October University for Modern Sciences and Art University

Faculty of Computer Science

Graduation Project Documentation

Early Diagnosis of Lung Cancer using machine learning

Semester (65)

Spring 2018

Submitted by:
Name: Abdullah Tarek Farag
ID: 153225

Supervised by:

Dr. Ahmed Farouk

Graduation Project Spring 2018

Prepared by

Abdullah Tarek Farag Ibrahim Farahat

Supervisor

Dr. Ahmed Farouk

Sponsored by

Domino Data lab

Abstract

Lung cancer is one of the deadliest types of cancer because it can spread to other tissue in the body damaging them as well. In 2017 there were 225,000 people who got diagnosed with lung cancer in the U.S only. It made an accumulative cost of 12 billion in health care costs. Early detection is one of the most effective methods to combat cancer. It increases the chance of survival drastically for the patient. A Doctor's job is to classify whether this scan is cancerous or not by analyzing nodules. But human beings can analyze nodules that are bigger than 7mm in diameter and doctors can sometimes make a patient wait to see whether this nodule will grow, or it won't because if it doesn't grow it will be a harmless nodule. And this gives a chance for a nodule to grow more undetected by the human eye. This project is aimed to detect nodules that are as small as 3mm to detect cancer as early as possible.

Machine learning and Deep Learning are now the state of the art techniques at almost all calcification problems. It can learn to classify a scan as accurately as a doctor. And it takes only hours of training not years of education and experience. The project first preprocess the CT scans and divides them into crops of 64X64X64. Then the crops enter a pipeline consisting of two 3d CNN. The first one is a binary classification that classifies whether this crop has a nodule or not. Then the crops that has a nodule enter another 3d CNN to classify which stage of malignancy is it. It predicts a number between 0 and 4, where zero is not malignant at all and four is the highest in malignancy. A number of different models were used in both classifications. The nodule classification got the best accuracy by the Googlenet model. And the malignancy classification got the best accuracy by the Lenet model.

Contents

Table of Figures

Chapter 1: Introduction

1.1 What is Lung Cancer

Lung Cancer is uncontrollable development of abnormal cells in either on or the two lungs. It prevents the Lung from functioning correctly and block air passage ways thus preventing the lung from nourishing the body with air and O2 fully. Those abnormal cells will keep on replicating till the form a tumor. Those tumors can be **benign**, remain in one place, or **malignant**, spread throughout the body through bloodstreams or something else, those are more harmful **(What Is Lung Cancer?, n.d.)**.

1.2 The Dangers of Lung cancer

Lung cancer is deadliest cancer there is, with 225,000 new of lung cancer were diagnosed in the U.S. and it is expected to hit two every five people in their life times in the U.S. Lung cancer causes the most death among all cancers. It had 12 billion dollars accumulative cost in the health care industry. Early detection is a very important to give patients a better chance of survival and minimizing the damage as low as possible, as it opens a range of treatment options that were not available when cancer is detected at later at more advanced stages. CT scans have low dose and high dose, low dose CT scans made a huge impact in the medical field and it saved lives up to 60% of the patients. (false positives from the initial screening) **(Team, 2016).**

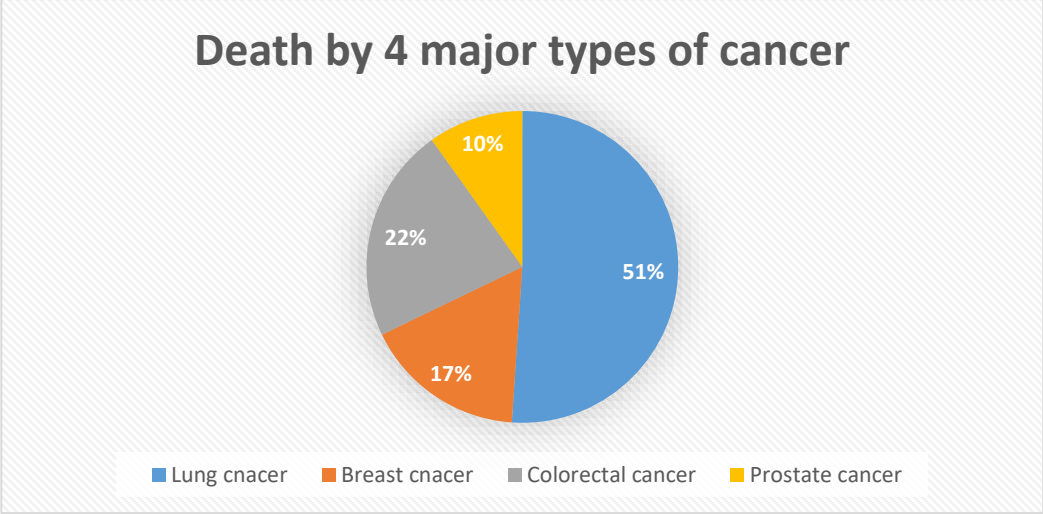


Figure 1 Death by cancer chart from the first reference

1.3 Benefits of early detection

Finding and treating cancer at an early stage saves lives. Cancer that is analyzed at am early stage, before it had the opportunity to get too huge or spread will probably be dealt with effectively and the patient will survive. If cancer spread, treatment turns out to be more troublesome, and for the most part a man's odds of surviving are much lower. Around 70% of

lung tumor patients will get by for no less than a year if analyzed at the soonest arrange contrasted with around 14% for individuals determined to have the most exceptional phase of ailment. Early determination can build odds of survival. However, enhancing survival rates isn't simply down to prior conclusion – guaranteeing patients get the best and proper treatment for them is additionally a critical piece of the jigsaw. The earlier it gets detected the better, because now actions are taken before damage becomes severe **(Jay W. Marks, n.d.).**

1.4 The limitations of doctor diagnosis

Now a normal procedure of how someone is diagnosed with lung cancer. The patient is first given a low dose CT-scan. CT- stands for computed tomography and it is a series of x-rays taken at different angels and because multiple angels are used a 3d volume of the things inside the body is constructed. If a doctor sees only blood vessels and non-growing clumps of cells this is fine and the patient does not have cancer. But if the mass of cells is big or is growing that means the patient has cancer. Here lies two problems the first is that there is no way for a doctor to determine that this clump of cells will grow or not the doctor has to make the patient wait at least 6 months and make another CT-scan, this gives an opportunity for cancer to become stronger and wastes time the patient can be treated in .The second problem is drawing the line between a small clump of cells and a big one. This is very hard task to be done and even experts get it wrong. The doctors have to make biopsies, an invasive way of taking a small part of this nodule, to determine whether this is cancerous or not for sure. And about 30% of those biopsies are wasted **(Team, 2016).**

1.5 Motivation

The wide spread and the damaging effects of lung cancer are great and actions should be taken to lessen its damaging effects. So as described early in the paper the benefits of early detection can be lifesaving. This is classifying whether the patient has early stages of lung cancer or not. And since Deep learning is now the state of the art classifying images with CNN. That was main approach for solving this problem using machine and deep learning to outperform doctors in early diagnosis of lung cancer. This project is aimed to how machine learning and computer science can be used to make early detection easy and as accurate as possible. Just by uploading the CT-scan and pressing a button, it doesn't even need a doctor to understand. As machine learning is beating accuracies in all sorts of fields from business to image recognition and classification.

1.6 Objective

This project is a classification problem to detect Lung cancer with and without early detection and stating the stage of the lung cancer a number from 0 to 4 is given where zero is no malignancy and 4 being the most malignant. Deep learning and machine learning is used to tackle this problem.

1.7 Aim

My Aim is to read CT scans and classify nodules with diameter as small as 3mm. 3mm diameter is very small and is hardly detected by radiologists. Radiologist can't classify nodules that are that small. This project aims to use methods from deep learning and computer vision particularly 2D CNN and 3D CNN to build an accurate classifier that will be able to diagnose lung cancer at its very early stages to gain the advantages of early detection as discussed above.

1.8 The complexity of the problem

This task may seem straightforward as just classifying an image, but actually it is way more complex than that. It is a needle in haystack problem, the classifier is looking for nodules as small as 3 mm in diameter in a large CT scan usually 200X300X400.The figure bellow shows how small a 5mm nodule can be in just one slice of CT scan. A CT scan can contain up to 300 slices. This makes it very hard for classifier to just take the scan as is and classify it **(S.Hawkins).**

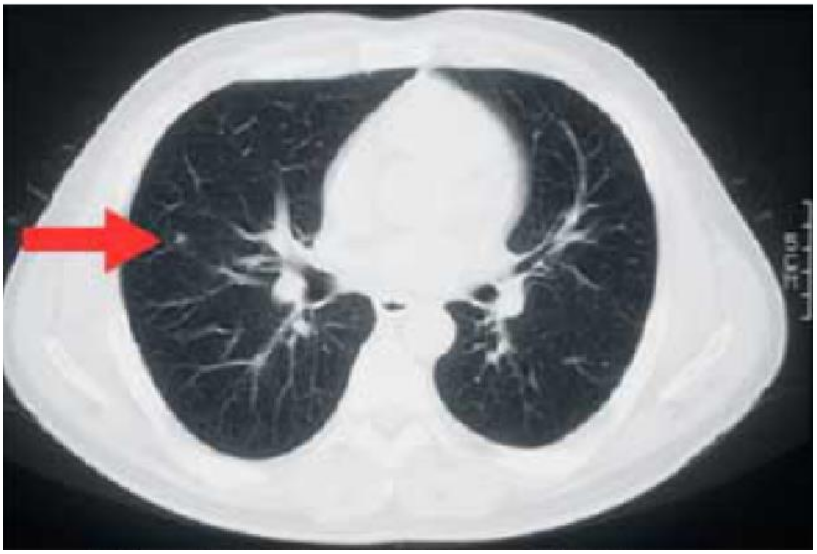


Figure 2 A nodule in one slice of CT scan from from reference 14

1.9 Background

As you have scene above the thing that classifies this large CT-scan is very small relative to the size of the whole CT-scan. Now this makes classification very hard for both human beings and machine learning algorithms. This project first tried to classify a full CT-scan as a whole whether it is cancerous or not, but failed getting less than 60% accuracy. Then the project tried to segment out the nodule using a segmentation but also suffered from the same problem which is the search space is too big relative to thing we want to segment. Next the project divided the CT-scans into 64X64X64 crops and classify each crop at a time decreasing the search space. This approach worked and got good accuracies and will be discussed below.

1- Reading annotations: csv file and xml file.
2- Image processing: change world coordinates to voxel coordinates, normalize planes.
3- Generating crops: generating patches of 64X64X64 from the whole scan.
4- Nodule candidate detection: Find the crops that has nodules in the scan.
5- Malignancy prediction for each nodule candidate: make a network that takes the nodule and predict a number from 0 to 4; The higher the number the higher the level of malignancy.
    - 0 → this nodule is not malignant
    - 1 → this nodule is malignant level 1
    - 2 → this nodule is malignant level 2
    - 3 → this nodule is malignant level 3
    - 4 → this nodule is malignant level 4

Chapter 2 Background and Previous work

2.1 Types of Diagnostic tests

2.1.1 Blood Tests

A full blood check is for the most part taken toward the beginning of any examination concerning conceivable ailment, including lung tumor. Changes in the quantity of red or potentially white platelets enable specialists to comprehend if the body is responding/reacting to a sickness. Different diverse normally happening substances, for example the antibodies that are in the region, and the bodies possess chemicals can vary from the ordinary range when there is a disease **(Mayo clinic, n.d.).**

2.1.2 Imaging Tests

2.1.2.1 Chest X-ray

Chest X-ray can distinguish tumors as little as 1 cm wide. Be that as it may, now and then a tumor might be holed up behind another structure e.g. a rib. Additionally imaging tests are by and large required to discount lung growth **(Mayo clinic, n.d.)**

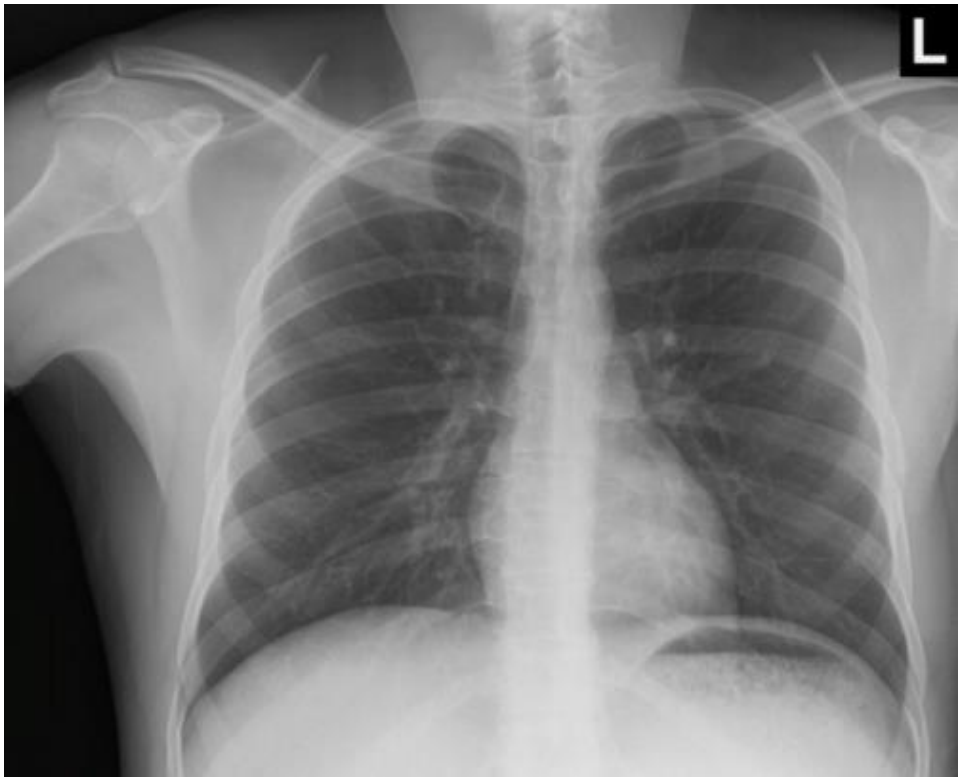


Figure 3X-ray of lung from reference 4

2.1.2.2 CT-Scans

A CT (automatic tomography) scan makes use of x-ray beams to take many pictures of the inner of your body and uses a pc to collect them into one exact, cross-sectional image. it is able to come across smaller tumors than those found by using chest x-rays, and presents precise facts approximately the tumor, the lymph nodes inside the chest and other organs ***(Lawrence M. Davis, n.d.)***.

CT scans are commonly done at a clinic or a radiology clinic. you'll be requested to fast (no longer devour or drink) for numerous hours earlier than the experiment to make the experiment pictures clearer and less complicated to examine. before the experiment, you may be given an injection of dye into a vein to your arm. This dye is called the contrast and it makes the pics clearer. The dye may additionally make you sense hot all over, and depart a bitter flavor on your mouth, and you can feel a sudden urge to bypass urine ***(Lawrence M. Davis, n.d.)***.

The CT scanner is a large, doughnut-fashioned gadget. you will lie flat on a table that moves inside and outside of the scanner. The test itself takes 10–20 mins, however you'll also want to prepare and then watch for the test. whilst a CT test may be noisy, it's far painless The dye utilized in a CT scan typically consists of iodine. when you have had an hypersensitivity to iodine or dyes at some point of a previous test, allow the character appearing the experiment realize in advance. You have to also allow them to know if you are diabetic, have kidney disorder or are pregnant ***(Lawrence M. Davis, n.d.)***.

