Winter 11-2019

AUTOMATIC LUNG CANCER DETECTION USING ARTIFICIAL INTELLIGENCE

Bardh Rushiti

Program for Mechatronics and Management

AUTOMATIC LUNG CANCER DETECTION USING ARTIFICIAL INTELLIGENCE

Bachelor Thesis

Bardh Rushiti

November / 2019
Prishtina

Program for Mechatronics and Management

Bachelor Thesis
Academic Year 2016 – 2019

Bardh Rushiti

AUTOMATIC LUNG CANCER DETECTION USING ARTIFICIAL INTELLIGENCE

Mentor: Dr. Sc. Bertan Karahoda

November / 2019

This paper has been compiled and submitted to meet the partial requirements for the Bachelor Degree

ABSTRACT

By far, lung cancer is the prominent cause of cancer deaths for both men and women around the world. In 2018, statistics for WCRF (Worldwide Cancer Research Fund) showed that out of 2.09 million people diagnosed with this disease, 1.76 million people died. The survival rate increases if detected in its earlier stages. Taking into consideration the complexity of the problem, many computer-aided diagnosis systems that increase the survival rate have been proposed and developed. Driven by the notable success of deep learning in the area of complex image classification problems, this paper presents the use of VGG16, VGG19, ResNet34, and ResNet50 convolutional neural network architectures or classifying images of patients with cancer. Moreover, to compare the performance evaluation Accuracy, Precision, Area Under Curve, and F1 score were calculated. In conclusion, ResNet50 architecture exhibited the best result for this classification problem, with 95.83 Precision, 88.89% Accuracy, and 88.46% F1 score. The strategy of using pre-trained deep learning models proved to be pertinent to this problem.

ACKNOWLEDGEMENTS

First, I am grateful to my *Artificial Intelligence and Computer Vision* subject professor and mentor Dr. Bertan Karahoda for the support, constant guidance during the design and development process of my Bachelor Thesis. He has been an engineering role model throughout my whole studies, and having him as a mentor was the right way to finish my Bachelor studies.

Second of all, I would like to thank all the professors in the Mechatronics Department and more for being there for my countless questions I had in the past years and will continue to have in the future.

Most importantly, I would like to thank my family for their constant support, in every possible way, during all of my educational journey and life. I am forever thankful to them.

Novermber, 2019,
Prishtine

CONTENTS

LIST OF ABREVIATIONS

SCLC - Small Cell Lung Cancer
NSCLC - Non-Small Cell Lung Cancer
ML - Machine Learning
SVM - Support Vector Machine
SVR - Support Vector Regression
ANN - Artificial Neural Network
MLP - Multi-Layer Perceptron
MSE - Mean Squared Error
CNN - Convolutional Neural Network
ReLU - Rectified Linear Unit
ResNet - Residual Neural Network

LIST OF FIGURES

1. INTRODUCTION

In this section, there will be a small introduction to the main concepts and purpose of this thesis. In other terms, there will be described how the importance and the reason for this research.

By far, lung cancer is the prominent cause of cancer deaths for both men and women around the world. In 2018, statistics from WCRF (Worldwide Cancer Research Fund) showed that of 2.09 million people are diagnosed with this disease (21.7% of all cancer diagnoses), 1.76 million people have died (36.5% of all cancer deaths) [1]. The survival rate increases if the cancer is detected in its early stages; however, approximately 80% of patients are accurately diagnosed during the intermediate or advanced stage of cancer [2]. Hence, the survival rate so low.

In the process of diagnosing potential cancerous lung nodules, the radiologists have two interconnected tasks: to detect an abnormality and categorize it as representing a specific type of disease. Diagnosticating is an arduous process, but pulmonary radiologists have a high degree of accuracy in diagnosis. In a study, the accuracy rate of the radiologists in the detection of lung cancer using CT scans revealed around 0.79%. Nonetheless, there remain problems in disease detection. Some of these problems consist of the miss rate for the detection of small pulmonary nodules, the detection of minimal interstitial lung disease, and the detection of changes in pre-existing interstitial lung disease. These problems are hard to overcome, even with high levels of clinical skills and experience.

For years CADx (Computer-Aided Diagnosis) systems have helped radiologists for early diagnose and increase of care services earlier, faster, and with higher accuracy. Traditionally, CADx systems identify tumors and different diseases of similar nature with complicated image processing techniques and segmentation methods. Consequently, making this process of extracting low-level features strenuous and intricating. However, in the recent research literature, machine learning and deep learning techniques (a subset of artificial intelligence) have been used to diagnose and classify cancer — [4, 5, 6]. Due to the nature of these methods, extracting low-level to high-level features from large amounts of datasets and classifying cancer of different types has successfully proven to be somewhat accurate [7].

In this paper, by taking into consideration machine learning and deep learning techniques as some of the state-of-the-art methods in terms of automatic feature extraction and automatic detection, different algorithms and architectures of these techniques, specifically convolutional neural network, were investigated.

2. LITERATURE REVIEW

In this section there will be a small introduction about the main concept and purpose of this thesis. In other terms, there will be described how the importance and the reason of this research.

Lung Cancer

First, cancer is a group of abnormal cells that grows until they spread into neighboring tissues. These cells can grow almost everywhere in the body. The human body is made out of tens of trillions of cells. These cells, to survive and spread their genes, they divide into new cells. So when old cells get old, they die, and new ones take their place. However, with cancer cells, this process of cell reproduction stage does occur. They are abnormal; in the sense that those cells contain errors or mutations in their genes. The old abnormal cells survive when they should have died, and new ones are created even though not needed, and this is how tumors grow.

Tumors can be malignant or benign. The difference between the two is that malignant tumors advance in nearby tissue, making it possible for cancer to travel, through blood or lymph system to distant organs, and cause harm far from the original location (organ/ tissue). On the other hand, benign tumors do not advance in other organs or tissues. Even though they can get big, once removed, they do not possess any harm to the patient (unless the location of the tumor is in the brain) [8].

Lungs are two cone-shaped breathing organs found in the chest. The lung's primary role is to bring oxygen in and release carbon dioxide when breathing out. Based on the terminology explained above, lung cancer is the rapid expansion of abnormal cells in one or both lungs. These cells are not specialized to do a specific function, as healthy cells. Instead, they cluster into a tumor and interfere with the lung's sole task as a part of the respiratory system. Based on the size of the cell in which cancer starts, there exist two types of lung cancer: small cell lung cancer (SCLC) and nonsmall cell lung cancer (NSCLC).

Moreover, NSCLC is going to be the topic of exploration for this thesis, and it consists of three groups that are identified based on the initial location of cancer. First, the most common type of lung cancer that emerges in the glandular cells on the outer part of the lung is called adenocarcinoma. NSCLC can also start "in flat, thin cells called squamous cells. These cells line the bronchi, which are the large airways that branch off from the windpipe (trachea) into the lungs, as it can be seen in the picture below. This type of cancer is called squamous cell carcinoma of the lung. Large cell carcinoma is another type of nonsmall cell lung cancer, but it is less common. There are also several rare types of nonsmall cell lung cancer. These include sarcoma and sarcomatoid carcinoma."[9] Below, it is presented the most common type of cancer, adenocarcinoma.

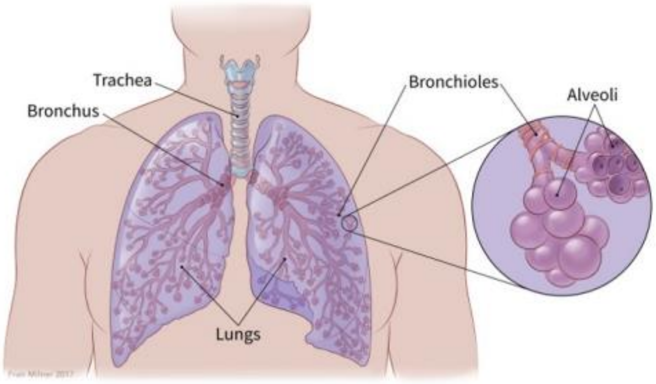


Figure 1 Non-small lung cancer - *Adenocarcinoma*

SCLC often starts in cells in the center of the lungs. The main types of small cell lung cancer are "small cell carcinoma and combined small cell carcinoma (mixed tumor with squamous or glandular cells)" [9].

For both types of cancer, the likelihood to cause lung cancer is increased by smoking (including cigars, tobacco, pipes, both past, and present). All studies and official websites point out the fact that smoking is the dominant risk factor for this disease. The longer the period of smoking, the earlier a person starts smoking, the more often the person smokes, the higher the risk. Additionally, other risk factors include exposure to secondhand smoke, asbestos, arsenic, chromium, beryllium, nickel, soot, tar, radiation, living in polluted air, having a family history of lung cancer, being infected by human immunodeficiency virus (H.I.V.), and many more. [10].

