

Symptom and Image based Disease Detection and Doctor Recommendation using Machine Learning

by

Mohammed Ahmed Hussain (Batch-SB-23-9-5 MLI)

Abstract

The cost of treatment, especially in corporate hospitals, is making the life of the citizens difficult. People dread falling sick for fear of falling into a debt trap. Most of the private hospitals vary their prices by imposing hidden fees on different people. In many situations, the cost of medicines, devices, tests, and interventions is so high and unessential that it is a sheer wastage of resources. It is evident that the market is in dire need of healthcare options. But the inflated prices in the common health services prove that there are not a lot of healthcare options for most of the people in India. There is a clear demand for healthcare services that alleviate the fear of huge debt and offer financial security. An app that uses machine learning to diagnose the symptoms and provide basic details about the disease before consulting a doctor will provide a suitable solution. For minor problems, it suggests a basic remedy and for a major problem, it recommends a doctor near you for consulting. Our system then recommends the right hospital near you and a list of doctors in your locality based on two criterion (i.e., cost of treatment and reviews of the doctor) which the patient can choose from accordingly. This eliminates the consultation fees for minor problems. An additional feature in the app will be skin and eye disease detection through image classification. As this type of disease is not minor, we will recommend a doctor based on the detected disease for all the classifications.

1. Problem Statement

About 55 million Indians were pushed into poverty in a single year due to patient-care costs, as per a study by the Public Health Foundation of India. The cost of treatment, especially in corporate hospitals, is making the life of the citizens difficult. People dread falling sick for fear of falling into a debt trap. Most of the private hospitals vary their prices by imposing hidden fees on different people. In many situations, the cost of medicines, devices, tests, and interventions is so high and unessential that it is a sheer wastage of resources. Finding the right doctor for your diagnosis can be a very painstaking process using up a lot of your time and financial resources.

2. Market/Customer/Business Need Assessment

It is evident that the market is in dire need of healthcare options. But the inflated prices in the common health services prove that there are not a lot of healthcare options for most of the people in India. There is a clear demand for healthcare services that alleviate the fear of huge debt and offer financial security. The consumers need clear and consistent information about the costs associated with medical treatments.

3. Target Specifications and Characterization

The proposed solution will cater to different types of customers in the following ways:

- Individuals and Families – By providing healthcare services, quality healthcare options at reasonable prices. and avoiding them from falling into financial hardship due to medical expenses.
- Healthcare Providers – For those who are interested in adopting more transparent pricing models and improving patient satisfaction and looking to expand their patient base by offering cost-effective services.
- Health Insurance Companies – For those who are seeking to partner up or integrate our solution to offer more affordable health insurance packages to their customers.
- Government Health Agencies – As they are responsible for healthcare policy and regulation and are interested in promoting affordable and accessible healthcare. They can look for data and insights for informed healthcare policy decisions.
- Corporations – For companies interested in employee wellness programs and healthcare cost management by providing cost-effective healthcare benefits to their employees.
- Medical Tourism Providers – Organizations involved in medical tourism that want to offer affordable and transparent healthcare options to international patients.

4. External Search

Here is the comparison of different treatment costs in New Delhi.

Treatment	Minimum Price	Maximum Price
Kidney disease	1,500/-	5,000/-
Cardio-vascular disease	1,70,000/-	4,50,000/-
Bladder Stones	21,800/-	2,10,000/-
Angioplasty	70,000/-	4,50,000/-

[Source: [Digit](#)]

These are just some of the figures and unfortunately the hospital does charge extra hidden fees on these costs. The problem is that most of the people do not earn this kind of money.

An article published by the World Health Organization in December 2021 clearly states how more than half a billion people pushed or pushed further into extreme poverty due to health care costs around the world due to the COVID-19 Pandemic. [Source: [WHO](#)]

An article by The Tribune states how medical debt is a major cause of poverty in India. [Source: [The Tribune](#)]

The article published by DownToEarth states how India's persistently high out-of-pocket health expenditure continues to push people into poverty. [Source: [DownToEarth](#)]

Recently, an article published by Scroll.in also presented the same information. [Source: [Scroll.in](#)]

5. Applicable Patents

- Disease prediction system using open source data [[Source](#)]
- Disease diagnoses-bases disease prediction [[Source](#)]
- Disease prediction system based on big data [[Source](#)]

6. Applicable Regulations

- The permission to collect patient's data.
- Unauthorized use of patient's data and medical information should be avoided.