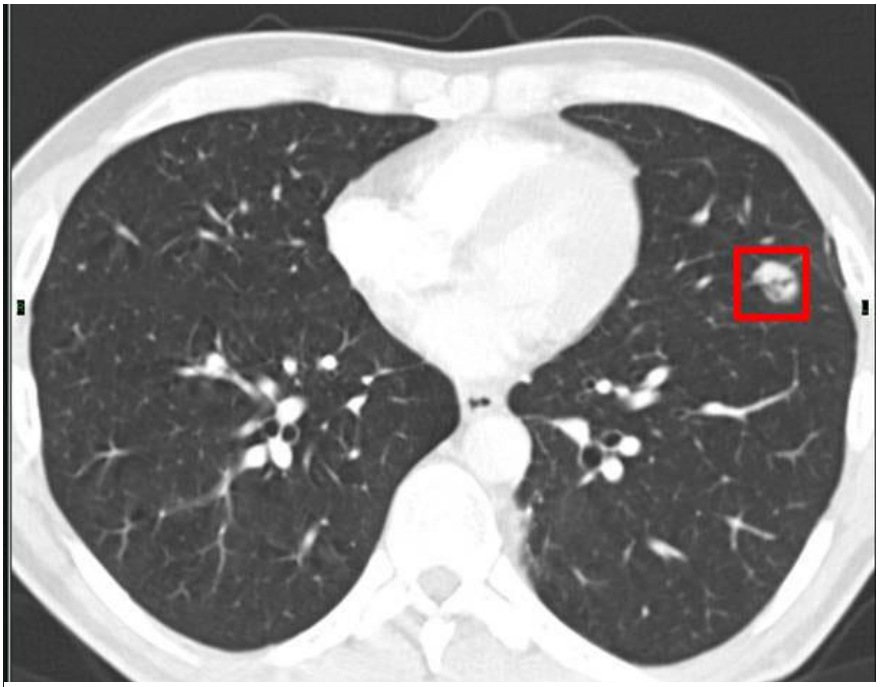


Figure 4 One slice of CT scan with nodule from reference 15

2.1.3 Pet scan

A PET SCAN filter out is a particular imaging test. An infusion of radioactive glucose is about into a vein to then circle during the body. The radioactive glucose

will frame in concentrated quantities round tumor destinations, permitting the sweep to distinguish these tumor locales on imaging. This check is precious in figuring out whether or not the tumor has spread to different parts of the frame **(Mayo clinic, n.d.)**.

A PET scan of someone with metastatic lung cancer.

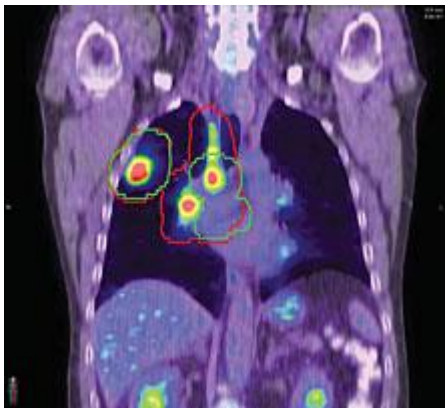


Figure 5Pet scan of lung from reference 14

2.1.4 MRI scan

A MRI SCAN (Magnetic Resonance Imaging) is utilized to look inside the structure and capacity of the lung. A MRI gives considerably more prominent complexity between the distinctive delicate cells of the body than a CT does. This is particularly helpful in growth scans **(Mayo clinic, n.d.).**

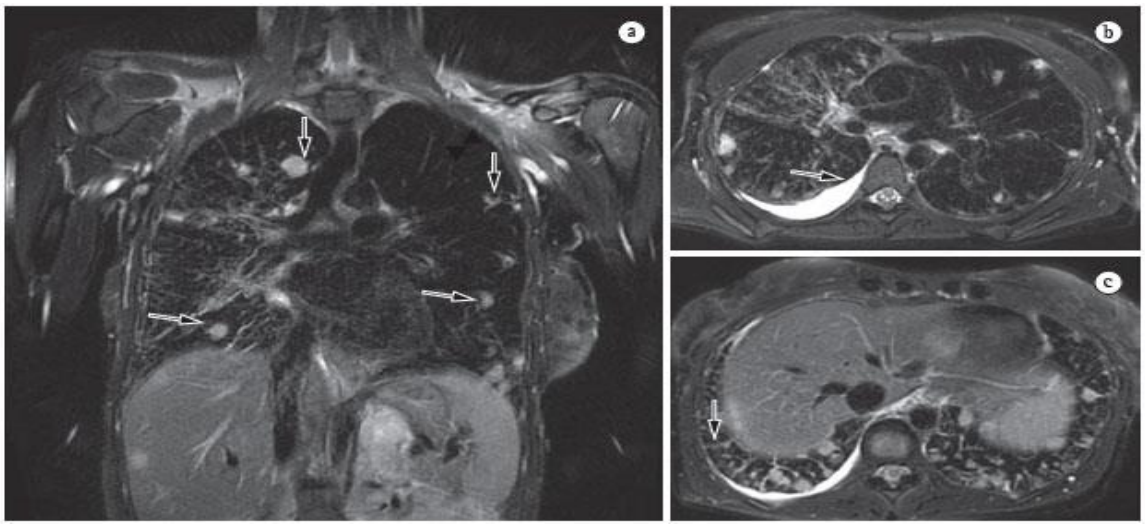


Figure 6MRI scan of lung from reference 15

2.1.5 Body Tissue tests

Another type of testing utilized as a part of seeing if the patient has a lung tumor is body tissue test. discharges or potentially genuine tissue are taken from the lung part of the

patient and tried in a research center, to identify assuming kind of malignancy cells are available e.g. little cell or non-little cell lung growth cells. Similarly as with imaging tests, a few (yet not really all) of these tests will be utilized while recognizing lung malignancy. The radiologist will examine it and will choose the test that is best for the patient **(Mayo clinic, n.d.)**

2.1.5.1 Sputum cytology

Sputum (phlegm) definition is the fluid ingredient that is all over the lung tissue. Changes in sputum amount, shading and the wideness are ordinarily found in the tumor. This test inspects a sputum test under a magnifying lens and search for strange or tumor cells and is helpful in identifying changes that are occurring in the lungs. Sputum that are done in the early morning are the best, as this is the best time to get a clean example from liquid which was made during the night **(Mayo clinic, n.d.)**.

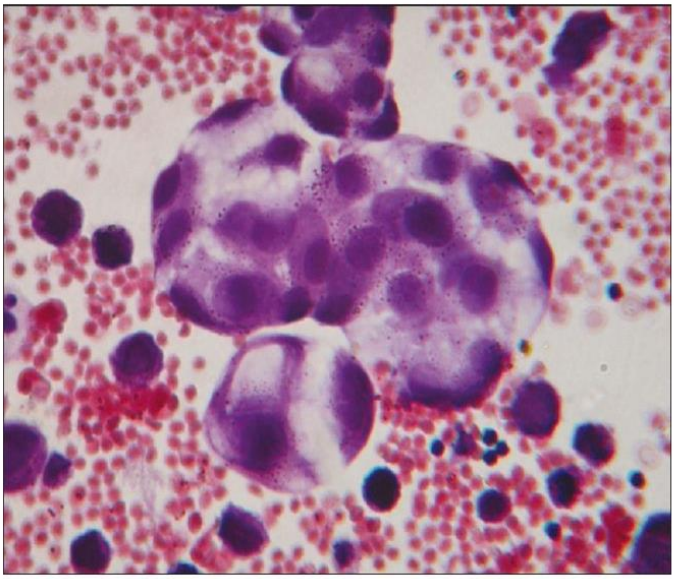


Figure 7Sputum cytology from reference 4

2.1.5.2 Bronchial biopsy

In the event that a tumor is distinguished in the mouth bronchoscopy, (little tissue test) might be gathered for examination. The Tissue is put under a magnifying instrument, to identify changes in singular cells **(Mayo clinic, n.d.)**.

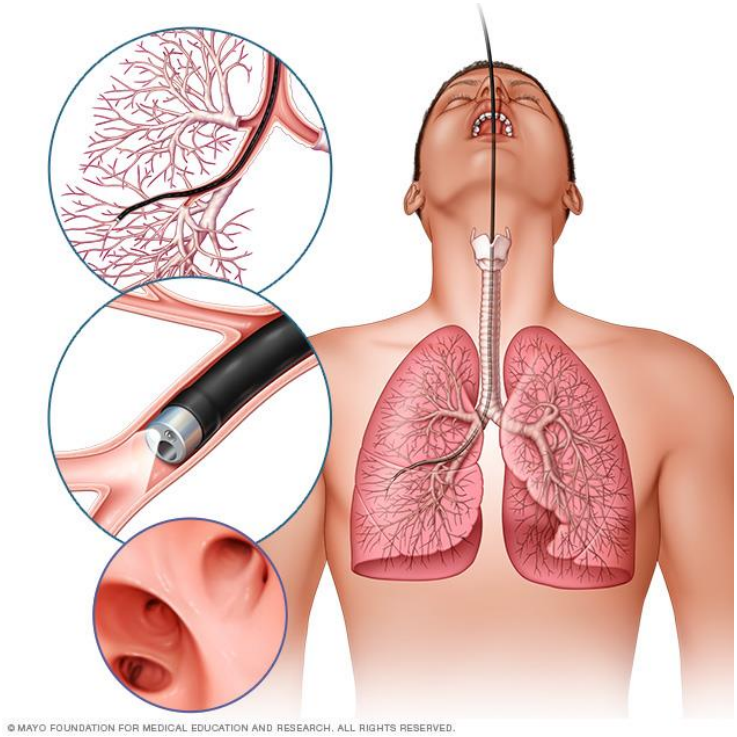


Figure 8Bronchial biopsy from reference 15

2.1.5.3 Bronchoscopy

A bronchoscopy allows your medical doctor to appearance at once into your airways (bronchi), and if required, take biopsy samples of lung tissue. This system is executed the use of a bendy tube called a bronchoscope, that is inserted via your nostril or mouth and down your windpipe. The bronchoscope may additionally feel uncomfortable, but it should now not be painful. you will be given either a mild sedation or a standard anesthetic and the returned of your throat is numbed with a local anesthetic **(Mayo clinic, n.d.)**.

2.2 The current process of diagnosing lung cancer

If you are suspected of having lung cancer, you are first given a low dose CT scan, A CT scan is bunch of x-rays taken at different angels of your lung and the result will be a 3d representation of what your lung looks like, Then they search for suspicious Tumors in the lung. If they find suspicious tumors then they do biopsy, is a surgical procedure where a needle is inserted inside the human body and cells and tissue is extracted and then analyzed to see if the tissue is cancerous or not, now this is very invasive, very uncomfortable and very costly for everyone **(Mayo clinic, n.d.).**

2.2.1 The problems of that approach

Radiologist classify a lung if it is cancerous or not by examining the CT scan. When they only see blood vesicles in the lung then this is fine, but if they see a large mass or a growing mass those are likely metastatic tumors. But this is way harder than it sounds. Radiologist get wrong reading about 30% of the time. So biopsy are wasted 30% wasting time and money for everyone and making the patient more uncomfortable. This is not as easy as it seems it is way harder than it seems. For example drawing the line if this is a small nodule or a blood vesicles. If the mass is large then it is mostly a tumor and biopsy is made. If it is small radiologist have to make the patient wait a couple of months and make another CT scan to see if the suspected area grows or stays the same. If it grows then it is a growing tumor and needs to be dealt with and if it stays the same it is not a nodule. Now this is makes the patient wait and making the cancer grows which is will deteriorate the patient's health. And this is also a waste of money, and makes the doctors operating on the procedure job much harder.

2.3 Prediction with column features:

This approach gives you dataset consisting features that have been extracted from an image and put those features onto columns. The data set is a table containing many rows. Now we can filter features or make new ones based on the already given data and put the preprocessed table into a neural network to train. The neural networks gets trained on the dataset and hopefully get good accuracies. And it doesn't even have to be a neural network, applying a simple KNN classifier got 77% accuracy in the UCI lung cancer data set. Working with already extracted features as you can see can be very simple and get good results. But it takes a lot of time to extract the features and put them into tables and it takes people to manually extract the features and put them into rows. It can be prone error also as human faults at data entry is imminent. CNNs in image classification automatically extracts features from the image and it can extract features that a human being can't detect or see easily. And if the CT scans are used the time, money and workforce needed to make the table will greatly reduce. Also using the scan as a whole makes this a usable applications that patients and doctors can use to diagnose lung cancer as early as possible **(Lung Cancer Data Set , n.d.).**

2.4 Deep learning and medicine

There many application in Deep learning and medicine in all fields from cancer to to finding new cures for the common cold to automatic diagnosis of disease. , Startup Enlitic uses significant making sense of the least difficult approach to take a gander at radiographs and CT and tomography channels. Boss Igor Barani, among the previous a teacher of radiation solution at the University of American state in city, says Enlitic's estimations beated four radiologists in particular and gathering viscus handles as kind or debilitating. (The work has not been peer investigated, and conjointly the development has not but rather learned specialist underwriting.)

Merck is endeavoring to use significant making sense of the least difficult approach to stimulate steady disclosure, very kind of a city startup raised as Atomwise. Neural frameworks investigate 3D pictures—a Brobdingnagian differ of molecules that may fill in as pharmaceutical hopefuls—and predict their sensibility for preventive the a piece of an operator. Such associations unit of estimation using neural nets to attempt to zest up what individuals starting at directly do; others attempt and attempt and to things individuals can't act any methods. lead celestial host Otte, 27, World Health Organization incorporates a pH scale.D. in machine science, started Freenome, that intends to examine development from blood tests. it's at compound parts among the dissemination framework that unit of estimation disgorged by cells as they chomp the soil. Using significant learning, he asks for that PCs find associations between's though not cell compound and maybe a couple of malignancies. "We're seeing novel denotes that haven't been imagined by threat researchers yet," says Otte **(Charles D. Fenimore, 2011).**

At the point once Andreessen performer was contemplating partner enthusiasm for Freenome, AH's Pande sent Otte five outwardly impeded illustrations—two standard and three unsafe. Otte got the greater part of the five right, says Pande, whose firm contributed. While a pro could even observe an immense change of pictures for the duration of his life, a PC is additionally shown millions. "It's not crazy to verify that this picture issue could be disentangled higher by PCs," Pande says, "in light-weight of the specific unquestionable truth that they will drive through a magnificent arrangement more data than an identity's could ever do." The potential central focuses don't seem, by all accounts, to be just further critical exactness and speedier examination, but instead assemble activity of organizations. because of the development at last grounds up typical, among the long run every patient will benefit **(Charles D. Fenimore, 2011)**.

Chapter 3 Requirement analysis

3.1 Requirements for training Dataset

- Cloud :

  The Dataset is about 65 GB. And one sample can reach up to 300 mb. This makes very hard to train this huge dataset on my Intel core i7 laptop. I couldn't even train this type of data when the batch size is one. So this project needed cloud computing to deal with this dataset. In the next part we will talk about the Dataset in details. This project was sponsored by Domino Datalab and they gave this project all the cloud computing It needed and with as much storage as the project wants. Now this was very crucial in loading and training this huge dataset

  - Cloud requirements
    - 80 GB storage
    - 32 GB RAM
    - 4 GPU with cuda
- Testing and GUI on my computer

◦ **Hardware**

- NVIDIA GPU

◦ **Software**

- Anaconda
- Python
- keras
- Tensorflow
- Python libraries (numpy, pandas , opencv2 ,tkinter)

3.2 Dataset

My Primary dataset is the LUNA lung cancer dataset. It has above 65 GB of annotated 3d CT scans of lung. Each CT scan in in .mhd format. One .mhd file is one patient with the whole 3d scan. The size of the scan is 512x512x a variable depth that varies from 150 to 300 and I had to resize the data to make it standard and fit into machine learning algorithms. This Data is very spacious, one patient can reach up to 300mb and. The dataset when read into python consists of multiple 2d slices which makes the 3d object. The annotations in the LUNA dataset gives me x, y and z world positions of and class ether zero or one. Zero is no nodule and 1 there is a nodule. Then I knew that the LUNA dataset is subset from a bigger dataset called LIDCR it had annotations for the malignancy of the nodules in xml format. The Z is in world coordinates and x and y are in voxel positions **(Lung nodule analysis 2016, n.d.) (Charles D. Fenimore, 2011).**

3.3 Functional requirements

- User can upload his CT scan.

