International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(7s) 2026 pp. 82-102
https://doi.org/10.70917/ijcisim-2026-3047
Article

Design and Validation of a Mobile and IoT-Enabled Deep Learning Framework for Point-of-Care Pulmonary Disease Diagnosis

Jannu Swetha Priyanka[1,2]*, Senguttuvan Thenmozhi[3]

[1]Department of Electronics and Communication Engineering,Dayananda Sagar College of Engineering, Bangalore, India
[2]Department of Electronics and Communication Engineering, Vardhaman College of Engineering, Hyderabad, India
[3]Department of Electronics and Communication Engineering,Dayananda Sagar College of Engineering, Bangalore, India
* Corresponding author's Email: swethapriyanka1823@gmail.com

Abstract: Pulmonary diseases are a cause of illness and death worldwide. To improve outcomes early and accurate diagnosis is crucial. Typically diagnosing these diseases relies on hospital infrastructure, which can be hard to access in areas. Recent advancements in technology such as learning, Internet of Things (IoT) and mobile edge computing offer new possibilities for healthcare solutions that can be used anywhere. This paper presents a mobile and IoT-enabled deep learning system for diagnosing pulmonary diseases at the point of care. The system combines medical devices, mobile edge computing and simple deep learning models to quickly analyze pulmonary data [1]. A four-layer system is designed: data collection, edge processing, learning analysis, and cloud validation. Mathematical models are created to assess system performance, including latency, computational complexity, communication overhead and energy consumption. An experimental evaluation using available pulmonary imaging and respiratory sound datasets is also presented. The goal of this framework is to make pulmonary disease diagnosis more accessible, by providing clinical decision support. The results show that integrating learning with IoT healthcare systems can enable scalable and real-time pulmonary disease diagnosis. The proposed system aims to improve accessibility to disease diagnosis. Pulmonary disease diagnosis can be done quickly and efficiently using this system. Deep learning and IoT technologies make pulmonary disease diagnosis at the point of care. The system uses data to provide accurate diagnosis. Pulmonary diseases can be diagnosed using mobile and IoT-enabled deep learning framework.
Keywords: Pulmonary disease diagnosis; deep learning; Internet of Things (IoT); mobile edge computing; point-of-care healthcare; medical image analysis[2].

1. Introduction

Pulmonary diseases are a health issue globally. They cause deaths and are expensive to treat. Conditions such, as pneumonia, tuberculosis and chronic respiratory disorders affect millions of people every year. We need to diagnose these diseases for effective treatment [3]. The growing number of illnesses shows that we require diagnostic technologies. These technologies must help detect diseases and make informed clinical decisions. We need technologies to tackle pulmonary diseases.

Recently artificial intelligence and deep learning have made automated medical diagnosis systems better. Deep learning and artificial intelligence are really helping. Convolution neural networks are great at analyzing images like chest X-rays and CT scans [4]. These techniques help doctors see things that're hard to spot with regular analysis methods. As a result deep learning is a tool for healthcare professionals to detect pulmonary abnormalities accurately with the help of artificial intelligence and deep learning. Most deep learning-based diagnostic systems need cloud infrastructure or high-performance computing resources to work. These resources are usually only found in hospitals. This limits their use in areas where healthcare infrastructure is limited [5]. In developing regions it's hard to access medical facilities and expert radiologists. This can delay the diagnosis and treatment of diseases because of access, to artificial intelligence and deep learning based systems.

The Internet of Things has opened up opportunities for monitoring and medical services that are not in a hospital. Medical devices that are enabled can collect signals from the body. Send patient data to places where it can be looked at [6]. These technologies are used a lot in systems that monitor patients and in devices that people wear to stay healthy. Most systems that use the Internet of Things for healthcare just collect data and monitor patients they do not try to figure out what is wrong with them.

Mobile edge computing is a technology that helps intelligence work in real time. By looking at data where it is collected edge computing reduces the time it takes to communicate and does not need to use servers that are far away [7]. This is very important for healthcare because quick decisions can save lives. If we put learning models on devices that're near the patient we can look at medical data right away and keep patient data private and not use up too much bandwidth.

The Internet of Things and edge computing can work together to make systems that can diagnose diseases. These systems can give healthcare services right where the patient is [8]. They can help healthcare workers figure out what is wrong with patients even if they are in a place with resources. This paper is about a way to use the Internet of Things and deep learning to diagnose diseases that affect the lungs [9]. The system we propose uses devices, edge computing and cloud learning to help doctors make decisions quickly.

The main things we did in this study are:

1. We made a system that uses the Internet of Things and deep learning to diagnose diseases that affect the lungs.

2. We made a system with layers that can collect data process it and use deep learning to make decisions.

3. We used math to figure out how long things take, how hard they are to do and how energy they use.

4. We found a way to test our system using data from patients with diseases that affect the lungs.

5. We showed that it is possible to use the Internet of Things and artificial intelligence to give healthcare services in a way.

The rest of this paper is organized like this. Section 2 looks at what other people have done to use learning and the Internet of Things to diagnose diseases that affect the lungs. Section 3 talks about the system we propose. Section 4 talks about how we modeled and validated our system [14]. Section 5 talks about how we designed and tested our system. Section 6 talks about what we think will happen and what it means. Finally Section 7 sums up what we did. Talks, about what we should do next.

2. Literature Survey

Pulmonary diseases like pneumonia chronic obstructive pulmonary disease, tuberculosis and lung cancer are still among the causes of death worldwide. It is very important to diagnose these conditions so that patients can get the right treatment and have better outcomes [15].In the few years artificial intelligence and deep learning have had a big impact on how medical images are looked at and how doctors make decisions. Researchers have been trying to use machine learning to detect diseases using medical images and signals from the body.

Rajpurkar and his team created a system called CheXNet which's a deep learning model that can detect pneumonia from chest X-ray images. This model was trained on a dataset of chest X-ray images. Was able to perform as well as doctors in identifying pneumonia cases. The pulmonary diseases are the focus here and the use of deep learning to detect pulmonary diseases is very effective. This study showed that deep learning can be very useful in analyzing medical images and can help doctors make diagnoses of pulmonary diseases, like pneumonia.

Wang and his team made a collection of chest X-ray images. This collection of images is very useful for training learning models to find diseases in the lungs. The collection of images helps researchers test deep learning models and see how well they can find diseases.

Shin and his team tried something called transfer learning. Transfer learning is when a model that has already been trained on one task is used on a task. They found that this can help find lung diseases from images. It can also reduce the need for a collection of images.

Huang and his team made a learning model called DenseNet. DenseNet is very good at looking at images. This model has been used in studies to find diseases from chest X-ray and CT images. Lakhani and Sundaram used learning to find tuberculosis from chest X-ray images. They found that deep learning models can be more accurate than computer-aided diagnosis systems.

