# Medical Image Recognition For Diagnosis

## COMMUNITY SERVICE PROJECT

*Submitted by*


**CH. GIREESH (99210041425)**

**B. RAKESH (99210041019)**

**A. ABHIRAM (99210041138)**

**B. KARTHIK (99210041455)**


*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**


**SCHOOL OF COMPUTING**

**COMPUTER SCIENCE AND ENGINEERING**

**KALASALINGAM ACADEMY OF RESEARCH**

**AND EDUCATION**

**KRISHNANKOIL 626 126**


Academic Year 2023-2024

# DECLARATION

We affirm that the project work titled **"Visualizing Health: Advancements in Medical Image Recognition for Accurate Diagnoses."** being submitted in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.


**CH. GIREESH**

(99210041425)

**B. RAKESH**

(99210041019)

**A.ABHIRAM**

(99210041138)

**B.KARTHIK**

(99210041455)

KALASALINGAM
ACADEMY OF RESEARCH & EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

# BONAFIDE CERTIFICATE

Certified that this project report **"Visualizing Health: Advancements in Medical Image Recognition for Accurate Diagnoses."** is the bonafide work of "**CH. GIREESH, B. RAKESH, A. ABHIRAM, B. KARTHIK "** who carried out the project work under my supervision.

**SUPERVISOR**

**Dr.V.Sathya Narayanan**
Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankoil 626126

**HEAD OF THE DEPARTMENT**

**Dr. N. Suresh Kumar**
Professor & Head
Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankoil 626126

Submitted for the Project Viva-voce examination held on......................................

\

**Supervisor** **Faculty Advisor** **External Examiner (s)**

# ACKNOWLEDGEMENT

First and foremost, I wish to thank the **Almighty God** for his grace and benediction to complete this Project work successfully. I would like to convey my special thanks from the bottom of my heart to my dear **Parents** and affectionate **Family members** for their honest support for the completion of this Project work.

I express deep sense of gratitude to "Kalvivallal" Thiru. **T. Kalasalingam** B.com., Founder Chairman, "Ilayavallal" **Dr. K. Sridharan,** Ph.D., Chancellor, **Dr. S. Shasi Anand**, Ph.D., Vice President (Academic), **Mr. S. Arjun Kalasalingam** M.S., Vice President (Administration), **Dr. S. Narayanan**, Vice-Chancellor, **Dr. V. Vasudevan,** Ph.D., Registrar, **Dr. P. Deepalakshmi,** Ph.D., Dean (School of Computing), **Dr. N. Suresh Kumar**, Professor & Head, Department of CSE, Kalasalingam Academy of Research and Education for granting the permission and providing necessary facilities to carry out Project work.

I would like to express my special appreciation and profound thanks to my enthusiastic Project Supervisor **Dr.V.Sathya Narayanan**, Assistant Professor/CSE of Kalasalingam Academy of Research and Education (KARE) for his inspiring guidance, constant encouragement with my work during all stages. I am extremely glad that I had a chance to do my Project under my Guide, who truly practices and appreciates deep thinking.

I will be forever indebted to my Faculty Advisor '**Mr. D. Bala Krishnan** for all the time. He gave me the moral support and the freedom I needed to move on.

Besides my Project guide, I would like to thank the committee members, all faculty members and non-teaching staff for their insightful comments and encouragement. Finally, but by no means least, thanks go to all my school and college teachers, well wishers, friends for almost unbelievable support.

KALASALINGAM
ACADEMY OF RESEARCH & EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

## School of Computing

## Department of Computer Science and Engineering

## Project Summary

| | | |
|---|---|---|
| Project Title | Visualizing Health: Advancements in Medical Image Recognition for Accurate Diagnoses | |
| Project Team Members (Name with Register No) | 1. CH. GIREESH (99210041425)<br>2. B. RAKESH (99210041019)<br>3. A. ABHIRAM (99210041138)<br>4. B. KARTHIK (99210041455) | |
| Guide Name/Designation | Dr.V.Sathya Narayanan, Assistant Professor, Department of Computer Science and Engineering | |
| Program Concentration Area | Medical Image Recognition For Diagnosis | |
| Technical Requirements | Android Studio is used by the developer to complete the project. | |
| Engineering standards and realistic constraints in these areas: (Refer Appendix in page 4 of this doc.) | | |
| **Area** | **Codes & Standards / Realistic Constraints** | **Tick ✓** |
| Economic | | |
| Environmental | | |
| Social | | |
| Ethical | | |
| Health and Safety | This project is mainly used to get awreness about disease in prior | ✓ |

| | | |
|---|---|---|
| Manufacturability | Reuse the code for future development | ✓ |
| Sustainability | | |

**Realistic Constraints:**

**Health and Safety:**

In order to protect patients and healthcare workers alike, medical image recognition systems must incorporate health and safety protocols. In order to safeguard patient information, data security and privacy must be preserved first and foremost, adhering closely to laws like HIPAA. It is important to have procedures for quality control and validation in place to reduce the possibility of inaccurate treatment recommendations or misdiagnoses. To get precise and trustworthy findings, imaging equipment needs to be calibrated and maintained on a regular basis. Additionally, to identify any drift or decline in performance, the AI systems must be continuously trained and monitored.

**Manufacturability:**

A medical image recognition system that can be manufactured should be able to process enormous amounts of data, be compatible with many types of imaging, and adhere to the strict quality and safety standards that are necessary in the medical industry. Its successful adoption and widespread use in clinical settings also depend on its ease of use and smooth integration with various medical devices and information systems.It includes factors including the technology's scalability, ease of interaction with current medical systems, cost-effectiveness, and compliance with regulations.

# ABSTRACT

Using Android Studio for medical image recognition is a cutting-edge field that improves patient care and diagnostics by utilizing mobile technologies. The goal of this research project is to create a reliable and intuitive Android application that has high accuracy for analyzing and interpreting medical pictures, including MRIs, CT scans, and X-rays. Convolutional neural networks (CNNs) are among the deep learning algorithms that the system will use to identify anomalies and abnormalities in the photos. This will help healthcare professionals make important decisions by detecting anomalies and abnormalities in the images. The program will include an easy-to-use user interface that allows for quick diagnosis feedback, real-time processing, and picture input. This cutting-edge Android-based medical picture recognition application has the potential to transform healthcare delivery by speeding up diagnosis and improving accuracy. The ability to recognize medical images has become essential to contemporary healthcare, allowing for more precise diagnosis and treatment planning. An overview of current developments and difficulties in the field of medical image recognition is given in this paper. Automation systems that can help medical professionals interpret digital imaging modalities like MRIs, CT scans, X-rays, and ultrasounds are becoming more and more necessary as these machines become more common. The numerous methods and algorithms used in medical image recognition—such as deep learning, convolutional neural networks, and image segmentation—are covered in this review.

