A Deep Learning Approach to Detect Lung Cancer Using Alexnet and kNN

Submitted by

Md. Maksudur Rahman

163-35-1778

Department of Software Engineering

Daffodil International University

Supervised by

Md. Shohel Arman

Lecturer

Department of Software Engineering

Daffodil International University

This Thesis report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Science in Software Engineering.

© All right Reserved by Daffodil International University

© Daffodil International University

ACKNOWLEDGEMENT

At first, I would like to thanks the Almighty Allah Who gives me the courage and keeps me healthy to successfully complete this thesis.

Secondly, I am grateful to my honorable teacher **Md. Shohel Arman**, **Lecturer**, **Department of Software Engineering, Daffodil International University** for his boundless calmness, scholastic guidance, non-stop inspiration, constant and active oversight, organic criticism, precious advice, reading lots of inferior disasters and fixing them at every level have made it feasible to successfully finish this thesis.

I am also thankful to all the other faculties and staff individuals from our department for their co-activity and help.

© Daffodil International University

TABLE OF CONTENT

© Daffodil International University

© Daffodil International University

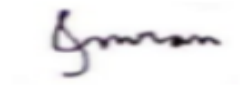
TABLE OF FIGURES

© Daffodil International University

APPROVAL

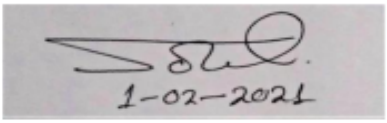
This **thesis** titled on "**A Deep Learning Approach to Detect Lung Cancer Using Alexnet and kNN**", submitted by Student Name (ID: **163-35-1778**) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University



Internal Examiner 1

Md. Khaled Sohel
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Md. Maruf Hassan
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

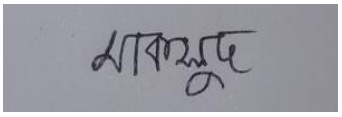
External Examiner

Prof. Dr. Mohammod Abul Kaashem
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering and Technology, Gazipur

© Daffodil International University

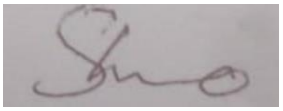
DECLARATION

I therefore declare that I have done this thesis under the oversight of **Md. Shohel Arman, Lecturer, Department of Software Engineering, Daffodil International University.** I also declare that neither entire record nor any portion of this record has been submitted somewhere else for award of any degree.



…….
Md. Maksudur Rahman
ID: 163-35-1778
Batch: 21st Batch
Department of Software Engineering
Daffodil International University

Certified by:



…….
Md. Shohel Arman
Lecturer
Department of Software Engineering
Daffodil International University

© Daffodil International University

ABSTRACT

In the race of all cancerous diseases, lung cancer is in the first place. Every year lots of people died because of cancer and lung cancer is playing the leading role among them. In the year of 2018, 9.6 million people died because of cancer where 1.76 million death occurred due to lung cancer. In this study, we experiment with a deep learning model with kNN classifier to extend the success rate in diagnosing lung cancer. The dataset used in this study is a publicly accessible resource SPIE-AAPM. We used data augmentation on the training dataset to expand the dataset and convolutional neural network (CNN) to extract the related features. Extracted features from CNN used as input to the kNN classifier with cross-validation. The experiment hit accuracy of 90% by predicting the dataset with the help of selected features and kNN classifier.

© Daffodil International University

CHAPTER 1

INTRODUCTION

1.1 Background

Cancer is an unintentional enhancement of cells which could outspread in an uncontrolled manner, and sometimes it spreads. From hundreds of cancer diseases, lung cancer has turned out one of the most common cancer in the world [18]. The beginning of lung cancer happens from the unusual development of lung cells. The lung absorbs oxygen when the human breath in, and it discharges carbon dioxide when they breath out. So, the lung of the human body is a very important and sensitive organ for humans. A very vast amount of people dies every year because of lung cancer [1]. According to the Global Cancer Statistics, lung cancer and breast cancer had the same amount of new cases detected (lung cancer 11.6% of total cases and breast cancer also 11.6% of total cases). But, in death ratio lung cancer is three times greater than the breast cancer (lung cancer 18.4% of total deaths and breast cancer 6.6% of total deaths). Total 2.1 million people were affected and 1.76 million people died because of lung cancer worldwide in 2018 [19]. The survival rate is too low because of late diagnosis. Hence, early and faultless diagnose of lung cancer may reduce the number of deaths.

The use of artificial intelligence in medical department is increasing day by day at a massive rate. In biomedical field AI based solutions are having great success. In recent few years, deep learning is proved as a more effective technique than machine learning because of the power of relative feature extraction in the biomedical field.

© Daffodil International University

1.3 Motivation of the Research

Every year lots of people is beaten by lung cancer and passes away to the afterlife. In 2018 2.1 million people affected by lung cancer and from there 1.76 million people stops their journey and passed away to the afterlife. Due to late diagnosis only, few people can survive it. Early diagnosis may reduce the number of death and increase the survival rate. There are various models were developed to identify precisely this disease earlier.

1.3 Problem Statement

It is really difficult and time consuming for a field expert to diagnose the lung cancer from CT scan images manually. Sometimes they cannot even diagnose the cancer precisely. As a result, the patient has to face the confusing situation. The manual process of diagnosing the lung cancer is also quite expensive.

1.4 Research Questions

According to the background, motivation and problem statement the following questions are raised:

- Does the model used in this study can detect lung cancer accurately?
- Does the technique can provide better result comparing to the existing solutions?
- Is this model more effective than others when it comes to comparison?

1.5 Research Objectives

- To propose a pretrained model which can perform with better perfection.
- To produce a significant model which can help to detect lung cancer precisely.
- To provide a better solution which can provide better results among the existing solutions.

© Daffodil International University

1.6 Research Scope

The purpose of this study is to detect only lung cancer in CT scan images. Any other diseases or any other kind of cancer cannot be diagnosed using this system. The input images must be CT scan images of the lung, not other images like CT images of the abdomen, CT images of the brain, x-ray of the chest, etc. This implementation detects like binary 0 and 1, whether the image is cancerous or non-cancerous. The system cannot detect the stages of lung cancer. The system can be used in a hospital where lung cancer patients exist or come for diagnosing lung cancer.

1.5 Paper Organization:

The reference section was organized using APA referencing procedure. In the organization of this paper, six chapters are included as follows:

In chapter 1, background, motivation, problem statement, research question, research objective, and research scope of this research are discussed. In chapter 2, earlier reading on lung cancer detection are discussed. In chapter 3, description of the dataset implemented in this study. In chapter 4, a brief discussion of the methodology applied in this study. In chapter 5, the implementation process and step by step improvement have been discussed. In chapter 6 and 7, the results and discussion of this experiment and conclusion of the study respectively.

© Daffodil International University

CHAPTER 2

LITERATURE REVIEW

D. P. Kaucha et al [1], detect early-stage lung cancer in CT scan images using an image processing technique where the images were pre-processed by segmentation of the ROI of the lung and GLCM to extract the features. Then extracted features were fed into SVM classifier and then they got an accuracy of 95.16%, sensitivity of 98.21% and specificity of 78.69%.

Suren Makaju et al [2], implement image processing and machine learning to detect lung cancer in CT scan images. They used watershed segmentation and SVM classifier and they achieved an accuracy of 92%, sensitivity of 100% and specificity of 50%.

Bhatia et al [3], detect lung cancer from CT scan images using deep residual learning where Unet and Resnet models were used for feature extraction and XGBoost and Random Forest used for classification. They achieved an accuracy of 84% utilizing group of Random Forest and XGBoost classifier on LIDC-IDRI dataset.

Rehman et al [4], built a lung cancer detection model using Convolutional Neural Networks (CNN) and the dataset used in this study was taken from Japanese Society of Radiological Technology (JSRT) and they got an accuracy of 88%.

Shakeel et al [5], predict lung cancer using improved deep neural network and ensemble classifier on cancer imaging archive (CIA) dataset. In this paper, they preprocessed images using multilevel brightness preserving approach and then segmentation using an improved clustering technique. Then feature extraction and feature selection and then selected features applied in ensemble classifier for prediction. They got an accuracy of 96%, specificity of 98%, precision of 97% and recall of 98%.

Sujitha, R., & Seenivasagam [6], introduce a technique to detect lung cancer stages by examining machine learning and apache spark. They applied binary classification with T-BMSVM to detect lung cancer. They got 86% accuracy in it.

© Daffodil International University

Potghan et al [7], detect lung cancer using CT scan images from LIDC-IDRI dataset. In this study, separation of lung volume is conducted by a k-means clustering algorithm. Feature extraction was applied using deep learning architecture. And then classification performed using k NN and MLP. The approach accomplished 98.30% accuracy with kNN classifier and 98.31% accuracy with MLP classifier.

Liu, K., & Kang, G [8], used a Multiview CNN for the classification of lung cancer on LIDC-IDRI dataset. In MV-CNN 7-view sides were noticed. In binary classification, they acquire the undermost error rate is 5.41%. In the ternary classification, the undermost error rate found is 13.91%.

Cengil, E., & Cinar, A [10], conduct a deep learning-based technique to detect lung cancer. In this study, they used SPIE-AAPM dataset. Convolutional neural networks (CNN) was implemented for image classification. Classification accuracy accomplished in this study was 70%.