7. Applicable Constraints

- Continuous collection and management of data.
- Continuous expansion, training, and improvement of ML algorithm.
- Domain Knowledge from medical experts.
- Access to a large network of Healthcare Professionals and Health Insurance Providers.

8. Business Model

The proposed idea is to create an app that uses machine learning to diagnose the symptoms and provide basic details about the disease before consulting a doctor. For minor problems, it suggests a basic remedy and for a major problem, it recommends a doctor for consulting. Our system then recommends the right hospital near you and a list of doctors in your locality based on two criterion (i.e., cost of treatment and reviews of the doctor) which the patient can choose from accordingly. This eliminates the consultation fees for minor problems. Upon consultation the doctors will be charged a commission for providing them patients.

We will partner with Medical Insurance providers to provide health insurance in a simple and selective manner. The patients will have the liberty to choose what kind of health insurance they need. Furthermore, for corporates and organizations, we will charge a subscription fee for seamless utilization of the service on a larger scale.

An additional feature in the app will be skin and eye disease detection through image classification. This feature will be accessible with a basic subscription to customers as a premium feature. As this type of disease is not minor, we will recommend a doctor based on the detected disease for all the classifications.

9. Concept Generation

The people around the country need something that helps them make their life easier. This solution will cater to a large number of people as it would eliminate the mostly unnecessary consultation costs, provide them with viable options according to their problems and help them to move forward without the fear of a financial disaster.

10. Concept Development

The patient's disease will be classified based on their symptoms and a solution will be provided. They will have the option to choose a doctor according to their financial capabilities only if there is a serious health concern. If not, they can save the basic consultation fees by performing a basic remedy.

11. Final Product Prototype

The patient upon logging into the application, they can give their symptoms and their disease will be detected. Here is the functioning of the app:

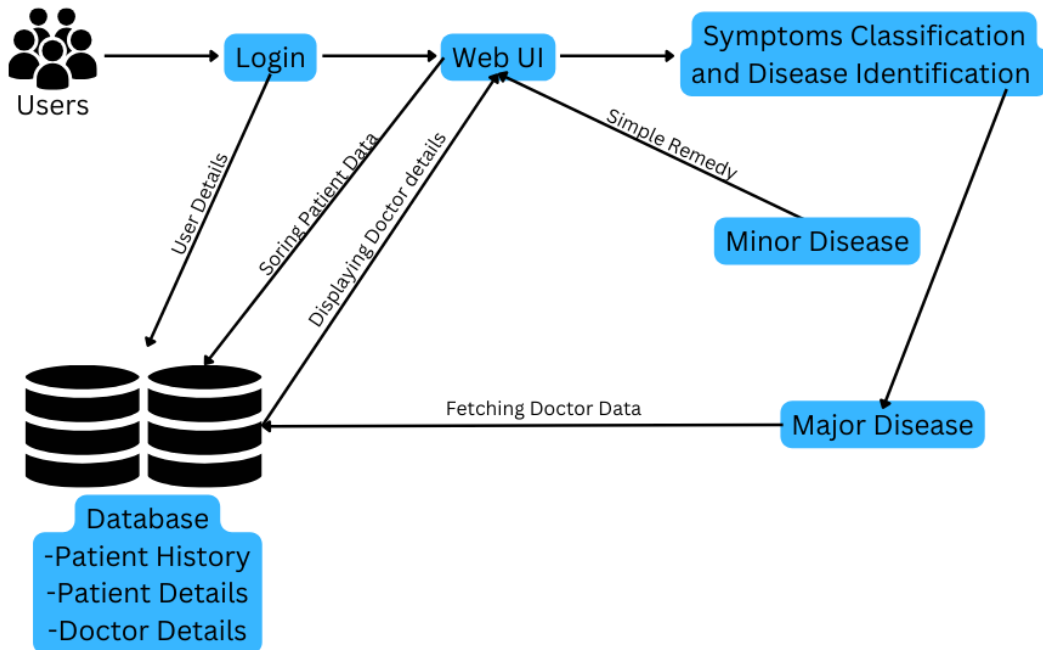


Fig. Basic Functioning of the app

Here is how the skin/eye disease detection feature works:

