AUTOMATED LUNG CANCER NODULE DETECTION

Monica Ramakrishnan, Surya Rajasekaran, Barsa Nayak, Akshay Bhagdikar

SANTA CLARA UNIVERSITY

1 PRE-INTRODUCTION

1.1 PREFAFACE

The purpose of this project is to develop a model that utilizes various concepts from image processing, data mining, and machine learning to detect lung cancer nodules amongst high risk patients. Various concepts of image processing were also utilized. This report has been made in fulfillment of the requirement for the subject: Pattern Recognition & Data Mining in June 2017 under the supervision of Dr. Ming-Hwa Wang

1.2 ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to Dr. Ming-Hwa Wang for providing us with an opportunity to explore our interests in data mining as well as image processing. Without his tremendous support, encouragement as well as valuable inputs, this project couldn't have materialized. The guidance and support received from all the members who contributed and who are contributing to this project was vital for our success.

1.3 TABLE OF CONTENTS

1.4 LIST OF TABLES & FIGURES

1.5 ABSTRACT:

Lung cancer is a disease of uncontrolled cell growth in tissues of the lung. Since lung cancer is one of the leading causes of death, early detection of malignant tumors is imperative for a successful recovery. In general, early stage lung cancer diagnosis techniques mainly utilize X-ray chest films, CT, MRI, etc. Computed tomography (CT) produces a series of cross-sectional images covering a part of the human body. For our case specifically, we will focus on the thoracic region. Visually identifying and examining these images for potential abnormalities is a challenging and time consuming task due to the large amount of information that needs to be processed, and the short amount of time given. The subject of medical image mining is currently an up and coming topic and shows a lot of research potential in the area of computational intelligence. By auto-analyzing a patient's records and images through data mining and image processing techniques, we would reduce the risk of human error in nodule detection. By applying a combination of techniques in data preprocessing, feature extraction, and classification, we ultimately seek to increase the accuracy rate of cancer detection, while simultaneously reducing the false positive diagnosis rate. In this project, we propose to use a deep artificial neural network architecture, which is a combination of CNN along with RNN for the fully-automated detection of pulmonary nodules in CT scans. The architecture of the VGG16 convolutional neural network is trained to distinguish pixels across images, and can be utilized in our case to extract nodule information. Our project will demonstrate that by leveraging these techniques, we substantially increase the sensitivity to detect pulmonary nodules, without inflating the false positive rate. Thus, from the available LIDC/IDRI dataset consisting of around 1500 CT scans, we have provided an innovative approach of implementing CNN using the pretrained VGG model for feature extraction and RNN for feature classification for identification of pulmonary nodules in lung cancer detection.

2 INTRODUCTION

2.1 OBJECTIVE:

The objective of this project is to improve the current cancer detection rate by reducing the false positives while maintaining a low false negative rate. The false positive detection of cancer is dangerous, as an erroneous diagnosis utilizes precious resources, causes unnecessary apprehension for the patient, and finally, poses a variety of legal threats to the doctors. Thus, to prevent this, our objective is to develop an algorithm that efficiently reduces the false positive rate, while maintaining the overall cancer detection accuracy.

2.2 WHAT IS THE PROBLEM:

Lung carcinoma, also known as lung cancer, is characterized by malignant tumors from when gene changes in the DNA of the cells mutate and promote unnatural growth. Lung cancer is the most common type of cancer with approximately 225K new cases in 2016 alone, which led to $12 billion in annual health care costs. The most common age at diagnosis is 70 years. Overall, the lung cancer survival rate in the United States is extremely low – only 17.4% of patients diagnosed with lung cancer survive five years post diagnosis. Further, uncontrolled cell growth can spread to surrounding areas or metastasize to other organs if it not detected early. Doctors currently use Low-Dose CT scans to help assess if a person is at risk of lung cancer, or even other pulmonary diseases.

Using a data set of thousands of high-resolution lung scans provided by the National Cancer Institute, we will develop a model that accurately determines whether lesions in the lungs of high risk patients are cancerous. Current models have extremely high false positive rates, which does not allow oncologists and radiologists to focus on patients that have an imminent cancer threat. Further, the high false positive rates lead to unnecessary patient anxiety, additional follow up imaging, and interventional treatments. Thus, by building a classification model that reduces the number of both false positives and false negatives, patients can have earlier access to life-saving interventions, as well as give doctors an opportunity to prioritize patient care.

2.3 PROJECT RELATION TO CLASS:

The approach we will take to model an accurate prediction system of lung cancer will utilize multiple techniques in data mining and pattern recognition. Classification problems are rooted in feature processing, clustering feature extraction, feature engineering, and machine learning models. First, by mining a large data set of CT scans, we utilize techniques in data mining and image processing that are crucial for accurate feature extraction. We also plan to utilize both unsupervised (clustering) and supervised (classification) models in order to extract characteristics of patients' lungs and finally classify patients as cancerous or not cancerous. By using these various techniques, we are combining widely used techniques in data mining and pattern recognition.

2.4 WHY OTHER SOLUTIONS ARE INADEQUATE:

Most of the standard mechanisms for classification go for either PCA or SVM or Random Forest and Xboost Classifier to classify the data. Other classifiers fail if they haven't taken the below scenarios into account:

1. Detecting the location of Pulmonary Nodule(s): The current approaches for detecting pulmonary nodules generally include manual intervention. When the location of the nodules is not provided in the data set, it becomes an extremely tedious task to manually parse the images to find potentially malignant nodules. This proves itself even more difficult if the images are rotated or twisted, leading to further erroneous processing.
2. Feature Extraction: The generation of a small number of features leads to a loss of data. Once the nodule position is determined, the nodule is then extracted from the entire image. Then, the features such as total area, average area, maximum area and average eccentricity, average equivalent diameter, standard equivalent diameter, weightedX, weightedY, number of nodes, and number of nodes per slice were calculated. With this approach, there is a lot of information lost.

2.5 WHY OUR APPROACH IS BETTER:

Our proposed approach has the following two steps:

1. Feature Extraction Phase: Unlike manual intervention, we make use of the image processing techniques to first highlight the lung region and then apply our state of the art pre trained Convolutional Neural Network model for feature extraction from the images. This reduces the human effort of the nodule position detection, and since our model is not restricted to the position of the nodule, it does not get affected if the image is rotated or twisted.
2. Classification Phase: Recurrent Neural Networks with multilevel perceptron will be used to classify the CT scans. Although random forest classification models typically require more data for a similar accuracy, they typically generate a robust model. Thus, we will use random forest to set a baseline accuracy for our analysis. On the other hand, deep learning is more favorable as complex problems such as image classification can be handled better, and this is the base of pulmonary nodule detection.

t

The benefits of using deep learning (Recurrent Neural Networks) are:

1. Automatic feature extraction without having to extract the nodule position information and other features.
2. In case of datasets which are complex 3D images, deep learning gives better classification results as compared to other methods.
3. Tree based models (the one that XGBoost is good at) solve tabular data very well but a deep network can capture things like image, audio and possibly text quite well by modeling the spatial temporal locality.
4. Neural network based deep learning is an accuracy-focused method whereas Xgboost is an interpretation-focused method.

2.6 STATEMENT OF PROBLEM

Although CT scans are established means for detecting pulmonary nodules, the small lesions in the lung still remain difficult to identify – especially when using a single detector CT scan. This poses itself as a challenge when attempting early detection of lung cancer. Since early detection is the key for a successful remission and recovery, the inability to manually see the small lesions further hinders the possibility of early detection. Current CT scanners produce up to 300 cross-sectional 2D images, each of which must be individually evaluated under a time constraint. Despite the diagnostic benefits provided by the CT imaging, the increased manual workload that is required to read 200-300 slices per exam leads to the increasing error rate of cancer detection. Given that nodules can appear in different positions, and depending on the patient, the process to detect lung cancer becomes extremely labor-intensive and manual. This manual process further increases the probability of human error – either the doctor detects cancer in a patient who is cancer free, or the doctor fails to detect the malignant nodule. Current studies have demonstrated that the rate of erroneous CT interpretation and analysis ranges from 7% - 15% when a radiologist performs more than 20 CT examinations per day (Bechtold, 1997). In order to address these issues, there has been a sudden increase of research and development of CAD (computer-aided-diagnosis) systems for high accuracy pulmonary nodule detection. Using the CT scans, the adoption of the CAD systems led to an improvement on the sensitivity of current detection algorithms – present day systems are successfully able to detect nodules with a 3mm diameter. Thus, several approaches are being proposed to overcome the challenges to detect the pulmonary nodules and thus maximize the chances of the survival of the patients.