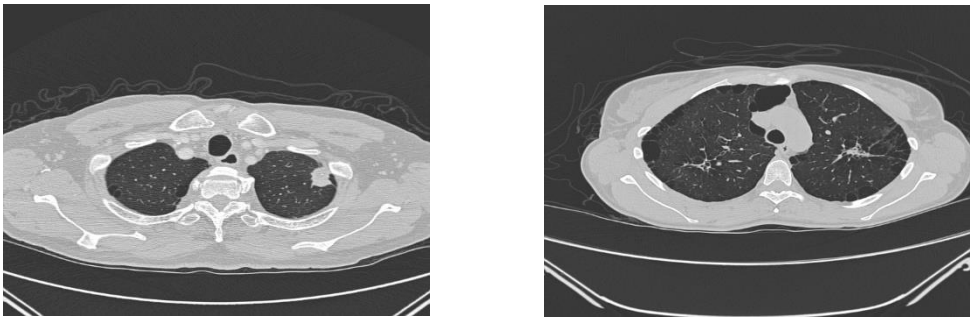
Toğaçar et al [9], used deep learning approach with mRMR feature selection to detect lung cancer. They used a public dataset TCGA-LUAD in their work. LeNet, AlexNet, VGG-16 deep learning models were used in this study where AlexNet performed better result. As a result, the features from the utmost entirely associated layer of AlexNet were implemented as input to the different classifier (LR, LDA, DT, SVM, kNN, Softmax). From six different classifier kNN performed better, so they used kNN for classification. Then they used mRMR feature selection to improve the model. They perform cross-validation in each experiment. After all of this, they accomplished an accuracy, sensitivity, and specificity of 99.51%, 99.32%, 99.71% respectively.

© Daffodil International University

CHAPTER 3

DATASET

The dataset used in this experiment is a publicly accessible resource. SPIE – with the help of American Association of Physicists in Medicine (AAPM) and the National Cancer Institute (NCI) able to create the dataset. So that others can apply their methods or algorithms on this dataset. CT scan images picked up from 70 different patients for this dataset [11]. Total of 81 labelled images was used in this experiment in which 42 images are benign and, 39 images are malignant. The image format is in DICOM. The images were changed to JPEG format for the purpose of image preprocessing. Two sample images (non-cancerous and cancerous) from the dataset is displayed in figure 1.



(a) (b)

Figure 1: (a) Non-Cancerous, (b) Cancerous

© Daffodil International University

CHAPTER 4

METHODOLOGY

4.1 Data Augmentation

Data augmentation is a strategy that utilized in AI to improve model precision, speculation, and to control overfitting [12]. The image augmentation procedures used in this study are acknowledged by the Keras library in Python [13]. In image augmentation, rotation, width and height shift, shear, zoom, horizontal flip and fill operations are implemented. The rotation range fix to 40 and width and height shift range, the sheer range and zoom range fix to 0.2 and horizontal flip fix to true. The augmentation was only applied to training dataset and on the other hand, the test dataset used without augmentation.

4.2 Convolutional Neural Network (CNN)

In this study, AlexNet was used from deep learning models. AlexNet was fundamentally planned by Alex Krizhevsky. It was disclosed with Ilya Sutskever and Krizhevsky's doctoral counsellor Geoffrey Hinton, and is a Convolutional Neural Network or CNN. In the wake of contending in ImageNet Large Scale Visual Recognition Challenge, AlexNet shot to popularity. It accomplished a top-5 mistake of 15.3%. It was trained by 1.2 million images [14]. From three deep learning methods, AlexNet performed better to detect lung cancer [9]. The AlexNet architecture formed by five convolutional layers, some of these are ended by maximum pooling layers and afterwards three entirely joint layers lastly a 1000-way softmax classifier. The AlexNet architecture used in this study detailed in table 1.

© Daffodil International University

Table 1: AlexNet architecture

Layer		Feature Map	Size	Kernal Slze	Stride	Activation
Input	Image	1	227x227x3	-		-
1	Convolution	96	55x55x96	11x11	4	relu
	Max pooling	96	27x27x96	3x3	2	relu
2	Convolution	256	27x27x256	5x5	1	relu
	Max pooling	256	13x13x256	3x3	2	relu
3	Convolution	384	13x13x384	3x3	1	relu
4	Convolution	384	13x13x384	3x3	1	relu
5	Convolution	256	13x13x256	3x3	1	relu
	Max pooling	256	6x6x256	3x3	2	relu
6	FC	-	4096	-	-	relu
7	FC	-	4096	-	-	relu
8	FC	-	1000	-	-	relu
output	Softmax	-	1000	-	-	Softmax

4.3 Optimization

The primary reason for optimization techniques is to refresh the loads at each level until the best training in CNN design is figured it out. Every technique plays out the update measure with its own calculation. In this study, we used stochastic gradient descent (SGD) optimization technique in Keras [17]. Stochastic gradient descent (SGD) interestingly plays out a boundary update for each training instance x and label y [15][16].

© Daffodil International University

4.4 Machine Learning Model

Machine learning model is employed to separate the classes. In this experiment, we used the K-Nearest Neighbor algorithm (K-NN) to classify lung cancer. When it comes to classification, kNN machine learning algorithm is going to set which objects goes under which class by investigating the features of the sample. According to the maximum votes of the neighbors, the sample is sent to the appropriate class. The kNN performed outstanding from various machine learning algorithms [9]. In this experiment, we observed the value of k in the range of 1 to 7 discussed in figure 4. And finally, the value of k is set to 3.

4.5 Proposed Methodology

The dataset applied in this study is a public dataset from SPIE-AAPM. Firstly, the data augmentation technique applied to training dataset and the test data set were kept as it was. Then the AlexNet deep learning architecture applied to the augmented dataset. Parameters for all the layers of AlexNet kept as mentioned in table1. And then stochastic gradient descent (SGD) optimization technique applied for feature optimization. After that relative features were turned out from the last fully-connected layer of AlexNet. Lastly, extracted features from AlexNet are applied to kNN with cross validation for separating cancerous and non-cancerous images. The flowchart of this model is described in figure 2.

© Daffodil International University

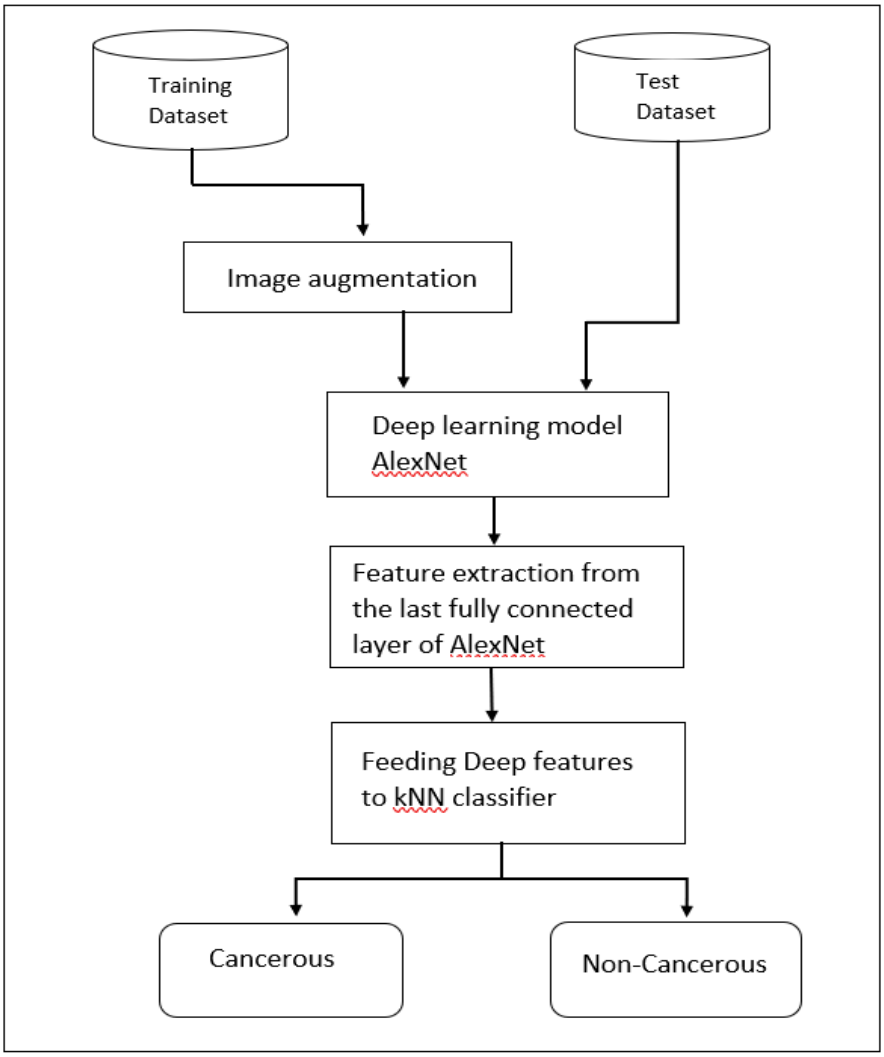


Figure 2: Design of the model

© Daffodil International University

CHAPTER 5

IMPLEMENTATION

4.1 Dataset Preparation

The dataset implemented in this study was downloaded from the cancer imaging archive (TCIA) by using NBIA data retrieving software. The dataset contains a total 22,489 CT scan images from 70 different patients. The DICOM images were opened using a DICOM viewer software and converted to JPG format for using as input to the model. Total 81 cancerous and noncancerous images were separated from the whole dataset using their given xls sheet. Then the separated images divided into two folder one for training and another for testing. In training folder, cancerous images were kept into one folder named as ‘lc’ and non-cancerous images were kept into another folder named as ‘be’ and for test dataset, images were kept as same as training dataset.

4.2 Dataset labeling

As previously discussed, there are no disease stages being identified in this study. as a result, the dataset was labelled into binary labelling. In the labelling non-cancerous images were considered as 0 and cancerous images were considered as 1.

© Daffodil International University

4.3 Training Improvement

In the first step, all the images converted to 227x227 size and implemented to alexnet architecture presented in table 1. In this step image augmentation was not used and the epoch was set to 30. Consequently, the training accuracy and loss respectively accomplished 1.28 and 84.51%. In the second step, image augmentation was used and the epoch was set as before it was. Consequently, the training accuracy and loss respectively accomplished 0.0894 and 97.18%. In the third step, the epoch was changed and set to 60. Consequently, the training accuracy and loss respectively downgraded to 0.3305 and 92.96%. In the fourth step, data augmentation was used and the epoch was set to 120. Consequently, the training accuracy and loss respectively accomplished 0.0063 and 100%. All the steps of training accuracy improvement are presented in table 2.