- User can view the scan slice by slice
- User can get the diagnosis and if he has lung cancer at what stage

3.4 Nonfunctional requirements

- System take the lung and make image preprocessing
- Run the preprocessed images in the nodule classifier
- Run the nodules in a malignancy classifier

3.5 Use case Diagram

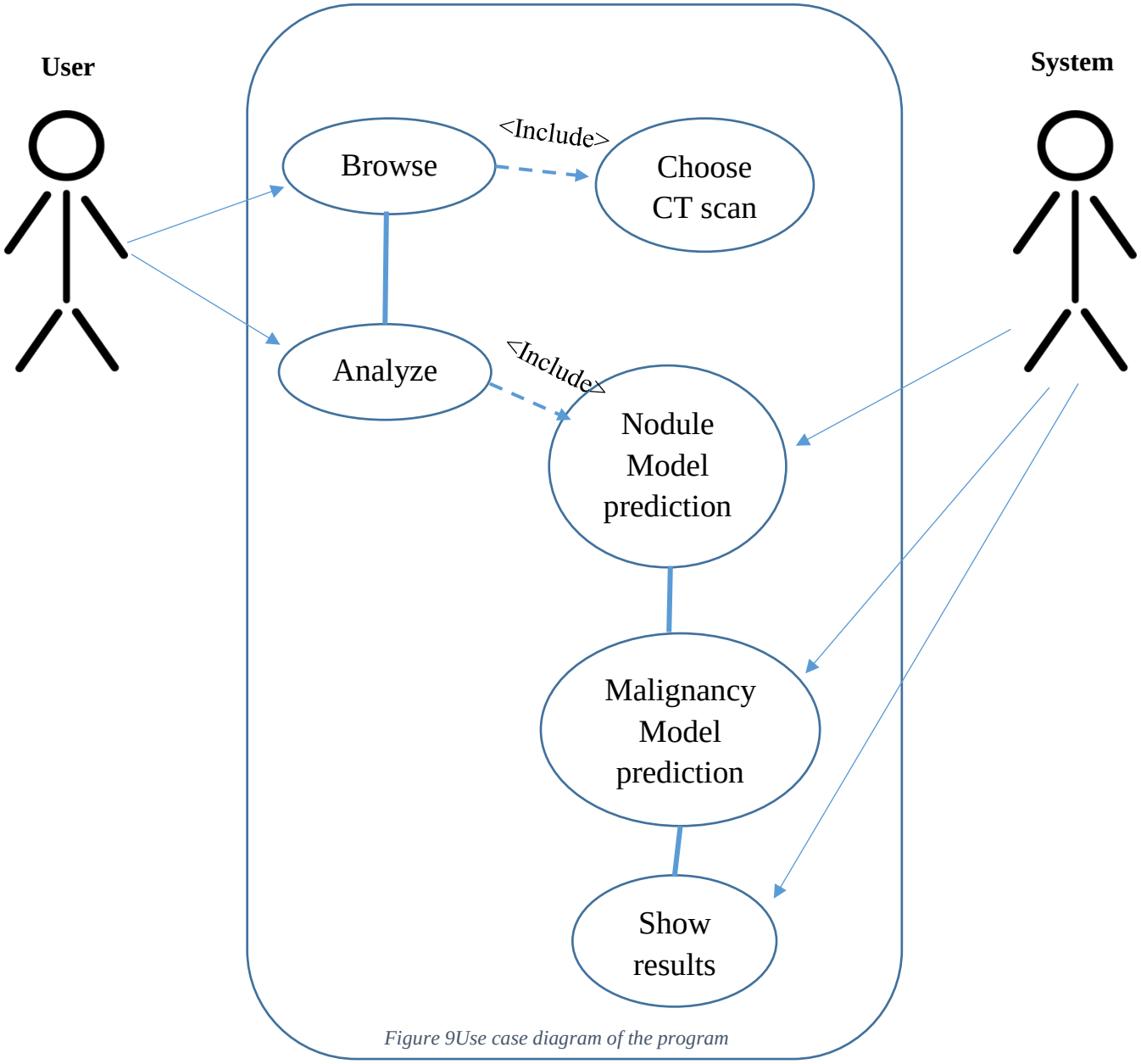


Figure 9Use case diagram of the program

Chapter 4 The proposed solution

4.1 Introduction to supervised Learning:

Supervised learning is where an algorithm is given input-output pairs and tries to predict the mapping function between them. This kind of data set that have input-output pairs is called a labeled dataset. The goal of the algorithm is to make a mapping function so well so if the function was given new sample it will predict the output of it correctly. Supervised learning have to branches regression problems and classification problems. Regression problems are when the output value is a real number. This real number represents things like money, temperature and weight. Classification problems are when the output value is a category, such as 'Dog' or 'cat' or 'disease' and 'no disease'. This project will focus on the classification problem since the output of the scan are categories ('nodule', 'no-nodule', 'number between 0 and 4'). In supervised learning for each input we know the exact output of it. Training is the process where the algorithms is finding the best function to map the input and output. It gets better and better incrementally with every sample it trains on. So having a larger dataset will eventually result in better accuracies **(Castle, 2017).**

4.2 Neural Networks:

One of the most common algorithms used in supervised machine learning is neural networks. Neural networks is loosely based on the human brain analogy where neurons fire when they are given enough stimuli. Also neural networks have neuron they are called nodes and the connections between that nodes that directs which nodes to fire and which not to fire are called weights. Neural networks consists of layers, a group of nodes, nodes and weights between the layers connecting the nodes with each other. Every node might have an activation function to help the network learn non-linear problems. The diagram bellow will illustrate the neural network parts **(Dormehl, 2018)**.

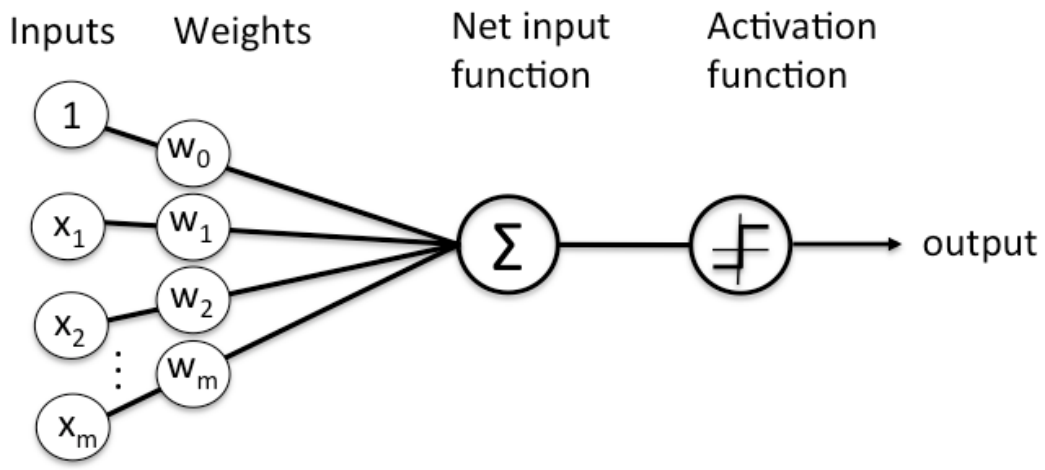


Figure 10Neural Network blocks from the 10th refrence

Neural Network have Input layers, output layer and hidden layers. A hidden layer is any layer that is between the input and output layer. Neural networks find the features and extracts them automatically without human intervention then make a decision whether those features belong to a certain class or another. Now neural networks have to be trained first. The training of the neural networks is slight changes to weights to get closer and closer to the desired output. The steps of making a neural network will be discussed below **(Dormehl, 2018)**.

4.2.1 Making the architecture of the neural network

Some of the layers of the architectures are determined by the input and output. The input layers must have as many nodes as there are inputs. If there are three inputs that means there is 3 nodes in the input layer. And sometimes it four nodes because a bias node is added. A bias node is always gives an input of one but its weights to the nodes in the following layer are tuned normally. This bias helps to get to the routing function between input and output better. And the output layer is determined by the number of classes. If you have four different classes then you have four nodes in the out layer. Now the hidden layers are parameters to tune to get the best accuracies. How many hidden layers and how many nodes in each hidden layer is a parameter that can be tuned to get the best accuracies. A deep network means it have many hidden layers and the more hidden layer the more complex problems the network can solve and the more time it will take training. Training is going through each sample of the dataset and forward propagating through the network then backward propagating. This will be discussed in the following sections **(Dormehl, 2018).**

4.2.2 Forward propagation

$$\sum XiWi$$

The above equation is the forward propagation equation. It takes the each of the inputs from the previous layer and multiplies it with the corresponding weights then adds them all together to produce one node on the upcoming layer. Now this is done to all the nodes in the upcoming layers as well. To optimize this usually matrix multiplication is used. Because a matrix multiplication multiplies an element of the matrix with the corresponding element in the other matrix (row and column) and adds them all together. Using matrices is not only more readable and easy to understand it is way more efficient in GPU computing and takes a lot less time. Now after the new value of the preceding node is calculated it usually goes through an activation function. Now the purpose of the activation function is to make the numbers non-linear as this way the network can learn to generalize and to learn more. There are many activation functions to use and this can be parameter to use in the training process. Here some will be discussed. There is sigmoid, relu , tanh and softmax activation functions. All of which are used after the summation of multiplied elements. Now the softmax is used in the output layer. It produces number from 0 to 1 among all nodes. And the summation if the nodes after the softmax will be 1. The number that node ends up with is the percentage that this is class is the appropriate class for the input. So the softmax is

usually used only in the output layer. The figures bellow show some of the activation functions and how they behave **(Dormehl, 2018).**

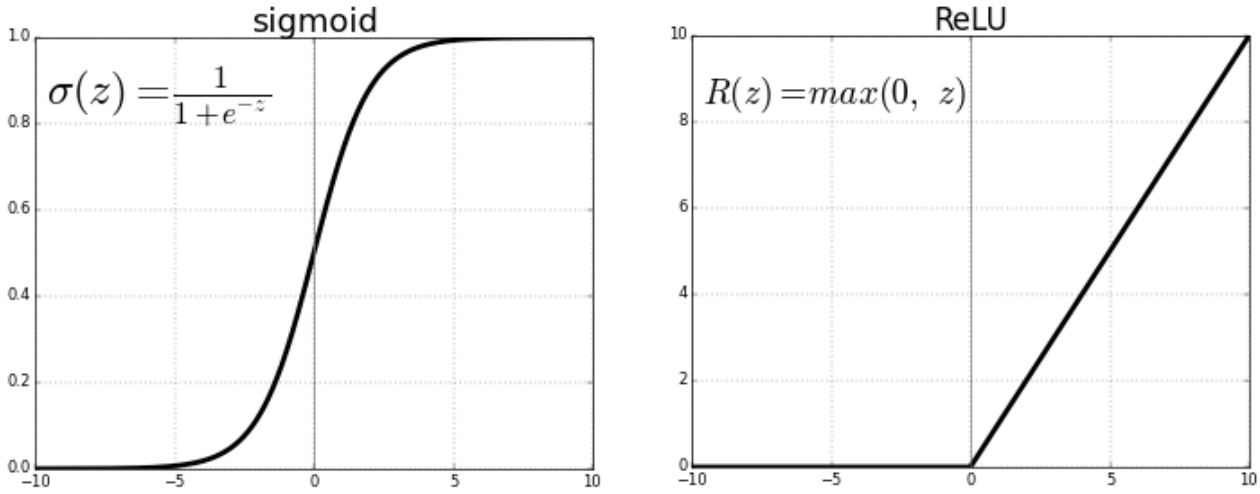


Figure 11 activation function examples from the 7th refrence

4.2.3 Loss function

Now that we got the network to predict the output of a give input we need to compare how different the output is from the desired output. This is where the loss function comes in. a loss function can be something as simple as the squared difference between the predicted output and the desired output. The graph of the loss function should be something like a U shape if it only has one parameter to tune where the Y axis is the loss and the x is the paramters tuned. An example of a loss function is below **(Dormehl, 2018).**

$$\sum(ypredicted - y\ desired)\hat{}2$$

4.2.4 Backpropigation

Now if we want to reduce the loss the function beginning in any random point we get the slope and we minus that from the point and get another point that have a slightly lower value of the y which means a lower loss and more accuracy. The slope is got by the derivative of the loss function. Then old weights is the derivative of the slope this will get you the new weights. And this is the backpropigation in a nutshell **(Brownlee, 2016)**..

4.3 Deep learning:

Deep learning networks are neural networks but with many hidden layers in between the input and the output. In each nodes in each layer learn to extract features. Deeper layers detect more complex features. For example detecting a face, early layers detect edges, deeper layers detect more complex features like eyes, nose and mouth. And deeper layers detects the whole face. This is called feature hierarchy. The figure illustrates the previous example.

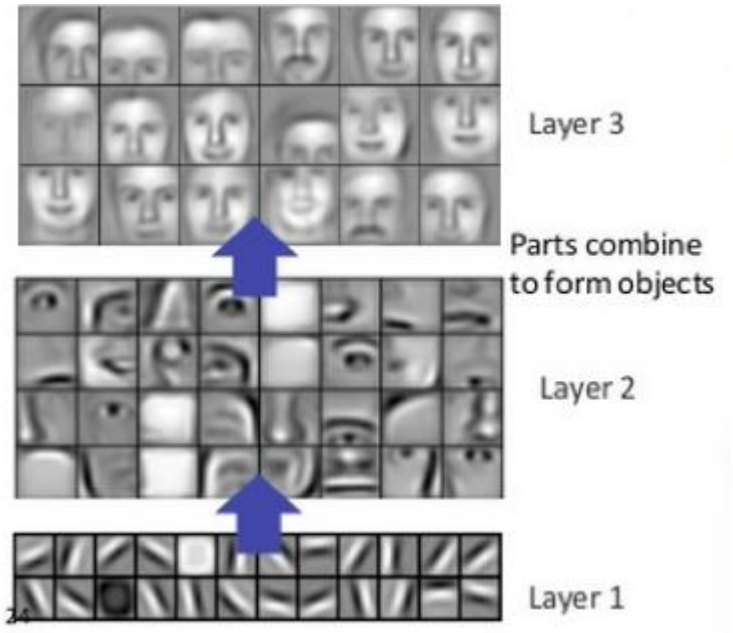


Figure 12 what network learns in each layer from reference 16

Deep learning preforms automatic feature extraction and the more layers it has the more complex features it can detect. There are many deep learning models each optimized to make a certain task and each has a paper proving that this architecture works best to solve a certain problem. Deep learning suffers vanishing and exploding gradients. Vanishing gradient means that the gradient the updates the weights gets smaller and smaller thus not affecting the weights of the earlier layers this result in the model not training as well. Exploding gradients mean that the gradients gets very big at the earlier layers and this will make weights change in a fashion that will not reach the optimal solution thus not training the model effectively. A number of different approaches are used to combat this. First the relu activation was invented to decrease this problem and it certainly did. Then batch normalization and normalizing inputs are very two important ways that decreases this phenomenon. Batch normalization is normalizing every node at a certain layer making their number range between zero and one. Input normalization is the same thing but with the input **(Brownlee, 2016)**.