# CONTENTS

## LIST OF FIGURES



## LIST OF ABBREVIATIONS

| Abbreviation | Full form |
|---|---|
| **CNN** | Convolutional Neural Network |
| **SGD** | Stochastic gradient descent |
| **HIPAA** | Health Insurance Portability and Accountability Act |
| **CNTK** | Cognitive Toolkit |

# 1.INTRODUCTION

The creation of a mobile application that makes use of cutting-edge image processing and machine learning techniques to analyze and interpret medical images—such as X-rays, MRIs, CT scans, or ultrasounds—directly on Android devices is required for medical image identification using Android Studio. By giving patients and medical professionals alike quick and easy access to diagnostic help, this technology has the potential to completely transform the healthcare industry. Developers can use Android Studio to make portable, user-friendly apps that can access pre-existing image databases or take pictures with a device's camera. After that, these photos are processed using deep learning models that have been taught to recognize and categorize a variety of medical issues found in the photos, including tumors, fractures, and malformations. In order to facilitate medical decision-making, the application can provide real-time diagnosis, risk assessments, or therapy recommendations. Furthermore, incorporating secure cloud services can make data sharing, storing, and distant consultations easier, improving the application's overall usefulness and scalability. To put it briefly, medical image identification with Android Studio leverages mobile technology, artificial intelligence, and image analysis to improve patient care and healthcare outcomes by increasing accessibility, efficiency, and accuracy of medical diagnosis.

## 1.1.OVERVIEW

A state-of-the-art technological application that uses Android Studio for medical image recognition helps medical professionals diagnose and treat a wide range of medical conditions. Using cutting-edge machine learning and computer vision techniques, this novel approach combines the power of the well-known integrated development environment, Android Studio. This technology allows the direct analysis of medical images, such as X-rays, MRIs, and CT scans, on smartphones and tablets by integrating deep learning models, like convolutional neural networks (CNNs), with the Android platform. This could expedite diagnosis times, lessen the burden on medical staff, and increase access to healthcare in underserved or remote locations. This could expedite diagnosis times, lessen the burden on medical staff, and increase access to healthcare in underserved or remote locations. Additionally, it can help in the early detection of illnesses like cancer and neurological conditions, which can result in more efficient treatment. However, there are certain difficulties with this application, such as maintaining data security, adhering to regulations, and requiring high-quality datasets in order to train precise models. Android Studio's medical image recognition technology has enormous potential to transform the healthcare sector and enhance patient outcomes as technology develops.

## 1.2.CONVOLUTIONAL NEURAL NETWORK

An especially developed class of deep neural networks called Convolutional Neural Networks (CNNs) is used to process and analyze visual data, including images and videos. Because of their exceptional capacity to automatically extract meaningful features from input data, they have emerged as a key component of computer vision, image recognition, and other machine learning applications. Convolutional, pooling, fully connected, and frequently a final softmax layer for classification tasks are among the layers that make up a CNN. The convolutional layer, which applies a collection of learnable filters or kernels to the input data, is the primary innovation of CNNs. Feature maps are produced by sliding these filters over the input and applying element-wise summations and multiplications. These feature maps record various elements of the input, including textures, edges, and more intricate patterns. Convolutional layers are followed by pooling layers, which lower the spatial dimensions of feature maps and increase the computational efficiency and robustness of the network against slight input distortions and translations. For this, max-pooling or average-pooling are typically employed.Towards the end of the network, fully connected layers are used. They receive input in the form of flattened feature maps and are in charge of making final decisions like object recognition or classification. In a classification problem, the softmax layer is frequently used to generate probability scores for each class. CNNs learn hierarchical representations of the input data by using large datasets for training, and by adjusting their internal parameters (weights and biases) through gradient descent and backpropagation. With the growing popularity of transfer learning, which reduces the need for large datasets and computation, pre-trained CNNs are adjusted for particular tasks.With the ability to perform tasks like object detection, face recognition, medical image analysis, and autonomous driving, CNNs have completely changed the fields of image and video analysis. Their versatility and impact on different domains of artificial intelligence are demonstrated by their applications, which go beyond images to include sequential data and even natural language processing.

**Architecture of Convolution Neural Network**

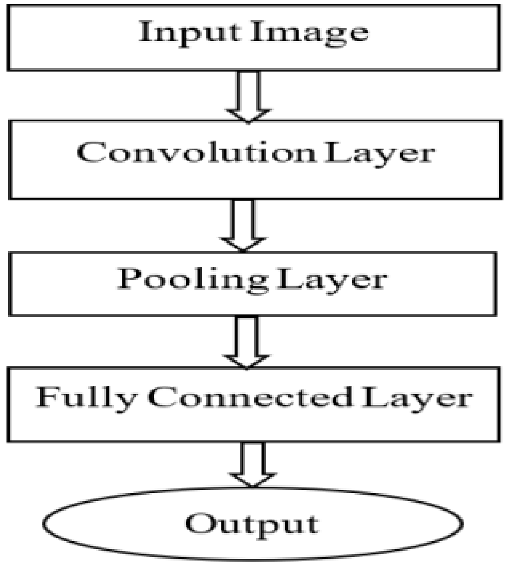


**Figure 1**

A CNN is a neural network that includes both convolutional and additional layers. Multiple filters make up a convolutional layer, which performs the convolutional operation. Convolutional layers are used with bi-dimensional inputs and have gained a lot of notoriety for their outstanding image categorization abilities. They are founded on the discrete convolution of a tiny kernel, K, and a bi-dimensional input, whereby the input may be the result 22 of another convolutional layer. Fig 3. shows the network architecture of CNN. The Basic Building Block of CNN is: It has five layers namely:

a) Input Layer

b) Convolution Layer

c) Activation Layer

d) Pooling/Sub-sampling Layer

e) Locally Connected Laye

a) Input Layer: The input layer is the first layer of a Convolutional Neural Network (CNN). Its main function is to receive the input image data and transform it into a format that can be processed by the subsequent layers of the network. The input layer typically consists of a grid of neurons that corresponds to the dimensions of the input image. Each neuron in the input layer corresponds to a single pixel in the input image and its value represents the intensity or color of the corresponding pixel. The input layer feeds the processed data into subsequent layers such as convolutional layers, pooling layers, and fully connected layers for feature extraction and prediction.

b) Convolution Layer: The convolution layer is a key component of a Convolutional Neural Network (CNN). Its main function is to extract features from the input image through a process of convolution. The convolution operation involves sliding a small matrix called a filter over the input image and computing the dot product between the filter and each corresponding region of the image. The output of this operation is a feature map that highlights specific patterns and features in the input image. Multiple filters can be used in a single convolution layer to extract different features. The output feature map is then passed on to subsequent layers for further processing and analysis. The convolution layer plays a critical role in the success of CNNs in tasks such as image recognition, object detection, and natural language processing.

