ML_Final_Project
Team 07

Team members
Akhila Bezawada – ab1545
Sridhar balaji - sg1237
Sai Arvind Reddy – sd1253
Vineeth Appala - va1025

In our ML project we have used two datasets. We initially started working on this project from start but since it's working on only Neural networks. We are not sure about the project evaluation how it will be going on if we have applied only neural network on lung cancer detection (without using csv, we have used PNG files(CT scan) to determine cancer cells, so we have chosen other dataset(wine quality) to apply classification and regression methods.
1. Wine quality detection where we have applied regression and classification methods to determine the quality of the wine

2. Lung cancer detection where it's an ongoing project we have used PNG images to determine the cancer by using all the patient records like CT scan reports, we tried to determine the cancer among patients by using neural network techniques.

Dataset 1: Wine Quality		
Preprocessing		Introduction
Methods		Related Work: Classification, Regression
Evaluation		Evaluation Methodology
Results		Results
1		INTRODUCTION
	1.1	Dataset 2: Lung Cancer
	1.2	Tests to diagnose Lung Cancer
	1.3	Why early detection is necessary?
2		BACKGROUND THEORY
	2.1	Deep Learning
	2.2	Convolutional Neural Network
	2.3	Technologies Used
3		METHODOLOGY
	3.1	Data Understanding
	3.2	Data Preparation
	3.2.	Averaging Method
	3.2.2	Image Enhancement and Lung Segmentation
4		RESULT ANALYSIS
	4.1	Training Phase
	4.2	Results
5		CONCLUSIONS & FUTURE SCOPE
	5.1	Work Conclusions
	5.2	Future Scope of Work
REFERENCES		

Introduction

Dataset used: winequality.csv dataset from UCI repository to check the quality of the wine this datset variants of the Portuguese "Vinho Verde" wine. This dataset can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones).

Problem: The price of wine depends on a rather abstract concept of wine quality; this problem enables wine makers to resolve the quality of wine so that certification and quality assessment and assurance process is more controlled.

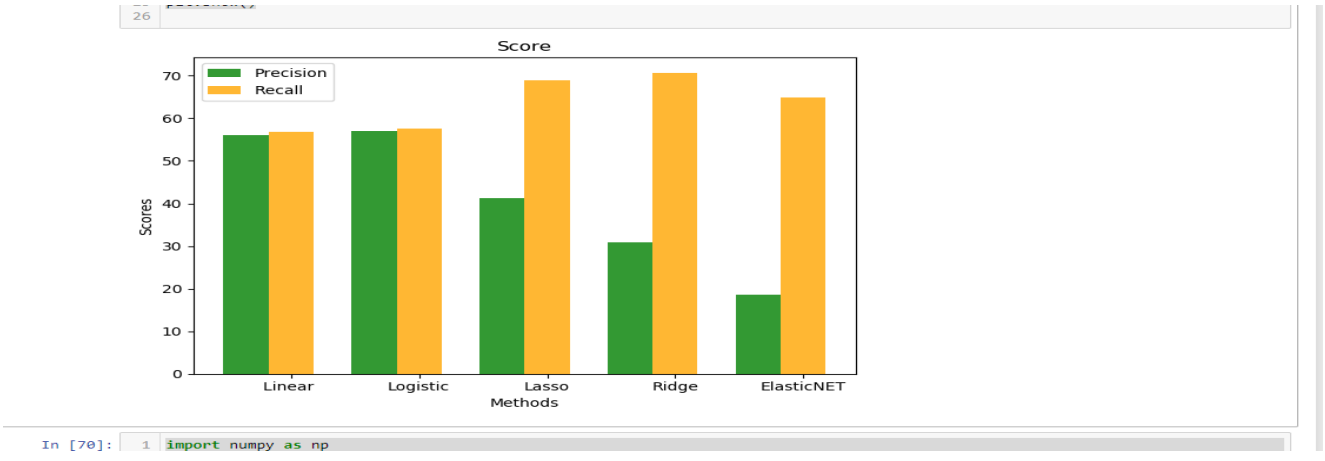
Inference: Since, it's a theory formed around implicit analysis based on factors like acidity, pH level, the presence of sugar and other chemical properties, it's an inference.
It is So, we have applied following regression and classification methods to test the quality of the wine. We are getting good test and train accuracy to determine the quality.

Related work:
In paper[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. They have proposed a data mining approach to predict human wine taste preferences that is based on easily available analytical tests at the certification step.

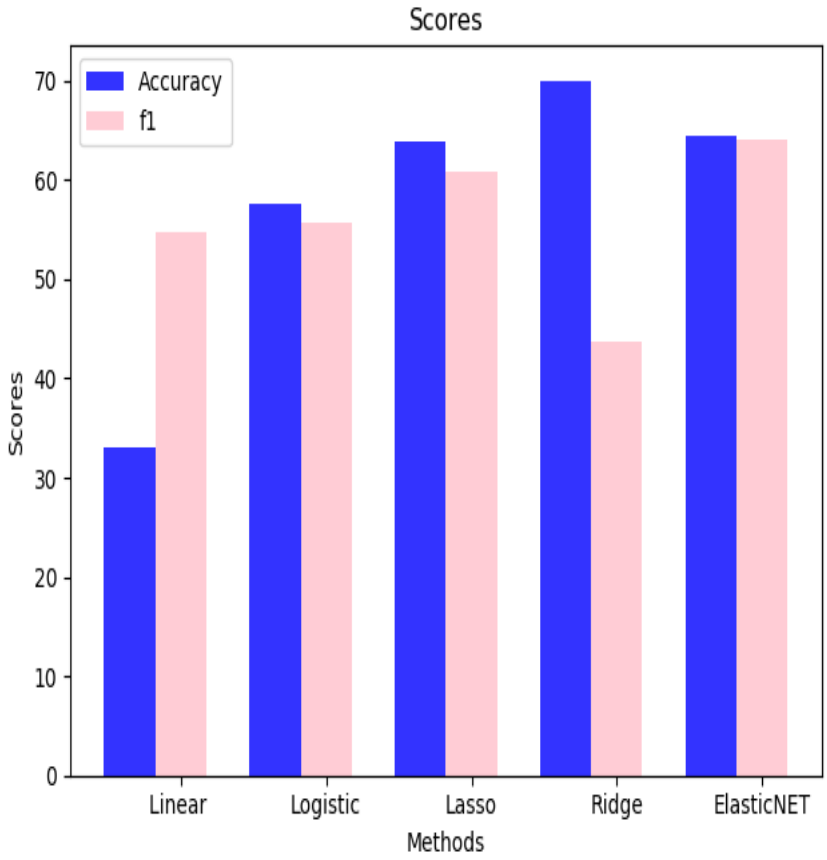
In paper[2] S. Aeberhard, D. Coomans and O. de Vel, Comparison of Classifiers in High Dimensional Settings,Tech. Rep. no. 92-02, (1992. They proposed three different wine 'categories' and goal will be to classify an unlabeled wine according to its characteristic features such as alcohol content, flavor, hue etc.