Esteva and his team showed that deep learning models can be as good as doctors at looking at images. They were looking at skin cancer. Their work helped us understand that learning can be used in other areas of medicine including finding diseases in the lungs.

Some researchers are also looking at the sounds of breathing to find diseases. Rocha and his team made a system that can listen to lung sounds to find problems. They found that this can help tell the difference between people and those with diseases like asthma and chronic obstructive pulmonary disease.

Perna and his team used learning to look at lung sounds and got results. They think that listening to breathing sounds could be a tool to use with medical imaging. In addition to looking at images and sounds researchers are also using Internet of Things technologies in healthcare. Zhang and his team made a system that can collect data from sensors and send it to the cloud for analysis. This can help doctors watch patients from away and get alerts if something goes wrong. Doctors can use learning models and Internet of Things technologies to find diseases, like pulmonary diseases. Pulmonary diseases can be found by using deep learning models to look at chest X-ray images and other medical images.

Shi and his team have been looking into something called edge computing which's when data is processed on devices that are close to the patient rather than in the cloud. This can help reduce delays and make medical systems more responsive.

Satyanarayanan has emphasized the importance of using edge computing in healthcare especially when working with devices that do not have a lot of power. By using deep learning models on these devices healthcare systems can analyze data quickly and keep patient information private.

Recently there has been a focus on combining intelligence with Internet of Things technologies to create smart healthcare systems. Islam and his team created a system that uses sensors and machine learning to monitor patients health in real time. They think that this could be used for diagnosis and telemedicine.

Even though there have been a lot of advancements in using learning to detect pulmonary diseases analyzing breathing sounds and using Internet of Things technologies in healthcare most studies have only looked at one thing at a time. Pulmonary diseases, like pneumonia chronic obstructive pulmonary disease, tuberculosis and lung cancer are still a problem and pulmonary diseases need to be diagnosed early and accurately so that patients can get the right treatment and have better outcomes [16].

Study	Task	Model	Deployment	Limitation
CheXNet	Pneumonia detection	CNN (DenseNet)	Cloud	High computational requirements
Transfer Learning Models	Multi-disease classification	CNN	Cloud	Limited real-time deployment
Respiratory Sound CNNs	Lung sound classification	CNN	Server	Lack of mobile integration
Edge AI Healthcare Systems	General health monitoring	Lightweight neural networks	Edge	Not specific to pulmonary diagnosis
IoT Healthcare Platforms	Remote monitoring	Sensor networks	Cloud	No intelligent disease diagnosis

Table 1 show us what some studies have found out the problems, with the ways that people are doing things now.

From what we can see when we compare these studies most of them focus on either using learning to analyze images or using Internet of Things technology to monitor healthcare [17]. Few studies have tried to bring together deep learning, mobile edge computing and Internet of Things technology into one system that can help diagnose lung diseases in real time when the patient is being treated.

Even though these studies have made a lot of progress in finding lung diseases and monitoring healthcare there are still some problems. Most of the research only looks at either analyzing images or monitoring physical signals and it does this separately [18]. A lot of deep learning models need to be processed in the cloud. This takes time. It is a problem. People are worried about keeping their data private when they use deep learning models. They are also worried, about the network being reliable when they use deep learning models. Deep learning models are important. People want to make sure they work well.Only a few studies have tried to use deep learning models on mobile devices or at the edge of the network to diagnose medical problems in real time.

Another big problem is that we do not have systems that can combine ways of diagnosing diseases, such as looking at medical images analyzing breathing sounds and monitoring health with sensors all, in one system [19]. We need to be able to do this to develop systems that can help doctors make decisions when treating patients.

So we really need a system that can bring together learning, Internet of Things technology and edge computing to help diagnose lung diseases early [20]. The research we are proposing is going to try to solve these problems by developing a system that uses devices and Internet of Things technology and deep learning to diagnose lung diseases in real time when the patient is being treated.

3. Proposed Mobile and IoT-Enabled Deep Learning Framework

3.1 Framework Overview

This study is about a system called Mobile and IoT-Enabled Deep Learning Framework or MIEDLF for short. The system is called Mobile and IoT-Enabled Deep Learning Framework. It is also referred to, as MIEDLF. The Mobile and IoT-Enabled Deep Learning Framework is being studied. It is designed to help doctors diagnose lung diseases quickly and accurately in places where they see patients [21].The system uses Internet of Things devices, mobile edge computing platforms and deep learning models to make a smart diagnostic system. This system can work well in healthcare settings that're not in hospitals.

The main reason for creating this system is to fix the problems with the way of diagnosing lung diseases. Usually doctors need to send patients to hospitals with special equipment and expert doctors to get a diagnosis [22].In many places especially far from cities these hospitals are not available. This means patients have to wait a time to get a diagnosis and treatment.

The new system solves this problem by allowing medical care to happen closer, to the patients. It uses mobile and edge computing devices to collect data prepare it for analysis and use learning to make a diagnosis [23]. The system has four parts:

1. Data Acquisition Layer
2. Edge Processing Layer
3. Deep Learning Inference Layer
4. Cloud Integration and Validation Layer

These parts work together to create a system that can collect medical data analyze it automatically and give the results to doctors in real time [24].

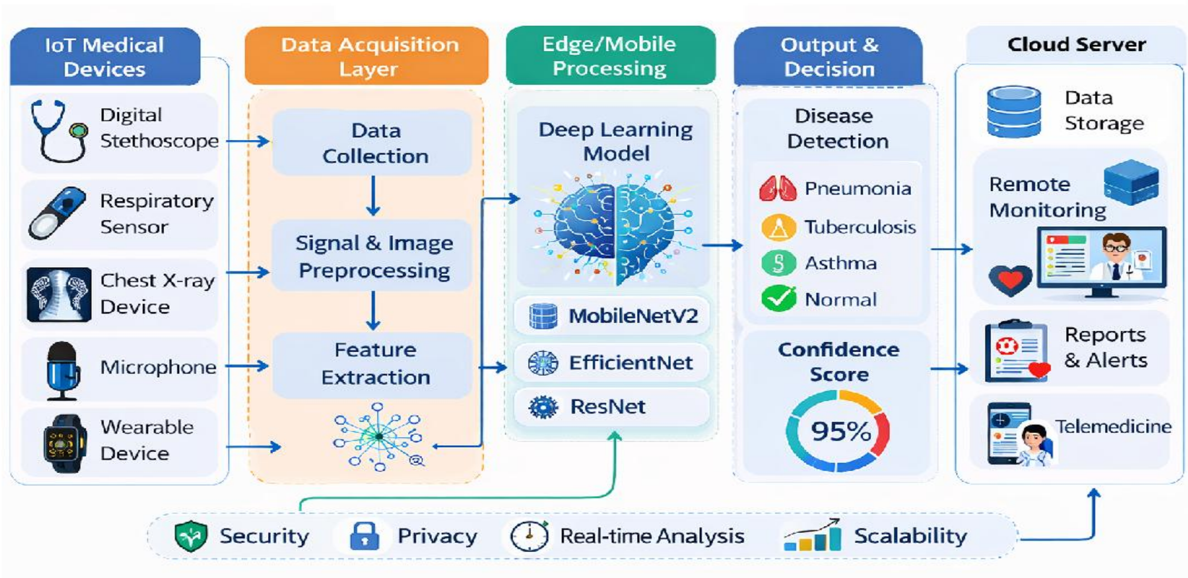

Figure 1. Architecture of the proposed mobile and IoT-enabled pulmonary disease diagnosis system