Machine Learning

Machine learning is the science of programming computers to detect patterns in data and use the knowledge gained to make predictions and or conditional decisions. "[Machine learning is the] field of study that gives computers the ability to learn without being explicitly programmed." the definition of machine learning by Arthur Samuel.

Nowadays, data is being generated like never before, not only by people but also by mobile phones, computers, cars, and other devices. Up to the year 2005, humans have generated around 130 Exabytes of data; this includes pictures, videos, songs, books, everything we have created. These numbers are increasing exponentially every year. Started from 1,200 exabytes in 2010, this number is estimated to reach 40,900 exabytes by 2020. [11] With enough accessible data and available computing power, this is the perfect time for this science to flourish. Below, it is presented the graph showing this exponential growth of the rate of data production.

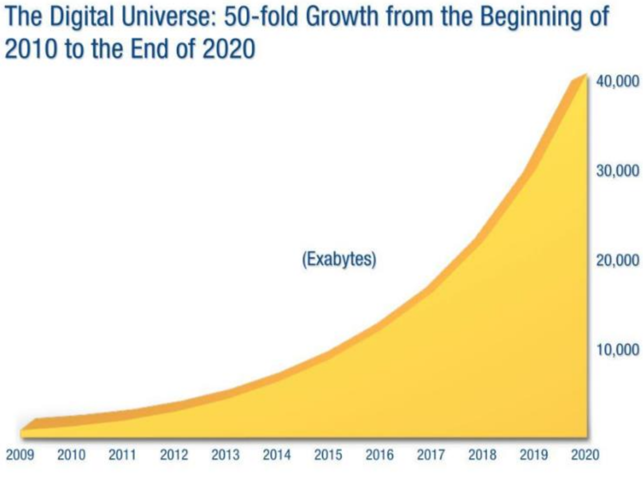


Figure 2 The growth of data created from 2010 to 2020

Machine learning, as a subfield of artificial intelligence, is well known for solving various complex problems. Some areas where this science thrives are Predictions - With large amounts of data, these techniques are suitable for detecting patterns and then making predictions on future new data. For example, in predictive maintenance, companies use ML and old data from their sensory of machines in the production line to predict the upcoming machine failures. Image recognition - ML is used to detect faces and saves them in different categories for each person. For example, the "tag recommendation" system on Facebook. Speech recognition - Word to text converters, text to word converters, data entry, and many different applications using only voice. These techniques are widely applied in Google Translate, and Caption in YouTube videos. Medical diagnoses - Machine learning techniques are used to detect and diagnose cancerous tissue. The financial industry and trading - many banks use ML in fraud detection and credit check. As can be seen from above, the variety of problems these techniques can solve is infinite. [12]

Types of Machine Learning

Based on how much supervision they get during training, machine learning algorithms divide into three main categories: supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, the data inputted into the algorithm requires the desired solution as well, called labels. The idea is to try to approximate the function so well that new data, the function can predict the desired outputs. The spam filter is a great example to show this. It is presented in Figure 2.2 (A). Two typical supervised learning tasks are classification and regression. As the name suggests, classification tasks classify the output into known labels (spam/not spam, cat/dog, cancer/no cancer). On the other hand, regression type problems output constant value, such as price or weight. Figure 2.2 (B) presents this example. [12]

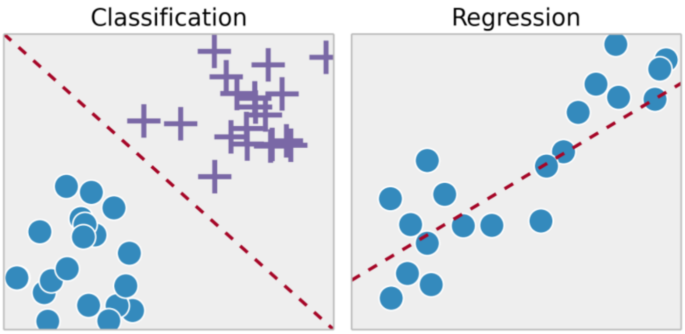


Figure 3 Classification (A) and regression (B) type problems in supervised learning

In unsupervised learning, the data is uncategorized and unstructured. In other words, the algorithm tries to "learn without a teacher" [insert citation here - Hands on ML]. Main unsupervised learning tasks are clustering, pattern search, and dimensionality reduction. Clustering problems include grouping data with similar characteristics. An example to illustrate this would be grouping customers by purchasing behavior. Pattern search, or association, algorithms try to discover rules, patterns or make associations for some given data. For example, in shopping data, the algorithm if people item X, they also tend to buy item Y. In dimensionality reduction, the idea is "to simplify the data without losing too much information" [12]

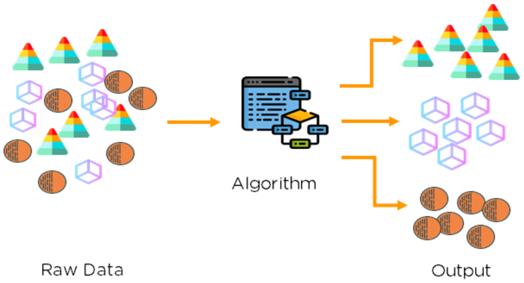


Figure 4 Unsupervised learning

In reinforcement learning, agents are operating in an environment. They analyze the environment, make a decision on the next action, and receive a reward based on it (positive if the decision has brought it closer to the goal, and contrary if it has not). Then the agent learns his strategy to act in order to bring the most favorable rewards over time, and this is called policy. A typical example of Reinforcement learning is shown below.

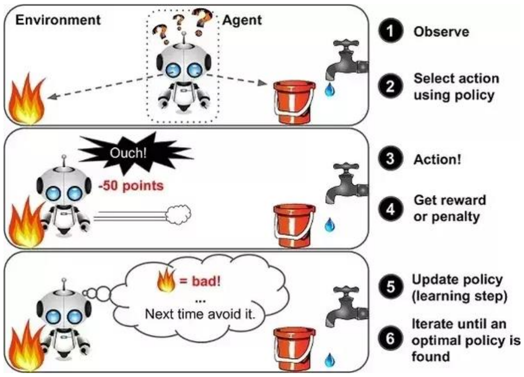

Environment
Agent
1 Observe
2 Select action using policy
Ouch!
-50 points
3 Action!
4 Get reward or penalty
= bad!
...
Next time avoid it.
5 Update policy (learning step)
6 Iterate until an optimal policy is found

Figure 5 Reinforcement Learning

Machine Learning Algorithms

There is a large variety of Machine Learning algorithms, and they are used for even a wider variety of problems. Below we will discuss some of the leading models used in the research and production world of data science.

Naive Bayes is a probabilistic approach to determine the class of the new input data. This classification algorithm is based on Bayes Theorem. Even though it is relatively simple, it performs well in particular areas. This algorithm requires that the features of the data are independent (so the correlation between variables should be zero) before making a classification.

Support Vector Machine algorithm relies on the idea of finding the maximum margin line that minimizes the error of classification among classes. It is a powerful and versatile algorithm, which efficiently performs linear, nonlinear classifications, regressions, and outlier detection. The reason these algorithms perform so good is that they take extreme cases (what might be confusing for computers to identify) of classes, which are close to the margin line, and it uses that to construct its analysis.

CART stand for classification and regression tree is a term that encompasses two types of decision trees: classification and regression trees. Concerning classification, decision trees can be seen as algorithmic trees, which try to separate the dataset based on different decisions (rules). This separation is tried to be as pure (most if not all the nodes datapoint only from one class) as possible in order for the decision to be more accurate. The tree on its own is a weak and straightforward model, but when combined, they create robust and powerful models called Random Forest. The combination of weaker algorithms into more powerful ones is generally known as ensemble learning [13].

Regarding regression, linear and logistic regression is commonly used due to their simplicity and low costs for implementation. Linear regression estimates the variables that present the best line for the provided data.

SVM supports linear and nonlinear regression, referred to as SVR (Support Vector Machine). SVR instead of trying to maximize the margin between two classes, it tries to fit the most significant number of points between that margin. The size of the margin is determined by the hyperparameter Epsilon. This algorithm has shown excellent performance in the areas of weather forecasting and financial data.

Artificial neural networks are a robust and classification algorithm. By imitating how the brain works, ANN has shown impressive results many times over in the research community. It is well known for solving complex problems from various fields.