Table 2: Training improvement

Process	Epoch	Loss	Accuracy (%)
Alexnet	30	1.28	84.51
Image Augmentation + Alexnet	30	0.0894	97.18
Image Augmentation + Alexnet	60	0.3305	92.96
Image Augmentation + Alexnet	120	0.0063	100

4.3 Accuracy Improvement

In the first step, the model accuracy was evaluated by only using alexnet and the accuracy accomplished 60%. In the Second step, image augmentation was applied to the dataset and then evaluated the result and the accuracy accomplished by the model was 70%. In the final step image augmentation, alexnet, feature extraction from alexnet and kNN classifier was applied and the accuracy reached to 90%.

© Daffodil International University

Table 3: Accuracy improvement

Process	Epoch	Accuracy (%)
Alexnet	30	60
Image augmentation + Alexnet	30	70
Image augmentation + Alexnet + Feature extraction + kNN	120	90

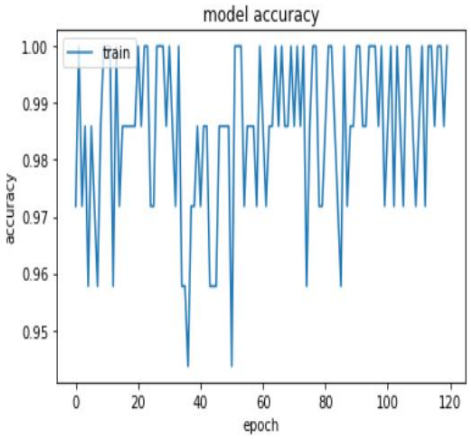
© Daffodil International University

CHAPTER 6

RESULTS AND DISCUSSION

The image augmentation and AlexNet architecture was implemented by using anaconda jupyter notebook and python. The parameters used in AlexNet described in table 1. The batch size was set as default mini batch size 32. For training, the number of epochs was 120. A simple rescaling is performed for every image before applying them as an input to CNN. The operating system used to run the experiment was 64-bit windows 10. And the hardware of the computer was core i5 and 4GB RAM.

In the training section 71 images taken from dataset for training. And then 120 epochs being ran for them and the training accuracy accomplished 100%. The detailed of the training along per epoch of the experiment is presenting in figure 3.




```
3/3 [==============================] - 1s 391ms/step - loss: 0.0063 - accuracy: 1.0000
[0.006323544308543205, 1.0]
```

Figure 3: Training accuracy

After training, the input of the AlexNet applied to a new model to extract the features from utmost entirely associated (softmax) layer. And then the extracted features were being applied as input to the kNN classifier with cross validation. For kNN classifier we observed the value of the k from range 1 to 7 figure 4.

© Daffodil International University

```
Out[22]: Text(0, 0.5, 'cross-Validation Accuracy')
```

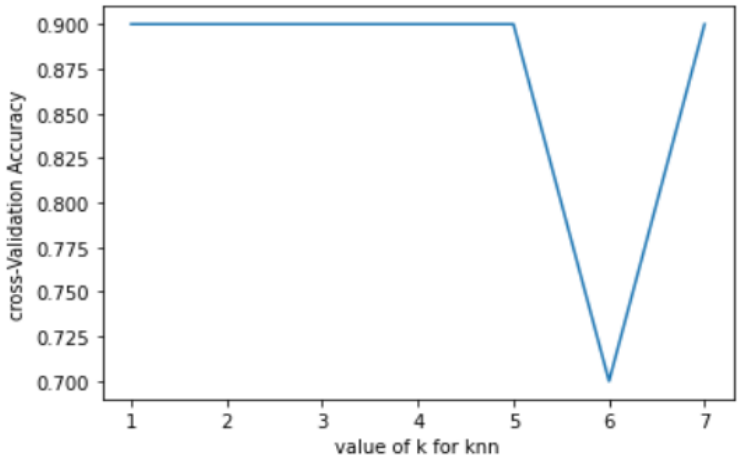


Figure 4: Accuracy graph for different value of k

According to figure 4, the highest accuracy provided by k was in between 1 to 5. So, we set the value of k to 3. And a 5-fold cross-validation was implemented in this experiment. Then we calculated the accuracy, precision, recall and, f1 score for the proposed model. The performances of the model are described in detail in confusion matrix in figure 4. The model accomplished an accuracy of 90%, precision of 83%, recall of 100% and, f1 score of 90.9% where

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{FP+TP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1 Score} = \frac{2xTP}{2xTP+FP+FN}$$

TP = True Positive

TN = True Negative

FP = False Positive

© Daffodil International University

FN = False Negative

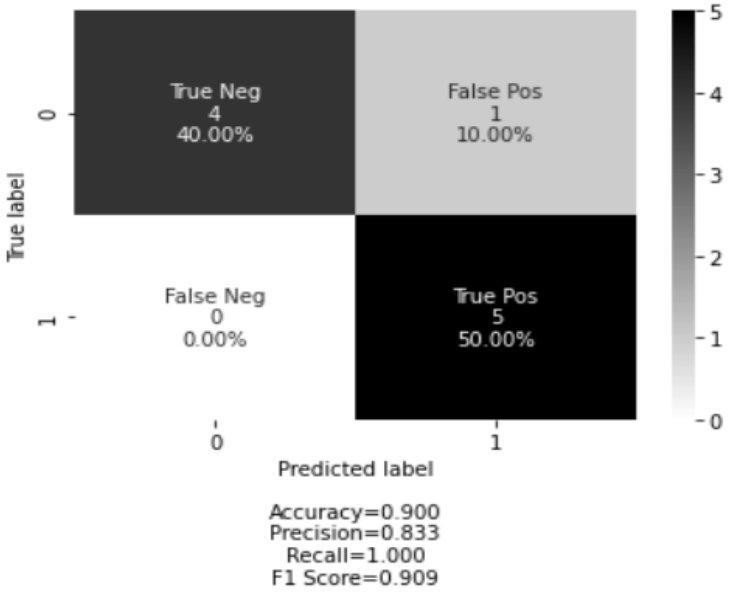


Figure 5: Confusion matrix of system performance

Lung cancer is a leading scorer type cancer from various cancer diseases exist in the world. The outbreak of this dangerous disease is growing gradually. If this disease is not identified earlier, the possibility of death is quite high. The immediate identification of this disease is connected with fast and faultless outcomes of the image processing techniques. On this occasion, the CNN models have an incredible favorable position regarding giving quicker and better outcomes contrasted with machine learning models. Moreover, the great thing of this model is the worthiness of extracting the relative local features. In addition, it can reflect this worthiness to the classification method. Therefore, numerous studies concentrated on machine learning and deep learning techniques to early identification of lung cancer. Even though comparison with one study is not effective because of various datasets and techniques.

© Daffodil International University

Cengil, E., & Cinar, A [10], implement CNN to classify the lung cancer as benign and malignant, on the same dataset we used in our study. They used 3D CNN architecture for classification. The success rate of their model was 70%. Rehman et al [4], also used CNN on JSRT dataset and the images were preprocessed into gray-scale images. They acquired an accuracy of 88%. D. P. Kaucha et al [1], convert images to gray scale to enhance the quality of the image in image processing. The DWT technique is used in the ROI images for segmentation process. The Gray Level Co-occurrence Matrix (GLCM) implemented to extract the features and SVM classifier to classify the lung cancer. This model accomplished an accuracy and sensitivity of 95.16%, 98.21% respectively. Potghan et al [7], also used GLCM for feature extraction. In classification, kNN and MLP classifier was used and achieved an accuracy of 98.30% and 98.31% respectively. Toğaçar et al [9], experimented three deep learning models with different classifier on a public dataset TCGA-LUAD. In their fourth experiment, AlexNet and kNN achieved 98.74%. In their last experiment they used mRMR feature selection process to improve the model and the model performs better than previous and the accuracy, sensitivity, and specificity of the model got in last experiment respectively 99.51%, 99.32% and, 99.71%.

In this experiment, we accomplished an accuracy of 90% and recall of 100%. Some of the models mentioned above provide better accuracy than our experiment but, in this experiment, we have better recall (sensitivity) than their model.

© Daffodil International University

CHAPTER 6

CONCLUSION

In this study, a deep learning technique with kNN is used to predict lung cancer in CT scan images. The CT scan images were preprocessed using image augmentation technique. Then the augmented dataset is used in deep learning model (AlexNet) for training. And then features were extracted from the last layer of AlexNet and the extracted features from AlexNet was applied as input to kNN classifier. And then the 5-fold cross validation was also applied to the experiment to achieve generalized result. The model implemented on a publicly available SIPE-AAPM dataset consisting total 81 labeled chest CT images. The experiment achieved an accuracy of 90%, precision of 83%, recall of 100% and, f1 score of 90.9%. The limitation of this study, because of hardware lacking (graphics card) of our computer, we could not apply feature selection technique. The dataset used in this study was small, in future, we will work with a big dataset and an improved model.

© Daffodil International University

REFERENCES

[1] Kaucha, D. P., Prasad, P. W. C., Alsadoon, A., Elchouemi, A., & Sreedharan, S. (2017, September). Early detection of lung cancer using SVM classifier in biomedical image processing. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)* (pp. 3143-3148). IEEE.

[2] Makaju, S., Prasad, P. W. C., Alsadoon, A., Singh, A. K., & Elchouemi, A. (2018). Lung cancer detection using CT scan images. *Procedia Computer Science*, *125*, 107-114.

[3] Bhatia, S., Sinha, Y., & Goel, L. (2019). Lung cancer detection: A deep learning approach. In *Soft Computing for Problem Solving* (pp. 699-705). Springer, Singapore.

