SCTR's Pune Institute of Computer Technology
(PICT) Pune

AN
INTERNSHIP REPORT
ON

Lung Cancer Detection and Classification

SUBMITTED BY

Name: Omkar R. Nevse
Class: TE - 05
Roll no: 32130

Under the guidance of

Dr. R. Sreemathy, Associate Professor.
Ms. Ankita K. Patel, Assistant Professor.
SCTR's Pune Institute Of Computer Technology.

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
ACADEMIC YEAR 2022-23

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

SCTR's Pune Institute of Computer Technology (PICT), Pune
Maharashtra 411043

CERTIFICATE

This is to certify that the internship report titled
"(Lung Cancer Detection and Classification)"
Submitted by
(Omkar R. Nevse)
(ExamNo. T190053203)

- has satisfactorily completed the curriculum-based internship under the guidance of Prof. Dr. R. Sreemathy and Prof. Ms. Ankita K. Patel, PICT towards the partial fulfilment of third-year Electronics and Telecommunication Engineering Semester VI, Academic Year 2022-23 of Savitribai Phule Pune University.

Dr. M.V.Munot

Internship Mentors Internship Coordinator Head of the Department (E&TE)

Dr. R. Sreemathy

Ms. Ankita K. Patel

Place:

Date

Acknowledgement

It gives me great pleasure to present the internship report on ”Lung Cancer Detection and Classification”.

I would like to take this opportunity to thank my internship guide Dr. R. Sreemathy ma’am and Ms. Ankita K. Patel ma’am for giving me all the help and guidance needed. I am really grateful for her kind support and valuable suggestions that proved to be beneficial in the overall completion of this internship.

I am thankful to our Head of the Electronics and Telecommunication Engineering Department, Dr. M.V.Munot, for her indispensable support and suggestions throughout the internship.

I would also genuinely like to express my gratitude to the Department Internship Coordinator for her constant guidance and support and for the timely resolution of the doubts related to the internship process.

Finally, I would like to thank my mentor, Dr. R. Sreemathy and Ms. Ankita K. Patel for their constant help and support during the internship.

Contents

List of figures

List of Tables

Title: Lung Cancer Detection and Classification

● Introduction

Lung cancer is the second most common cancer in both men and women that affects millions of people each year. Nearly 1 out of 4 cancer deaths are from lung cancer, more than colon, breast, and prostate cancers combined. Globally there were an estimated 2.1 million lung cancer cases and 1.8 million deaths in 2018.[1] Early detection of the cancer can allow for early treatment which significantly increases the chances of survival. Lung cancer screening is performed with a CT scan that collects hundreds of images to build a full 3D composite of the lung. Next, small growths called pulmonary nodules need to be detected. These nodules show up as small, circular structures on the CT scans.

In some cases, the nodules are not obvious and may take a trained eye and a considerable amount of time to detect. Building a machine learning algorithm that can automatically detect the nodules can save considerable time and money, thus opening the accessibility of prescreening, ultimately saving lives. Additionally, most pulmonary nodules are not cancerous as they can also be due to non-cancerous growths, scar tissue, or infections. The task is then to determine the features of a nodule that are associated with malignancy. Current state-of-the-art methods yield a 25% false positive rate in CT lung cancer screenings. A convolutional neural network may be used to determine the features associated with cancerous or non-cancerous pulmonary nodules and may reduce the false positive rate of CT lung cancer screenings.

- **Literature Survey:**

1. "Lung Cancer Detection using Deep Convolutional Networks" by Jelo Salomon et al. (2018). [2]

   This study provides an overview of deep learning approaches used for lung cancer detection on CT scans. It discusses various architectures and highlights their performance in terms of sensitivity and specificity.

2. "Automated Lung Cancer Detection Using Artificial Intelligence (AI) Deep Convolutional Neural Networks: A Narrative Literature Review" by Jaikaran Singh et al. (2017). [3]

   The paper reviews machine learning models used for lung cancer diagnosis and prognosis. It discusses different features, algorithms, and validation methods employed in these models, along with their respective performance metrics.

