DR Detect

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SYED MAHBOOB ABRAR ALI	**(1604-18-733-088)**
MOHD ABDUL AZEEM	**(1604-18-733-089)**
MOHAMMED SUFIYAN ABDULLAH GHORI	**(1604-18-733-094)**

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(Affiliated to Osmania University)

Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34

2022

Date: / / 2022

CERTIFICATE

This is to certify that the project dissertation titled " ***Dr Detect*** " being submitted by

1. SYED MAHBOOB ABRAR ALI (1604-18-733-088)
2. MOHD ABDUL AZEEM (1604-18-733-089)
3. MOHAMMED SUFIYAN ABDULLAH GHORI (1604-18-733-094)

in partial fulfillment of the requirements for the award of the degree Of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2021-22 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide | Head CSED

(Mr. Md. Zainuddin Naveed) | (Dr. A.A. Moiz Qyser)

(Assistant Professor)

External Examiner

DECLARATION

This is to certify that the work reported in the major project entitled "**DR Detect**" is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. SYED MAHBOOB ABRAR ALI (1604-18-733-088)
2. MOHD ABDUL AZEEM (1604-18-733-089)
3. MOHAMMED SUFIYAN ABDULLAH GHORI (1604-18-733-094)

ACKNOWLEDGEMENT

Our hearts are filled with gratitude to the Almighty for empowering us with courage, wisdom and strength to complete this project successfully. We give him all the glory, honor and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to us and make us promising professionals for tomorrow.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Asst Prof. Md. Zainuddin Naveed** for his valuable suggestions and interest throughout the course of this project.

We are happy to express our profound sense of gratitude and indebtedness to **Prof. Dr. Ahmed Abdul Moiz Qyser**, Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educating, guidance, constant encouragement right throughout our work and making us successful.

With a great sense of pleasure and privilege, we extend our gratitude to **Mrs. Maniza Hijab**, Associate Professor of Computer Science and Engineering Department, project in-charge, offered valuable suggestions as a pre-requisite to carry out support in every step.

We are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

SYED MAHBOOB ABRAR ALI
MOHD ABDUL AZEEM
MOHAMMED SUFIYAN ABDULLAH GHORI

ABSTRACT

Early recognition of results of medical reports is the key to preventing risky complications (even death), even more so now that millions of COVID-19 patients require care . Deep learning algorithms may be able to automatically detect abnormalities. Once alerted, clinicians can take appropriate decisions to avoid life threatening complications

Delay or Lack of proper diagnosis of various health-related issues is a major concern, so Early Detection of such reports is even more important as COVID-19 cases continue to surge, these steps can be time consuming and are still prone to human error, especially in stressful situations when hospitals are at capacity.

Deep learning algorithms simplify complex data analysis, so abnormalities are determined and prioritized more precisely. The insights that Convolutional Neural Networks (CNNs) provide, help medical professionals to notice health issues of their patients on time and more accurately.

This is a Computer Vision task that can be solved by segmenting or classifying the different components in the medical reports accordingly. Making use of State of the Art Computer Vision Algorithms would be an ideal approach.

Providing AI solutions for Healthcare issues like: Diabetic Retinopathy,Brain Tumor, Breast Lesions, Pneumonia, Common Pigmented Skin Lesions, Knee Osteoarthritis.

The proposed solution will help to understand the result in the state of an emergency even by a paramedic/Jr Doctor in absence of the specialist.

CONTENTS

Title	
Certificate	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Figures	viii
List of Tables	viii
1. Introduction	1
2. Literature Survey	5
2.1. Diabetic Retinopathy	6
2.1.1. Objective	6
2.1.2. Introduction	6
2.1.3. Conclusion	6
2.1.4. Findings	6
2.2. Brain Tumor	7
2.2.1. Objective	7
2.2.2. Introduction	7
2.2.3. Conclusion	8
2.2.4. Findings	8
2.3. Breast Lesions	8
2.3.1. Objective	8
2.3.2. Introduction	8
2.3.3. Conclusion	9
2.3.4. Findings	9
2.4. Pneumonia	9
2.4.1. Objective	9
2.4.2. Introduction	10
2.4.3. Conclusion	10
2.4.4. Findings	10
2.5. Common Pigmented Skin Lesions	11
2.5.1. Objective	11
2.5.2. Introduction	11

2.5.3. Conclusion 11
2.5.4. Findings 11
2.6. Knee Osteoarthritis 12
2.6.1. Objective 12
2.6.2. Introduction 12
2.6.3. Conclusion 12
2.6.4. Findings 13

3. System Analysis 14
3.1. Problems with existing system 14
3.2. Proposed System 15
3.3. Feasibility Study 16
3.3.1. Types of Feasibility 16
3.3.1.1. Technical Feasibility 17
3.3.1.2. Operational Feasibility 17
3.3.1.3. Economical Feasibility 17
3.4. Software Requirement Specification 18
3.4.1. Introduction 18
3.4.1.1. Purpose 18
3.4.1.2. Project Scope 18
3.4.2. Overall Description 18
3.4.2.1. Product Perspective 18
3.4.2.2. Product Functions 19
3.4.2.3. Operating Environment 19
3.4.3. External Interface Requirements 20
3.4.3.1. User Interfaces 20
3.4.3.2. Hardware Interfaces 20
3.4.3.3. Software Interfaces 20
3.4.4. System Features 21
3.4.4.1. Disease Overview 21
3.4.4.2. Diagnosis of the reports 21

4. System Design 22
4.1. System Design Description 23

5. Implementation 26
5.1. Web Application Sample 27
5.2. Model Creation 28
5.3. Loading the model and prediction 31
5.4. Testing the behavior of the web application 33
5.5. Dockerfile 33
5.6. Result Analysis 35

6. Testing 37

6.1. Pre-train Tests 37
6.2. Post-train Tests 38

7. Screenshots 41

8. Conclusion 46

9. Future Prospects 47

References 48

LIST OF FIGURES

Figure No.	Description	Page No.
Fig 3.4.2.1.1	Project Block Diagram	19
Fig 4.1	System Design	22
Fig 5.1	Project Directory	26
Fig 7.1	About Page	41
Fig 7.2	Preface Page	42
Fig 7.3	List of Diseases	42
Fig 7.4	Textual Explanation	43
Fig 7.5	Visual Explanation	43
Fig 7.6	Model Prediction	44
FIg 7.7	Sample Prediction	44
Fig 7.8	Azure WebApp Log Stream	45
Fig 7.9	Heroku WebApp Log Stream	45

LIST OF TABLES

Table No.	Description	Page No.
Table 2.1	Literature Survey Summary	5
Table 3.4.3.3.1	Softwares Used	20
Table 5.6.1	Result Analysis	35

1. INTRODUCTION

AI, machine learning, and deep learning have gained a lot of attention for quite some time now. Healthcare is one such industry that implements these technologies the most. As health is a priority, medical experts are continually trying to find ways to implement new technologies and provide impactful results. Deep learning in healthcare offers path breaking applications. Deep learning gathers a massive volume of data, including patients' records, medical reports, and insurance records, and applies its neural networks to provide the best outcomes.

Deep learning is assisting medical professionals and researchers to discover the hidden opportunities in data and to serve the healthcare industry better. Deep learning in healthcare provides doctors the analysis of any disease accurately and helps them treat them better, thus resulting in better medical decisions.

Deep learning in healthcare helps in discovery of medicines and their development. The technology analyzes the patient's medical history and provides the best treatment for them. Moreover, this technology is gaining insights from patient symptoms and tests.

Medical imaging techniques such as MRI scans, CT scans, ECG, are used to diagnose dreadful diseases such as heart disease, cancer, brain tumor. Hence, deep learning helps doctors to analyze the disease better and provide patients with the best treatment.

Many of the industry's deep learning headlines are currently related to small-scale pilots or research projects in their pre-commercialized phases.

However, deep learning is steadily finding its way into innovative tools that have high-value applications in the real-world clinical environment.

Some of the most promising use cases include innovative patient-facing applications as well as a few surprisingly established strategies for improving the health IT user experience.

One type of deep learning, known as convolutional neural networks (CNNs), is particularly well-suited to analyzing images, such as MRI results or x-rays.

As a result, some CNNs are approaching – or even surpassing – the accuracy of human diagnosticians when identifying important features in diagnostic imaging studies.