4.4 Convolutional Neural Networks:

One of the most common fields that deep learning is taking a part of is Convolutional neural networks. Now this type of network is optimized to be used for images. A lot of deep neural networks are CNNs. CNNs has three main layers a convolution layer, a pooling layer and dense layer **(Zawadzki, 2018).**

4.4.1 Convolution layer

Convolution is used to decrease the number of weights that is tuned. Images can get really big and having a weight for each pixel is just inefficient. It will be time and memory consuming to train all those weights. So better way to do it is to use convolution. Those will be just a couple of weights that are run through all the image. The figure bellow shows how the convolution operation is performed. The weights get multiplied against the corresponding input and then added all together **(Zawadzki, 2018).**

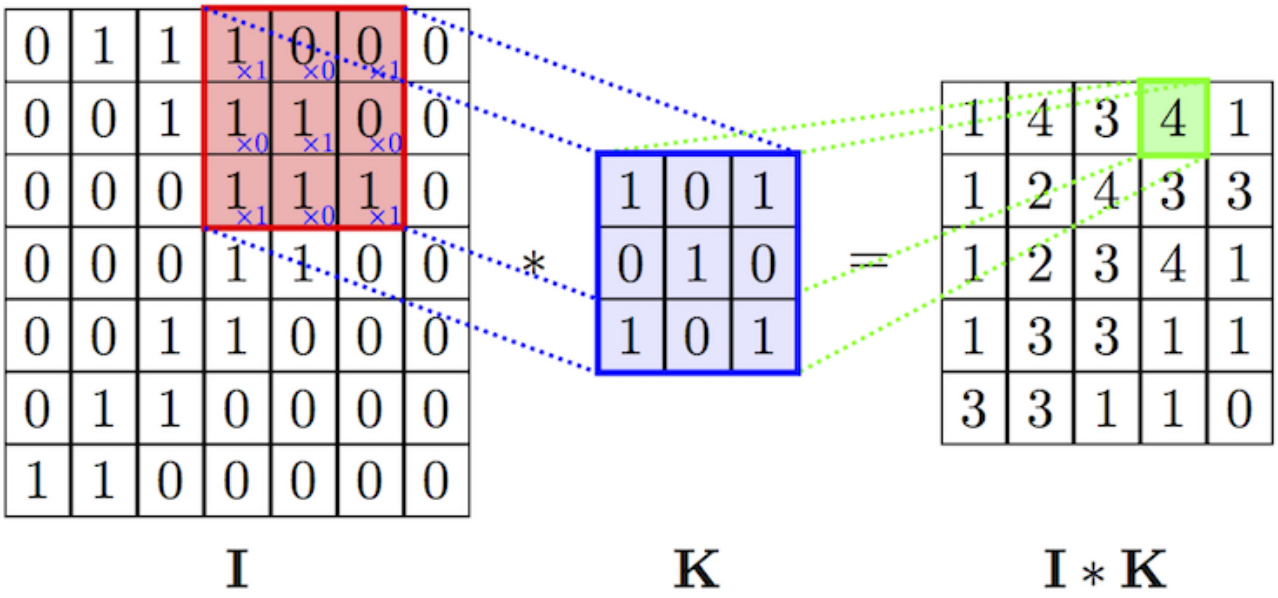


Figure 13convolution example from reference 16

4.4.1.1 Padding

As you can see in the figure above the dimensions of output 'I*K' is smaller than the dimensions of the Input 'I' due to the convolution operation by multiplying it with the kernel 'K'. So in order to take control of the shape or dimensions of the output we add padding and we change the stride. Padding is just adding zeroes to the edges of the input so when the dimensions shrink it shrinks to its original size. Sometimes

we need the dimensions to not shrink in deep CNN so that the image dimensions will not be zero **(Zawadzki, 2018).**

4.4.1.2 Strides

A stride is another way to control the dimensions of the output layer. It is the number of cells that the kernel jumps to make the next operation. In the figure bellow shows a stride of two **(Zawadzki, 2018).**

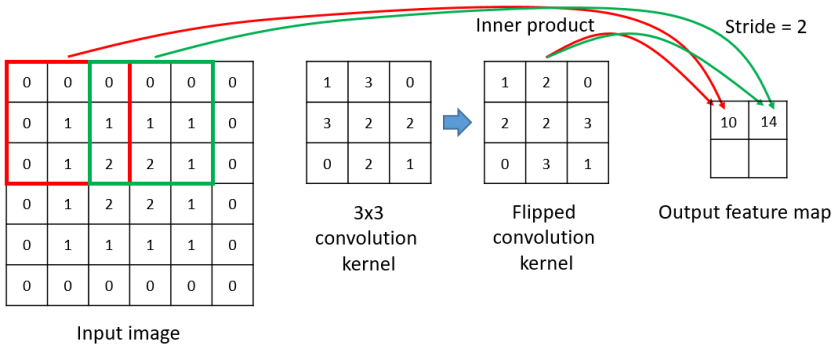


Figure 14stride "2"example from reference 16

4.4.2 Pooling layers

Pooling layers are layers that reduce the size of the input. By taking the only the most important features of the image. There are two common types of pooling average pooling and max pooling. Average pooling takes the average values in the kernel and represent them as one value. The maximum just take the maximum value in the kernel and put it as output **(Zawadzki, 2018).**

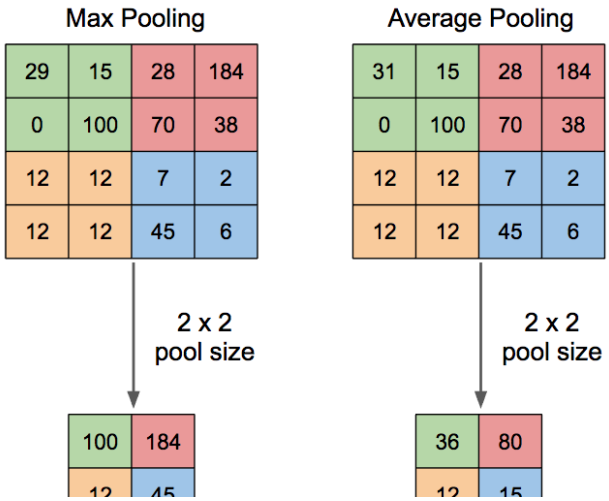


Figure 15 Max pool and average pool example from reference 16

4.4.3 Fully connected layers

Now this layer usually starts by flattening out the features. That means putting whatever shape from that was generated from the convolution and pooling layers into a single column of nodes. Then it proceeds by putting at least one more layer of nodes. The last layer must also have the number of nodes equal to the number of outputs. This helps the network to use the extracted features to classify the image.

4.4.4 Up convolution

This type of operation is only performed in segmentation models where we want the dimensions to expand after it shrunk. Now this is a normal convolution but expands the input before making the convolution by two ways. Padding and fractional strides. Padding becomes very big so that the output shape will be larger than the input shape. Another way to approach this is by using fractional strides this puts the zeroes between pixels to increase the dimensions. This and the other operations will be used in the U-net in the upcoming chapters **(Zawadzki, 2018).**

4.4 Deep CNNs beating the state of the art techniques at classification

Deep CNNs are beating accuracies of every single algorithm there is at Classifying images. And with more and more data the CNNs are doing better and better. Also with faster GPU we can train deeper and deeper networks. CNNs can classify now many things accurately for example, types of dogs, types of animals and places all over the world. But what is most fascinating is that it can classify medical images as accurate as a doctor **(Zawadzki, 2018).**

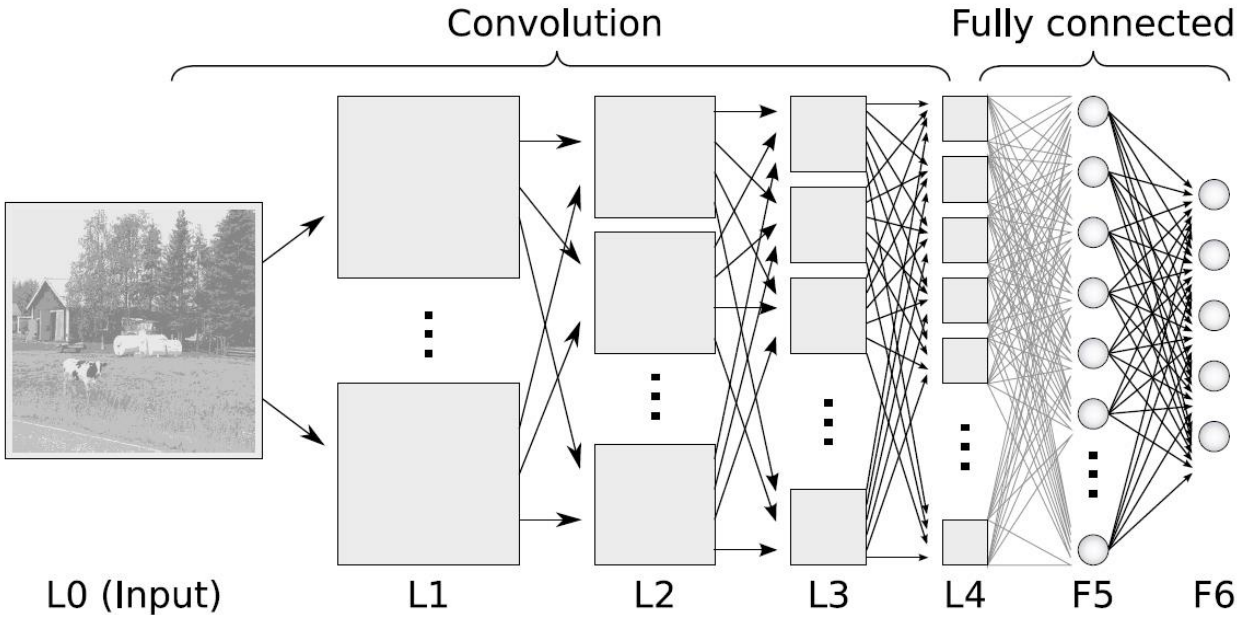


Figure 16CNN Example from reference 10

4.5 Using Deep CNNs for early diagnosis of Lung cancer

4.5.1 The needle in a haystack problem

Now as stated earlier this problem is not as easy as any image classification problem simply because the search space is too big and what the model is searching for is too small. This is called a needle in a haystack problem where the search is space of a problem is too big and what the model is looking for is relatively small. In this project the CT scan is 512X512Xdepth and the depth has a minimum of 124 and the nodule dimensions have a maximum diameter of 15 mm which is just a couple of pixels wide. In this situation the model trains poorly on the dataset and does not easily find the features it is looking for **(S.Hawkins).**

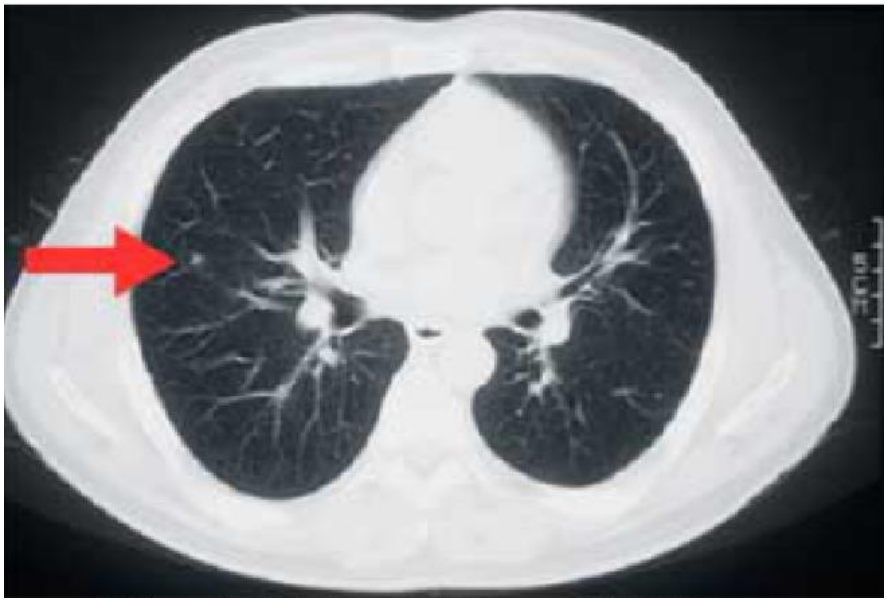
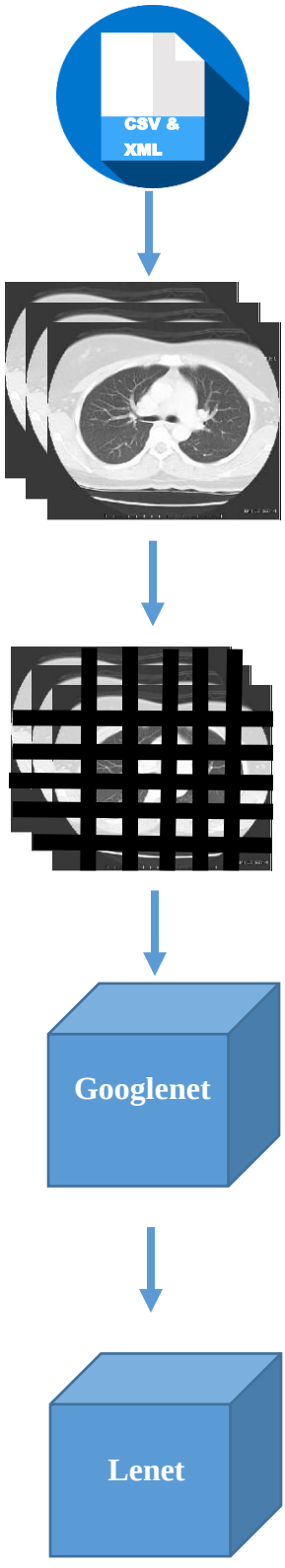


Figure 17 how small a nodule in a CT- scan slice

4.6 Machine learning pipeline for early classification

This chapter will only give the steps taken to diagnose a CT-scan. This will not include the trials and failed attempt. It will only include the result the last working pipeline. This pipeline has two models one for determining nodules and the other for determining malignancies.



1. Reading annotations: Reading nodule positions and characteristics from both a csv file and an xml file.

2. Loading CT-scan and preprocessing: Load CT in python and change world coordinates to voxel coordinates and normalizing the CT-scan.

3. Generate crops: Divide the CT into crops of 64 X64X64 patches to narrow down search space.

4. Nodule classification model: Train a model to detect if there is a nodule or not.

5. Malignancy classification: If there is a nodule put that patch in another model to classify which stage of malignancy is it by giving a number between 0 and 4. Zero being the least malignant and four being the most malignant.

Figure 18machine learning models Pipeline

Chapter 5 Implementation

5.1 Loading and preprocessing CT scans:

5.1.1 Read the .mhd CT scan and the csv file and xml annotations

This project used a python library called SimpleITK to read in the 3D CT scan which is a .mhd file. It reads the .mhd file as a list of 2d images of an undetermined size. This project used simpleITK.readImage to read in the CT-scan. This function returned a list of 2d slices. Each slice is a Hounsfield value ranging from 0 to 3000. Then those slices will be preprocessed in the following steps. This project reads the annotations.csv, which is a csv file that has locations of the nodules for each scan if it contains any. We iterate over each row in the csv and read the corresponding and proceed with the preprocessing. And lastly the annotations for the malignancy are xml files. Now the xml files contains information about each nodule in the scan. It contains many features one of which is the malignancy sore which ranges from 0 to 4 , 0 being the least malignant and four being the most malignant. The nodule position is referred by an edge map. Now an edge map is the positions of the pixels that are on the borders of the nodule. So to get the centroid of nodule I had to use the equation below. The summation of all the positions of x, y and z separately the divide it by the number of positions for each one to get the center for x, y and z.

$$(\sum Xi) \,/\, n$$
$$(\sum Yi) \,/\, n$$
$$(\sum Zi) \,/\, n$$

5.1.2 Converting world coordinates to voxel coordinates

The csv file gives us the position of each nodule but in world coordinates, but we can't locate the nodule on the scan using the world coordinates we had to use voxel coordinates. So a function was made to convert all CT scans. It takes as input the world coordinates, the origin and the spacing and return the voxel coordinates using a function given with the dataset. The equation it subtracts the origin from the world coordinates and then divide the result with the spacing.

$$voxel\ coordinates = (worldCoordinates - Origin\ ) \,/\, spacing$$

5.1.3 Normalizing the CT-scan

Now the CT-scan have Hounsfield values that range from 0 to 3000. This can be a problem neural networks. Because this will help the problem of vanishing and exploding gradients in deep learning. Vanishing gradients is a problem that occurs when the gradients used to update the weight become smaller and smaller as it reaches the beginning layers of the model. This will not change the weights of the beginning layers of the model and thus not learning as effectively. And the exploding gradients is the gradients getting bigger and bigger and thus changing the weights in the early drastically thus not giving a change for the model effectively.

Normalizing the input solves that problem and makes the learning process faster as the numbers are smaller to calculate. So first we determined the maximum and minimum range we want thet to normalize in. The lung Hounsfield values were between 400 and 1000 so we normalize accordingly. Where minHU is 400 and maxHU is 100.

$$Normalized = (currentHU - minHU) / (maxHU - minHU)$$

5.1.4 Generating training data

The search space in 3d CT scan is enormous, one .mhd can reach to 300 megabytes. The scan 512X512Xdepth where the depth can vary from 120 to 300. Now this search space is very huge and a deep CNN can't get good accuracies this way. In the testing and validation more will be said about the different approaches taken, but the best results are only discussed. This project now extracts patches of 64X64X64 given from the csv file annotated by nodule or no nodule. Then prepares the training data by saving those nodules as Numpy arrays and saving the corresponding Label as "one hot array", a one hot array is an array that has a length of number of labels in that case two "nodule" or no "nodule and all the array is zero except the index of the appropriate class is one this is proved to be optimal for training.