Evaluation methodology
Preprocessing
We have done cleaning the data since it contains some categorical data, we have converted to numeric to get good accuracy in preprocessing step. We have divided train and test data into 70% and 30% respectively.
We have implemented following algorithms in Python programming language. We get the following graphs for precision and recall when we train the data set. We have done bootstrapping to to improve the stability and accuracy of machine learning algorithms. We have plotted graphs for test and train accuracy and f1 and validated the results are good.

Regression Models



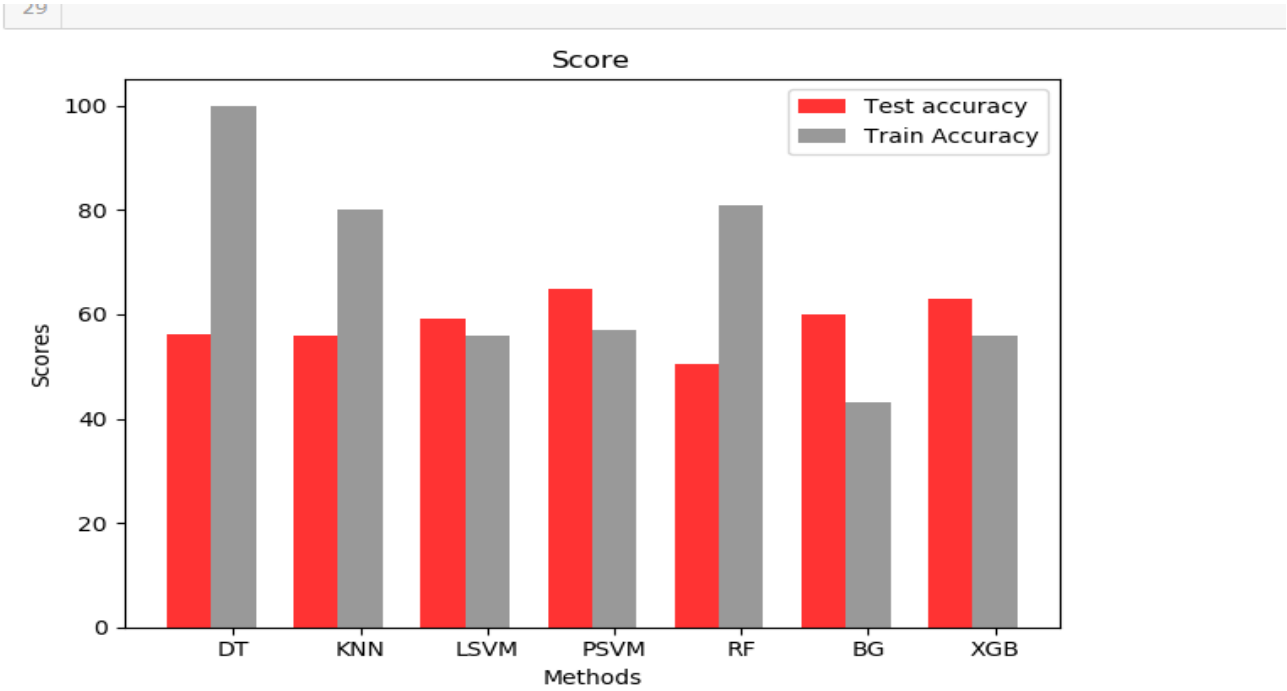
```
32 plt.show()
```



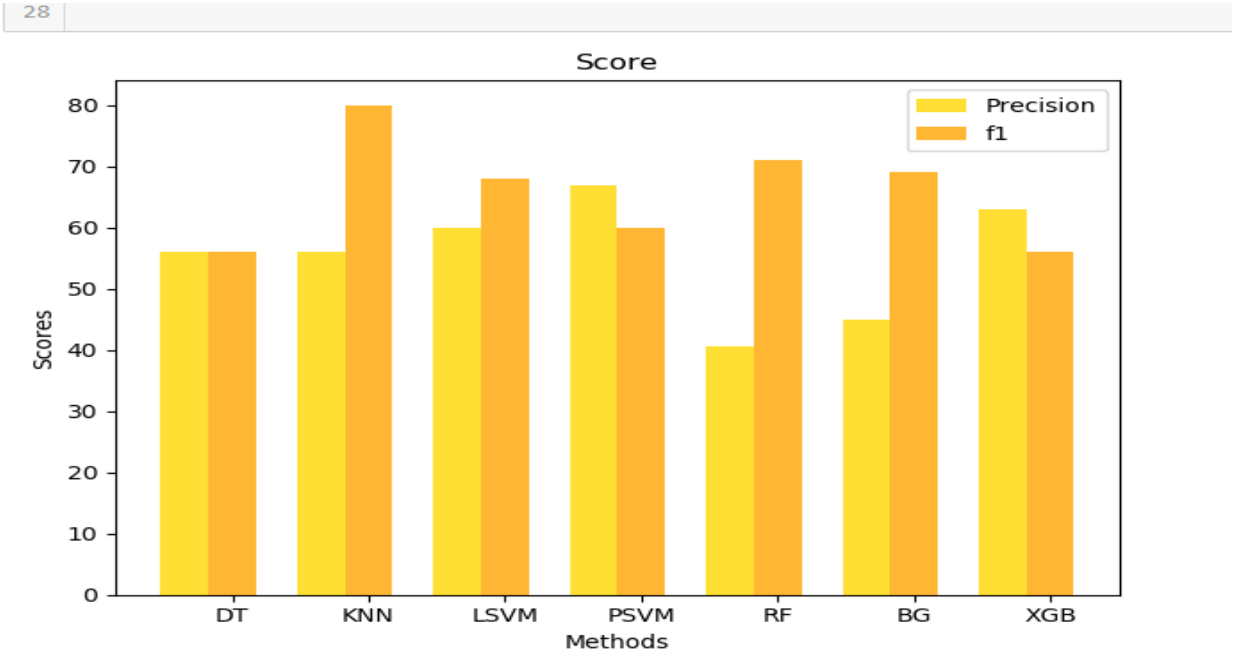

```
[75]: 1 import matplotlib.pyplot as plt; plt.rcdefaults()
```

	Accuracy	Precision	Recall	F1
Linear	33.10%	56.74%	56.74%	54.30%
Logistic	57.08%	57.43%	57.08%	55.30%
lasso	63.2%	68.92%	41.67%	60.2%
Ridge	70.66%	70.21%	30.21%	42.01%
Elasticnet	64.21%	64.01%	18.21%	40.31%

Classification Model



01
21



	Test Acc	Train Acc	Precision	Recall	F1
Decision	0.56	1.00	0.56	0.54	0.56
KNN	0.56	0.80	0.56	0.63	0.68
LSVM	0.59	0.56	0.60	0.30	0.68
PSVM	0.64	0.57	0.66	0.34	0.60

RF	0.70	0.81	0.40	0.60	0.70
Bagging	0.60	0.40	0.47	0.74	0.69
Boosting	0.62	0.46	0.56	0.56	0.568

Results

From our analysis the best method is Random Forest. The results of the Random Forest model on the test set are as follows: -

Train Accuracy of Model: 81.0 %
Test Accuracy of Model: 70.0 %
Precision of Model: 40.2678 %
Recall of Model: 60.8397 %
F1 score of the model: 70.10237 %

Conclusion

Various models were implemented and the best method we came across among all the models is Random Forest. In the future we plan to implement time series on given datasets and even try to improvise algorithms to get better results.

