Cairo University - Faculty of Engineering
Department of Electronics & Electrical Communications
Second Year-MTH2175

Early prognosis of lung cancer using neural networks

Lungevity team

LUNGEVITY

Name	Section	ID
Erini Hosny Maher	1	9220175
Eman Yasser Gamal Eldin	1	9220181
Aya Hassan Abdelrahman	1	9220184
Manar Abdo Ali	4	9220868
Mennatallah Karam Shehata	4	9220875
Hana Khaled El-Sayed	4	9220945
Hana Maged Mohamed	4	9220947
Yara Ayman Abdelmaksoud	4	9220955
Yara Soliman Mohamed	4	9220956

Under the supervision of:

Dr. Samah El-Shafiey

Table of Contents

Table of Figures

1. Abstract

The innovative integration of CT scans, AI, and IoT presents a transformative solution for lung cancer patients. By employing advanced technology, we address the challenges faced by individuals dealing with lung cancer. From now the suspected patient could be able to face all these challenges through our AI-IoT based solution ,the patient is only required to input their CT scans to the AI model and it will firstly detect whether they have a tumor or not then if there were a tumor, it will classify whether it is malignant or benign case and after that the role of IoT solution will appear as the information of the patient (age, contact information) will be published on a server (MQTT server)connecting the laboratories with the hospitals and cancer treatment center , however, publishing the information of malignant and benign cases on the same topic (a topic is a string used in the MQTT protocol to identify and route messages) is not the best solution regarding the organization point of view , so we decided to establish two different topics on the server: one would be for malignant cases and it is supposed to be subscribed by hospitals specialized in treating cancer , the other would be for benign cases and it is supposed to be subscribed by the hospitals specialized in treating this type of tumor ,so by applying this solution we can see that the treatment and detection process will be significantly speeded up which will reflect positively on our patient's health and will highly increase the probability of survival. We chose ResNet-50 to be our AI model based on researches that showed that ResNet-50 reaches very high accuracy levels which was proved through testing our model and we also found out that the number of epochs is a very effective parameter in determining the model's accuracy we concluded that as the number of epochs increases the accuracy increases , this made us look forward to finding whether another factor like using another dataset or changing the number of neural network layers would affect the accuracy or not.

2. Introduction

The human body is made up of trillions of cells which represent the basic building block of all living organisms. Normally, these cells grow and multiply (through a process called cell division) to form new cells as the body needs them, and when these cells grow old or become damaged, they die, and new cells take their place, but sometimes this orderly process breaks down which makes the abnormal or damaged cells grow and multiply when they shouldn't. These abnormal cells may form tumors, which are lumps of tissue. These tumors may be cancerous (malignant) or not cancerous(benign).

Benign Tumors are generally harmless, grow slowly and they do not spread to other parts of the body, while malignant tumors, otherwise known as cancers, are very harmful to the body organs, can grow rapidly and spread throughout the body.

Lung cancer is an important health problem with an increasing incidence. It is a disease caused by uncontrolled cell division in your lungs, that eventually keep your lungs from working properly, Lung cancer starts in the lungs can spread to other organs in the body, such as lymph nodes or the brain. Cancers in other organs can also spread to the lungs. The spread of cancer cells from one organ to another is called metastasis.

There are many cancers that affect the lungs, but we usually use the term "lung cancer" for two main kinds:

1- **Non-small cell lung cancer (NSCLC)** is the most common type of lung cancer. It accounts for over 80% of lung cancer cases.
2- **Small cell lung cancer (SCLC)** grows more quickly and is harder to treat than NSCLC. It's often found as a relatively small lung tumor that's already spread to other parts of your body.

We are going to specifically focus on lung cancer due to:

1- It is one of the most common cancers.
2- It's a huge danger on human lives.

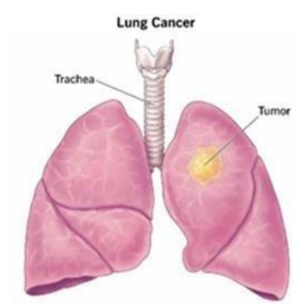

Figure 1:Lung Cancer

In most European countries, lung cancer has increased so dramatically that it may be considered one of the major health problems in the last century [1]. The most common causes of cancer-related death are cancers of the lung and bronchus (24%), prostate (10%) and colorectum (9%) in men, and lung and bronchus (23%), breast (15%), and colorectum (8%) in women [2]

Lung cancer is a significant public health concern, causing a considerable number of deaths globally. GLOBOCAN 2020 estimates of cancer incidence and mortality produced by the International Agency for Research on Cancer (IARC) show as lung cancer remains the leading cause of cancer death, with an estimated 1.8 million deaths (18%) in 2020

According to the latest WHO data published in 2020 Lung Cancers Deaths in Egypt reached 5,677 or 1.06% of total deaths. The age adjusted Death Rate is 8.02 per 100,000 of population ranks Egypt #115 in the world.

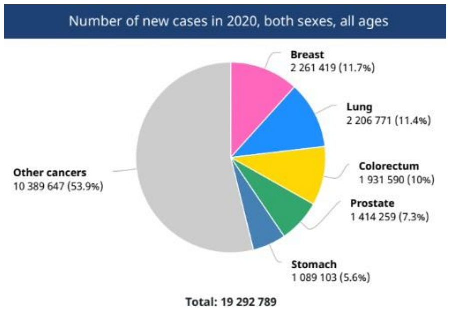

Figure 2: Number of new cancer cases in 2020 [3]

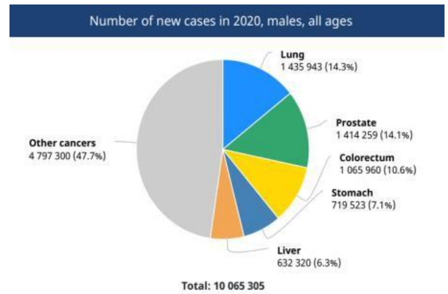

Figure 3:Number of deaths in 2020 [3]

Lung cancer is considered dangerous due to several factors:

1. **Firstly**, it often goes undetected in its early stages, leading to a higher likelihood of the cancer spreading to other parts of the body. This makes treatment more challenging and reduces the chances of successful outcomes.
2. **Additionally**, lung cancer is associated with a high mortality rate. It is one of the leading causes of cancer-related deaths globally.

3. **Furthermore**, there are many common causes of lung cancer as it is strongly linked to smoking and exposure to secondhand smoke or to harmful substances such as asbestos, radon and air pollution and the family history with lung cancer.

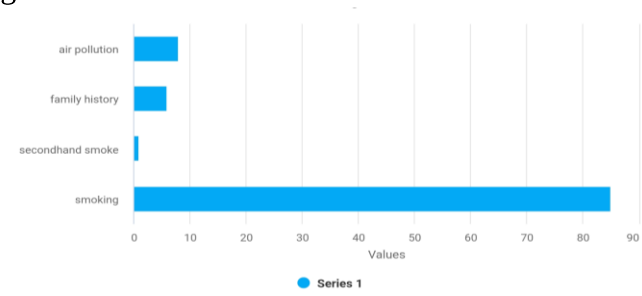

Figure 4: Causes of Lung Cancer

4. **Lastly,** lung cancer symptoms may not manifest until the disease has progressed significantly. This delay in symptom presentation can result in delayed diagnosis and treatment, further contributing to the potential dangers associated with lung cancer. Lung cancer symptoms may include:

- Coughing that gets worse or doesn't go away.
- Chest pain.
- Shortness of breath.

- Wheezing.
- Coughing up blood.
- Feeling very tired all the time.
- Weight loss with no known cause.

Our contribution:

as it is believed that early detection, improvement in cancer screening, treatment and prevention can significantly increases the rate of survival, this strongly encourages us to establish a software tool using neural networks and IoT to speed up the diagnosis process, this tool will require some lung scans from the patient to serve as an input to the program which in returns would:

1- inform the patient whether they are normal or have a tumor then classify this tumor to malignant or benign. give some results that show the status of the patient.

2- Connect the patient, the hospitals, clinics, and institutes that treat lung cancer to one server in order to establish a strong network and increase the connectivity between them and the patient which will in return speed up the treatment process.

The role of partial differential equations:

- We will use a PDE that determines to what extent lung cancer has spread, to know the most recent status of the patient.
- The deep learning algorithm which we are going to use to classify the status of the patient as mentioned above is strongly based on PDEs to improve its learning process which results in increasing the efficiency of the program to give a more and more accurate results.

the challenges faced by the patients in receiving health services:

1. Late-stage Diagnosis.
2. Limited Treatment Options: (limited effective therapies, especially for advanced stages).
3. Financial Burden.
4. High Relapse Rates: Lung cancer has a high rate of recurrence, necessitating ongoing monitoring and management even after successful treatment.
5. Comorbidities: Lung cancer patients often have other health conditions, such as chronic obstructive pulmonary disease (COPD) or cardiovascular issues, which complicate treatment planning and increase healthcare complexity.
6. Limited Screening Programs: The lack of widespread lung cancer screening programs contributes to late-stage diagnoses.
7. Access to Specialized Care.

8- Delay in starting the treatment process: spend long time searching for a place in some hospital.

3. Literature Review

- **Imaging techniques for diagnosis of lung cancer:**

X-rays:

X-rays provide 2D images of the chest and are often used as an initial screening tool. They can identify abnormalities, such as lung nodules or masses, but lack the detail of other imaging methods. X-rays offer numerous advantages for imaging, including their effectiveness in detecting various conditions, quick and painless nature, non-invasive procedure, and affordability. However, they also have limitations such as limited soft tissue differentiation, exposure to ionizing radiation, challenges in visualizing complex structures, and inability to capture dynamic images. It is essential for healthcare professionals to weigh the benefits and risks associated with X-rays to ensure the most appropriate imaging modality is selected for each patient's specific needs.

Computed tomography (CT) scans:

One of the most accurate imaging modalities for identifying infected lung nodules is computed tomography (CT) [4]. CT slices provide excellent resolution and are non-invasive and painless, all of which are important in the early stages of lung cancer detection and categorization into malignant and benign groups. Through the development of numerical approaches [5]and machine learning methods used for cancer detection, CT biomedical images have been widely employed in cancer diagnoses. There are various techniques used for lung cancer classification using CT images based on convolutional neural.

- **Lung Cancer Prediction Using Deep Learning:**

A study was published in March 2020 [6]to propose a computer aided detection (CADe) system for the early detection of lung nodules from low dose computed tomography (LDCT) images.

The proposed system initially preprocesses the raw data to improve the contrast of the low dose images. Compact deep learning features are then extracted by investigating different deep learning architectures, including Alex, VGG16, and VGG19 networks. To optimize the extracted set of features, a genetic algorithm (GA) is trained to select the most relevant features for early detection. three different types of classifiers are investigated to obtain the best detection accuracy, namely, KNN, decision trees, and SVM. The proposed system, using VGG19 architecture and SVM classifier, achieves the best detection accuracy of 96.25%, sensitivity of 97.5%, and specificity of 95%. Compared to other methods. The results show that using SVM achieves the best detection accuracies. it works by using a transfer function to non-linearly map input vectors into a high dimensional feature space, which helps to reduce the complexity of optimization, and it allows working directly with a high-dimensional vector space and it is still effective in cases where number of dimensions is greater than the number of samples. BUT SVM is unsuitable for large datasets, it will use a lot of processing time due to complexities in calculations. This will result large time to train the datasets itself. And SVM does not perform very well, when the data set has more noise.