2.7 SCOPE OF INVESTIGATION

For the purposes of this project, we will specifically focus on lesions in the lungs of high risk patients. The general approach can potentially be applied to different organs and in patients of various demographics, but for this investigation, we will limit the scope of our investigation.

3 THEORETICAL BASES AND LITERATURE REVIEW

3.1 DEFINITION OF THE PROBLEM:

Since lung cancer is one of the leading causes of death in the United States and given that early detection increases the probability of a successful remission, our problem statement revolves around creating an automated, high accuracy model for nodule detection. Current methodologies place a heavy focus on reducing false negative rates, but at the expense of significantly over predicting the cancerous class. Further, our research has suggested that current approaches to nodule detection all require some level of manual image processing. That is, the images are parsed by hand for the coordinates of the nodule. Thus, the solution will focus on 1) automating nodule detection, and 2) reducing the false positive rate of cancer detection while maintaining a good false negative rate. The availability of large data given by the National Cancer Association provides an opportunity for further research in data mining. The important problem in this area is to make an efficient detection algorithm to aid with early detection.

3.2 THEORETICAL BACKGROUND OF THE PROBLEM:

With the advancement of Technology and Computer Aided Diagnosis (CAD), scientists have encouraged a lot of automated systems to address the issue of reducing false positive while estimating the presence of pulmonary nodules in the CT scans of the patients. Thus, today we have a surplus of data pertaining to the CT scan patients. From here, we have the opportunity to use current topics in image processing, data mining, and machine learning to identify hidden patterns in nodule size, location, structure, etc. and construct a model to increase the probability of malignant tumor detection.

With the advent of the pattern recognition and machine learning, data scientists have proposed many approaches which were robust in finding the hidden patterns and reducing the false positives. With this, as the data for the patients detected with lung cancer increased, the CT scans tend to differ more significantly from each other, across patients. Consequently, the deep learning has come into picture for the complex image classification in order to ensure that outliers and anomalies are properly handled in the model and thus, reducing the false positive rate for malignant pulmonary nodule detection.

3.3 RELATED RESEARCH TO SOLVE THE PROBLEM:

There has been a lot of research in recent times on the development of computer-aided diagnosis (CAD) systems for pulmonary nodule detection using CT imaging. Advancements in image processing field has increased the accuracy in the prediction of cancer from CT scans. There are plenty of research papers which discuss the various methods and outputs. A few examples include: ‘Recurrent Convolutional Networks for Pulmonary Nodule Detection in CT Imaging’, ‘Combining deep neural network and traditional image features to improve survival prediction accuracy for lung cancer patients from diagnostic CT’, ‘Computerized Detection of Lung Tumors in PET/CT Images’, and ‘Lung cancer classification using neural networks for CT images’.

3.4 ADVANTAGE/DISADVANTAGE OF THOSE RESEARCH

Advantages

The current research is extremely helpful to many students who would like to implement the findings into their projects and as well as to do further research. The advantage of these research findings is that it also helps in improving the detection of cancer at an early stage and to reduce the detection error.

Disadvantages

The disadvantages are that usually the data training set is very huge which increases the time taken to train. In order to obtain a beyond-average accuracy, we are requird to use an extremely large data set in order to train the model. Although it is time consuming for the code to train the network, it is a one-time process that can be redone if there are new types of cancers that need to be trained.

3.5 SOLUTION TO PROBLEM

In order to detect cancerous tissue in the lungs with a relatively high accuracy, our solution will include the following steps:

1. Image Processing
2. Feature Extraction & Engineering
3. Clustering
4. Classification Model

This hybrid approach allows us to combine the most efficient individual methodologies into an end to end model. We believe that this combination will extract the advantages from each approach, and ultimately provide more conclusive results. Our goal is to build an automated nodule detection model with high accuracy in order to aid the detection of lung cancer in patients.

3.6 WHERE SOLUTION IS DIFFERENT FROM OTHERS

Our solution will utilize a combination of machine learning techniques and place a unique focus on minimizing the false positive rate. While most models place the primary focus on total cancerous lungs predicted, we will place our focus on reducing the false positive rate, in order to make our results more reliable. Further, most of the techniques being used for lung cancer detection pre-process the CT scan manually, which is the most critical step for high accuracy detection. Current solutions require manual feature extraction, which increases the probability of human error. On the contrary, our method will input the entire CT scan as DICOM image into the classification system and thus, we eliminate the need for a manual preprocess.

3.7 WHY SOLUTION IS AN IMPROVEMENT ON CURRENT METHODS

Our solution is aimed at reducing the number of false positives in the classification model. This can be achieved by a combination of feature engineering (i.e. selecting features and characteristics that are highly correlated with lung cancer) as well as training a classification model using bootstrapping and various other resampling techniques given our unbalanced data set. By doing so, our data set will even out the distribution between the classes of patients. Creating a weighting function for rare class (cancerous scans) forces the model to not over fit to the data, which ultimately results in a minimal number of false positives.

Further, as previously mentioned, for any job which is done in a pattern, machines have a much larger capacity and increased efficiency for image processing. By eliminating the need for human interaction, we have proposed an automated tool which will input raw CT scans, perform the required amount of image processing, model training, and final classification.

3.8 TERMINOLOGY

This section highlights the common terminology used in lung cancer and image processing.

Terminology	Definition
CT	Computerized Topology uses computer-processed combination of X-Ray images taken at different angles to produce the scan
Pulmonary Nodule	Mass in the lung that usually represents cancerous lesions
DICOM	Digital Imaging and Communications in Medicine
Instance Number	Identifies the sequence of images in a DICOM
Slice Thickness	Thickness of slice depends on thickness of CT detection machine
Houndsfield Units	Quantitative Scale used to measure density of substances found in body
ROI	Region of Interest

4 HYPOTHESIS

4.1 MULTIPLE HYPOTHESIS

Hypothesis: By using the solution outlined above, our model will reduce the number of false positives, while maintaining a good accuracy rate.

4.2 POSITIVE/NEGATIVE HYPOTHESIS:

Positive Hypothesis: When extracting and quantifying feature for ROI, the feature structure design is irrational, hence 3D features are taken into account by recombining the slices while extracting the features to get more accuracy.

Negative: The pulmonary nodule can be of any shape and could merge with the blood vessel, which could cause issues when detecting the nodule accurately. Since the accuracy is not 100%, there leaves room for error. However, this is still better than a manual detection done by humans.

5 METHODOLOGY

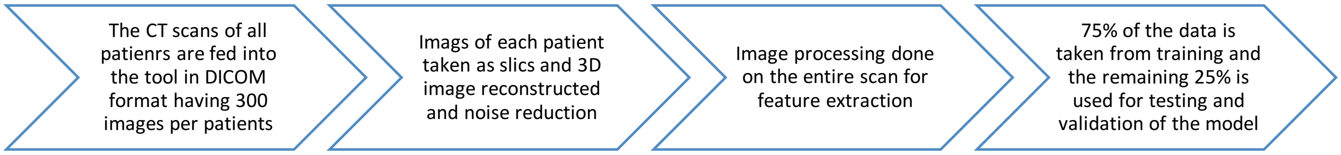
5.1 HOW TO GENERATE/COLLECT INPUT DATA:

Our input data is taken from Kaggle, which has the data of 1,000 high risk patients. The data set is approximately 130 GB – which required a significant amount of computing power to process. In this dataset, we are given over a thousand low-dose CT images from high-risk patients in DICOM format. Each DICOM image contains a series of 2D gray-scale images that contain multiple axial slices of the chest cavity – that is, there are approximately 300 images per patient. Each image has a variable number of 2D slices, which can vary based on the type of machine used in the scan. The DICOM files have a header that contains the necessary information about the patient id, as well as other scan parameters such as the slice thickness. The data set also included a csv containing the classification information as cancerous or non-cancerous per patient id. The images in our data set vary in quality, depending on when the scan was taken. For example, older scans were imaged with less sophisticated equipment and thus, have a lower resolution than more recent scans. Thus to summarize it has the following info – around 300 images which represent the slices of the thoracic region CT scan in slices in a DICOM format:

1. Patient ID, name, date of birth and other metadata of the patient and image information
2. CSV file mapping to each patient id mentioning of the person has cancer or not

5.2 HOW TO SOLVE THE PROBLEM:

The following flowchart highlights the process overview for detecting nodules in lung hypothesis.