Dataset 2: Lung cancer detection

ABSTRACT

Lung cancer is one of the dangerous and life taking disease in the world. However, early diagnosis and treatment can save life. Although, CT scan imaging is best imaging technique in medical field, it is difficult for doctors to interpret and identify the cancer from CT scan images. Therefore, computer aided diagnosis can be helpful for doctors to identify the cancerous cells accurately. Many computer aided techniques using image processing and machine learning has been researched and implemented. It is difficult to detect because it arises and shows symptoms in final stage. However, mortality rate and probability can be reduced by early detection and treatment of the disease. Best imaging technique CT imaging are reliable for lung cancer diagnosis because it can disclose every suspected and unsuspected lung cancer nodule. However, variance of intensity in CT scan images and anatomical structure misjudgement by doctors and radiologists might cause difficulty in marking the cancerous cell. Recently, to assist radiologists and doctors detect the cancer accurately computer Aided Diagnosis has become supplement and promising tool.

The first stage starts with taking a collection of CT images (normal and abnormal). The second stage consists in making a basic model without any changes to the given dataset. The third stage consists image enhancement to acquire quality in images. The fourth stage applies image segmentation which plays an effective role in image processing stages. The fifth stage is training our model using Convolutional Neural Networks (CNN).

Our basic model has shown decent accuracy without any changes in the given dataset. But the accuracy has been improved with the application of image enhancement and segmentation techniques.

LIST OF TABLES

LIST OF FIGURES

Introduction

1.1: Lung Cancer

Cancer is when cells in the body change and grow out of control. Your body is made up of tiny building blocks called cells. Normal cells grow when your body needs them, and die when your body does not need them any longer.
The lungs are sponge-like organs in your chest. Their job is to bring oxygen into the body and to get rid of carbon dioxide. When you breathe air in, it goes into your lungs through your windpipe (trachea). The trachea divides into tubes called bronchi, which enter the lungs. These divide into smaller branches called bronchioles. At the end of the bronchioles are tiny air sacs called alveoli. The alveoli move oxygen from the air into your blood. They take carbon dioxide out of the blood. This leaves your body when you breathe out (exhale).

Lung cancer is cancer that starts in the cells that make up the lungs. Many other types of cancer, such as breast or kidney, can spread to the lungs. Specific type of cancer is based on the site of the original tumor.

1.2: Tests to diagnose Lung Cancer

Present Techniques used to diagnose Lung Cancer:

Imaging tests. An X-ray image of your lungs may reveal an abnormal mass or nodule. A CT scan can reveal small lesions in your lungs that might not be detected on an X-ray.

Sputum cytology. If you have a cough and are producing sputum, looking at the sputum under the microscope can sometimes reveal the presence of lung cancer cells.

Tissue sample (biopsy). A sample of abnormal cells may be removed in a procedure called a biopsy.

This method is called bronchoscopy, in which the doctor examines abnormal areas of your lungs using a lighted tube that's passed down your throat and into your lungs; mediastinoscopy, in which an incision is made at the base of your neck and surgical tools are inserted behind your breastbone to take tissue samples from lymph nodes; and needle biopsy, in which your doctor uses X-ray or CT images to guide a needle through your chest wall and into the lung tissue to collect suspicious cells.
A biopsy sample may also be taken from lymph nodes or other areas where cancer has spread, such as your liver.

1.3: Why early detection is necessary?

Most lung cancers could be prevented, because they are related to smoking (or second-hand smoke), or less often to exposure to radon or other environmental factors. But some lung cancers occur in people without any known risk factors for the disease. It is not yet clear if these cancers can be prevented.

Most lung cancers have already spread widely and are at an advanced stage when they are first found. These cancers are very hard to cure. Survival from lung cancer is directly related to its growth of the detection time but people do have a higher chance of survival if the cancer can be detected in the early stages.

BACKGROUND THEORY

2.1: DEEP LEARNING

Deep learning excels on problem domains where the inputs (and even output) are analog. Meaning, they are not a few quantities in a tabular format but instead are images of pixel data, documents of text data or files of audio data.
Computer programs that use deep learning go through much the same process. Each algorithm in the hierarchy applies a nonlinear transformation on its input and uses what it learns to create a statistical model as output. Iterations continue until the output has reached an acceptable level of accuracy. The number of processing layers through which data must pass is what inspired the label deep.

Neural Network:
A neural network is a model that has been inspired by the brain, the brain consists of nearly 10 billion neurons with 60 trillion connections between each other. A neuron (Fig 2.1) consists of a cell body called the soma where the nucleus is found, many dendrites where input signals are received and transmitted and a synapse which is basically connections between neuron to neuron.

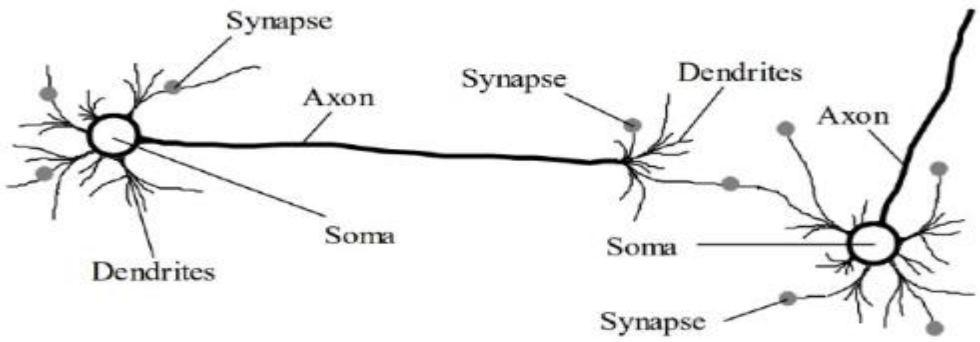


Fig 2.1 – Neuron of Brain

Artificial Neuron:
Like a brain neuron, neural networks consist of an artificial neuron. An artificial neuron (see Fig 2.2) consists of a weight to determine the strength of a connection, a linear function that needs to be computed

and an activation function that computes the weighted sum of the linear function that is then compared to a threshold value.
It is also worthy to note that artificial neurons can be modified to contain different linear function algorithms such as Gradient Descent, Logistic Regression, etc and also different activation functions such as ReLU (Rectified Linear Unit) or TanH functions which depends on the problem to solve.