Features are the input data that are fed into the machine learning algorithms; therefore, making features suitable for these algorithms consumes most of the data scientist's time because they, more than anything else, influence the result of the algorithm. Feature engineering is done with two goals in mind, set up the data appropriately for the algorithm requirements, and increase the accuracy of the ML models. Numerous area specialists and data scientists look to find and create high-quality feature subsequent applying various evaluation methods, statistical analysis, and performance tests of models. [14]

Deep Learning

Deep learning is a branch of machine learning modeled after the signal transmission in neurons and synapses of the brain. The elementary unit of a neural network is a perceptron, invented by Frank Rosenblatt in 1957, can be visualized as a simple linear classifier. A model of a single perceptron is shown in figure 5.

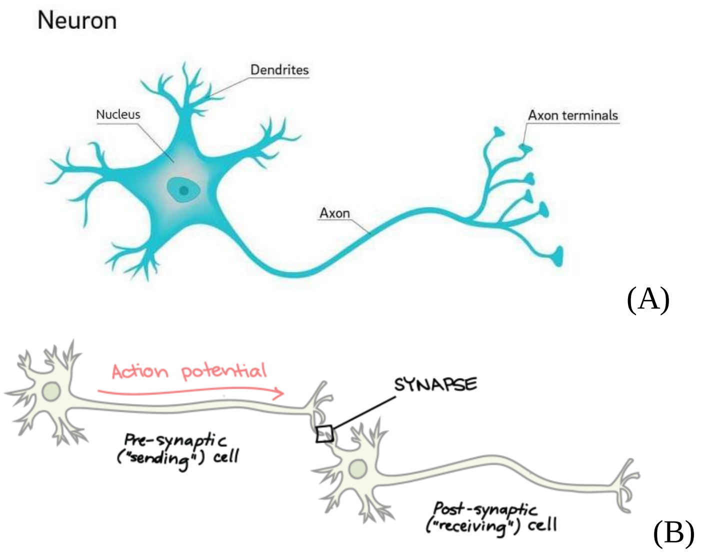


Figure 6 (A) Neuron and its elements (B) Synapse between neurons

Biologically, neuron transmits nerve impulses to other cells depending on the strength of the action potential. The neuron usually contains a nucleus (cell body), dendrites, axon, and axon terminal. In engineering terms, dendrites are like the receiver of the signal; the nucleus is where the processing happens; axon and axon terminal is like the transmitters of the signal to nearby neurons. The connection between one neuron's axon and another's dendrites is called synapse. Most synapses are chemical connections and are used to send electrical impulses between neurons; this exchange of electrical impulses is known as an action potential.

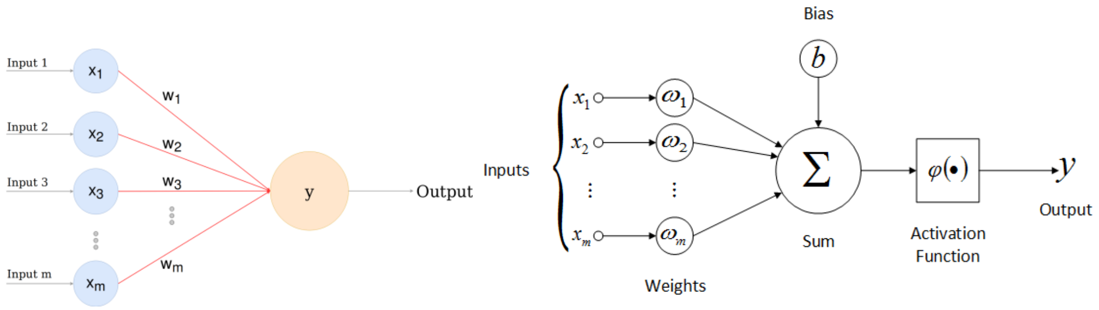


Figure 7 Mathematical model of the neuron

Above is presented the visual representation of the mathematical model of the perceptron. The first part of the neuron, x1 to xm, is called the input layer. This layer is where the input signal or input data from other neurons come in. The last part is called the output layer, and as the name suggests, it calculates the output of the neuron. In the middle, there is a nucleus where all the data is processed. In the second picture, we can see a more detailed model, which includes bias b, and the activation function φ. The bias serves as a threshold for neuron activation; the activation function redistributes the function in another more suitable form for perceptron. Before inputting to the neuron, values in the activation function can vary. In order to have values, for example, ranging from 0 to 1, we put the activation function, which does that for us. To summarize, the neuron calculates a "weighted sum" of its inputs, compares it to a threshold b, and then transforms the output into a specific range. [15]. Below, we can see the mathematical formulation of the perceptron in a compact way (3).

$$z = (\mathrm{w}_1 \times \mathrm{x}_1 + \mathrm{w}_2 \times \mathrm{x}_2 + \mathrm{w}_3 \times \mathrm{x}_3 + \ldots + \mathrm{w}_n \times \mathrm{x}_n) + b \qquad (1)$$

$$\hat{y} = \varphi(z) \qquad (2)$$

or

$$\hat{y} = \varphi\left(\textstyle\sum_{i=1}^{n} \mathrm{w}_i \times \mathrm{x}_i + b\right) \qquad (3)$$

At the time of the discovery, scientists found many flaws with the perceptron's ability to solve problems; it was limiting in many ways. However, Multi-Layer Perceptron (MLP), which is stacked perceptrons on top of each other, eliminates most of these problems. Similarly, the first layer of the MLP is the input layer, the last one is the output layer, and hidden layers are the layers in between. More complex networks are constructed from many perceptron units. If a single perceptron is a linear classifier, then MLP is a much more complex set of linear classifiers, where each unit aids in the overall process of correctly classifying an input [12, 16]. Below it is presented an MLP neural network with an input layer(containing eight units), three hidden layers (each containing nine units), and an output layer.

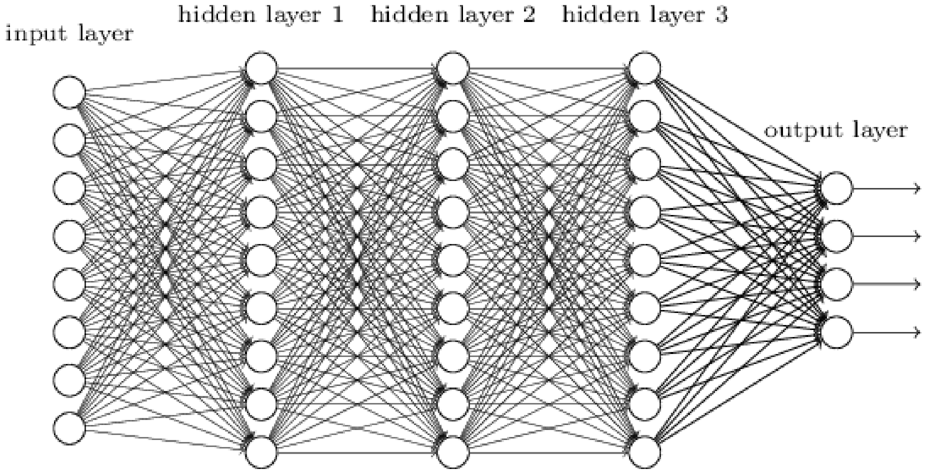


Figure 8 A Multi-Layer Perceptron (MLP) Neural Network

As we have mentioned earlier, the learnable variables (the variables that change to adapt to the problem) of the network are weights ω and biases b. Each unit contains a bias, and each connection contains a weight, so in the example of Figure 7, 8 * 9 (first and second layer connections) + 9*9 (second and third layer connections) + 9*9 (third and fourth layer connections) and 9*4 (fourth and fifth layer connections) + 8+9+9+9+4 (the number of total units in the network) = 309 learnable parameters.