Even when human clinicians were equipped with background information on patients, such as age, sex, and the body site of the suspect feature, the CNN outperformed the dermatologists by nearly 7 percent.

The images use patterns learned from real scans to create synthetic versions of CT or MRI images. The data can be randomly generated and endlessly diverse, allowing researchers to access large volumes of necessary data without any concerns around patient privacy or consent.

As intriguing as these pilots and projects can be, they represent only the very beginning of deep learning's role in healthcare analytics.

Excitement and interest about deep learning are everywhere, capturing the imaginations of regulators and rule makers, private companies, care providers, and even patients.

This targeted form of AI and deep learning helps the overburdened radiologist by flagging items that are of concern and thereby allows the healthcare professional to direct patients with greater control and efficiency. It also reduces admin by integrating into workflows and improving access to relevant patient information.

The technology that supports the medical profession is becoming increasingly capable of integrating AI-based algorithms that can streamline and simplify complex data analysis and improve diagnosis. It can be trained and it can learn. It can reduce reporting delays and improve workflows. And it can be used to shift the benchmarks of patient care in a time and budget strapped economy.

Deep learning in healthcare will continue to make inroads into the industry, especially now that more and more medical professionals are recognizing the value it brings. This technology can only benefit from intense collaboration with industry and specialist organizations. It needs to remain agile and able to adapt to ensure that it always remains relevant to the profession.

While deep learning in healthcare is still in the early stages of its potential, it has already seen significant results. The benefits it brings have been recognized by leading institutions and medical bodies, and the popularity of the solutions has reached a fever pitch

The future still lies in the hands of the medical professionals, but they are now being supported by technology that understands their unique needs and environments and reduces the stresses that they experience on a daily basis.

AI is in its infancy and its long-term implications are uncertain. Future applications of AI in healthcare delivery, in the approach to innovation and in how each of us thinks about our health, may be transformative. A future can be imagined in which population-level data from wearables and implants change our understanding of human biology and of how medicines work, enabling personalized and real-time treatment for all. What is real today and what will enable innovation and adoption tomorrow, rather than exploring the long-term future of personalized medicine. Faced with the uncertainty of the eventual scope of application of emerging technologies, some short-term opportunities are clear, as are steps that will enable health providers and systems to bring benefits from innovation in AI to the populations they serve more rapidly.

AI Healthcare Industry can play a fundamental role in catalyzing the introduction and scaleup of Artificial Intelligence by the following:

- Develop a regional or national AI strategy for healthcare, defining a medium- and longer-term vision and goals, specific initiatives, resources and performance indicators. Define use cases to support through targeted funding and incentives to enable scaling of AI solutions across the system; ensure these deliver against both clinical and operational outcomes.

- Set standards for digitization, data quality and completeness, data access, governance, risk management, security and sharing, and system interoperability; incentivize adherence to standards through a combination of performance and financial incentives.

- Redesign workforce planning and clinical-education processes to address the needs of both future healthcare and AI-focused professionals; and invest upfront in upskilling frontline staff and designing lifelong-learning programs through continuing professional development and degrees or diplomas for healthcare professionals.

- Provide incentives and guidance for healthcare organizations to collaborate in centers of excellence/clusters of innovation at the regional or national level.

- Address AI regulation, liability and funding issues, creating the right environment for appropriate, safe and effective AI solutions to be adopted but minimizing the risk to practitioners.

- Ensure this is reflected in funding and reimbursement mechanisms for innovation in healthcare—the number one priority for survey respondents from health systems, alongside simplifying data-governance and data-sharing processes.

2. LITERATURE SURVEY

Table 2.1 : Literature Survey Summary

Healthcare Issue	**Research Paper Title**	**Proposed Solution**	**Metrics**
Diabetic Retinopathy	Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs [1]	Deep Convolutional Neural Network	AUC-ROC
Brain Tumor	Brain tumor classification in MRI image using convolutional neural network [2]	Deep Convolutional Neural Network	Accuracy, Loss
Breast Lesions	Segmentation and recognition of breast ultrasound images based on an expanded U-Net [3]	Deep Convolutional Neural Network, Auto Encoders	Accuracy, Loss
Pneumonia	Pneumonia detection in chest X-ray images using an ensemble of deep learning models [4]	Deep Convolutional Neural Network	Accuracy, Loss
Common Pigmented Skin Lesions	Deep Learning for Two-Step Classification of Malignant Pigmented Skin Lesions [5]	Deep Convolutional Neural Network	Accuracy, Loss
Knee Osteoarthritis	Automatic Detection of Knee Joints and Quantification of Knee Osteoarthritis Severity Using Convolutional Neural Networks [6]	Deep Convolutional Neural Network	Accuracy, Loss

2.1. Diabetic Retinopathy

2.1.1. Objective

To apply deep learning to create an algorithm for automated detection of diabetic retinopathy and diabetic macular edema in retinal fundus photographs.

2.1.2. Introduction

Automated grading of diabetic retinopathy has potential benefits such as increasing efficiency, reproducibility, and coverage of screening programs; reducing barriers to access; and improving patient outcomes by providing early detection and treatment. To maximize the clinical utility of automated grading, an algorithm to detect referable diabetic retinopathy is needed. Machine learning has been leveraged for a variety of classification tasks including automated classification of diabetic retinopathy. Deep learning is a machine learning technique that avoids such engineering by learning the most predictive features directly from the images given a large data set of labeled examples. This technique uses an optimization algorithm called back-propagation to indicate how a machine should change its internal parameters to best predict the desired output of an image.

2.1.3. Conclusion

In this evaluation of retinal fundus photographs from adults with diabetes, an algorithm based on deep machine learning had high sensitivity and specificity for detecting referable diabetic retinopathy.

2.1.4. Findings

The development of similar high-performing algorithms for medical imaging using deep learning has 2 prerequisites.

First, there must be a collection of a large developmental set with tens of thousands of abnormal cases. While performance on the tuning set saturated at 60 000 images, additional gains might be achieved by increasing the diversity of training data (ie, data from new clinics).

Experimenting various architectures to achieve good performance with the limited amount of dataset available.

Second, data sets used to measure final performance (tuning and the clinical validation data sets) should have multiple grades per image. The reference standard used for this study was the majority decision of all ophthalmologist graders. This means the algorithm may not perform as well for images with subtle findings that a majority of ophthalmologists would not identify which is out of scope for this project.

2.2. Brain Tumor

2.2.1. Objective

Introduction to the convolutional neural network (CNN) approach along with Data Augmentation and Image Processing to categorize brain MRI scan images into cancerous and non-cancerous.

2.2.2. Introduction

A brain is the most sensitive organ of our body, which controls the core functions and characteristics in the human body and according to the National Brain Tumor Society, in the United States, about 700,000 people live with a brain tumor, and the figure will rise to 787,000 by the end of 2020

Brain tumor has a lasting and psychological impact on patient life.The Brain tumor is caused by tissue abnormality that develops within the brain or in the central spine, interrupting proper brain function.

The Brain tumor is diagnosed using several techniques such as CT scan, EEG, but Magnetic Resource Image (MRI) is the most effective and widely used method. In the past few years because of AI and Deep learning, significant advancement has been made in the medical science like Medical Image processing technique which helps doctors to the diagnose disease early and easily, before that, it was tedious and time-consuming.So in our study with the help of Brain MRI Images, we provide a method for classification of brain tumors into cancerous and non-cancerous using data augmentation technique and convolutional neural network model.

2.2.3. Conclusion

Our proposed system can play a prognostic significance in the detection of tumors in brain tumor patients. To further boost the model efficiency, comprehensive hyper-parameter tuning and a better preprocessing technique can be conceived.

2.2.4. Findings

First, using the image edge detection technique, we found the region of interest in MRI images and cropped them. Then, we used the data augmentation technique for increasing the size of our training data.

Second, we provide an efficient methodology for brain tumor classification by proposing a simple CNN network. For sophisticated and accurate results neural networks require a large amount of data to train on. We can prolong this approach in other scientific areas as well where there is a problem in the availability of large data or we can use the different transfer learning methods with the same proposed technique.