c) Activation Layer: The activation layer is a key component of a Convolutional Neural Network (CNN). Its main function is to introduce nonlinearity into the network, which allows it to learn complex patterns and relationships in the input data. The activation layer applies a nonlinear activation function to the output of the previous layer, which determines whether a neuron should be activated or not based on its input. The most commonly used activation functions include ReLU (Rectified Linear Unit), Sigmoid, and Tanh. ReLU is the most popular 23 choice due to its simplicity and effectiveness. The activation layer is typically inserted after the convolutional and fully connected layers to improve the network's performance. Without the activation layer, the network would be limited to linear relationships and unable to model the complex nonlinearities present in most real-world data.

d) Pooling/Sub-sampling Layer: The pooling/sub-sampling layer is a key component of a Convolutional Neural Network (CNN). Its main function is to reduce the spatial size of the feature maps and extract the most important features from them. The pooling operation involves dividing the input feature map into non-overlapping regions and computing a single output value for each region. The most commonly used pooling operations are max pooling and average pooling, which respectively return the maximum and average value of each region. The pooling layer helps to reduce the dimensionality of the feature maps, which reduces the computational complexity of the network and helps to prevent overfitting. It also helps to make the network more robust to small shifts and variations in the input data. The pooling layer is typically inserted after the convolutional layers and before the fully connected layers in the network.

e) Locally Connected Layer: The locally connected layer is a type of layer in a Convolutional Neural Network (CNN) that shares some similarities with the convolutional layer. While the convolutional layer applies the same filters to the entire input image, the locally connected layer applies different filters to different regions of the image. Specifically, each filter is only connected to a subset of the input image rather than the entire image. This can help to capture more fine-grained local patterns and variations in the input data. The locally connected layer is typically used in situations where the input data has a strong spatial structure or where the same feature may have different meanings in different regions of the image. However, the locally connected layer requires more computational resources than the convolutional layer and may be more prone to overfitting due to the increased number of parameters

## 1.2.LITERATURE REVIEW

| S.NO | TITLE | AUTHOR | METHODLOGY | REMARKS | ADVANTAGES/ DISADVANTAGES |
|---|---|---|---|---|---|
| **1)** | Fuzzy Algorithms for Pattern Recognition in Medical Diagnosis | Shahin Ara Begum and O. Mema Devi | Dealing with uncertainties is a common problem in pattern recognition and the use of fuzzy set theory has given rise to a lot of new methods of pattern recognition for medical diagnosis | it is important to perform quantitative analysis in addition to qualitative evaluation of the medical data. Medical imaging process plays a vital role in early diagnosis. A | Diagnosis and prognosis is the task of medical science. The most important problems in medical diagnosis and prognosis are (i) limited observation and subjectivity of the specialist, (ii) uncertainties and incompleteness in medical knowledge |

| | | | | | |
|---|---|---|---|---|---|
| **2)** | Hybrid Intelligence-Driven Medical Image Recognition for Remote Patient Diagnosis in Internet of Medical Things | Zhiwei Guo , Yu Shen | Convolutional neural networks (CNN) have been proved effective in feature extraction with reasonable deployment of convolution operators and pooling operators. Thus, the DML module is designed with the use of a relatively effective CNN architecture | The DML approaches have been the effective ways for medical image recognition tasks, and have received much attention by researchers from many related academic fields | the growing business volume of healthcare, the daily works of medical staffs are encountered with heavier pressure and challenges The utilization of intelligent and automatic technologies to alleviate current circumstances. |
| **3)** | Artificial Convolution Neural Network for Medical Image Pattern Recognition | SHIH-CHUNG B. Lo, HEANG-PING CHAN, | The structure of the artificial neural network is a simplified network structure of the neocognitron. Two-dimensional local connection as a group is the fundamental architecture for the signal propagation in the convolution | Medical image pattern recognition using feature extraction as an input has been proposed in the detection of disease patterns . Since only a small number of inputs are used , less computation is necessary for | the performance of the CNN in detecting disease was improved significantly by administering these training methods |

| | | | | | |
|---|---|---|---|---|---|
| | | | neural network | training | |
| **4)** | MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network | Zizhao Zhang, Yuanpu Xie | In this paper, we present a unified network, namely MDNet, that can read images, generate diagnostic reports, retrieve images by symptom descriptions, and visualize network attention, to provide justifications of the network diagonosis process | This paper presents a novel unified network, namely MDNet, to establish the direct multimodal mapping from medical images and diagnostic reports | The inability to interpret the model prediction in semantically and visually meaningful ways is a well-known shortcoming of most existing computer-aided diagnosis methods |

# 3. PROBLEM STATEMENT

Medical Image Recognition for Diagnosis aims to address the challenge of improving the accuracy, efficiency, and reliability of disease detection and diagnosis through the application of advanced computer vision and machine learning techniques to various types of medical images. The goal of using Android Studio to develop a medical image recognition application is to provide a mobile tool that is easy to use and can help medical professionals diagnose and identify medical conditions based on visual data, such as images from CT scans, MRIs, and X-rays. The problems of laborious manual analysis, the possibility of human error, and the requirement for quick and accurate medical image interpretation are all addressed by this application. The objective is to create a reliable and effective Android app that can improve the speed and accuracy of medical image recognition, thereby enhancing patient care and healthcare efficiency, by utilizing cutting-edge machine learning and computer vision techniques. This solution has the potential to improve patient care and save lives by speeding up the diagnostic process and making healthcare more accessible in underserved or remote areas. The application's precision in identifying and classifying medical conditions, its usability for healthcare practitioners, and its potential for seamless integration into current healthcare systems will be used to gauge the project's success.

# 4. REQUIREMENTS

## 4.1.Requirement Description

To create medical image recognition software with Visual Studio, there are a few requirements you need to fulfill. Make sure you understand the principles of medical image analysis and the relevant algorithms thoroughly before moving further. You'll also need Visual Studio or an equivalent integrated development environment (IDE) to write and compile your code. In addition, to train and test your models, you will require access to an anonymized and consented medical image dataset. Observe all relevant laws and regulations, including HIPAA in the US, that deal with patient data privacy. Moreover, consider using high-performance hardware, such as GPUs, and deep learning libraries, such as TensorFlow or PyTorch, to expedite model training. Finally, prioritize testing, validation, and clinical collaboration above all else to ensure the accuracy and security of your medical image recognition system.