3. "Radiomics and artificial intelligence in lung cancer screening" by Franciszek Binczyk et al. (2021). [4]

   This comprehensive review explores the utilization of radiomics-based machine learning models for lung cancer diagnosis, prognosis, and treatment response prediction. It covers feature extraction techniques, model architectures, and performance evaluation metrics.

4. "Machine learning application in personalised lung cancer recurrence and survivability prediction" by Yang et al. (2020). [5]

   The study focuses on machine learning approaches for predicting lung cancer recurrence. It discusses different feature selection methods, algorithms, and model evaluation techniques applied to clinical and genomic data.

5. "Lung Cancer Survival Prediction via Machine Learning Regression, Classification, and Statistical Techniques" by James et al. (2021). [6]

   This research paper examines machine learning models for predicting overall survival in lung cancer patients. It explores various clinical and molecular features used in these models and evaluates their predictive accuracy.

6. "A systematic review and meta-analysis of diagnostic performance and physicians' perceptions of artificial intelligence (AI)-assisted CT diagnostic technology for the classification of pulmonary nodules" by Guo Huang et al. [7] (2021). The paper provides an overview of machine learning techniques employed for lung nodule classification and prediction. It compares different algorithms, feature selection methods, and performance measures used in the studies reviewed.

7. "Automated Extraction and Classification of Pulmonary Lung Nodules from CT Scans" by Mike Huang et al. (2018).

   This Project uses image segmentation and masking techniques which help in achieving a more efficient training process and improving the overall output of their model.

These literature sources provide a comprehensive overview of machine learning models utilized in lung cancer research, including detection, diagnosis, prognosis, recurrence prediction, survival prediction, nodule classification, treatment outcome prediction, and risk assessment.

● Problem statement

1. Perform image processing to extract nodule features from images of lung CT scans.
2. Design a deep learning model to predict whether or not a patient is diagnosed with lung cancer.

● Objectives and scope

Objectives:

1. Collect lung CT scan images and create a dataset.
2. Explore different existing deep learning and machine learning methods.
3. Develop a deep learning model for image classification.
4. Train model on CT scan images.

Scope:

This project focuses on developing a model for effective lung cancer detection and classification. The scope includes collection, preprocessing, model development, training and model evaluation.

The system should be capable of accepting CT scan images that will be utilized by the Deep Learning Model. The system should be able to detect lung cancer within the CT scan images that users have uploaded. The system should be able to provide information that our users can appropriately understand and gain insight from.

● Methodological details

1. Data Collection and Preprocessing:

- Identify or curate a dataset of paired text and corresponding images. This dataset should cover all three main types of lung cancer along with a dataset for normal healthy lungs and provide sufficient examples for training.

- Normalize and resize the images to a consistent resolution, ensuring compatibility with the model architecture.

- Label and store the images in separate folders according to their type.

- For the purpose of this project, a dataset of 1000 CT scan images is used for training and testing.

2. Model Architecture:

- The model uses the following techniques:

1. **VGG16** (Visual Geometry Group 16): VGG16 is a deep convolutional neural network trained on the imagenet dataset. Its trained weights are leveraged in the model. This step in the model acts as a feature extractor. The VGG16 architecture consists of 16 convolutional and fully connected layers. Here's a breakdown of its structure:

1.1 Input Layer: The network takes an input image of size 224x224 pixels.

1.2 Convolutional Layers: VGG16 consists of 13 convolutional layers, each followed by a rectified linear unit (ReLU) activation function and a 3x3 filter. The convolutional layers are responsible for capturing and learning various image features at different levels of abstraction.

1.3 Max Pooling Layers: After some of the convolutional layers, VGG16 includes max-pooling layers with a 2x2 window and a stride of 2. Max pooling helps reduce the spatial dimensions and extract the most important features while preserving their spatial relationships.