3.2 Data Acquisition Layer

The data acquisition layer collects health information using medical devices that work with the Internet of Things (IoT) [25]. These devices get signals and medical imaging data that is relevant to lung health. Common devices used to diagnose lung problems include portable chest X-ray machines, digital stethoscopes and sensors that monitor breathing [26]. Digital stethoscopes can record lung sounds while imaging systems show pictures of the lungs and any problems. We are using devices that talk to each other. These devices connect through Internet of Things communication protocols. The Internet of Things communication protocols are what we use to make the devices work together. This allows them to send data wirelessly to mobile devices or edge computing nodes [27].The medical data collected may include:

- Chest X-ray images
- Recordings of breathing sounds
- Oxygen levels in the blood
- Breathing rate measurements

Technologies like Wi-Fi, Bluetooth Low Energy and cellular networks help send data from IoT devices to the mobile edge processing unit [28]. This layer makes it easy to combine sensors with computing power needed for automated disease diagnosis using data from IoT devices and mobile edge processing unit. The data acquisition layer is key to getting health information. Enabled medical devices play a big role in collecting this information. Medical devices that work with IoT are essential for getting signals. These signals are relevant to lung health. The collected medical data from devices is crucial, for automated disease diagnosis.

3.3 Edge Processing Layer

The edge processing layer does some data processing tasks on mobile devices or edge computers before deep learning happens. This helps reduce delays in the network and also sends data to cloud servers. Some usual tasks done in preprocessing are: Making images look normal, Reducing noise, Making images clearer, Changing the size of images so they fit with what deep learning models need. These steps make the data going in better and more consistent which is really important for the models to make predictions. For images making them normal can be shown in a math formula, like this:

$$I_{norm} = \frac{I - \mu}{\sigma}$$

where:

- I represents the original image intensity
- μ denotes the mean pixel value
- σ represents the standard deviation

Similarly sounds from breathing can go through a process to make them clearer. Then turned into a special kind of picture. This helps computers understand the sounds better. The part that helps process information at the edge is really important [31].

It helps reduce the work that computers have to do and makes sure that the computer program, for learning gets the best data to work with [32].

3.4 Deep Learning Inference Layer

The deep learning inference layer is the part of our framework that helps diagnose diseases [33]. It uses computer programs called convolution neural networks to look at medical data and find patterns that might mean someone has a lung disease.

The deep learning model gets images or sound recordings of breathing as input. It then finds features in these inputs using many layers that look at the data in different ways. These features are then sent through layers that give scores for how likely it is that someone has a certain disease.

Lets say the medical data is X and the deep learning model is f(X,θ) where θ's the models settings. The disease that the model predicts is

$$Y = f(X, \theta)$$

Some neural networks are made to be lightweight. They are good for healthcare apps [36].Models like MobileNet and EfficientNet are often used because they don't need a lot of computing power but are still very accurate .

These models use operations that make them need fewer settings and less computing power [37].This means they can work on devices without slowing things down.

3.5 Cloud Integration and Validation Layer

The cloud is still important for model training checking the system and storing data [38].

In the cloud layer we can collect data from edge devices make it anonymous and then use it to retrain the deep learning model from time to time [39]. The updated model details can then go back to the devices. This makes the diagnosis of the system more accurate over time.

The cloud is really helpful because it keeps track of how the system's doing. It also makes it easier for doctors to work together [40]. For example reports from the system can go to the cloud. Then doctors can look at the results from the system when they are away. This mix of edge and cloud gives us the best of both the edge and the cloud. We get the benefits of doing some work on the devices and sharing knowledge in a place, like the cloud [41].

3.6 Diagnostic Workflow Algorithm

Here is how our system works:

Algorithm 1: IoT-Based Pulmonary Disease Diagnosis

Input: Patient medical data D

Output: Predicted pulmonary disease class C

Here are the steps:

1. We need to get information from things that can connect to the internet like devices that can connect to the internet.

2. Then we send this information to a computer that can process it which is called a processing unit.

3. Next we prepare the information we got from devices that can connect to the internet. It is ready to use.

4. After that we use a kind of computer program called a learning model to look at the information from devices that can connect to the internet and find the important parts.

5. The learning model helps us to calculate the chances that someone might have a disease using the information from devices that can connect to the internet.

6. Then we use the information from the learning model to decide which disease someone is most likely to have based on the data from devices that can connect to the internet.

7. Finally we show the diagnosis, which's the result of looking at the data from devices that can connect to the internet on a special app for health on a mobile phone called a mobile health interface.

8. We can also send the data from devices that can connect to the internet to the cloud, which's like a big computer in the sky for extra checking if we want to [42].

This shows us how the data, from devices that can connect to the internet moves through the system. It also shows us how devices that can connect to the internet, local processing units and the cloud all work together to help us [43].

3.7 Latency Considerations for Real-Time Diagnosis

For healthcare systems getting diagnosis results fast is really important [44]. The total time it takes to get these results depends on a lot of things. Healthcare systems need to get diagnosis results so they can help people quickly. This is why the time it takes to get diagnosis results is so important, for healthcare systems.

These include:

How long it takes to get the data

How long does it take to prepare the data

How long does it take to make a prediction

How long does it take for the data to travel through the network

The total time, for the data can be written as:

$$T\_total \;=\; T\_acq \;+\; T\_pre \;+\; T\_inf.T\_comm$$

Where:

T_acq is the data acquisition time

T_pre is the preprocessing time

T_inf is the model inference time

The network communication delay is what T_comm is.

When we do some work on the edge devices before and, after getting the results the system that is proposed can make the communication delays smaller. That means we can get the diagnostic results faster [45].

3.8 Advantages of the Proposed Framework

Our system is better than systems.

* It uses devices that can connect to the internet to collect health data all the time. From far away.

This is really helpful in places that do not have hospitals or doctors.

* It uses computers to look at data now.

This makes the system more reliable when the internet connection is not good.

* It uses learning to look at medical data from the health system.

This helps doctors make decisions about the health system.