## 4.2.SOFTWARE REQUIREMENTS

- Programming Languages: Python is the most common programming language for developing medical image recognition algorithms due to its extensive libraries for data manipulation, machine learning, and deep learning, such as TensorFlow, PyTorch, and Keras.
- Deep Learning Frameworks: Frameworks like TensorFlow, PyTorch, and Keras provide pre-built functions and tools for building, training, and evaluating deep learning models.
- TensorFlow: Google created the open-source machine learning framework TensorFlow. It provides an adaptable platform for creating and training neural networks and is widely used for a range of machine learning and deep learning applications. TensorFlow offers both a low-level API for fine-grained control over model architecture and a high-level API for rapid model development. Transformer models, recurrent neural networks (RNNs), and convolutional neural networks (CNNs) are just a few of the many neural network architectures that it supports.
- PyTorch: The AI Research lab (FAIR) at Facebook created the open-source deep learning framework PyTorch. Its growing popularity can be attributed to its adaptability, simplicity of use, and dynamic computation graph. Because PyTorch offers comprehensive support for neural network models, it is especially well-liked by researchers and developers in the artificial intelligence and machine learning domains. . Due to its extensive ecosystem of libraries and tools, PyTorch is a good option for a wide

range of deep learning tasks, including computer vision, reinforcement learning, and natural language processing.

- Keras: Python-based Keras is an open-source, high-level neural network API. It is a well-liked and approachable library that makes deep learning model construction and training easier. Developers can easily switch between different deep learning frameworks, such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK), thanks to Keras's consistent and user-friendly interface. Its modular design makes it a popular option for both novice and seasoned machine learning practitioners, facilitating simple model construction, quick prototyping, and extensive customization. Because Keras makes it possible to quickly develop a wide range of applications, from image and text processing to reinforcement learning, it has been instrumental in democratizing deep learning.

- Streamlit: An open-source Python library called Streamlit makes it easier to create web applications for machine learning and data science. Without requiring in-depth knowledge of web development, developers can quickly transform data scripts into shareable, interactive web apps with Streamlit. It provides a simple and easy-to-use API for creating applications and comes with a large selection of widgets for interactive elements, data visualization, and user input. Data scientists and engineers frequently use Streamlit for dashboard creation, sharing insights, and prototyping because of its real-time, automatic updates that facilitate dynamic data and result display.

## 4.3 HARDWARE REQUIREMENTS

Central Processing Unit (CPU): A powerful CPU is necessary for tasks such as data preprocessing, model architecture design, and training algorithm development.

Graphics Processing Unit (GPU): GPUs, especially those optimized for machine learning tasks, accelerate the training of deep neural networks.

Memory (RAM): Sufficient RAM is essential for loading and processing large medical image datasets.

# 5. PROPOSED APPROACH

Medical image recognition using Convolutional Neural Networks (CNN) is a popular and effective approach for diagnosing medical conditions. CNNs are a type of deep learning algorithm that can learn to recognize patterns in images. They have been used successfully in many medical image recognition tasks, including brain tumor detection, breast cancer diagnosis, and lung nodule detection.

## 5.1 METHODS AND ALGORITHM

Data collection and preprocessing: The first stage entails compiling a representative and varied dataset of medical images associated with the particular ailment of interest, such as mammograms for the diagnosis of breast cancer or MRI scans for the detection of brain tumors. To protect patient privacy, these photos need to be properly anonymized and labeled. To improve the consistency and quality of the data, preprocessing techniques like noise reduction, normalization, and resizing can be used.

- Model Architecture: Create a CNN architecture specifically for the task of medical imaging. Usually, this consists of pooling layers for spatial downsampling, multiple convolutional layers for feature extraction, and fully connected layers for classification. The complexity of the problem may influence the selection of the architecture, the number of layers, and the hyperparameters.

- Training: Using the labeled dataset, the CNN model is trained. The model gains the ability to recognize pertinent details and patterns in the images during training. The difference between the expected and actual labels is measured by loss functions, and optimization methods such as stochastic gradient descent (SGD) modify the model's parameters to minimize this loss.

- Data Augmentation: Techniques for data augmentation can be used to improve the model's resilience and capacity for generalization. These consist of arbitrary flips, rotations, and zooms in addition to lighting-condition adjustments.

- Validation and Testing: To fine-tune hyperparameters and prevent overfitting, the model's performance is assessed using a different validation dataset. In addition, the accuracy and generalizability of the model are evaluated using a test dataset.

- Deployment: The CNN model can be used in a clinical setting after it performs well enough. To help medical professionals diagnose patients, this entails integrating the model into an interface that is easy to use, like a web application or a hospital information system.
- Interpretability: Efforts should be made to make the CNN's decisions interpretable in order to guarantee the model's credibility and acceptance in a medical setting. Methods like saliency maps and attention maps can be used to draw attention to the areas of the picture that influenced the model's choice.
- Continuous Improvement: As new research and techniques are developed, as more data become available, and as medical image recognition models are updated, they should be modified on a regular basis.
- Regulatory Compliance: Throughout the development and deployment process, adherence to regulatory standards for patient data privacy, such as HIPAA (in the US), is crucial.
- Clinical Collaboration: To validate the model's efficacy and make sure it fits with the clinical workflow, collaboration with clinicians and medical experts is essential.