Another study was published in April 2019 [7]. They used an ensemble of CNNs to try to improve upon classification by using both unsmoothed and smoothed images in separate networks.

They train CNN on all the pixels, however that may increase the machine price and coaching time. Therefore, instead they simply set to crop the photographs round the coordinates provided within the annotations. The annotations were provided in philosopher coordinates. In order that they had to be born-again to voxel coordinates. Conjointly the image intensity was outlined in Hounsfield scale. Therefore, it had to be rescaled for image process functions. While the script higher than under-sampled the negative category specified each one in half-dozen pictures had a nodule. The information set remains immensely unbalanced for coaching.

They tested their CNN model with 1623 pictures. They had associate degree validation accuracy of 93%. They have used the images of lung cancer as observation symbols, whereas the types of cancer have been states of their system. Investigational results show the performance and effectiveness of their system and demonstrate the usefulness of learning and enhancing the medical industry. Their model contains a preciseness of 89.3 try to recall of 72 %. The model contains a specificity of 98.2 %. There are many other systems available, but their model will help to get the accuracy at the earlier stages of the cancer.

There is another paper aims to examine and analyze four widely acknowledged complex neural network models, namely VGG16, VGG19, ResNet50, and Inception V3. the aim is to determine the model that offers the highest level of accuracy in lung cancer prediction:

Neural Network Models:

The VGG16 architecture consists of 13 convolution layers followed by three fully connected layers. The convolutional layers play a crucial role as an automatic entity that efficiently extracts features, capturing distinct patterns that can help distinguish between different categories of diseases. The initial convolutional layers acquire basic features like edges, while the subsequent convolutional layers combine these features together to generate intricate and complex characteristics [8].

The VGG19 architecture consists of 16-19 convolution layers. The convolutional kernel utilized in this architecture has dimensions of 3x3. Notably, the input to this architecture is a three- dimensional tensor with dimensions 224x224x3. The alternating arrangement of multiple convolutional layers and non-linear activation layers in a neural network architecture is more advanced compared to a single convolutional layer. This structure allows for better extraction of image features, downsampling through Maxpooling, and the use of a linear activation function (ReLU) to select the highest value within each image region. The downsampling layer helps the network resist image distortion while preserving important features and reducing the number of parameters

ResNet (Residual Networks) is a widely recognized deep neural network that forms the backbone of numerous computer vision applications. The Most Famous ResNet Model is ResNet50. It consists of five convolutional blocks placed on top of one another and it contains 50 hidden layers. The depth of the network is large. Because of the increased depth, the network correctly classifies the samples more efficiently. This network also automatically extracts features from input images and classifies the images.

The Inception v3 model, widely employed for image recognition, has consistently demonstrated remarkable accuracy exceeding 78.1 percent. Inception V3 is built upon the concept of "inception modules," which integrate multiple convolutional layers within a single layer. These modules allow the network to capture information at varying scales, leading to enhanced feature learning and representation.
Upon implementing the four models on the dataset, analysis reveals that the ResNet50 model exhibits the highest level of accuracy, closely trailed by the VGG16 and VGG19 models. Conversely, the Inception V3 model demonstrates the lowest accuracy out of the four models tested.

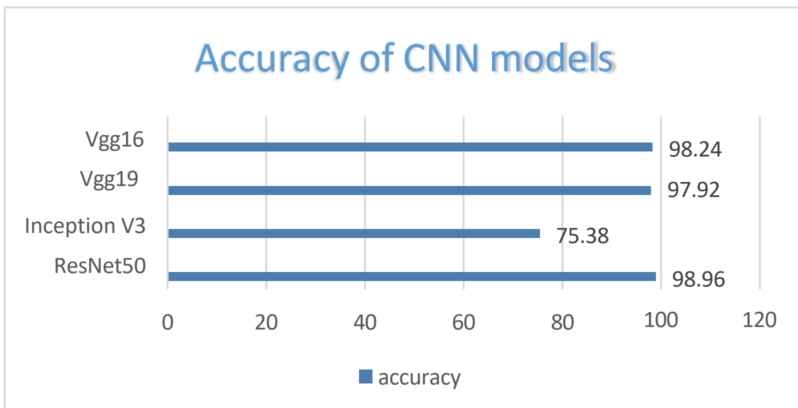

Figure 5:Accuracy of CNN models

- ## Comparison between different Neural Network Models:

Neural Network Models	Advantages	Disadvantages
VGG16	1. performs well on a wide range of vision tasks. 2. extract features from the images with high accuracy.	Training the model from scratch is time- consuming.
VGG19	1. has a simple architecture which it is easy to understand and implement.	has a large number of parameters and requires high memory.
ResNet 50	1. training networks with large number of layers without increasing the training error percentage. 2. used to tackle with certain issues like vanishing gradient problem.	The training process can be time-consuming and complex.
Inception V3	1. High-performance gain on convolutional neural networks. 2. Efficient utilization of computing resource with minimal increase in computation load for the high- performance output	time-consuming and requires high computational power.

- ## IoT

In order to connect distant devices to each other to allow the transmission of messages among these connected devices, there are many network protocols were developed. Historically, the main characterization of networks was based on the geographic scope and the span of the administration of the network. Networks were categorized as local-area networks **(LANs)** or wide-area networks **(WANs)**.

A **LAN** is a network infrastructure that spans a small geographical area. Specific features of LANs include:

- LANs interconnect end devices in a limited area such as a home, school, office building, or campus.
- A LAN is usually administered by a single organization or individual. The administrative control that governs the security and access control policies are enforced at the network level.

While A **WAN** is a network infrastructure that spans a wide geographical area. WANs are typically owned by enterprises or by Internet Service Providers. WANs are often managed by service providers (SP) or Internet Service Providers (ISP). An enterprise network may be managed internally but use WAN services from an ISP. Specific features of WANs include the following:

- WANs interconnect LANs over wide geographical areas such as cities, states, provinces, countries, or continents.
- WANs can include network segments administered by multiple service providers.
- WANs typically provide slower speed links between LANs.

Network protocols: For Devices to communicate, whether in small range using LANs or wide range using WANs, they must first conform to common communication rules, These rules are called network protocols. Like human languages, network devices take advantage of these communication rules to ensure the messages are sent and received, and that they can be understood. Two very important families of network protocols are **Ethernet** and **TCP/IP**.

- **Ethernet** is a family of technologies and protocols ruling the communication between local devices, ensuring they can successfully communicate while sharing the same communication media. Without Ethernet, signals placed on shared network media by network devices could interfere and become unreadable.
- **TCP/IP** protocol model for internetwork communications was created in the early 1970s, it is the family of protocols that interconnects billions of devices from networks throughout the Internet.

Notice that while Ethernet and TCP/IP are both communication protocols, they have very distinct functions and roles in network device communication. Ethernet ensures proper use of the local media, while TCP/IP facilitates remote communication and is independent of the media, and as we mentioned before our solution is aimed to be available for users' kilometers apart from each other, so we are going to focus on TCP/IP protocol in our solution.

IoT protocols:

IoT designers develop set of protocols created specifically for the IoT to meet the different requirements needed to establish an IoT solution. While most of the success of the Web is based on the use of a Client/Server approach using the HTTP protocol for exchanging messages, in the case of the IoT networks different approaches are emerging. CoAP and MQTT are two data protocols common in the IoT.

CoAP (Constrained Application Protocol): is a protocol intended for resource-constrained IoT devices that enables IoT devices to communicate with the Internet. CoAP is based on HTTP and the REST model where resources are retrieved from a server using URIs/URLs, The clients use the well-known methods of GET, PUT, POST, and DELETE to manipulate these resources. CoAP can be used via other mechanisms, such as SMS on mobile communication networks.

CoAP is designed to provide multicast support, low overhead, and simplicity. It is designed to work on microcontrollers with as low as 10 KB of RAM and 100 KB of storage space while also providing strong security.

MQTT (Message Queuing Telemetry Transport): is a lightweight protocol. MQTT is best suited for systems that rely on low bandwidth connections and require code with a small footprint. MQTT protocols uses the concept of publish-subscribe communications among nodes.

The publish-subscribe schema requires the presence of an intermediate node called a message broker. Every source of data must publish the data element on the broker node indicating to which "topic" the data belongs. The nodes interested in receiving data on a specific topic must subscribe to that topic on the MQTT broker. The broker will then distribute the messages to interested clients based on the topic of a message. An MQTT broker server is an essential component of the MQTT protocol, acting as a central message hub that facilitates communication between IoT devices and applications. It receives messages from devices (publishers) and delivers them to interested applications (subscribers) based on their subscription topics. The broker ensures reliable message delivery, maintains persistent sessions, and handles security measures to maintain a secure and scalable messaging infrastructure for IoT networks. We are going to apply MQTT protocol to our IoT solution due to the following:

1. MQTT provides high data integrity and security for remote environment.
2. MQTT is faster and more stable communication protocol to handle massive data than HTTP protocol.
3. The availability of an open-source MQTT broker server which can be accessible by anyone which is HiveMQ: A cloud-based MQTT broker offering high availability and scalability, and this will enable us to use and test the results of our solution.

MQTT Broker Server:

An MQTT broker server is an essential component of the MQTT protocol, acting as a central message hub that facilitates communication between IoT devices and applications. It receives messages from devices (publishers) and delivers them to interested applications (subscribers) based on their subscription topics. The broker ensures reliable message delivery, maintains persistent sessions, and handles security measures to maintain a secure and scalable messaging infrastructure for IoT networks.

Key Features of an MQTT Broker Server:

- Publish/Subscribe Architecture: MQTT employs a publish/subscribe pattern where devices publish messages to specific topics, and applications subscribe to those topics to receive relevant messages.
- Persistent Sessions: MQTT brokers maintain persistent sessions with clients, allowing clients to reconnect quickly after network disruptions.
- Security: MQTT supports TLS/SSL encryption for secure message transmission and client authentication mechanisms to prevent unauthorized access.
- Scalability: MQTT brokers can handle large numbers of connected clients and message volumes, making them suitable for large-scale IoT deployments.

- The innovative integration of CT scans, Al, and loT presents a transformative solution for lung cancer patients. By employing advanced technology, we address the challenges faced by individuals dealing with lung cancer. The smart CT scans, connected to the internet through loT, enable early and precise detection of tumors. A super-smart CT machine, hooked up to the internet, takes detailed images of their lungs. These images are then sent to an Al model trained specifically for tumor detection. The Al algorithm analyzes the intricate details within the scans, swiftly identifying the presence of any tumors. Upon detection, the Al model further classifies the tumor, distinguishing between benign and malignant cases with a high degree of accuracy. The critical phase follows as the patient's data, tagged with specific identifiers denoting benign as "N" and malignant as "M," is transmitted in real- time to hospital databases. This dynamic sharing mechanism enables hospitals to maintain a live record of available treatment spaces, ensuring rapid response and allocation of resources. designed to detect tumors swiftly and accurately. By streamlining the communication between diagnostic processes, classification, and treatment coordination, this integrated approach ensures that patients receive the necessary assistance faster, significantly improving the efficiency of lung cancer care.

4. Mathematical Modeling and Methodology:

In today's interconnected and data-driven world, the need for impactful solutions for complex problems is paramount. Traditional solutions often fall short due to their insufficiency to capture the intricacies and nonlinearities in real word phenomena. Here artificial intelligence excels in such situations, so in our methodology, we leveraged the power of deep learning by employing the ResNet50 architecture, a robust and widely recognized model for image recognition and classification. Our task is to make a binary classification of a patient with normal diagnosis and a patient who has a tumor and if he has a tumor, we will make one more binary classification between malignant and benign cases and using computed tomography (CT) scans as dataset.

