes care of GET requests automatically. It is possible to override this selection, though, by passing a new set of methods to the route() decorator. To illustrate the POST method's role in URL routing, let's build an HTML form and then send data from the form through POST to an existing URL. This will allow us to show how the POST method is used in URL routing. The user's name, age, gender, and any other disease-related input may be automatically generated on the HTML login page using this approach.

At last, we get to save this HTML page and give this python script a shot at launching the server. When the development server has started up, open your HTML file in a browser, fill up the text box, and hit the submit button to send the information to the server. A result indicating whether or not the illness was detected will be shown.

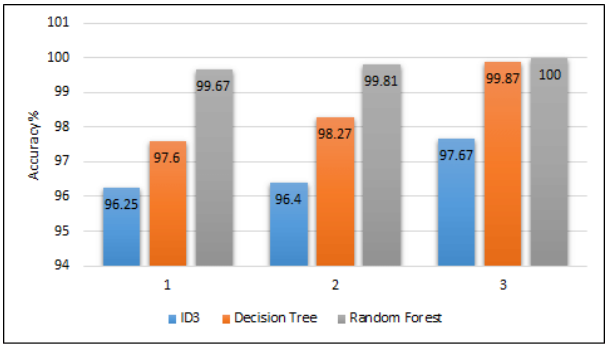
Multi-Page Creation

Alzheimer's, cancer, heart disease, lung disease, renal disease, cancer, and pneumonia are only few of the diseases that we simulate in

various variants. Next, using the flask module and python notebook, we compile everything into a single HTML file.