[4] Rehman, M. Z., Nawi, N. M., Tanveer, A., Zafar, H., Munir, H., & Hassan, S. (2020, January). Lungs Cancer Nodules Detection from CT Scan Images with Convolutional Neural Networks. In *International Conference on Soft Computing and Data Mining* (pp. 382-391). Springer, Cham.

[5] Shakeel, P. M., Burhanuddin, M. A., & Desa, M. I. (2020). Automatic lung cancer detection from CT image using improved deep neural network and ensemble classifier. *Neural Computing and Applications*, 1-14.

[6] Sujitha, R., & Seenivasagam, V. Classification of lung cancer stages with machine learning over big data healthcare framework.

[7] Potghan, S., Rajamenakshi, R., & Bhise, A. (2018, March). Multi-Layer Perceptron Based Lung Tumor Classification. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 499-502). IEEE.

[8] Liu, K., & Kang, G. (2017). Multiview convolutional neural networks for lung nodule classification. *International Journal of Imaging Systems and Technology*, *27*(1), 12-22.

[9] Toğaçar, M., Ergen, B., & Cömert, Z. (2020). Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks. *Biocybernetics and Biomedical Engineering*, *40*(1), 23-39.

© Daffodil International University

[10] Cengil, E., & Cinar, A. (2018, September). A Deep Learning Based Approach to Lung Cancer Identification. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)* (pp. 1-5). IEEE.

[11] Armato III, Samuel G.; Hadjiiski, Lubomir; Tourassi, Georgia D.; Drukker, Karen; Giger, Maryellen L.; Li, Feng; Redmond, George; Farahani, Keyvan; Kirby, Justin S.; Clarke, Laurence P. (2015). **SPIE-AAPM-NCI Lung Nodule Classification Challenge Dataset**. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2015.UZLSU3FL

[12] Bloice, M. D., Stocker, C., & Holzinger, A. (2017). Augmentor: an image augmentation library for machine learning. *arXiv preprint arXiv:1708.04680*.

[13] Building powerful image classification models using very little data n.d. https://blog.keras.io/building-powerful-imageclassification-models-using-very-little-data.html (accessed July 20, 2020).

[14] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

[15] An overview of gradient descent optimization algorithms https://ruder.io/optimizing-gradient-descent/ (accessed January 4, 2021).

[16] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[17] SGD https://keras.io/api/optimizers/sgd/ (accessed September 4, 2020).

[18] Minna, J. D., Roth, J. A., & Gazdar, A. F. (2002). Focus on lung cancer. *Cancer cell*, *1*(1), 49-52.

[19] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, *68*(6), 394-424.

© Daffodil International University

[20] Sun, W., Zheng, B., & Qian, W. (2016, March). Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical imaging 2016: computer-aided diagnosis* (Vol. 9785, p. 97850Z). International Society for Optics and Photonics.

[21] Suguna, N., & Thanushkodi, K. (2010). An improved k-nearest neighbor classification using genetic algorithm. International Journal of Computer Science Issues, 7(2), 18-21.

© Daffodil International University

APPENDIX A

Dataset Collection and Categorization

The dataset SPIE-AAPM lung CT challenge is collected from the cancer imaging archive. A 12.1 GB DICOM images contains in this dataset. The dataset was downloaded using NBIA data retriever software. The dataset contains 22,489 DICOM images from 70 different patients. DICOM images were opened using a DICOM viewer software and converted to jpg format. A total 81 images categorized into benign and malignant classes for training and testing from there given XLS sheet. The whole process described below with figures