The CT scans of all patienrs are fed into the tool in DICOM format having 300 images per patients
Imags of each patient taken as slics and 3D image reconstructed and noise reduction
Image processing done on the entire scan for feature extraction
75% of the data is taken from training and the remaining 25% is used for testing and validation of the model

The steps are outlined as follows:

1. Data Input: Download data for around 1,500 patients from Kaggle – gives 2D slices of CT scan images for each patient, which are in the DICOM format. There are approximately 300 slices per patient.
2. Image Processing: Reconstruct the 3D structure of lungs, remove the noise from the image, and make the data set of all the patients' uniform and thus, eligible for feature extraction, which includes sorting, morphological, dilation, erosion, segmentation, masking etc.
3. Classification: Recurrent neural Network with MLP as its core is used for the classification. MLP, or multi-level perceptron, has a memory element, LSTM (long short term memory) associated with it. This further enhances the classification by emphasizing significant features while de-emphasizing asspects which are unimportant.
4. Validation: 75% of training data is given to train the model and 25% is used for testing purposes.

5.3 ALGORITHM DESIGN:

As a part of our algorithm design, we will implement two different approaches in order to determine the model with the highest prediction accuracy for detecting lung cancer. Our image processing steps will remain constant throughout the approaches. The difference will lie within the feature extraction and classification techniques. As our cursory analysis, we will design a relatively simple method of feature extraction and classification.

1. Image Processing: – given the sets of CT scans, our algorithm will construct the 3D lung scan, and extract only the lung region. This is especially important that edge detection is done with high accuracy to minimize the error rate of our model. We also utilize segmentation and machine learning techniques to pre-process the image, including auto-detecting boundaries that surround the volume of interest. The images represent the 2D slices of the patient's thoracic region in DICOM format. Thus before processing, it is very important to reconstruct the image in 3D form as follows:

    a. **Arrange the Slices**: Arrange the slices in the non-decreasing order of their Instance Number to make sure the CT scan can be reconstructed. As mentioned, the CT scan is in the form of numerous 2D images (slices) which need to be rearranged in numerical order to be able to recreate the 3D structure. This information lies in the InstanceNumber of the slices, which gives the order in which it has to appear when generating 3D view. An example of three slices in a particular DICOM image can be see below:

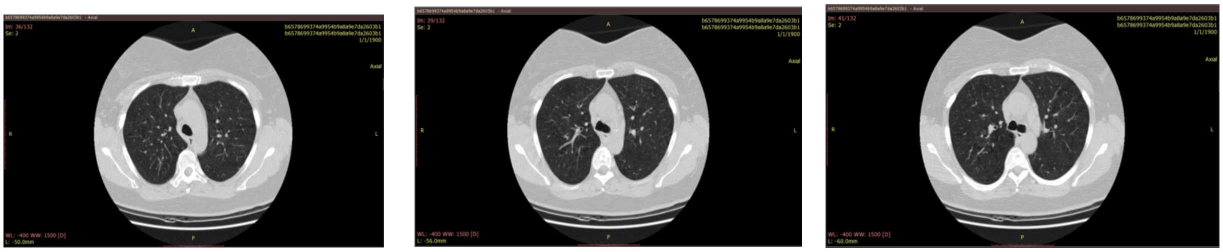


Figure 1: Sample slices of DICOM image

    b. **Calculate Slice Thickness:** Once the image is formed, we have to ensure that slice thickness is taken into account. The slice thickness is calculated based on the metadata present in the DICOM format. Since the input data has been taken from many sources, its thickness varies for different scanners. Thus, we have two formats of metadata to calculate thickness where one can consider either the ImagePositionPatient attribute or SliceLocation attribute of the DICOM format based on which is available.

    c. **Convert the Image into HU:** The image is then converted to HU (Hounsfield units) to isolate the lung region. Since we have an image of the entire thoracic region, it is important to distinguish lung region from non-lung region. The first step of the conversion is setting these values to 0, which corresponds to the HU unit of air. Then, the HU units of each region is obtained by multiplying the rescale slope and adding the intercept (which are conveniently stored in the metadata of the scans). For reference, the table below contains the average range HUs per substance:

Substance	HU
Air	-1000
Lung	-500
Fat	-100 to -50
Blood	+30 to +45
Water	0
Muscle	+10 to +40

Figure 2: Houndsfield Units for Various Substances

When adding a colored filter to a particular slice of the DICOM images, we are better able to visualize the and outline various substances in the lung:

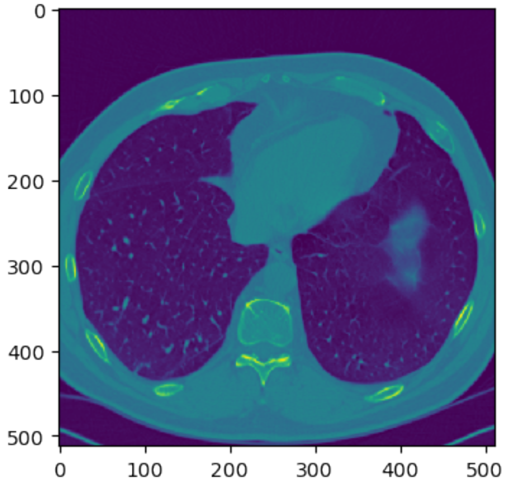


Figure 3: CT Scan Slice Colored

d. **Resampling:** Then, we apply resampling to obtain a homogeneous dataset. Uniformity is of prime concern in image processing, so we do not bias any set of images. A common method of dealing with problem is to resample the full dataset to a particular isotropic resolution. Thus, we go for a resampling technique to ensure that the spacing of the images is uniform for the entire input dataset.

e. **Interpolation:** The process of zoom interpolation using spline methodology is done to smooth the image and reduce visual distortion.

f. **Thresholding** – While there exist various kinds of thresholding techniques, for the purposes of our project, we will utilize a clustering base method, where the gray-level samples are clustered in two parts as background and foreground (object). Thus, we apply the k-means algorithm in order to separate the lung region from the background. An example of a slice post thresholding can be found below. Note that we have successfully isolated the lung region, but can still visibly see the noise in the image.

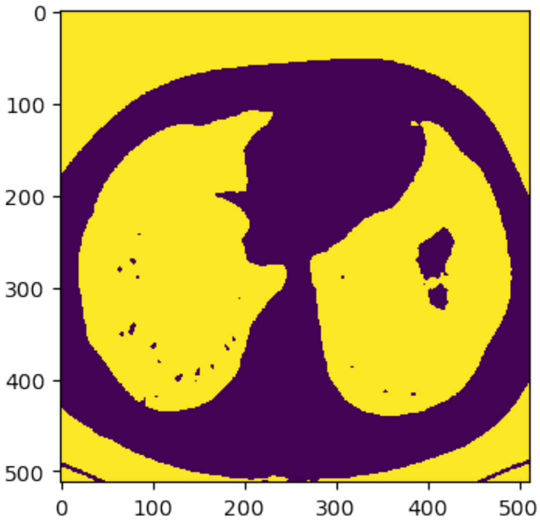


Figure 4: After thresholding

g. **Erosion and Dilation** – We then apply erosion – a technique for shrinking the image in order to ensure the unwanted noise introduces further diminishes. Then, when we dilate the image, we regain the original size of the image, keeping the actual size of lungs and the nodules intact. These two processes overall reduce the noise in the image. Examples of noise in CT scans include veins, capillaries, etc.

