MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

JUNE 2020

DETECTING AND RECOGNIZING LUNG CANCER USING CONVOLUTIONAL NEURAL NETWORKS

MAJOR PROJECT REPORT

PROJECT MENTOR
Dr. Namita Tiwari

PROJECT COORDINATOR
Dr. Saritha Khetwat

SUBMITTED BY

ABHISHEK PANDEY	161112001
LOKESH LOVEWANSHI	161112031
SUDHANSHU RANJAN	161112046
SHUBHAM KOSE	161112049

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Abhishek Pandey, Lokesh Lovewanshi, Sudhanshu Ranjan, and Shubham Kose**, students of B. Tech 4th Year (Computer Science and Engineering), have successfully completed their project **"Detecting and Recognizing Lung Cancer using Convolutional Neural Networks"** in partial fulfillment of their minor project in Computer Science and Engineering.

Dr. Namita Tiwari
Assistant Professor
Project Guide

Dr. Saritha Khetwat
Assistant Professor
Project Coordinator

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, hereby, declare that the following report, which is being presented in the Minor Project Documentation entitled **"Detecting and Recognizing Lung Cancer using Convolutional Neural Networks"** is for the partial fulfillment of the requirements of the final year (eighth semester) Major Project in the field of Computer Science and Engineering. It is an authentic documentation of our own original work carried out under the able guidance of Dr. Namita Tiwari. The work has been carried out entirely at Maulana Azad National Institute of Technology, Bhopal. The following project and it's report, in part or whole, has not been presented or submitted by us for any other purpose in any other institute or organization.

We, hereby, declare that the facts mentioned above are true to the best of our knowledge. In case of any unlikely discrepancies that may possibly occur, we will be the ones to take responsibility.

ABHISHEK PANDEY	**161112001**
LOKESH LOVEWANSHI	**161112031**
SUDHANSHU RANJAN	**161112046**
SHUBHAM KOSE	**161112049**

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide **Dr. Namita Tiwari** for her valuable help and guidance. We are thankful for the encouragement that she has given us in undertaking this project. Her rigorous evaluation and constructive criticism is of great assistance.

We are also grateful to our respected director for permitting us to utilize all the necessary facilities of the college.

Needless to mention is the additional help and support extended by our respected HOD, **Dr. Nilay Khare**, in allowing us to use the departmental laboratories and other services.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help.

Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing the much needed support and encouragement.

CONTENT

List of Figures

ABSTRACT

Lung cancer is one of the most dreadful diseases in the developing countries and its mortality rate is 19.4%. Early detection of lung tumor is done by using many imaging techniques such as Computed Tomography (CT), Sputum Cytology, Chest X-ray and Magnetic Resonance Imaging (MRI). Detection means classifying tumor two classes (i)non-cancerous tumor (benign) and (ii)cancerous tumor (malignant). The chance of survival at the advanced stage is less when compared to the treatment and lifestyle to survive cancer therapy when diagnosed at the early stage of the cancer. Manual analysis and diagnosis system can be greatly improved with the implementation of image processing techniques. A number of researches on the image processing techniques to detect the early stage cancer detection are available in the literature. But the hit ratio of early stage detection of cancer is not greatly improved. With the advancement in the machine learning techniques, the early diagnosis of the cancer is attempted by lot of researchers. Neural network plays a key role in the recognition of the cancer cells among the normal tissues, which in turn provides an effective tool for building an assistive AI based cancer detection. The cancer treatment will be effective only when the tumor cells are accurately separated from the normal cells. Classification of the tumor cells and training of the neural network forms the basis for the machine learning based cancer diagnosis. This major project presents a Convolutional Neural Network (CNN) based technique to classify the lung tumors as malignant or benign.

1. Introduction

1. 1. Neural Networks

A **neural network** in a modern sense is a network or circuit of artificial neurons, to build an artificial neural network. Thus a neural network is a artificial neural network, for solving artificial intelligence (AI) problems. The connections of the neurons are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections.

1. 2. Why Neural Networks?

Images (and movies) have become ubiquitous in both production and consumption.

- Neural networks have the ability to learn and model non-linear and complex relationships, which is really important in real-life.
- Neural networks can generalize - after learning from their initial inputs and their relationships, it can infer unseen relationships on unseen data.
- Unlike many prediction techniques, neural network doesn't impose any restrictions on the input variables

1. 3. Applications of Neural Networks

- **Character Recognition**
  It is an interesting problem which falls under the general area of Pattern Recognition. Many neural networks have been developed for automatic recognition of handwritten characters, either letters or digits. Following are some Neural Networks which have been used for character recognition −
  i. Multilayer neural networks such as Backpropagation neural networks.
  ii. Neocognitron

- **Forecasting**
  Forecasting is required extensively in everyday business decisions (e.g. sales, financial allocation between products, capacity utilization), in economic and monetary policy, in finance and stock market. More often, forecasting problems are complex, for example, predicting stock prices is a complex problem with a lot of underlying factors (some known, some unseen).