Artificial intelligence is the science of studying, developing theories and algorithms of simulating human intelligence by computer systems. Machine learning is a subset from artificial intelligence which is the science that makes machines capable of thinking, making human-like decisions, learning from data, and improving their performance over time. Deep learning is a subfield in machine learning, which employs artificial neural networks composed of multiple layers of interconnected nodes known as neurons to process complex data and learn from it. The word deep refers to the number of hidden layers in the neural network. Here are some of the deep learning algorithms CNN, RNN, LSTM, GANS and so forth [9].

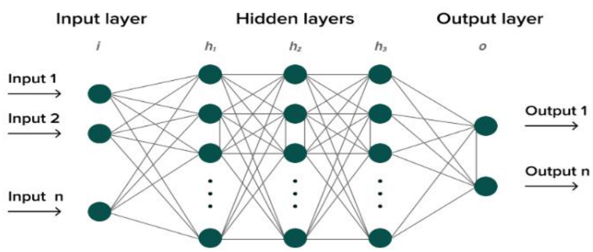

Figure 6:Schematic of a Deep Neural Network (DNN) [10]

Solution Methodology

Step 1: Image preprocessing

CT images underwent essential transformations to ensure their suitability for subsequent analysis. Here are four fundamental preprocessing steps:

1. **Image transformation**

Converting each image from JPG to RGB format, because JPG is a lossy compression format meaning that some information will be discarded, unlike RGB, a lossless color representation where each pixel is represented by a three-dimensional vector containing the red, green, and blue intensity values so it keeps all data information, in addition neural network training operations requires numerical data and RGB just meet the requirement. This makes RGB more convenient for image manipulation, feature extraction, cropping and resizing.

2. **Image resizing**

Image is resized to a fixed size of 224 x 224 x 3. The image resizing step is crucial for several reasons. It ensures compatibility with the network's filter size for accurate feature extraction. Resizing also improves efficiency by reducing the number of pixels, leading to faster training and inference. Furthermore, it allows for controlled feature extraction by defining the receptive field and the context the network considers. Additionally, resizing facilitates efficient batch processing by enabling memory allocation and parallel processing for multiple images.

3. **Converting images to arrays**

   It is considered the key ingredient here, as it offers a structured, homogenous, and organized representation of data, allowing for efficient access and manipulation. It also improves feature extraction by allowing for efficient manipulation and numerical operations. Additionally, it enhances the compatibility with machine learning libraries as most machine learning libraries and algorithms rely on data being in array format.

4. **Splitting dataset**

   Splitting the data into two parts, first part is the training dataset, it is also divided into two parts, one for the training and the other for validation, where the validation measures the performance of the model. Second part is the test dataset which is never seen by the model. 20% of the data is given to the test dataset and the remaining 80% goes to the training dataset.

STEP 2: Construction of the model

In continuation of the previous step, we employed the ResNet50 architecture for our modelling.

We loaded the ResNet50 from Keras library, then created variable named 'resnet_model' where we assigned a new sequential model then we set the pooling layer to average. A sequential model is a type of model that consists of a group of layers that are put together linearly or sequentially, every layer applies changes on the input image according to its specific function.

When the input image is passed through the sequential layers of the model it undergoes a series of transformations that extract and refine features of it, and the output of the previous layer is the input of the next layer.

Next, we added the flatten () layer to flatten the output of the ResNet50 model into a one-dimensional vector, this is because the dense layer only takes one-dimensional input.

Adding the first dense layer which consists of 512 neurons and with ReLU activation function that should be applied to the output of the dense layer.

Adding the second dense layer with 1 neuron only as it is required to output a single value which is the class label, with sigmoid activation function.

ResNet50

ResNet stands for Residual Networks. It is a CNN model such that it is a pretrained model; trained on an enormous image database which is 'ImageNet,' organized according to the WordNet hierarchy. It consists of 50 hidden layers, and it is a built-in model provided by Keras or TensorFlow deep learning libraries. ResNet50 consists of 16 residual blocks, with each block composed of several convolutional layers with residual connections. The architecture also includes pooling layers, fully connected layers, and a sigmoid output layer for classification.

In traditional deep neural networks, the vanishing gradient problem occurs when gradients become very small as they are propagated through many layers of a neural network. In simple words, as the number of layers increases training or test error gets worse which can lead to slower learning or inaccuracy as shown in

Figure 7. ResNet50 solves this problem by introducing skip connections, which allow the gradient to bypass certain layers in the network.

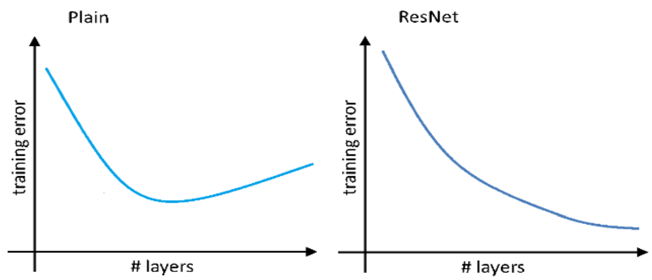

Figure 7: The graph to the right represents the training error for a plain network. The graph to the left represents the training error for a residual network. [11]

What is CNN?

Convolutional neural network (CNN) is a deep learning algorithm; it is also a feed forward neural network that excels at image processing such as image classification and object detection. It extracts the features of the image and converts it into lower dimension without losing its characteristics.

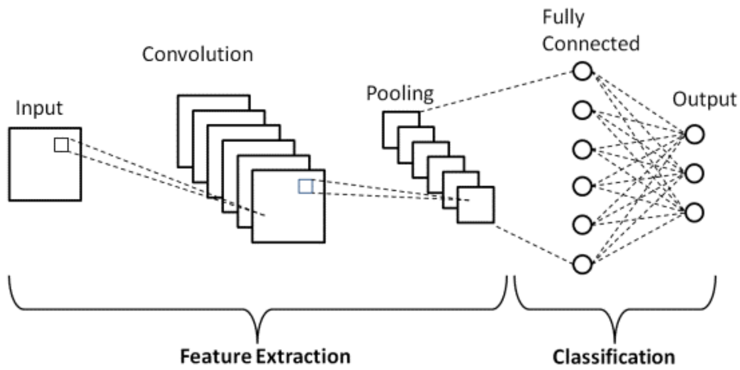

Figure 8: CNN architecture [12]

It encompasses an input layer, convolutional layer, pooling layer, dropout layer, flatten layer, fully connected layer, and output layer. The convolutional layer, pooling layer, dropout, flatten and fully connected layer are all the hidden layers.

What does Activation Function do?

It defines the output of a neuron given a set of inputs, the input value is given by computing the weighted summation of the neurons that points to a specific neuron, and pass this value to the activation function and then the activation function does some type of operation that transforms the summation value to number between some lower limit and upper limit, as the output value approaches the upper limit the more activated the neuron is, and when it approaches the lower limit the less activated the neuron is. If a model has no activation function, then the output is directly proportional to the input hence, the model would only be able to learn linear relationships between the inputs and the output. Activation functions introduce non-linearity to the neural network, allowing it to learn from complex, non-linear relationships in the data. We used two types of activation functions which are ReLU (Rectified linear unit) and sigmoid functions [13].

- **Sigmoid activation function:**

For instance, if the given input value is very negative the output will approach zero while if the input is very positive the output will approach one, so in this activation function the lower and upper limit is between 0 and 1. This makes it useful for binary classification problems.

The sigmoid function definition is:

$$S(X) = \frac{1}{1 + e^{-x}} \tag{1}$$

The sigmoid function is an S-shaped curve that maps any input to the range of 0 to 1 as shown in (**Figure 9)**.

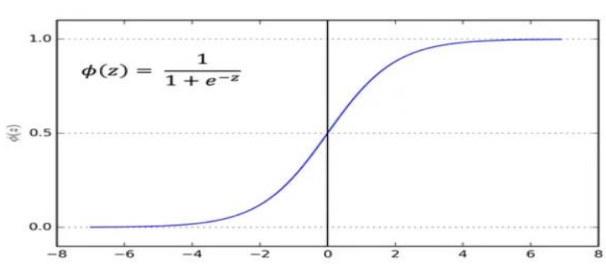

Figure 9: Sigmoid activation function [14]

- **ReLU:**

The ReLU (Rectified Linear Unit) function is given by:

$$R(X) = \max(0, x) \quad (2)$$

If the input is negative the output will be zero, while when the input is greater than or equal zero the output will be the number itself, so the limit here is between 0 and x (**Figure 10**).

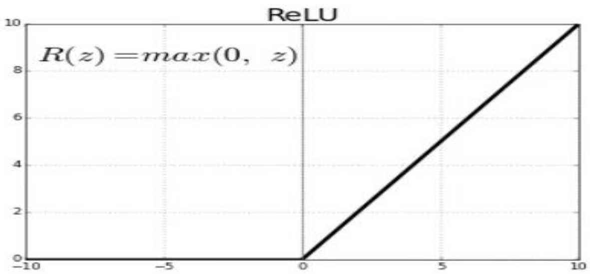

Figure 10: ReLU activation function [15]

As we see the concept of activation functions is inspired by the activity of our brains, if you smell a cookie scent some sort of neuron will be activated, and if you smell spoiled milk another neuron will be activated.

Math Behind Convolutional Neural Networks

The training process consists of a sequence of three fundamental stages: forward propagation, backpropagation, and parameter updating. Information flows through the network, errors are iteratively identified and corrected, and model parameters evolve to enhance predictive accuracy.

Forward Propagation

The forward propagation mechanism involves the gradual transmission of information from the input layer to the output layer. This process includes passing input data through the network's layers to generate an output or prediction. During computation, each neuron receives weighted inputs, applies an activation function, and passes the transformed information to the later layer. This process enhances the network's capacity to learn, notice patterns and make predictions from input data, enabling it to understand complex patterns and connections. Crucially, forward propagation allows neural networks to excel in various tasks such as classification, regression, and other machine learning objectives. [16]

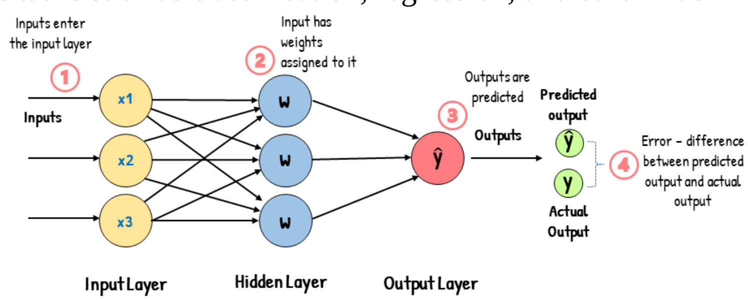

Figure 11: Feed-Forward Neural Network [17]

Backpropagation

Backpropagation refers to the method of calculating the gradient of neural network parameters. In short, the method traverses the network in reverse order [18], from the output to the input layer, according to the chain rule from calculus. The algorithm stores any intermediate variables (partial derivatives) required while calculating the gradient with respect to some parameters. Backpropagation sequentially calculates and stores the gradients of intermediate variables and parameters within the neural network in the reversed order.