Downloading the dataset using NBIA data retriever

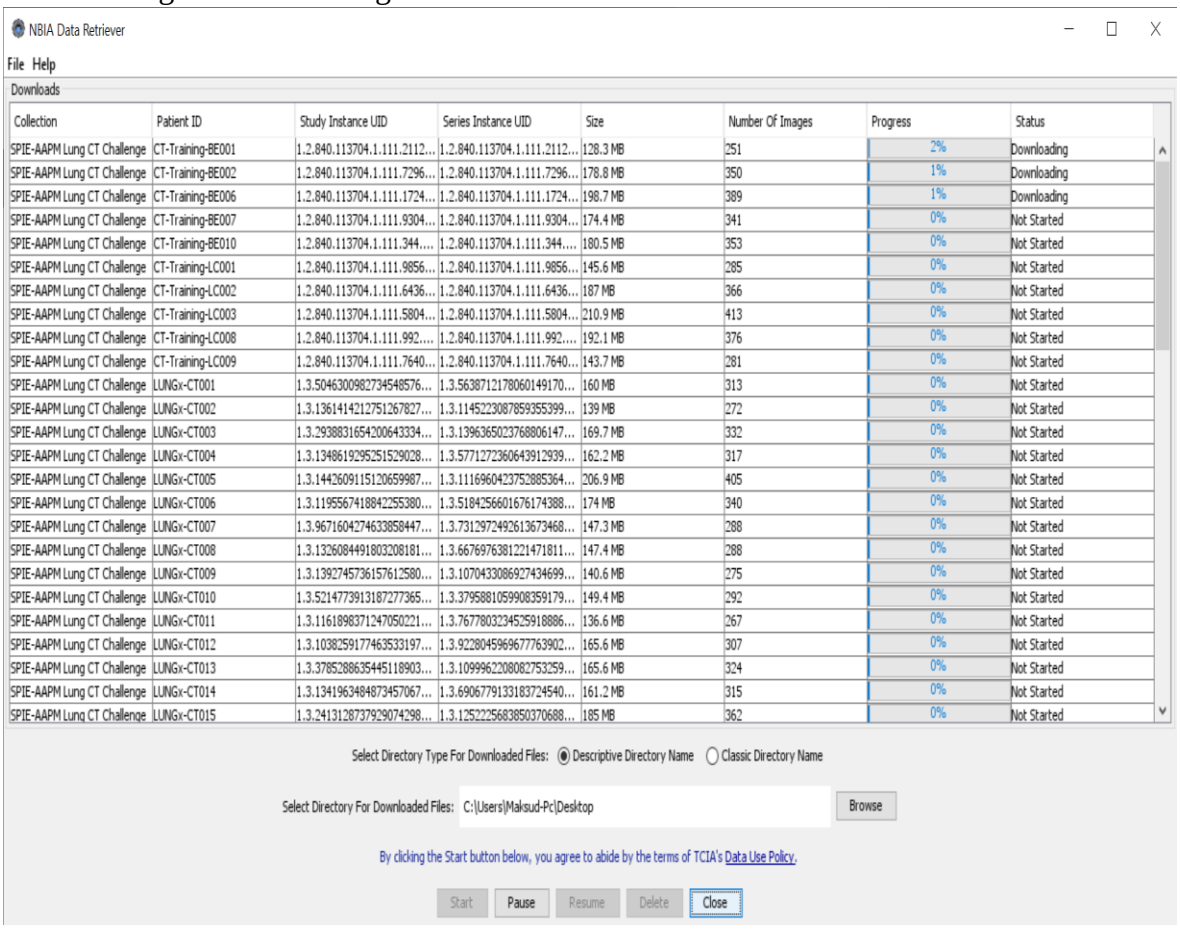


Figure A.1: Downloading dataset

© Daffodil International University

Downloaded DICOM CT scan images of a patient

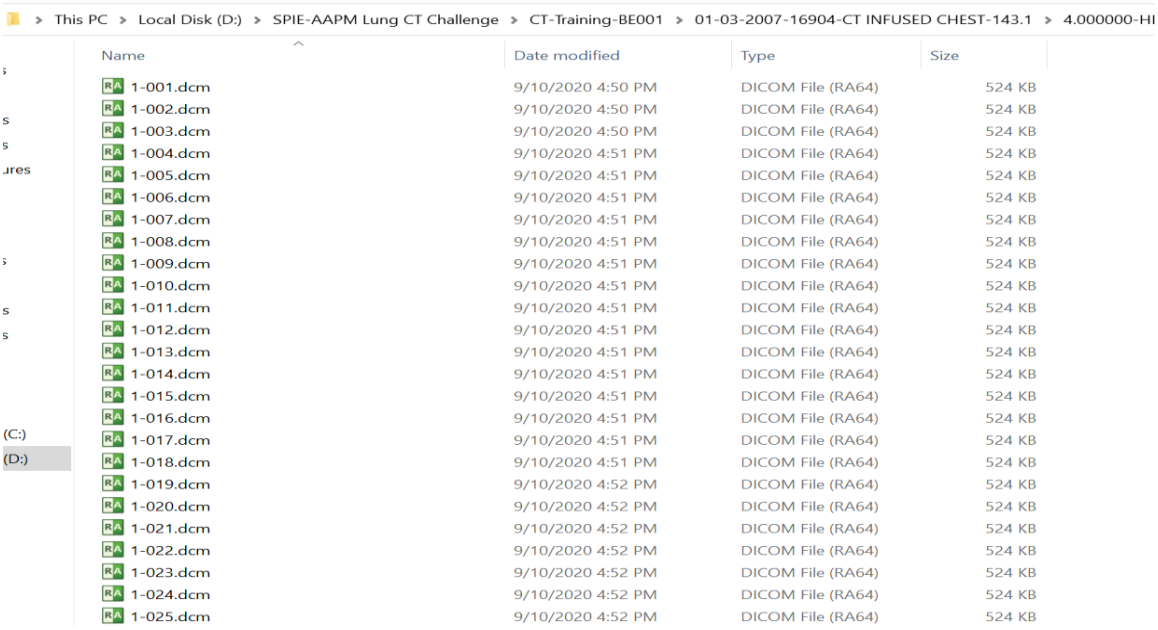


Figure A.2: Downloaded DICOM images

Opening A DICOM image using DICOM viewer software

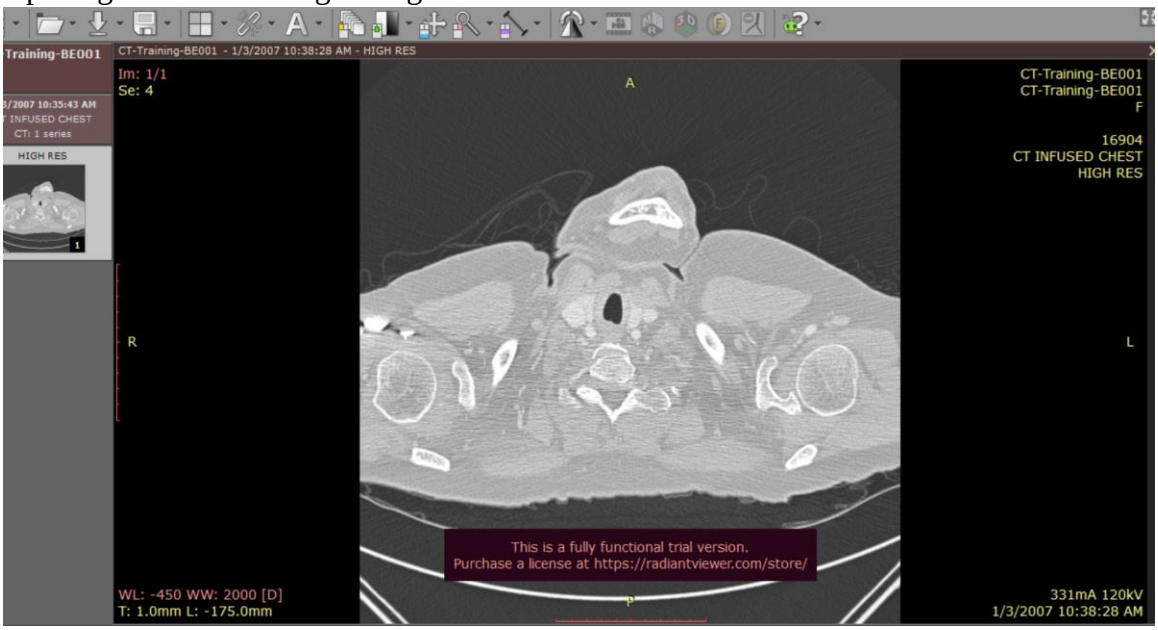


Figure A.3: Opening DICOM image

© Daffodil International University

JPG images were classified and kept into be(benign) and lc(malignant) folder

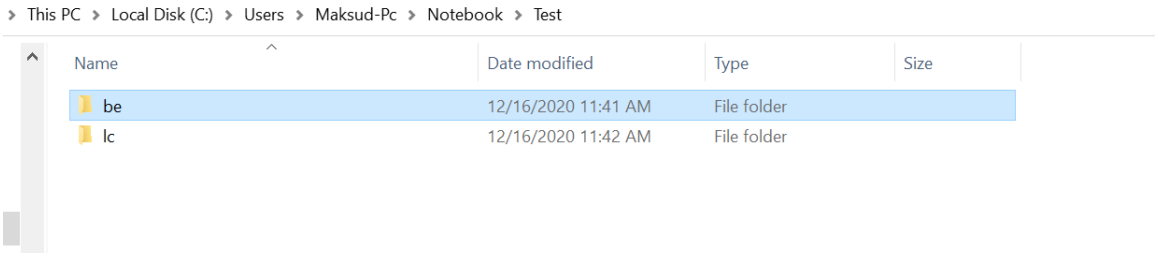


Figure A.4: Classifying the dataset

Benign JPG CT scan images in be folder for training

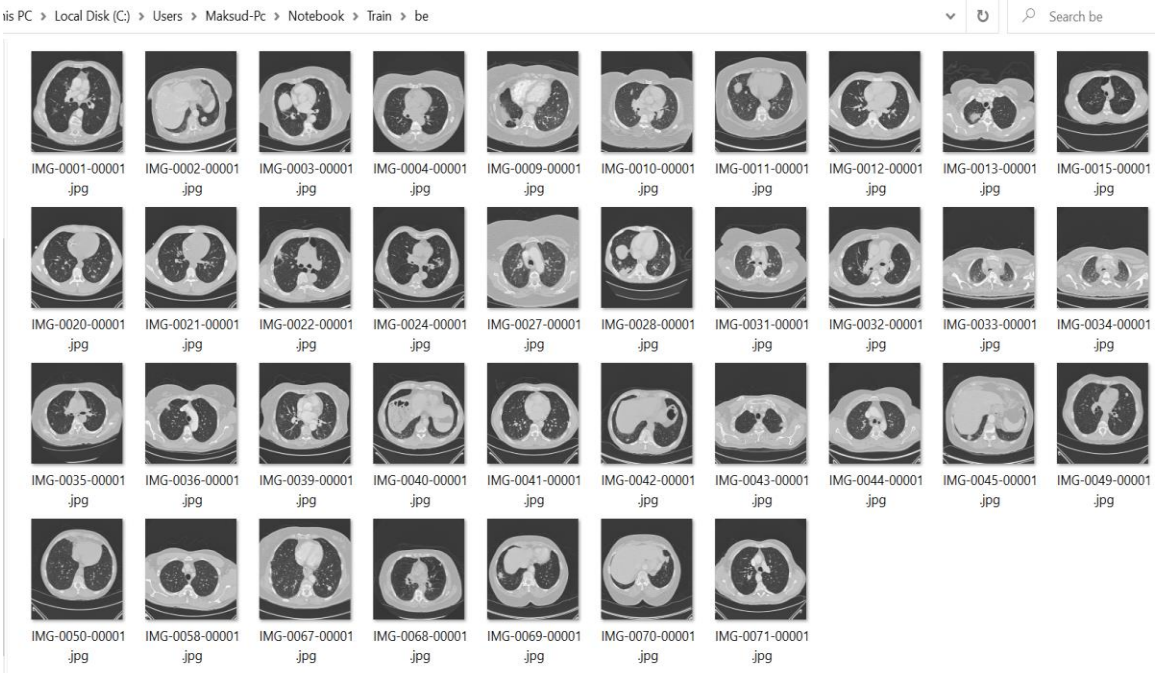


Figure A.5: Benign images

© Daffodil International University

APPENDIX B

Step by Step Improvement

The training dataset used without data augmentation and epoch is set to 30, as a result the training accuracy of the model is 0.84 and loss is 1.285r

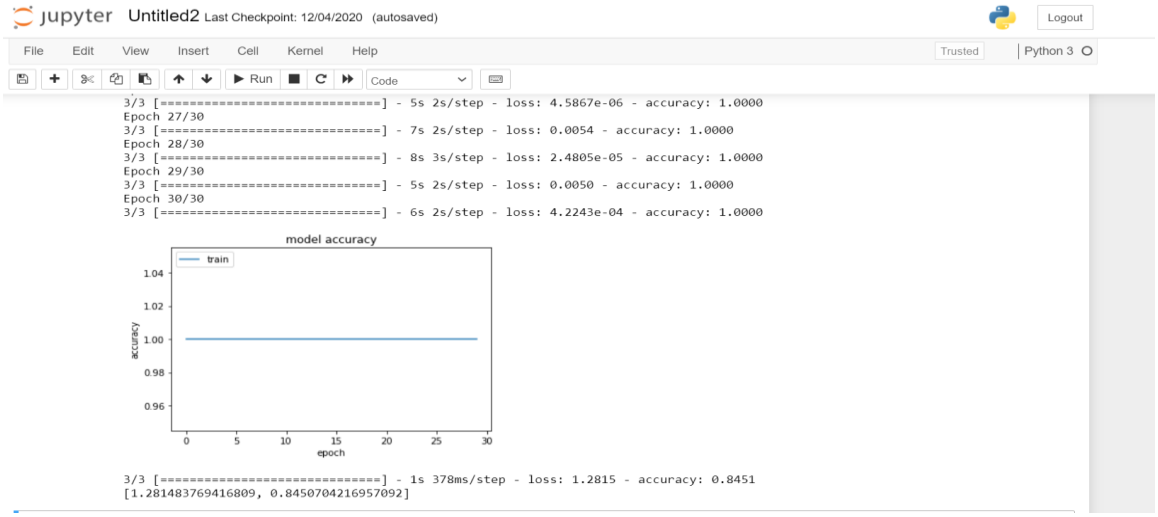


Figure B.1: First step training

After using data augmentation with same epoch, the training accuracy and loss is improved to 0.9718 and 0.0894

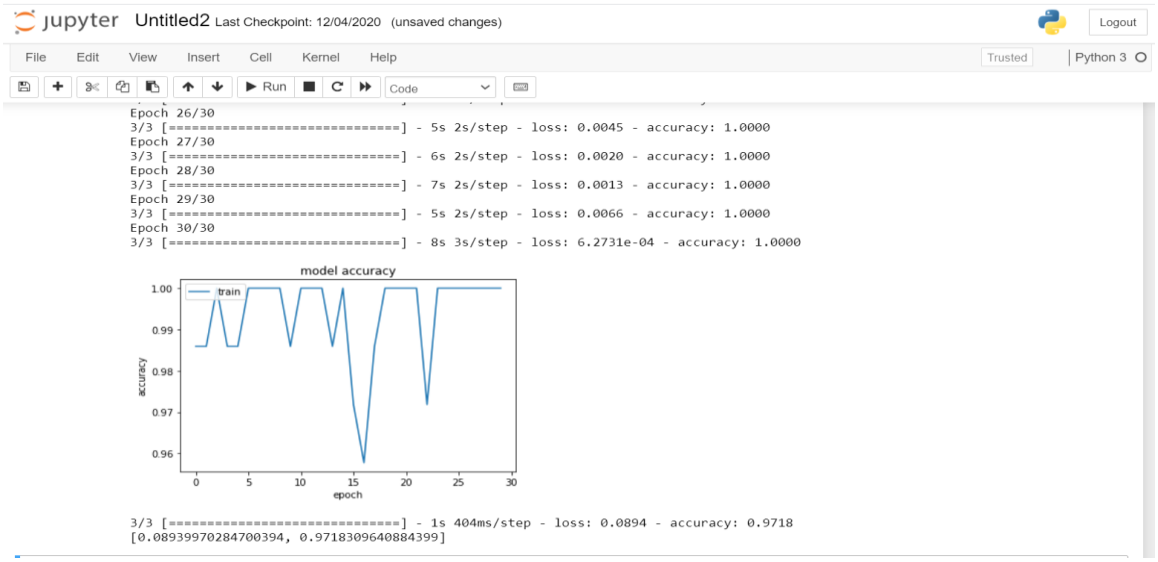


Figure B.2: Second step training

© Daffodil International University

The training accuracy and loss deteriorated to 0.9296 and 0.3305 after the epoch is set to 60