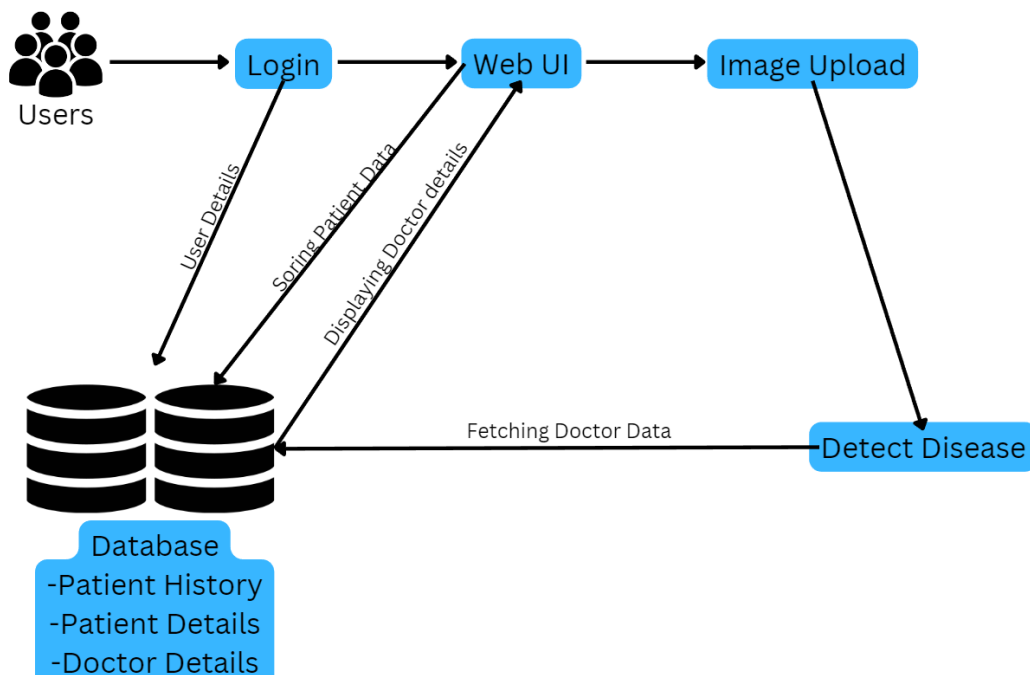


Fig. Skin/Eye disease detection feature

12. Product details

The product will consist of three parts that constitute our product.

12.1 Disease Detection

The app takes symptoms of the patient and classifies the disease. The algorithm that is used is a Decision Tree Classifier.

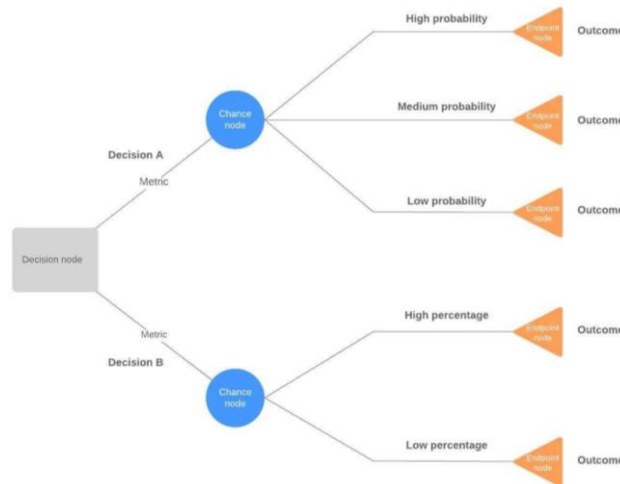


Fig. Decision Tree Classifier

As the product needs a lot of symptoms data we need to start with a small dataset and gradually scale to a large dataset. This will provide for a good overview of the product expansion. The dataset that can be use in development can be common disease dataset [Source: [Kaggle](#)]

12.2 Skin/Eye Disease Detection

The skin and eye disease detection feature can be done using a Convolutional Neural Network