How do these networks learn? The network starts by initializing these variables, producing random outputs for our classes. After that, we use the Mean Squared Error (MSE) cost function C (4) that calculates how far are the actual values from the desired ones, creating a colossal column vector of all with the weights and biases; if this algorithm outputs a large number, then it means that the result of the network is far from our desired output.

$$C(w,b) \equiv \frac{1}{2*n} \sum_x \quad || \, y(x) - a|| \, ^2 \tag{4}$$

The quadratic cost function is a practical way to measure how the network classifies the inputs. Usually, nudging the weights and biases does not have any effect on the number of training images classifies correctly. Experience shows that the network ends up making the same errors [17]. Afterward, the weights and biases are gradually optimized according to the problem with the help of optimization algorithms as stochastic gradient descent (5) and (5.1). [17]

$$w_{ij}^{(k+1)} := w_{ij}^{(k+1)} - \eta \frac{\partial C}{\partial w_{ij}^{k}} \tag{5}$$

$$b_{i}^{(k+1)} := b_{i}^{(k+1)} - \eta \frac{\partial C}{\partial b_{i}^{k}} \tag{5.1}$$

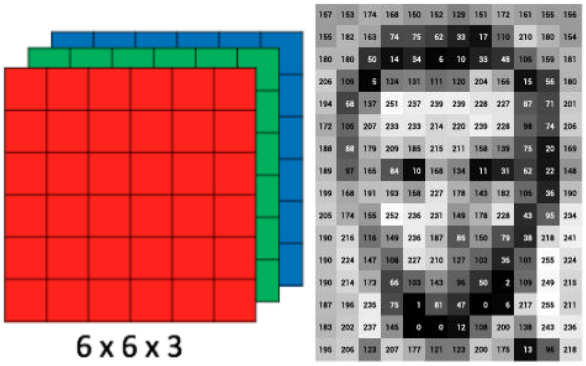
Taking into consideration that sometimes datasets contain very large amounts of data, and presenting this data to the gradient descent algorithm is time-consuming. Stochastic gradient descent only uses random data from the dataset in order to converge faster. This algorithm finds the way to the local optima with steps that are proportional to the negative steps of gradient function multiplied with the learning rate η. What it outputs is a large column vector of magnitudes of weights and biases. There is another algorithm that determines how much a single weights or biases contributes to the overall cost function in the whole network. This is done by calculating the partial derivatives of weights $\partial C/\partial w$ and biases $\partial C/\partial b$ for a single training example, and this process is repeated from the beginning; it is called backpropagation [19]. To summarize, neural networks learn by adjusting weights and biases to minimize a certain cost function.

When models get more complex, and architecture gets deeper, millions and millions of calculations are needed to fit the model; therefore powerful GPUs are needed to enable complex matrix operations to be finished in shorter amounts of time. These developments, both in advance models and in computing power have enabled improvements in many different research areas, as natural language processing, image classification, and voice recognition. Additionally, numerous open-source software libraries enable users to create such deep neural networks, like Keras, Tensorflow, Theano, Scikit-Learn, but not only.

Convolution Neural Network

Convolution neural networks (CNN) have presented an impressive performance in solving complex problems in various fields of computer vision, such as image recognition, face detection, natural language processing, and many more [6, 18, 19]. For all these fields, clear and abundant features are essential for the CNN model to work correctly.

For example, in classification tasks, CNN takes images as input and based on the training it classifies those in specific categories. Computers recognize coloured images as three-dimensional matrices of pixels where each dimension represents the percentage of red, green, and blue pixels. On the other hand, gray-scaled images as two-dimensional matrices of pixels, where pixels represent the intensity of the brightness of the pixel with values between 0 and 255.



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Figure 9 Colored image (three-dimensional) (left), grayscale image (two-dimensional) (right)

The CNN model goes through two phases in order to classify images correctly; in the first phase, the images are preprocessed to extract features for the algorithm to learn optimally, and in the second, those features are inputted in the fully connected neural network to make the classification. These two phases consist of four primary operations, which are the building blocks of every CNN: convolution (kernels, filters, feature map), non-linearity (ReLU), pooling (max pooling), classification (fully connected layer) as it can be seen below.

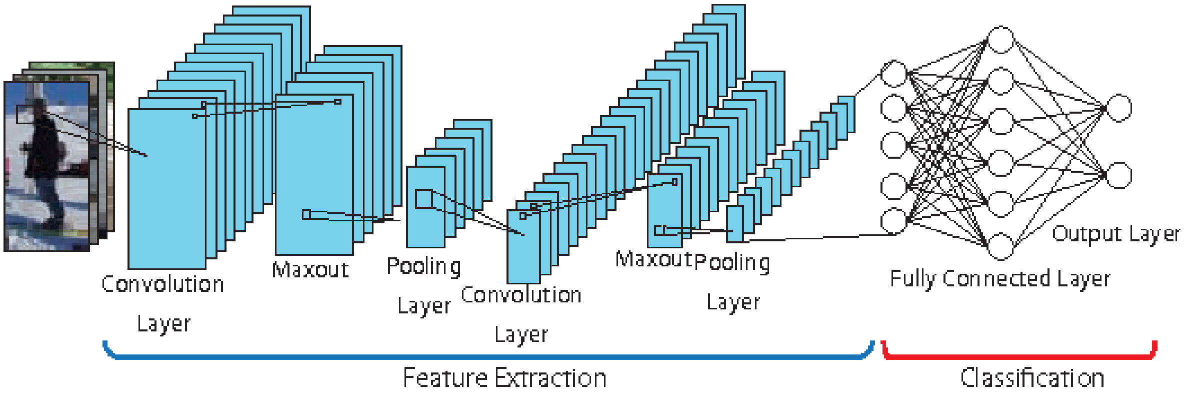
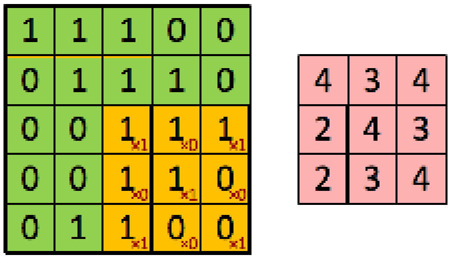


Figure 10 The two parts of convolutional neural network: feature extraction and classification

Convolutional operation is the first step in every convolutional neural network; its aim is to extract features from the image (known as feature maps). It preserves the spatial relationship among pixels from learning by feature maps. Let us take into consideration a five by five grayscale image (2-dimensional image, where each entry 0 corresponds to black, and 1 corresponds to white) and a three by three matrix called feature detector is shown above. The feature detector slides over the grayscale image and to find "features" that match the feature detector. Mathematically, elements of feature detector are multiplied with the elements of a grayscale image and then summed up to give one element of the feature map. This process repeats for all elements of the grayscale image.

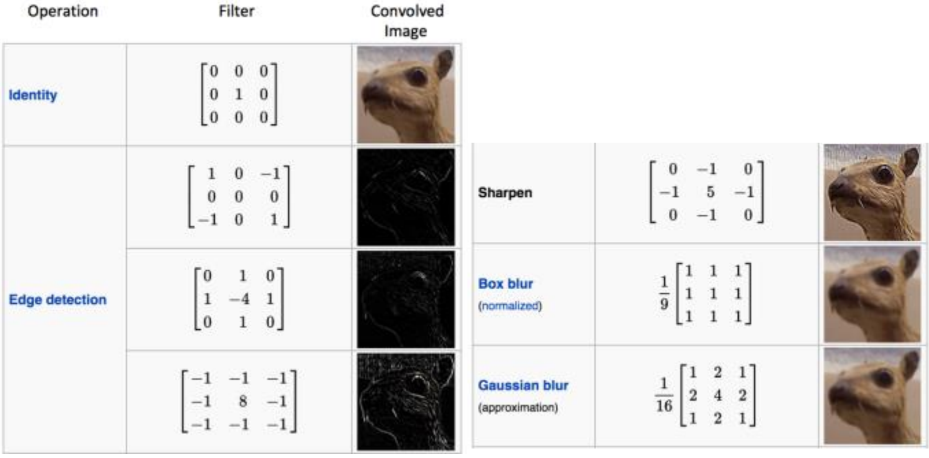


1	1	1	0	0
0	1	1	1	0
0	0	$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$
0	0	$1_{\times 0}$	$1_{\times 1}$	$0_{\times 0}$
0	1	$1_{\times 1}$	$0_{\times 0}$	$0_{\times 1}$

4	3	4
2	4	3
2	3	4

Figure 11 Convolution operation

Different filters (feature detectors) produce different feature maps, which can be seen in the image below. The more features the network is provided with, the more it learns the patterns in new images. Moreover, there are additional hyperparameters that influence the learning process, such as the number of filters, filter size, the architecture of the network, and many more.

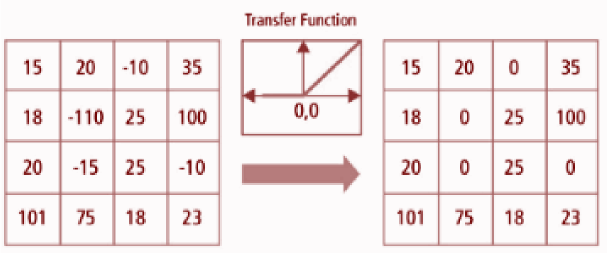


Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Figure 12 Image under different filter operations

Before the convolutional step, the ML engineer decides three hyperparameters that determine the size of these feature maps: depth, stride, and zero paddings. The depth of the feature map controls the number of feature maps we use for the convolutional step. The stride corresponds to the number of pixels the feature detector slides in our image matrix at a time. Zero-padding refers to the borders of the feature map. Sometimes the feature detector, depending on the stride, does not fit the input image flawlessly. If zero-padding is applied, the picture is pad with zeros; however, if it is not applied, it drops the parts the filter did not fit.