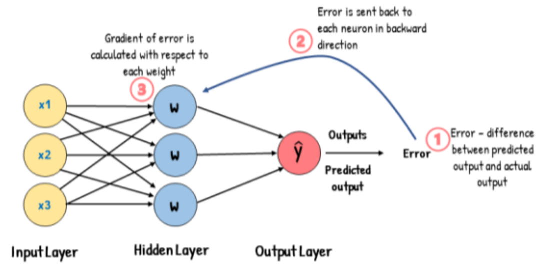

Figure 12: Backpropagation [17]

Updating weights & bias using Gradient Descent

Gradient Descent

Gradient descent is an optimization algorithm which is commonly used to train machine learning models and neural networks. Until the function is close to or equal to zero, the model will continue to adjust its parameters (weights and biases) to yield the smallest possible error (minimize the cost function) Minimizing the cost function means getting to the minimum point of the cost function. So, gradient descent aims to find a weight corresponding to the cost function's minimum point [19]. Gradient descent has three types: batch gradient descent, stochastic gradient descent (SGD) and mini-batch gradient descent.

Loss (or Cost) Function

Loss function is the function used for backpropagation. It computes the quantity that a model should seek to minimize during training. The choice of the Loss function completely depends on which task you want the network to do. Things get different in binary, multiclass, and regression problems.
In our model we use Binary Cross Entropy loss **Equation 3**. The binary cross-entropy loss function finds the dissimilarity between the predicted probability and the true binary label.

$$E(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \quad (3)$$

Where $E(y, \hat{y})$ represents the cross-entropy loss function between the true binary labels y and the predicted probabilities $\hat{y}$, n represents the number of samples in the dataset, y_i is the true label for the i-th sample where i ranges from 1 to n, and $\hat{y_i}$ is the predicted probability that the i-th sample belongs to class 1(positive class). Hence, the gradient of the loss function with respect to the predicted output (δ) is given by:

$$\delta = \frac{\partial E}{\partial \hat{y}} = -\frac{1}{n}(\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i}) \quad (4)$$

Learning rate

Also referred to as step size or the alpha is the size of the steps that are taken to reach the minimum. This is typically a small value, and it is evaluated and updated based on the behavior of the cost function. High learning rates result in larger steps but risks overshooting the minimum. Conversely, a low learning rate has small step sizes. While it has the advantage of more precision, the number of iterations compromises overall efficiency as this takes more time and computations to reach the minimum.

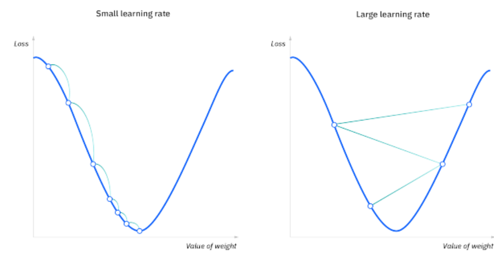

Figure 13: graph illustrating different learning rates within gradient descent. [20]

Optimizers

An optimizer is an algorithm or a method that adjusts the neural network attributes such as weights and learning rate, to improve accuracy and minimize losses by finding the optimal values for the model parameters. During the training of the model, the optimizer analyzes the gradient of the model's parameters with respect to the cost function. The optimizer then uses this information to update the parameters iteratively, aiming to reach a set of values that minimize the loss. Examples of Optimizers are Momentum, Adam, RMSProp and AdaGrad optimizers [21]. Here we will focus on Adam optimizer.

Adam optimizer (Adaptive moment estimation):

It is a combination of momentum and RMSProp optimizers. It maintains adaptive learning rates for each parameter using a term called first moment and the second moment of the gradient.

First, it initializes the first and second moment to zero. Then during iterations of the training process, the gradients of parameters are computed using backward propagation.

Second, updating the first and second moments: adam updates the first moment as mean and the second moment as variance using exponential moving averages.

Finally, adam computes the parameter update by combining the first and second moment estimates with the learning rate.

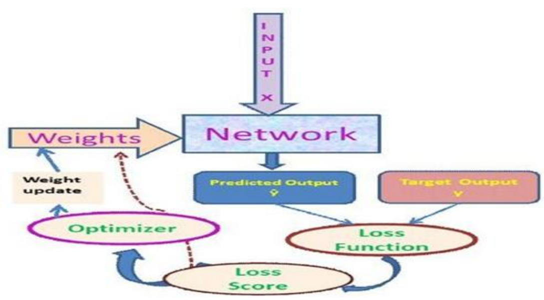

Figure 14: The process of the parameter updates in the network.

Learning process through layers

Let l be the layer number, x^l is the output at layer l, hence the input to the next layer. The coefficients defining the behavior of layer l are weights ω and biases b.

Forward propagation operation goes through two steps for each layer:

- Linear transformation

First, we calculate the weighted sum of inputs by multiplying input values with their corresponding weights and adding biases. The weighted input y is obtained via:

$$y^l = \omega^l \cdot x^{l-1} + b^l \tag{5}$$

- Activation function.

This weighted input then is fed elementwise into the activation function g^l to introduce nonlinearity to the network.

$$x^l = g^l(y^l) \tag{6}$$

For the backpropagation, let ($\delta = \frac{\partial E}{\partial x^l}$) be the gradient of the error with respect to the output of a layer.

The gradients with respect to the weights (ω), input (x^{l-1}), and bias (b) can be calculated as follows:

- Gradient with respect to the weights (W):

By substitution from **Equation 5** and taking the derivative:

$$\frac{\partial y^l}{\partial \omega^l} = \frac{\partial}{\partial \omega^l}(\omega^l \cdot x^{l-1} + b^l) = x^{l-1} \quad (7)$$

By applying the chain rule and substitution from **Equation 7**. So, gradient $\frac{\partial E}{\partial \omega^l}$ becomes,

$$\frac{\partial E}{\partial \omega^l} = \frac{\partial E}{\partial y^l} \cdot \frac{\partial y^l}{\partial \omega^l} = \frac{\partial E}{\partial y^l} \cdot [x^{l-1}]^T \quad (8)$$

- Gradient with respect to the inputs (x):

By applying chain rule, substitution from **Equation 5** and taking the derivative hence, the error gradient with respect to the inputs x is:

$$\frac{\partial E}{\partial x^{l-1}} = \sum_{i=1}^{n} \frac{\partial E}{\partial x^l} \cdot \frac{\partial x^l}{\partial y^l} \cdot \frac{\partial y^l}{\partial x^{l-1}} = \delta^l \cdot [\omega^l]^T * [g^{l-1}]'(x^{l-1}) \quad (9)$$

Where $*$ denotes the elementwise multiplication, n is the number of neurons in the layer l.

- Gradient with respect to the bias (b):

By applying chain rule, substitution from **Equations 5** and taking the derivative hence, the error gradient with respect to the bias term is:

$$\frac{\partial E}{\partial b^l} = \frac{\partial E}{\partial y^l} \cdot \frac{\partial y^l}{\partial b^l} = \frac{\partial E}{\partial y^l} \cdot \frac{\partial}{\partial b^l}\left(\omega^l \cdot x^{l-1} + b^l\right) \quad (10)$$

Hence, the error gradient with respect to the bias term is the same as the error gradient with respect to the layer output,

$$\frac{\partial E}{\partial b^l} = \frac{\partial E}{\partial y^l} \quad (11)$$

After calculating the error gradients with respect to the network parameters through backpropagation, these gradients are used to update the parameters to reduce the loss function as following:

- Updating the weights matrix ω

$$\omega_{ijk}^{new} = \omega_{ijk}^{old} - \alpha \cdot \frac{\partial E}{\partial \omega_{ijk}} \quad (12)$$

Where $\frac{\partial E}{\partial \omega_{ijk}}$ is the error gradient with respect to the weights matrix, α is the learning rate.

- Updating the bias matrix b:

$$b_j^{new} = b_j^{old} - \alpha \cdot \frac{\partial E}{\partial b_j} \quad (13)$$

Where $\frac{\partial E}{\partial b_j}$ is the error gradient with respect to the weights matrix, α is the learning rate.

Math Behind Residual Networks (ResNet)

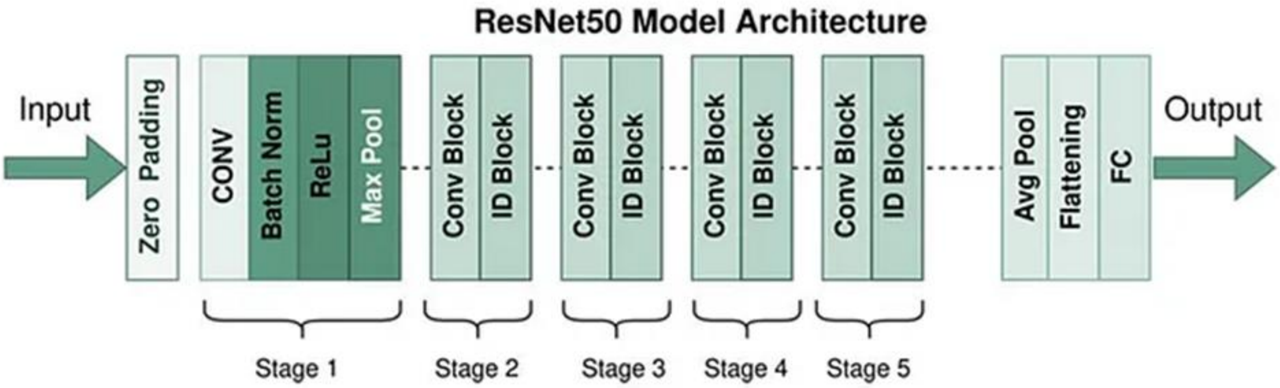

Figure 15: Keras ResNet-50 Model Architecture [22]

Vanishing Gradients Problem

Vanishing gradient is a problem that happens during the training of deep neural networks, where the gradients that are used to update the network become extremely small or "vanish" as they are backpropagated from the output layers to the earlier layers. The gradient of the early layers (the layers that are nearest to the input layer are derived by multiplying the gradients of the later layers (the layers that are near the output layer). Therefore, if the gradients of later layers are less than one, their multiplication vanishes at a particularly rapid pace. But, if a gradient is small, it won't be possible to effectively update the weights and biases of the initial layers with each training session. This means the weights of the neurons in the earlier layers learn very slowly or don't change at all. But earlier layers in the network are very significant because they are responsible for detecting simple patterns. This results in slower convergence & training instability, slower learning and even a complete lack of learning in very deep architectures. [23]

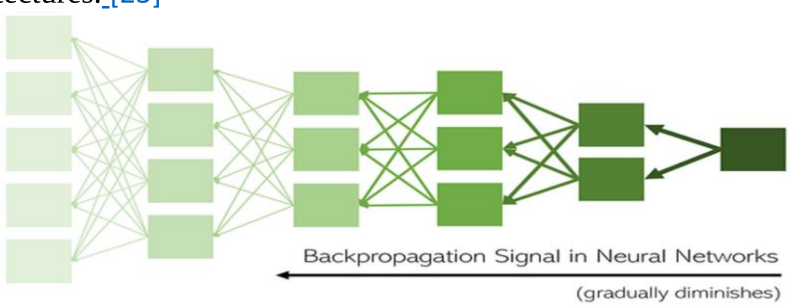

Figure 16: The vanishing of the gradients in the deep neural network [24]

Residual Connections

Residual connections, also known as skip connections or shortcut connections, were introduced in the ResNet architecture to address the vanishing gradient problem. Residual connections involve adding the input of a layer (identity mapping) to its output. The key idea is to let the network learn a residual function, which is the difference between the input and output. [25]

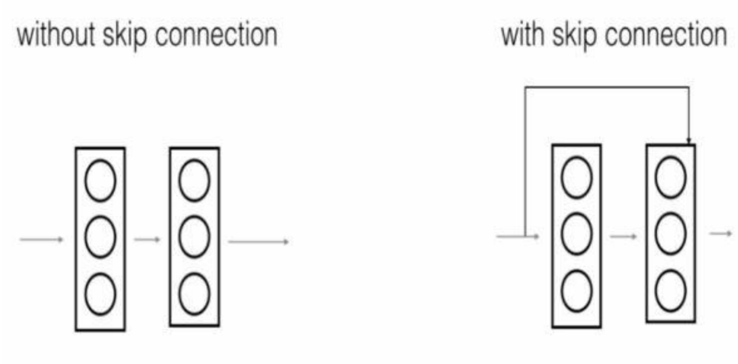

Figure 17: A ResNet block illustrating a skip-connection. [26]

In Forward Propagation:

Let x be the input to a layer, and $F(x)$ be the transformation applied by the layer. The output y of the layer with a residual connection is given by:

$$y = F(x) + x \tag{14}$$

This equation represents the addition of the residual function $F(x)$ and the input x to get the output y.