1.4 Fully Connected Layers: The network concludes with three fully connected layers. Each fully connected layer is followed by a ReLU activation, except for the last one. The final fully connected layer produces the output logits corresponding to different classes.

1.5 Softmax Activation: The output logits from the last fully connected layer are passed through a softmax activation function to obtain the class probabilities, indicating the likelihood of the image belonging to each class.

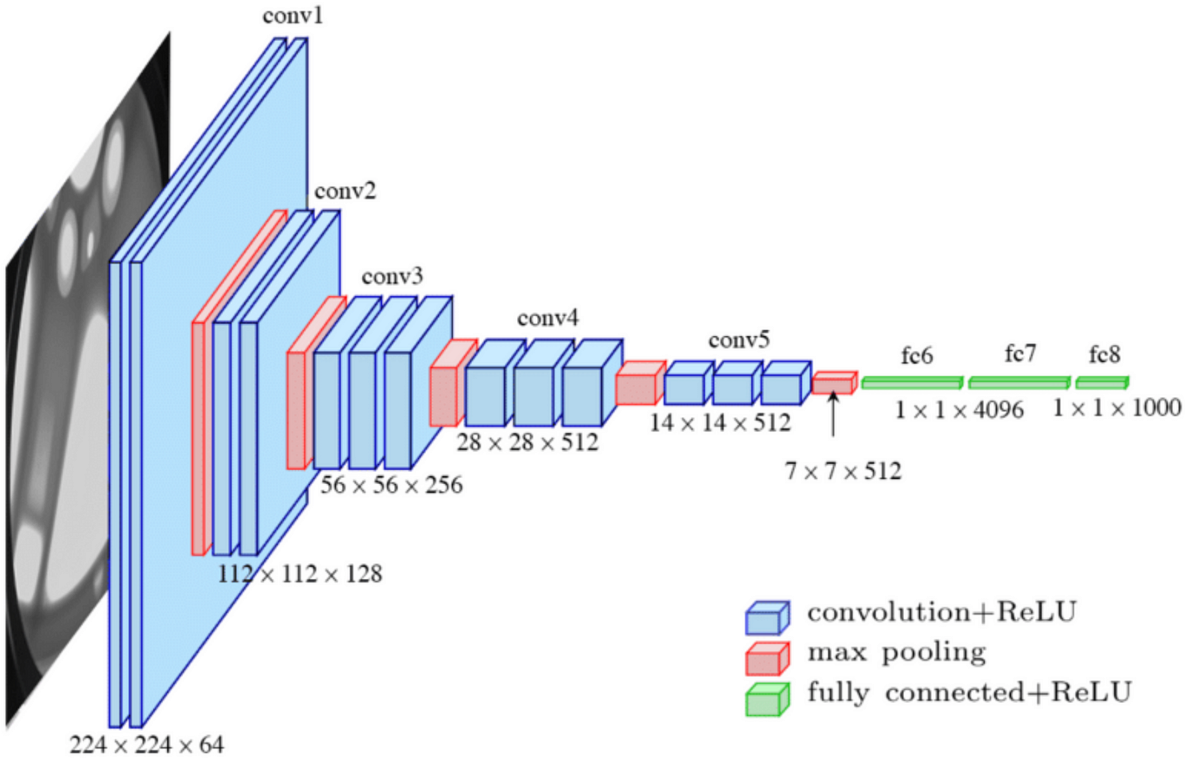


figure 1: VGG16 convolutional neural network visualization[8]

2. **Random Forest:** The Random Forest algorithm is a popular supervised machine learning technique used for both classification and regression tasks. In Random Forest, an ensemble of decision trees is created. Each decision tree is built using a random subset of the training data and a random subset of features. This randomness helps to reduce overfitting and improves the model's generalization ability.