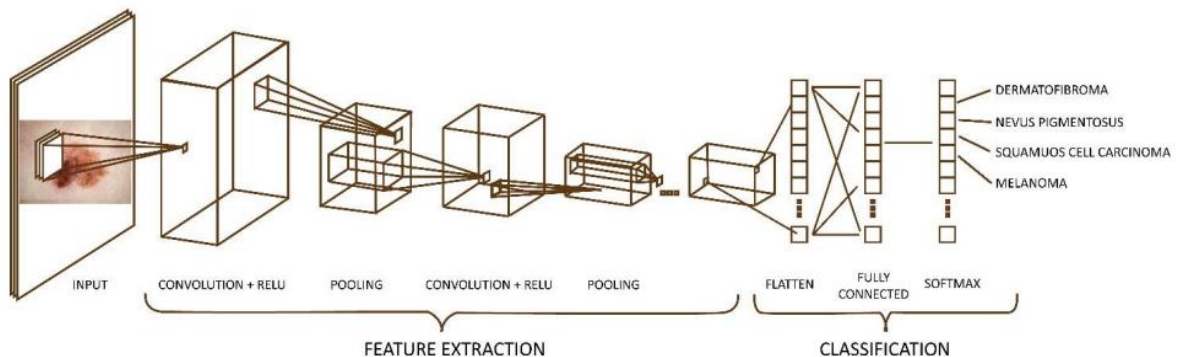


Fig. Architecture of CNN

The dataset that can be used is skin cancer ISIC [Source: [Kaggle](#)] and for eyes it is ocular disease dataset [Source : [Kaggle](#)].

12.3 Doctor Recommendation

Based on the diseases detected the doctors in the locality are displayed that are experts in treating that disease. The doctor's details need to be collected by doing research and surveys. The dataset of doctors should include doctor's details, their specialty, cost of treatment, etc. This is a major process in the development of the product.

12.4 Estimated Requirements

Software/Frameworks – Python, Pandas, TensorFlow, Keras, PyTorch, Scikit-Learn, Flask/Django, HTML, CSS, Node.js, React, Firebase, REST APIs and SQLite.

Team – Data Scientist, ML Engineer, Full-stack Web Developer, Data Engineer, Product Manager, and Domain Experts.

13. Implementation/Validation on Small Scale

Let us look at how these features work.

13.1 Common Disease Detection [[GitHub](#)]

Complete Dataset consists of 2 CSV files. One of them is training and the other is for testing your model. Each CSV file has 133 columns. 132 of these columns are symptoms that a person experiences and the last column is the prognosis. These symptoms are mapped to 42 diseases. You can classify these set of symptoms and train your model on training data and test it on testing data.

```
df.dtypes
itching                int64
skin_rash              int64
nodal_skin_eruptions   int64
continuous_sneezing    int64
shivering              int64
...
inflammatory_nails     int64
blister                int64
red_sore_around_nose    int64
yellow_crust_ooze       int64
prognosis              object
Length: 133, dtype: object
```

He is the possible prognosis that can be predicted.

```
df['prognosis'].unique()

array(['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
      'Drug Reaction', 'Peptic ulcer disease', 'AIDS', 'Diabetes ',
      'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
      'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
      'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
      'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
      'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
      'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
      'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
      'Osteoarthritis', 'Arthritis',
      '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
      'Urinary tract infection', 'Psoriasis', 'Impetigo'], dtype=object)
```