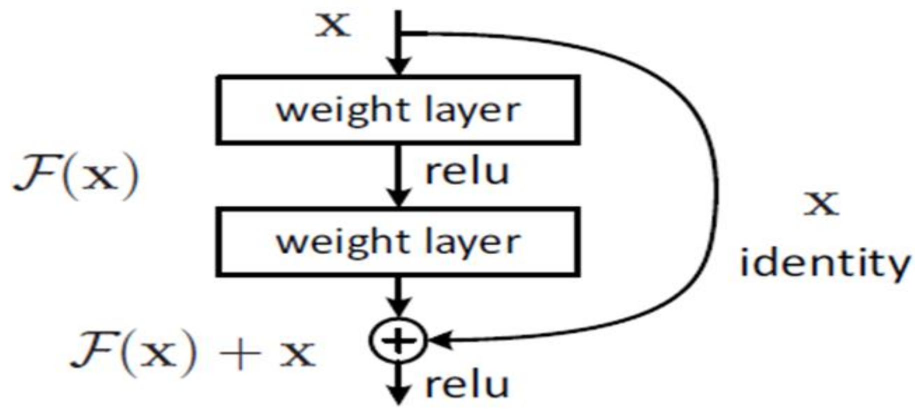

Figure 18: A building block for residual network [27]

In Backpropagation:

During backpropagation, when calculating gradients for the parameters of the layer, the gradient of the loss with respect to the input is calculated.

Let E be the loss and $\frac{\partial E}{\partial y}$ be the gradient of the loss with respect to the output of the layer. The gradient with respect to the input ($\frac{\partial E}{\partial x}$) is then calculated as:

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} \cdot \frac{\partial (F(x) + x)}{\partial x} = \frac{\partial E}{\partial y} \cdot F'(x) + \frac{\partial E}{\partial y} \tag{15}$$

The gradient of the input is directly propagated through the residual connection.
The inclusion of identity mapping allows the network to learn the residual function rather than the entire transformation. If the layer learns an identity mapping (i.e., $F(x) = 0$), the gradient can flow freely through the skip connection.

Batch Normalization

Batch normalization is another technique that has shown great success in solving the vanishing gradient problem and improving the training of DNNs [28]. As the name suggests, batch normalization is a normalization technique that we are applying to the input (current) batch of data by adjusting and scaling the activations to improve the stability and speed of training deep neural networks [29].

In Forward Propagation

Batch Normalization is a network layer that gets inserted between a hidden layer and the next hidden layer. The activations from the previous layer are passed as input to the Batch Norm. It has its own parameters: Two learnable parameters (beta and gamma), and two non-learnable parameters (Mean Moving Average and Variance Moving Average) are saved as part of the 'state' of the Batch Norm layer.

Let x be the input, γ $(mini\ batch\ mean), \beta$ (mini batch standard deviation) parameters to be learn, m number of values in the mini batch.

- **Mean and standard deviation calculation:**

For each activation vector, calculate the mean μ and variance σ^2 of all the values in the mini batch.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{16}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu)^2 \tag{17}$$

- **Normalization:** Batch normalization normalizes the input x within a mini batch by subtracting the mean and dividing by the standard deviation.

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{18}$$

Where ϵ is a small constant added to the denominator for numerical stability.

- **Scaling and Shifting:** After normalization, the values are scaled and shifted using learnable parameters (gamma and beta):

$$y = \gamma\hat{x} + \beta \tag{19}$$

In Backpropagation

We use chain rule to compute the gradients with respect to the batch norm parameters as follows:

- The gradient of the loss E with respect to the normalized input $\hat{x}_i$ of a layer in a neural network:

$$\frac{\partial \mathrm{E}}{\partial \hat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma \tag{20}$$

- The gradient of the loss E with respect to the variance σ^2 in Batch Normalization:

$$\frac{\partial E}{\partial \sigma^2} = \sum_{i=1}^{m} \frac{\partial E}{\partial \hat{x}_i} \cdot \frac{-(x_i - \mu)}{2(\sigma^2 - \epsilon)^{3/2}} \tag{21}$$

- The gradient of the loss E with respect to the mean μ in Batch Normalization:

$$\frac{\partial \mathrm{E}}{\partial \mu} = \sum_{i=1}^{m} \frac{\partial E}{\partial \hat{x}_i} \cdot \frac{-\mu}{\sqrt{\sigma^2 + \epsilon}} + \frac{\partial E}{\partial \sigma^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu)}{m} \tag{22}$$

- The gradient of the loss with respect to the input x_i of a layer in Batch Normalization:

$$\frac{\partial \mathrm{E}}{\partial x_i} = \frac{\partial \mathrm{E}}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2 + \epsilon}} + \frac{\partial E}{\partial \sigma^2} \cdot \frac{2(x_i - \mu)}{m} + \frac{\partial \mathrm{E}}{\partial \mu} \cdot \frac{1}{m} \tag{23}$$

- The gradient of the loss with respect to the scaling factor γ in Batch Normalization:

$$\frac{\partial \mathrm{E}}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial \mathrm{E}}{\partial y_i} \cdot \hat{x}_i \tag{24}$$

- The gradient of the loss with respect to the shifting factor β in Batch Normalization:

$$\frac{\partial \mathrm{E}}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial \mathrm{E}}{\partial y_i} \tag{25}$$

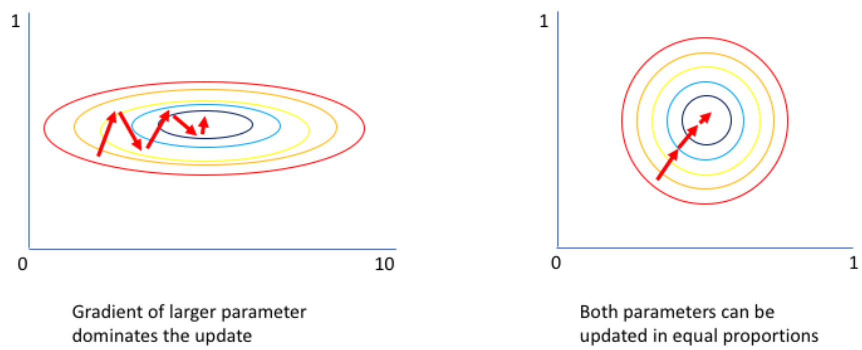

Figure 19: Batch norm, in effect, performs coordinated rescaling of its inputs. Normalized data helps the network converge faster. [30]

Each training iteration will become slower because of the extra normalization calculations during the forward pass and the additional hyperparameters to train during backpropagation. However, training should be faster overall as it should converge much more quickly, for the following reasons:

- **Higher learning rates are allowed:** Gradient descent usually requires small learning rates for the network to converge, this is because of gradient vanishing problem. As networks get deeper, gradients get smaller during back propagation, and so require even more iterations to converge (gradient vanishing problem). Using batch normalization allows much higher learning rates, increasing the speed at which networks train.
- **Easier weights initialization:** Choice of initial weights are crucial and can also influence training time. Weight initialization can be difficult, especially when creating deeper networks. Batch normalization helps reduce the sensitivity to the initial starting weights.
- **Makes more activation functions viable:** Some activation functions don't work well in certain situations.

To level up our hypothesis, we have integrated our model with a dedicated MQTT server. This vital link connects the laboratory directly to relevant cancer treatment authorities, enabling real-time data exchange and streamlining the flow of information. This not only accelerates the reporting of new discoveries but also allows for the immediate response and action necessary to optimize patient care.

IoT

- ## IoT Overview

The requirements to build our IoT network, that is going to connect all the laboratories all over Egypt and hence all the expected patients to the hospitals and the cancer treatment centers, are represented in having Computers that have access to the internet at the labs, hospitals and the treatment centers, in order to connect these end devices together , we are going to use the MQTT protocol and one of the free MQTT broker servers (HiveMQ) to act as a data center to this network.

- ## MQTT protocol

MQTT (Message Queuing Telemetry Transport) uses the concept of publish-subscribe communications among nodes.
The publish-subscribe schema requires the presence of an intermediate node called a message broker. Every source of data must publish the data element on the broker node indicating to which “topic” the data belongs. The nodes interested in receiving data on a specific topic must subscribe to that topic on the broker. The broker will then distribute the messages to interested clients based on the topic of a message. We are going to made use of it by establishing a connection (this connection is supposed to be owned by the labs in our solution) on an MQTT broker with these identifiers:
Host: mqtt-dashboard.com
Port: 8884
ClientID: clientId-LGfzGm5Mx2
Then we made two available topics on this connection one named ‘M’ and this topic must be subscribed by the hospitals and cancer treatment centers, so that the patient whose scans indicated that they got lung cancer, their personal information would be published on that topic, the other topic is named ‘B’, and this for non-cancerous cases.
To be able to test our solution we used one of the open-source MQTT brokers which is HIVEMQ, since this broker is on the internet then it uses the TCP /IP protocol as a protocol network.

- ## TCP /IP model

It is composed of four layers each of them has a certain function:

1. **Network access**: this layer is actually can be divided into: Physical Layer is responsible for generating the data and requesting connection and Data Link Layer, this layer is responsible for identifying the packet’s network protocol type, in this case, Error prevention and framing.

2. **Internet Layer:** This layer is responsible for the logical transmission of data over the entire network. The main protocols residing at this layer are as follows:

- **IP:** it stands for Internet Protocol, and it is responsible for delivering packets from the source host to the destination host by looking at the IP addresses in the packet headers. IP has 2 versions: IPv4 and IPv6. IPv4 is the one that most websites are using currently. But IPv6 is growing as the number of IPv4 addresses is limited in number when compared to the number of users.

- **ICMP:** it stands for Internet Control Message Protocol. It is encapsulated within IP datagrams and is responsible for providing hosts with information about network problems.

- **ARP:** it stands for Address Resolution Protocol. Its role is to find the hardware address of a host from a known IP address.

The Internet Layer is a layer in the Internet Protocol (IP) suite, which is the set of protocols that define the Internet. The Internet Layer is responsible for routing packets of data from one device to another across a network. It does this by assigning each device a unique IP address, which is used to identify the device and determine the route that packets should take to reach it.

3. **Transport Layer**: its role is to exchange data receipt acknowledgments and retransmit missing packets to ensure that packets arrive in order and without error.

4. **Application Layer:** It is responsible for end-to-end communication and error-free delivery of data. The most common protocol present in this layer is the HTTP and HTTPS: HTTP stands for Hypertext transfer protocol. It is used by the World Wide Web to manage communications between web browsers and servers. HTTPS stands for HTTP-Secure.