- **Speech Recognition**
  Neural Networks are playing a major role in this area. Following Neural Networks have been used for speech recognition −
  i. Multilayer networks
  ii. Multilayer networks with recurrent connections
  iii. Kohonen self-organizing feature map

The most useful network for this is Kohonen Self-Organizing feature map, which has its input as short segments of the speech waveform. It will map the same kind of phonemes as the output array, called feature extraction technique. After extracting the features, with the help of some acoustic models as back-end processing, it will recognize the utterance.

- **Human Face Recognition**
  It is one of the biometric methods to identify the given face. It is a typical task because of the characterization of "non-face" images. First, all the input images must be preprocessed. Then, the dimensionality of that image must be reduced. And, at last it must be classified using neural network training algorithm. Following neural networks are used for training purposes with preprocessed image −
    i. Fully-connected multilayer feed-forward neural network trained with the help of back-propagation algorithm.
    ii. For dimensionality reduction, Principal Component Analysis (PCA) is used.

1. 4. DIGITAL IMAGE PROCESSING

Digital image processing is the use of computer algorithms to perform image processing on digital images.

1. 5. APPLICATIONS OF DIGITAL IMAGE PROCESSING

- Image sharpening and restoration
- Medical Field
- Remote Sensing
- Transmission and encoding
- Machine/Robot vision
- Colour Processing
- Pattern Recognition
- Video processing
- Microscopic Imaging
- Others

- **Computer Vision VS Image Processing**
  Image processing studies image to image transformation. The input and output of image processing are both images.

  Computer vision is the construction of explicit, meaningful description of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

1. 6. Convolutional Neural Network (CNN)

- **What is Convolutional Neural Network?**
  CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, passes it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that was developed for neural networks still apply on CNNs.

- **Use of Convolutional Neural Network**
  - Image recognition
    CNNs are often used in image recognition systems. In 2012 an error rate of 0.23 percent on the MNIST database was reported. Another paper on using CNN for image classification reported that the learning process was "surprisingly fast"; in the same paper, the best published results as of 2011 were achieved in the MNIST database and the NORB database. When applied to lung caner detection, CNNs achieved a large decrease in error rate. Another paper reported a 97.6 percent recognition rate on "5,600 still images of more than 10 subjects". CNNs were used to assess video quality in an objective way after manual training; the resulting system had a very low root mean square error.

  - Video analysis
    Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream. Long short-term memory (LSTM) recurrent units are typically incorporated after the CNN to account for inter-frame or inter-clip dependencies.

  - Natural language processing
    CNNs have also been explored for natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

  - Drug discovery
    CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments. In 2015, Atomwise introduced AtomNet, the first deep learning neural network for structure-based rational drug design. The system trains directly on 3-dimensional representations of

chemical interactions. Similar to how image recognition networks learn to compose smaller, spatially proximate features into larger, complex structure, AtomNet discovers chemical features, such as aromaticity, sp^3 carbons and hydrogen bonding.

- Health risk assessment and biomarkers of aging discovery
  CNNs can be naturally tailored to analyze a sufficiently large collection of time series data representing one-week-long human physical activity streams augmented by the rich clinical data. A simple CNN was combined with Cox-Gompertz proportional hazards model and used to produce a proof-of-concept example of digital biomarkers of aging in the form of all-causes-mortality predictor

2. Literature Survey

In the last few years Convolutional neural networks have become the dominant approach in medical image analysis that outperforms algorithms based on handcrafted features. Esteva et al. proposed the CNN-based system for skin lesion classification that achieves performance comparable with dermatologist's assessment.

Ginneken et al (2001) described a survey in which the authors classify the lung regions extraction approaches into two different categories;

•Rule-based
•Pixel classification based category

In the rule-based technique, sequences of steps, tests and rules are used in the extraction process and most of the proposed techniques belong to this category. Thresholding, region growing, edge detection, ridge detection, morphological operations, and dynamic programming are some of the techniques used. Pixel classification is another approach used for the lung regions extraction process, where each pixel in the CT image is categorized into an anatomical class. Various types of neural networks, Markov random field modeling are the classifiers trained with a variety of local features including intensity, location, and texture measures.

Woten et al (2007) proposed a numerical investigation into the improvement of artificial neural network detection of breast cancer using a planar broadband antenna and a three-region breast approach. In this approach, Modified Four point antennas are used, which are capable of producing various wave polarizations. The effect of wave polarization on statistical detection is fully described in this approach by the author.