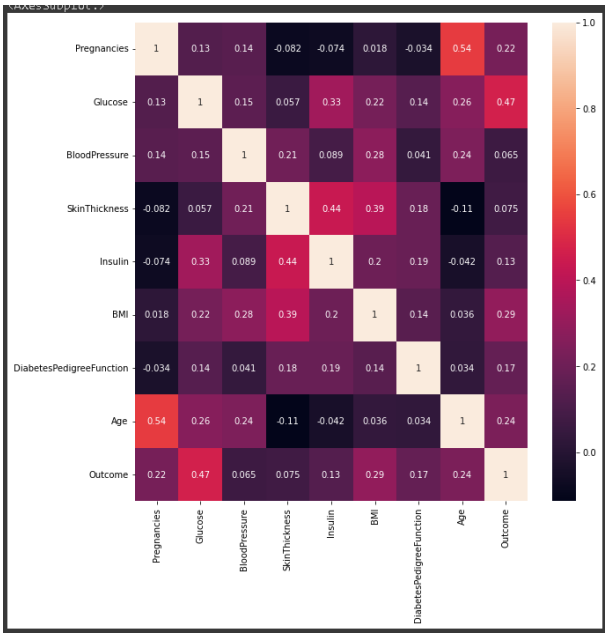
IV. EXPERIMENTAL RESULTS

Comparison of ML Algorithms

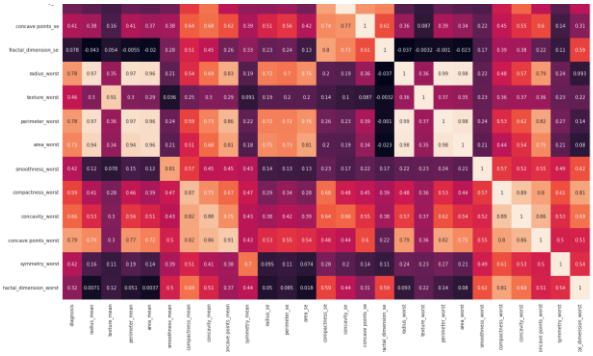


This shows that Random Forest is the most efficient algorithm

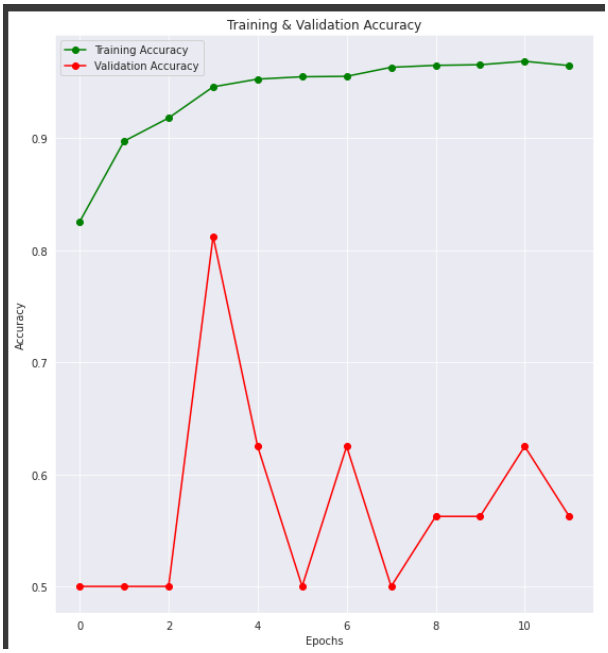
Diabetes Attribute Values

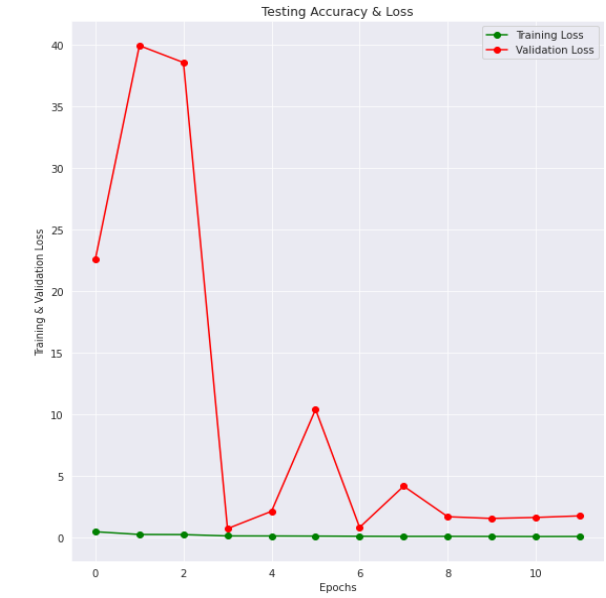


Cancer Attribute Values



Pneumonia Model Accuracy





V. CONCLUSION AND FUTURE WORK

Our goal is to create a multi-disease detection system utilising machine learning and CNN methods in order to solve the current accuracy issue. The technology will also aid in the reduction of mortality rates caused by chronic illnesses such as lung cancer,brain tumours, and cardiovascular disease. When a disease has users need to be made aware of the vulnerability and how to safeguard themselves if it has been discovered. Some other chronic diseases have large data sets, so in the future we may attempt applying this strategy to them. More illnesses and data sets may help increase accuracy.

In the future, it will be possible to predict cancer, diabetes, and pneumonia with a web application by deploying it in the cloud using bandwidth and sourcing the necessary inputs anywhere in the form of electronic health records (EHR) for cancer and diabetes as CSV datasets and for pneumonia as image datasets, with the resulting predictions being obtained in real time.

REFERENCES

[1] Liva Faes; Xiaoxuan Liu; Pearse A Kean, “A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis”, The Lancet Digital Health, Volume 1, Issue 6, October 2019

[2] M. Chen, Y. Hao, K. Hwang, L. Wang and L. Wang, "Disease prediction by machine learning over big data from healthcare communities", IEEE Access, vol. 5, no. 1, pp. 8869-8879, 2017

[3] Pushpa Singh; Narendra Singh; Krishna Kant Singh; Akansha Singh,” Diagnosing of disease using machine learning”,2021, Machine Learning and the Internet of Medical Things in Healthcare Pages 89-111, 2021

[4] Alexander Selvikvag; Lundervold; Arvid Lundervold; Katarzyna Węgrzyn-Wolska, “An overview of deep learning in medical imaging focusing on MRI”, Zeitschrift für Medizinische Physik, Volume 29, Issue 2, Pages 102-127, May 2019

[5] Yuanyuan Pan; Minghuan Fu; Biao Cheng; Xuefei Tao; Jing Guo,“Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform, IEEE Access (Volume: 8), May 2020,

[6] Dhiraj Dahiwade; Gajanan Patle; Ektaa Meshram,”Designing Disease Prediction Model Using Machine Learning Approach

[7] Dur-e-maknoon Nisar; Meshrif Alruily ; Sultan H. Almotir; Rashid Amin,“Healthcare Techniques Through Deep Learning: Issues, Challenges and Opportunities”, 2019

[8] Jer´ ome Thevenot; Miguel Bordallo Lopez;

Abdenour Hadid,“A Survey On Computer Vision For Assistive Medical Diagnosis From Faces”, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, VOL. 22, No. 5, 2018

[9] Rahatara Ferdousi; M. Anwar Hossain; Abdulmotaleb El Saddik, “Early-Stage Risk Prediction of Non-Communicable Disease Using Machine Learning in Health CPS”, IEEE Special Section On AI And IoT Convergence For Smart Health, July 2021

[10] Alice Othmania; Abdul Rahman Taleb; Hazem Abdelkawy; Abdenour Hadid, “Age estimation from faces using deep learning: a comparative analysis”, Computer Vision and Image Understanding, Volume 196, 102961, July 2020