2.3. Breast Lesions

2.3.1. Objective

This paper establishes a fully automatic real-time image segmentation and recognition system for breast ultrasound intervention

2.3.2. Introduction

Breast cancer is the most common malignancy in women and the main cause of cancer deaths among women worldwide. It accounts for 30% of cancer diagnoses in women, with increasing incidence in recent years

The main innovations of this paper are briefly summarized as follows: the research adopts the basic architecture of a U-Net [7], analyses the actual application scenarios of semantic segmentation of breast ultrasound images, adds dropout layers to the U-Net architecture to reduce the redundancy in texture details and prevent overfitting, and proposes an expanded training approach to obtain the expanded U-Net. It is difficult and expensive to obtain large-scale breast ultrasound images marked by professional doctors, which is far from meeting the needs of large-scale training data. Therefore, this paper uses a data enhancement approach based on geometric transformation to expand the scale of the dataset and solve the problem of overfitting during network training

2.3.3. Conclusion

Through theoretical analysis and experimental research, the paper draws the following conclusions.

(1) The context of breast ultrasound images can be extracted, and texture details and edge features of the tumor can be retained by an expanded U-Net.

(2) Using the expanded U-Net can quickly and automatically achieve precise segmentation and multi-class recognition of breast ultrasound images.

2.3.4. Findings

FCN accepts fixed-size input, its accuracy is not fine enough, context details are not fully extracted, and spatial consistency is lacking.

Compared with the general FCN, the input of the U-Net can be of any size, and the output is a heat map with the same size as the input. This end-to-end deep learning is conducive to interdisciplinary applications. This architecture has high requirements for data size and labeling quality, and accurate labeling of medical image data requires very high professionalism. It is very difficult to accurately label medical images in large quantities, which affects the performance of the segmentation model based on deep learning, and it is prone to overfitting and instability during the training process.

2.4. Pneumonia

2.4.1. Objective

employed deep transfer learning to handle the scarcity of available data and designed an ensemble of three convolutional neural network models: GoogLeNet [8], ResNet-18 [9], and DenseNet-121 [10]

2.4.2. Introduction

Pneumonia is a respiratory infection caused by bacteria or viruses; it affects many individuals, especially in developing and underdeveloped nations, where high levels of pollution, unhygienic living conditions, and overcrowding are relatively common, together with inadequate medical infrastructure

Chest X-ray imaging is the most frequently used method for diagnosing pneumonia. However, the examination of chest X-rays is a challenging task and is prone to subjective variability

Ensemble learning is a popular strategy in which the decisions of multiple classifiers are fused to obtain the final prediction for a test sample. Some of the ensemble techniques that were most frequently used in studies in the literature are average probability, weighted average probability, and majority voting.

In this study, we devised a novel strategy for weight allocation, where four evaluation metrics, precision, recall, f1-score, and area under receiver operating characteristics (ROC) curve (AUC), were used to assign the optimal weight to three base CNN models.

2.4.3. Conclusion

An ensemble framework, proposed for boosting the performance of the base CNN learners in pneumonia classification, was developed. For this purpose, a weighted average ensemble technique was adopted.To assist medical practitioners, an automated CAD system was developed in this study, which uses deep transfer learning-based classification to classify chest X-ray images into two classes “Pneumonia” and “Normal.”

2.4.4. Findings

Furthermore, because three CNN models are required to train the proposed ensemble, the computation cost is higher than that of the CNN baselines developed in studies in the literature.

a lightweight deep neural network, MobileNet [11] has fewer parameters and higher classification accuracy thereby reduces the parameters and the computation cost.

2.5. Common Pigmented Skin Lesions

2.5.1. Objective

An automatic detection of malignant pigmented skin lesions is investigated

2.5.2. Introduction

Skin cancer is one of the most common types of cancer. Its early detection drastically improves outcomes and saves human lives. Even though the diagnosis of these cancer types is done by a skin biopsy, automatic detection of skin cancer using computerized methods may lead to a faster and a more accurate diagnosis.

In this paper, an automatic detection of malignant pigmented skin lesions is investigated. For this, the two-step skin lesion diagnostic procedure of the dermatologists is followed. Using a deep learning model, the skin lesion is first classified as melanocytic or non-melanocytic and then malignant types are detected using other deep learning models. The performance evaluations show that melanocytic and nonmelanocytic skin lesions are detected with the highest accuracy. They also show that melanocytic malignant skin lesions can be classified with a higher accuracy than non-melanocytic malignant skin lesions.

2.5.3. Conclusion

This method is based on three deep learning models. A skin lesion can be classified as melanocytic using the first deep learning model. Then, the detection of melanoma skin lesions can be done using the second deep learning model. Similarly, a skin lesion can be classified as non-melanocytic using the first deep learning model and then malignant types (basal cell carcinoma, squamous cell carcinoma) can be detected using the third deep learning model.

2.5.4. Findings

It can be concluded from these results that since both melanocytic and non-melanocytic classes (first model) contain multiple skin lesion types, their detection is more difficult than melanoma and melanocytic nevus classes (second model). Furthermore, as the number of training images in the third model is not high enough, the deep learning architecture is not able to fully differentiate between similar non-melanocytic malignant and benign skin lesion images. Hence, non-melanocytic malignant skin lesion detection (third model) is also concluded to be more difficult than melanocytic malignant skin lesion detection (second model).

2.6. Knee Osteoarthritis

2.6.1. Objective

This paper introduces a new approach to automatically quantify the severity of knee OA (Osteoarthritis) using X-ray images

2.6.2. Introduction

Knee Osteoarthritis (OA) is a debilitating joint disorder that mainly degrades the knee articular cartilage . In its severe stages, it causes excruciating pain and often leads to total joint arthroplasty. Early diagnosis is crucial for clinical treatments and pathology Despite the introduction of several imaging modalities such as MRI, Optical Coherence Tomography and ultrasound for augmented OA diagnosis, radiography (X-ray) has been traditionally preferred, and remains the main accessible tool and "gold standard" for preliminary knee OA diagnosis.

This involves two main steps: 1) automatically detecting and extracting the region of interest (ROI) and localizing the knee joints, 2) classifying the localized knee joints.

We introduce a fully-convolutional neural network (FCN) based method to automatically localize the knee joints. A FCN is an end-to-end network trained to make pixel-wise predictions . Our FCN based method is highly accurate for localizing knee joints and the FCN can easily fit into an end-to-end network trained to quantify knee OA severity

2.6.3. Conclusion

We showed that the classification results obtained with automatically localized knee joints is comparable with the manually segmented knee joints. There is an improvement in the multi-class classification accuracy, precision, recall, and F1 score of the jointly trained network for classification and regression in comparison to the previous method

2.6.4. Findings

We demonstrate that classification accuracy can be significantly improved using deep convolutional neural network models pre-trained on ImageNet and fine-tuned on knee OA images. Furthermore, we argue that it is more appropriate to assess the accuracy of automatic knee OA severity predictions using a continuous distance-based evaluation metric like mean squared error than it is to use classification accuracy

3. SYSTEM ANALYSIS

3.1 Problems with existing system

Deep learning models are trained for specific image recognition tasks (such as nodule detection on chest computed tomography or hemorrhage on brain magnetic resonance imaging). However, thousands of such narrow detection tasks are necessary to fully identify all potential findings in medical images, and only a few of these can be done by AI today.

Clinical processes for employing AI-based image work are a long way from being ready for daily use. Different imaging technology vendors and deep learning algorithms have different foci: the probability of a lesion, the probability of cancer, a nodule's feature or its location. These distinct foci would make it very difficult to embed deep learning systems into current clinical practice.

Deep learning algorithms for image recognition require ‘labeled data’ – millions of images from patients who have received a definitive diagnosis of cancer, a broken bone or other pathology. However, there is no aggregated repository of radiology images, labeled or otherwise.

Substantial changes will be required in medical regulation and health insurance for automated image analysis to take off.

There are also a variety of ethical implications around the use of AI in healthcare. Healthcare decisions have been made almost exclusively by humans in the past, and the use of smart machines to make or assist with them raises issues of accountability, transparency, permission and privacy.

Humans are likely to encounter many ethical, medical, occupational and technological changes with AI in healthcare. It is important that healthcare institutions, as well as governmental and regulatory bodies, establish structures to monitor key issues, react in a responsible manner and establish governance mechanisms to limit negative implications. This is one of the more powerful and consequential technologies to impact human societies, so it will require continuous attention and thoughtful policy for many years.