Hany Ayad Bastawrous Fukumoto et al (2005) proposed a CAD approach used for the detection of Ground Glass Opacity (GGO) nodules in chest CT images. Gabor filter on the CT image was used in this proposed approach in order to enhance the detection process. Then some morphological processes including threshold process and labeling to extract the objects having high intensity values were performed. Then, to examine these objects some feature analysis was used which decides which of them are likely to be cancer candidates. A template matching between the potential cancer candidates and some Gaussian reference approaches was performed by following the feature analysis, to find the similarity between them. The approach was applied on 715 slices containing 25 GGO nodules and achieved detection sensitivity of 92% with False Positive (FP) rate of 0.76 FP/slice. Finally, Artificial Neural Network (ANN) was used to reduce the number of FP findings. The FP rate reduces to 0.25 FP/slice after applying ANN but at the expense of decreasing the detection sensitivity to 84%.

Kalman Filtering (KF). A novel nonlinear decision approach to enhance the performance of the classification was proposed. From the experimental observation, it is clearly observed that the sensitivity of this classification/detection is 100% with the false positive detection rate of less than 1 Micro-Calcification Clusters (MCCs) per image. The proposed approaches are automatic or operator independent and offer realistic image processing times as required for breast cancer screening programs.

Computer-Aided Diagnostic (CAD) approaches use a filter for enhancement of lesions as a preprocessing step for enhancing sensitivity and specificity. Thus, existing filters fail to improve actual lesions. Suzuki et al (2005) proposed a supervised filter for enhancement of lesions by use of a Massive-Training Artificial Neural Network (MTANN) in a CAD scheme for detection of lung nodules in CT. The MTANN filter was trained with actual nodules in CT images to improve actual patterns of nodules. By use of the MTANN filter, the sensitivity and specificity of this CAD approach were enhanced. With the database of 69 lung cancers, this CAD approach with the MTANN filter achieved 97% sensitivity with 6.7 false positives (FPs) per section, whereas a conventional CAD technique with a difference-image technique achieved 96% sensitivity with 19.3 FPs per section.

The techniques discussed here mainly deal with Lung Images, they can also be applied to mammogram images and brain images.

In (Mokhled, 2012) first images which were improved through Gabor filter. It has given better results than other enhancement techniques. They only worked on colored image enhancement and not extract the nucleus region and even not the cell region. In Features Extraction stage they acquire the general features of the enhanced and segmented image which later they used in Binarization. A refined Charged Fluid Model (CFM) along with improved Otsu's method was used for the automatic segmentation of MRI images in (Nagesj,

2012). This method gave enhanced results than the result given by the approaches used in previous experiments.

In (Nikita, 2012), a sober edge detection method was used which is based on finding the image gradient. This method tells that intensity of the image will be maximum where there is a separation of two dissimilar regions and thus an edge must exist there. On this basis they found the nodules in CT images. In (Parsh, 2011), a new variation level set algorithm without re-initialization was used. They also used thresholding to reduce the noise component of the images.

In (Sonith, 2012) an overview of entire process for processing digital images for lung cancer detection is given in this paper. This paper also describes all the essential steps required for the better performance starting from the pre-processing till the very end phase extraction of features

Regarding lung cancer diagnosis, methods proposed so far have dealt mostly with radiology. In image-based radiomics features strongly related to survival are extracted from positron emission tomography-computed tomography (PET/CT) scans. CNN is employed for classification of lung nodule images yielding accuracy of 86.4%. In digital pathology tasks CNNs chave been used on cell level for mitosis detection and cell nuclei detection. CAMELYON16 was the first challenge dealing with WSIs to detect breast cancer metastases in lymph nodes.

Thanks to the availability of large annotated training set in this challenge, it was possible to train deeper and more powerful CNN architectures like GoogLeNet, VGG-Net and ResNet. Method that gives the best result in this challenge is described in. It performs patch-based classification to discriminate tumor patches from normal patches using a combination of 2 GoogLeNet architectures where one of them is trained with and another without hard-negative mining.

Aim of TUPAC challenge was WSI based mitosis detection in breast cancer tissue and tumor grading prediction. In the best performing method ROI regions are firstly extracted from WSI based on cell density. This is followed by mitosis detection using ResNet CNN architecture. Finally, each WSI is represented by feature vector including the number of mitoses and cells in each patch as well as other features derived from statistics. This feature vector is fed to SVM classifier to predict tumor proliferation.

3. Methodology & Work Description

Module Description

For the purpose of the project, we are using Kaggle dataset (available at https://www.kaggle.com/c/data-science-bowl-2017/data) and LUNA dataset (available at https://luna16.grand-challenge.org/download/).

The CNN was developed with variable depths to evaluate the performance of these models for lung cancer recognition.

The first part of the network refers to M convolutional layers that can possess spatial batch normalization, dropout, and max-pooling in addition to the convolution layer and ReLU nonlinearity, which always exists in these layers. After M convolution layers, the network is led to N fully connected layers that always have Affine operation and ReLU nonlinearity, and can include batch normalization and dropout. Finally, the network is followed by the affine layer that computes the scores and softmax loss function.