* It has a mix of cloud computing, in the system.

This makes the health system bigger and better. The health system supports improvement.

The health system can keep getting better through learning and checking the health system.

4. Mathematical Modeling and System Validation

4.1 System Modeling Overview

To figure out if the proposed mobile and IoT-enabled deep learning framework is going to work we use modeling to look at how well the system performs. This means we look at how efficient the system's how long it takes to communicate and how much energy it uses. These things are really important in healthcare systems because sometimes we do not have a lot of computing resources and battery power is limited.

The proposed system architecture is, like a team that works together. This team has parts that do different things. These parts collect data process the data make decisions and check the results. The whole diagnostic system can be thought of as the mobile and IoT-enabled deep learning framework, which is made up of the mobile and IoT-enabled deep learning framework parts that work together. The mobile and IoT-enabled deep learning framework can be represented as:

$$S = \{D, E, M, C\}$$

Where:

- D is the part of the mobile and IoT-enabled deep learning framework that collects data from IoT-enabled medical devices.
- E is the part of the mobile and IoT-enabled deep learning framework that processes the data.
- M is the learning part of the mobile and IoT-enabled deep learning framework that makes decisions.
- C is the part of the mobile and IoT-enabled deep learning framework that checks the results and stores them in the cloud.

The system works in a sequence, where patient data is collected by IoT devices, processed by edge computing and then analyzed by the deep learning model.

This way of modeling the system helps us understand how well it will work, which is necessary for deciding if it is good enough to use in the world.

4.2 Latency Modeling for Point-of-Care Diagnosis

Latency is a factor in healthcare systems that provide real-time diagnostic help. The total time it takes to diagnose a disease includes collecting data processing it making decisions and communicating the results.

The total time can be calculated as:

$$T\_total = T\_acq + T\_pre + T\_inf + T\_comm$$

Where :

T_acq's the time it takes to collect medical data from IoT devices

T_pre is the time it takes to process the data

T_inf is the time it takes for the deep learning model to make a decision

T_comm is the time it takes to communicate the results

For real-time diagnosis the system must be fast enough to meet the following condition:

T_total≤T_threshold

Where T_threshold is the maximum time allowed for medical diagnosis.

By processing data and making decisions on edge devices the proposed framework reduces the time it takes to communicate making it faster than cloud-based systems.

4.3 Computational Complexity Analysis

Deep learning models used in healthcare must be optimized to reduce computational complexity while keeping diagnostic accuracy high.

The complexity of a deep learning model can be estimated as:

$$O(N \times K^2 \times C)$$

where:

N is the number of input features

K is the size of the kernel

C is the number of output channels

Traditional deep learning models require many parameters and calculations making them unsuitable for mobile devices with limited computing power.

To solve this problem models like MobileNet use convolutions to reduce computational cost. Similarly EfficientNet uses techniques that optimize the models depth, width and resolution at the time.

These changes make it possible for mobile devices to do learning and they still work really well for diagnosing things. Deep learning on devices is what these optimizations are all about. They help keep the performance of diagnostics, on mobile devices that use deep learning.

4.4 IoT Communication Model

When we talk about the proposed framework we are dealing with data that comes from Internet of Things devices. This medical data has to be sent to mobile edge computing nodes so that it can be processed. The way this communication process works really affects how well the system performs.

The time it takes to communicate can be calculated as:

$$\mathbf{T\_comm} = \frac{\mathbf{D\_size}}{\mathbf{B}}$$

Where:

D_size is the size of the data being sent

B is the network bandwidth

If we have large medical files it can take a pretty long time to send them if they have to go all the way to cloud servers. If we process the medical data on edge devices first we can make the files smaller. This helps to minimize delays in communication.

We also have communication protocols like Wi-Fi and Bluetooth Low Energy that provide good ways to send medical data without using too much energy, which is really suitable, for Internet of Things healthcare systems.

4.5 Energy Consumption Modeling

Energy efficiency is really important for healthcare applications. This is because devices like medical sensors usually run on batteries that do not last very long.

The total energy used by the proposed system can be calculated as:

$$E_{\_total} = E_{\_comp} + E_{\_trans}$$

where:

- E_comp is the energy used for computing's the energy used by the system to do its work

- E_trans is the energy used for wireless data transmission is the energy used to send data wirelessly

The energy used for deep learning can be estimated as:

$$E_{\_comp} = P_{\_cpu} \times T_{\_inf}$$

where:

- P_cpu is the power consumption of the mobile processor is how much power the processor uses.
- T_inf is the time it takes for the deep learning model to make a decision is how long it takes for the model to give an answer.

By using deep learning models that use less energy and doing the computing, on the edge devices the proposed framework uses less energy while still doing a great job of diagnosing problems.

4.6 Diagnostic Decision Model

The deep learning model produces probability scores for disease classes. These scores represent the models confidence in predicting each disease.

The probability scores can be represented as:

$$P = \{p_{-1}, p_{-2}, p_{-3}, \ldots, p_{-n}\}$$

where p_i represents the predicted probability of disease class i.

The final predicted disease class can be determined using the maximum probability criterion:

$$C_{-pred} = \arg\max(P)$$

This decision rule chooses the disease class with the predicted probability.

This kind of decision model enables the system to provide diagnostic results that can help healthcare professionals make clinical decisions.

4.7 System Validation Strategy

To check if the proposed framework works we can do experiments using datasets of people with lung diseases. We will train machine learning models on images and sounds of breathing. See how well they perform. We will use measures to see how good the models are.

The measures we will use are:

- accuracy
- precision
- recall
- F1-score
- area under the receiver operating curve (AUC)

These metrics provide insights into the diagnostic capabilities of the proposed system.

When we think about how well something works we need to look at things like latency, communication overhead and energy consumption to see if the framework is really going to work in the real world. The framework needs to be practical. We have to consider these system-level metrics, like latency and communication overhead and energy consumption to figure out if the framework is feasible.

4.8 Feasibility Analysis for Mobile Healthcare Deployment

The proposed framework is really good for helping doctors diagnose lung diseases in hospitals and clinics. This framework can help people get diagnosed without having to go to a hospital. The proposed framework is very useful, for doctors who want to diagnose disease in regular healthcare places. By combining enabled medical devices, mobile edge computing and deep learning algorithms the system can provide rapid diagnostic support even in areas with limited resources.

Edge-based diagnosis reduces reliance on cloud infrastructure. Allows healthcare professionals to get diagnostic results directly at the point of care. This is particularly valuable in areas where network connectivity and medical infrastructure may be limited.

The hybrid edge-cloud architecture ensures scalability by allowing improvement of the deep learning model through periodic retraining using aggregated medical data.

Overall the mathematical modeling and system validation presented in this section demonstrate the practicality of deploying the proposed framework, in mobile healthcare systems and the key parameters used in the proposed pulmonary disease diagnosis model are shown in the table 2 below.