3.2. Proposed System

Delay or Lack of proper diagnosis of various health-related issues is a major concern, so Early Detection of such reports is even more important as COVID-19 cases continue to surge, these steps can be time consuming and are still prone to human error, especially in stressful situations when hospitals are at capacity.

Creation of a software system available to the paramedic/Jr Doctor in absence of the specialist to understand the result in the state of an emergency which can help in Early Detection of medical reports and it is very crucial to help in preventing risky complications (even death).

Medical imaging techniques such as MRI scans, CT scans, ECG, are used to diagnose dreadful diseases such as heart disease, cancer, brain tumor. Hence, deep learning helps doctors to analyze the disease better and provide patients with the best treatment.

Deep learning algorithms may be able to automatically detect abnormalities. It is steadily finding its way into innovative tools that have high-value applications in the real-world clinical environment. Medical reports (like MRIs, XRAYs) are used to classify or detect diseases well beforehand and prevent mishappenings.

One type of deep learning, known as convolutional neural networks (CNNs), is particularly well-suited to analyzing images, such as MRI results or x-rays. As a result, some CNNs are approaching – or even surpassing – the accuracy of human diagnosticians when identifying important features in diagnostic imaging studies.

This technology can only benefit from intense collaboration with industry and specialist organizations. It needs to remain agile and able to adapt to ensure that it always remains relevant to the profession.

This targeted form of AI and deep learning helps the overburdened specialists by flagging items that are of concern and thereby allows the healthcare professional to direct patients with greater control and efficiency. It also reduces admin by integrating into workflows and improving access to relevant patient information. Once alerted, clinicians can take appropriate decisions to avoid life threatening complications

As intriguing as this pilot project can be, it represents only the very beginning of deep learning's role in healthcare analytics.

3.3. Feasibility Study

Feasibility study is the test of a system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources. It focuses on the evaluation of existing system and procedures analysis of alternative candidate system cost estimates. Feasibility analysis was done to determine whether the system would be feasible.

The development of a computer-based system or a product is more likely plagued by resources and delivery dates. Feasibility study helps the analyst to decide whether or not to proceed, amend, postpone or cancel the project, particularly important when the project is large, complex and costly.

Once the analysis of the user requirement is complementing, the system has to check for the compatibility and feasibility of the software package that is aimed at. An important outcome of the preliminary investigation is the determination that the system requested is feasible.

3.3.1. Types of Feasibility

1. Technical Feasibility
2. Operational Feasibility
3. Economical Feasibility

3.3.1.1. Technical Feasibility

The main goal of Machine Learning (ML) feasibility studies is to assess whether it is feasible to solve the problem satisfactorily using ML with the available data. We want to avoid investing too much in the solution before we have:

- Sufficient evidence that an ML solution would be the best technical solution given the business case
- Sufficient evidence that an ML solution is possible
- Some vetted direction on what an ML solution should look like

This effort ensures quality solutions backed by the appropriate, thorough amount of consideration and evidence.

3.3.1.2. Operational Feasibility

This proposed system can easily be implemented, as this is based on Java and cyber security. The database created is with Mysql which is more secure and easy to handle. The resources that are required to implement/install these are available. The personal of the organization already has enough exposure to computers. So, the project is operationally feasible and user friendly.

3.3.1.3. Economical Feasibility

If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action. This system is more economically feasible and budget friendly, so it is economically a good project.

3.4. Software Requirement Specification

3.4.1. Introduction

3.4.1.1. Purpose

The purpose of this document is to build a software system available to the paramedic/Jr Doctor in absence of the specialist to understand the result in the state of an emergency which can help in Early Detection of medical reports and it is very crucial to help in preventing risky complications (even death).

3.4.1.2. Project Scope

The purpose of the Healthcare AI system is to ease in diagnosing dreadful diseases such as Diabetic Retinopathy, Brain Tumor, Breast Lesions in UltraScans, Pneumonia via Xrays, Common pigmented skin lesions, Knee Osteoarthritis via XRays . Hence, deep learning helps doctors to analyze the disease better and provide patients with the best treatment.

3.4.2. Overall Description

3.4.2.1. Product Perspective

A software system made by applying deep learning algorithms that may be able to automatically detect abnormalities. It is steadily finding its way into innovative tools that have high-value applications in the real-world clinical environment. Medical reports (like MRIs, XRAYs) are used to classify or detect diseases well beforehand and prevent mishappenings.

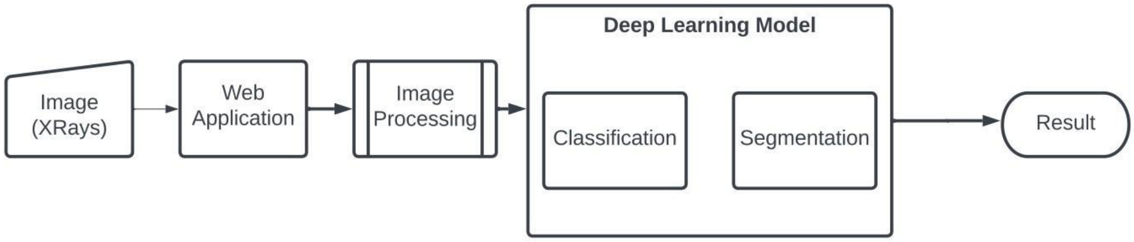

Fig 3.4.2.1.1 : Project Block Diagram

3.4.2.2. Product Functions

- **Input image**: The input Image is a XRay
- **Web Application:** A Web Application that takes the input and feeds to the deep learning model
- **Image Processing**: The image input is processed to appropriate dimensions
- **Deep Learning Model:** The deep learning model (CNN) takes input and computes and learns the features
    - **Classification**: Differentiates multiple instances of the report
    - **Segmentation**: Localisation of the feature of the report
- **Report Result**: Final Result that is studied before taking a final decision

3.4.2.3. Operating Environment

Operating environment for the HealthCare AI System

- Web Application
- Operating system: Windows,Linux,MacOS,wsl
- Language: python
- IDE: VS Code, Colab, Kaggle
- Libraries: Tensorflow, Streamlit
- Cloud Services : Azure, Heroku
- Containerization : Docker

3.4.3. External Interface Requirements

3.4.3.1. User Interfaces

- Front-end software: Streamlit (Python)
- Back-end software: Python

3.4.3.2. Hardware Interfaces

- Windows/Linux/MacOS
- Internet Connection.
- Every Browser

3.4.3.3. Software Interfaces

Following are the software used for

Table 3.4.3.3.1: Softwares Used

Software used	**Description**
Python	Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected
Tensorflow	TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks

Streamlit	Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time
Azure and Heroku	Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.
Docker	Docker is an open source containerization platform. It enables developers to package applications into containers—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment

3.4.4. System Features

3.4.4.1. Diseases Overview

The Application contains a brief Description discussing the Symptoms,causes and prevention techniques along with a Youtube Video explaining about the diseases : Diabetic Retinopathy,Brain Tumor,Breast Lesions in UltraScans,Pneumonia Chest via XRay, Common pigmented skin lesions, Knee Osteoarthritis via XRay

3.4.4.2. Diagnosis of the reports

Multiple Deep Learning models were devised to classify the presence of the disease , differentiate their severity and localize the affected part of the organ/tissue.