Fig. 1 - Depicts non cancer carrying lung on left, cancerous lung with nodule marked on the right.

The developed model gives the user the freedom to decide about the number of convolutional and fully connected layers, as well as the existence of batch normalization and max -pooling layers. Along with batch normalization techniques, regularization was included in the implementation. Furthermore, the number of filters and layers can be specified by user for the best results .

The images are processed in such a way that the lungs are almost centered and each set of lungs occupies about the same amount of space in each image. Each image has to be categorized into one of the two classes that express different cancer possibilities. The lungs have been categorized as:

0=Benign/Non-malignant/Non-cancerous,
1=Malignant/Cancerous

Figure 1 depicts one example for each possibility. In addition to the image class number (a number between 0 and 1), the given images are divided into two different sets which are training, test sets. There are about 5012 training images, 1239 images for testing. For the purpose of data augmentation, mirrored images were produced by flipping images in the training set horizontally. In order to classify the expressions, the features generated by convolution layers was mainly used, using the raw pixel data, give them as input features into Fully Connected (FC) layers.

Based on the results of previous publications, a decision was taken to create a CNN by oneself and train it from scratch. A 9-layer CNN with two convolutional layers, two pooling layers, and 4 fully connected dense layers along with a matrix flattening layer. The structure of the CNN is shown in Fig 2.

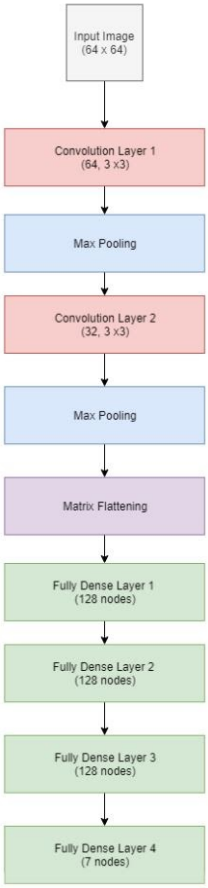


Fig. 2: Flow of Architecture

3. 1. Extracting Effective Features

In this module, first the system will take the image from the dataset taken. Then the input image is first checked for the lung x-ray features. In case if the image does not contain lung features, then it is not detected. If the input image contains lung features, then it detects the features. Lung is detected from the image as shown as Fig. 3

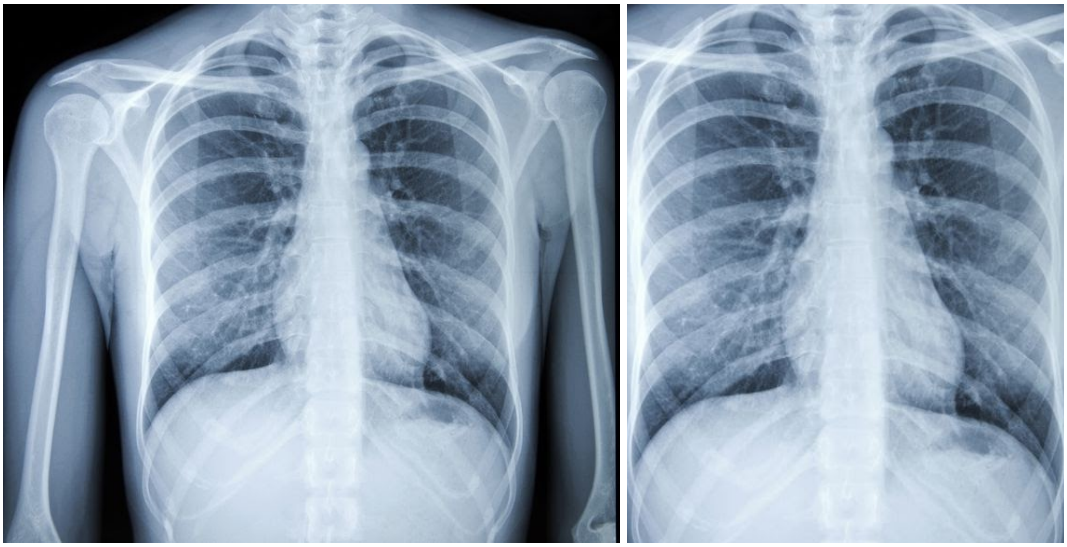


Fig. 3: Lung Extraction

3. 2. Feature Point Detection

For lung detection, first - convert the image from an RGB format to a binary format. The next step is to find the ribs from the binary image. System will start scanning from the middle of the image, after that it will look for continuous white pixels after a continuous black pixel. In this the goal is to find the maximum width of the white pixel by searching vertically both left and right side. Then, if the new width is smaller than half of the previous maximum width, then the scan is broken because in that case the scan will reach the diaphragm. Then the lung is cut from the starting position of the x ray and its height will be 1.5 multiple of its width. This processed image will have the lung, hotspot and body.