3. **SVM (Support Vector Machine):** is a supervised machine learning algorithm used for classification and regression tasks. In classification, SVM finds an optimal hyperplane that separates data points belonging to different classes with the maximum margin. The hyperplane is chosen to maximise the distance between the closest data points from each class, known as support vectors.

4. **Decision Tree Classifier:** A Decision Tree Classifier is a machine learning algorithm used for both classification and regression tasks. It creates a tree-like model of decisions and their possible consequences based on the input features.

5. **Multinomial Naive Bayes:** Multinomial Naive Bayes is a variant of the Naive Bayes algorithm that is specifically designed for handling discrete features with a multinomial distribution. It is commonly used for text classification tasks where the features represent word counts or frequencies.

3. Training Process:

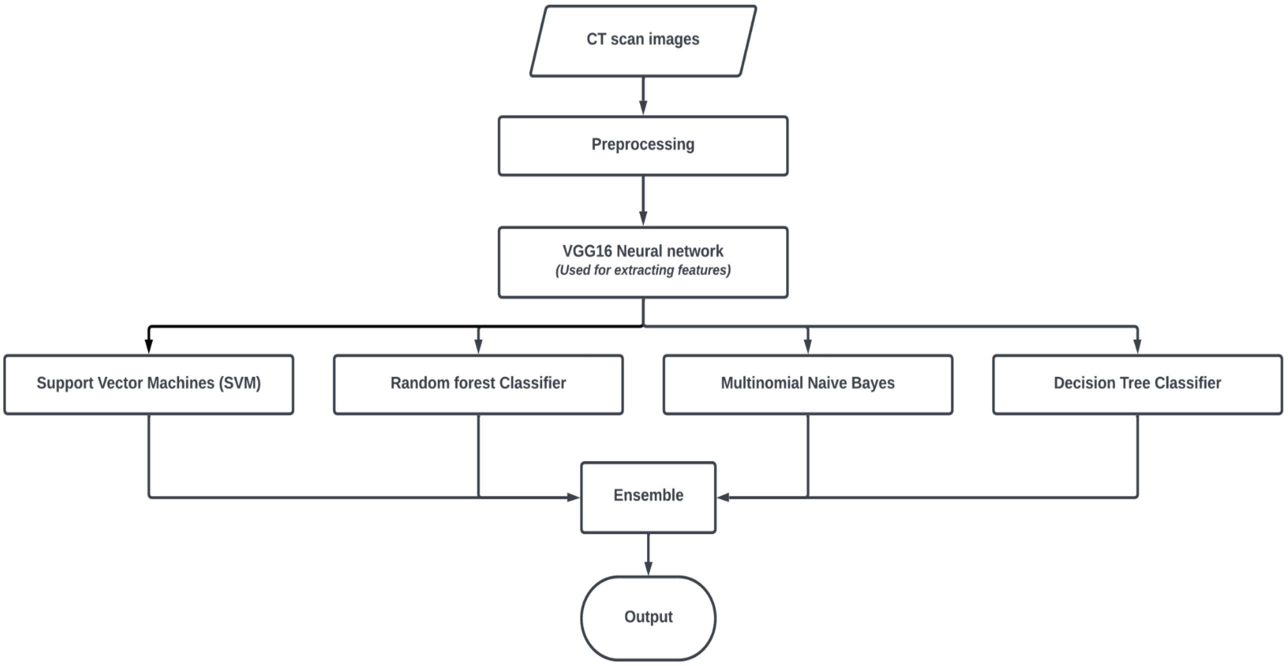


Figure 2: Flowchart of proposed model architecture

- Import required Dependencies
- Resize the images to 256x256
- Separate the dataset into Training, Testing and Validation folders (70-20-10 split)
- Normalise pixel values from 0-255 to 0-1. One-hot encode the label values for compatibility with VGG16 Neural Network.
- Load the VGG16 model without the fully connected layers, and use pre-trained ImageNet weights. Make Loaded layers non-trainable so that pretrained weights can be leveraged.
- Import RandomForestClassifier, SVM, Multinomial Naive Bayes and Decision Tree classifier from the sci-kit learn library
- Use the VGG16 model to extract features from the images. Pass the extracted features to all four classifiers.
- Train all four classifiers on these features.
- Ensemble the Predictions gained from RandomForest, SVM, MNB and DTC.
- Experiment with different hyperparameters to optimise the training process and achieve optimal classification accuracy.