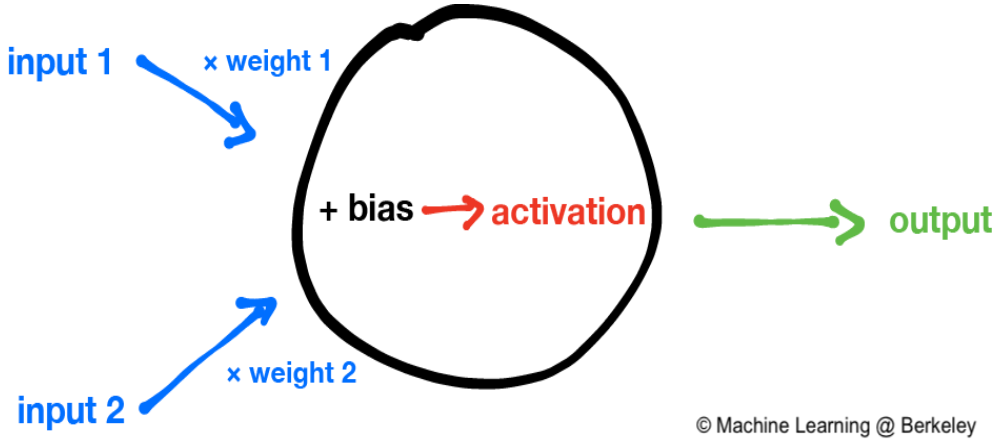


Fig 2.2 – Artificial Neuron

Artificial Neural Network:
ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data.
An Artificial Neural Network (see Fig 2.3) is an interconnected architecture where there exists an input layer where input data is placed, a hidden layer(s) where artificial neurons are stacked on top of each other and an output layer where the prediction or classification is made.

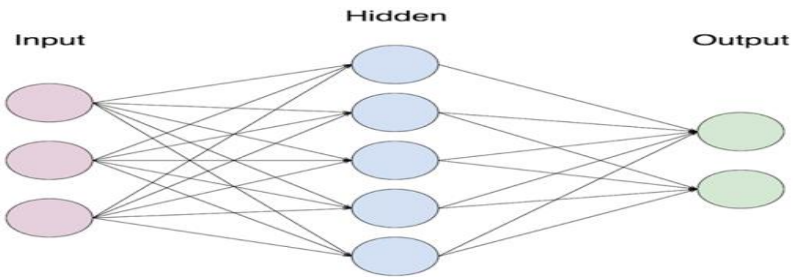


Fig 2.3 – Artificial Neural Network

Forward Propagation:
Forward propagation (see Fig 2.4) is technique in which data moves through from the corresponding input layer, hidden layers and output layer sequentially.
A hidden layer usually consists of a weight which is used in an optimization algorithm such as Gradient Descent, an activation function such as Sigmoid, TanH, LeakyReLU etc. and a Loss Function to calculate the loss or error of the function which we use to back propagate to adjust the weights

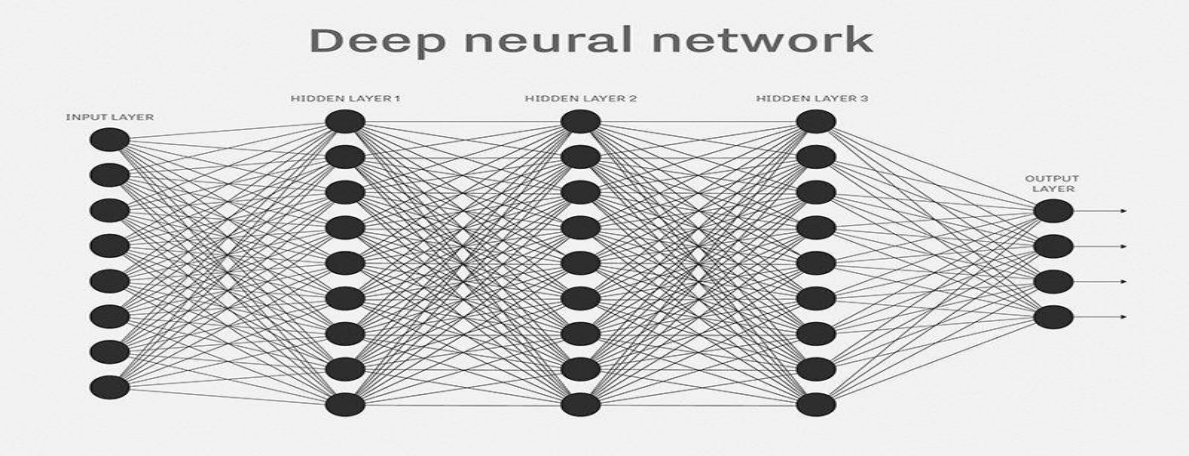


Fig 2.4 – Deep Neural Network

Back Propagation:
Back propagation is the opposite of Forward propagation as it feeds the network backwards. Back propagation is used to adjust the weight of the neural network after the errors have been computed by forward propagation algorithms.

Activation Function:

Activation functions (see Fig 2.5) are an important part of a neural network. They allow neural networks to create non-linear functions to solve problems. The 3 most used activation functions are Sigmoid, TanH and ReLU. Activation functions allow or stop neurons from firing into the next layer by comparing it into the activation function thresholds.

Activation functions are used both in the forward and backward propagation where in the forward propagation an activation function is used to calculate the loss where the output of a function is compared to the real number and in backward propagation they are used to update the parameters of the neural network.

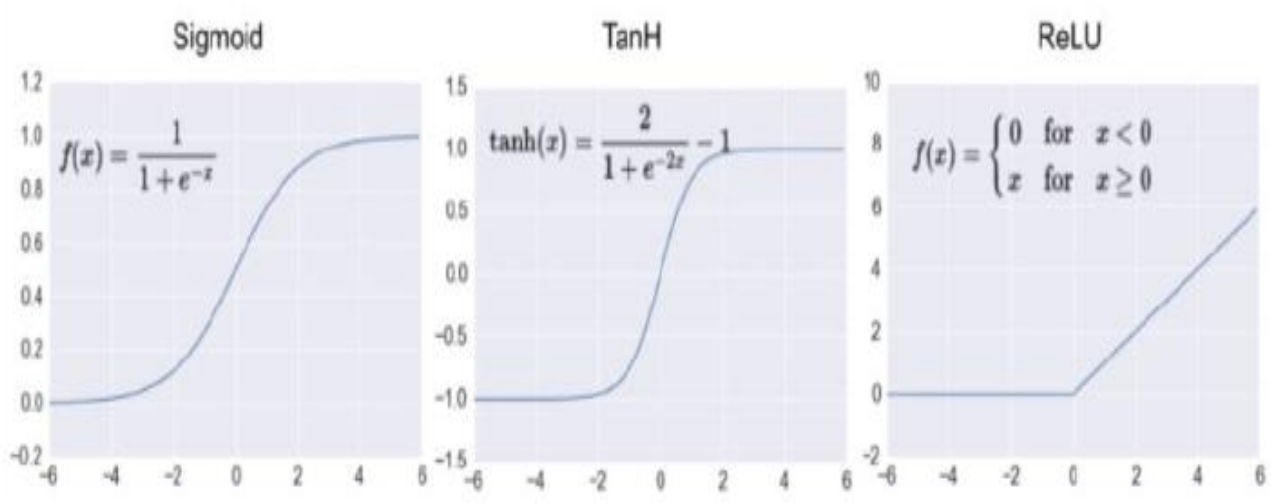


Fig 2.5 – Activation Function

<u>2.2:</u> Convolutional Neural Network
Convolutional Neural Networks is all about using Deep Learning with Computer Vision. A good way to gain intuition about this is to think about a Neural Network Architecture and how it is applied to

visual tasks i.e. Images and Video. Convolutional Neural Networks have allowed us to create Facial Recognition, Object Recognition (Fig 2.6), Self-Driving Cars and more.