The second operation on CNN is non-linearity. The reason why this is an essential part of CNN models is that it introduces non-linearity into the model. Non-linearity means that no combination of inputs and multiplication with scalars can reproduce the output. In other words, without a nonlinear activation function, the model would produce the same result as a neural network without any hidden layers, so a linear classifier. ReLU, which stands for Rectified Linear Unit, has proven to be an efficient nonlinear activation function, due to its simplicity [19].



15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Figure 13 The effect of ReLU operation

Presented above are the rectified linear unit and its effect as an operation in a matrix of values. Mathematically, for all non-positive values of x, the output is zero, and for positive values of x, the output is the same. In a pixelated picture, this has a slightly different effect. In figure 13, white pixels have positive values, gray pixels have zero, and black pixels have negative values. After the ReLU operation, it broke all linearities in the picture, containing only the non-negative values.

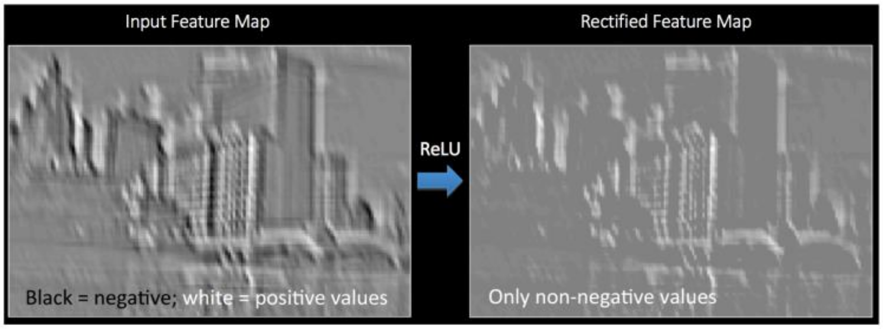


Figure 14 The effect of ReLU in picture

The pooling step (in literature, also known as subsampling or downsampling) aims to reduce the dimensionality of the feature map by different means. There are different types of pooling: Max Pooling, Average Pooling, Sum Pooling, depending on the operation, the pooling gets the name. For example, max-pooling preserves only the essential data (containing the highest values) after the rectified feature map. In the figure below, a two by two matrix with a stride of two slides into the feature map trying to capture the most crucial information.

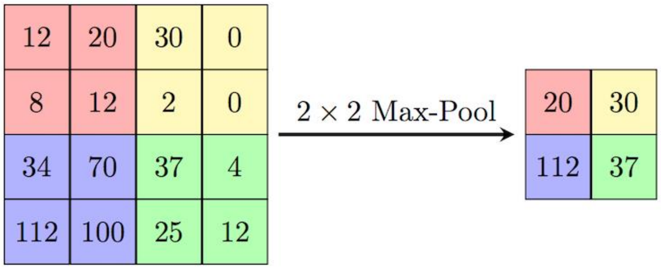


Figure 15 Max Pooling Operation

Pooling makes it easier for the network to manage large quantities of data. Additionally, by capturing the essential features from the input, it makes the network unconcerned from small variations, distortions, and translations in it. Consequently, it helps the network to converge to an equivariant representation of our input. This operation concludes the first phase of convolutional neural networks. In Figure 14, we can see the output of a convolutional operation, producing a four by four matrix of features. The max-pooling, which is a two by two matrix with a stride of two, goes through the features and out of the four elements it “sees” in every stride, it gets the most significant ones.

The second phase of the CNN algorithm is where the classification happens. In the next step, the matrix containing the most prominent features is flattened into vector form and used as input for the fully connected neural network, which is a standard multi-layer perceptron. In the problems with more than one class, a softmax function is usually applied. In mathematics, the softmax function normalizes the input and transforms it into non-negative numbers that add to 1, so it can also be represented as probabilities.

3. PROBLEM STATEMENT

Throughout the research process, artificial intelligence, even though a relatively young invention in the world, covers a wide range of areas of the industry, starting from medicine, sports, manufacturing, and countless more. The scientific community has focused more on a particular field of artificial intelligence known as deep learning. It is known for its ability to solve very complex problems and many times it has come close and gone beyond human capabilities.

In recent years, this tool has gained a lot of attention, especially in the health sector. With initiatives such as LUNA Challenge in 2016 and Kaggle's Data Science Bowl 2017 all around the globe trying to make the work of radiologists and oncologists a bit easier, and make healthcare services easier and cheaper to access.

In Kosova too, hospitals are overwhelmed with people, making the life of doctors and patients harder. Moreover, the patients have to wait long periods of time in order to get a proper diagnosis, and Consequently, needing to go abroad to seek different diagnoses.

In dire need of an effective solution that reduces the radiologist's time in detecting, classifying cancer, and writing reports for patients, an application that could reduce the time the day to day time consuming activities doctors have to go through in diagnosing a malignant and benign cancer is not only helpful but also would save lives.

4. METHODOLOGY

This part of the thesis presents the research approach, strategy and the method of classification used to analyze and accomplish this thesis.

The approach used for this research is an explorative one. Because of the main purpose of this thesis is to classify lung cancer, a dataset was formed with secondary data of snapshots of CT scans of patients with cancer and without cancer from the Cancer Imaging Archive. The cancer cases were chosen only those with nodules sizes of at least 3mm. Before training, to all of the images were applied some preprocessing techniques to enhance the features in the snapshots.

For training, I utilized the power of deep learning tools, particularly convolutional neural networks with four different architectures were used, VGG-16, VGG-19, ResNet-34, ResNet-50. The training was done in Google Colab platform utilizing power of a free GPU, some online free libraries in Python.

5. RESULTS

In this chapter will be discussed about the results of the project. More precisely here will be described about the accuracy and outcome of classification, which is made from our input dataset.

The Dataset

The data used for the research is obtained as secondary data from the publicly available Lung Image Database (LIDC-IDRI) [21]. It contains diagnostic, and lung cancer screening thoracic computed tomography scans with marked-up annotated lesions. This dataset is considered the benchmark for lung cancer research and data science challenges in the machine learning community. To create this dataset, seven academic centers and eight medical imaging companies collaborated. Four experienced thoracic radiologists did the two-phase annotation process. In the initial blinded-read phase, each radiologist independently reviewed each CT scan and maker lesions belonging to one of three categories ("nodule > or =3 mm," "nodule <3 mm," and "non-nodule > or =3 mm"). In the subsequent unblinded-read phase, each radiologist independently reviewed their marks along with the anonymized marks of the three other radiologists to render a final opinion. For our purposes, the dataset and labels used were from the first initial phase of the process, to ensure that that the algorithm will capture the full range of cancer cases from patients. Additionally, due to a lack of resources, only snapshots of the CT scan were acquired. In other words, the classifier learned from images rather than from full CT scans. In cancer cases, the snapshot was deliberately chosen from the layer containing the nodule.

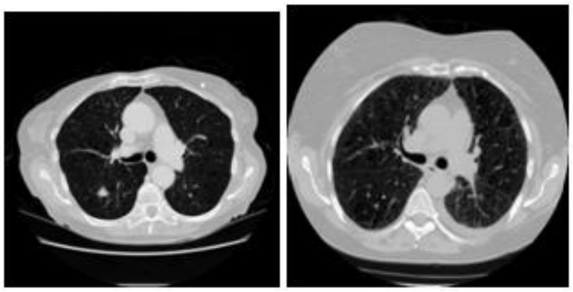


Figure 16 Two snapshots of lung screening (Left – Cancer; Right – No Cancer)

Above, we can see the algorithm's input data. On the left, we have a nodule on the left side of the CT snapshot, which is labeled as cancerous. On the right, we have an input that does not have a nodule inside the lung area, which has the label labeled as no cancerous.

The Algorithms and Training

For our particular problem, we used four different pre-trained CNN networks that were inside the fastai library: VGG-16, VGG-19, ResNet-34, and ResNet-50. These models were previously trained on ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, a dataset consisting of approximately 1.2 million training images. Moreover, these models are capable of recognizing 1000 different object categories, similar to things we come across in our daily lives with high accuracy.