Fig. 4: Tumor Detection

4. Tools and Technology to be Used

4. 1. Software Requirements

- Python
- Keras Library
- Anaconda Navigator
- Numpy Library

4. 2. Hardware Requirements

- Windows XP or Above
- 2GB of RAM
- Any Dual Core Processor or above

5. Implementation and Coding

The main challenge of the project was the implementation. Many failed iterations of the project led to finally land upon the correct implementation that allowed to achieve decent performance giving optimum results.

At the end of many tries, the project stemmed from the following self created architecture.

S. No	Layer	Shape
1	Convolution2D	(Filters(64, 3, 3), input_shape = (64, 64, 3), activation = ‘relu’)
2	MaxPooling2D	(pool_size = (2, 2))
3	Convolution2D	(Filters(32, 3, 3), activation = ‘relu’)
4	MaxPooling2D	(pool_size = (2, 2))
5	Flatten	Flatten the matrix
6	Dense	(output_dim = 128, activation = ‘relu’)
7	Dense	(output_dim = 128, activation = ‘relu’)
8	Dense	(output_dim = 128, activation = ‘relu’)
9	Dense	(output_dim = 7, activation = ‘softmax’)

From the above architecture, the following code was successfully implemented.

We used the following code to train the model:

```
"""
Trains a CNN model using tflearn wrapper for tensorflow
"""


import tflearn
import h5py
import numpy as np
from cnn_model import CNNModel


# Load HDF5 dataset
h5f = h5py.File('../data/train.h5', 'r')
X_train_images = h5f['X']
Y_train_labels = h5f['Y']


h5f2 = h5py.File('../data/val.h5', 'r')
X_val_images = h5f2['X']
Y_val_labels = h5f2['Y']


## Model definition
convnet  = CNNModel()
network = convnet.define_network(X_train_images)
model = tflearn.DNN(network, tensorboard_verbose=0,\
            checkpoint_path='nodule3-classifier.tfl.ckpt')
model.fit(X_train_images,  Y_train_labels,  n_epoch  =  70,
shuffle=True,\
                  validation_set  =  (X_val_images,  Y_val_labels),
show_metric = True,\
                  batch_size = 96, snapshot_epoch = True, run_id =
'nodule3-classifier')
model.save("nodule3-classifier.tfl")
print("Network trained and saved as nodule2-classifier.tfl!")

h5f.close()
h5f2.close()
```