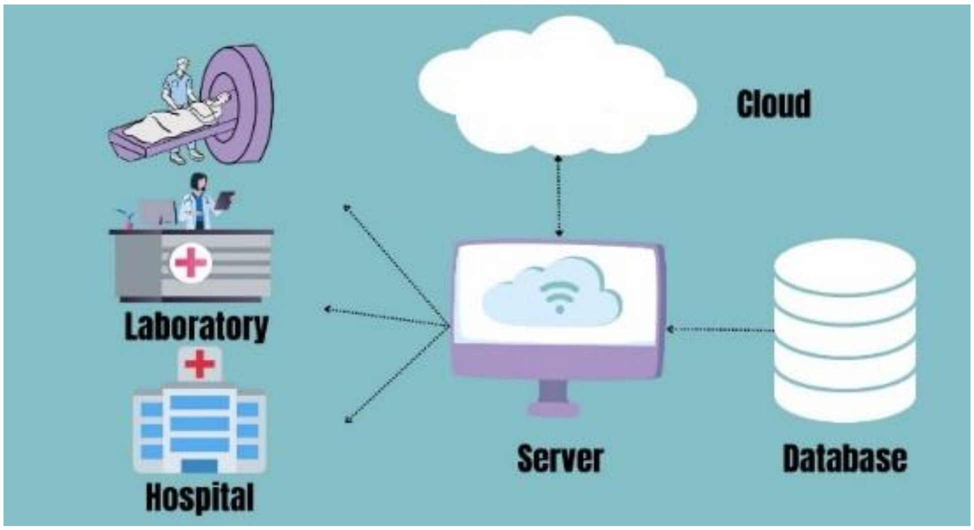

Figure 20:IOT all scenario

5. Experimental work

- **Dataset:**
We used two different datasets for training. All datasets were published publicly on Kaggle and Google datasets. First, the dataset consisted of 364 CT scan images that 238 of them belong to people who are suffering from lung cancer 126 of the rest belong to healthy people. The CT images are collected from the hospital situated in Iran [31]. Second, The dataset consisted of 1097 CT scan images that categorized into three classes (120 are benign, 561 are malignant, and 416 are normal) the cases vary in gender, age, educational attainment, area of residence and living status. The CT images are collected from two specialist hospitals The Iraq-Oncology Teaching Hospital/ National Center for Cancer Diseases (IQOTH/NCCD) [32].

- **Evaluation criteria:**
In order to focus on evaluating the performance of our ResNet50 model, we will take specific criteria into account which are accuracy that defines the model's overall ability to distinguish between different classes, simply it provides a general overview of its performance, sensitivity (recall) that tells us how good the model at identifying the true positives, precision that represents how efficient the model at avoiding false positives and F1 score, it is known as the harmonic mean of precision and sensitivity (recall), it provides a valuable tool for assessing the performance of the model combining precision and recall into a single, balanced metric [33].

- These four criteria are considered the most common and most important criteria to evaluate any deep learning model generally. However, Taking into consideration the misdiagnosis rate is very crucial in medical field applications , misdiagnosis here is meant to be predicting malignant cases as benign ones and benign cases as malignant one, because misdiagnoses would lead to incorrect treatment, which would be very dangerous on our patients' health ,also it can delay the correct treatment for serious cases which may decrease the probability of survival, the above mentioned criteria are calculated through the following formulas [34]:

Performance metric	Formula
Accuracy	$\frac{TP + TN}{P + N}$
Misdiagnosis rate	$\frac{FP}{FP + TN}$
Sensitivity or recall	$\frac{TP}{P}$
Precision	$\frac{TP}{TP + FP}$
F1 score	$\frac{2 \times precision \times recall}{precision + recall}$

- True positive (TP) where the predicted class matched the actual class and the actual class here is positive.
- True negative (TN) where the predicted class matched the actual class and the actual class here is negative.
- False positive (FP) is also called type I error, where the predicted class didn't match the actual class, the actual class was negative while the predicted class was positive.
- False negative (FN) is also called type II error, where the predicted class didn't match the actual class, the actual class was positive while the predicted class was negative.
- Positive (P) it indicates the number of real positive cases in the data, which is tumor class and malignant class in our model.
- Negative (N) it indicates the number of real negative cases in the data, which is normal class and benign class in our model.

- **The confusion matrix**

It is one of the representational tools in python imported from sklearn module to easily show the performance of the model. It is a square matrix of n rows and n columns that evaluates the performance and functionality of a model [35], where n represents the number of classes. In our model we have two consecutive classifying stages, one classifying between normal and tumor classes and the other is classifying between malignant and benign classes if there was tumor. So, n here is equal to 2, as we can see the previous mentioned criteria can be calculated from the values shown in the confusion matrix.

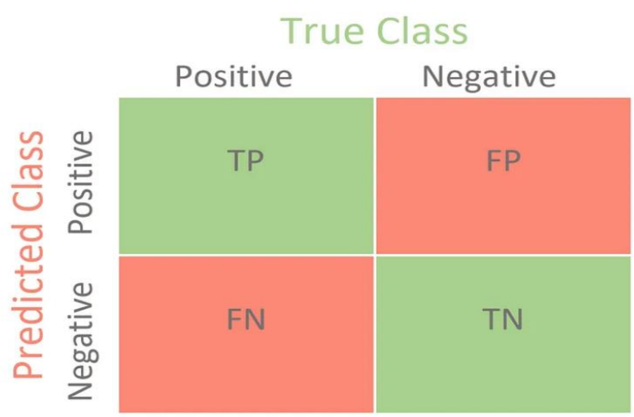

Figure 21:Confusion matrix structure

Analyzing the confusion matrix:

- The class label has two values: Positive or Negative.
- The columns represent the actual values of the class label.
- The rows represent the predicted values of the class label.
- The diagonal cells represent correct prediction where prediction class matched the actual class.
- The off-diagonal cells represent misclassification where prediction class didn't match the actual class.

- ## Fitting dataset into the model

In order to fit the dataset into the model, we imported Model from tensorflow. keras.models library, then we used model.fit () function which is a function used to take the training data and the model architecture as input then iteratively exposes the model to small batches of training data, calculates the model predictions, and compares them to the actual target values. The model architecture is represented by the following parameters:

➔ Batch size:

It represents the number of samples of training dataset processed by the model before updating its weights. large batch size may speed up the training process but will lead to overfitting, so we found out that choosing the batch size to be 32 is considered the best value to ensure balancing the two sides: the prevention of overfitting and keeping the model run at accepted speed. Moreover, to ensure the total avoiding of overfitting we employed the dropout technique from keras. Layers library [36].

Dropout: is a normalization technique used commonly in neural networks, it prevents overfitting and improves the generalization ability of the model. Dropout means that some nodes (input and hidden layers) dropped out or deactivated during the training process, in simple words dropout means ignoring some features (extra features) which are not necessary to be taken into consideration during the training of the model [37].

➔ Epochs:

It represents how many times the model passes through the entire dataset. For instance, if epochs = 10, it means that the model will see the training dataset 10 times.

➔ Shuffle:

It is a parameter that determines whether to rearrange the order of training dataset before each epoch. We set it to false, so the training data won't be shuffled before each epoch.

6. Results and Analysis

- **Testing scenarios:**

At this stage we are framing the testing scenarios designed to comprehensively evaluate the performance and functionality of our ResNet50 model. These scenarios will assess the model's ability to meet the requirements which are defined as achieving high accuracy of diagnosis, maintaining sensitivity, minimizing misclassification (false positive or false negative), in addition to generalizability such that the model performs well on unseen data.

We will conduct our testing with two different configurations primarily regarding the number of epochs. Firstly, we'll set the number of epochs to a small value while keeping the batch size and the shuffle fixed, subsequently we will increase the number of epochs to analyze its impact on accuracy and sensitivity. Also, we will employ a combination of positive and negative test cases to verify both predicted and unpredicted behavior. Through the execution of these testing scenarios, we aim to achieve the following:

- Verify that the system functions as intended according to the defined requirements.
- Identify any functional or non-functional defects.
- Gain valuable insights into the system's performance and behavior under different conditions.

Scenario 1: Epochs = 10

For normal and tumor classification

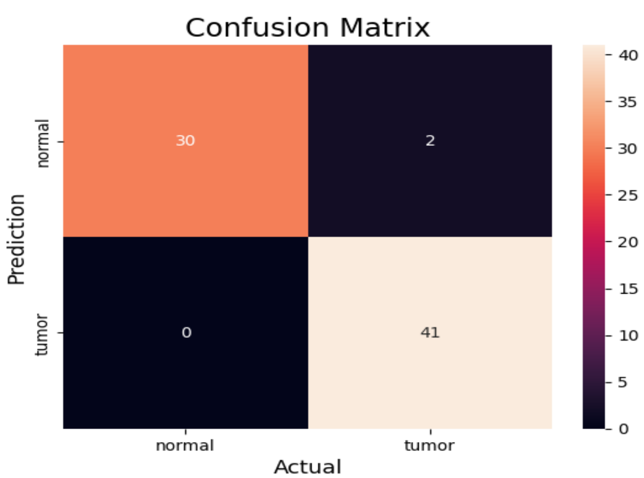

Figure 22: Confusion matrix for normal and tumor classification

Accuracy = 97.260274%.
Precision = 95.348837%.
Sensitivity (recall) = 100%.
F1 score = 97.6190476%.
Misdiagnosis rate = 0.0.

For malignant and benign classification

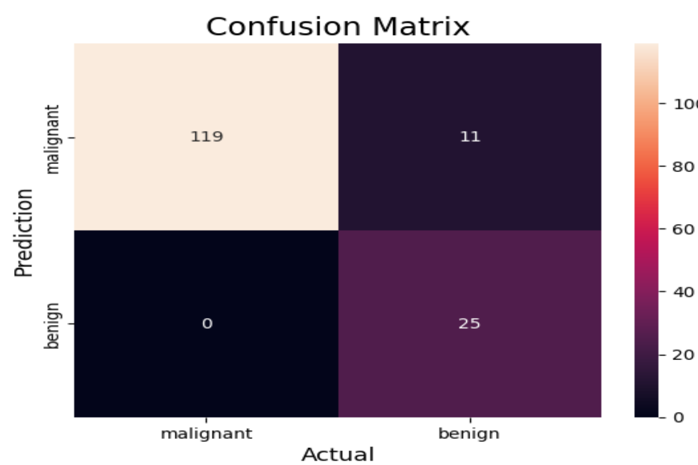

Figure 23: Confusion matrix of malignant and benign classification

Accuracy = 92.903226%.
Precision = 69.444445%.
Sensitivity (recall) = 100%.
F1 score = 81.967213%.
Misdiagnosis rate = 30.56%.

As we have mentioned before, we are able to calculate the evaluation criteria from the results obtained from the confusion matrix, although to ensure there will be no error in calculations, we decided to obtain their values by importing metrics library which is a part from scikit-learn library, then we used the following functions: metrics.accuracy_score() to obtain the accuracy, metrics.precision _score() to obtain the Precision, metrics.sensitivity _score() to obtain the sensitivity, metrics.f1_score() to obtain the f1 score, and we got the previous results.