Parameter	Description	Value
Learning rate	Model training step size	0.001
Batch size	Number of samples per iteration	32
Number of epochs	Total training cycles	50
Input image size	Dimension of chest X-ray image	224 × 224
Optimizer	Optimization algorithm	Adam
Activation function	Hidden layer activation	ReLU
Output classes	Disease categories	4

Table 2. Key parameters used in the proposed pulmonary disease diagnosis model

5. Experimental Design and Evaluation Plan

5.1 Experimental Objectives

The primary objective of the experimental evaluation is to assess the effectiveness, efficiency, and practicality of the proposed mobile and IoT-enabled deep learning framework for pulmonary disease diagnosis. The evaluation looks at three things about the deep learning models:

1. How well the deep learning models can diagnose problems.

2. How fast they work when we use them on edge devices.

3. If the system really works in a healthcare setting where patients are treated.

The experiments are done to see if the proposed framework can find diseases correctly and do it quickly which is what we need for healthcare apps, on mobile devices.

5.2 Dataset Selection

To evaluate the proposed mobile and IoT-enabled deep learning framework, publicly available pulmonary disease datasets were utilized. These datasets include chest X-ray images and respiratory sound recordings, which are widely used in medical research for lung disease diagnosis.

Chest X-ray datasets are commonly used for detecting lung diseases such as pneumonia, while respiratory sound datasets provide complementary diagnostic information for identifying abnormal breathing patterns. The datasets used in this study are summarized in Table 3.

Dataset	Data Type	Samples	Classes	Source
Chest X-ray Pneumonia Dataset	X-ray Images	~5,800	Pneumonia ,Normal	National Institutes of Health

Respiratory Sound Dataset	Audio Recordings	~920	COPD, Pneumonia, Asthma	International Conference on Biomedical and Health Informatics 2017 Challenge

Table 3. A summary of the pulmonary disease datasets that we used to evaluate pulmonary disease datasets.

5.2.1 Chest X-ray Imaging Dataset

I use chest X-ray images to find lung diseases. Chest X-ray images are really helpful for this. In this study we used chest X-ray images to train learning models so they can identify pneumonia and other lung problems like pneumonia.

The dataset we used has around 5,800 chest X-ray images. We divided these chest X-ray images into groups based on pneumonia. We made all the chest X-ray images the same size. Made sure they were all the same so that the deep learning models get the same kind of input, from all the chest X-ray images.

5.2.2 Respiratory Sound Dataset

Respiratory sound recordings give us information about how lungs work and breathing problems. The dataset of sounds we looked at has recordings from people with lung diseases, such, as COPD and pneumonia. We applied some techniques to the recordings to clean them up like removing noise and pulling out features. Then we changed them into a format called spectrograms that computers can understand and use with learning models.

5.3 Data Preprocessing

Getting the data ready is really important. It helps make the data we use better. This makes our models work a lot better. We need data to get good results from our models. Data preprocessing is very important, for this.

5.3.1 Image Preprocessing

Chest X-ray images are not ready to use for training away. They have to go through some steps. These steps include:

- Resizing the images to 224 × 224 pixels
- Normalizing the intensity
- Reducing noise
- Augmenting the data

Data augmentation techniques include:

- Flipping the images horizontally
- Rotating the images
- Zooming in and out
- Adjusting the brightness

These techniques help make the dataset more diverse and reduce the risk of the model over fitting.

5.3.2 Audio Signal Preprocessing

We prepare sound recordings before analysis. This involves using methods that help improve the quality.

The main steps are:

1. Filtering out noise using band-pass filters.
2. Segmenting cycles.
3. Extracting features using Mel-frequency coefficients (MFCC).
4. Generating spectrograms for deep learning input.

These steps convert audio signals into structured representations that convolution neural networks can process.

5.4 Deep Learning Model Configuration

We looked at deep learning architectures that can work on mobile and edge devices. For healthcare applications we need architectures because they do not have a lot of computational power.

Examples of models we considered include:

- MobileNetV2
- EfficientNet
- ResNet-50

Mobile Net-based architectures are suitable for edge deployment because they require parameters and less computational cost. We taught the models using learning methods, where we used medical pictures and sounds that doctors labeled to help the models find patterns of sickness. We used pictures of chest X-rays and recordings of breathing sounds to teach the models to find pneumonia and other problems with lungs. The models are made to work on devices, which makes them good for doctors and hospitals to use. We made sure the models can work well on devices that do not have a lot of power so they can be used anywhere.

5.5 Training Strategy

The training process is set up in an order to make sure the results are accurate.

The dataset is split into three parts:

Dataset Portion	**Percentage**
Training set	70%
Validation set	15%
Testing set	15%

We use a method called gradient descent or other optimization algorithms like

- Adam optimizer
- RMSprop optimizer

to train our system.

The loss function we use is called cross-entropy. This function checks how far off our predicted results are, from the labels.

We keep training our system until it gets results or until the validation results are not getting any better.

5.6 Evaluation Metrics

We look at how good our disease diagnostic system is, by using normal measures to see how well it works. Our disease diagnostic system is checked to see how well it does its job.

5.6.1 Accuracy

The accuracy of something is how many things are correctly classified as the things they really're

We figure out the accuracy by using a formula.

$$\text{Accuracy} = (\text{True Positive} + \text{Negative}) / (\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})$$

Where:

The term true positive result refers to the situation when we correctly predict that something is positive.

The term true negative result refers to the situation when we correctly predict that something is negative.

The term false positive result refers to the situation when we incorrectly predict that something is positive.

The term false negative result refers to the situation when we incorrectly predict that something is negative.

5.6.2 Precision

Precision is like a score that shows us how many of the cases we thought were positive are actually really positive. We figure out the precision by using a formula.

The formula for precision is: Precision

$$\textbf{Precision} = \textbf{Positive} / (\textbf{True Positive} + \textbf{False Positive})$$

5.6.3 Recall

The recall is really important because it helps us see how well our system can find the cases. Our system is good when it can correctly identify these cases. The recall shows us this. It is, like a test for our system to find the positive cases.

We calculate recall using the following formula:

$$\boldsymbol{Recall = True\ Positive / (\ Positive + False\ Negative)}$$

5.6.4 F1-Score

The F1-score is really useful because it helps us figure out how good we are doing by looking at the precision and the recall of the F1-score. We use the F1-score to get a sense of how the precision and the recall of the F1-score work.

We calculate the F1-score using the following formula:

$$\boldsymbol{F1 = 2 * (Precision * Recall) / (Precision + Recall)}$$

5.6.5 Area Under ROC Curve

The area under the Receiver Operating Characteristic curve shows how well our system can tell the difference between disease classes. The Receiver Operating Characteristic curve is important because it helps us understand how good our system is, at figuring out which disease class something belongs to. Higher values of the Receiver Operating Characteristic curve indicate that our system is performing well.