## Design Algorithm

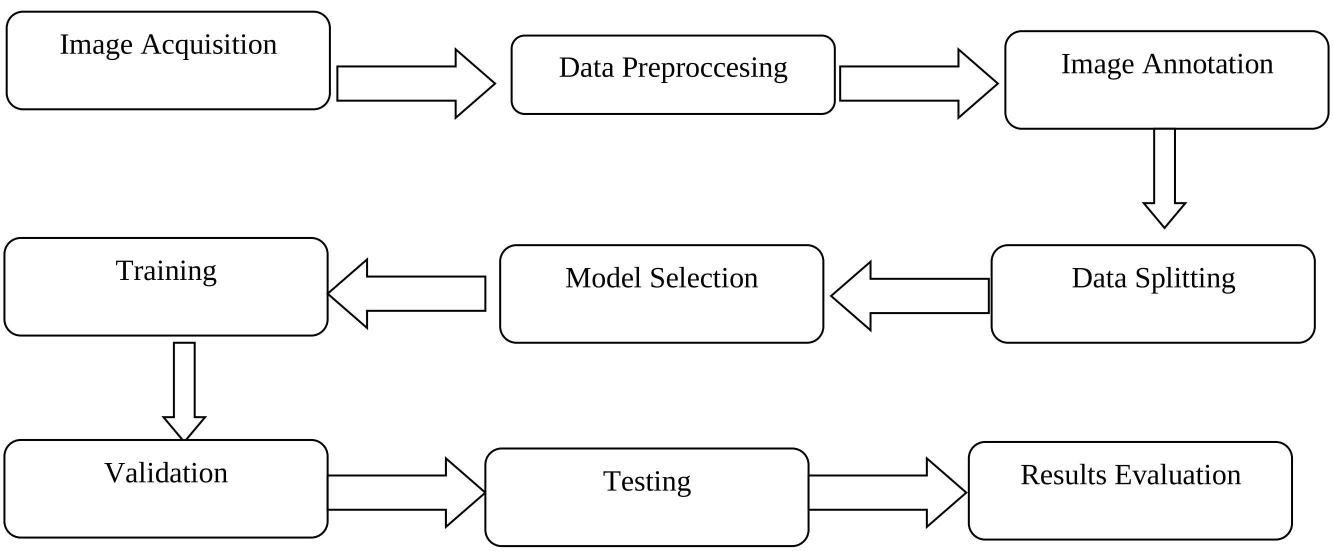


**Figure 2**

# 6. IMPLEMENTATION AND RESULT

## 6.1.Coding

```
import tensorflow as tf
import cv2
import numpy as np
import streamlit as st
def classify(img):
    im = img
    lt = ["other","Bone","Brain","eye","kidney","chest","skin"]
    im = cv2.resize(im,(52,52))
    model = tf.keras.models.load_model("all-in-one.h5",compile=False)
    result = model.predict(np.array([im]))
    a = np.argmax(result)
    c=""
    if a==0:
        st.write('_______________________________________________')
        return "Provide the medical Imaging of the mentioned categories",""
        # st.write('_______________________________________________')
    if a==1:
        st.write('_______________________________________________')
        c= bone_net(im)
        # st.write('_______________________________________________')
    if a==2:
        st.write('_______________________________________________')
        c= brain_net(im)
        # st.write('_______________________________________________')
    if a==3:
```

```
        st.write('______________________________________________________')
        c= Eye_net(im)
        # st.write('______________________________________________________')
    if a==4:
        st.write('______________________________________________________')
        c= kidney_net(im)
        # st.write('______________________________________________________')
    if a==5:
        st.write('______________________________________________________')
        c= chest_net(im)
        # st.write('______________________________________________________')
    if a==6:
        st.write('______________________________________________________')
        c= skin_net(im)
    return c
def bone_net(img):
    # img = cv2.resize(img,(224,224))
    lt = ["Normal",'Fractured']
    model = tf.keras.models.load_model("fracture.h5",compile=False)
    result = model.predict(np.array([img]))
    ans = np.argmax(result)
    a=""
    b=""
    if ans==0:
        a="Normal "
        b="There is no bone facture detected"
    if ans==1:
         a="Fractured"
         b="There is  bone facture.\nconsult doctor Immediately\nApply ice: Apply ice to the affected area to reduce swelling and pain. Wrap the ice in a cloth or towel and apply it to the area for 15-20 minutes at a time, several times a day."
```

```
    return a,b


def brain_net(img):
    lt = ['pituitary', 'notumor', 'meningioma', 'glioma']
    # img = cv2.resize(img,(52,52))
    model = tf.keras.models.load_model("brain.h5",compile=False)
    result = model.predict(np.array([img]))
    ans = np.argmax(result)
    b = ""
    c = ""
    if ans==3:
        b = f"Glioma: Can range from low-grade (mild) to high-grade (severe)."
        c = "\n\nAvoid activities that could cause head injury, such as contact sports.\nTake steps to manage seizures, which can be a common symptom of gliomas.\nAvoid exposure to radiation, as this can increase the risk of developing a glioma.\nTake steps to reduce stress, which can exacerbate symptoms and affect overall health.\nWork with a healthcare provider to manage any other underlying medical conditions that could impact treatment."
    if ans==2:
        b = f"Meningioma: Most are low-grade."
        c = "\n\nAvoid activities that could cause head injury, such as contact sports.\nWork with a healthcare provider to manage any underlying medical conditions, such as high blood pressure, that could impact treatment.\nBe aware of potential symptoms, such as headaches or changes in vision, and seek medical attention promptly if they occur.\nTake steps to reduce stress, which can exacerbate symptoms and affect overall health.\nFollow a healthy lifestyle, including a balanced diet, regular exercise, and adequate sleep."
    if ans==1:
        b = "No tumor: N/A \n\n Enjoy the Life "
        c = "No Tumor has been detected \n Follow the precautions be healthy"
    if ans==0:
        b = f"Pituitary: Can be either benign or malignant."
        c = "\n\n Work with a healthcare provider to manage any underlying medical conditions, such as diabetes, that could impact treatment.\nBe aware of potential symptoms, such as headaches or changes in vision, and seek medical attention promptly if they occur.\nFollow a healthy lifestyle, including a balanced diet, regular exercise, and adequate sleep.\nTake steps to
```

```
manage any hormonal imbalances that may result from the tumor.\nAvoid exposure to radiation, as this can increase the risk of developing pituitary tumors"
    return b,c


def chest_net(img):
    lt = ['PNEUMONIA', 'NORMAL']
    # img = cv2.resize(img,(224,224))
    model = tf.keras.models.load_model("chest.h5",compile=False)
    result = model.predict(np.array([img]))
    ans = np.argmax(result)
    d=""
    e=""
    if ans==0:
        d=f" Might be you are effected by pneumonia  "
        e=f"Vaccination is the best protection against pneumonia."
    if ans==1:
        d=f"Normal"
        e=f"Practice good hygiene by washing your hands frequently and avoiding touching your face"
    return d,e


def Eye_net(img):
    lt = ['glaucoma', 'normal', 'diabetic_retinopathy', 'cataract']
    # img = cv2.resize(img,(224,224))
    model = tf.keras.models.load_model("eye.h5",compile=False)
    result = model.predict(np.array([img]))
    f=""
    g=""
    if ans==3:
        f=f"You are Effected by Cataract"
        g=f'Protect your eyes from harmful UV rays with sunglasses or a hat.'
```

```
    if ans==2:
        f=f"Diaetic Retinopathy"
        g='Control your blood sugar levels to prevent diabetic retinopathy.'
    if ans==1:
        f=f"normal CONDITION"
        g='You are in Normal condition , Maintain same '
    if ans==0:
        f=f'glaucoma detected'