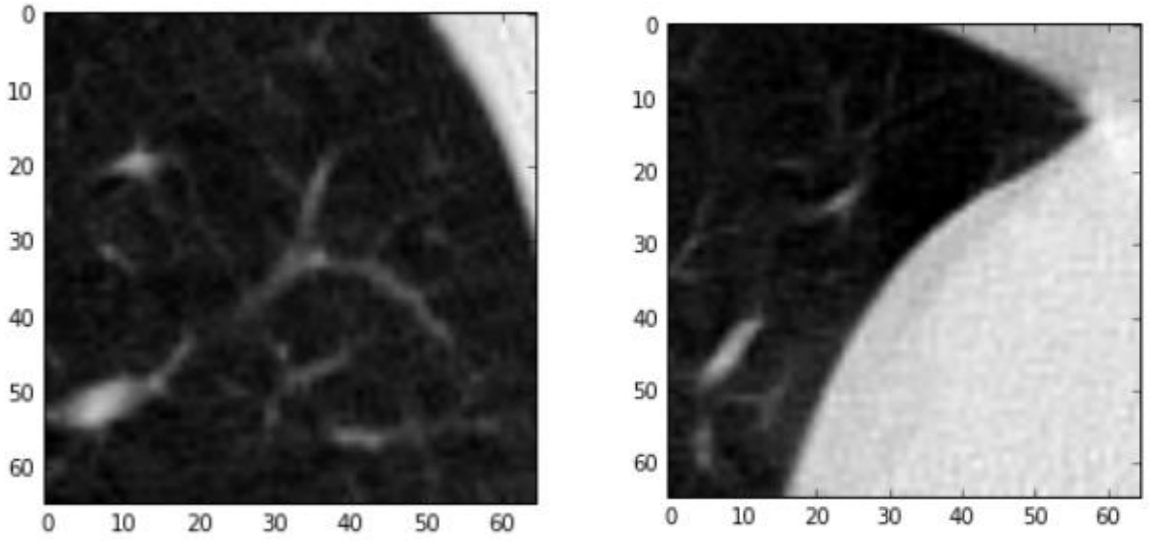


Figure 19Crops generated

5.1.5 Training the nodule classifier

Now that the training set is ready, we can put it into different classification models to predict whether this crop has a nodule or not. This project used three architectures Lnet 3D, Vanilla 3d and Googlent. All three models use cross entropy loss function, the Adam optimizer,

Relu activation function and dropout after each layer. The best accuracies was made by Goolenet reaching 99.6% more about the results in the testing section. Later this chapter all three architectures will be shown and discussed.

5.1.5.1 Dealing with skewed data

Skewed Data is to have a very big difference between the number of positive and negative samples in the training data. Now this can be a problem for example if 90% of the training data is negative and only 10% are positive. A model can get 90% accurate very fast but in fact all it's doing is predicting a zero (the negative sample) every single time. It will not predict one (the positive sample at all).The confusion matrix will be something like the figure bellow. There were no positives predicted. Now apparently this is model doesn't work well. In order to fix that we have to train the model in a different way introducing negatives incrementally. So for example assuming we have 80 negative sample and 20 positive ones. We first train the model with 10 positives and 10 negative samples till it reaches an above 90% accuracy that way make sure that the model predicts both negatives and positives alike. Then we increase the number of negatives and positives but the number of negatives increased is larger than the number of positives and continue training. And repeat that last step till you finish the data set and hope fully it will predict positives and negatives correctly.

	Negative	Positive
Negative	80(True negatives)	0(false negative)
Positive	20(false positive)	0(True positives)

Figure 20 Example of a confusion matrix of a skewed dataset

5.1.6 Malignancy classifier and reading xml annotations

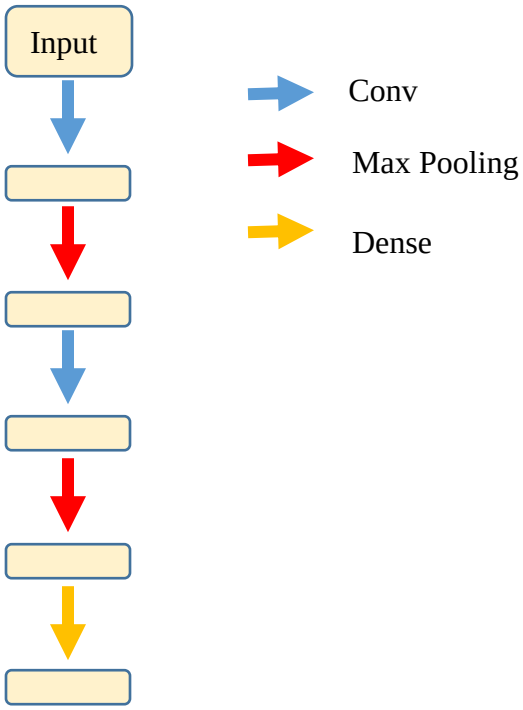
This project then LIDCR annotations and given the fact that the LUNA dataset is a subset from LIDCR, the project only downloaded the xml file annotations and used the already downloaded LUNA dataset. The annotations have nodules position in site map so getting the nodule centroid was calculated and there were features for this nodule in the annotations including malignancy. It ranged from 0 to 4 with zero meaning the nodule is benign and 4 meaning it is malignant level four which is more than the levels before it. Now that we have the centroid of the nodule and the malignancy level. The projects generates 64X64X64 patches from the centroid and makes the malignancy level as a one hot array. Then the same three architectures were trained the Lenet 3D, vanilla 3d and the googlent 3d. Here the best results were given by the Lenet which is unexpected as it was the least complex model. It reached 96% accurate on the training set. More on the models and accuracies reached later.

5.1.7 Machine learning model

All of the models used a convolution layer, max-pooling layer, and dense layers. There is also a simple model that is one convolution and one max and then dense layers, but it will not be shown as it is very simple and didn't reach good accuracies.

5.1.7.1 Lenet

The Lenet model is a simple model, it consists of 6 layers. It uses a dropout of 0.3 after each convolution and after the dense layer. It also uses the Adam optimizer with learning rate of 0.001.

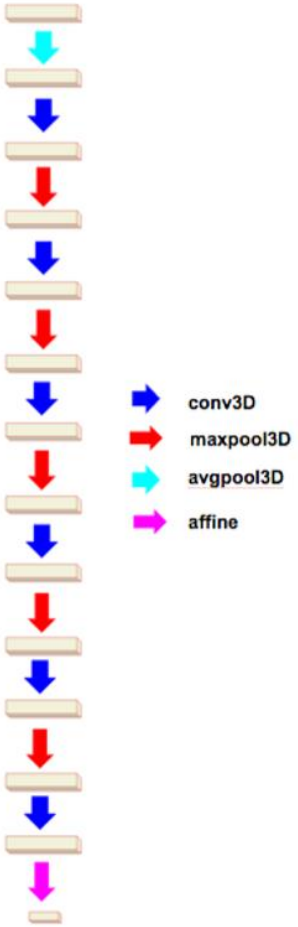


Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 64, 64, 20)	32020
activation_1 (Activation)	(None, 64, 64, 20)	0
dropout_1 (Dropout)	(None, 64, 64, 20)	0
max_pooling2d_1 (MaxPooling2	(None, 32, 32, 20)	0
conv2d_2 (Conv2D)	(None, 32, 32, 50)	25050
activation_2 (Activation)	(None, 32, 32, 50)	0
dropout_2 (Dropout)	(None, 32, 32, 50)	0
max_pooling2d_2 (MaxPooling2	(None, 16, 16, 50)	0
flatten_1 (Flatten)	(None, 12800)	0
dense_1 (Dense)	(None, 500)	6400500
activation_3 (Activation)	(None, 500)	0
dropout_3 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 6)	3006
activation_4 (Activation)	(None, 6)	0

Figure 21Lnet model

5.1.7.2 Vanilla 3D

The Vanilla 3d architecture has 13 layer. It uses average pool as well as the max pool. A dropout of 0.2 was used after every convolution layer and Adam optimizer was used with a learning rate of 0.003. IT has 3,475,366 trainable parameters.



Layer	Params	Activation	Output
Input			64 x 64 x 64 x 1
AvgPool	2x1x1		32 x 64 x 64 x 1
Conv1	3x3x3	ReLu	32 x 64 x 64 x 32
MaxPool	2x2x2		16 x 32 x 32 x 32
Conv2	3x3x3	ReLu	16 x 32 x 32 x 64
MaxPool	2x2x2		8 x 16 x 16 x 64
Conv3	3x3x3	ReLu	8 x 16 x 16 x 128
MaxPool	2x2x2		4 x 8 x 8 x 128
Conv4	3x3x3	ReLu	4 x 8 x 8 x 256
MaxPool	2x2x2		2 x 4 x 4 x 256
Conv5	3x3x3	ReLu	2 x 4 x 4 x 256
MaxPool	2x2x2		1 x 2 x 2 x 256
Conv6	3x3x3	ReLu	1 x 2 x 2 x 512
Dense			2

Figure 22Vanilla 3d model

5.1.7.3 Googlent

A new type of layer is used here, it's called an inception layer. In CNN you have to choose whether to do a convolution layer, or a pooling layer and you have to choose the kernel size, will it be 1 or 3 or 5. Why not do them all this is the inception layer it does all of that and concatenates the result. The figure bellow illustrates the inception layer. And when you concatenate all those possibilities it was proved that it learns better.

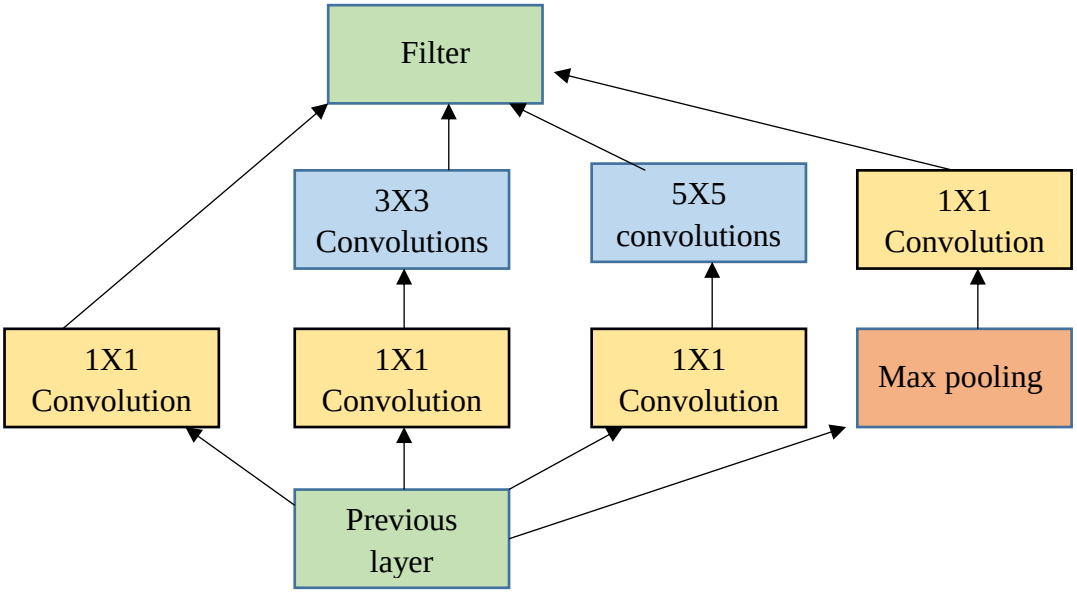
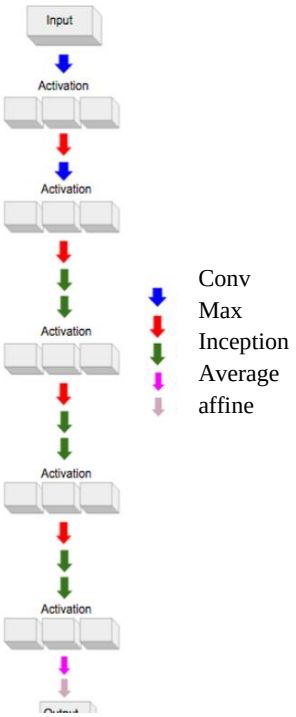


Figure 23inception layer

This model is the deepest model used here. It consists of 14 layers, 6 of which are inception layers which is practically 7 layers concatenated together as shown in the figure above. Dropout was 0.3 after and inception layer. The Adam optimizer was used with 0.0001 learning rate. It has 6,511,718 trainable parameters.



Layer	Params	Activation	Output
Input			64 x 64 x 64 x 1
Conv1	7x7x7	ReLu	32 x 32 x 32 x 32
MaxPool	2x2x2		16 x 16 x 16 x 32
Conv2	3x3x3	ReLu	16 x 16 x 16 x 64
MaxPool	2x2x2		8 x 8 x 8 x 64
Inception1	1,3,5	ReLu	8 x 8 x 8 x 128
Inception2	1,3,5	ReLu	8 x 8 x 8 x 128
MaxPool	2x2x2		4 x 4 x 4 x 128
Inception3	1,3,5	ReLu	4 x 4 x 4 x 256
Inception4	1,3,5	ReLu	4 x 4 x 4 x 256
MaxPool	2x2x2		2 x 2 x 2 x 256
Inception5	1,3,5	ReLu	2 x 2 x 2 x 512
Inception6	1,3,5	ReLu	2 x 2 x 2 x 512
AvgPool	2x2x2	ReLu	1 x 1 x 1 x 512
Dense			2

Figure 24Googlenet Model

5.1.8 GUI: Putting it all together

This project then proceeds by making an application. First it segments the lung from the whole scan using morphological techniques then make 64X64X64 patches and puts each one in the nodule classifier and the positive patches enter in the malignancy classifier then It prints out whether he has cancer or not , and if he has cancer at what stage is it. And all this is put into a GUI made with python so the user can interact with the program easily. The GUI displays the CT-scan once your browse the .mhd file. Then once you press it will display the nodules at and will also display the nodule count and the malignancy score for each nodule.

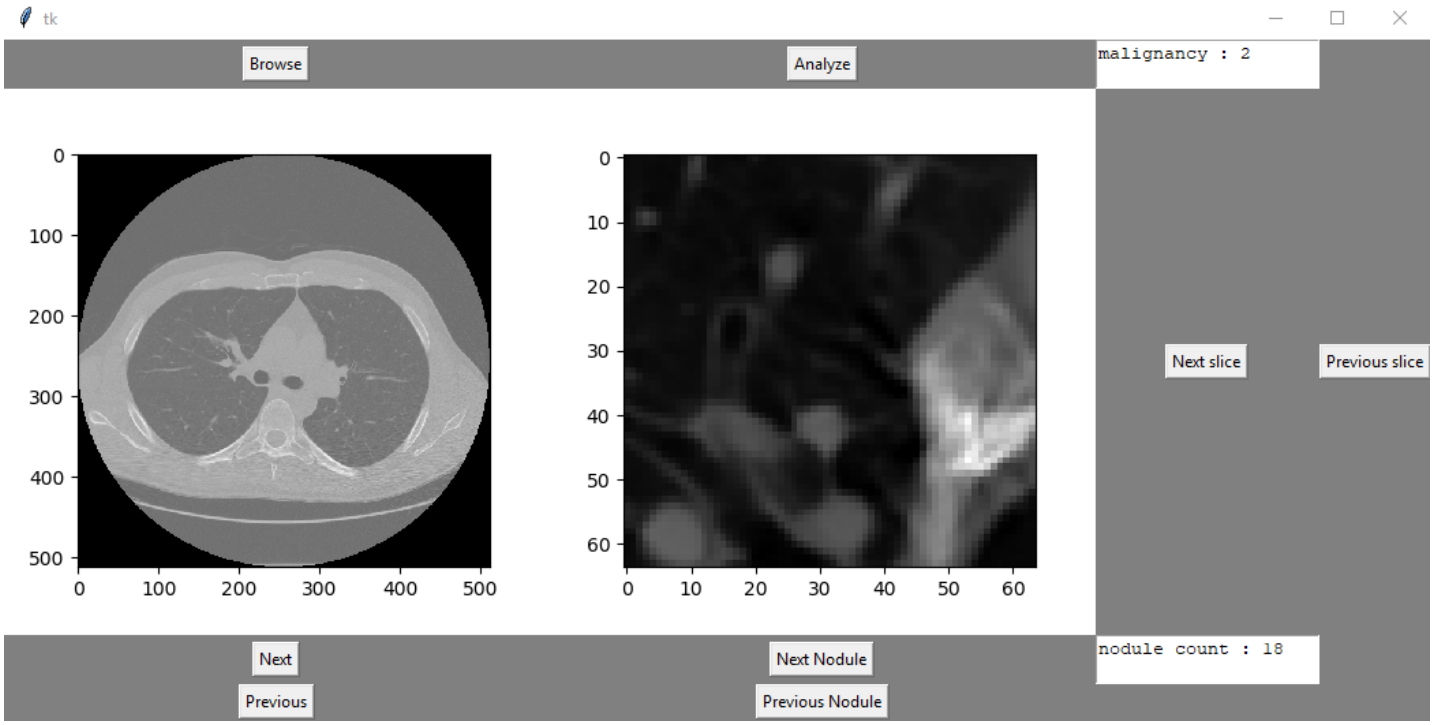


Figure 25GUI snapshot

Chapter 6 Results and Testing

A number of different approaches were taken. It took many tries to get to the accuracy that is reached now. This chapter will the many failed approaches that were taken and the different results of the different models.