After some data preprocessing, we fit the data in a Decision Tree classifier with Gini Index for the information gain.

```
model = DecisionTreeClassifier(criterion='gini')

from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.7,random_state=3)

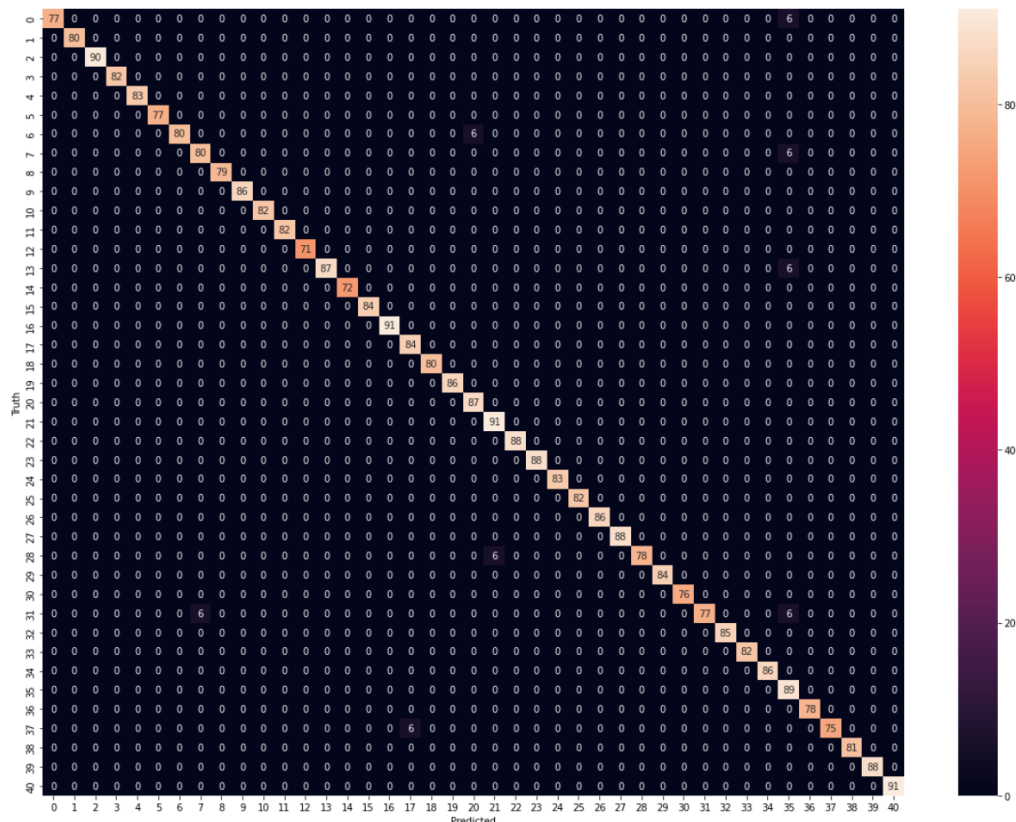
x_train.shape, x_test.shape
((1476, 132), (3444, 132))

y_train.shape, y_test.shape
((1476,), (3444,))

model.fit(x_train,y_train)

DecisionTreeClassifier()
```

Here is the heatmap between predicted values and real values.



With the given data, the model performs quite well and is highly accurate.

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	1.00	0.93	0.96	83
1	1.00	1.00	1.00	80
2	1.00	1.00	1.00	90
3	1.00	1.00	1.00	82
4	1.00	1.00	1.00	83
5	1.00	1.00	1.00	77
6	1.00	0.93	0.96	86
7	0.93	0.93	0.93	86
8	1.00	1.00	1.00	79
9	1.00	1.00	1.00	86
10	1.00	1.00	1.00	82
11	1.00	1.00	1.00	82
12	1.00	1.00	1.00	71
13	1.00	0.94	0.97	93
14	1.00	1.00	1.00	72
15	1.00	1.00	1.00	84
16	1.00	1.00	1.00	91
17	0.93	1.00	0.97	84
18	1.00	1.00	1.00	80
19	1.00	1.00	1.00	86
20	0.94	1.00	0.97	87
21	0.94	1.00	0.97	91
22	1.00	1.00	1.00	88
23	1.00	1.00	1.00	88
24	1.00	1.00	1.00	83
25	1.00	1.00	1.00	82
26	1.00	1.00	1.00	86
27	1.00	1.00	1.00	88
28	1.00	0.93	0.96	84
29	1.00	1.00	1.00	84
30	1.00	1.00	1.00	76
31	1.00	0.87	0.93	89
32	1.00	1.00	1.00	85
33	1.00	1.00	1.00	82
34	1.00	1.00	1.00	86
35	0.79	1.00	0.88	89
36	1.00	1.00	1.00	78
37	1.00	0.93	0.96	81
38	1.00	1.00	1.00	81
39	1.00	1.00	1.00	88
40	1.00	1.00	1.00	91
accuracy			0.99	3444
macro avg	0.99	0.99	0.99	3444
weighted avg	0.99	0.99	0.99	3444

```
model.score(x_test,y_test)
```

```
0.9860627177700348
```