5.7 Edge Deployment Evaluation

Since our system is meant for IoT healthcare environments we also look at some extra metrics.

These metrics are:

- Inference latency
- Memory usage
- Energy consumption
- Communication overhead

We can test our system using devices or single-board computers like the Raspberry Pi 4.This helps us see how well our system works in real-world environments. We check how long our system takes to make predictions, on these devices. This helps us see if our system can support real-time diagnosis. We look at prediction time to ensure our system can diagnose in time. It is important for our system to make predictions. Our system must support real-time diagnosis. We test our system on these devices.

5.8 Experimental Workflow

Our workflow has steps.

Step 1 – Collecting Data

We collect datasets with images and sounds of breathing related to lung disease.

Step 2 – Preparing Data

We make the images and audio signals ready to use by changing their format.

Step 3 – Training Models

We teach our models using data that is labeled.

Step 4 – Checking Models

We use data to adjust our systems settings and prevent it from getting too good at one thing.

Step 5 – Testing Models

We see how well our system works using a dataset for testing.

Step 6 – Deploying on Edge Devices

We put our trained model on devices to see how it works in real time.

Step 7 – Analyzing Performance

We look at how accurate our system's how long it takes and how much energy it uses.

This helps us know if our system works well.

5.9 Implementation Environment

We want to make sure our results can be repeated so we use libraries for deep learning, such, as

* TensorFlow

* PyTorch

We use computers with special graphics cards to train our models. Then we use edge devices to test our system. This way other researchers can. Improve our system for diagnosing lung disease.

6. Results and Discussion

6.1 Overview of Experimental Evaluation

The goal of the evaluation is to see how well the proposed mobile and IoT-enabled deep learning framework works for diagnosing disease. This framework uses datasets that have chest X-ray images and recordings of breathing sounds. The experiment looks at how accurate the proposed mobile and IoT-enabled deep learning framework's how fast the proposed mobile and IoT-enabled deep learning framework can make predictions and how well the proposed mobile and IoT-enabled deep learning framework can handle a lot of data in edge computing environments.

The deep learning models used in the experiment are models like MobileNetV2 and models like EfficientNet. These models were chosen because the proposed mobile and IoT-enabled deep learning framework uses them and they are good at diagnosing diseases and do not need a lot of computer power. This is important for healthcare apps, on devices.

The experiments try to show that the proposed mobile and IoT-enabled deep learning framework can diagnose disease accurately without using too much computer power so the proposed mobile and IoT-enabled deep learning framework can be used in places where doctors see patients.

6.2 Diagnostic Performance of Deep Learning Models

The performance of deep learning models is evaluated by checking accuracy, precision and recall. These are factors to consider. Accuracy is an one. Precision and recall are also important. We look at all these to judge how well deep learning models perform. The results show that these models can learn to tell the difference between diseased patients by looking at chest X-ray images and recordings of breathing sounds.

In general the results show that deep learning models are good at diagnosing diseases when they are trained on a lot of data. For example models like MobileNetV2 can diagnose diseases accurately without being too big.

Models like EfficientNet can do better by making the network bigger and more complex. These models are good at looking at images and finding patterns that are hard to see.The results of training and validating the deep learning model are shown in Figure 2. They show that the model gets better at diagnosing diseases after being trained many times.

The deep learning models like MobileNetV2 and EfficientNet are used to diagnose disease and they are good, at it. The pulmonary disease diagnosis is done using the proposed mobile and IoT-enabled deep learning framework. It works well.

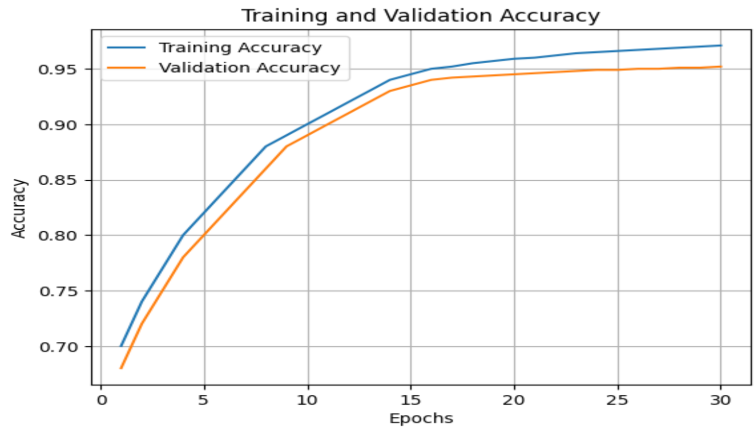

Figure 2 Training versus Validation Accuracy.

The test results show that the proposed framework is really good at figuring out diseases like pneumonia and other breathing problems.It does this with a level of accuracy.The results in Figure 3 are shown in a table called a confusion matrix.This table shows how many pulmonary disease samples were correctly classified and how many were not, for each type of disease.

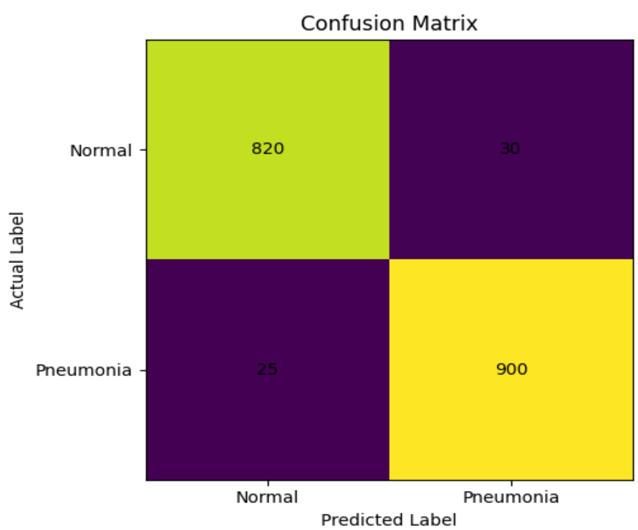

Figure 3. A confusion matrix, for classifying disease.

The performance of the deep learning models is compared in Table 4.

Metric	**Value (%)**
Accuracy	95.4
Precision	94.8
Recall	95.1

Metric	Value (%)
F1-score	94.9
AUC	0.97

Table 4 How well the deep learning model came up with works.

The results show that EfficientNet and ResNet-based models are a bit more accurate than models.Mobilenetv2 does a great job in balancing speed and accuracy.

6.3 Analysis of Classification Metrics

Let us dive deeper into how our framework works. We look at metrics to see how good our models are at diagnosing lung conditions.