4. Evaluation Metrics:

- The model is evaluated on the basis of its accuracy and the model's ability to correctly classify the input CT scan images as per the given four classes.
- Hyperparameters are then tweaked and the model is retrained until optimal desired accuracy is achieved.

● Usage of Modern engineering tools

The author had the opportunity to use specialised computers available on the institute campus. The model was trained with an Nvidia RTX 3090 GPU.

Software components used in the internship include:

1. Anaconda data science application
2. Tensorflow
3. Nvidia CUDA
4. Nvidia CuDNN libraries
5. Keras
6. Jupyter Notebooks and JupyterLab IDE.

● Outcome/ results of internship work (screenshots of work done)

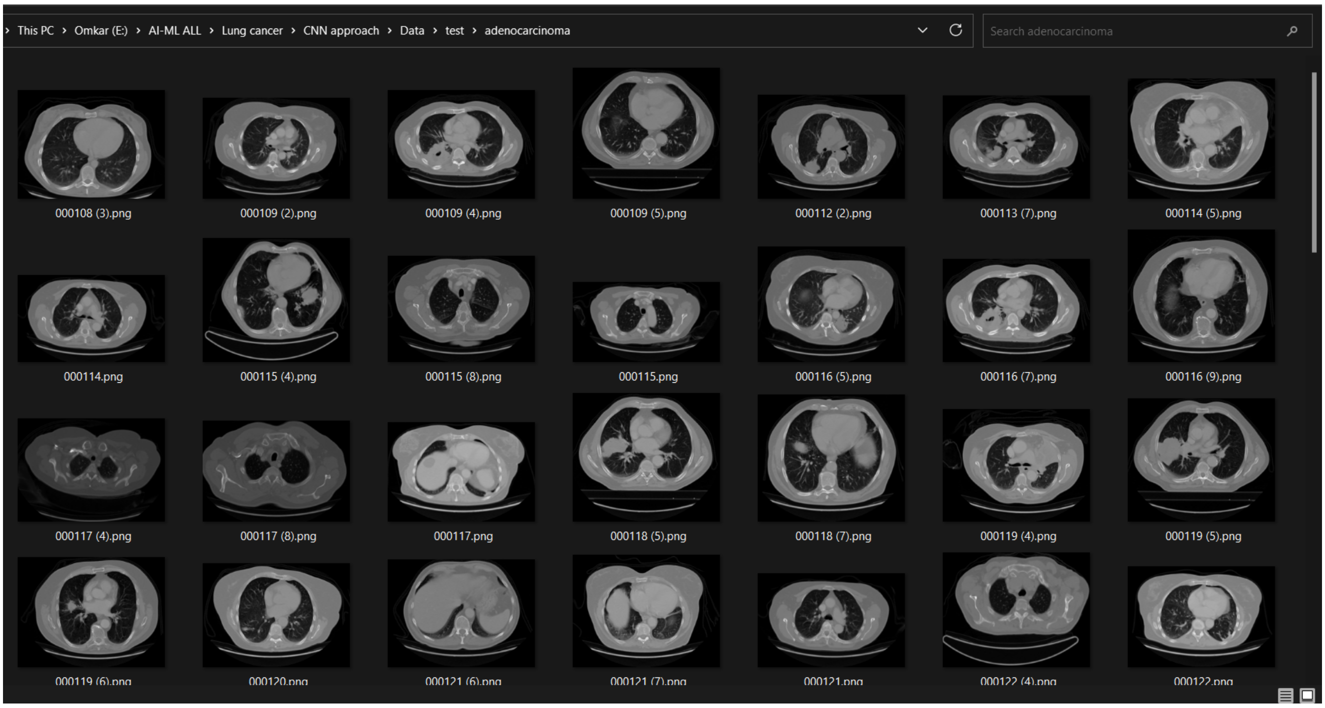
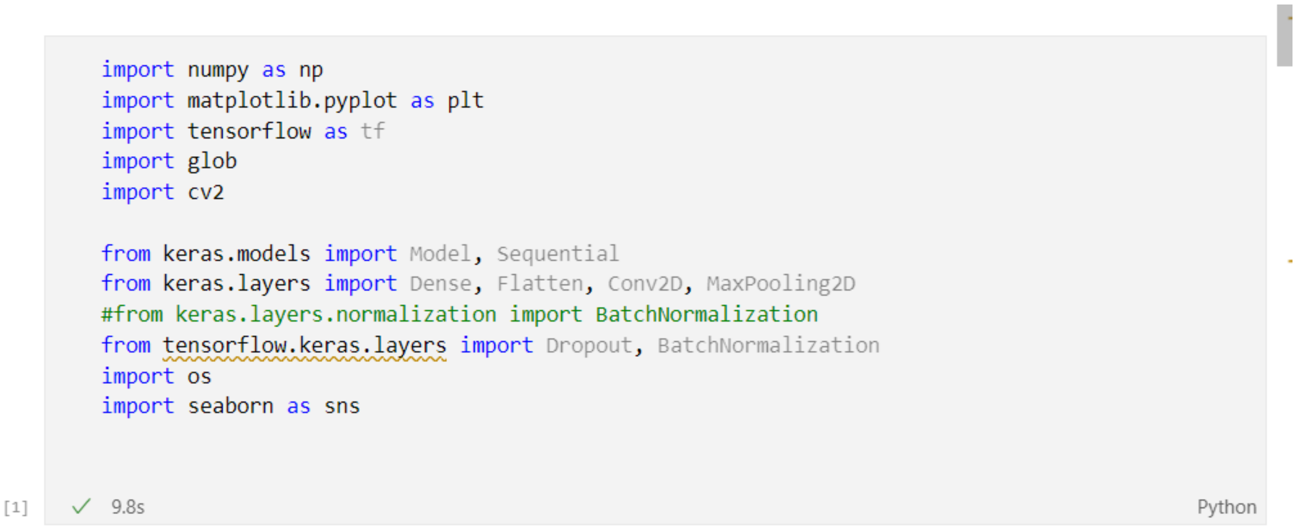