Here are the diseases it detects based on prognosis.

```
diseases
```

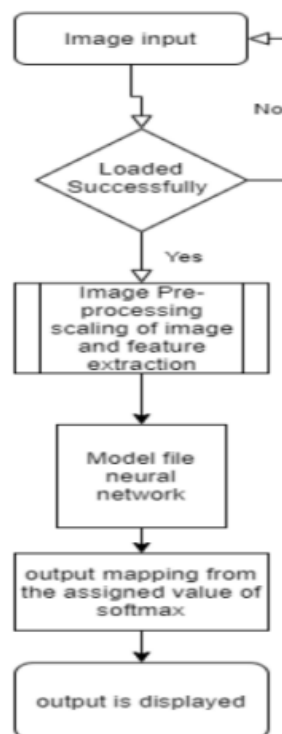
```
array(['(vertigo) Paroysmal Positional Vertigo', 'AIDS', 'Acne',
      'Alcoholic hepatitis', 'Allergy', 'Arthritis', 'Bronchial Asthma',
      'Cervical spondylosis', 'Chicken pox', 'Chronic cholestasis',
      'Common Cold', 'Dengue', 'Diabetes ',
      'Dimorphic hemorrhoids(piles)', 'Drug Reaction',
      'Fungal infection', 'GERD', 'Gastroenteritis', 'Heart attack',
      'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
      'Hypertension ', 'Hyperthyroidism', 'Hypoglycemia',
      'Hypothyroidism', 'Impetigo', 'Jaundice', 'Malaria', 'Migraine',
      'Osteoarthritis', 'Paralysis (brain hemorrhage)',
      'Peptic ulcer disease', 'Pneumonia', 'Psoriasis', 'Tuberculosis',
      'Typhoid', 'Urinary tract infection', 'Varicose veins',
      'hepatitis A'], dtype=object)
```

Here is the prediction.

```
predict_disease(['shivering', 'skin_rash', 'itching'])  
C:\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names  
warnings.warn(  
'Allergy'
```

With expansion of data and domain expertise there are a lot of improvements that can be made to the model.

13.2 Skin Disease Detection [[GitHub](#)]



This set consists of 2357 images of malignant and benign oncological diseases, which were formed from The International Skin Imaging Collaboration (ISIC). All images were sorted according to the classification taken with ISIC, and all subsets were divided into the same number of images, with the exception of melanomas and moles, whose images are slightly dominant.

The model can be designed to pre-process the image to a different texture and color-reverse which has a better option to increase its efficiency. The accuracy will definitely increase if both are treated as different models.

The data is imported and trained on a neural network.

```
prediction = Dense(len(folders), activation='softmax')(x)

# create a model object
model = Model(inputs=inception.input, outputs=prediction)
```

```
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

The data is preprocessed and divided into a train and test set.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
training_set = train_datagen.flow_from_directory('Datasets/train',
                                                target_size = (224, 224),
                                                batch_size = 32,
                                                class_mode = 'categorical')
```

Found 7455 images belonging to 11 classes.

```
test_set = test_datagen.flow_from_directory('Datasets/test',
                                           target_size = (224, 224),
                                           batch_size = 32,
                                           class_mode = 'categorical')
```

Found 742 images belonging to 11 classes.

The data is trained for 20 epochs.

```
#model ggene.
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=20,
    steps_per_epoch=len(training_set),
    validation_steps=len(test_set)
)
```