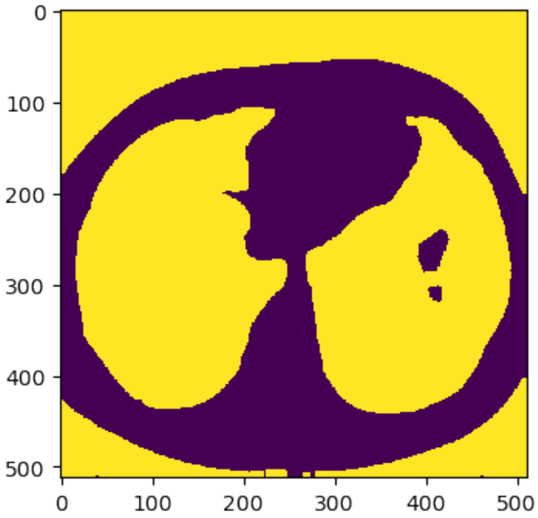
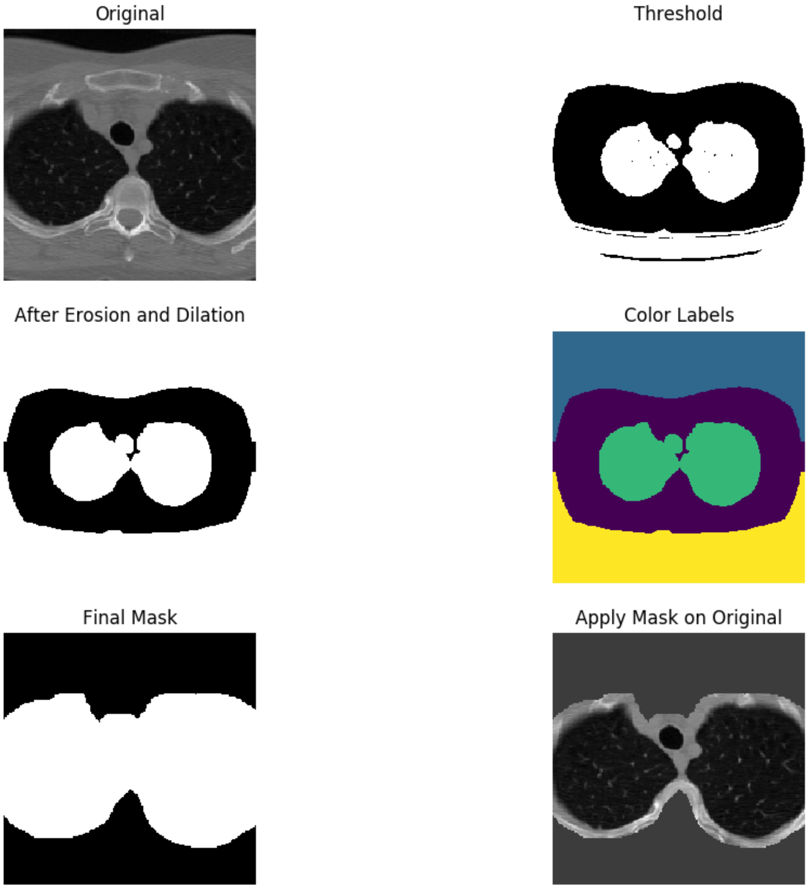


Figure 5: After erosion and dilation

h. **Segmentation** - Masking of the image is done by highlighting ROI (Region of Interest). After getting rid of noise and reducing false positives, the image is segmented. Thus, a mask is made by outlining the edges of the lung (distinguishing the lung from the muscular surroundings) as well as the nodules contained within the organ. We then apply the mask to isolate the ROI.

The below image demonstrates an overview of the entire process described above:



Original

Threshold

After Erosion and Dilation

Color Labels

Final Mask

Apply Mask on Original

2. Feature Extraction & Engineering – Using the 3D scan of the lungs allows us to retain as much of the information as possible. As explained, we then convert the CT scans (in Hounsfield Units) to pixel, allowing us to input a pixel array into our classifier as well as the features that we extract. Here, we will have utilized a two tiered approach. The first approach will focus on a simplistic method of feature extraction, then followed by our second approach which will place a heavy emphasis on deep learning. By doing so, we allow the simplistic model to set a baseline for accuracy, and we are then able to fully understand the impact that deep learning has on our precision

   Approach 1:

   a) For every patient we divided the set of images into 4 sections. For example, if the patient has 200 images, then divide the set in a section of 50 images each. By doing this we essentially divide the lung into 4 regions – top, two middle, and a bottom.
   b) After preprocessing the image, we get an image with binary pixel values as 1 or 0's. We read every image array in a section, then sum those arrays and finally average out and store these averaged values in a new array. This new array is a mean representation of that specific

section of lung area. By doing so, we average out the dark areas per region, with the hypothesis that the higher the average value, the larger the probability for nodule detection.

c) Similarly, we find averaged array for every section. The values stored in this array ranges from 0 to 1. We round off these values and finally get four arrays with 0s and 1s for every patient.
d) We calculate the number of 1s in every array. So finally, for every patient we get four values – number of 1s in every section.
e) These values are then used as the feature input for our classification model.

Approach 2:

First, feature extraction is largely dependent on image identification. Thus any customized CNN model first have to fed with lot of training data unlike human beings for it to understand which category the image belongs to. In the process it does so by updating its weights i.e. applying appropriate filters, and thus emphasizing on the feature to be considered for recognition. This process is time consuming as with each input, it updates the weights and learns to identify the feature. The process outlined below was created by Visual Geometry Group (VGG) who have trained a CNN model with at least 22,000 categories to have the best possible weights for feature extraction. For our case, we can use the model directly for feature extraction without having to manually update the weights. With 16 layers, VGG16 is an extremely efficient image processing algorithm. Using a pre-trained model allows us to utilize the most up to date technologies in machine learning and data mining. By running this CNN on the data, we are able to auto detect features in each DICOM image. Thus, our output from the CNN is a data frame with approximately 40,000 processed features per patient. The architecture for VGG is below:

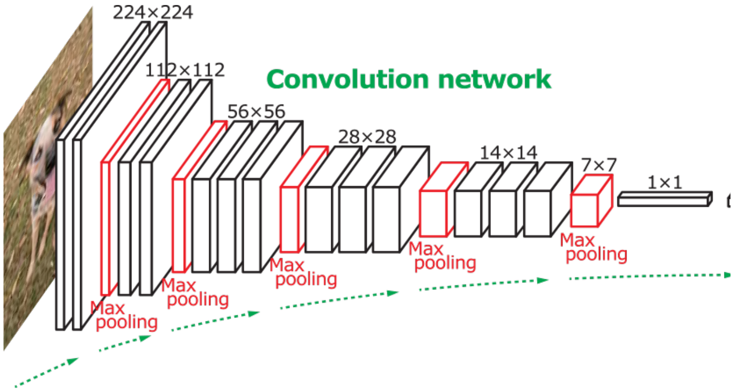


Figure 6: VGG16 Architecture

This model is available for both the Theano and TensorFlow backend, and can be built both with "channels_first" data format (channels, height, width) or "channels_last" data format (height, width, channels).

3. Classification – Finally, we can input our data into a classification model. The data frame will have each individual patient as a row, and the different features as a column. The final column in the data frame will be a binary value: 1 if the patient has cancer, and 0 if the patient does not have cancer. For validation purposes, we will use 75% percent of our original data for training the classifier, and 25% for validation. From previous discussion, we mentioned we are taking two different tracks in order to further analyze the accuracy differences between using a simple vs complex model. Thus, our classification steps for both approaches are as follows:

   Approach 1:

   Once our images are processed and the pixel array is extracted from each CT scan from approach 1 mentioned above, we can feed the data frame into a clustering algorithm to group similar patients together. As such, using the four features previously generated, we utilized the K-Means clustering algorithm in order to partition the ~1,500 patients into k=2 clusters – positive or negative for cancer – where each patient will belong to the cluster with the nearest mean. By partitioning the data space as such, we are able to gain a preliminary understanding of the distinguishing characteristics between the two groups. The K-Means algorithm partitioned the data into the two groups with relatively high accuracy.

   We were then able to add the K-Means clustering classification as an input parameter to our next classification method: Random Forest. Since Random Forests are known to perform well on various tasks, including unscaled data and variable selection, we decided to employ this approach in order to understand the effectiveness of our feature extraction methodology. If our Random Forest returns a low accuracy, we can conclude that our feature extraction is not adequate, setting a basis for our deep learning approach.

   Approach 2:

   The classification of the data set produced by our second approach using convolution neural networks is done by a multilevel perceptron (MLP) which follows the structure in the image below.

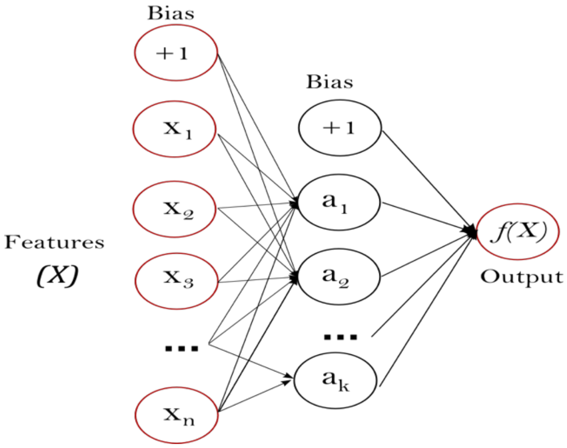


Figure 7: MLP Process

Using a multilevel perceptron has various advantages, including the capability to learn both linear and non-linear models, and most importantly, the capability to learn models in real-time (on-line learning) using partial fit.

The MLP then follows the steps highlighted below:

- Checks if there a pre trained model to load it before training.
- Sequential model used for training the model.
- Model uses relu activation with various activation layers
- Dropout after every stage to prevent overfitting
- Model is compiled with ‘binary_crossentropy’ - loss which is the best for binary classification, optimizer – RMSPROP being used
- Checkpointer - used to save the best model based on accuracy metrics
- After training, evaluation is done to show the efficiency of the trained model.