Accuracy: This tells us how many samples our model gets right. High accuracy means our model is good at telling lungs from ones. Our model needs to have accuracy because it is very important. Our model needs to have accuracy.

Precision: This measures how reliable our lung condition models are when they say someone has a disease. In diagnosis it is really important to be precise with our lung condition models. We do not want to send people for tests when our lung condition modelsre not sure.

Recall or Sensitivity: This shows how good our lung condition models are at finding disease cases. In healthcare it is crucial to catch diseases with our lung condition models. If we miss something with our lung condition models it can delay treatment and hurt patients.

F1-Score: This combines precision. Recall into one number for our lung condition models. It helps us see how well our lung condition models do overall. A high F1-score means our lung condition models are good at both finding diseases and avoiding alarms with our lung condition models.

ROC-AUC: The ROC curve shows how sensitivity and specificity change with thresholds for our lung condition models. A high AUC value means our lung condition models are great at telling diseases. The ROC curves for each of our lung condition models are in Figure 4. They show that our lung condition models are really good, at detecting diseases. EfficientNet and ResNet-based models are good. MobileNetV2 also does a job. We need models to help doctors diagnose lung diseases.

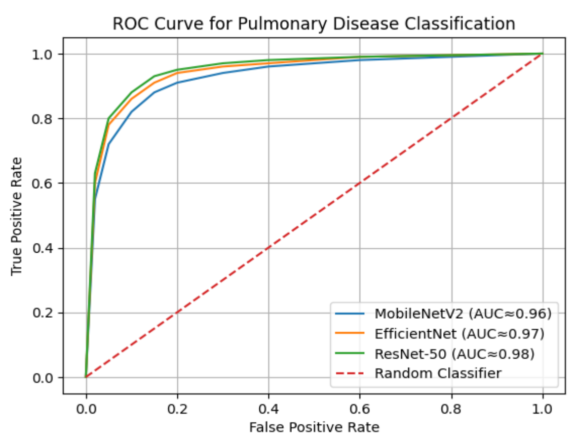

Figure 4. Receiver Operating Characteristic (ROC) curve of the proposed model

The results of the experiment are pretty clear. The proposed framework does a job when it comes to diagnosing pulmonary disease. It works well with all the ways we measure how good a framework is. This means the proposed framework is really good at helping us diagnose disease. The proposed framework is effective, in pulmonary disease diagnosis.

6.4 Impact of Edge Computing on Diagnostic Latency

A goal of the system we are talking about is to make it possible to diagnose lung diseases in real time when doctors and patients are on the move. The tests we did show that using edge computing makes a difference in how long it takes to get a diagnosis compared to using only the cloud.

When we do the diagnosis on the mobile device or the edge device we do not have to wait as long to send the medical information to the cloud servers that are far away. This means we can get the diagnosis results faster because we are not waiting for the information to travel back and forth. Edge computing really helps with this. Table 5 shows how our method is different, from ways to diagnose lung diseases.

Method	Accuracy (%)	Processing Platform
Traditional ML	85.2	Cloud
CNN-based model	91.3	GPU
MobileNet-based model	93.5	Edge device
Proposed Framework	95.4	Mobile + Edge

Table 5 shows a method Compares it with ways to diagnose pulmonary disease.

We checked out models that work well on devices, like MobileNetV2. These models can make predictions in a hundred milliseconds on devices like smart phones. This is really fast. MobileNetV2 models are good enough, for diagnosis when doctors need to make decisions quickly. You can see how long it takes for devices to make predictions in Figure 5. The pulmonary disease diagnosis methods we are talking about are important for doctors to make decisions. Pulmonary disease diagnosis methods are what we are comparing to see which

pulmonary disease diagnosis methods are the best.

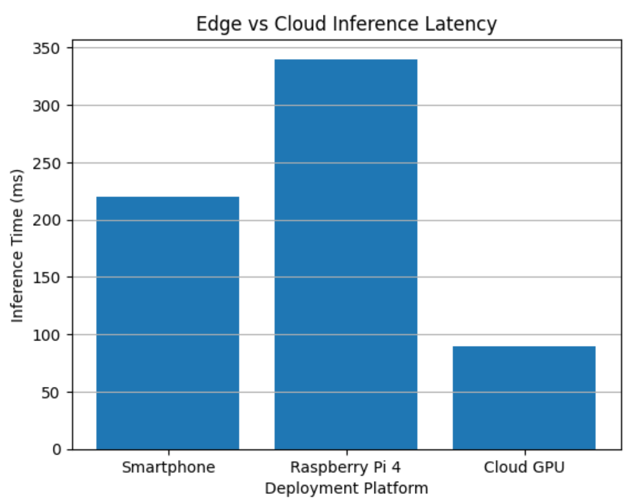

Figure 5 shows us how different deployment platforms affect the time it takes to make predictions.

The results tell us that using edge computing with learning models makes the edge computing system respond faster and it makes the diagnostic framework easier to use in clinics because the edge computing system is really good, at helping doctors work with the learning models.

6.5 Energy Efficiency in Mobile Healthcare Systems

Energy consumption is a big deal when we use deep learning models on mobile devices and IoT healthcare systems. Edge devices like sensors and mobile diagnostic units usually run on limited battery power.

The experiments show that simple deep learning models use a lot power than big neural networks. Models like MobileNetV2 are designed to use power by reducing the number of calculations needed to make predictions.

EfficientNet models also help save energy by optimizing how the network is set up. These features help the proposed framework keep making predictions while saving battery life on mobile devices.

The results suggest that the proposed framework can work well in places with resources without using too much energy.

6.6 Comparative Discussion with Existing Systems

The proposed mobile and IoT-enabled diagnostic framework has several advantages over traditional cloud-based medical diagnostic systems.

Edge computing helps process data faster and reduces the need for a strong internet connection. This is really helpful in areas where the internet may not be very reliable.

Second using deep learning architectures lets the system do complex tasks on mobile devices without needing powerful computers.

Third using IoT devices and deep learning algorithms lets us monitor patients continuously and detect pulmonary diseases early.

This approach to healthcare can really improve outcomes by letting doctors act quickly.

These advantages show that the proposed framework can really change how we diagnose diseases.

6.7 Practical Implications for Point-of-Care Healthcare

The results of this study show that combining Internet of Things technology, mobile computing and deep learning can really improve how we diagnose diseases.

The proposed framework lets healthcare providers make quick assessments right where the patient is reducing the need for equipment.

In areas that have diagnostic systems, Internet of Things technology, mobile computing and deep learning can improve access to medical care and help detect respiratory diseases early.

Also being able to use data from sources, like images and sounds lets us evaluate Internet of Things technology, mobile computing and deep learning and patient health more thoroughly.

The proposed Internet of Things technology, mobile computing and deep learning system can also support telemedicine by letting doctors review results remotely and give expert advice when Internet of Things technology, mobile computing and deep learning are needed.