The VGG architecture was first proposed in 2015 by Simonyan et al called "Very Deep Convolutional Networks for LSIR (Large-Scale Image Recognition)". In 2014, it won the 2nd place in the ILSVRC challenge for object detection, the 1st place for object localization, and the first model that achieved error rate under 10%. The total architecture of VGG model is simple; stacked convolution layer of on top of each other; but what sets this model apart is the small feature detector, which is 3 by 3. By contrast of AlexNet [19] and ZFNet [18] used larger feature detectors, 11 by 11 and 7 by 7, respectively. Using smaller feature detectors requires less parameter to be learned, consequently converging faster, and reducing the overall overfitting problem. Below, the number in VGG-16 and

VGG-19 refers to the number of layers that these architectures contain; hence, 16 layers and 19 layers, respectively.

Until 2016, evidence showed that the number of layers exhibited a strong correlation to the accuracy rate of models [18, 19, 22, 24]. However, as the number of layers increases, the accuracy satures and then the performance of the model degrades quickly[25], and overfitting is not the problem. As counterintuitive as it might sound, shallower networks were learning better. The question arises, why not just short circuit the extra layers in the network where the learning degrades? He et al. proposed this solution by adding identity mapping to network. Instead of hoping for the network to learn better with depth, he added skip connections (residual connections or identity shortcuts) to every few layers to increase the accuracy. This way, the network skips the layers that it does not learn anything from and only continues with layers which provide new features that increase its performance. Below it can be seen the difference between VGG-19, 34-layer plain, and 34-layer residual. Similarly, below ResNet-34 and ResNet-50, refer to the number of layers these architectures contain; hence, 34 layers and 50 layers, respectively.

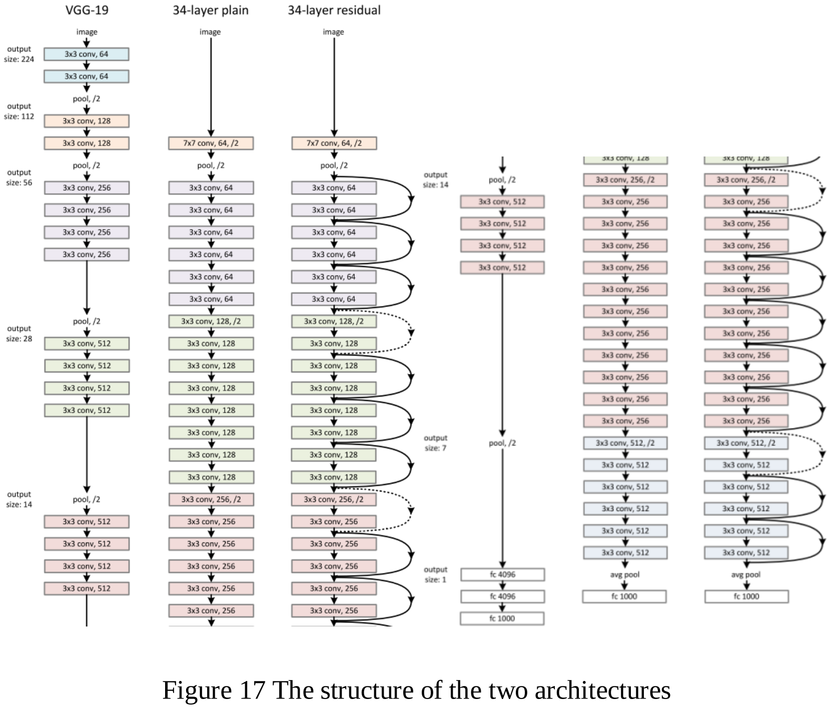


Figure 17 The structure of the two architectures

Two hundred stratified inputs are used to train the algorithm, of which 75% were used to train the model, and the remaining 25% were used to validate it. Fastai is the library that I used to create the pipeline for this algorithm, including cleaning up the data, creating the model, training it, and then validating it. Everything mentioned above was done in Google Colab platform, where the model was trained with one Tesla K80 having 2496 CUDA cores, compute 3.7 MHz, 12GB(11.439GB Usable) GDDR5 VRAM.

Transfer Learning

Having a small amount of data highly limits the algorithm's ability to learn to generalize the problem at hand, that is why I used transfer learning. This technique utilizes the power of a trained model on a very large dataset (in our case, ILSVRC dataset) [25] and then channel the new learning for a different computer vision problem. The basic principle behind transfer learning is that the model has already learned to recognize patterns on different images, and then you use that knowledge to solve new problems [26]. This technique is particularly helpful in small datasets, compared to starting from a randomly initialized model [27].

Moreover, transfer learning is implemented by only re-training the linear layers (known as the head) of the convolutional models. This way, convolutional layers, which extract features and analyze the images, are remained with the same pre-trained weights on ImageNet, and only the head is trained and initialized randomly. The approach used in this research paper is separated into two phases: in the first phase, the convolutional layers remain with the same weights, and only the head is retrained and in the second phase, the convolutional layers are unfreezed and trained gradually by fine-tuning the whole model's learning rate.

Why Python

In the flourishing machine learning community it has become an unwritten standard to use Python for developing artificial intelligence algorithms that solve complex problems. In the graph below the growth Python has had in the past five years in the field of machine learning.

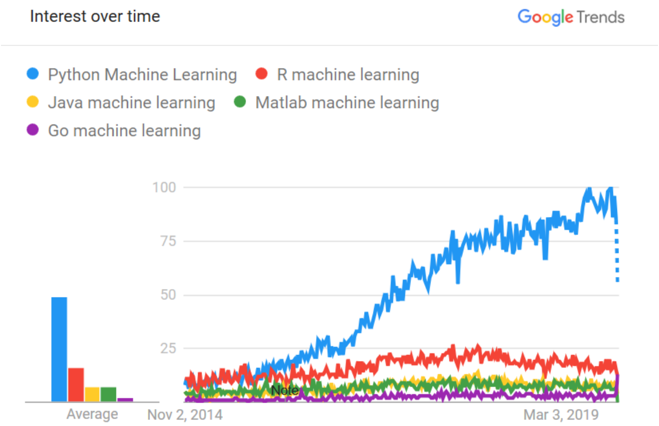


Figure 18 The growth of the five most common machine learning programming languages over the last five years

I used Python for various reasons. First of all, Python is easy to use. Its straightforward and readable syntax is what attracts most people. The abundance of libraries and frameworks that can be used for a variety of problems makes Python the go-to language for most ML developers. Some libraries that are beneficial especially for the artificial intelligence community are Keras is an open-source library that is helpful for experimenting with deep neural networks; TensorFlow is a free software useful for creating neural network applications; Scikit-learn is a software library that contains many classification, clustering, and regression algorithms related to machine learning; Fastai is a high level software library that makes the machine learning and deep learning process easy for newcomers.

Moreover, being in the market since 1990, Python has a well established a community and corporate support. Having a well-built community, new engineers can easily improve their knowledge on the topic, which consequently and simultaneously leads to a better community and machine learning engineers.Last but not least, Python allows operations in cross-platforms and cross-language operations which makes it very portable and extensible. Earlier we mentioned the steps GPU have had in the overall development of these algorithms, Python has proven to be very easy and well-suited for such tasks.

In this section, we are going to discuss the performance of the classifiers we have trained for the lung cancer problem. For the training part, 400 images were used in total, 75% of which were used for the training set, and the remaining 25% were used for validating it. The algorithms trained for this problem were VGG16, VGG19, ResNet34, and ResNet50.

The Experiment

The first phase of the training, was done only for the top layers of the model, without changing the convolutional layers deeper down in the model.

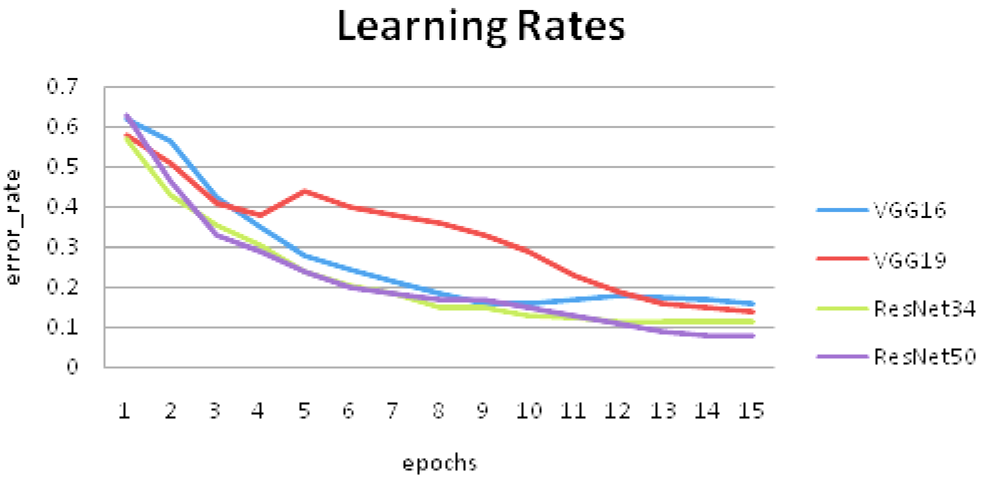


Figure 19 Learning rate of the algorithm expressed in error_rate per epoch

In Figure 18, it can be seen that out of the four pre-trained models, ResNet50 performed the best during training, reaching an all time low error rate of 0.07886. After ResNet50, ResNet34 scored the best with error rate of 0.11111, VGG19 with 0.14023 and VGG16 with 0.16059. The outputs for this performance we believe are the micro networks in residual architecture, since they don’t let the learning go astray. After the first phase of training, we used a learning rate optimizer in order to find the best learning rate for the algorithm for the problem at hand. This optimizer explores a wide range of learning rates and plots them to their loss, as below.