6.1 Classifying the whole CT scan as cancerous or not

In this part of the project Kaggale's Data science Bowl data Set was used. First it reads all the scans and resized it and changed it from pixels to Hounsfield values and then we put it into 3 different Deep CNN architectures Lenet 3D, Vanilla 3D and Googlnet 3D best of which got 52% accuracy and Normal dataset where the number of positives equal to the number of negatives. Also tried to resize the data to smaller scans (150X150X20) and still got very bad accuracies. So classifying the scan as whole as cancerous or not was way out of the question.

6.2 Unet to segment nodules in 2D

This approach tried to narrow down the search space by segmenting the nodules using 2d masks. The masks were generated from the annotations as it gives x, y and z voxel coordinates and a diameter. From this information a mask was generated. The Unet architecture is one of the famous architectures for segmenting medical images. So it was predicted that it will work with flying colors. But in reality it generated masks that are all black. And although the accuracy is about 97% but that was because 97% of the image is black and we need the 3% percent which is the nodule. The result was that the generated masks was useless. The figure bellow shows what this paragraph is talking about.

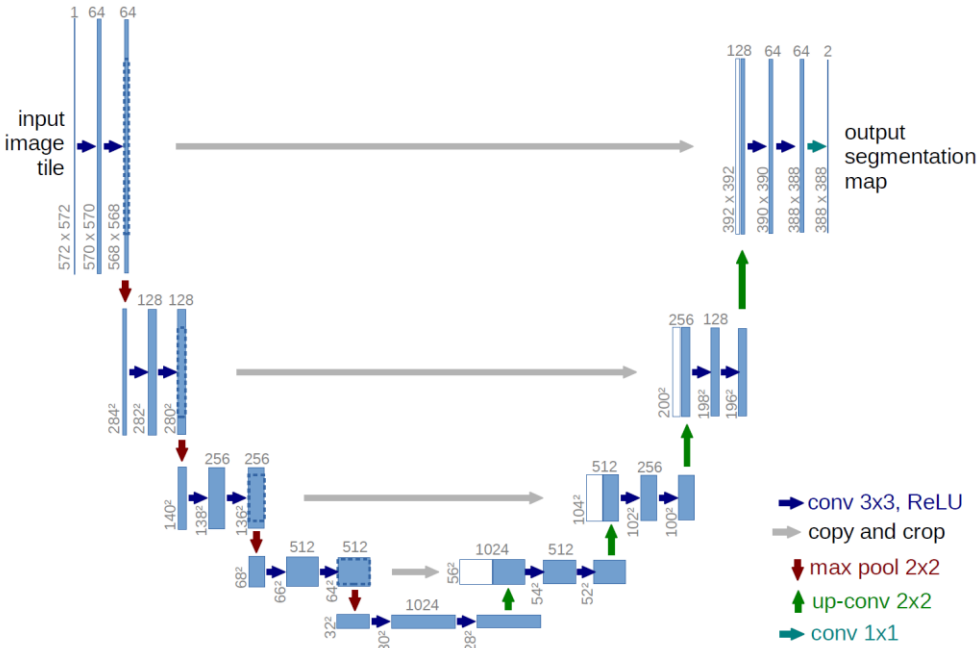
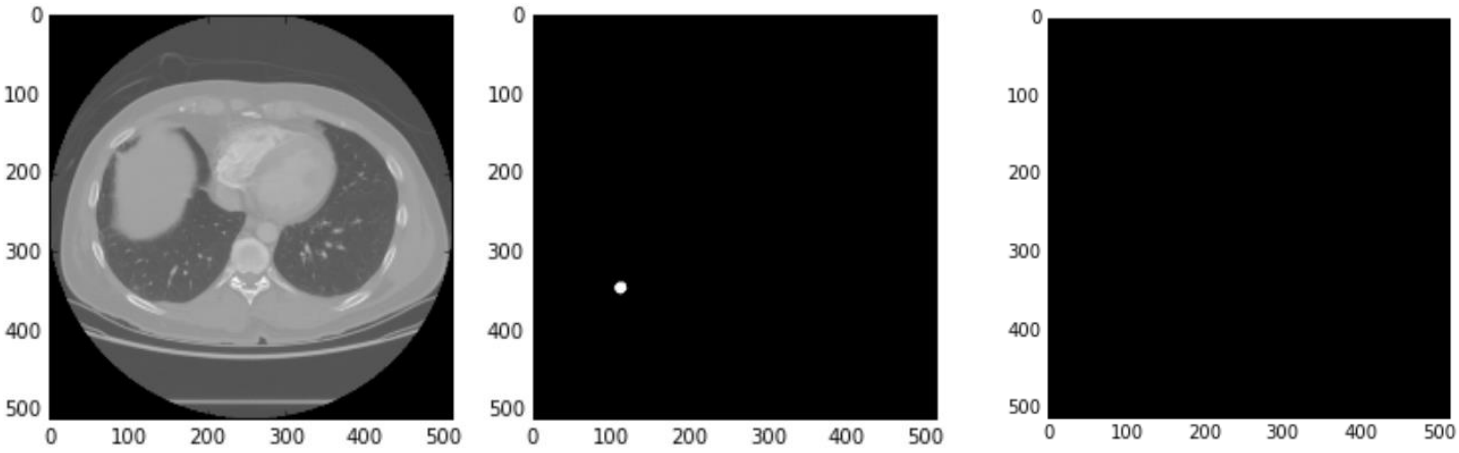


Figure 26 U-net segmentation model from reference 16



CT scan slice Desired Mask Generated Mask

Figure 27Unet generated masks

6.3 RADIO python library for medical imaging

RADIO is a python library made for processing medical imaging on python. This library simplifies a lot of preprocessing and it has some models like the U-net and the v-net for segmenting the nodule. This library was used to preprocess like converting from word coordinates to voxels and then make crops and masks. After the masks and crops are saved. Then it was used for the unet and vnet models. Unfortunately the results for both models are the same as the above model. The segmentation accuracy is about 93% but as you see here that is because 93% of the image is black and the 7% that is not correct is the white part which is the nodule. The nodule part is the only thing we will use in the next phases of the pipeline and it was not segmented. So those generated masks were unusable.

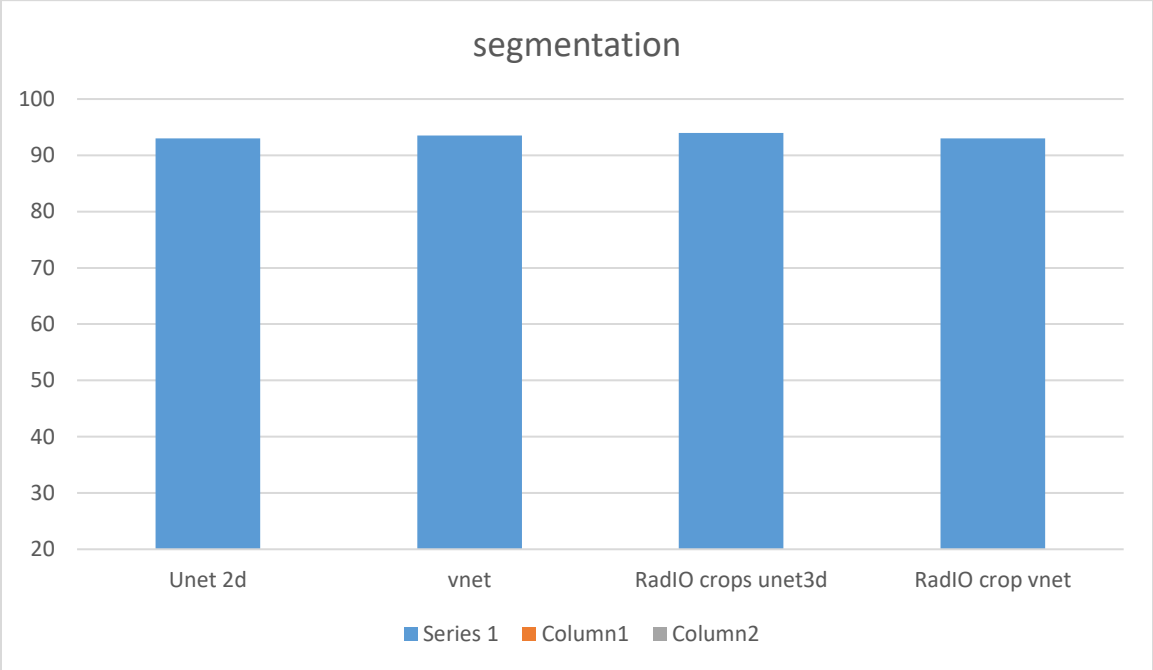


Figure 28segmentation accuracies

6.4 Generating 2 crops

Generating 2D crops was the beginning of accuracies. The crops were 64x64 in one slice only. This is where results were beginning to be promising. The project generated a dataset consisting of 64x64 images and it's desired whether it contains a nodule or it didn't. Then it was trained on Lenet model where it got 83% accuracy. This data was 3d data so the project also tried 3d crops of 64X64X64 on the same Lenet model and it got 95% accuracy. Taking advantage of the 3d nature of the dataset resulted in a higher accuracy. With this fact the project proceeded to make some more models and tweak the learning on the 3D crops.

6.5 Generating 3d crops for nodule classification

Generating the training set as 64X64X64 and classifying it as it contains a nodule or not as discussed in the above began to give some promising results with a simple CNN model. After those results, three more models were tested to reach the best accuracies. The Lenet 3d model, vanilla 3d and googlent. All got very good results but the best results we given by the googlent. The figure bellow show the accuracies for each model.

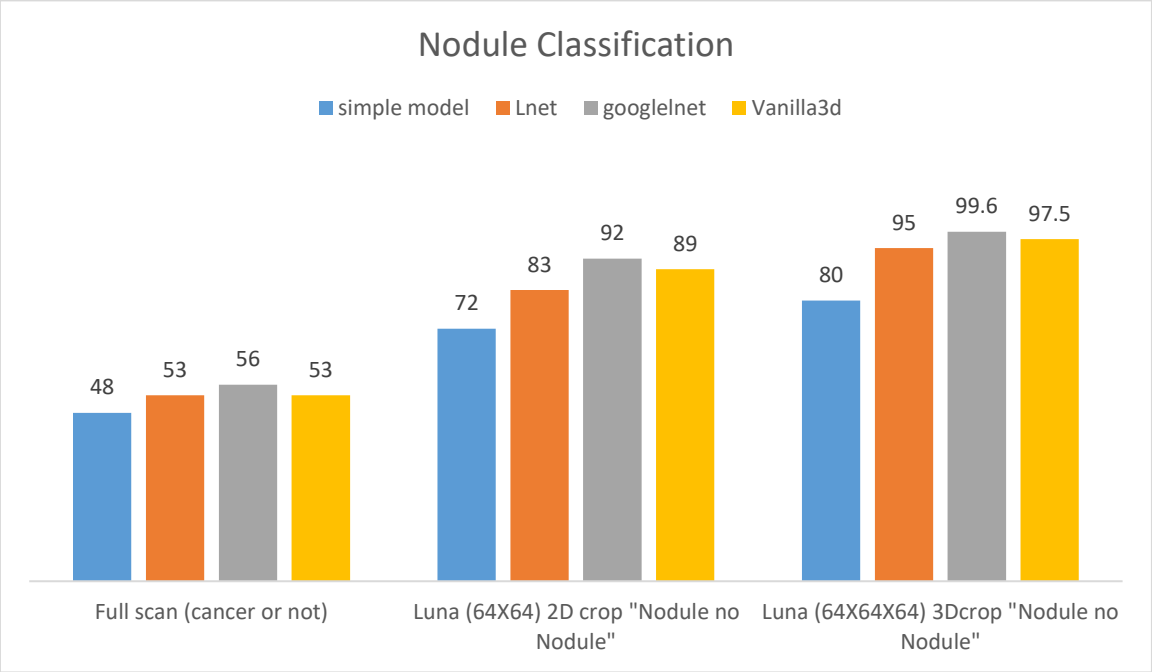


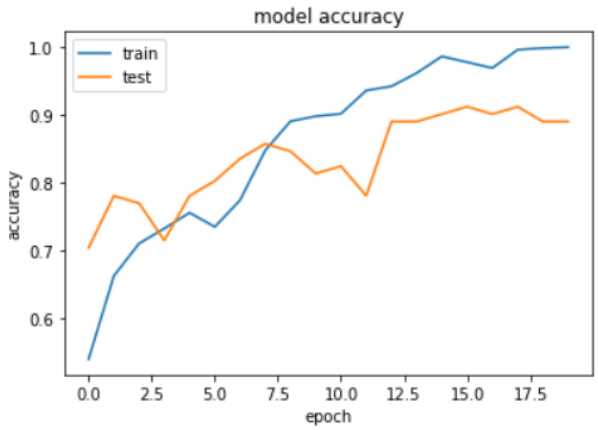
Figure 29Nodule classification accuracies

6.5.1 Loss and accuracy plots

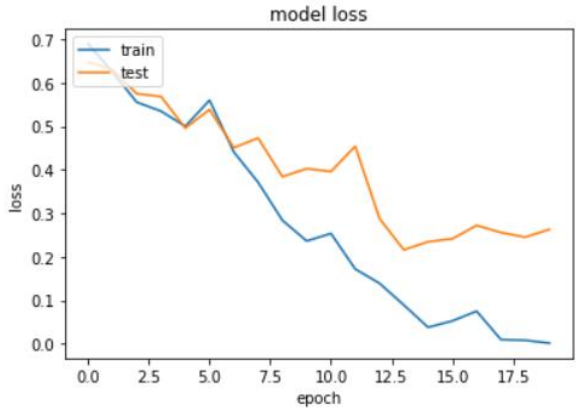
6.5.1.1 Vanilla 3D

A couple of learning rates were tested at this architecture to reach the best accuracy in the best times with the least number of iterations. In the end a learning rate of 0.0003 was chosen.

Learning rate 0.0003



Accuracy graph

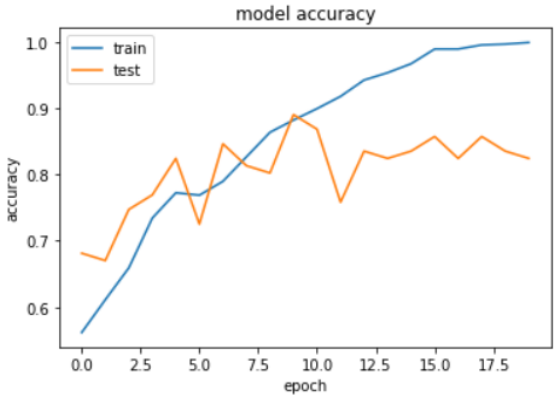


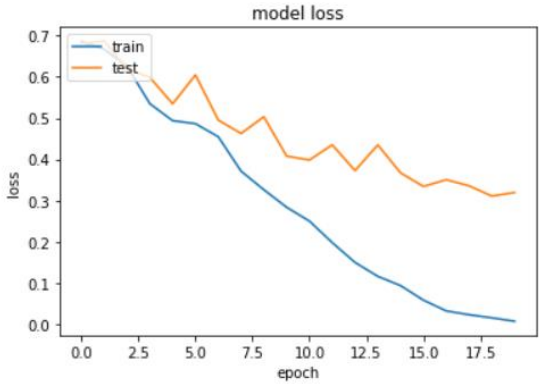
Loss graph

Figure 30Vanilla3d loss and accuracy graphs at learning rate 0.0003

At learning rate 0.0003 the accuracy got higher at a nice rate and the data did not over fit reaching nice accuracies in both the training and test sets. The accuracy of the training set is increasing and it plateaued of near the 96%. Also the loss of the training set is decreasing in a normal fashion that is close to the log slope. It did not reach In 0.0001 it took more time to get to the accuracies and it over fitted a bit. In 0.001 it did not learn very well.