And by the end we have an accuracy of 0.8993

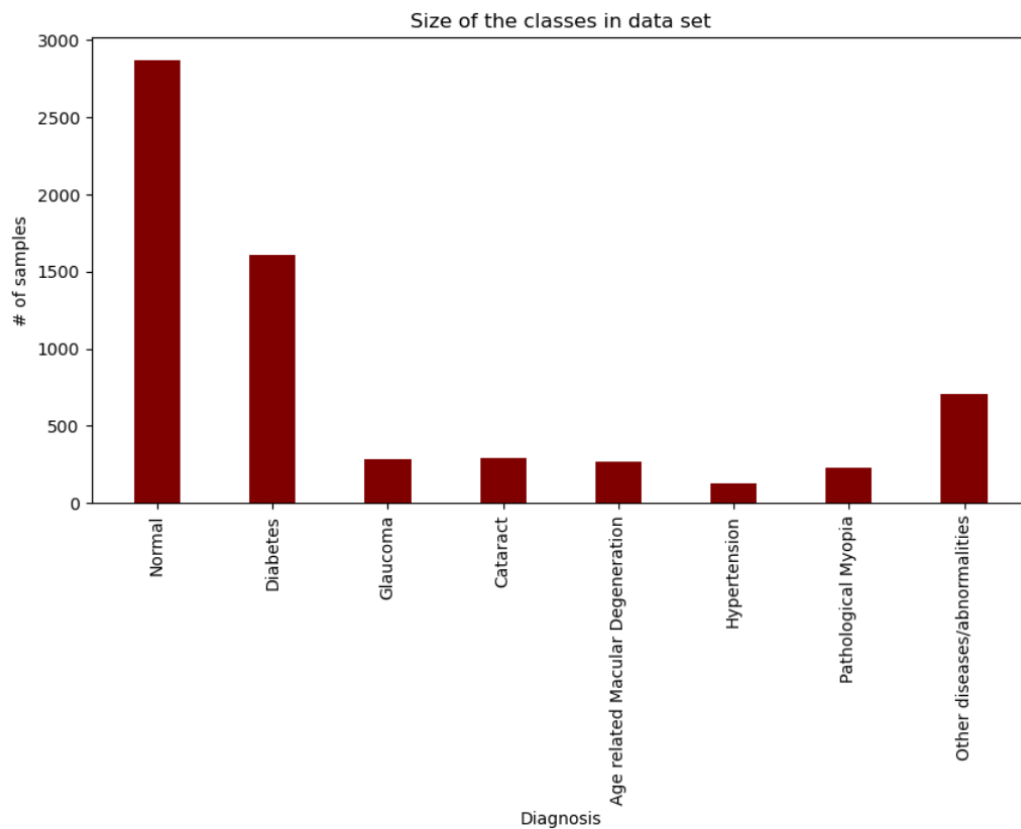
```
Epoch 20/20
233/233 [=====] - 415s 2s/step - loss: 1.8030 - accuracy: 0.8993 - val_loss: 6.5997 - val_accuracy: 0.8032
```

13.3 Eye Disease Detection [[GitHub](#)]

This dataset is meant to represent “real-life” set of patient information collected by Shangong Medical Technology Co., Ltd. from different hospitals/medical centers in China. In these institutions, fundus images are captured by various cameras in the market, such as Canon, Zeiss and Kowa, resulting in varied image resolutions.

Annotations were labeled by trained human readers with quality control management. They classify patients into eight labels including Normal (N),

Diabetes (D), Glaucoma (G), Cataract (C), Age related Macular Degeneration (A), Hypertension (H), Pathological Myopia (M), Other diseases/abnormalities (O).



Notice that the dataset is imbalanced, however there are more than 100 sample images for each class therefore there should be enough data per class to train a classification problem.

The data goes through extensive preprocessing and is fed to a neural network in batches.

```

# instantiate transfer Learning model
resnet_model = models.resnet50(pretrained=True)

# set all parameters as trainable
for param in resnet_model.parameters():
    param.requires_grad = True

# get input of fc layer
n_inputs = resnet_model.fc.in_features

# redefine fc layer / top layer/ head for our classification problem
resnet_model.fc = nn.Sequential(nn.Linear(n_inputs, 2048),
                                nn.ReLU(),
                                nn.Dropout(p=0.4),
                                nn.Linear(2048, 2048),
                                nn.ReLU(),
                                nn.Dropout(p=0.4),
                                nn.Linear(2048, 8),
                                nn.LogSigmoid())

# set all parameters of the model as trainable
for name, child in resnet_model.named_children():
    for name2, params in child.named_parameters():
        params.requires_grad = True

# Distribute the model to all GPU's
resnet_model = nn.DataParallel(resnet_model)

# set model to run on GPU or CPU abased on availability
resnet_model.to(device)

# print the transfer Learning NN model's architecture
resnet_model

```

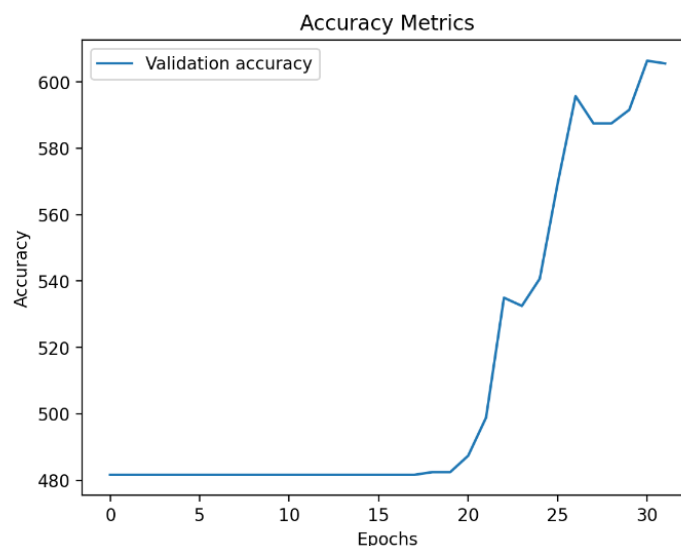
The model is trained for 32 epochs and the accuracy comes out as 73.16% and validation accuracy is 59.13%.

```

Epoch 32 Batch 120
Accuracy: 73.16 % Loss: 0.7215 Duration: 5.05 minutes
Validation Accuracy 59.13% Validation Loss: 1.2904