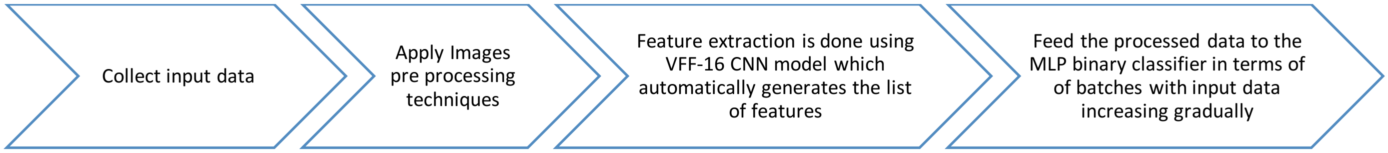
5.4 LANGUAGES AND TOOLS USED:

Our project will utilize libraries from both R and python. We will primarily use python to do the majority of our data extraction and image processing, while the classification models will be built and analyzed using both R and python.

5.5 HOW TO GENERATE OUTPUT:

The program reads the input image (in DICOM) and applies the image processing functions, followed by the prediction algorithm in order to generate binary classification of lung cancer detection.

The flow chart below describes the process taken to generate the prediction for lung cancer detection:


Collect input data
Apply Images pre processing techniques
Feature extraction is done using VFF-16 CNN model which automatically generates the list of features
Feed the processed data to the MLP binary classifier in terms of of batches with input data increasing gradually

5.6 HOW TO TEST AGAINST HYPOTHESIS:

Once we acquire a dataset, we intend to divide it into two subsets:

> **Training**: Here, we have taken 75% of the entire dataset to train the predictive model
>
> **Test**: Here, we take the remaining 25% of the data to assess the likely future performance of the model. If our model fit based on the training data set is a much better fit than the test set, we will likely have an overfitting problem.

Since our focus is on the reduction of false positives, and we wish to prevent overfitting, we have accounted for both these cases by using a dropout layer in the neural network. The purpose of such a layer is to prevent regularization – which may have falsely produced a smooth lose curve, but over fit the data, ultimately leading to low convergence.

6 IMPLEMENTATION:

6.1 CODE

Image Processing & Extraction:

The following python scripts have been created in order to process the DICOM images as well as perform the feature extraction:

1. settings.py
2. common.py
3. preprocess_step_1.py
4. preprocess_step_2.py

Classification

The following python scripts have been created in order to apply the classification model to our processed data set:

1. mlp_binary_classifcation.py

Snippets of code:

Given that the code of the project is too vast, as it contains various aspects of image processing as well as feature extraction and classification, we have provided the important functions below. The descriptions for these functions can be found in section 6.2: the design document.

preprocess_step_1.py

```
def load_scan(src_dir):
    slices = [dicom.read_file(src_dir + '/' + s) for s in os.listdir(src_dir)]
    slices.sort(key=lambda x: int(x.InstanceNumber))
    try:
        slice_thickness = np.abs(slices[0].ImagePositionPatient[2] -
slices[1].ImagePositionPatient[2])
    except:
        slice_thickness = np.abs(slices[0].SliceLocation - slices[1].SliceLocation)

    for s in slices:
        s.SliceThickness = slice_thickness

    return slices
```

```
def resample(image, scan, new_spacing=[1, 1, 1]):
    # Determine current pixel spacing
    spacing = map(float, ([scan[0].SliceThickness] + scan[0].PixelSpacing))
    #print spacing
    spacing = np.array(list(spacing))
    resize_factor = spacing / new_spacing
    new_real_shape = image.shape * resize_factor
    new_shape = np.round(new_real_shape)
    real_resize_factor = new_shape / image.shape
    new_spacing = spacing / real_resize_factor
    image = scipy.ndimage.interpolation.zoom(image, real_resize_factor)
    return image, new_spacing

def make_lungmask(img, display=False):
    row_size = img.shape[0]
    col_size = img.shape[1]
    mean = np.mean(img)
    std = np.std(img)
    img = img - mean
    if std > 0:
        img = img / std
    # Find the average pixel value near the lungs to renormalize washed out images
    middle = img[int(col_size / 5):int(col_size / 5*4),
              int(row_size / 5):int(row_size / 5 * 4)]
    mean = np.mean(middle)
    max = np.max(img)
    min = np.min(img)
    #To improve threshold finding,move underflow and overflow to pixel spectrum
    img[img == max] = mean
    img[img == min] = mean
    # Use Kmeans to separate foreground(soft tissue/bone) and background(lung/air)
    kmeans = KMeans(n_clusters=2).fit(np.reshape(middle,
              [np.prod(middle.shape), 1]))
    centers = sorted(kmeans.cluster_centers_.flatten())
    threshold = np.mean(centers)
    thresh_img = np.where(img < threshold, 1.0, 0.0)  # threshold the image
# First erode away the finer elements, then dilate to include some of the       pixels
surrounding the lung as we don't want to accidentally clip the lung.
    eroded = morphology.erosion(thresh_img, np.ones([3, 3]))
    dilation = morphology.dilation(eroded, np.ones([8, 8]))
    labels = measure.label(dilation)  # Different labels are displayed in different colors
    label_vals = np.unique(labels)
    regions = measure.regionprops(labels)
    good_labels = []
    for prop in regions:
        B = prop.bbox
        if B[2] - B[0] < row_size / 10 * 9 and B[3] - B[1] < col_size / 10 * 9 and B[0] >
row_size / 5 and B[
            2] < col_size / 5 * 4:
            good_labels.append(prop.label)
    mask = np.ndarray([row_size, col_size], dtype=np.int8)
    mask[:] = 0
```

```
  #  After just the lungs are left, we do another large dilation
  #  in order to fill in and out the lung mask
  for N in good_labels:
    mask = mask + np.where(labels == N, 1, 0)
  mask = morphology.dilation(mask, np.ones([10, 10]))  # one last dilation

  mask = mask[int(col_size / 5):int(col_size / 5 * 4), int(row_size / 5):int(row_size /
5 * 4)]
  img = img[int(col_size / 5):int(col_size / 5 * 4), int(row_size / 5):int(row_size / 5
* 4)]
  masked_lung = mask * img

  if (display):
    fig, ax = plt.subplots(3, 2, figsize=[12, 12])
    ax[0, 0].set_title("Original")
    ax[0, 0].imshow(img, cmap='gray')
    ax[0, 0].axis('off')
    ax[0, 1].set_title("Threshold")
    ax[0, 1].imshow(thresh_img, cmap='gray')
    ax[0, 1].axis('off')
    ax[1, 0].set_title("After Erosion and Dilation")
    ax[1, 0].imshow(dilation, cmap='gray')
    ax[1, 0].axis('off')
    ax[1, 1].set_title("Color Labels")
    ax[1, 1].imshow(labels)
    ax[1, 1].axis('off')
    ax[2, 0].set_title("Final Mask")
    ax[2, 0].imshow(mask, cmap='gray')
    ax[2, 0].axis('off')
    ax[2, 1].set_title("Apply Mask on Original")
    ax[2, 1].imshow(mask * img, cmap='gray')
    ax[2, 1].axis('off')
    plt.show()

  masked_lung = mask * img
  return masked_lung
```

preprocess_step_2.py

```
def dump_features(patient_id):
    patient_id = patient_id[:-2]
    try:
        global COUNT
        COUNT = COUNT + 1
        print("{}, {}".format(patient_id, COUNT))
        model = applications.VGG16(include_top=False, weights='imagenet',
input_shape=(settings.ct_width, settings.ct_height, 3))
        features_train = model.predict_generator(generator(patient_id),
steps=settings.ct_depth/3)
        features_train = np.reshape(features_train, (1, -1))
        if settings.is_csv:
            np.savetxt("{}/{}.csv".format(settings.preprocess_step2_csv, patient_id),
features_train, fmt ='%.2f', delimiter=',')
        else:
            pickle.dump(features_train,
                    open("{}/{}.b".format(settings.preprocess_step2, patient_id), 'wb'))
    except Exception as e:
        print e
```

mlp_binary_classifcation.py

```
def create_model():
    model = Sequential()
    model.add(Dense(128, input_dim=input_dim, activation='relu',
kernel_initializer="normal",
                    #kernel_regularizer=regularizers.l1(),
                    # activity_regularizer=regularizers.l1(0.0001)
                    ))
    model.add(Dropout(0.3))
    model.add(Dense(64, activation='relu', kernel_initializer="normal"))
    model.add(Dropout(0.3))
    model.add(Dense(32, activation='relu', kernel_initializer="normal"))
    model.add(Dropout(0.3))
    model.add(Dense(16, activation='relu', kernel_initializer="normal"))
    model.add(Dropout(0.3))
    model.add(Dense(8, activation='relu', kernel_initializer="normal"))
    model.add(Dropout(0.3))
    model.add(Dense(1, activation='sigmoid'))

    sgd = SGD(lr=0.01, decay=1e-2, momentum=0.9)

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