We used the following code to test the model:

```
"""
A script to predict nodules using conv net model and for analysis of
results
"""

import tflearn
from cnn_model import CNNModel
```

```
import tensorflow as tf

import pickle
import pandas as pd
import numpy as np
import h5py
from sklearn.metrics import roc_curve, auc, confusion_matrix
import itertools

import matplotlib.pyplot as plt

hdfs_file = '../data/test.h5'

def create_mosaic(image, nrows, ncols):
  """
  Tiles all the layers in nrows x ncols
  Args:
  ------
  image = 3d numpy array of M * N * number of filters dimensions
  nrows = integer representing number of images in a row
  ncol = integer representing number of images in a column

  returns formatted image
  """

  M = image.shape[1]
  N = image.shape[2]

  npad = ((0,0), (1,1), (1,1))
  image = np.pad(image, pad_width = npad, mode = 'constant',\
    constant_values = 0)
  M += 2
  N += 2
  image = image.reshape(nrows, ncols, M, N)
  image = np.transpose(image, (0,2,1,3))
  image = image.reshape(M*nrows, N*ncols)
  return image


def format_image(image, num_images):
  """
  Formats images
  """
  idxs = np.random.choice(image.shape[0], num_images)
  M = image.shape[1]
  N = image.shape[2]
  imagex = np.squeeze(image[idxs, :, :, :])
  print (imagex.shape)
  return imagex



def plot_confusion_matrix(cm, classes,
```

```
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Purples):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    #plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
             for   i,   j   in   itertools.product(range(cm.shape[0]),
range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    #plt.grid('off')
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

def load_images(filename):
  """
  Loads images contained in hdfs file
  """
  h5f2 = h5py.File(filename, 'r')
  X_test_images = h5f2['X']
  Y_test_labels = h5f2['Y']
  return X_test_images, Y_test_labels

def plot_predictions(images, filename):
  """
  Plots the predictions mosaic
  """
  imagex = format_image(images, 4)
  mosaic = create_mosaic(imagex, 2, 2)
  plt.figure(figsize = (12, 12))
  plt.imshow(mosaic, cmap = 'gray')
  plt.axis('off')
  plt.savefig(filename + '.png', bbox_inches='tight')
```

```
def get_predictions(X_test_images, Y_test_labels):
  """
  Args:
  ------
  Given hdfs file of X_test_images and Y_test_labels

  returns:
  --------
  predictions: probability values for each class
  label_predictions: returns predicted classes
  """

  ## Model definition
  convnet  = CNNModel()
  network = convnet.define_network(X_test_images)
  model = tflearn.DNN(network, tensorboard_verbose=0,\
            checkpoint_path='nodule3-classifier.tfl.ckpt')
  model.load("nodule3-classifier.tfl")

  predictions = np.vstack(model.predict(X_test_images[:,:,:,:]))
                                    #label_predictions                  =
np.vstack(model.predict_label(X_test_images[:,:,:,:]))
  score = model.evaluate(X_test_images, Y_test_labels)
  label_predictions = np.zeros_like(predictions)
   label_predictions[np.arange(len(predictions)), predictions.argmax(1)]
= 1
  return predictions, label_predictions

def get_roc_curve(Y_test_labels, predictions):
  """
  Args:
  -------
  hdfs datasets: Y_test_labels and predictions

  Returns:
  --------
  fpr: false positive Rate
  tpr: true posiive Rate
  roc_auc: area under the curve value
  """
        fpr,    tpr,    thresholds    =    roc_curve(Y_test_labels[:,1],
predictions[:,1], pos_label=1)
  roc_auc = auc(fpr, tpr)
  return fpr, tpr, roc_auc


def get_metrics(Y_test_labels, label_predictions):
  """
  Args:
  -----
  Y_test_labels, label_predictions

  Returns:
  --------
```

```
  precision, recall and specificity values and cm
  """
  cm = confusion_matrix(Y_test_labels[:,1], label_predictions[:,1])

  TN = cm[0][0]
  FP = cm[0][1]
  FN = cm[1][0]
  TP = cm[1][1]

  precision = TP*1.0/(TP+FP)
  recall = TP*1.0/(TP+FN)
  specificity = TN*1.0/(TN+FP)

  return precision, recall, specificity, cm

def plot_roc_curve(fpr, tpr, roc_auc):
  """
  Plots ROC curve

  Args:
  -----
  FPR, TPR and AUC
  """
  plt.figure()
  lw = 2
  plt.plot(fpr, tpr, color='darkorange',
    lw=lw, label='(AUC = %0.2f)' % roc_auc)
  plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
  plt.axis('equal')
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.legend(loc="lower right")
  plt.savefig('roc1.png', bbox_inches='tight')


def main():
  X_test_images, Y_test_labels = load_images(hdfs_file)

  predictions, label_predictions = \
  get_predictions(X_test_images, Y_test_labels)

  fpr, tpr, roc_auc = get_roc_curve(Y_test_labels, predictions)
  plot_roc_curve(fpr, tpr, roc_auc)

  precision, recall, specificity, cm =\
   get_metrics(Y_test_labels, label_predictions)

  print (precision, recall, specificity )

  # Plot non-normalized confusion matrix
  plt.figure()
  plot_confusion_matrix(cm, classes=['no-nodule', 'nodule'], \
    title='Confusion matrix')
  plt.savefig('confusion_matrix.png', bbox_inches='tight')
```

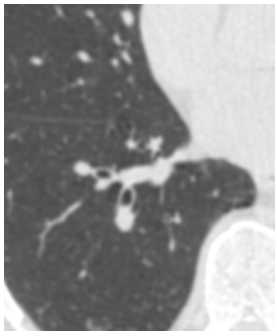
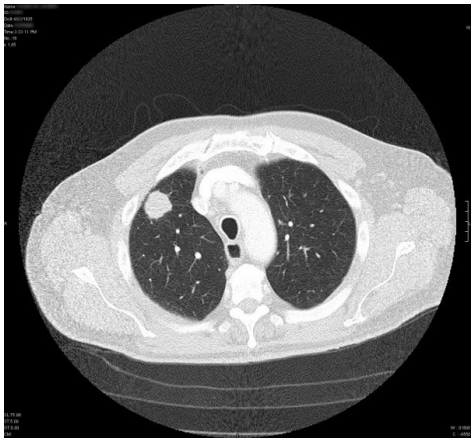
```
  # Plot all inputs representing True Positives, FP, FN, TN
       TP_images   =   X_test_images[(Y_test_labels[:,1]   ==   1)   &
(label_predictions[:,1] == 1), :,:,:]
       FP_images   =   X_test_images[(Y_test_labels[:,1]   ==   0)   &
(label_predictions[:,1] == 1), :,:,:]
       TN_images   =   X_test_images[(Y_test_labels[:,1]   ==   0)   &
(label_predictions[:,1] == 0), :,:,:]
       FN_images   =   X_test_images[(Y_test_labels[:,1]   ==   1)   &
(label_predictions[:,1] == 0), :,:,:]

  ## Choose 16 images randomly
  plot_predictions(TP_images, 'preds_tps')
  plot_predictions(TN_images, 'preds_tns')
  plot_predictions(FN_images, 'preds_fns')
  plot_predictions(FP_images, 'preds_fps')

if __name__ == "__main__":
  main()
```