```

The accuracy tends to increase at the later epochs moving forward.



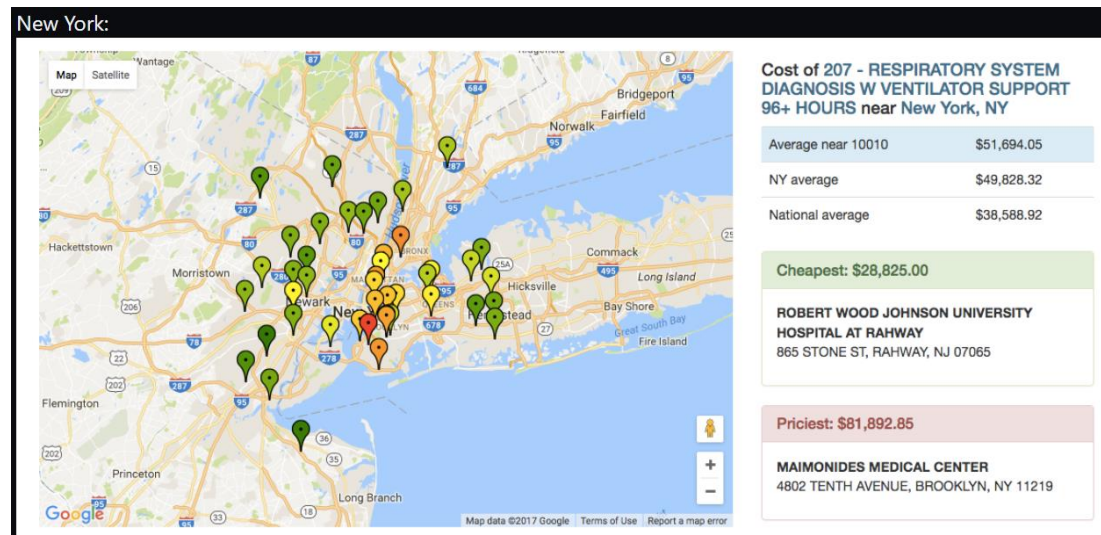
As the accuracy is not that great, more computational power and higher quality of data is required.

13.4 Doctor Recommendation [[GitHub](#)]

As of now there is no proper way to recommend doctors as it varies from place to place. Although there is a GitHub repo that might help solve this problem.

Medi-Compare is a Web app that helps patients to compare the costs of medical procedures at hospitals in their region. By making hidden healthcare costs more transparent, we're empowering citizens to take a stand on their health and wallet.

This project was done for cities in the United States.



If this is implemented for India with proper data and integrated into our system, it will revolutionize the Healthcare industry forever.

This is a combination of services that can help the masses and eliminate the fear of medical debt in people forever.

14. Conclusion

In conclusion, our product represents a significant step forward in addressing the pressing issue of healthcare accessibility and affordability in the modern age. The convergence of machine learning, medical data, and intelligent recommendation systems has empowered us to create a solution that holds the promise of revolutionizing the healthcare landscape. Through rigorous experimentation and careful algorithm selection, we can demonstrate the feasibility of accurately detecting diseases based on symptoms and medical images.

Our product is not without its challenges and limitations. The availability of diverse and high-quality medical data remains a hurdle, and the interpretability of machine learning models in healthcare continues to be an area of active research. However, these challenges serve as catalysts for future work and innovation, encouraging us to explore new avenues and improve upon our existing framework. Furthermore, our product contributes to the broader field of machine learning in healthcare. It aligns with the growing body of research focused on harnessing AI to tackle real-world medical challenges, ultimately improving the quality and accessibility of healthcare services.