```
def train_model():
    model = get_model()
    patient_ids = common.get_patient_ids()
    train_pct = 0.7
    steps_per_epoch = int(len(patient_ids)*train_pct/settings.batch_size)

    validation_steps = max(1, steps_per_epoch * (1 - train_pct))
    checkpointer = ModelCheckpoint(filepath=settings.model_filename, monitor='acc', verbose=1, save_best_only=True)
    early_stopping = EarlyStopping(monitor='acc', min_delta=0.001, patience=10, verbose=1, mode='auto')
    history = model.fit_generator(
        common.generate_arrays_from_file(input_shape=(input_dim,), train_pct=train_pct),
        steps_per_epoch=steps_per_epoch,
        epochs=settings.epochs,
        validation_data=common.generate_arrays_from_file(input_shape=(input_dim,), train_pct=train_pct, is_test=True),
        validation_steps=validation_steps,
        #validation_data=read_validation_data(30, input_shape=(input_dim,)),
        workers=4,
        pickle_safe=False,
        callbacks=[checkpointer])


    score = model.evaluate_generator(
        common.generate_arrays_from_file(input_shape=(input_dim,),  train_pct=0.80, is_test=True),
        steps=1)
    print score
    return model, history
    #score = model.evaluate(x_test, y_test)
    #print score

model, history = train_model()
model.save(settings.model_filename)
pickle.dump(history.history, open(settings.model_dump_dir + "/cnn_history.b", "wb"))
#plot_model(model, to_file=settings.output_dir + "/" + "mip_model.png")
common.plot_history(history)
```

6.2 DESIGN DOCUMENT:

Documentation of Functions Used:

Common.py

Method	Description
get_patient_ids()	This function fetches the patient IDs for all the patients
threadsafe_generator(f)	This is a decorator that takes a generator function and makes it thread safe
generate_arrays_from_file(input_shape, train_pct=0.9, is_test=False)	Generates an array from values in the file
plot_history(history)	This function is used to plot the training accuracy and the test accuracy

preprocess_step_1.py

Method	Description
load_scan(src_dir)	This function loops over the image files and stores everything into a list.
get_pixels_hu(scans)	This function is used for segregating the lung part by converting pixels to HU units
sample_stack(stack, rows=6, cols=6, start_with=10, show_every=3)	Used sample the stack of 2D images and display them
resample(image, scan, new_spacing=[1, 1, 1])	Resampling is done by this function to make sure that the distance between the slices is uniform
reshape_ct(image, new_shape=[settings.ct_depth, settings.ct_width, settings.ct_height])	Reshape the 2D images as per the desired shape for the next input
make_lungmask(img, display=False)	Function to standardize the pixel values

preprocess(data_path)	This function processes all the above functions defined
dump_preprocess_data(patient_id)	Used to save the preprocessed image in the form of binary file/csv to be used by the feature extraction model
preprocess_all():	Function used to iterate over every patient and preprocess each slice per patient to make sure it is ready for feature extraction

Pre_process_step_2.py

Method	Description
read_preprocess_1(patient_id):	This function is used to load the preprocessed images
generator(patient_id):	This function takes the preprocessed image and generates new images with dimension, which can be fed in to #VGG-16 model.
dump_features(patient_id):	The features extracted using the CNN model (VGG-16) are saved to be fed to the classifier in the next stage in this function
preprocess2_all():	Function used to extract the features for each of the scans by VGG 16 model

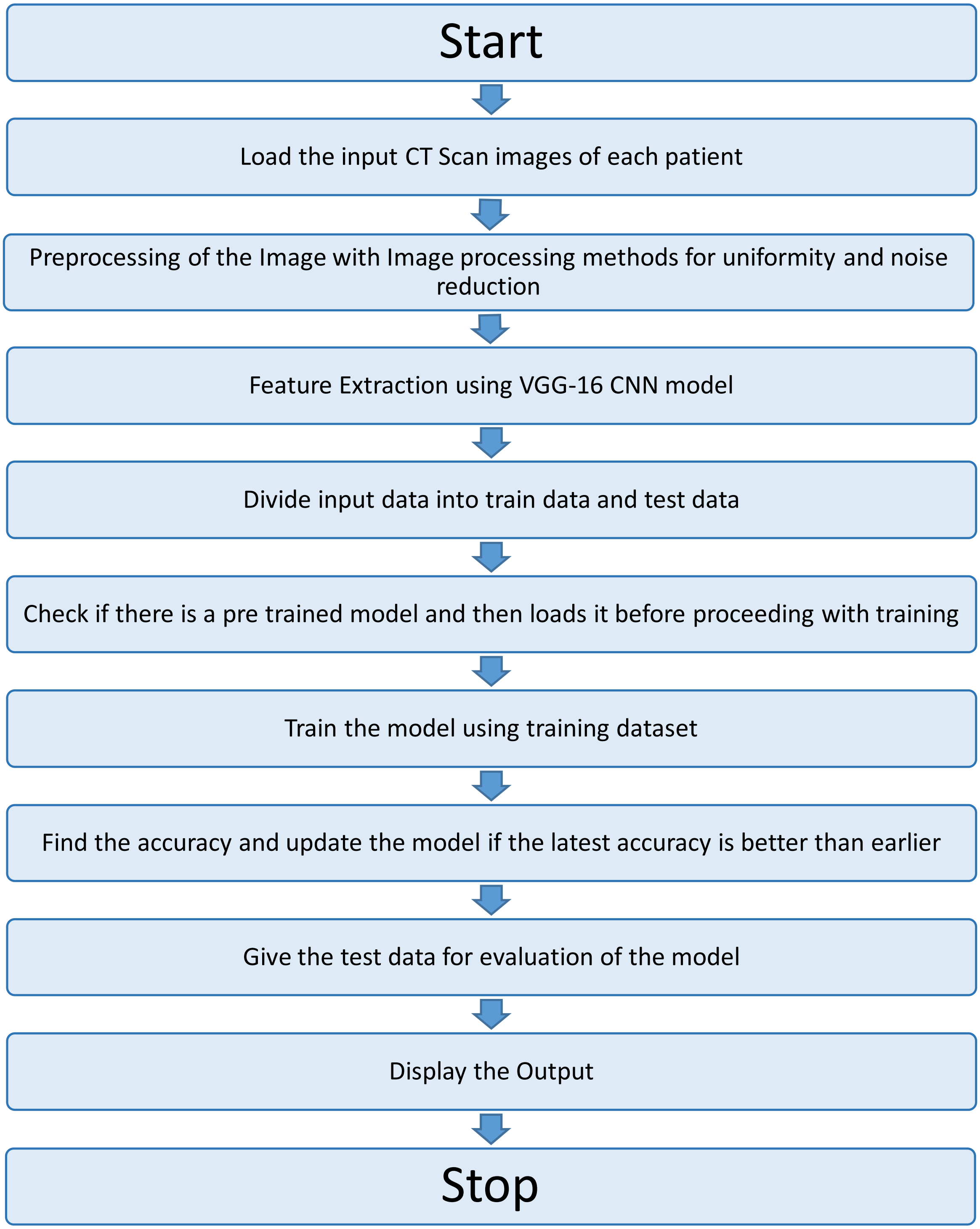
Mlp_binary_clasification.py

Method	**Description**
create_model ()	This function is used to create a sequential model that is used for simple binary classification.
get_model():	Function used to load a pre-existing trained model

train_model():	This function get a pre trained model if present or creates a training model and then trains it in epochs (as per the number of epochs mentioned)
evaluate_model():	Once the model is trained, this function is used to evaluate the test data

6.3 DESIGN DOCUMENT AND FLOWCHART

The following section outlines the flowchart of our analysis.


Start
Load the input CT Scan images of each patient
Preprocessing of the Image with Image processing methods for uniformity and noise reduction
Feature Extraction using VGG-16 CNN model
Divide input data into train data and test data
Check if there is a pre trained model and then loads it before proceeding with training
Train the model using training dataset
Find the accuracy and update the model if the latest accuracy is better than earlier
Give the test data for evaluation of the model
Display the Output
Stop

7 DATA ANALYSIS AND DISCUSSION

7.1 OUTPUT GENERATION:

The program reads the input of a CT scan in the form of a DICOM image per patient, and applies the prediction algorithm to it to generate the output. The images are 2D slices of the CT scan. We then train the model and analyze the accuracy against the actual result for each patient (i.e. cancerous or non-cancerous). We then find the difference of the outputs and update the weights by -30% to avoid overfitting. The final output of our algorithm is a classification of each patient as either positive or negative for cancer. This output is generated for all selected patients.

7.2 OUTPUT ANALYSIS:

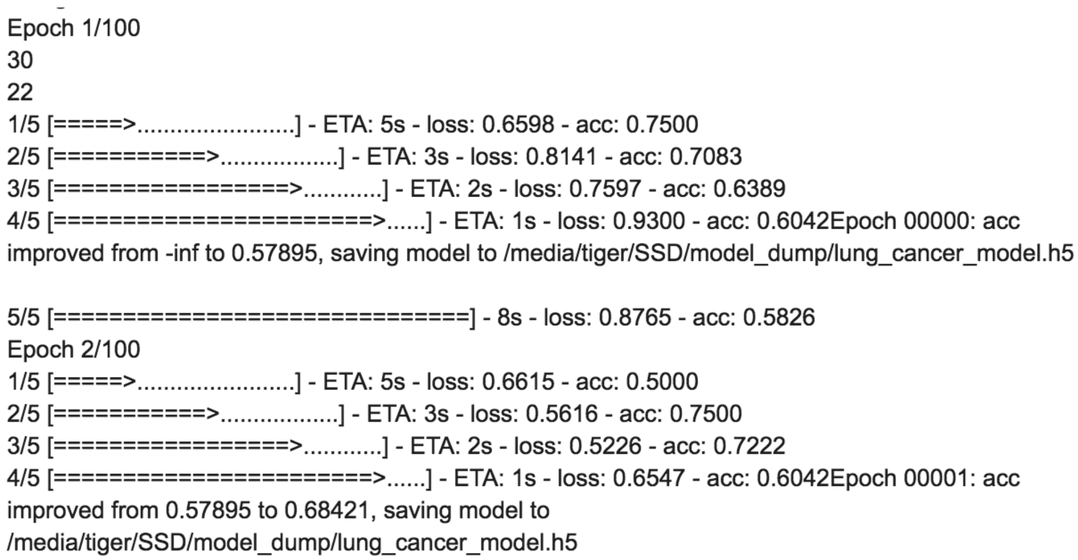
Approach 1: As previously mentioned, our first approach utilizes the Random Forest classifier on the simplistic feature extraction model. As such, the confusion matrix is as follows:

	PREDICTED: NO CANCER	**PREDICTED: YES CANCER**
ACTUAL: NO CANCER	301	383
ACTUAL: YES CANCER	117	128

Here, confusion matrix displays the total number of correct and incorrect predictions made by the classification model in comparison to the actual outcomes. For our purposes, the performance of our models will be evaluated using the data in the matrix. From the above output, we can see that out of 684 patients who did not have cancer, 383 were predicted incorrectly, indicating that our false positive rate is 56%. Similarly, out of the 245 patients who have cancer, only 128 were predicted correctly, indicating that our false negative rate is 47%. As we can see, the above input indicates that our feature extraction process is clearly inadequate for detecting lung cancer nodules. Our accuracy rate is around 50% for both the false positives and false negatives, and thus, we can conclude that the simplistic model based on high-level feature extraction and basic classification is not able to predict lung cancer nodules at a desirable accuracy. However, the simplistic approach provides us with a baseline for further analysis. As we proceed to analyze our more complex model involving a pre-trained feature extraction convolutional neural network, we can judge our accuracy not only by a raw percentage, but also as a relative increase compared to our simplistic model.

Approach 2: Now, as we move forward with the deep learning approach as previously mentioned, we used a convolutional neural network to classify patients as either cancerous or non-cancerous. Through the process of training the classifier, we look at the epoch output as followed:

At each step, the CNN calculates the Estimate Time of Arrival, loss, accuracy in each stage, total time, total loss and total accuracy for entire step. Sample output is found below:

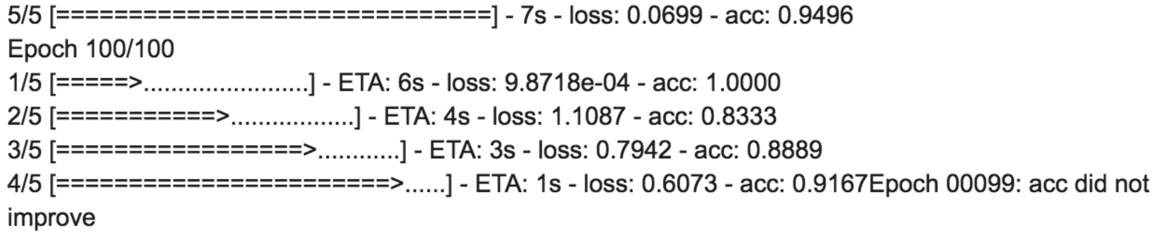