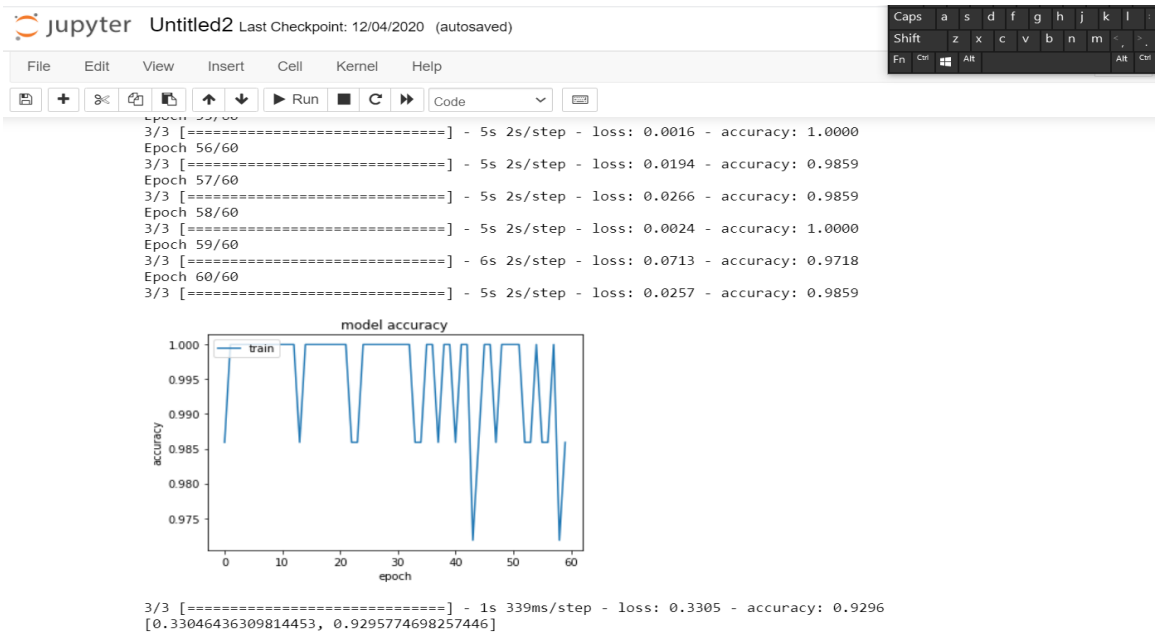


Figure B.3: Third step training

When the epoch is set to 120, the training accuracy and loss is improved to 1.0 and 0.006

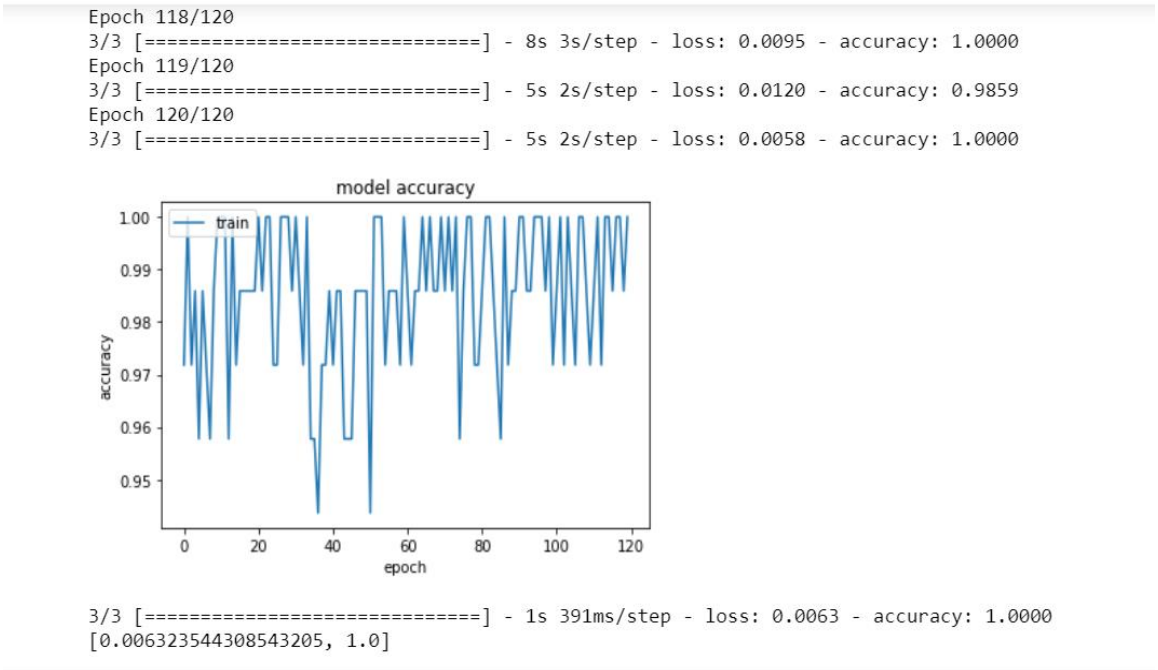


Figure B.4: Fourth step training

© Daffodil International University

Model accuracy for 30 epochs using only alexnet

```
In [11]: score = alex.evaluate(tests, verbose=1)
         print(score)

         1/1 [==============================] - 0s 0s/step - loss: 1.7481 - accuracy: 0.6000
         [1.7480876445770264, 0.6000000238418579]
```

Figure B.5: First step model accuracy

Model accuracy for 30 epochs using image augmentation and alexnet

```
In [15]: score = alex.evaluate(tests, verbose=1)
         print(score)

         1/1 [==============================] - 0s 0s/step - loss: 2.1050 - accuracy: 0.7000
         [2.104989528656006, 0.699999988079071]
```

Figure B.6: Second step model accuracy

Model accuracy improved after using feature extraction and kNN classifier with 120 epochs