For normal and tumor classification

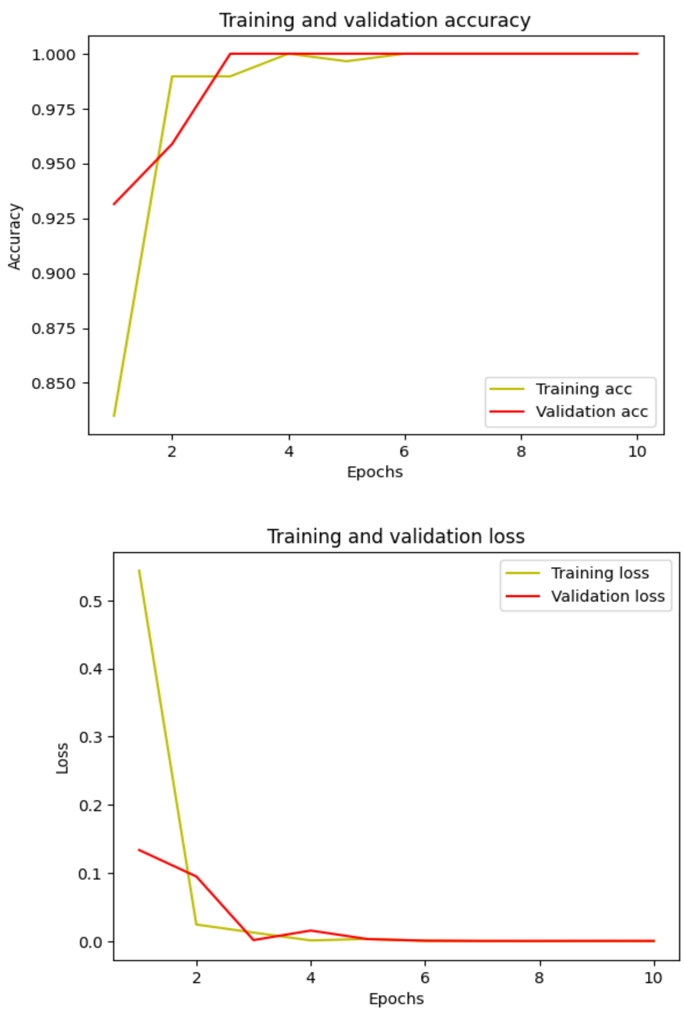

Figure 24: Training, validation accuracy and loss vs epochs

For malignant and benign classification

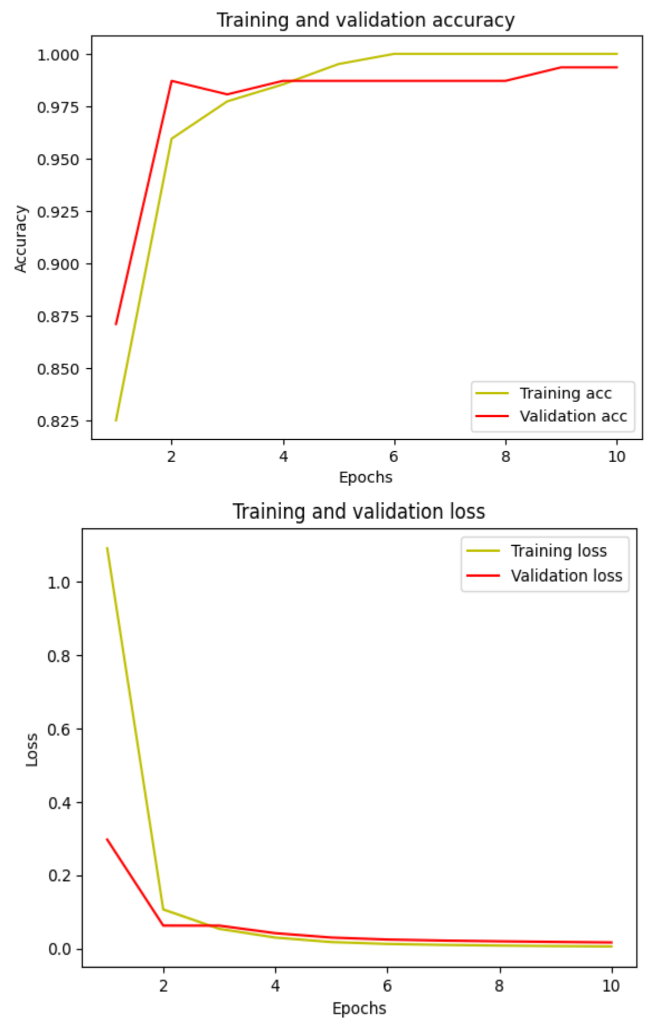

Figure 25: Training, validation accuracy and loss vs epoch

As illustrated from **figure 24 and 25**, we can observe that as the number of epochs increases, both the validation and training accuracy increases and both the validation and training loss decreases, it almost reaches zero.

Scenario 2: Epochs = 50

For normal and tumor classification

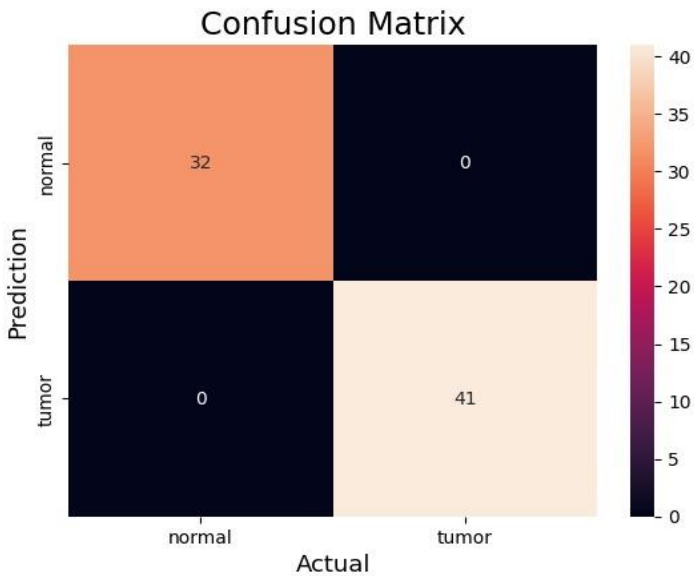

Figure 26: Confusion matrix for normal and tumor classification

For malignant and benign classification

Figure 27: Confusion matrix of malignant and benign classification

Accuracy = 100%
Precision = 100%
Sensitivity (recall) = 100%
F1 score = 100%
Misdiagnosis rate = 0.

Accuracy = 98.0645161%.
Precision = 89.285714%.
Sensitivity (recall) = 100%.
F1 score = 94.339623%.
Misdiagnosis rate = 10.714%.

For normal and tumor classification

For malignant and benign classification

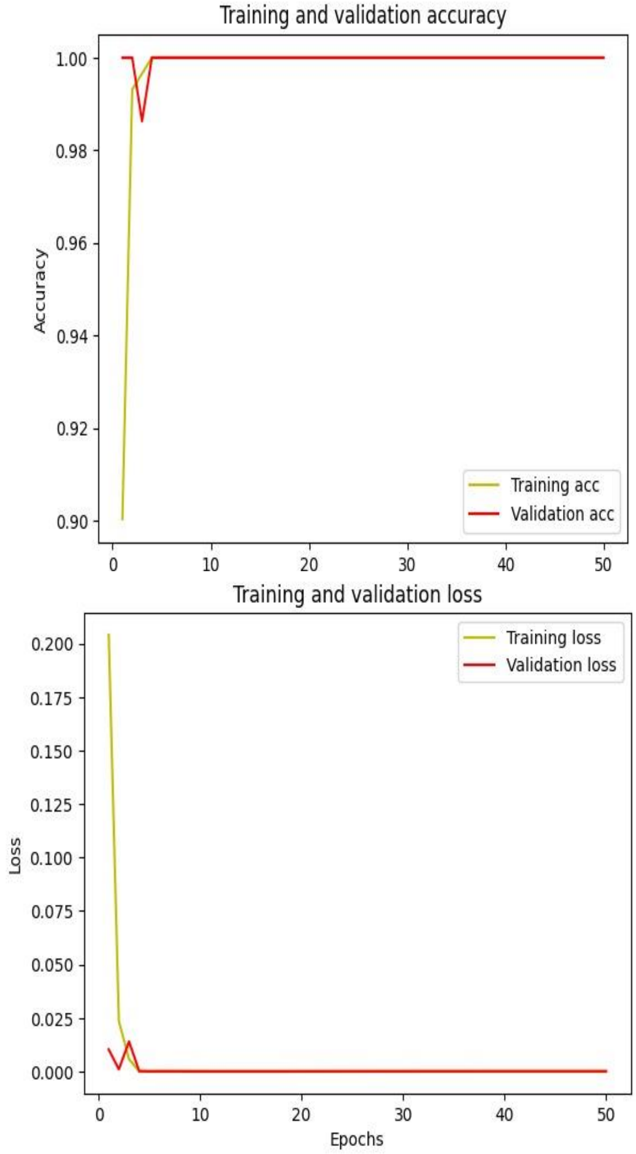

Figure 28: Training, validation accuracy and loss vs epochs

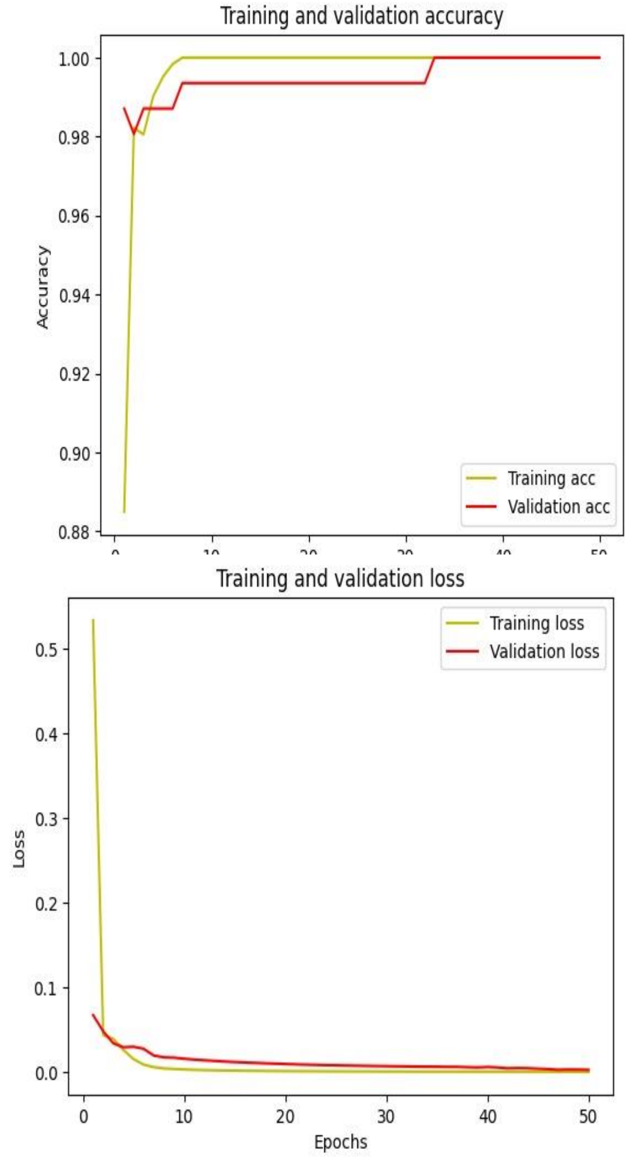

Figure 29: Training, validation accuracy and loss vs epochs

At this point we can notice that jumping from 10 epochs to 50 epochs has affected the evaluation criteria greatly, it achieved the maximum accuracy, precision, sensitivity, F1 score, So we can conclude that by increasing the number of epochs the model will reach a state of flawless performance and will learn very well.