        g="Follow your doctor's instructions for taking glaucoma medications."
    ans = np.argmax(result)
    return f,g

def kidney_net(img):
    lt = ['Cyst', 'Tumor', 'Stone', 'Normal']
    # img = cv2.resize(img,(224,224))
    model = tf.keras.models.load_model("kidney.h5",compile=False)
    result = model.predict(np.array([img]))
    ans = np.argmax(result)
    h=""
    i=""
    if ans==0:
        h="Cyst"
        i="Avoid drinking alcohol, as it can irritate the kidneys and cause cysts to form.\nMaintain a healthy weight to reduce the risk of developing cysts."
    if ans==1:
        h="Tumor"
        i="Quit smoking to reduce the risk of kidney tumors.\nMaintain a healthy diet, rich in fruits and vegetables, to prevent the development of tumors.\nStay physically active to reduce the risk of developing tumors."
    if ans==2:
        h="Stone"
```

```
        i="Drink plenty of water to prevent the formation of kidney stones.\nLimit your intake of salt and animal protein to reduce the risk of developing stones.\nEat a diet rich in fruits and vegetables to prevent the formation of stones."
    if ans==3:
        h="Normal"
        i="You are in Normal Condition\n Be safe"
    return h,i


def skin_net(img):
    lt = ['pigmented benign keratosis', 'melanoma', 'vascular lesion', 'actinic keratosis', 'squamous cell carcinoma', 'basal cell carcinoma', 'seborrheic keratosis', 'dermatofibroma', 'nevus']
    # img = cv2.resize(img,(224,224))
    model = tf.keras.models.load_model("skin.h5",compile=False)
    result = model.predict(np.array([img]))
    ans = np.argmax(result)
    l=""
    m=""
    if ans==0:
        l="pigmented benign keratosis"
        m="Protect your eyes from excessive exposure to sunlight or other sources of UV radiation.\nAvoid rubbing or scratching the affected area to prevent irritation or injury."
    if ans==1:
        l="melanoma"
        m="Regular eye exams with a qualified healthcare professional can help detect early signs of melanoma in the eye.\nWearing UV-protective eyewear and avoiding prolonged exposure to sunlight can help reduce the risk of developing melanoma in the eye."
    if ans==2:
        l="vascular lesion"
        m="Vascular lesions in the eye can be fragile and prone to bleeding, so it is important to avoid any activities that can increase intraocular pressure, such as heavy lifting or straining."
    if ans==3:
        l="actinic keratosis"
```

```
        m="Protect your eyes from UV radiation by wearing UV-blocking sunglasses or a hat with a brim.\nAvoid prolonged exposure to sunlight, especially during peak hours when the sun's rays are strongest."
    if ans==4:
        l="squamous cell carcinoma"
        m="Avoid smoking and limit alcohol consumption as they increase the risk of developing squamous cell carcinoma in the eye."
    if ans==5:
        l="basal cell carcinoma"
        m="Wear protective eyewear.\nMonitor your skin for any changes or abnormalities."
    if ans==6:
        l="seborrheic keratosis"
        m="Seborrheic keratosis near the eye should be examined by a healthcare professional to rule out the possibility of malignant growth.\nAvoid rubbing or scratching the affected area around the eye to prevent irritation or injury."
    if ans==7:
        l="dermatofibroma"
        m="Avoid scratching or picking at the dermatofibroma to prevent infection and further irritation.\nProtect the dermatofibroma from direct sunlight or excessive heat exposure to prevent discoloration or changes in texture."
    if ans==8:
        l="nevus"
        m="Regularly monitor the nevus and report any changes to your healthcare provider.\nProtect your eyes from excessive sun exposure by wearing sunglasses and a hat with a brim."
    return l,m
import io
import os
import numpy as np
import streamlit as st
import requests
from PIL import Image
from model import classify
```

```
import cv2

@st.cache(allow_output_mutation=True)
# def get_model():
#     return bone_frac()
# pred_model = get_model()
# pred_model=bone_frac()
def predict():
    c=classify('tmp.jpg')
    st.markdown('#### Predicted Captions:')
    st.write(c)
st.title('Health Vision')
st.markdown('### What we can do?')
st.write('-Detect Brain tumors')
st.write('-Detect Pnuemonia')
st.write('-Detect Bone Fractures')
st.write('-Detect Skin infections')
st.write('-Detect Kidney Stones')
st.write('-Detect Eye infections')
st.write('')
st.write('(Note:The results may not be correct always its better to have a second opnion)')
# img_url = st.text_input(label='Enter Image URL')

# if (img_url != "") and (img_url != None):
#     img = Image.open(requests.get(img_url, stream=True).raw)
#     img = img.convert('RGB')
#     st.image(img)
#     img.save('tmp.jpg')
#     predict()
#     os.remove('tmp.jpg')
```

```
hide_streamlit_style = """
            <style>
            #MainMenu {visibility: hidden;}
            footer {visibility: hidden;}
            </style>
            """
st.markdown(hide_streamlit_style, unsafe_allow_html=True)
# st.markdown('<center style="opacity: 70%">OR</center>', unsafe_allow_html=True)
img_upload = st.file_uploader(label='Upload Image', type=['jpg', 'png', 'jpeg'])
if img_upload != None:
    img = img_upload.read()
    img = Image.open(io.BytesIO(img))
    img = img.convert('RGB')
    img=np.asarray(img)
    print(img)
    # img=cv2.imread(img)
    # img.save('tmp.jpg')
    st.image(img)
    c,b=classify(img)
    st.markdown('#### Possible Disease Prediction:')
    st.write(c)
    st.write('___________________________________________________')
    st.markdown('#### Precautions To Be Taken :')
    st.write('___________________________________________________')
    st.write(b)
    # predict()
    # os.remove('tmp.jpg')
```