Fig 2.6 – Object Detection

Architecture:
Like a Neural Network, a typical Convolutional Neural Network (see Fig 2.7) consists of a multiple hidden layers called a Convolutional Layer where the linear function computes the strided convolutions over an image to extract features.

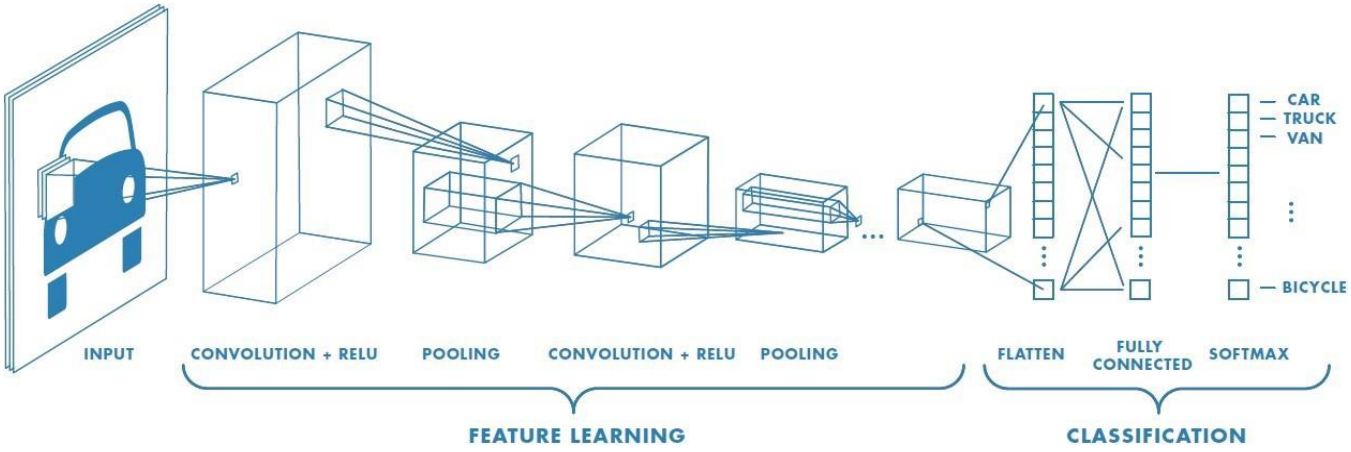


Fig 2.7 – CNN Architecture

Convolution Layer:

In a Convolutional Neural Network (Fig 2.8), the linear function that is used is called a convolutional layer. Each node in the hidden layer extracts different features by using image processing feature detectors. For example, in the first layer, the first node may extract the horizontal edges of an image, the second node may extract vertical edges and etc. These features are extracted using a kernel.

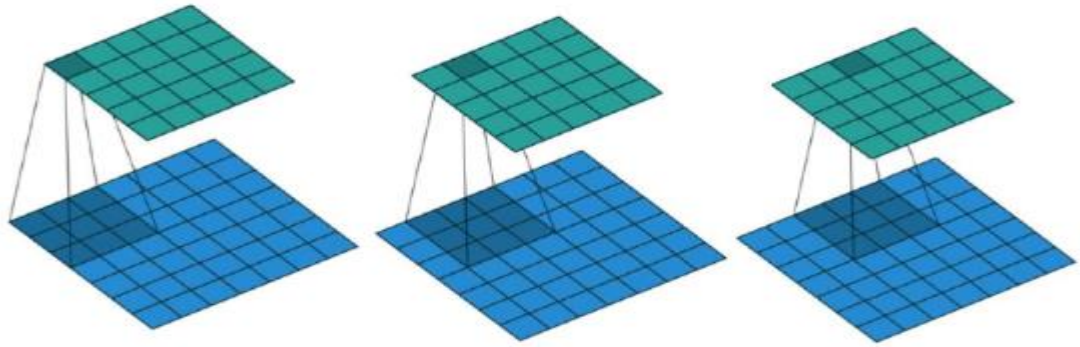


Fig 2.8 – Convolution Layer

Stride:

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 (see Fig 2.9).

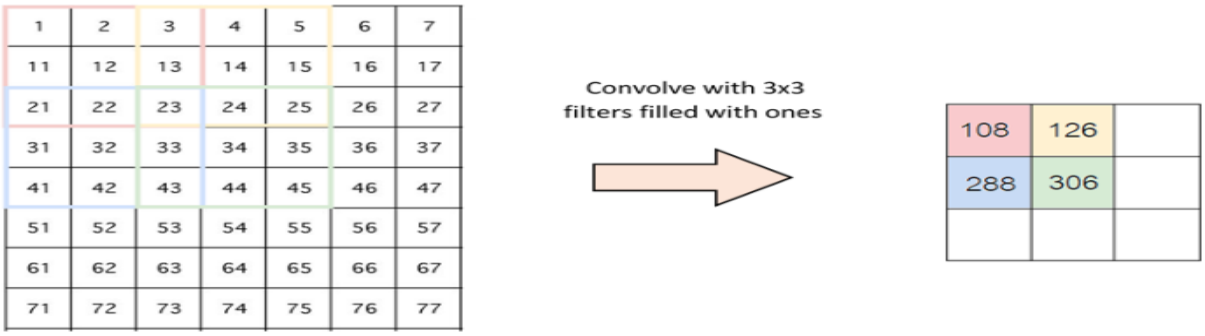


Fig 2.9 – Stride of 2

Padding:

Without padding (Fig 2.10) the edges of an image might be neglected in feature extraction. To ensure that no feature extraction of edges is missed, we apply padding to improve the efficieny of the model.

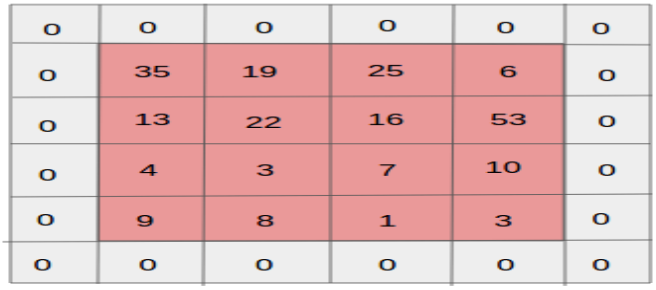


Fig 2.10 - Padding

Pooling Layer:

The pooling layer (Fig 2.11) happens to be computed after the convolutional layer.