➢ IoT Experimental work and analysis

As shown in **(Figure 30)** we used the open-source broker HIVEMQ as an example of MQTT servers to show how our model is able to connect to MQTT servers, the left-side screenshot shows the connection information of our topics:

Host: mqtt-dashboard.com
Port: 8884
ClientID: clientId-LGfzGm5Mx2

The right-side screenshot from the broker shows the results of two runs to the model: in the first one we entered a malignant case as an input ,as we can see the personal information of the patient is published on topic "M" as intended ,in the other run we were testing the result in case of benign cases and it is obvious that the message took our correct route to be published on the "B" topic which was specialized for benign cases.

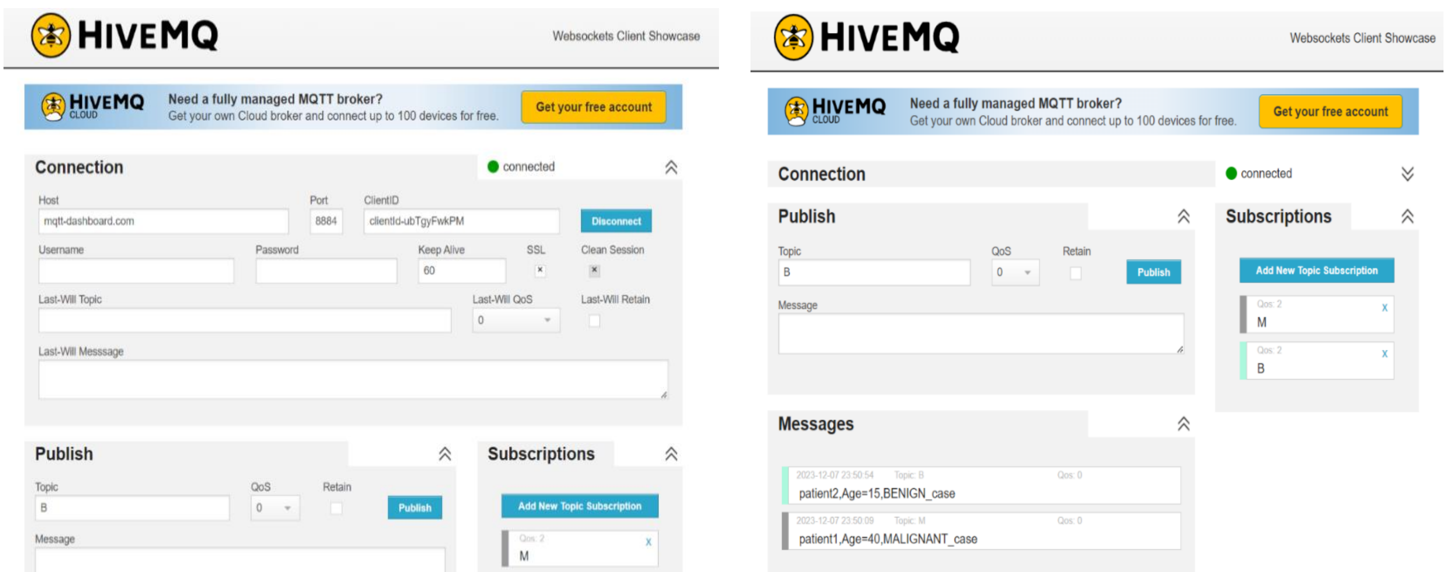

Figure 30: Connecting the patient to the MQTT server

7. Conclusion

We have presented a novel approach for the diagnosis of lung cancer using ResNet50 architecture, a pre- trained Convolutional network (CNN) model, which we selected after reviewing studies that demonstrated its high accuracy. We confirmed this by testing our model and discovered that the accuracy depended largely on the number of epochs. We observed that more epochs led to higher accuracy. Our mode also can achieve sensitivity and specificity in detecting if tumors are cancerous (malignant) or not cancerous(benign). We also use IOT to help patients to speed up their treatment process in order to increase the probability of survival by connect the patient, the hospitals, clinics, and institutes that treat lung cancer to one server in order to establish a strong network and increase the connectivity between them and the patient.

8. Future work

We are looking forward to extending our current researches to include the following important points:

1. Perform a detailed comparative analysis of ResNet-50 against competing models (AlexNet and Inception-V3) to determine the model with the highest accuracy for cancer detection.
2. Evaluate ResNet-50's performance across diverse datasets to understand its adaptability in cancer detection and classification.
3. Instead of employing a two-level binary classification for cancer detection and classification, we aim to transition towards a one-level multi-class classification approach, showing its impact on the model's accuracy.
4. Regarding the IoT solution, the current situations in Palestine makes Egypt try to bring some of the Palestinian patients who fight cancer to be treated in hospitals across Egypt, this motivates us to think about using the IoT solution that we created to serve those patients by making it available for them so they can publish their contact information and the cancer type that they fight on a topic ,on our server, which is subscribed by the Egyptian cancer treatment centers which will facilitate finding a place for each patient according to their case and age.

9. References

[1] Cheng TY, Cramb SM, Baade PD, Baade PD, Youlden DR, Nwogu C, Reid ME, "The International Epidemiology of Lung Cancer: Latest Trends, Disparities, and Tumor Characteristics," J Thorac Oncol, 2016.

[2] Siegel RL, Miller KD, Jemal A, "Cancer statistics," CA Cancer J Clin, 2019.

[3] Sung H, Ferlay J, Siegel RL, et al. Global Cancer Statistics 2020, " GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," CA Cancer J Clin. 2021;71(3):209-249., [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33538338/.

[4] Dehmeshki, Jamshid, Jun Chen, Manlio Valdivieso Casique, and Mustafa Karakoy, "Classification of lung data by sampling and support vector machine," In The 26th Annual International Conference of the IEEE Engineering, 2004.

[5] Lakshmanaprabu, S. K., Sachi Nandan Mohanty, K. Shankar, N. Arunkumar, and Gustavo Ramirez, "Optimal deep learning model for classification of lung cancer on CT images," Future Generation Computer Systems 92 , 2019.

[6] A. Elnakib, H. M. Amer, and F. E.Z. Abou-Chadi, "Early Lung Cancer Detection using Deep Learning Optimization," Int. J. Onl. Eng., vol. 16, no. 06, pp. pp. 82–94, 2020.

[7] Madan, Bhagyashree and Panchal, Akshay and Chavan, Dilip, "Lung Cancer Detection Using Deep Learning," 2nd International Conference on Advances in Science & Technology (ICAST), Mumbai,India, 2019.

[8] Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint , 2014.

[9] H. K. Anthony , M. M. Adam and N. A. Peter, "Comparison of two artificial intelligence-augmented ECG approaches: Machine learning and deep learning," *Journal of Electrocardiology,* vol. 79, pp. 75-80, 2023.

[10] Mackay CT, Nowell D., " Informed machine learning methods for application in engineering," A review. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2023.

[11] P. Sharma, "A Comprehensive Tutorial to learn Convolutional Neural Networks from Scratch (deeplearning.ai Course #4)," 14 July 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/12/guide-convolutional-neural-network-cnn/.

[12] "Recent Advances in Deep Learning Techniques for Face Recognition," Scientific Figure on ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Basic-CNN-Architecture-56_fig1_350326846 . [Accessed 13 Dec 2023].

[13] S. Sharma, S. Sharma and A. Athaiya , "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology,* vol. 4, no. 12, pp. 310-316, 2020.

[14] "MRFGRO: a hybrid meta-heuristic feature selection method for screening COVID-19 using deep features," Scientific Figure on ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Graphical-representation-of-sigmoid-function_fig3_357068475. [Accessed 13 Dec 2023].

[15] "Multi-Task Convolutional Learning for Flame," Scientific Figure on ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Graph-for-ReLU-activation-function_fig5_344399787 . [Accessed 13 Dec 2023].

[16] A. Zhang, Z. C. Lipton, M. Li and A. J. Smola, "Forward Propagation, Backward Propagation, and Computational Graphs," in *Dive into Deep Learning*, Cambridge University Press, 2023, pp. 166-169.

[17] M. Kalirane, "Gradient Descent vs. Backpropagation: What's the Difference?," 5 April 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/.

[18] M. Kalirane, "Gradient Descent vs. Backpropagation: What's the Difference?," Analytics Vidhya, 02 Jan 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/. [Accessed 07 12 2023].

[19] IBM, "What Is Gradient Descent? | IBM," www.ibm.com, [Online]. Available: https://www.ibm.com/topics/gradient-descent.

[20] Y. Adari, "Beyond Backpropagation: Enhancing Neural Network Training with Optimizers," [Online]. Available: https://medium.com/@yaswanth.adari0/beyond-backpropagation-enhancing-neural-network-training-with-optimizers-2630500d475b.

[21] A. K. Jain, . P. . P. Rao and V. Sharma, "Optimizers in Deep Learning: An Imperative Study and Analysis," *Third International Conference on Science & Technology Metrics,* 2021.

[22] a. d. serej, "ResNet-50," Medium, 23 Dec 2022. [Online]. Available: https://medium.com/@arashserej/resnet-50-83b3ff33be7d.

[23] "Vanishing gradient problem," Engati, [Online]. Available: https://www.engati.com/glossary/vanishing-gradient-problem. [Accessed 11 12 2023].

[24] A. Ye, "Batch Normalization: An Incredibly Versatile Deep Learning Tool," Medium, 27 May 2020. [Online]. Available: https://towardsdatascience.com/batch-normalization-the-greatest-breakthrough-in-deep-learning-77e64909d81d.

[25] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[26] "Performance Analysis On Bangla Handwritten Digit Recognition Using CNN And Transfer Learning," Figure on ResearchGate, 14 Dec 2023. [Online]. Available: https://www.researchgate.net/figure/Flattened-as-the-fully-connected-layer-12-RESIDUAL-NETWORK-RESNET-CNN-has-different_fig1_354212966.

[27] K. Bhavnagri, "Vanishing Gradients," The Data Science Swiss Army Knife, 7 Jan 2020. [Online]. Available: https://www.kamwithk.com/vanishing-gradients.

[28] Amanatullah, "Vanishing Gradient Problem in Deep Learning: Understanding, Intuition, and Solutions," Medium, 12 Jun 2023. [Online]. Available: https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54. [Accessed 11 12 2023].

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *International conference on machine learning. pmlr,* pp. 446-456, 2015.

[30] Kristinelpetrosyan, "When and how to normalize your Data," Medium, 31 May 2021. [Online]. Available: https://kristinelpetrosyan.medium.com/when-and-how-to-normalize-your-data-7966d4bd1268.

[31] M. Data, "CT-Scans image," [Online]. Available: https://data.mendeley.com/datasets/p2r42nm2ty/1. [Accessed 5 12 2023].

[32] Kaggle, "The IQ-OTH/NCCD lung cancer dataset," [Online]. Available: https://www.kaggle.com/datasets/hamdallak/the-iqothnccd-lung-cancer-dataset. [Accessed 5 12 2023].

[33] N. Jiang and H. Liu, "Understand System’s Relative Effectiveness Using Adapted Confusion Matrix," *International Conference of Design, User experience, and Usability,* vol. 8012.

[34] D. Chicco, N. Tötsch and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluatio," *BioData Mining,* 2021.

[35] M. Navin J R and Pankaja R , "Performance Analysis of Text Classification Algorithms using Confusion Matrix," *International Journal of Engineering and Technical Research (IJETR),* vol. 6, no. 4, 2016.

[36] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express,* vol. 6, no. 4, pp. 312-315, 2020.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout:A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning,* 2014.

Appendix

Link to our AI model and IOT repository on Github:

https://github.com/hannahmagedd609/Lungevity/tree/main