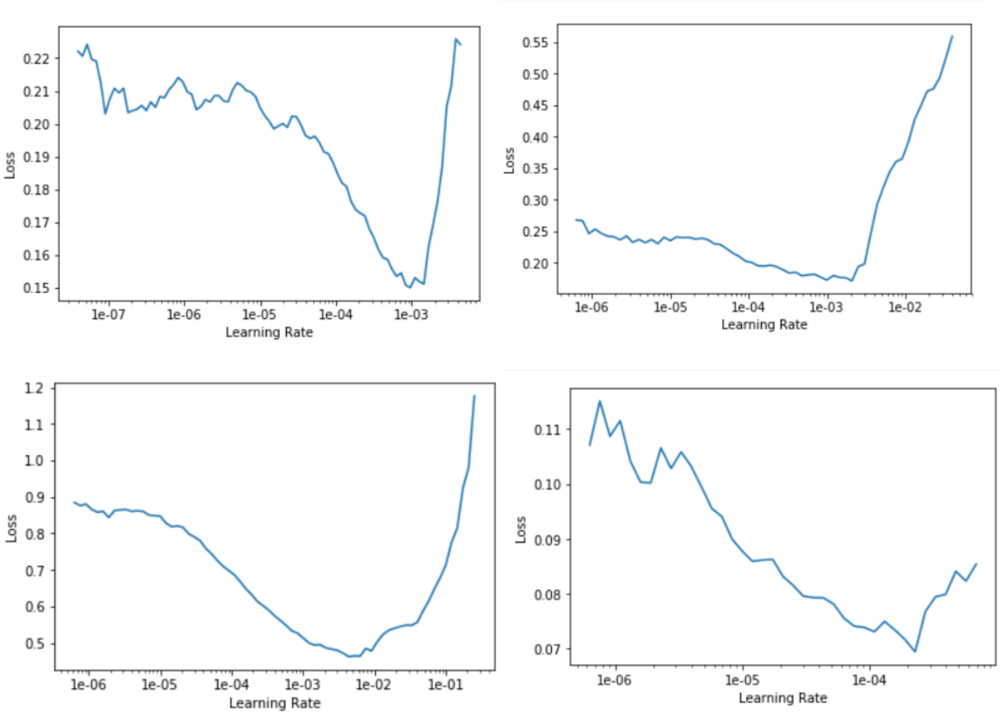


Figure 20 Learning rate optimizer used for all models (top left - VGG16, top right - VGG19, bottom left ResNet34, bottom right ResNet50)

After the recalibration of the network, the second phase of training begins. In this phase the convolutional layers containing the weights are unfreezed, and further training for 15 more epochs is done. Conclusively, their performances are evaluated on the validation set in terms of Sensitivity, Specificity, Precision, Accuracy and F1 Score. Each of these scores representations of the confusion matrix, which is a summary table of our algorithm performance.

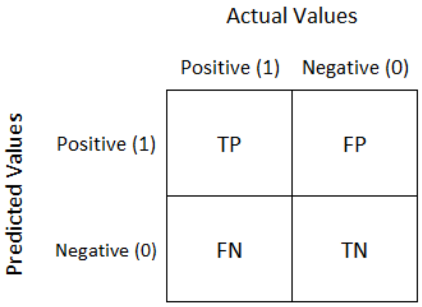


Figure 21 Confusion Matrix

Recall (Sensitivity) is the ratio of actual cancer cases from the total number of correct classifications. It is a statistical measure of the performance of a binary classification test (6). Precision shows the number of correct positives in all the classifications the algorithm has made (7). Accuracy is defined as the ratio of total correct classifications by the total classifications (8). F1 Score is a “harmonic mean of precision and recall” as Aurelien Geron put it in Hands on Machine Learning (9).

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$\mathrm{A} = \frac{TP + FN}{TP + TN + FN + FP} \tag{8}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{9}$$

Measure	VGG16	VGG19	ResNet34	ResNet50
Sensitivity	0.7500	0.8519	0.7778	0.8214
Specificity	0.8846	0.7931	0.8889	0.9615
Precision	0.8750	0.7931	0.875	0.9583
Accuracy	0.8148	0.8214	0.8333	0.8889

F1 Score	0.8077	0.8214	0.8235	0.8846

Table 1. Shows the scores after calculating the confusion matrix

6. DISCUSSIONS AND CONCLUSIONS

Referring to the results presented in the previous section, in this section we are going to draw conclusions from the experiments made above, and understand how they paint the big picture of Lung Cancer Classifier using Artificial Intelligence.

Lung cancer is a complex problem and deep learning tools proved to be the right tool to tackle it. From the result of the experiment, it can be seen that ResNet50 deep learning architecture was performed especially well in extracting features and classifying the images. Clustering the results presented in Table 1, we can see the performance of all four algorithms grouped in measurement of Precision, Accuracy, and F1 Score. The scores below show us that ResNet-50 has had the best scores in every performance measure that was set in the beginning of the project. It scored a ~96% Precision, ~89% Accuracy, and ~88% F1 Score.

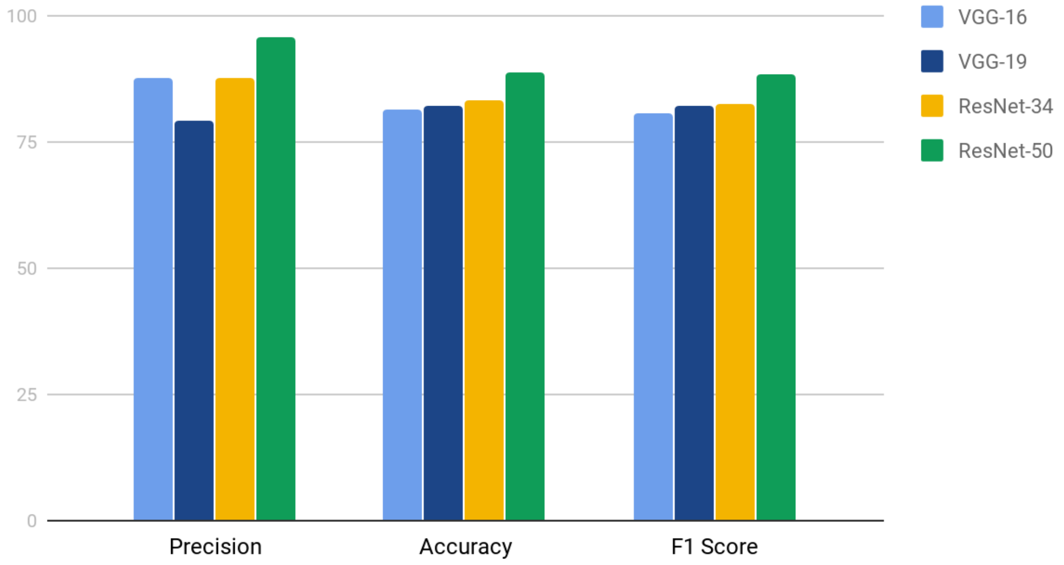


Figure 22 The performance of all four algorithms in Precision, Accuracy, and F1 Score

The data suggests that Residual Neural Networks architecture performed better. Due to its architecture and micro-architecture, this network’s ability to refrain itself from learning “bad” features which degrade overall learning has helped it get a higher accuracy score in the midst of other architectures. Additionally, the increase of number of layers proved to be very effective for solving complicated problems.