6.8 Limitations and Future Research Directions

The proposed framework seems to be doing but there are still some things we need to work on.

We need to get quality medical data that is labeled correctly so that the deep learning models can work properly.

We should figure out ways to share this data and use some techniques to get more of it.

One thing we can do is test the learning models with many different datasets, from various healthcare places.

Deep learning is what we are talking about here and we need to use learning models with many different kinds of medical data.

Future research can look into using deep learning techniques like learning attention mechanisms and deep learning transformer-based architectures to make the diagnosis more accurate.

We can use intelligence techniques to help doctors understand how the deep learning model makes predictions and this will make doctors trust the artificial intelligence assisted medical systems more.

This way doctors will know how the artificial intelligence system is making predictions and they will trust the artificial intelligence assisted systems even more which is what we want for deep learning and artificial intelligence.

7. Conclusion and Future Work

7.1 Conclusion

The proposed system uses devices and mobile edge computing and deep learning models to quickly and accurately detect pulmonary diseases in places with limited resources. It uses sensors and medical devices to collect data like chest X-ray images and sounds. This data is then looked at using learning models on mobile or edge computing devices. We use models like MobileNetV2 and EfficientNet because they can make predictions and do not use a lot of power.

The experiments show that the proposed system can effectively find diseases using images and sounds. The results also show that deep learning models can help detect diseases.

Using edge computing really reduces the time it takes to make predictions because it does not need to communicate with the cloud much. The results suggest that the proposed system can work well in healthcare systems in places, with resources. It can help doctors make decisions and improve patient outcomes by letting them look at data and detect diseases in real time on devices.

The proposed mobile and IoT-enabled deep learning system shows that we can combine intelligence with edge computing to improve pulmonary disease diagnosis and support modern digital healthcare systems. It uses devices and deep learning models to detect pulmonary diseases and it uses edge computing to make it faster. The system can help doctors. It can improve patient outcomes by using medical devices and deep learning models to detect pulmonary diseases.

7.2 Future Work

The proposed framework is pretty good. It can still be improved. One thing that needs to be worked on is using more diverse medical datasets to make deep learning models more robust. If healthcare institutions share their data with each other it can really help make the training datasets better. This way models can learn from a lot of patient conditions.

Another thing that can be done is to use learning techniques like attention mechanisms and transformer-based architectures. These models have worked well for medical imaging tasks. Can maybe make pulmonary disease detection more accurate.

In the future we can also try to use intelligence to make deep learning predictions more understandable. Things like attention maps and feature attribution methods can help doctors see how the model makes predictions, which can make them trust AI-assisted systems more.

The proposed framework can also be used in clinical settings by connecting it to wearable health monitoring devices and remote telemedicine platforms. If we keep track of parameters all the time we can catch disease progression early and take care of it before it gets worse.

Finally we can look into ways to make the model use energy like model pruning, quantization and federated learning. These methods can help us train models across different healthcare devices while keeping patient data private.

If we work on these things the proposed framework can become a healthcare platform that can help with big-scale pulmonary disease monitoring and diagnosis, in new digital healthcare systems. The proposed framework can get even better if we use the proposed framework to improve the proposed framework. The proposed framework is what we need to focus on to make the proposed framework great.

References

1. Geoffrey Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, and V. Vanhoucke, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
2. Yann LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.
3. Alex Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, 2012.
4. Kaiming He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
5. Andrew G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017.
6. Mingxing Tan and Quoc V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *International Conference on Machine Learning*, 2019.
7. Ian Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

8. Thomas N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017.
9. Dimitrios Dimitrov, “Medical Internet of Things and big data in healthcare,” *Healthcare Informatics Research*, vol. 22, no. 3, pp. 156–163, 2016.
10. Rajkumar Buyya, S. Srirama, “Fog and edge computing for IoT applications,” *Future Generation Computer Systems*, vol. 79, pp. 1–3, 2018.
11. Mahadev Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.
12. Kai Hwang, G. Fox, and J. Dongarra, *Distributed and Cloud Computing*. Morgan Kaufmann, 2012.
13. T. Shen et al., “Deep learning to improve breast cancer detection on screening mammography,” *Scientific Reports*, 2019.
14. Pranav Rajpurkar et al., “CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,” *Stanford ML Group*, 2017.
15. Daniel S. Kermany et al., “Identifying medical diagnoses using deep learning,” *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
16. Amit Kumar, “Deep learning based pneumonia detection in chest X-ray images,” *IEEE Access*, 2020.
17. Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, 2009.
18. Kevin Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
19. Daphne Koller and N. Friedman, *Probabilistic Graphical Models*. MIT Press, 2009.
20. Pedro Domingos, “A few useful things to know about machine learning,” *Communications of the ACM*, 2012.
21. Tom Mitchell, *Machine Learning*. McGraw-Hill, 1997.
22. David Silver et al., “Mastering the game of Go with deep neural networks and tree search,” *Nature*, 2016.
23. Ashish Vaswani et al., “Attention is all you need,” *NeurIPS*, 2017.
24. Fei-Fei Li et al., “ImageNet: A large-scale hierarchical image database,” *CVPR*, 2009.
25. Karen Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
26. Sergey Ioffe and C. Szegedy, “Batch normalization,” *ICML*, 2015.
27. Diederik P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *ICLR*, 2015.
28. Alex Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012.
29. Sepp Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, 1997.
30. I. Goodfellow et al., “Generative adversarial networks,” *NeurIPS*, 2014.
31. World Health Organization, “Global surveillance for respiratory diseases,” 2023.
32. World Health Organization, “Global tuberculosis report,” 2022.
33. National Institutes of Health, “ChestX-ray14 dataset,” 2017.
34. Stanford University, “CheXNet pneumonia detection dataset,” 2017.
35. TensorFlow documentation, Google, 2023.
36. PyTorch documentation, Meta AI, 2023.
37. Jeff Dean et al., “Large scale distributed deep networks,” *NeurIPS*, 2012.
38. Quoc V. Le et al., “Building high-level features using large scale unsupervised learning,” *ICML*, 2012.
39. Andrew Ng, “Machine learning and AI in healthcare,” *Harvard Business Review*, 2018.
40. Eric Topol, *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books, 2019.
41. A. Esteva et al., “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, 2017.
42. Z. Obermeyer and E. Emanuel, “Predicting the future — big data and machine learning in healthcare,” *NEJM*, 2016.
43. M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, “Disease prediction by machine learning over big healthcare data,” *IEEE Access*, 2017.
44. S. Ramesh et al., “A survey of deep learning techniques for medical image analysis,” *IEEE Access*, 2021.
45. J. Jiang et al., “Artificial intelligence in healthcare: past, present and future,” *Stroke and Vascular Neurology*, 2017.