## 6.2Output screenshot

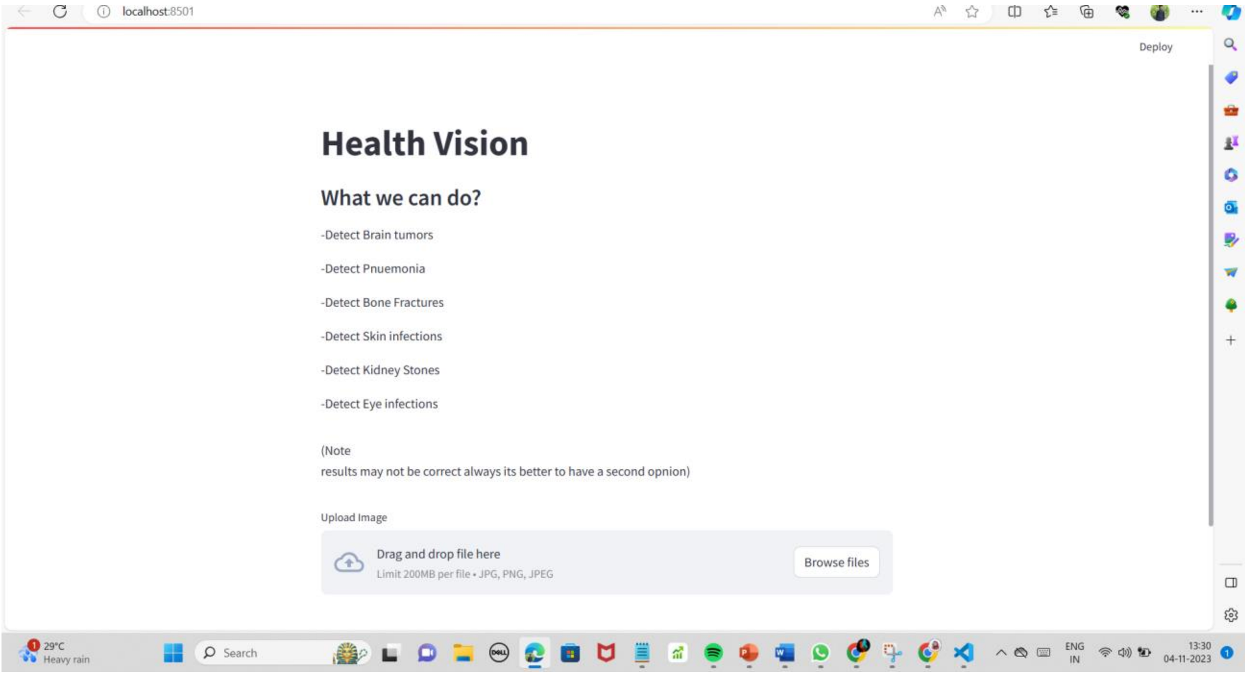


**Figure 3**

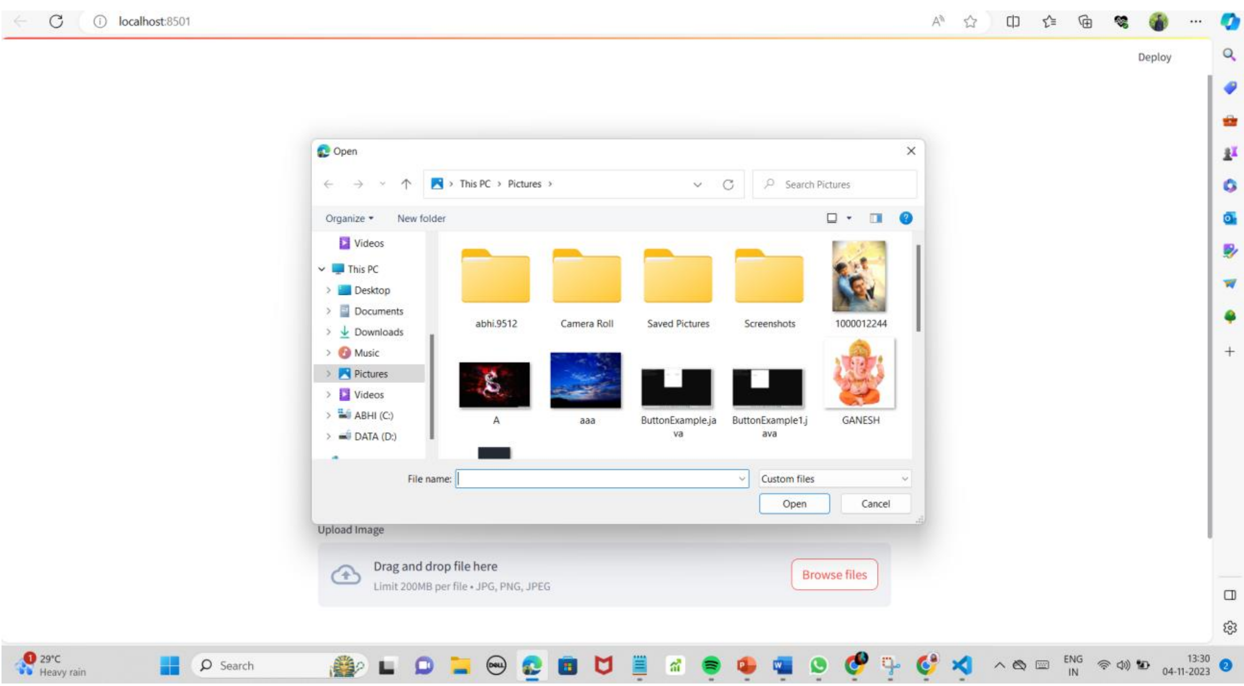


**Figure 4**

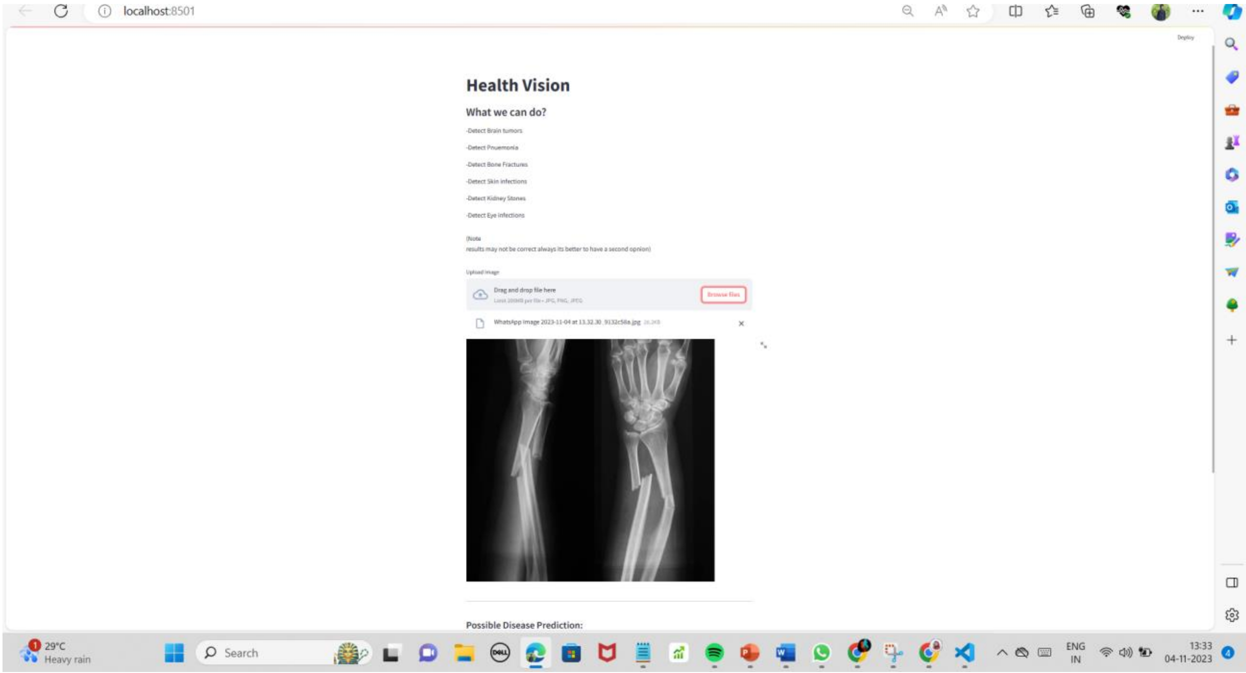


**Figure 5**

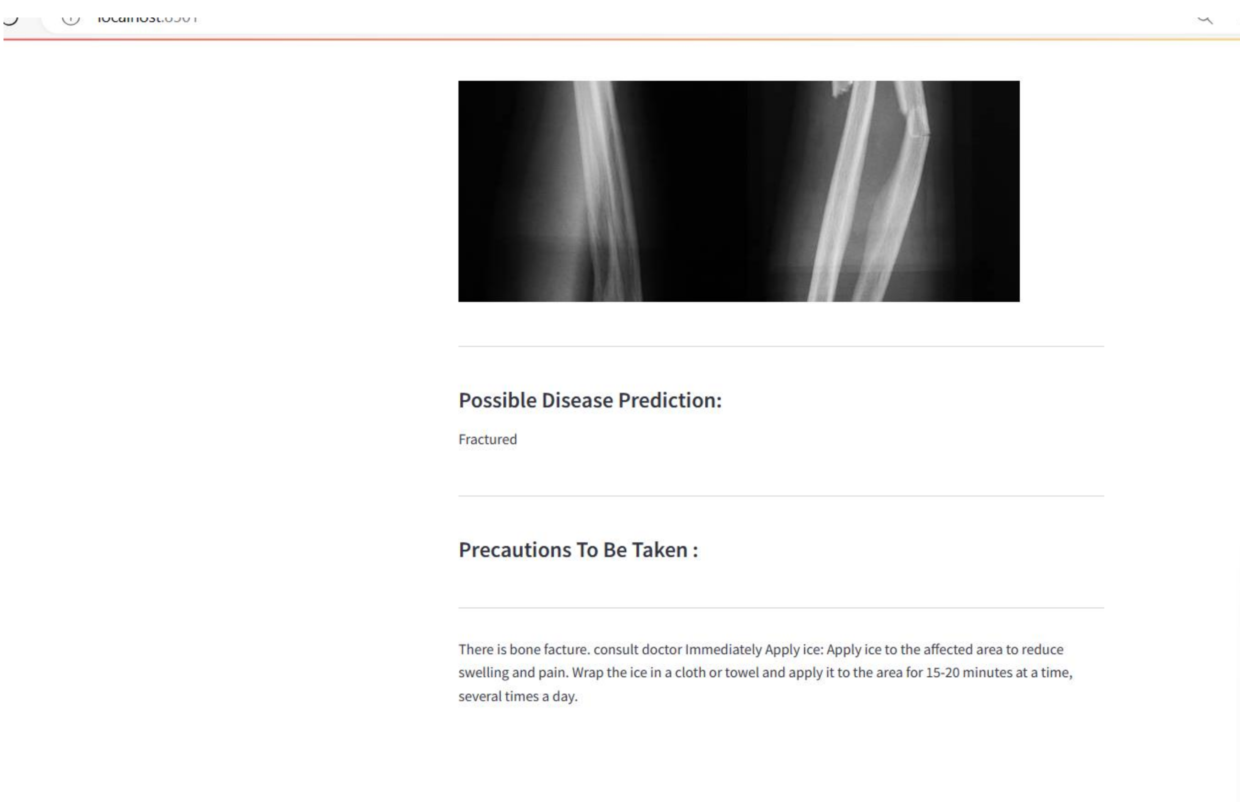


**Figure 6**

# 7. CONCLUSION AND FUTURE ENHANCEMENT

## 7.1. Conclusion

In conclusion, the healthcare sector has a great deal of potential for medical image recognition. Artificial intelligence and deep learning have advanced to the point where they are now vital tools for doctors, aiding in the early diagnosis and detection of everything from cancer to neurological disorders. Because these systems can quickly and accurately analyze large amounts of medical imagery, they have the potential to significantly improve patient outcomes, reduce healthcare costs, and speed up the diagnostic process. In order for medical image recognition technologies to be safely and successfully used in clinical settings, concerns about data privacy, model interpretability, and ongoing algorithm improvement must be addressed. Medical image recognition is positioned to become more and more important as technology develops, improving patient care and transforming the healthcare sector.

## 7.2 Future Enhancement

Medical image recognition has a bright future ahead of it as it continues to transform healthcare. More precise and effective diagnosis of a variety of medical conditions, including cancer, neurological disorders, and cardiovascular diseases, is now possible thanks to developments in artificial intelligence and deep learning. Medical image recognition systems will become even more potent with the integration of big data and cloud