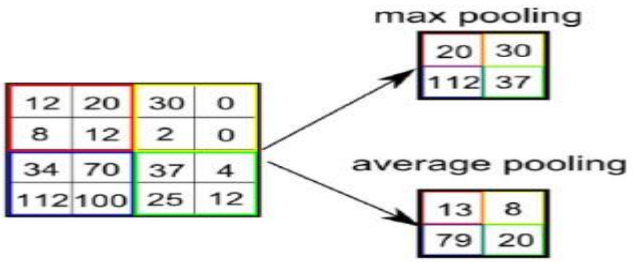


Fig 2.11 - Pooling

Flattening:
Flattening is an important step where pooled features map into a column. After the flattening step, we end up with a long vector of input data that you pass through the artificial neural network to have it processed further.

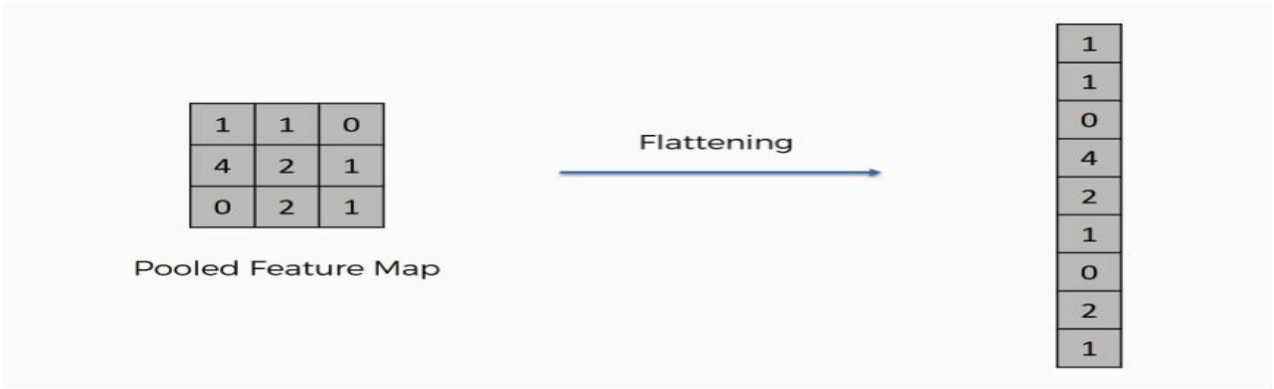


Fig 2.12 - Flattening

2.3: TECHNOLOGIES USED

Python: Python is a high-level interpreted language used for general purpose programming
Keras: Keras is also a Deep Learning Framework that abstracts much of the code in the other Frameworks like Tensorflow and Theano. Compared to the other frameworks Keras is more minimalist.
Jupyter Notebook IDE: The Anaconda distribution comes with a variety of software that includes Jupyter Notebooks for scientific computing. Jupyter Notebooks is an open source software IDE that allows developers to create and share documents that contain live code and more.
Spyder: Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging and introspection features.

METHODOLOGY

3.1: DATA UNDERSTANDING

This phase is about collecting the data, gaining familiarity and ultimately understanding the data. This data deals with DICOM (Digital Imaging and Communications in Medicine) files. DICOM is the

international standard to transmit, store, retrieve, print, process and display medical imaging information. DICOM is used for communication and management of medical imaging information and related data. It is most commonly used for storing and transmitting medical images. They are used for creating visual representations of the interior of a body for clinical analysis. This process is done by analysing DICOM files which visually represent the working of organs or tissues present inside the human body.

(0008, 0000) Group Length	UL: 358
(0008, 0005) Specific Character Set	CS: 'ISO_IR 100'
(0028, 1053) Rescale Slope	DS: '1'
(0028, 1052) Rescale Intercept	DS: '-1024'
(0008, 103e) Series Description	LO: 'Axial'
(0010, 0000) Group Length	UL: 64
(0010, 0010) Patient's Name	PN: '0015ceb851d725lb8f399e39779dle7d'
(0010, 0020) Patient ID	LO: '0015ceb851d725lb8f399e39779dle7d'
(0010, 0030) Patient's Birth Date	DA: '19000101'
(0018, 0060) KVP	DS: ''
(0020, 0000) Group Length	UL: 390
(0028, 0030) Pixel Spacing	DS: ['0.693359', '0.693359']
(0028, 0100) Bits Allocated	US: 16
(0028, 0101) Bits Stored	US: 16
(0020, 0032) Image Position (Patient)	DS: ['-177.500000', '-177.500000', '-384.109985']
(0028, 0102) High Bit	US: 15
(0020, 1041) Slice Location	DS: '-384.109985'
(0028, 0000) Group Length	UL: 200
(0028, 0120) Pixel Padding Value	US or SS: b'0\xf8'
(7fe0, 0010) Pixel Data	OB or OW: Array of 524288 bytes

Table 3.1 – Metadata of DICOM file

3.2: DATA PREPARATION

3.2.1: Averaging Method

The downloaded data consists of many 2D-slices (DICOM files), which, when combined, produce a 3-dimensional rendering of the scans. In this case, that's the chest cavity of the patient. We've got scans about 1368 patients, and then we've got another file that contains the labels for this data.

There exists a Python package called 'pydicom' for reading DICOM files. Once read, we get the list of patients by their IDs, and their associated labels in a dataframe (see Table 3.2). Now, we can begin to iterate through the patients and gather their respective image data. We're almost certainly going to need to do some pre-processing of this data.

ID	Cancer Label
0a0c32c9e08cc2ea76a71649de56be6d	0
0c0de3749d4fe175b7a5098b060982a1	1
0bd0e3056cbf23a1cb7f0f0b18446068	0
0c60f4b87afcb3e2dfa65abbbf3ef2f9	1
00cba091fa4ad62cc3200a657aeb957e	0

Table 3.2 – Dataframe of labels

Each patient consists around 200 scans and the resolution is about 512 x 512. Below is the sample data consisting the resolution and no. of scans of 10 patients.

```
(512, 512) 195
(512, 512) 265
(512, 512) 233
(512, 512) 173
(512, 512) 146
(512, 512) 171
(512, 512) 123
(512, 512) 134
(512, 512) 135
(512, 512) 191
```

This means our 3D rendering is a 200 x 512 x 512 which is huge. So, we are going to absolutely need to resize the data. So, a DICOM file has many attributes but the 'pixel_array' attribute consists of the image data. A sample CT scan as shown in the (Fig 3.1).