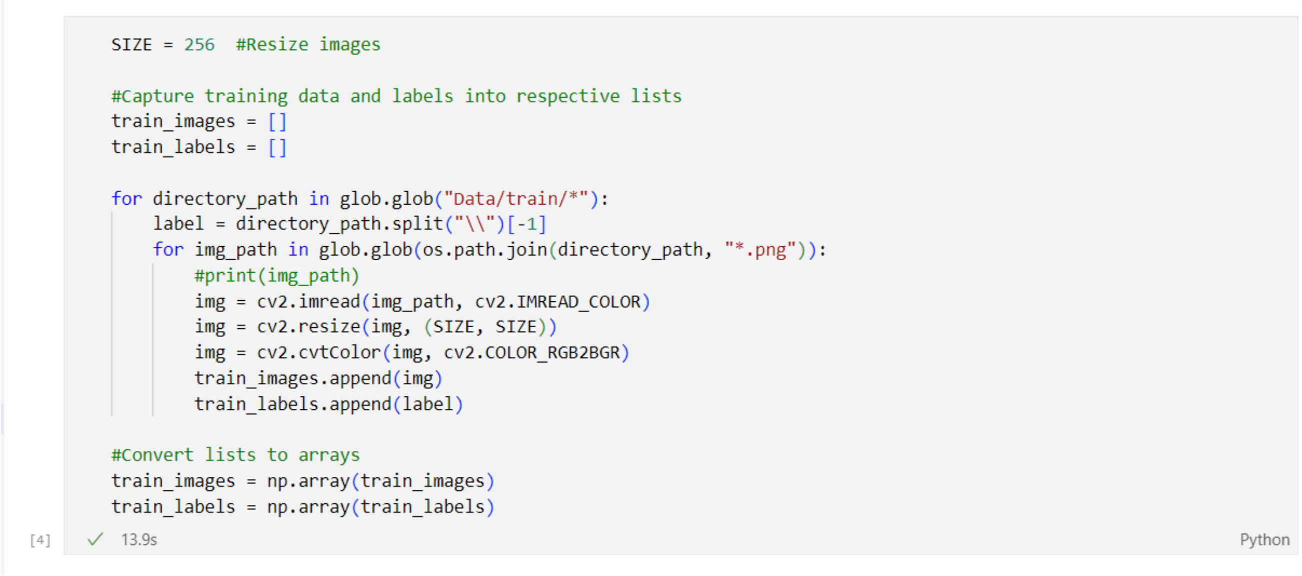
Figure 3: CT scan images used as training and testing dataset



```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import glob
import cv2

from keras.models import Model, Sequential
from keras.layers import Dense, Flatten, Conv2D, MaxPooling2D
#from keras.layers.normalization import BatchNormalization
from tensorflow.keras.layers import Dropout, BatchNormalization
import os
import seaborn as sns
```

```
[1]  ✓ 9.8s                                                          Python
```