Learning rate 0.0001





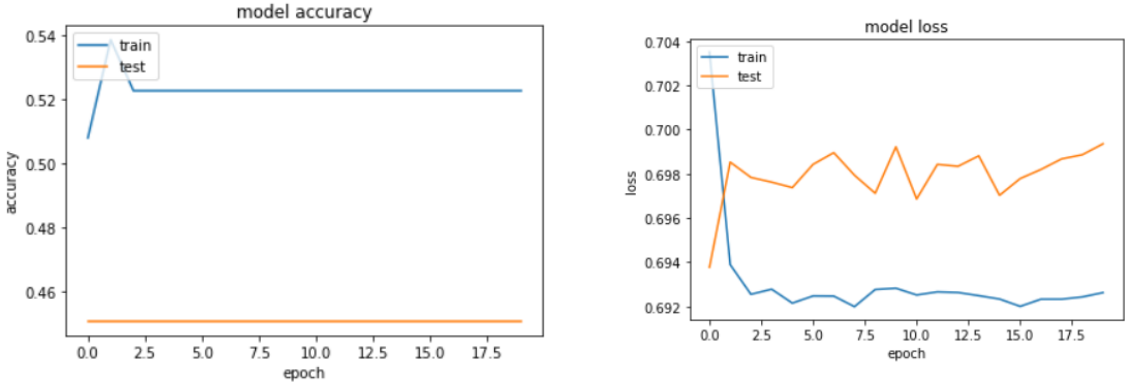
Accuracy graph

Loss graph

Figure 31Vanilla3d loss and accuracy graphs at learning rate 0.0001

In this graph it shows that using the same number of epochs it reached a slightly lower accuracy with the same number of epochs. Now this can reach a higher accuracy and a lower loss as the graph shows it did not plateau, but it will take more epochs and thus more time to train and in the end, it will reach the same accuracy as the above. But the above learning rate will use less time than this one accuracy, thus using the above learning rate and not this one.

Learning rate 0.001



Accuracy graph

Loss graph

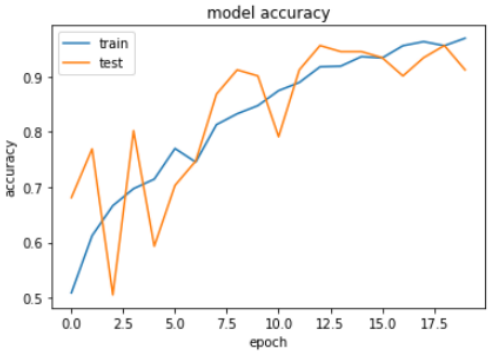
Figure 32Vanilla3d loss and accuracy graphs at learning rate 0.001

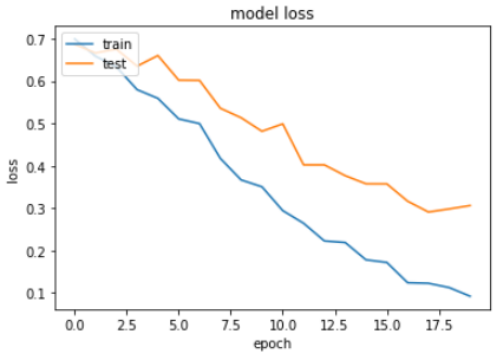
This learning rate was very large so the model start jumping of and missing the global optimum thus not learning at all and thus a flat line is presented above. This shows that this learning rate is very high for this problem and had to be decreased so that the model can actually learn something and decrease the loss.

6.5.1.2 Googlent

Also in this architecture a couple learning rates were tested to reach the best accuracy. 0.0001, 0.0003 and 0.001. The best results were from 0.0001 as this learning rate got the highest accuracies.

Learning rate 0.0001





Accuracy graph

Loss graph

Figure 33Googlenet loss and accuracy graphs at learning rate 0.0001

This learning rate is better than the last leaning rate as you can see here that the model generalized well, and the test and training accuracy were very close to the testing accuracy this learning rate got the best results in both training and test accuracies.

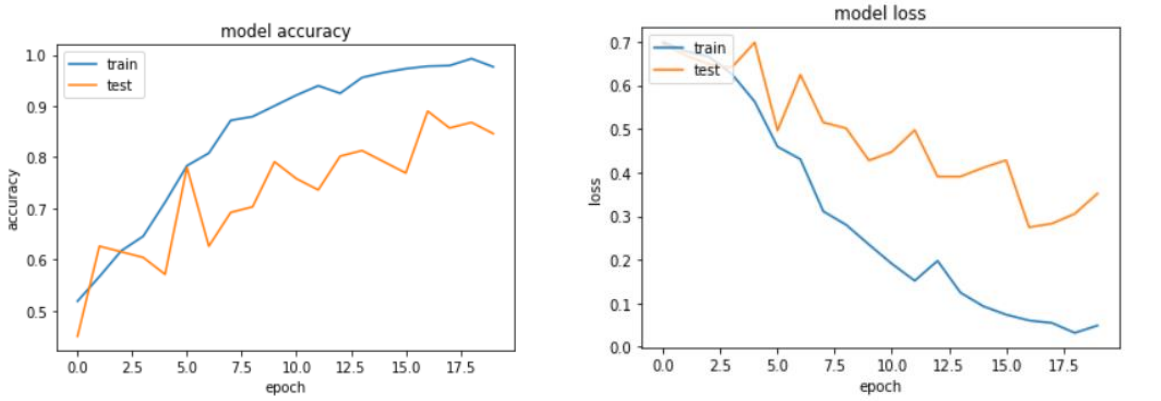
```
from sklearn.metrics import confusion_matrix
predicted = googlenet.predict(xtrain[:])
pre = np.argmax(predicted,axis=1)
tru = np.argmax(ytrain[:],axis=1)
confusion_matrix(tru, pre)
```

```
array([[ 507,   23],
       [  12, 4283]])
```

Figure 34Confusion matrix at best results

This is the confusion matrix show the true positive which is the top left and true negatives (bottom right), false positive (top right) and false negative (bottom left) as you can see the true positive and negative are the biggest numbers and thus good results.

Learning rate 0.0003



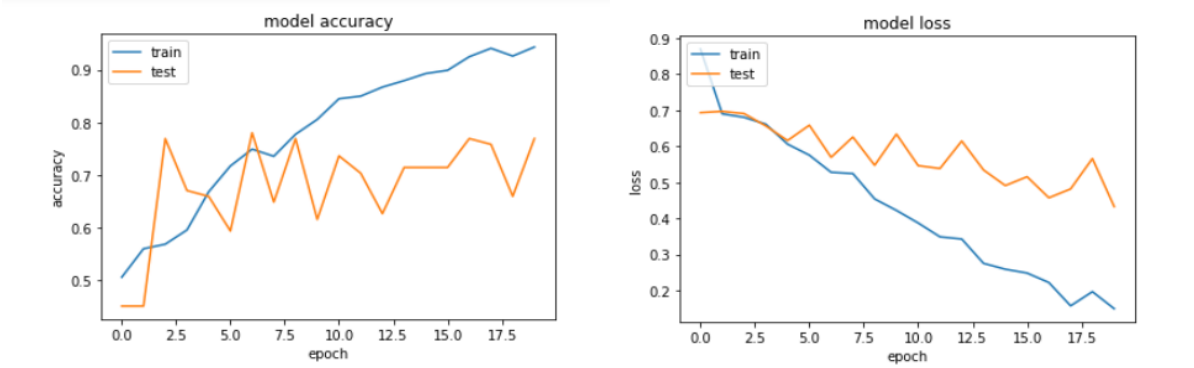
Accuracy graph

Loss graph

Figure 35Googlenet loss and accuracy graphs at learning rate 0.0003

This learning rate here is performing fine and plateaued off at a high accuracy but eventually they were big steps for the model to be able to generalize well and as you can see in the there is a big difference between the training accuracy and the test accuracy. better results were made in the bellow learning rate.

Learning rate 0.001



Accuracy graph

Loss graph

This learning rate plateaued off at a lower accuracy than the two above it so this learning took too big steps for this problem to solve this problem so it just bounced off around the global optimum resulting in a lower accuracy.

6.6 Malignancy classifier

The same three networks were trained on the malignancy classification problem. And unexpectedly the Lenet Model got the highest accuracies. It was trained on 64X64X64 crops and got the annotations from LIDCR xml file. All models tried different learning rates to reach best accuracies and all models used the Adam optimizer.

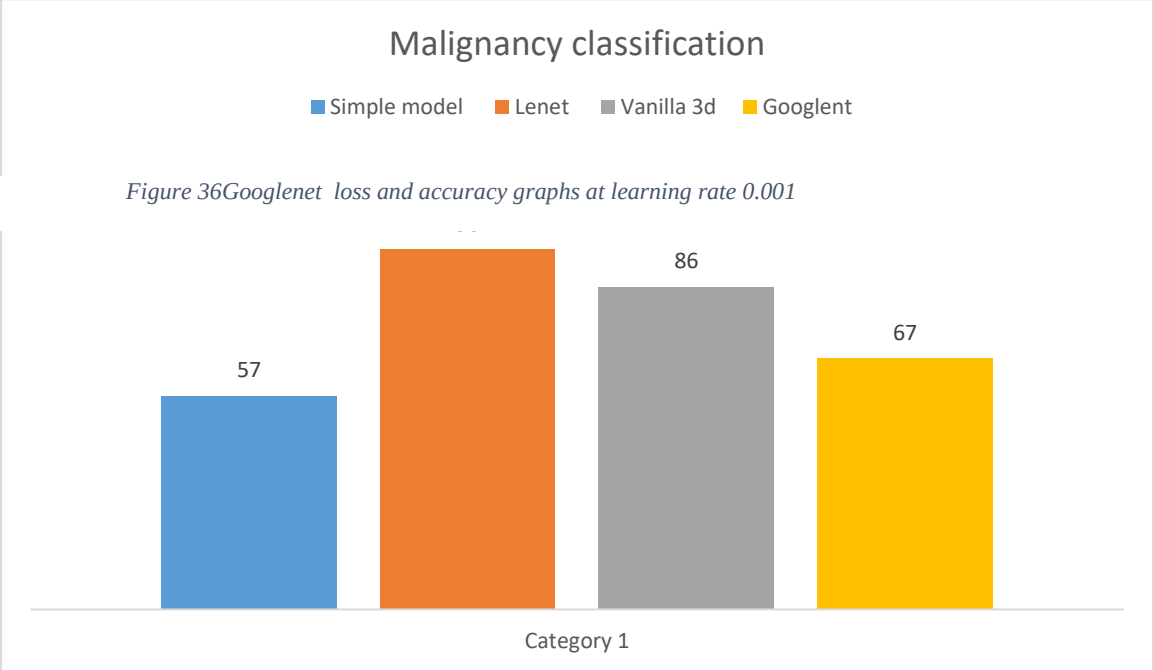


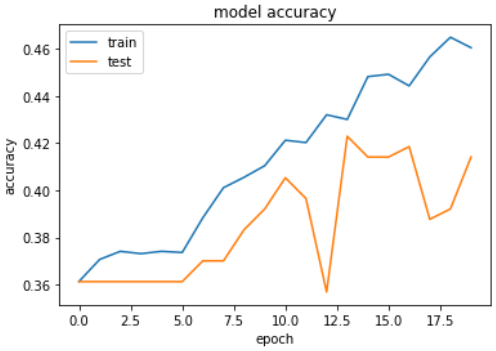
Figure 36Googlenet loss and accuracy graphs at learning rate 0.001

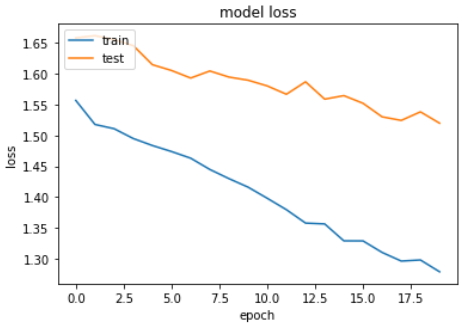
Figure 37Accuracy graphs of malignancy classification

6.6.1 Googlent

Although this is the deepest CNN it performed poorly on this problem and it only got 67% accuracy. Three learning rates were tested 0.00001, 0.0001 and 0.001. The Adam optimizer was used and a dropout rate of 0.2 after each convolution and inception layer.

Learning rate 0.00001





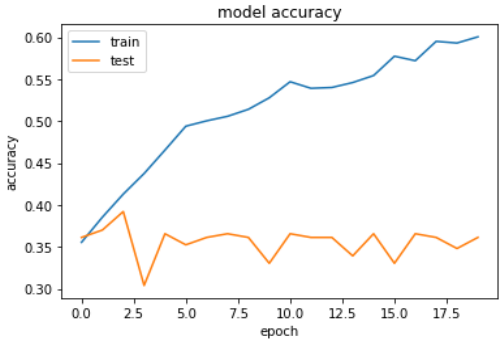
Accuracy graph

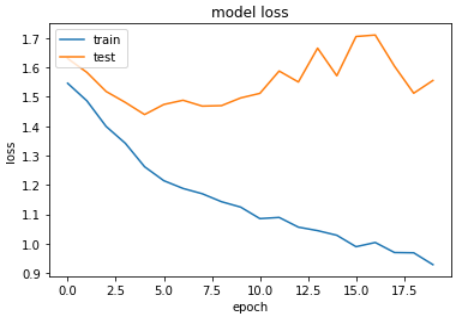
Loss graph

Figure 38Googlenet loss and accuracy graph for malignancy at learning rate 0.00001

Googlent did not reach accuracies as you can see above it reached around a 50% accuracy. It was trained on more epochs, but the model got stuck here and did not increase the accuracy. This learning rate is the lowest among the others used bellow it should get the highest accuracy so the others are not expected to do good as well.

Learning rate 0.0001





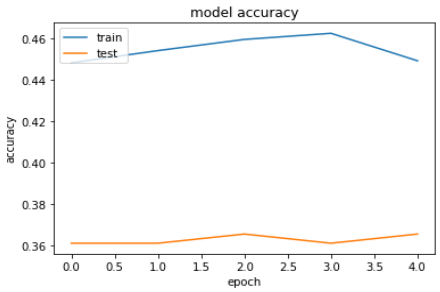
Accuracy graph

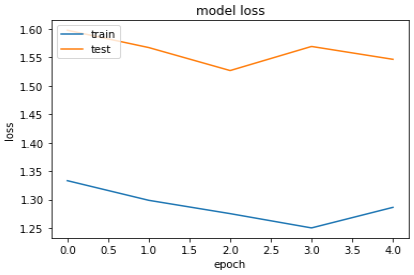
Loss graph

Figure 39Googlenet loss and accuracy graph for malignancy at learning rate 0.0001

Here the project used another lower learning rate just to make sure that the above accuracies was not a glitch. It eventually plateaued of at the same point. Thus this learning rate and all the model is not efficient at this problem.

Learning rate 0.001





Accuracy graph

Loss graph

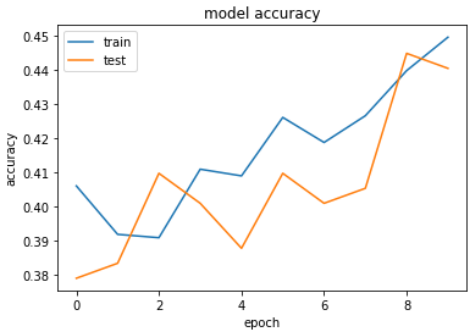
Figure 40Googlenet loss and accuracy graph for malignancy at learning rate 0.001

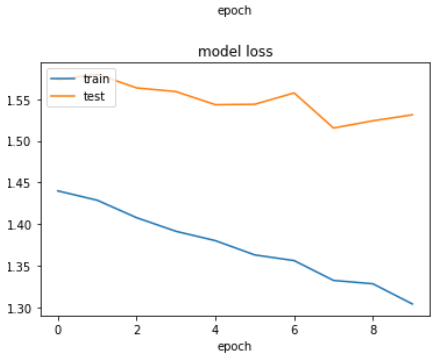
This learning rate was very large so the model start jumping of and missing the global optimum thus not learning at all and thus a flat line is presented above. This shows that this learning rate is very high for this problem and had to be decreased so that the model can learn something and decrease the loss.