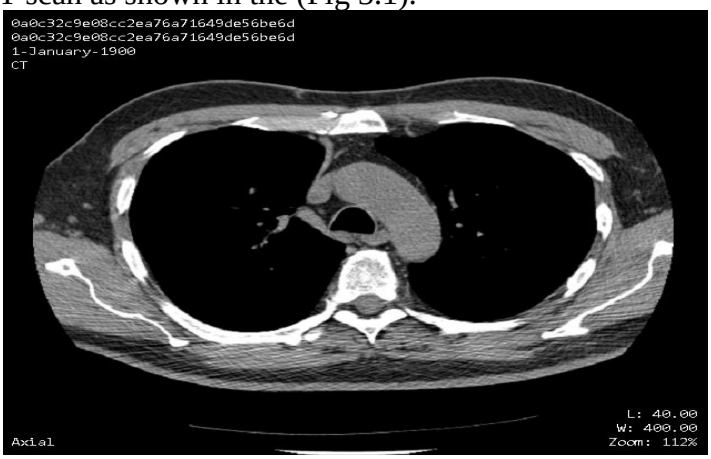


Fig 3.1 – CT Scan

The number of slices for each patient is way too big for a convolutional neural network(CNN) to handle without some serious computing power. So, we need to down sample this data. We resize our images from 512 x 512 to 100 x 100 which is also huge data but a GPU can handle this data. The resize is done using the 'openCV' tool.

Having around 200 scans for each patient is troublesome, so, we reduce it to 20 scans instead.

```
195 20
265 20
233 20
173 20
146 20
```

Now we've got pre-processed, normalized data which is ready to be fed through a neural network(CNN).

3.2.2: Image Enhancement and Lung Segmentation

This is another method adopted to observe the accuracy differences obtained in both the methods. In this method we are going to apply image enhancement or segmentation techniques for the original data.

Gabor Filter.

The pre-processing steps followed are:

Loading the DICOM files: These files contain a lot of metadata but the requirement for a neural network is the pixel data of an image which can be derived from a 'pixel_array' attribute.

Apply Gabor Filter: On the image data derived from the DICOM files, Gabor Filter has been applied as an image enhancement technique.

Hounsfield Units(HU): The unit of measurement in CT scans is the Hounsfield Unit (see Table 3.3), which is a measure of radio density. In this step, we convert the pixel values of the image to Hounsfield units and these unit values correspond to specific tissues.

Substance	HU
Air	−1000
Lung	−500
Fat	−100 to −50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

Table 3.3 – Hounsfield (HU) Table

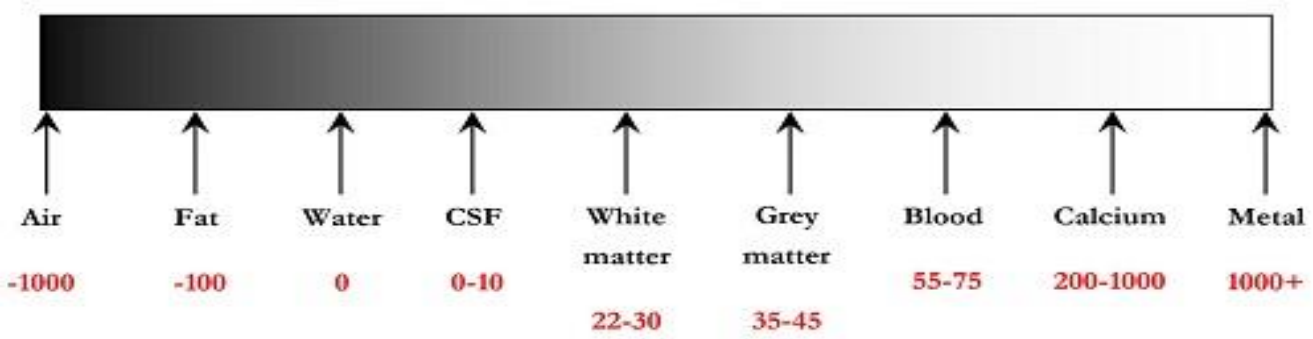


Fig 3.2 – Hounsfield Intensity Scale

By default, the slices do not return values in this unit. The pixel values of the image can be converted into HU units by multiplying with the rescale slope and adding the rescale intercept. (which are stored in the meta data of the scans). The Hounsfield intensity scale is shown in the (Fig 3.2)

Below is a graph between Hounsfield units and its Frequency.

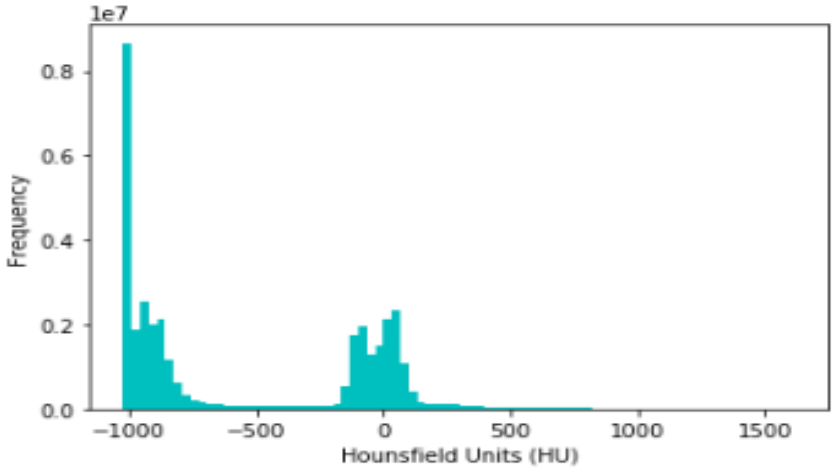


Fig 3.3 – Histogram

Resampling: A scan may have pixel spacing of [2.5, 0.5, 0.5], which means that the distance between slices is 2.5 millimetres. For a different scan this maybe [1.5, 0.75, 0.75], this can be problematic for automatic analysis. A common method of dealing with this is resampling the full dataset to a certain isotropic resolution. If we choose to resample everything to 1mm x 1mm x 1mm pixels we can use 3D convnets without worrying about slice thickness invariance.

3D plotting: For visualization it is useful to be able to show a 3D image of the scan. We used matplotlib to visualize the 3D image.

3D plotting of Bone Structure:

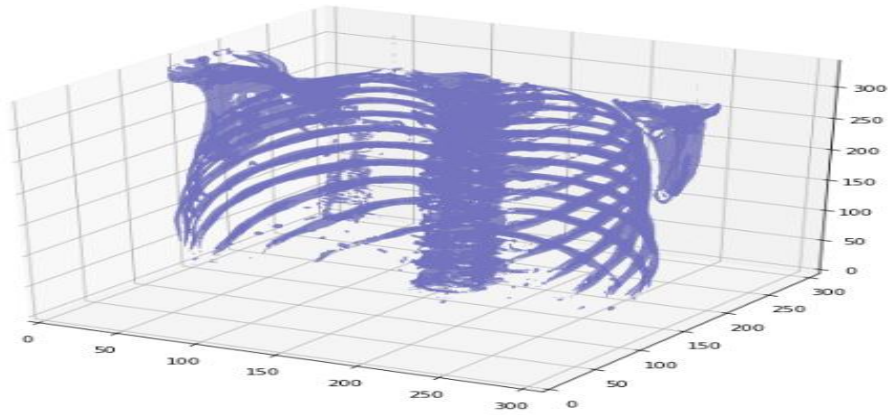


Fig 3.4 – 3D plotting of Bone Structure

Lung Segmentation: In order to reduce the problem space, we can segment the lungs and usually some tissue around it. Our values currently range from -1024 to 2000. Anything above 400 is not interesting to us as these are simply bones with different radio density. To remove unwanted tissues, we set a threshold limit. In this case - 400 HU to + 400 HU is a good threshold limit and the lower and higher values can be neglected.