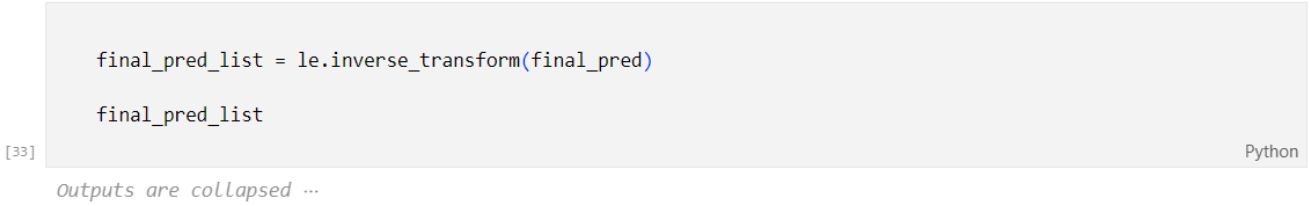
Figure: Imported dependencies



```
SIZE = 256  #Resize images

#Capture training data and labels into respective lists
train_images = []
train_labels = []

for directory_path in glob.glob("Data/train/*"):
    label = directory_path.split("\\")[-1]
    for img_path in glob.glob(os.path.join(directory_path, "*.png")):
        #print(img_path)
        img = cv2.imread(img_path, cv2.IMREAD_COLOR)
        img = cv2.resize(img, (SIZE, SIZE))
        img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
        train_images.append(img)
        train_labels.append(label)

#Convert lists to arrays
train_images = np.array(train_images)
train_labels = np.array(train_labels)
```

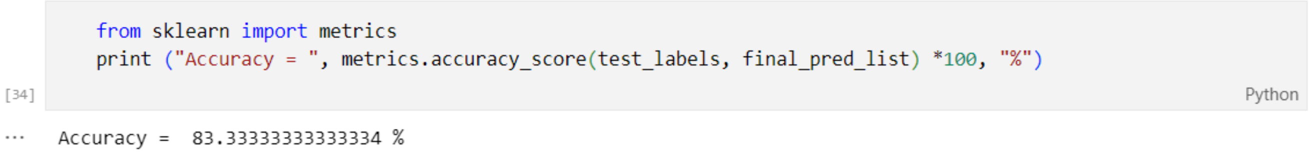
```
[4]  ✓ 13.9s                                                         Python
```