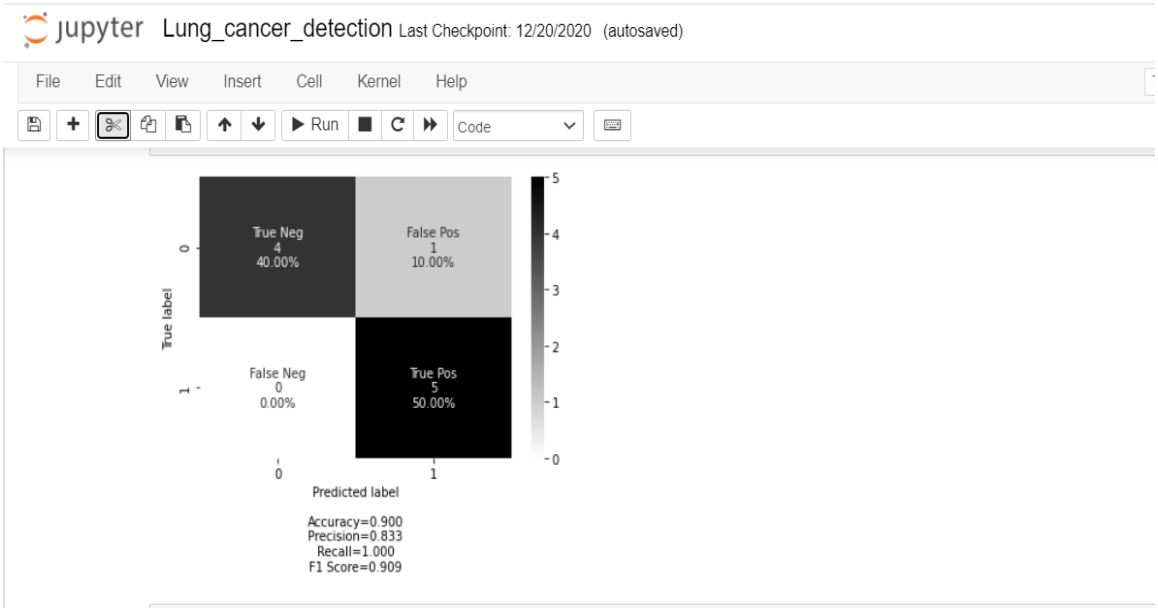


Figure B.7: Final improved model accuracy

© Daffodil International University

APPENDIX C

Sample of Implemented Code

Importing all the modules required

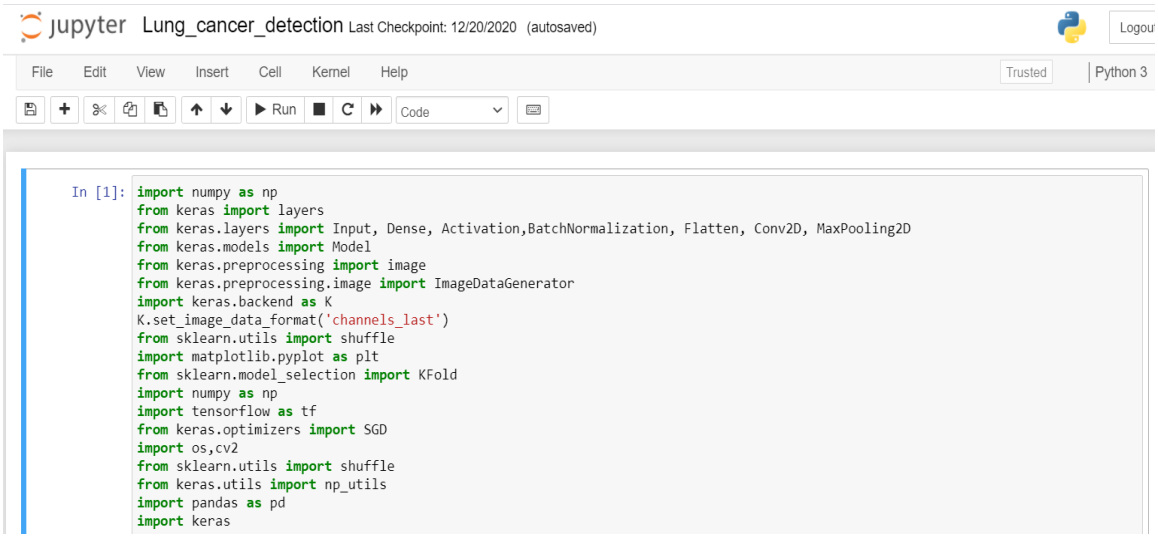


Figure: C.1: Importing module

Defining train and test dataset for binary labeling

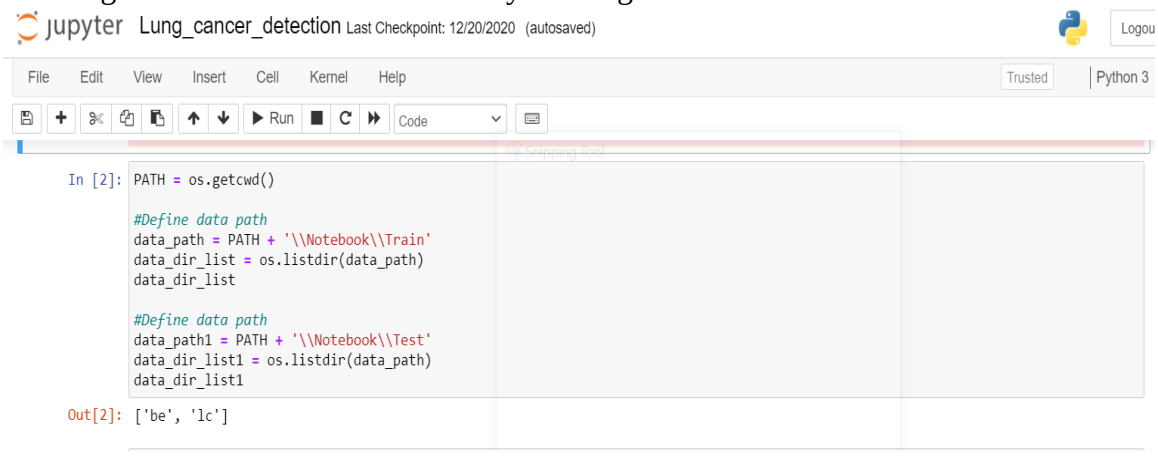


Figure C.2: Defining dataset

© Daffodil International University

Image preprocessing and image augmentation

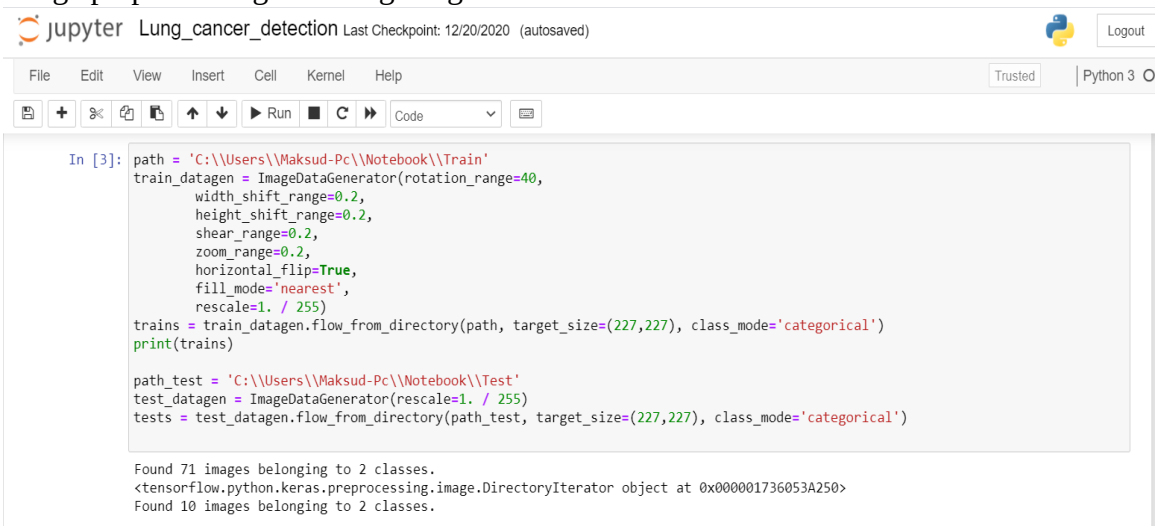


Figure C.3: Image preprocessing and augmentation

Binary labeling the test dataset

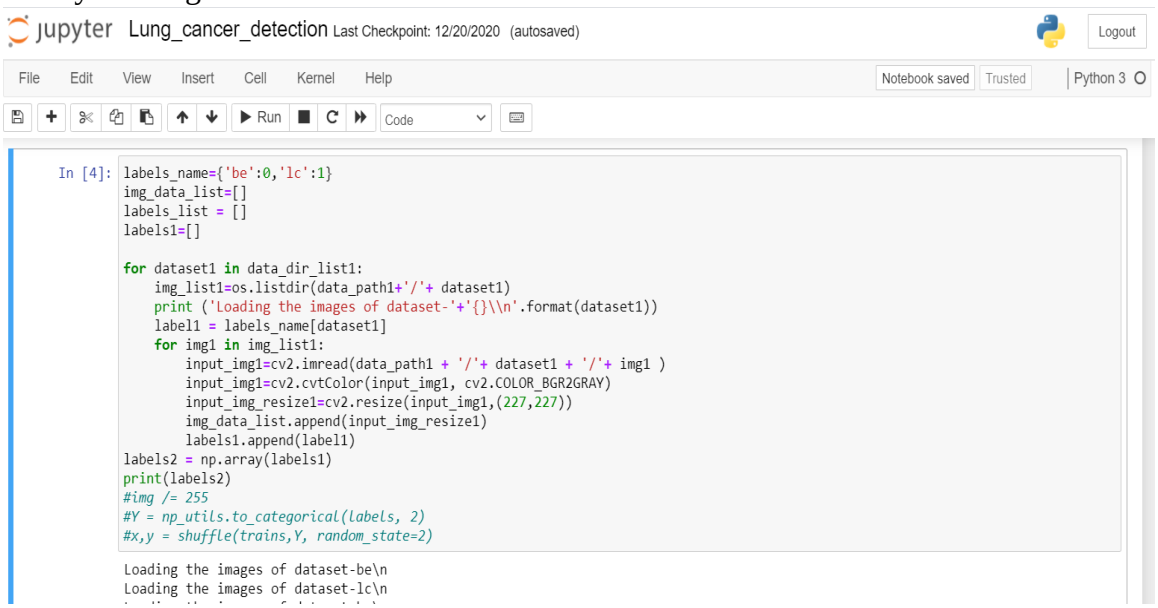


Figure C.4: Labeling

© Daffodil International University

Alexnet model

```
def AlexNet(input_shape):

    X_input = Input(input_shape)

    X = Conv2D(96,(11,11),strides = 4,name="conv0")(X_input)
    X = BatchNormalization(axis = 3 , name = "bn0")(X)
    X = Activation('relu')(X)

    X = MaxPooling2D((3,3),strides = 2,name = 'max0')(X)
    X = BatchNormalization(axis = 3 ,name='bn1')(X)
    X = Activation('relu')(X)

    X = Conv2D(256,(5,5),padding = 'same' , name = 'conv1')(X)
    X = BatchNormalization(axis = 3 ,name='bn2')(X)
    X = Activation('relu')(X)

    X = MaxPooling2D((3,3),strides = 2,name = 'max1')(X)
    X = BatchNormalization(axis = 3 ,name='bn3')(X)
    X = Activation('relu')(X)

    X = Conv2D(384, (3,3) , padding = 'same' , name='conv2')(X)
    X = BatchNormalization(axis = 3, name = 'bn4')(X)
    X = Activation('relu')(X)

    X = Conv2D(384, (3,3) , padding = 'same' , name='conv3')(X)
    X = BatchNormalization(axis = 3, name = 'bn5')(X)
    X = Activation('relu')(X)

    X = Conv2D(256, (3,3) , padding = 'same' , name='conv4')(X)
    X = BatchNormalization(axis = 3, name = 'bn6')(X)
    X = Activation('relu')(X)

    X = MaxPooling2D((3,3),strides = 2,name = 'max2')(X)
    X = BatchNormalization(axis = 3 ,name='bn7')(X)
    X = Activation('relu')(X)

    X = Flatten()(X)

    X = Dense(4096, activation = 'relu', name = "fc0")(X)

    X = Dense(4096, activation = 'relu', name = 'fc1')(X)

    X = Dense(1000, activation='relu',name = 'fc2')(X)

    X = Dense(2, activation='softmax',name = 'softmax')(X)

    model = Model(inputs = X_input, outputs = X, name='AlexNet')
    return model

alex = AlexNet(trains[0][0].shape[1:])

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
alex.compile(optimizer = sgd , loss = 'categorical_crossentropy' , metrics=['accuracy'])
```