4. SYSTEM DESIGN

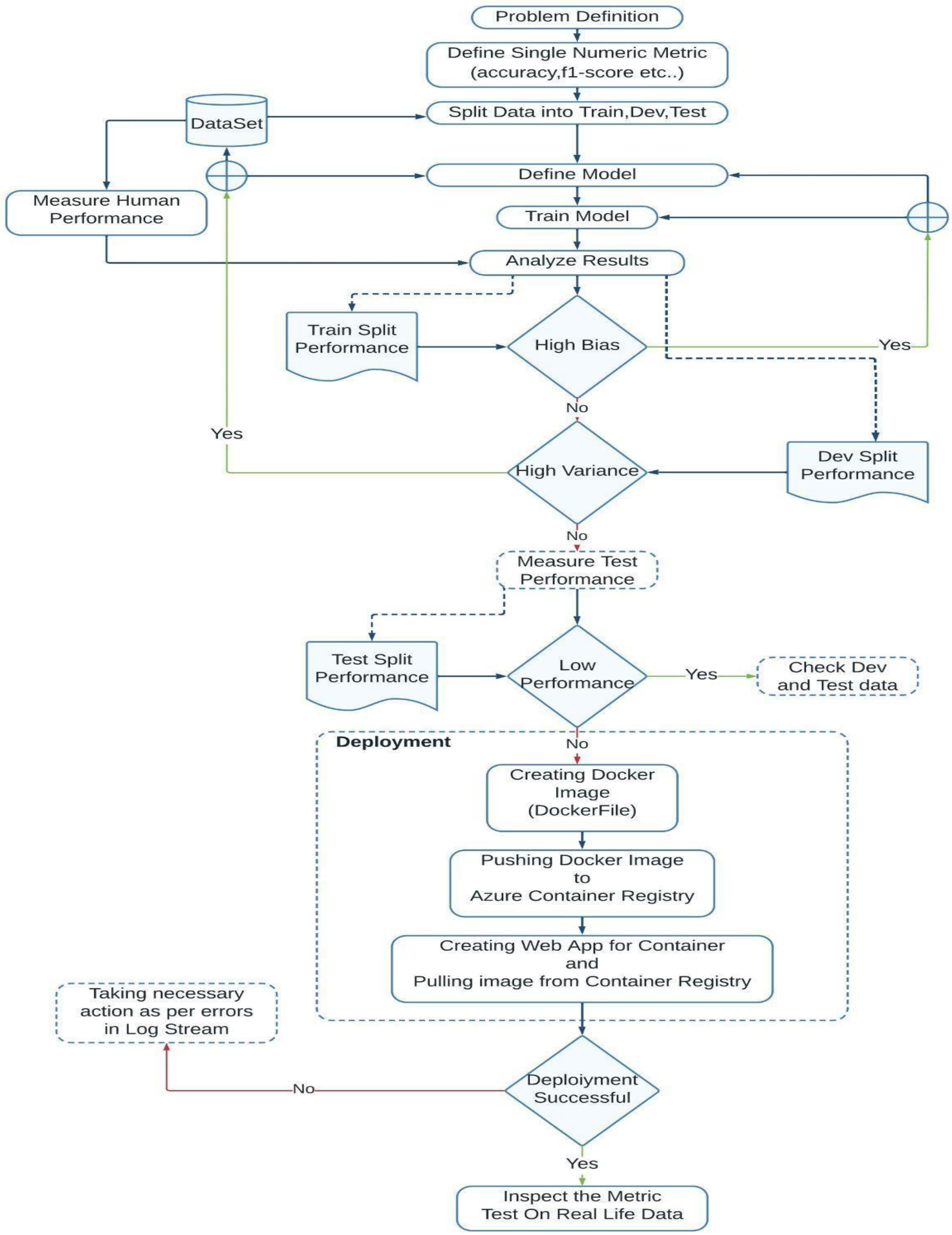

Fig 4.1 : System Design

4.1. System Design Description

Collection of data

- As Data is very crucial in a computer vision problem , it should be in correct usable format and Up-to-date with the market trends.
- Data should be collected from verified and reputable sources only.

Understanding the data

- It is important to know about the data used.
- Brief Domain Knowledge is required to solve (in our case Medical Knowledge).

Identifying the type of problem

- Is the data available suited for a classification or segmentation problem ?
- Identify the type of disease or detecting the location of the abnormality

Creating a baseline architecture

- Creating a basic Neural Network to achieve a baseline score
- Deciding the appropriate metrics to measure performance

Improving the baseline score

- Creating a better architecture with the help of 'State Of The Art Neural Networks' by Transfer Learning.
- Achieving a better score.

Testing the models performance

- Testing the created model using unknown data to understand how it works on real-time data.
- Monitoring the models performance and modifying if required.

Creating a web application

- Creation of a Web Application for the end users.
- Streamlit python package used to create the web front.

Testing the web application

- Testing the behaviour of the web application when deployed using ngrok
- Making changes if necessary

Containerizing the application

- Containerizing the entire application using Docker to avoid any software dependencies conflicts
- Dockerized applications are easier to work with

Deployment

- Pushing the Docker Image of the Application on to Azure container registry
- Creating the web application by pulling the image from the container registry.

Architectures used

DenseNet

DenseNet is a network architecture where each layer is directly connected to every other layer in a feed-forward fashion (within each *dense block*). For each layer, the feature maps of all preceding layers are treated as separate inputs whereas its own feature maps are passed on as inputs to all subsequent layers. This connectivity pattern yields state-of-the-art accuracies on CIFAR10/100 (with or without data augmentation) and SVHN. On the large scale ILSVRC 2012 (ImageNet) dataset, DenseNet achieves a similar accuracy as ResNet, but using less than half the amount of parameters and roughly half the number of FLOPs.

MobileNetv2

MobileNetv2 paper introduces a new neural network architecture that is specifically tailored for mobile and resource constrained environments. This network pushes the state of the art for mobile tailored computer vision models, by significantly decreasing the number of operations and memory needed while retaining the same accuracy.

UNet

The typical use of convolutional networks is on classification tasks, where the output to an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization, i.e., a class label is supposed to be assigned to each pixel. Moreover, thousands of training images are usually beyond reach in biomedical tasks. Hence, Ciresan et al. trained a network in a sliding-window setup to predict the class label of each pixel by providing a local region (patch) around that pixel as input. First, this network can localize. Secondly, the training data in terms of patches is much larger than the number of training images.

5. IMPLEMENTATION

- Implemented a software system as a Web Application available to the paramedic/Jr Doctor which in the absence of the specialist helps understand the result in the state of an emergency which can help in Early Detection of medical reports and it is very crucial to help in preventing risky complications (even death).
- Achieved the solution to the proposed system applying multiple models for multiple diseases.
- Modular code is maintained to allow development to be divided by splitting down a program into smaller programs in order to execute a variety of tasks. This has enabled us to work simultaneously and minimize the time taken for development.

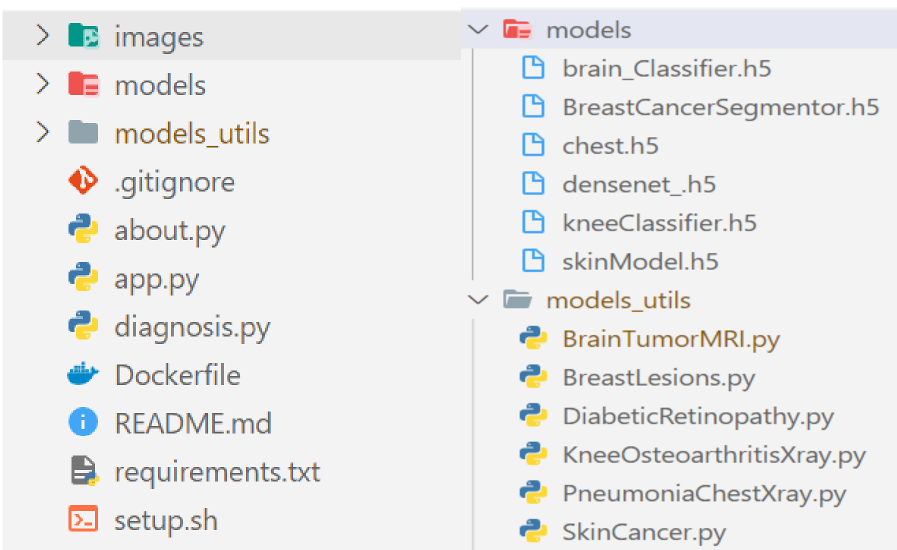

Fig 5.1 : Project Directory

5.1. Web Application sample

- The User Interface of the Web Application was created using Streamlit.
- Objects of every disease were instantiated for simplicity.
- Multiple Streamlit UI components were utilized to fulfill different requirements of the application