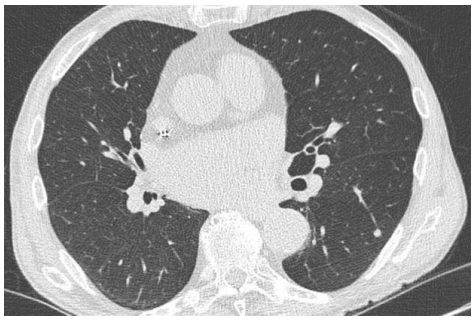
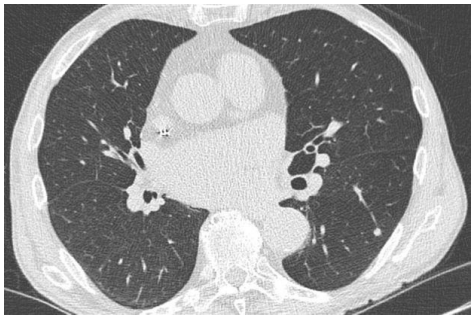
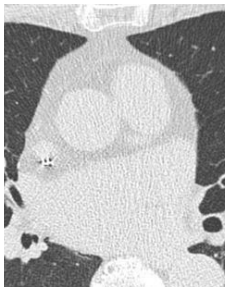
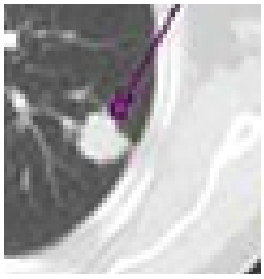
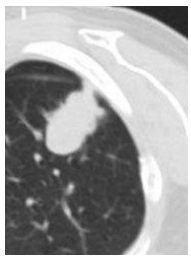
6. Result Analysis

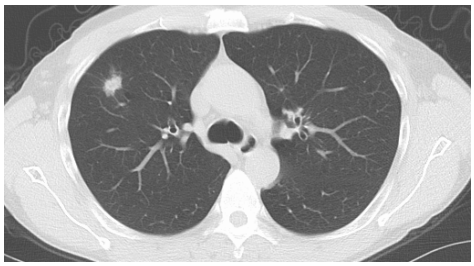
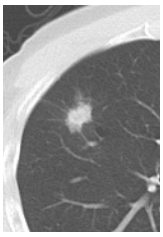
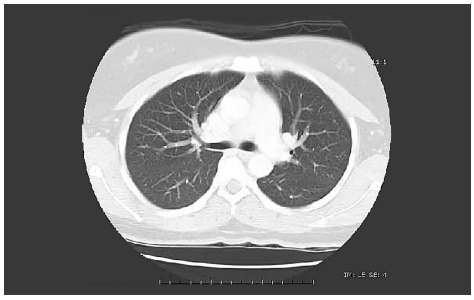
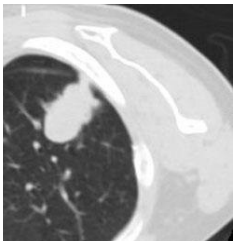
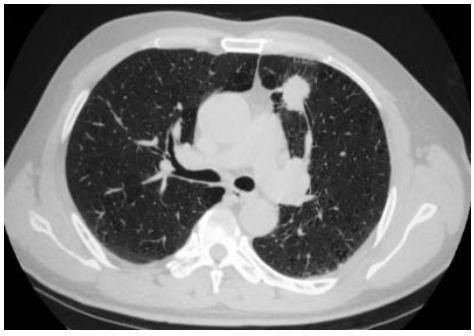
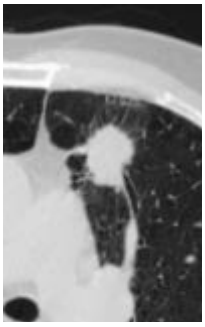
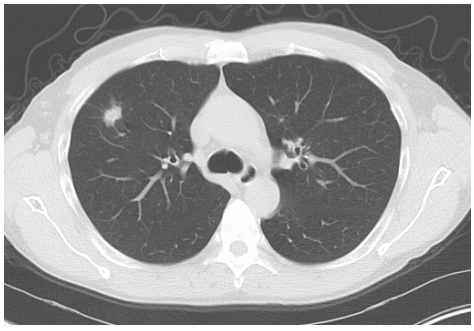
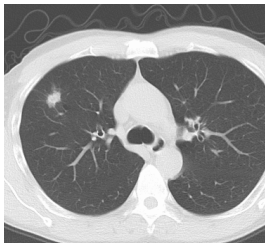
The main aim of this project was to showcase the technology of lung cancer detection. To be able to allow a machine and a program to detect the various cancers, first cancers need to be selected. A total of 2 results were finalized depending upon the most common scenarios of lung cancer detection. The options are:

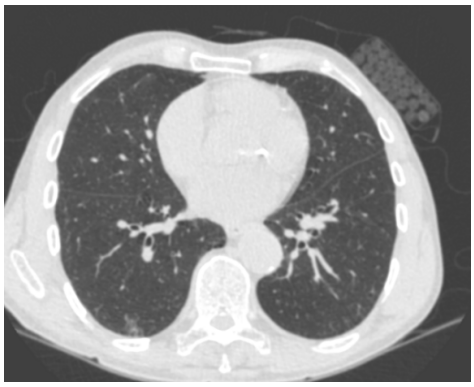
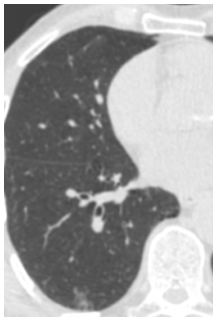
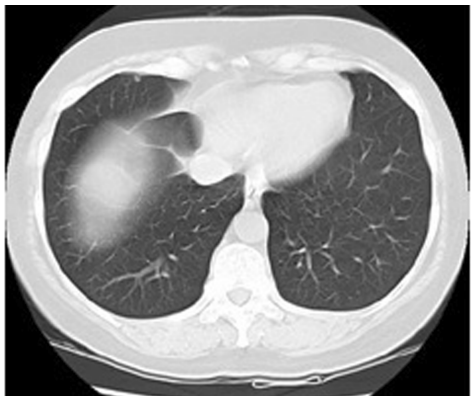
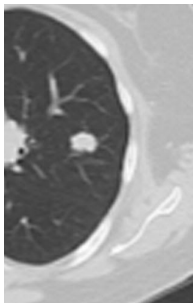
1. 0=Benign/Non-malignant/Non-cancerous,
2. 1=Malignant/Cancerous

The program was fed 10 different types of images from a static data set. The following are the results of the project.

S. No	Image	Output	Accuracy
1			Correct
2			Correct

3			Incorrect
4			Incorrect
5			Incorrect
6			Correct
7			Correct

8			Correct
9			Correct
10			Correct
11			Correct
12			Incorrect

13			Correct
14			Correct
15			Correct

After the above observations, it can be noted that out of 10 instances, the program was able to identify lung cancer11 times correctly, and 4 times incorrectly.

The accuracy can then be calculated as:

Accuracy on the above images = $\frac{Number\ of\ correct\ instances}{Number\ of\ total\ instances}$ x 100

Accuracy on the above images = (11/15) x 100

Accuracy on the above images = 73.33%

Accuracy on training dataset = 98%

Accuracy on the test dataset = 76%

7. Conclusion and Future Scope

The first novelty in our paper is using the K-means algorithm to pre-classify the pictures into piles of same slice images, where the DNN can specialize in image classification of same slice images.

The second novelty is that the additional convolution layer with edge sharpening filters, to thoroughly search for cancer. Finally, the most novelty is testing our Deep Neural Network with carcinoma images from Tx stages 2, 3 and 4 and determining at which Tx stage the 2 algorithms can detect the possibility of cancer. The results were analyzed with medical personnel from the oncology department and were marked as satisfactory to see cancer in T3 phase.

For future work, we plan on making an extra analysis, where we are going to change the DNN to output 2 values (0 and 1) and determine which one has higher certainty of classification. This way, we can classify the image not even as being a decimal value between 0.0 or 1.0, but also compare what proportion is 0 (not cancer) and the way much is 1 (cancer). For extra future work, almost like Cruz-Roa and Ovalle, who used RGB (color) images to spotlight the realm of malignant cells, we plan on modifying the DNN to indicate to us where (the location) on the CT image it's detected a cancer.

8. REFERENCES

1. "Lung cancer detection using digital image processing and artificial neural networks". Published by IEEE 2017 international conference, authored by S. Kalaivani, Pramit Chatterjee, Shikhar Juyal, Rishi Gupta
   https://ieeexplore.ieee.org/document/8212773
2. "A Deep Convolutional Neural Network for Lung Cancer Diagnostic" by Mehdi Fatan Serj, Bahram Lavi, Gabriella Hoff and Domenec Puig Valls
3. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6232785/
4. "Computer aided lung cancer detection". Published by IEEE 2015 Global Conference, authored by Sruthi Ignatious, Robin Joseph
5. "Using Double Convolutional Neural Network for Lung Cancer Stage Detection" by Goran Jakimovski and Danco Davcev
6. "A review paper on computer aided system on lunc cancer detection". Published by IEEE 2017 International Conference, authored by Snehal Dabade, Shubhangi Chaudhari, Sneha Jadhav, Arjun Nichal
7. https://www.researchgate.net/publication/319453582_Lung_Cancer_Detection_and_Classification_with_3D_Convolutional_Neural_Network_3D-CNN
8. https://clincancerres.aacrjournals.org/content/25/11/3266
9. "Lung Cancer Detection Using Image Processing Techniques" by Mokhled S. AL-TARAWNEH

10. https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1002711
11. https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1002711