Figure C.5: Alexnet model

Training the model with preprocess train dataset

```
checkpoint_filepath = 'D:\\Users\\Checkpoint'
model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
filepath=checkpoint_filepath,
save_weights_only=True,
monitor='loss',
mode='min',
save_best_only=True)

alex.load_weights(checkpoint_filepath)

history=alex.fit(trains, epochs=120, verbose=1, batch_size=32, callbacks=[model_checkpoint_callback])

# alex.load_weights(checkpoint_filepath)

# # Generate generalization metrics
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper left')
plt.show()

scores = alex.evaluate(trains, verbose=1)
print(scores)

# print(f'Score for fold {fold_no}: {alex.metrics_names[0]} of {scores[0]}; {alex.metrics_names[1]} of {scores[1]*100}%')
# acc_per_fold.append(scores[1] * 100)
# loss_per_fold.append(scores[0])

#fold_no = fold_no + 1
```

```
Epoch 1/120
3/3 [==============================] - 37s 12s/step - loss: 3.0863e-04 - accuracy: 1.0000
Epoch 2/120
3/3 [==============================] - 26s 9s/step - loss: 1.6407e-04 - accuracy: 1.0000
Epoch 3/120
3/3 [==============================] - 5s 2s/step - loss: 0.0030 - accuracy: 1.0000
Epoch 4/120
3/3 [==============================] - 17s 6s/step - loss: 1.4002e-04 - accuracy: 1.0000
Epoch 5/120
3/3 [==============================] - 6s 2s/step - loss: 6.3975e-04 - accuracy: 1.0000
Epoch 6/120
3/3 [==============================] - 7s 2s/step - loss: 2.4798e-04 - accuracy: 1.0000
Epoch 7/120
3/3 [==============================] - 7s 2s/step - loss: 7.1349e-04 - accuracy: 1.0000
Epoch 8/120
```

Figure C.6: Training the model

© Daffodil International University

Feature extraction from the last layer of Alexnet

```
In [8]: from sklearn.model_selection import cross_val_score
new_model=Model(inputs=alex.input,outputs=alex.get_layer('softmax').output)
#Let's obtain the Input Representations
train_x=new_model.predict(trains)
test_x=new_model.predict(tests)
print(train_x)
```

```
[[5.7275083e-05 9.9994278e-01]
 [9.9999928e-01 7.5957820e-07]
 [9.9999893e-01 1.1286087e-06]
 [1.0000000e+00 5.8821975e-08]
 [7.0447350e-05 9.9992955e-01]
 [1.0000000e+00 4.1575682e-08]
 [8.2390919e-08 9.9999988e-01]
 [9.9914801e-01 8.5201766e-04]
 [9.9999952e-01 4.6673702e-07]
 [9.9999988e-01 8.2616737e-08]
 [9.9954993e-01 4.5000124e-04]
 [1.1551637e-04 9.9988449e-01]
 [1.5823772e-06 9.9999845e-01]
 [9.9999750e-01 2.5534171e-06]
 [1.9336184e-08 1.0000000e+00]
 [2.3530968e-06 9.9999762e-01]
 [9.9999881e-01 1.2103077e-06]
 [1.0000000e+00 5.4546689e-08]
 [2.4221376e-09 1.0000000e+00]
 [1.1228966e-07 9.9999988e-01]
 [1.5618851e-06 9.9999845e-01]
 [9.9999833e-01 1.6692364e-06]
 [2.3937190e-08 1.0000000e+00]
 [9.9995863e-01 4.1401279e-05]
 [9.9999952e-01 4.2097869e-07]
 [1.0000000e+00 8.9243786e-09]
 [5.1902832e-10 1.0000000e+00]
 [9.9999666e-01 3.3292552e-06]
```

Figure C.7: Feature extraction

Implementing the extracted features to kNN and finding model accuracy for different value of k

```
In [90]: k_range=range(1, 8)
k_scores=[]
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, test_x, labels1, cv=5, scoring='accuracy')
    k_scores.append(scores.mean())
print (k_scores)
```

```
[0.9, 0.8, 0.9, 0.9, 0.9, 0.9, 0.5]
```

```
In [91]: plt.plot(k_range, k_scores)
plt.xlabel('value of k for knn')
plt.ylabel('cross-Validation Accuracy')
```

```
Out[91]: Text(0, 0.5, 'cross-Validation Accuracy')
```

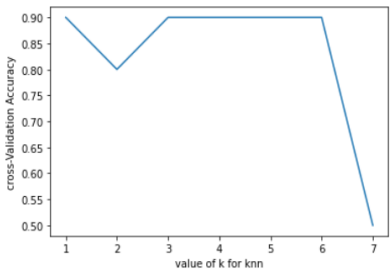


Figure C.8: Model accuracy for different value of k

© Daffodil International University

Confusion matrix for model accuracy and the value of k is set to 4

jupyter Lung_cancer_detection Last Checkpoint: 12/20/2020 (unsaved changes)

File Edit View Insert Cell Kernel Help

```
def make_confusion_matrix(cf,
                          group_names=None,
                          categories='auto',
                          count=True,
                          percent=True,
                          cbar=True,
                          xyticks=True,
                          xyplotlabels=True,
                          sum_stats=True,
                          figsize=None,
                          cmap='Blues',
                          title=None):

    # CODE TO GENERATE TEXT INSIDE EACH SQUARE
    blanks = ['' for i in range(cf.size)]

    if group_names and len(group_names)==cf.size:
        group_labels = ["{}\n".format(value) for value in group_names]
    else:
        group_labels = blanks

    if count:
        group_counts = ["{0:0.0f}\n".format(value) for value in cf.flatten()]
    else:
        group_counts = blanks

    if percent:
        group_percentages = ["{0:.2%}".format(value) for value in cf.flatten()/np.sum(cf)]
    else:
        group_percentages = blanks

    box_labels = [f"{v1}{v2}{v3}".strip() for v1, v2, v3 in zip(group_labels,group_counts,group_percentages)]
    box_labels = np.asarray(box_labels).reshape(cf.shape[0],cf.shape[1])


    # CODE TO GENERATE SUMMARY STATISTICS & TEXT FOR SUMMARY STATS
    if sum_stats:
        #Accuracy is sum of diagonal divided by total observations
        accuracy  = np.trace(cf) / float(np.sum(cf))

        #if it is a binary confusion matrix, show some more stats
        if len(cf)==2:
            #Metrics for Binary Confusion Matrices
            precision = cf[1,1] / sum(cf[:,1])
            recall    = cf[1,1] / sum(cf[1,:])
            f1_score  = 2*precision*recall / (precision + recall)
            stats_text = "\n\nAccuracy={:0.3f}\nPrecision={:0.3f}\nRecall={:0.3f}\nF1 Score={:0.3f}".format(
                accuracy,precision,recall,f1_score)
        else:
            stats_text = "\n\nAccuracy={:0.3f}".format(accuracy)
    else:
        stats_text = ""


    # SET FIGURE PARAMETERS ACCORDING TO OTHER ARGUMENTS
    if figsize==None:
        #Get default figure size if not set
        figsize = plt.rcParams.get('figure.figsize')

    if xyticks==False:
        #Do not show categories if xyticks is False
        categories=False


    # MAKE THE HEATMAP VISUALIZATION
    plt.figure(figsize=figsize)
    sns.heatmap(cf,annot=box_labels,fmt="",cmap=cmap,cbar=cbar,xticklabels=categories,yticklabels=categories)

    if xyplotlabels:
        plt.ylabel('True label')
        plt.xlabel('Predicted label' + stats_text)
    else:
        plt.xlabel(stats_text)

    if title:
        plt.title(title)

labels = ['True Neg','False Pos','False Neg','True Pos']
make_confusion_matrix(cm,
                      group_names=labels,
                      cmap='binary')
```

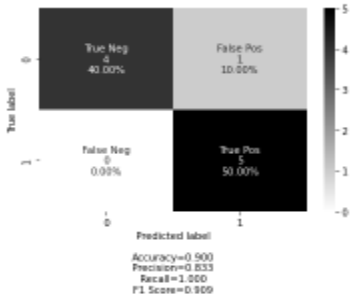
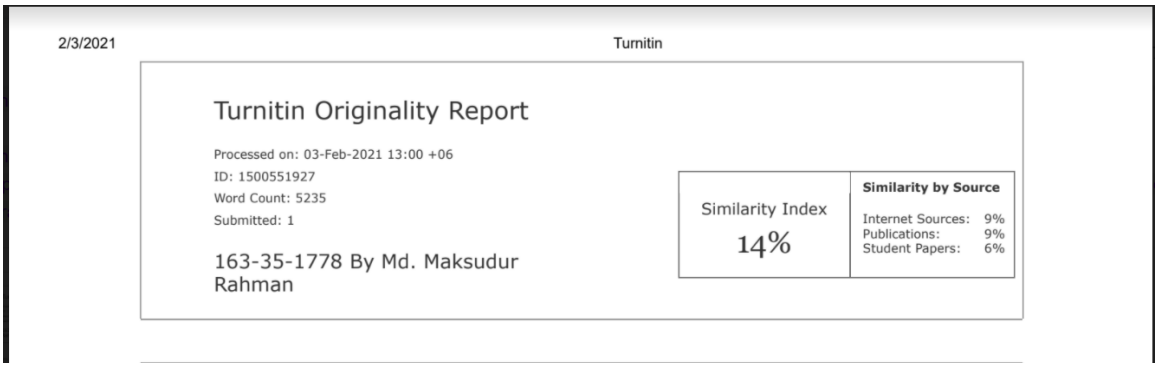


Figure C.9: Confusion matrix

© Daffodil International University

APPENDIX D



2/3/2021 Turnitin

Turnitin Originality Report

Processed on: 03-Feb-2021 13:00 +06
ID: 1500551927
Word Count: 5235
Submitted: 1

163-35-1778 By Md. Maksudur Rahman

Similarity Index	Similarity by Source	
14%	Internet Sources:	9%
	Publications:	9%
	Student Papers:	6%

© Daffodil International University