Code

```
from models_utils import BrainTumorMRI, DiabeticRetinopathy,BreastLesions,
KneeOsteoarthritisXray, PneumoniaChestXray, SkinCancer

DR=DiabeticRetinopathy.DiabeticRetinopathy_Model()
SC=SkinCancer.SkinCancer_Model()
PCX=PneumoniaChestXray.PneumoniaChestXray_Model()
BTM=BrainTumorMRI.BrainTumorMRI_Model()
BLT=BreastLesions.BreastLesions_Model()
KOX=KneeOsteoarthritisXray.KneeOsteoarthritisXray_Model()

    ## Diabetic-Retionapathy ##
    if diagnose == 'Diabetic-Retinopathy':
        DR.DiabeticRetinopathy_Predict()

    ## Skin-Cancer ##
    if diagnose == "Skin-Cancer":
        SC.SkinCancer_Predict()

    ## Pneumonia-Chest-XRAY ###
    if diagnose == "Pneumonia-Chest-XRAY":
       PCX.PneumoniaChestXray_Predict()


    ## Brain-Tumor-MRI ##
    if diagnose == "Brain-Tumor-MRI":
```

```
    BTM.BrainTumorMRI_Predict()
  ## Breast-Lesions-Tumor-Cancer ##
  if diagnose == 'Breast-Lesions-Tumor-Cancer':
    BLT.BreastLesions_Predict()


  ## Knee-Osteoarthritis-XRAY ##
  if diagnose == 'Knee-Osteoarthritis-XRAY':
    KOX.KneeOsteoarthritisXray_Predict()
```

5.2. Model Creation

- Tensorflow - a Deep Learning framework is used to create Convolutional neural networks (CNNs), which is particularly well-suited to analyzing images, such as MRI results or x-rays

Code

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Activation,Flatten,Dropout
from tensorflow.keras.layers import Conv2D,MaxPooling2D
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping

model=Sequential()

model.add(Conv2D(128,(3,3),input_shape=data.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
#The first CNN layer followed by Relu and MaxPooling layers

model.add(Conv2D(64,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
#The second convolution layer followed by Relu and MaxPooling layers
```

```
model.add(Conv2D(32,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
#The thrid convolution layer followed by Relu and MaxPooling layers

model.add(Flatten())
#Flatten layer to stack the output convolutions from 3rd convolution layer
model.add(Dropout(0.2))

model.add(Dense(128,activation='relu'))
#Dense layer of 128 neurons

model.add(Dropout(0.1))
model.add(Dense(64,activation='relu'))
#Dense layer of 64 neurons

model.add(Dense(5,activation='softmax'))
#The Final layer with two outputs for two categories

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

checkpoint = ModelCheckpoint("Classifier.h5",monitor = "val_accuracy",save_best_only =
True,verbose=1)

earlystop = EarlyStopping(monitor='val_accuracy',patience=10,verbose=1)

history=model.fit(x_train,y_train,
        epochs=100,
        validation_split=0.2,
        callbacks = [checkpoint,earlystop],
        verbose=1)
```

```
Model: "sequential"

Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 254, 254, 128)     1280

activation (Activation)      (None, 254, 254, 128)     0

max_pooling2d (MaxPooling2D) (None, 127, 127, 128)     0

conv2d_1 (Conv2D)            (None, 125, 125, 64)      73792

activation_1 (Activation)    (None, 125, 125, 64)      0

max_pooling2d_1 (MaxPooling2 (None, 62, 62, 64)        0

conv2d_2 (Conv2D)            (None, 60, 60, 32)        18464

activation_2 (Activation)    (None, 60, 60, 32)        0

max_pooling2d_2 (MaxPooling2 (None, 30, 30, 32)        0

flatten (Flatten)            (None, 28800)             0

dropout (Dropout)            (None, 28800)             0
...
Total params: 3,788,645
Trainable params: 3,788,645
Non-trainable params: 0
```

5.3. Loading the model and prediction

- Every disease's objects were instantiated and called along with their respective saved model.
- Image was preprocessed before being fed into the neural network as input

Code

```
if option == 'Model':

    brainClasses = ['Tumor Not Found', 'TUmor Found']
    brainModel = load_model('models/brain_Classifier.h5')

    st.title("Welcome to Brain Tumor CLassifier")
    st.header("Identify what's the MRI result!")

    if st.checkbox("These are the classes of BRAIN Tumor "):
        st.write(brainClasses)

    file = st.file_uploader("Please upload a BRAIN MRI Image", type=[
                            "jpg", "png", "jpeg"])

    if file is None:
        st.info("Please upload an Image file")
    else:
        image = Image.open(file)
        st.image(image, use_column_width=True)
        demo = np.array(image)
        demo = demo[:, :, ::-1].copy()
        demo = tf.image.convert_image_dtype(demo, tf.float32)
        demo = tf.image.resize(demo, size=[150, 150])
        demo = np.expand_dims(demo, axis=0)
```

```
    pred = brainModel.predict(demo)
    result = np.argmax(pred)
    st.write(result)
    if result == 0:
        st.write("Prediction : Normal ")
    elif result == 1:
        st.write("Prediction : BRAIN TUMOR FOUND")

def predict_sample_img(file_path_selected):
    image = Image.open(file_path_selected)
    #st.image(image, use_column_width=True)
    demo = np.array(image)
    demo = demo[:, :, ::-1].copy()
    demo = tf.image.convert_image_dtype(demo, tf.float32)
    demo = tf.image.resize(demo, size=[150, 150])
    demo = np.expand_dims(demo, axis=0)

    pred = brainModel.predict(demo)
    result = np.argmax(pred)
    st.write(result)
    if result == 0:
        st.write("Prediction : Normal ")
    elif result == 1:
        st.write("Prediction : BRAIN TUMOR FOUND")
```

5.4. Testing the behavior of the Web Application

- The behavior of Web Applications is ambiguous due to the large number of dependencies , hence before deploying it on the public cloud , it is tested using ngrok.

Code

```
!unzip /content/drive/MyDrive/DR_DETECT/ngrok-stable-linux-amd64.zip

!./ngrok authtoken <private token>

get_ipython().system_raw('/content/drive/MyDrive/DR_DETECT/ngrok http 8501 &')


!curl -s http://localhost:4040/api/tunnels | python3 -c \
  'import sys, json; print("Execute the next cell and the go to the following URL: " +json.load(sys.stdin)["tunnels"][0]["public_url"])'


!streamlit run ./app.py
```

5.5. Dockerfile

- A container is an isolated and self-contained portable computing environment. It contains everything an application needs to run, from binaries to dependencies to configuration files
- Dockerfile acts as a manifest file that has all commands which need to be run for building a given image.

Code

```
FROM python:3.8

COPY . /usr/app/
WORKDIR /usr/app/

RUN pip install -r requirements.txt \
   &&  apt-get update && apt-get install libsm6 libxrender1 libfontconfig1 libice6 ffmpeg libxext6
-y

CMD sh setup.sh \
   && streamlit run app.py
```

5.6. Result Analysis

Table 5.6.1 : Result Analysis

Disease	Images per Class	Architecture used	Train Score	Validation Score
Brain MRI	No- 98 YES- 155	MobileNet-v2	Loss: 0.3178 accuracy: 0.8571	Val_loss 0.1950 Val_accuracy 0.9200
Chest Pneumonia	Normal- 1583 Pneumonia- 4273	MobileNet-v2	Loss: 0.2355 accuracy: 0.8976	Val_loss 0.2811 val_accuracy 0.8594
Knee-OsteoArthritis	Normal- 514 Doubtful- 488 Mild- 232 Moderate- 221 Severe- 206	Custom	loss: 0.7531 accuracy: 0.6360	Val_loss 0.7796 val_accuracy 0.6189
Skin Lesions	Melanocytic nevi - 6705 Melanoma - 1113 Benign keratosis-like lesions - 1099 Basal cell carcinoma - 514 Actinic keratoses - 327 Vascular lesions - 142 Dermatofibroma - 115	Custom	Loss: 0.7457 accuracy: 0.7358	Val_loss 0.8075 val_accuracy 0.6870

Diabetic Retinopathy	No DR 1805 Mild 999 Moderate 370 Severe 295 Proliferative DR 193	DenseNet-121	Loss: 0.4067 accuracy: 0.8474	Val_loss 0.4511 val_accuracy 0.85091
Breast Lesion	Normal - 133 Benign 445 Malignant 210	UNET	Loss 0.0067	Val_loss 0.0193

6. TESTING

The reason behind testing was to find errors. Every program or software has errors in it, against the common view that there are no errors in it if the program or software is working. Executing the programs with the intention of finding the errors in it is therefore testing; hence a successful test is one which finds errors. Testing is an activity, however, it is restricted to being performed after the development phase is complete, but is carried out parallel with all stages of system development, starting with requirement specification.

Several issues are considered while testing a Deep Learning Project:

- Check the general logic of the model (not possible in the case of deep neural networks so go to the next step if working with a DL model).
- Control the model performance by manual testing for a random couple of data points.
- Evaluate the accuracy of the ML model.
- Make sure that the achieved loss is acceptable for your task.
- If you get reasonable results, jump to unit tests to check the model performance on the real data.

There are two general kinds of tests carried out :

6.1. Pre-train tests

This type of test is performed early on and allows you to catch bugs before running the model. They do not need training parameters to be run.

An example of a pre-train test is a program that checks whether there are any labels missing in your training and validation datasets.

Checking if an image is uploaded and is preprocessed before training