Figure: preprocessing input images



```
final_pred_list = le.inverse_transform(final_pred)

final_pred_list
```

```
[33]                                                                 Python
Outputs are collapsed ...
```



```
from sklearn import metrics
print ("Accuracy = ", metrics.accuracy_score(test_labels, final_pred_list) *100, "%")
```

```
[34]                                                                 Python
...  Accuracy =  83.33333333333334 %
```

Figure: Final output accuracy

- Comparing accuracy between different ML algorithms and DL classifiers

ML algorithms	SVM	Random Forest	Multinomial NB	Decision Tree Classifier	Ensemble
Detection Accuracy	88.89%	81.95%	70%	57%	83.4%

Table 1: Using VGG16

ML algorithms	SVM	Random Forest	Multinomial NB	Decision Tree Classifier	Ensemble
Detection Accuracy	88.89%	79.17%	68.05%	61.11%	81.95%

Table 2: Using DenseNet201

ML algorithms	SVM	Random Forest	Multinomial NB	Decision Tree Classifier	Ensemble
Detection Accuracy	86.11%	73.69%	66.67%	62.5%	80.55%

Table 3: Using Xception

ML algorithms	SVM	Random Forest	Multinomial NB	Decision Tree Classifier	Ensemble
Detection Accuracy	45.83%	63.89%	47.22%	48.66%	56.94%

Table 4: Using ResNet50

As seen above, the model leveraging VGG16 weights manages to get better accuracy as compared to others.

- **Any achievement (Job opportunity, project sponsorship, patent, commercial product, research publications, pre-placement offers, a strong professional network etc.)**

The author got the opportunity to learn new topics in the domain along with increasing his professional network in the domain of research-oriented machine learning and deep learning. The author is also moving ahead with publishing his research in a prominent publication.

● References:

1. Cancer Statistics: https://canceratlas.cancer.org/the-burden/lung-cancer/

2. Jelo Salomon, Bianca Schoen Phelan (2018). Lung Cancer Detection using Deep Convolutional Networks. *Cureus*, *12*(8). http://dx.doi.org/10.13140/RG.2.2.33602.27841

3. Sathyakumar, K., Munoz, M., Singh, J., Hussain, N., & Babu, B. A. (2020). Automated Lung Cancer Detection Using Artificial Intelligence (AI) Deep Convolutional Neural Networks: A Narrative Literature Review. *Cureus*, *12*(8) https://doi.org/10.7759%2Fcureus.10017

4. Binczyk, F., Prazuch, W., Bozek, P., & Polanska, J. (2021). Radiomics and artificial intelligence in lung cancer screening. *Translational Lung Cancer Research*, *10*(2), 1186-1199. https://doi.org/10.21037/tlcr-20-708

5. Yang, Y., Xu, L., Sun, L., Zhang, P., & Farid, S. S. (2022). Machine learning application in personalised lung cancer recurrence and survivability prediction. *Computational and Structural Biotechnology Journal*, *20*, 1811-1820. https://doi.org/10.1016/j.csbj.2022.03.035

6. Bartholomai, J. A., & Frieboes, H. B. (2018). Lung Cancer Survival Prediction via Machine Learning Regression, Classification, and Statistical Techniques. *Proceedings of the ... IEEE International Symposium on Signal Processing and Information Technology. IEEE International Symposium on Signal Processing and Information Technology*, *2018*, 632. https://doi.org/10.1109/ISSPIT.2018.8642753

7. Huang, G., Wei, X., Tang, H., Bai, F., Lin, X., & Xue, D. (2021). A systematic review and meta-analysis of diagnostic performance and physicians' perceptions of artificial intelligence (AI)-assisted CT diagnostic technology for the classification of pulmonary nodules. *Journal of Thoracic Disease*, *13*(8), 4797-4811. https://doi.org/10.21037/jtd-21-810

8. An overview of VGG16 and NiN models: https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484