```
Epoch 1/100
30
22
1/5 [=====>........................] - ETA: 5s - loss: 0.6598 - acc: 0.7500
2/5 [===========>..................] - ETA: 3s - loss: 0.8141 - acc: 0.7083
3/5 [=================>............] - ETA: 2s - loss: 0.7597 - acc: 0.6389
4/5 [=======================>......] - ETA: 1s - loss: 0.9300 - acc: 0.6042Epoch 00000: acc
improved from -inf to 0.57895, saving model to /media/tiger/SSD/model_dump/lung_cancer_model.h5

5/5 [==============================] - 8s - loss: 0.8765 - acc: 0.5826
Epoch 2/100
1/5 [=====>........................] - ETA: 5s - loss: 0.6615 - acc: 0.5000
2/5 [===========>..................] - ETA: 3s - loss: 0.5616 - acc: 0.7500
3/5 [=================>............] - ETA: 2s - loss: 0.5226 - acc: 0.7222
4/5 [=======================>......] - ETA: 1s - loss: 0.6547 - acc: 0.6042Epoch 00001: acc
improved from 0.57895 to 0.68421, saving model to
/media/tiger/SSD/model_dump/lung_cancer_model.h5
```

Figure 8: Sample Epoch Output

As we can see from the above output, the CNN classifier categorizes the data, calculates the loss function, and then adjusts the weights accordingly. This process is repeated until the ACC no longer improves, shown in the output below:



```
5/5 [==============================] - 7s - loss: 0.0699 - acc: 0.9496
Epoch 100/100
1/5 [=====>........................] - ETA: 6s - loss: 9.8718e-04 - acc: 1.0000
2/5 [===========>..................] - ETA: 4s - loss: 1.1087 - acc: 0.8333
3/5 [=================>............] - ETA: 3s - loss: 0.7942 - acc: 0.8889
4/5 [=======================>......] - ETA: 1s - loss: 0.6073 - acc: 0.9167Epoch 00099: acc did not
improve
```

Thus, after our classification model is trained with the data, we can compute the confusion matrix as follows:

	PREDICTED: NO CANCER	**PREDICTED: YES CANCER**
ACTUAL: NO CANCER	744	1
ACTUAL: YES CANCER	57	198

Here, we able to see that by utilizing concepts in deep learning, we were able to significantly increase our accuracy rate. Not only were we able to predict the lung cancer nodules with an accuracy of 77%, we were also able to reduce the false positive rate to 0.001% in our training data. Now, when we take out trained classification model, and apply it to our testing data, we see that there is an overall accuracy of 70%. A potential reason for the drastic difference between the accuracy of the test and train data is overfitting – a modeling error which occurs when a function is too closely fit to a limited set of data points. While a point for further research includes handling the overfitting of the data, our achieved accuracy of 70% is much higher than our simplistic model prediction. Further, given the complexity of the problem, and the lack of manual analysis, we can conclude that an overall accuracy of 70% is adequate.

7.3 COMPARE OUTPUT AGAINST HYPOTHESIS

The efficiency of the model depends on the preprocessing, the feature extraction and the classification model. It also largely depends on the dataset used, the quality of CT scan images and also the volume of data or the number of patients. With increasing rate of lung cancer with time, more data will be available and thus it can improve the efficiency of the model accordingly. With the dataset of around 1,500 patients with low dose CT scan images (that has around 300 images per patient) where 75% of data is taken as input and 25% as output, we managed to get a 90% accuracy with our training data and a 70% accuracy rate with our test data with complete automated pulmonary nodule detection.

7.4 ABNORMAL CASE EXPLANATION

Since the data available was of low dose CT scan, the image quality could also have potentially affected the efficiency. With the advancement of technology, better CT scanners have been developed which could take CT scans with ultimate precision and image quality, hence being a better input for the model.

7.5 DISCUSSION:

We selected 1,000 patients to train our CNN classification model, and used the remaining 200 patients to test the algorithm. Many iterations or testing various parameters were attempted before concluding with the above model. As seen, we may have a potential problem with overfitting, but for now, we will conclude that our model is adequate for predicting the detection of lung cancer nodules.

8 CONCLUSION AND RECOMMENDATIONS

8.1 SUMMARY AND CONCLUSIONS:

In this paper, we study the use of image processing, data mining, and machine learning techniques to predict lung cancer nodules in high risk patients. Based on the research and analysis conducted for this project using a publicly available data set of lung CT scans, we were able to develop a successful model for lung cancer nodule detection. By using a hybrid of approaches in image processing and classification, we were able to develop an end to end process that detects lung cancer nodules with high accuracy. Further, by placing a heavy emphasis on automation of image processing as well as a reduction of false positives, we were able to develop a full model that runs with 70% accuracy on test data. Given the difficult nature of the problem, we faced various challenges throughout the process. First, the segmentation of lungs is a very challenging problem due to inhomogeneity in the lung region, pulmonary structures of similar densities such as arteries, veins, bronchi, and bronchioles, and different scanners and scanning protocols and difference in quality of the CT scans, the images had to be made uniform before processing. The CT scans being in hundreds of images per patient had a memory constraint while processing and also was a time consuming process since data per patient was relatively high (around 300 images per patient with around 1,500 patients).

8.2 RECOMMENDATIONS FOR FUTURE STUDIES:

For feature extraction, we have use VGG16 the simplicity of the model makes it easy to implement. However, there are other pre-trained CNN models available too for feature extraction. For complicated process like image classification, many other deep learning technologies are proposed. At each stage, one could use a novel approach to retain the maximum features, which would overall lead to a better model. Other pre-trained models could be used like Resnet, Googlenet, etc. for feature extraction and other modules for deep learning could be combined with deferent loss function, layers and optimization techniques for better results. As previously mentioned, our classification model is potentially overfitting the data, and thus, there exists a clear opportunity to research different methodologies to combat this problem. Finally, the image classification being used is MLP classifier with sequential model, however, other regularization techniques and functions could be explored to update weights in order to increase the accuracy of the model.

8.3 BIBLIOGRAPHY

"1.17. Neural Network Models (supervised)" *1.17. Neural Network Models (supervised) — Scikit-learn 0.18.1 Documentation*. Google, n.d. Web. 12 June 2017.

American Cancer Society. Cancer facts and figures, 2015. URL http://www. cancer.org/research/cancerfactsstatistics/index.

Bechtold, Robert E., Michael Y. M. Chen, David J. Ott, Ronald J. Zagoria, Eric S. Scharling, Neil T. Wolfman, and David J. Vining. "Interpretation of Abdominal CT: Analysis of Errors and Their Causes." *Journal of Computer Assisted Tomography* 21.5 (1997): 681-85. Web.

Chauhan, Divya, and Varun Jaiswal. "An Efficient Data Mining Classification Approach for Detecting Lung Cancer Disease." *2016 International Conference on Communication and Electronics Systems (ICCES)* (2016): n. pag. Web.

"Getting Started with the Keras Sequential Model." *Guide to the Sequential Model - Keras Documentation*. N.p., n.d. Web. 12 June 2017.

Golan, Rotem, Christian Jacob, and Jorg Denzinger. "Lung Nodule Detection in CT Images Using Deep Convolutional Neural Networks." *2016 International Joint Conference on Neural Networks (IJCNN)* (2016): n. pag. Web.

Hawkins, Samuel H., John N. Korecki, Yoganand Balagurunathan, Yuhua Gu, Virendra Kumar, Satrajit Basu, Lawrence O. Hall, Dmitry B. Goldgof, Robert A. Gatenby, and Robert J. Gillies. "Predicting Outcomes of Nonsmall Cell Lung Cancer Using CT Image Features."*IEEE Access* 2 (2014): 1418-426. Web.

Jafar, Iyad, Hao Ying, Anthony F. Shields, and Otto Muzik. "Computerized Detection of Lung Tumors in PET/CT Images." *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (2006): n. pag. Web.

Juma, Kassimu, Ma He, and Yue Zhaoc. "Lung Cancer Detection and Analysis Using Data Mining Techniques, Principal Component Analysis and Artificial Neural Network."*American Scientific Research Journal for Engineering, Technology, and Sciences* (n.d.): n. pag. Web.

Kuruvilla, Jinsa, and K. Gunavathi. "Lung Cancer Classification Using Neural Networks for CT Images." *Computer Methods and Programs in Biomedicine* 113.1 (2014): 202-09. Web.

Paul, Rahul, Samuel H. Hawkins, Lawrence O. Hall, Dmitry B. Goldgof, and Robert J. Gillies. "Combining Deep Neural Network and Traditional Image Features to Improve Survival Prediction Accuracy for Lung Cancer Patients from Diagnostic CT." *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2016): n. pag. Web.

Rao, R. Bharat, Jinbo Bi, Glenn Fung, Marcos Salganicoff, Nancy Obuchowski, and David Naidich. "LungCAD." *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07* (2007): n. pag. Web.

Raschka, Sebastian. "KDnuggets." *KDnuggets Analytics Big Data Data Mining and Data Science*. N.p., n.d. Web. 12 June 2017.

Rosebrock, Adrian. "ImageNet: VGGNet, ResNet, Inception, and Xception with Keras."*PyImageSearch*. Pyimagesearch, 03 May 2017. Web. 12 June 2017.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting."*Journal of Machine Learning Research* (2014): n. pag. Web.

Ypsilantis, Petros-Pavlos, and Giovanni Montana. "Recurrent Convolutional Networks for Pulmonary Nodule Detection in CT Imaging." (2016): 1-36.*Https://arxiv.org/pdf/1609.09143.pdf*. Web. May 2017.

Zisserman, Andrew, and Karen Simonyan. "Very Deep Convolutional Networks for Large-Scale Visual Recognition." *Visual Geometry Group Home Page*. Visual Geometry Group Department of Engineering Science, University of Oxford, n.d. Web. 12 June 2017.

8.4 PROGRAM FLOWCHART

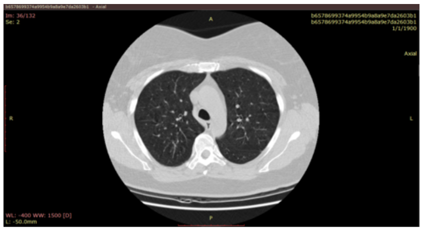
The program flow chart can be found in section 6.3 and detailed instructions for running the code can be found in the submitted README file.

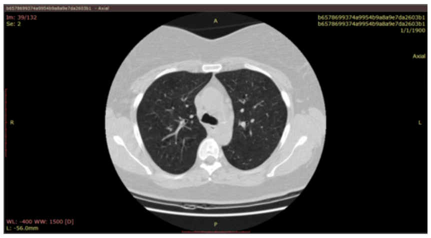
8.5 PROGRAM SOURCE CODE WITH DOCUMENTATION

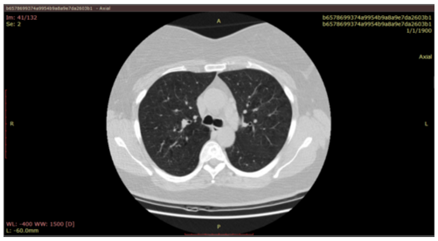
The code can be found in the submitted files, along with a README file containing the full documentation.

8.6 INPUT/OUTPUT LISTING

Input: As previously mentioned, the input files are in the form of a DICOM image. Examples of slices in a DICOM image for one patient can be found below.


b6578699374a99954b9a8a9e7da2603b1 - Axial
Im: 36/132
Se: 2
b6578699374a99954b9a8a9e7da2603b1
b6578699374a99954b9a8a9e7da2603b1
1/1/1900
A
Axial
R
L
WL: -400 WW: 1500 [D]
L: -50.0mm
P


b6578699374a99954b9a8a9e7da2603b1 - Axial
Im: 39/132
Se: 2
b6578699374a99954b9a8a9e7da2603b1
b6578699374a99954b9a8a9e7da2603b1
1/1/1900
A
Axial
R
L
WL: -400 WW: 1500 [D]
L: -56.0mm
P


b6578699374a99954b9a8a9e7da2603b1 - Axial
Im: 41/132
Se: 2
b6578699374a99954b9a8a9e7da2603b1
b6578699374a99954b9a8a9e7da2603b1
1/1/1900
A
Axial
R
L
WL: -400 WW: 1500 [D]
L: -60.0mm
P

Output: The final output is a classification value: 1 for cancerous, and 0 for non-cancerous.