Measure	VGG16	VGG19	ResNet34	ResNet50
Sensitivity	0.7500	0.8519	0.7778	0.8214

Figure 23 Sensitivity (Recall) from the Confusion Matrix from each algorithm

In medical tests, sensitivity score, or known as the true positive rate, is an important measure for statistical tests; it shows, in our case, the number of cancer cases from all correct classifications. In the set of four different classification algorithms, ResNet50 performed the best with 88.89% accuracy. However, VGG19 had a better Sensitivity (Recall) score than the entire set of tested algorithms. In real world examples, with patients waiting their CT scan diagnoses, it is more important for the cases that have cancer to be classified as such, than cases not having cancer being classified as such.

Measure	VGG16	VGG19	ResNet34	ResNet50
Specificity	0.8846	0.7931	0.8889	0.9615

Figure 24 Specificity from the Confusion Matrix from each algorithm

On the other hand, specificity score, or known as the true negative rate, presents the actual negatives that are correctly classified as such. In our example, these would represent those patients who do not have cancer, and are identified as non-cancer cases. Consequently, the value patients get from sensitivity (the true positive rate) is higher than the value they get from specificity (the true negative rate), for the sole reason of being correctly classified as a cancer case in early stages of cancer, means saving a life [2].

The strategy of using pre-trained deep learning models has proved to be useful for such complicated tasks. On the other hand, in the research community Chen et al. got an 84% accuracy score combining CNN with SVM classifier [30], Monkam et al. reached accuracy of 88.28% pure CNN [28], Da Nóbrega et al. achieved accuracy of 88.41% with ResNet50 architecture[29], Wang et al. got scores 94.78% from pure CNN [30], from Da Silva et al. scored an accuracy of 97.62% [31]. As it can be seen, the accuracy scores are very diverse in the sense of accuracy and even more diverse in the tools and techniques used to find nodules, extract features, and classify the cancer as benignant or malignant.

The reliability of this research is impacted by the small amount of data available for training algorithms. Even with techniques as transfer learning which has proved to be practical in such cases, rare are cases where systems are impacted by the large amount of data. Additionally, the methodological choices were constrained by the lack of previous expertise on the topic of radiology and oncology. Nonetheless, the results remain valid for two-dimensional images of lung screenings.

Future recommendations for improving the accuracy of this model would be utilizing the power of semantic segmentation, particularly the U-Net Segmentation algorithm for biomedical segmentation. This algorithm would help by segmenting the lung area, from the rest of the organs, and this way it would make the feature extraction and classification process more accurate. In order for this algorithm to be considered for real life cases, it has to be susceptible of inputting full CT scans. In the future I look forward to upgrading this machine learning pipeline into a more practical, capable of handling real-life examples.

7. REFERENCES

1. “Worldwide Cancer Data.” *World Cancer Research Fund*, 17 July 2019, www.wcrf.org/dietandcancer/cancer-trends/worldwide-cancer-data.

2. Knight, Sean Blandin, et al. “Progress and Prospects of Early Detection in Lung Cancer.” *Open Biology*, vol. 7, no. 9, 2017, p. 170070., doi:10.1098/rsob.170070.

3. Mohammad, B. Al, et al. “Radiologist Performance in the Detection of Lung Cancer Using CT.” *Clinical Radiology*, vol. 74, no. 1, 2019, pp. 67–75., doi:10.1016/j.crad.2018.10.008.

4. Madan, Bhagyashree, et al. “Lung Cancer Detection Using Deep Learning.” *SSRN Electronic Journal*, 2019, doi:10.2139/ssrn.3370783.

5. Lee, June-Goo, et al. “Deep Learning in Medical Imaging: General Overview .” *Korean Journal of Radiology*, 2017, doi: http://dx.doi.org/10.3348/kjr.2017.18.4.570.

6. Yamashita, Rikiya, et al. “Convolutional Neural Networks: an Overview and Application in Radiology.” *Insights into Imaging*, 22 June 2018, doi:https://doi.org/10.1007/s13244-018-0639-9.

7. Polat, Huseyin, and Homay Danaei Mehr. “Classification of Pulmonary CT Images by Using Hybrid 3D-Deep Convolutional Neural Network Architecture.” *Applied Sciences*, vol. 9, no. 5, 2019, p. 940., doi:10.3390/app9050940.

8. “What Is Cancer?” *National Cancer Institute*, www.cancer.gov/about-cancer/understanding/what-is-cancer.

9. “What Is Lung Cancer? - Canadian Cancer Society.” *Www.cancer.ca*, www.cancer.ca/en/cancer-information/cancer-type/lung/lung-cancer/?region=qc.

10. “Small Cell Lung Cancer Treatment (PDQ®)–Patient Version.” *National Cancer Institute*, www.cancer.gov/types/lung/patient/small-cell-lung-treatment-pdq.

11. "IDC Digital Universe Study: Big Data, Bigger Digital Shadows and Biggest Growth in the Far East - Sponsored by EMC." 12 Dec. 2012.

12. "The Machine Learning Landscape." *Hands-on Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*, by Géron Aurélien, O'Reilly, 2019.

13. Sanjeevi, Madhu. "Chapter 4: Decision Trees Algorithms." *Medium*, Deep Math Machine Learning.ai, 23 Oct. 2018, medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1.

14. Rencberoglu, Emre. "Fundamental Techniques of Feature Engineering for Machine Learning." *Medium*, Towards Data Science, 1 Apr. 2019, towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114.

15. V, Avinash Sharma. "Understanding Activation Functions in Neural Networks." *Medium*, The Theory Of Everything, 30 Mar. 2017, medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0.

16. Eremenko, Kiril, and Hadelin de Ponteves. "Online Courses - Anytime, Anywhere." *Udemy*, 2017, www.udemy.com/course/deeplearning/learn/lecture/6743910?start=165#content.

17. Nielsen, Michael A. *Neural Networks and Deep Learning*. Determination Press, 2015.

18. Zeiler, Matthew D., and Rob Fergus. "Visualizing and Understanding Convolutional Networks." *Computer Vision – ECCV 2014 Lecture Notes in Computer Science*, 2014, pp. 818–833., doi:10.1007/978-3-319-10590-1_53.

19. Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM*, vol. 60, no. 6, 2017, pp. 84–90., doi:10.1145/3065386.

20. Rumelhart, David E., et al. “Learning Representations by Back-Propagating Errors.” *Nature*, vol. 323, no. 6088, 1986, pp. 533–536., doi:10.1038/323533a0.

21. Armato III, Samuel G., McLennan, Geoffrey, Bidaut, Luc, McNitt-Gray, Michael F., Meyer, Charles R., Reeves, Anthony P., … Clarke, Laurence P. (2015). Data From LIDC-IDRI. The Cancer Imaging Archive. http://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX

22. Szegedy, Christian, et al. “Going Deeper with Convolutions.” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, doi:10.1109/cvpr.2015.7298594.

23. He, Kaiming, et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, doi:10.1109/iccv.2015.123.

24. He, Kaiming, and Jian Sun. “Convolutional Neural Networks at Constrained Time Cost.” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, doi:10.1109/cvpr.2015.7299173.

25. Kornblith, Simon, et al. “Do Better ImageNet Models Transfer Better?” May 2018.

26. West, Jeremy; Ventura, Dan; Warnick, Sean (2007). *"Spring Research Presentation: A Theoretical Foundation for Inductive Transfer"*. Brigham Young University, College of Physical and Mathematical Sciences. Archived from *the original* on 2007-08-01. *Retrieved 2007-08-05.*

27. Howard, Jeremy. “Vision.learner.”, Sep. 2015, *Vision.learner | Fastai*, docs.fast.ai/vision.learner.html#Get-predictions.

28. Da Nóbrega , Raul Victor Medeiros, et al. “ Lung Nodule Classification via Deep Transfer Learning in CT Lung Images .” *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, 2018.

29. P, Monkam, et al. “CNN Models Discriminating between Pulmonary Micro-Nodules and Non-Nodules from CT Images.” *Biomed Eng Online,* 26 July 2018.

30. Wang, Xing, et al. “An Appraisal of Lung Nodules Automatic Classification Algorithms for CT Images.” *Sensors (Basel)*, Jan. 2019, doi:10.3390/s19010194.

31. Da Silva , GLF, et al. “Convolutional Neural Network-Based PSO for Lung Nodule False Positive Reduction on CT Images.” *Comput Methods Programs Biomed*, Aug. 2018.

8. APPENDIX

- The code for ResNet classificators can be found in this link:

https://colab.research.google.com/drive/1RM7Q19SaAeNuZh5jsjUoyGchv6ZCEm7c

- The code for VGG classificators can be found in this link:

https://colab.research.google.com/drive/1lq5YWshLXtUrY6URpTqE6NsLUvw0Ve2N#scrollTo=Q0aeOvnfT-f3