```
if file is None:
        st.info("Please upload an Image file")
    else:
        image = Image.open(file)
        st.image(image, use_column_width=True)
        demo = np.array(image)
        demo = demo[:, :, ::-1].copy()
        demo = tf.image.convert_image_dtype(demo, tf.float32)
        demo = tf.image.resize(demo, size=[150, 150])
        demo = np.expand_dims(demo, axis=0)
```

6.2. Post-train tests

These tests are performed on a trained model and check whether it performs correctly. They allow us to investigate the logic behind the algorithm and see whether there are any bugs there. There are three types of tests that report the behavior of the program:

- **Invariance tests.** Using invariance tests, we can check how much we can change the input without it affecting the performance of the model. We can pair up input examples and check for consistency in predictions. For example, if we run a pattern recognition model on two different photos of red apples, we expect that the result will not change much.
- **Directional expectation tests**. Unlike invariance tests, directional expectation tests are needed to check how perturbations in input will change the behavior of the model. For example, when building a regression model that estimates the prices of houses and takes square meters as one of the parameters, we want to see that adding extra space makes the price go up.
- **Minimum functionality tests.** These tests enable us to test the components of the program separately just like traditional unit tests. For example, you can assess the model on specific cases found in your data.
- **Unit test**: Check the correctness of individual model components.

- **Regression test**: Check whether your model breaks and test for previously encountered bugs.
- **Integration test**: Check whether the different components work with each other within your machine learning pipeline.

ngrok

- ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort.
- The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.
- Similar functionality can be achieved with Reverse SSH Tunneling, but this requires more setup as well as hosting of your own remote server.

The behavior of Web Applications is ambiguous due to the large number of dependencies , hence before deploying it on the public cloud , it is tested using ngrok.

Unit Tests were performed on individual Disease models, all the modules were individually tested following bottom to top approach, starting with the smallest and lowest modules and then testing one at a time

Once the unit test was over, all the disease modules were integrated for integration testing. External and internal interfaces are implemented and work as per design, the performance of the module is not degraded.

Code

```
!unzip /content/drive/MyDrive/DR_DETECT/ngrok-stable-linux-amd64.zip

!./ngrok authtoken <private token>

get_ipython().system_raw('/content/drive/MyDrive/DR_DETECT/ngrok http 8501 &')


!curl -s http://localhost:4040/api/tunnels | python3 -c \
   'import sys, json; print("Execute the next cell and the go to the following URL: " +json.load(sys.stdin)["tunnels"][0]["public_url"])'