computing, enabling quicker image processing and smoother data sharing between healthcare providers. Additionally, the advancement of wearable technology and telemedicine applications will broaden the application of medical image recognition by allowing for remote patient monitoring and real-time condition evaluation. With improved early disease detection, treatment planning, and personalized medicine, this technology has the potential to improve patient outcomes and save healthcare costs.

# 8. REFERENCES


- S.Begum,O.Devi, Published 4 May 2011,Computer Science,Assam University Journal of Science and Technology.
- IEEE J Biomed Health Inform 2022 Dec;26(12):5817-5828. doi: 10.1109/JBHI.2021.3139541. Epub 2022 Dec 7
- Front. Mol. Biosci., 28 January 2021 Sec. Molecular Diagnostics and Therapeutics.
- Computer Vision and Pattern Recognition , Zizhao Zhang, Yuanpu Xie, Fuyong Xing, Mason McGough, Lin Yang University of Florida
- Greenspan, Hayit, Zhou, S. Kevin, and Shen, Dinggang. (2017). Deep Learning for Analysis of Medical Images.
- Gustavo Carneiro, Andrew P. Bradley, and Jacinto C. Nascimento (2019). Deep Learning in the Analysis of Medical Images.
- Li (2017), Qin (2017), Sun (2017), et al. An Overview of Deep Learning for Image Analysis in Medicine.
- Sankararaman, Sriram, and Arjun Sharma. (2019). An overview of the concepts and a survey of the state of the art for deep learning in radiology.
- Bejnordi, B. E., Kooi, T., Litjens, G., et al. (2017). Recent Developments and Future Directions in Deep Learning for Medical Image Analysis.
- Hinton, Geoffrey E., Sutskever, Ilya, and Alex Krizhevsky (2012). Deep Convolutional Neural Networks for ImageNet Classification.