3D plotting of Lungs:

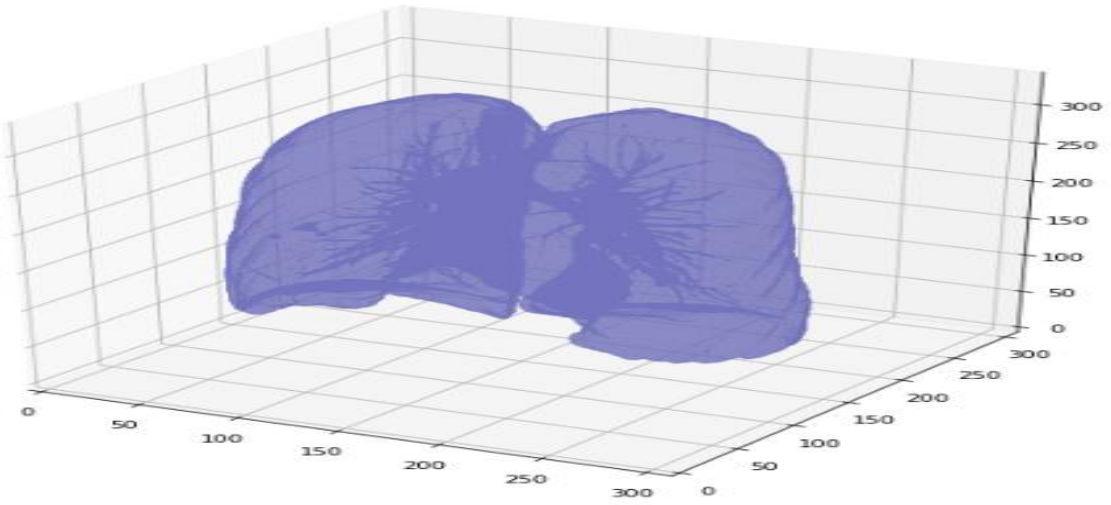


Fig 3.5 – 3D plotting of Lungs

3D plotting of segmented Lungs:

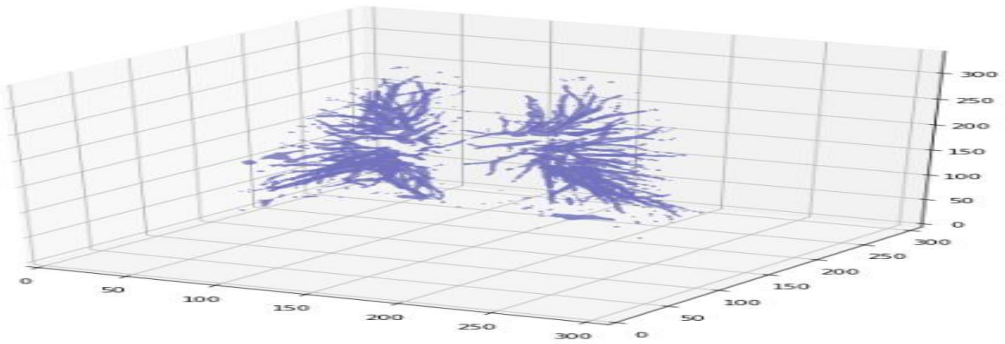


Fig 3.6 – 3D plotting of segmented Lungs

With these steps our images are ready for consumption by CNN.

RESULT ANALYSIS

4.1: Training Phase

In this experiment, Nuronics Labs has provided a total of 1368 patient's data, where we used the data generated from 1094 patients scans as the train data. The remaining 274 patient's scans are used as test data. This experiment was conducted on a desktop equipped with Intel i7 processor, 20GB memory and a GeForce GTX 1050 GPU.

4.2: Results

In the basic model created, to check the working nature of the model, we trained it with a data of 100 patients (training data – 80 patients, testing data – 20 patients) and the results are as follows:
Training Accuracy – 48.6%
Testing Accuracy – 49.8%

On the final working model, with the averaged method, we trained it with complete data and the results are as follows:

Training Accuracy – 71.2%
Testing Accuracy – 72.7%

On the final working model, with the image enhancement and lung segmentation method, we acquired better results than the averaged method and the results are as follows:

Training Accuracy – 76.2%
Testing Accuracy – 76.8%

Accuracy Table:

	Training Data	Testing Data	Training Accuracy	Testing Accuracy
Basic Model	80	20	48.6	49.8
Avg Method	1094	274	71.2	72.7
Lung Segmentation Method	1094	274	76.2	76.8

Table - 4.1

5.1 CONCLUSION AND FUTURE SCOPE

Doctor's working in this field are prone to observer fatigue from viewing so many CT scan images. Due to this, the risk of errors to be made by doctors while analysing these scans rise. For example, for every 200 – 300 scans only 2 – 3 scans would show cancer nodules depending on the stage of the patient. Therefore, by the usage of Deep Learning techniques in this field could gradually decrease the errors made.

In this study, various optimization algorithms have been evaluated to detect the tumor. Medical images often need pre-processing before being subjected to statistical analysis. We used CNN based procedures to detect nodules in CT scans of Lung. We study and compare the two methodologies we adopted, one with the averaging technique, and the other with image enhancement and lung segmentation techniques. We adopted Gabor filter as an image enhancing technique to enhance the edges of an image for better feature extraction. And with additional lung segmentation technique the results were further improved. With this filter, we got better results than the averaging method.

5.2: Future Scopes

The further enhancement of the proposed methods can be focused on the following ideas for better performance and efficient images. Removal of noise from the images can better the performance of a model.
Better learning techniques can be used to increase the performance and quality of the method.
Improving the quality of CT scans.
An improved segmentation technique on enhanced CT scans could deliver better results.
By the application of predefined Neural Networks on the model, we are likely to observe an improvement in the accuracy. The implementation of pre-nets on the model is termed as "Transfer Learning".

REFERENCES

Journal / Conference Papers

[1] Mokhled S. AL-TARAWNEH, "Lung Cancer Detection Using Image Processing Techniques", Leonardo Electronic Journal of Practices and Technologies, Issue 20, Jan–June 2012, p.147-158

[2] S Vishukumar K Patela, "Implementation of Medical Image Enhancement Using Gabor Filter", International Journal of Current Engineering and Technology, 2012

[3] S. Sasikala, M. Bharathi, B. R. Sowmiya, "Lung Cancer Detection and Classification Using Deep CNN", Lung Cancer Detection and Classification Using Deep CNN, Volume-8 Issue-2S December, 2018