6.6.2 Vanilla3d

In this architecture three different learning rates were tested, 0.00001, 0.0003 and 0.0001.It got 86% accurate at its best. It used the Adam optimizer and a dropout rate of 0.3 after each convolutional layer.

Learning rate 0.00001





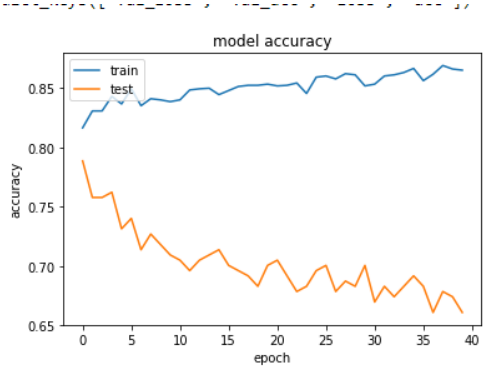
Accuracy graph

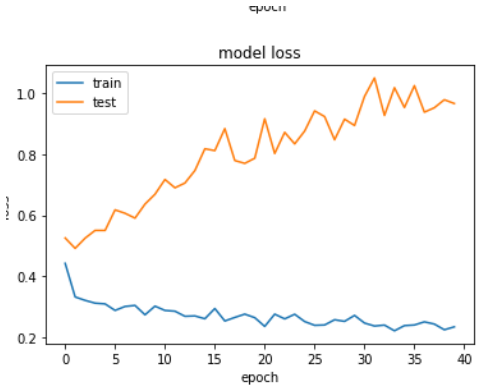
Loss graph

Figure 41Vanilla 3d loss and accuracy graph for malignancy at learning rate 0.00001

Vanilla 3d suffered from the same problem googlnet suffered from it was not able to learn above a certain point and. This learning rate is very low so if the global optimum of this was lower than 50% this learning rate should actually reach there, but eventually it did not reach good accuracies and the following accuracies are not expected to good as well.

Learning rate 0.0003





Accuracy graph

Loss graph

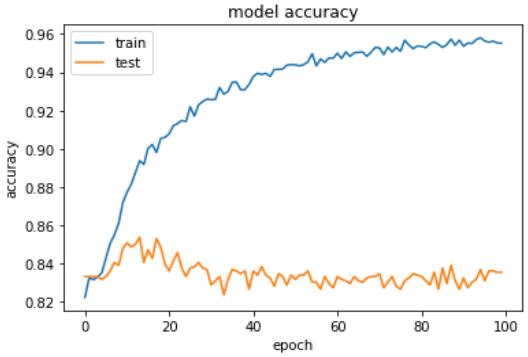
Figure 42Vanilla 3d loss and accuracy graph for malignancy at learning rate 0.0003

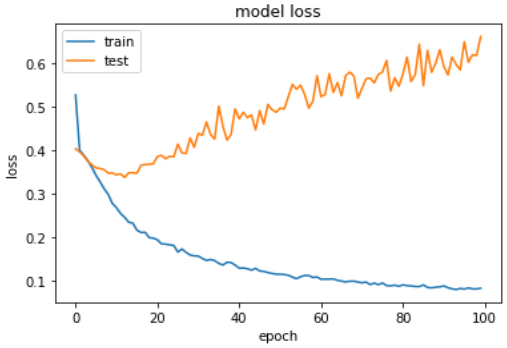
As you can see above the accuracy graph is near to horizontal and the model is not learning much. It is jumping around the global optimum and it is not going down. This shows that this learning rate is too huge for this problem.

6.6.3 Lenet

In this architecture three different learning rates were tested, 0.0001, 0.0003 and 0.001. This model got the highest result in this problem, reaching an accuracy of 96.5% at training accuracy. The best learning rate was 0.0001. It also used the Adam optimizer and a dropout of 0.2 after each convolution layer.

Learning rate 0.0003





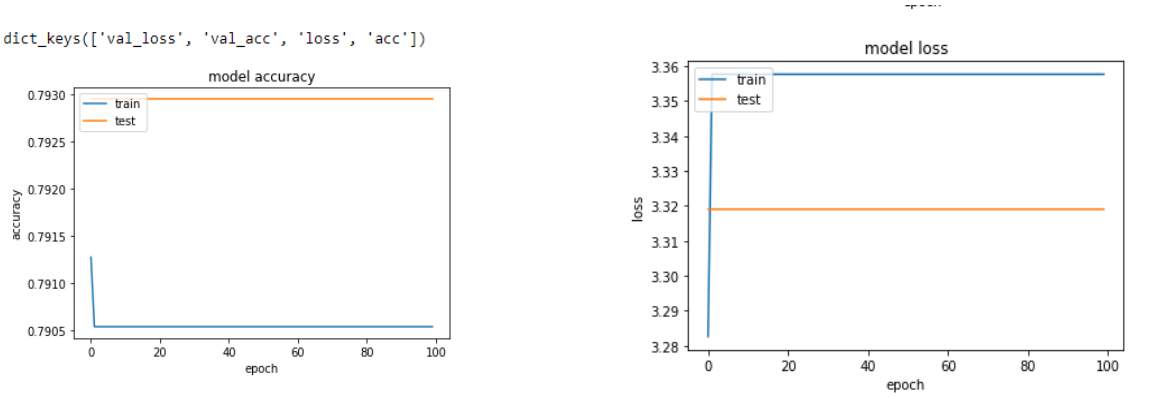
Accuracy graph

Loss graph

Figure 43Vanilla 3d loss and accuracy graph for malignancy at learning rate 0.0003

This accuracy got pretty good results for the training set but the validation was lower the training by a noticeable amount, so the model here did not generalize well. This will not do well if implemented in an application. So other learning rates were able to do this without training it with more epochs thus, saving time.

Learning rate 0.001



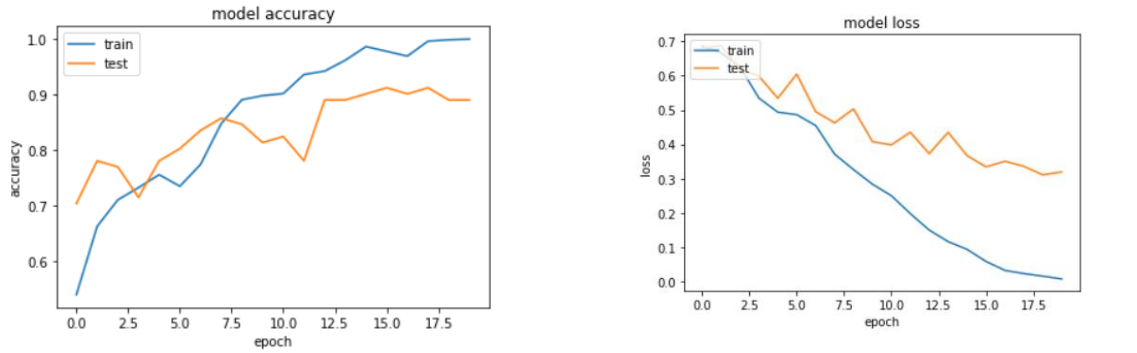
Accuracy graph

Loss graph

Figure 44Vanilla 3d loss and accuracy graph for malignancy at learning rate 0.0001

This learning rate was very large so the model start jumping of and missing the global optimum thus not learning at all and thus a flat line is presented above. This shows that this learning rate is very high for this problem and had to be decreased so that the model can learn something and decrease the loss.

Learning rate 0.0001



Accuracy graph

Loss graph

Figure 45 lenet in malignancy classification at learning rate 0.0001

Now This learning accuracy got the best results and it generalized nicely getting good accuracies in both the training and validation data, as you can see here it plateaued off in the 90s accuracies and the difference between the training and validation.

```
from sklearn.metrics import confusion_matrix
predicted = model1.predict(xtrain[:])
pre = np.argmax(predicted,axis=1)
tru = np.argmax(ytrain[:],axis=1)
confusion_matrix(tru, pre)
```

```
array([[198,  10,   9,   2,   2,   0],
       [ 17, 241,  45,  15,   4,   0],
       [ 16,  28, 737,  45,  17,   0],
       [  7,  11,  67, 435,  12,   0],
       [  1,   5,  15,  10, 314,   0],
       [  0,   0,   0,   0,   0,   1]])
```

Figure 46Vanilla 3d loss and accuracy graph for malignancy at learning rate 0.001

This is the confusion matrix of the 5 classes of the second model. As you can see here the diagonal have the biggest number and the diagonal is the true classes and the rest is the false classes, so this backs up the accuracies shown above.

6.7 Unit testing

6.7.1 Reading CT-scans and meta data

This function was simple. After the CT-scan matplotlib was used to display the scan and it displayed it correctly. Then I opened the scan using CT-scan viewer to see the meta data and compare with the ones read by python and they were the same.

6.7.2 Changing world coordinates to voxel coordinates

After this function was implemented. Check that no value is in negative as voxel coordinates have no negative values only world coordinates have those. Then the csv file was to annotate a big nodule in diameter and then display the position of it. To make sure the coordinates are write and the nodule can be scene.

6.7.3 Normalizing scans

This function should only return a scan that have values between 0 and 1. No negative values and no values greater than 1. So after this function was called a slice was displayed at random using numpy array and indeed all values were between zero and one.

6.7.4 Cropping patches

All patches should be 64X64X64 no bigger and no smaller because the model can give an error saying that this is not the shape required and not run. After this function

was run through a whole scan. The crops were put in a list. Then iterate over the list and display the shape of each crop.

6.7.5 Nodule classification

A set called validation set is used to determine the accuracy of the nodule classification model on scans it has never seen before. The validation set was about 1000 crops. The model predicted 1000 samples and we compare the models output with the real output and find that it 98% similar. This accuracy is called the validation accuracy, when you make the network predict the output of samples it was not trained on before and you have the desired outputs for this sample. This is a very good way to see if your model is doing well on data other than it is training on or not.

6.7.6 Malignancy classification

Also here a validation set is used to determine the accuracy of the model on untrained samples. It got 92% accuracy this is slightly less the training accuracies but this decrease is expected as long as there is not a huge gap between the two it is fine. The model was working fine on untrained data.

Chapter 7 Conclusion and future work

7.1 Conclusion

Deep learning was able to diagnose lung cancer one year before any other doctor by detecting features that are hard to be detected by the human eye. Although the classification problem was hard to solve because of the needle in a haystack problem, the problem was solved by dividing the classification into two parts and dividing the CT-scan into crops. Very good accuracies were reached in both models. In the nodule classification the best accuracies were reached by a model called Googlent and it reached up to 97% accurate. The malignancy classification also performed well. The best accuracies were reached by a model called Lenet and it reached 96% accurate. At the end a simple GUI was made to be able to use those models to classify unknown scans.

This project will help early diagnosis of lung cancer. It will reduce the number of false biopsies taken. It will make doctors confident of their decision and it will catch nodules that are not caught by doctors at early stages. This will make patients get treatment 1 year earlier than the catching it with the human eye. And when patients get diagnosed early this will drastically improve their chances of survival and making treatment a lot less vigorous. The program also gives you each nodule it has detected with the corresponding malignancy so the doctor can see and examine it giving the best medication for that patient. If this project can be made to scale it can save the lives of thousands each year and an annual checkup for people will make it very easy for radiologist to actually treat the patient and give him or her direct treatment. This will help reduce the fear of getting cancer and it can be just as simple as a few month of treatment.

In this project I gained many skills and learned new things. I learned machine learning from Coursera's machine learning course. That was an online course teaching you many machine learning algorithms like linear regression, neural network, SVM and much more with the math behind every single one. In this course I implemented every single algorithm from scratch using matlab understanding it fully. Then I took Coursera's Deep learning specialization, this is a five course program that helps you learn how to use tensorflow and many deep learning techniques and parameters. I learned how to use python, tensorflow and keras and learned the many parameters that can be tuned in a deep learning model like the learning rate the dropouts and even the architecture itself. Also self-discipline and time management are very important gained skills that without it I would have not completed the project. Another important skill I gained is reading articles and papers. Dr.Ahmed Farouq really was the one and only source that taught me how to read papers and how to get the information you need from a paper. This project is one of the hardest deep learning/ computer vision problems that one can tackle, so to be able to tackle this problem effectively gave me confidence about my machine learning that I gained. This project had many struggles as you have seen in the previous chapters and dealing with them

and solving them gave me hands n experience with working with models and gave me experience working under pressure.

7.2 Future work

Locating the nodules using segmentation or image localization will save a lot of time while classifying unknown scans. Now scans can take up to five minutes running to get a classification in the GUI. But this time can drastically decrease if we do not run the nodule classification model for each crop. An average scan can have about 128 crops so running forward propagation 128 times is time consuming. If the U-net worked it would only forward propagate once and segment all nodules in just one shot. And having the ability to segment out the nodule narrows the aria of search that the nodule might be in. In this project the area of search is 64X64X64 crops, but in segmentation will narrow down the area of search and ideally the area of search will be just the nodule.

If the nodules were segmented correctly we can combine the segmented nodule into a single crop and get the malignancy of the whole scan with just one forward propagation. Also this will decrease run time in the GUI it might help improve the accuracy.

References

1. browniee, J. (2016, March 16). *Want help with algorithms? Take the FREE Mini-Course.* Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/
2. Brownlee, J. (2016, August 16). *What is Deep Learning?* Retrieved from machinelearningmastery.com: https://machinelearningmastery.com/what-is-deep-learning/
3. Castle, N. (2017, 7 13). *Supervised vs. Unsupervised Machine Learning*. Retrieved from datascience.com: https://www.datascience.com/blog/supervised-and-unsupervised-machine-learning-algorithms
4. Charles D. Fenimore, S. A. (2011, january 28). *The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI)*. Retrieved from nist.gov: https://www.nist.gov/publications/lung-image-database-consortium-lidc-and-image-database-resource-initiative-idri
5. *Data Science Bowl 2017*. (n.d.). Retrieved from Kaggle: http://www.kaggle.com/c/data-science-bowl-2017
6. Dormehl, L. (2018, 5 11). *What is an artificial neural network?* Retrieved from digitaltrends.com: https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/
7. Jay W. Marks, M. (n.d.). *Lung Cancer*. Retrieved from emedicinehealth.com: https://www.emedicinehealth.com/lung_cancer/article_em.htm
8. Lawrence M. Davis. (n.d.). *CT Scan (CAT Scan, Computerized Axial Tomography)*. Retrieved from emedicinehealth.com: https://www.emedicinehealth.com/ct_scan/article_em.htm
9. *Lung Cancer Data Set* . (n.d.). Retrieved from UCI: https://archive.ics.uci.edu/ml/datasets/lung+cancer
10. *Lung nodule analysis 2016*. (n.d.). Retrieved from LUNA16: https://luna16.grand-challenge.org
11. Mayo clinic. (n.d.). *Lung Cancer*. Retrieved from mayoclinic.org: https://www.mayoclinic.org/diseases-conditions/lung-cancer/symptoms-causes/syc-20374620
12. *Nodulee Pipeline – Weights and Biases – Medium*. (2017, July 16). Retrieved from Medium: https://medium.com/weightsandbiases/nodulee-pipeline-bd3b6a48842b
13. S.Hawkins. (n.d.). Predicting malignat nodules from screening CT scnas. *Journal of Thoracic Oncology* .
14. Team, M. E. (2016, january 5). *Lung Cancer: Facts, Types and Causes*. Retrieved from medicalnewstoday.com: https://www.medicalnewstoday.com/info/lung-cancer
15. *What Is Lung Cancer?* (n.d.). Retrieved from lungcancer.org: https://www.lungcancer.org/find_information/publications/163-lung_cancer_101/265-what_is_lung_cancer
16. Zawadzki, J. (2018, february 7). *Convolutional Neural Networks For All*. Retrieved from medium.com: https://medium.com/machine-learning-world/convolutional-neural-networks-for-all-part-ii-b4cb41d424fd