!streamlit run ./app.py
```

7. SCREENSHOTS

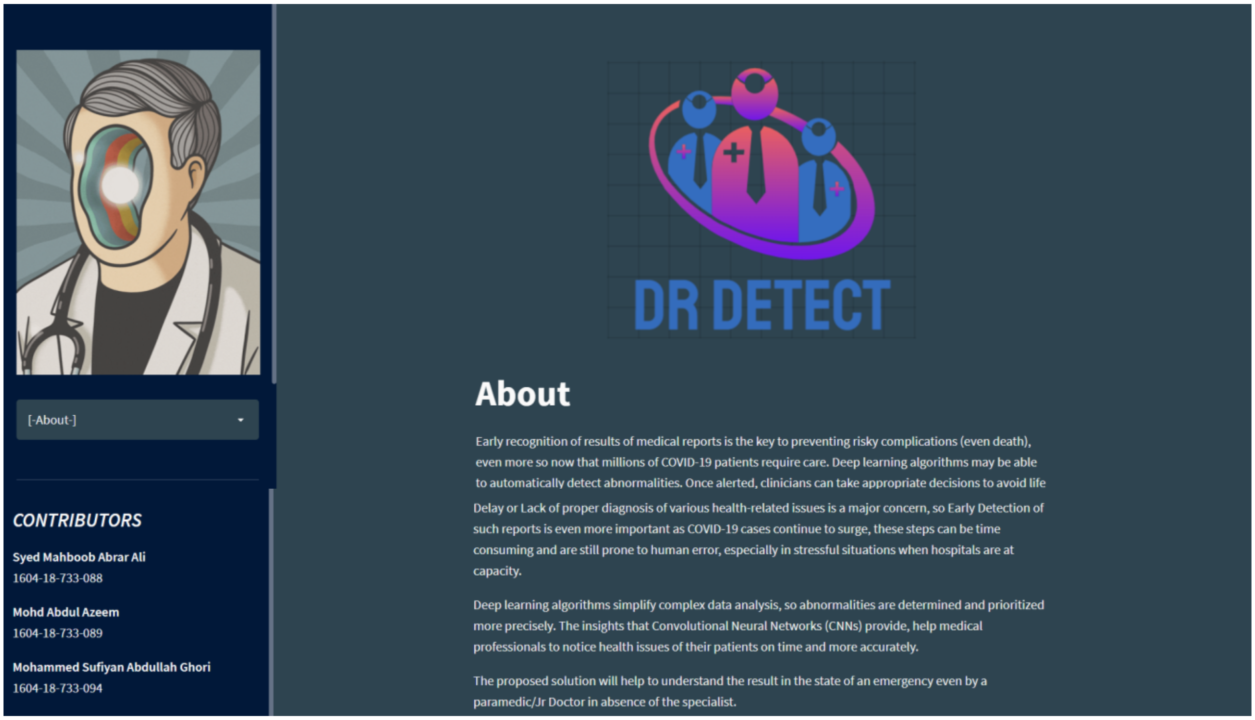

Fig 7.1 : About Page

The website has been deployed on two public clouds:

- Azure : https://drdetect.azurewebsites.net/
- Heroku : https://drdetect.herokuapp.com/

The sidebar consists of

- Drop Down consisting
  - About page : containsAbstract
  - Diagnosis page : it contains the dropdown for every disease
- Contributors list with roll numbers

A github repository of the project is attached. https://github.com/MSufiyanAG/dr_detect

The Application contains a brief Description discussing the Symptoms,causes and prevention techniques along with a Youtube Video explaining about the diseases with their respective models.

Fig 7.2 : Preface Page

Fig 7.3 : List of Diseases

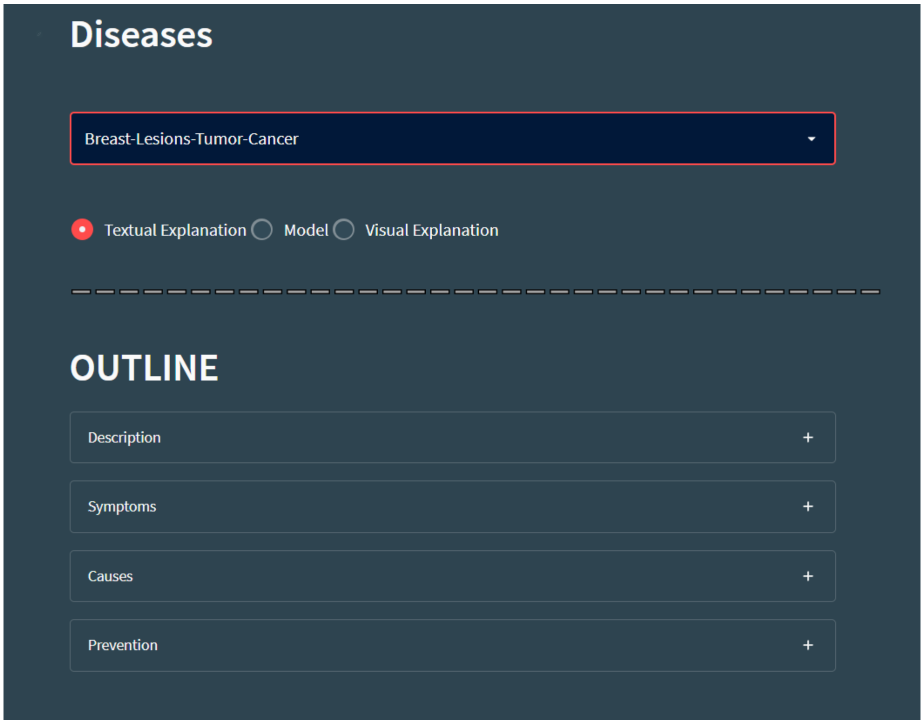

Fig 7.4 : Textual Explanation

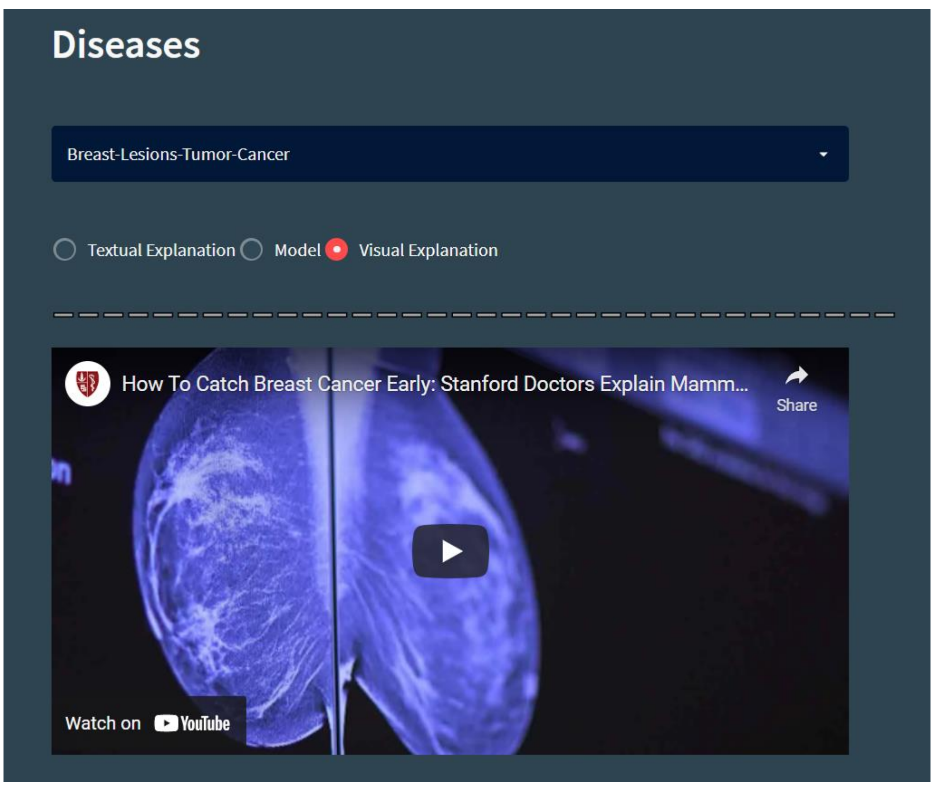

Fig 7.5 : Visual Explanation

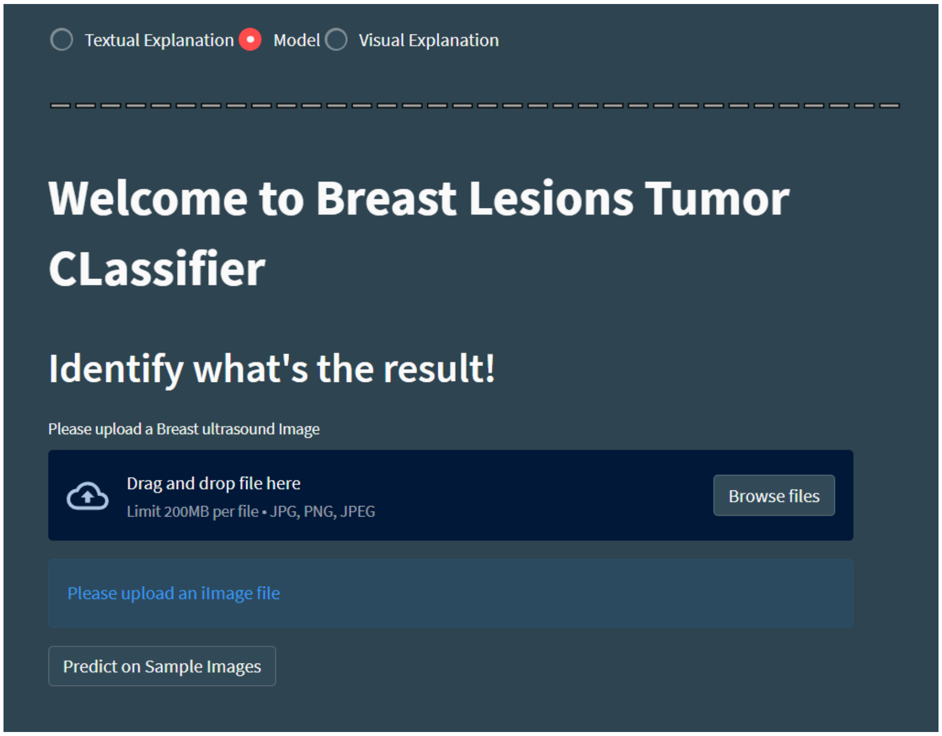

Fig 7.6 : Model Prediction

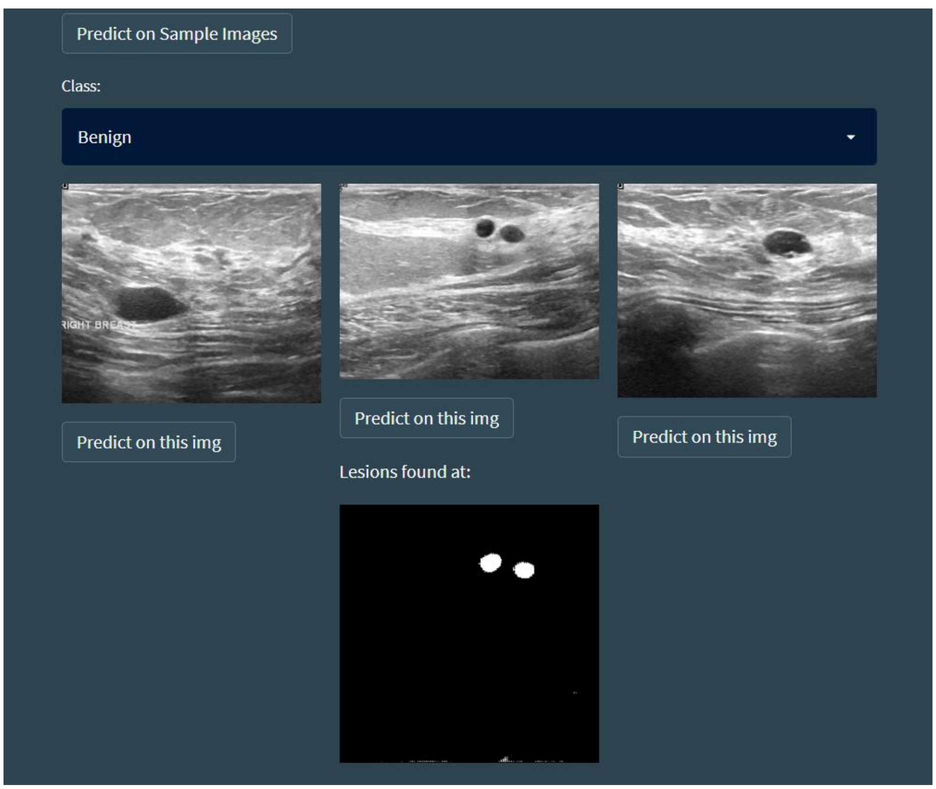

Fig 7.7 : Sample Prediction

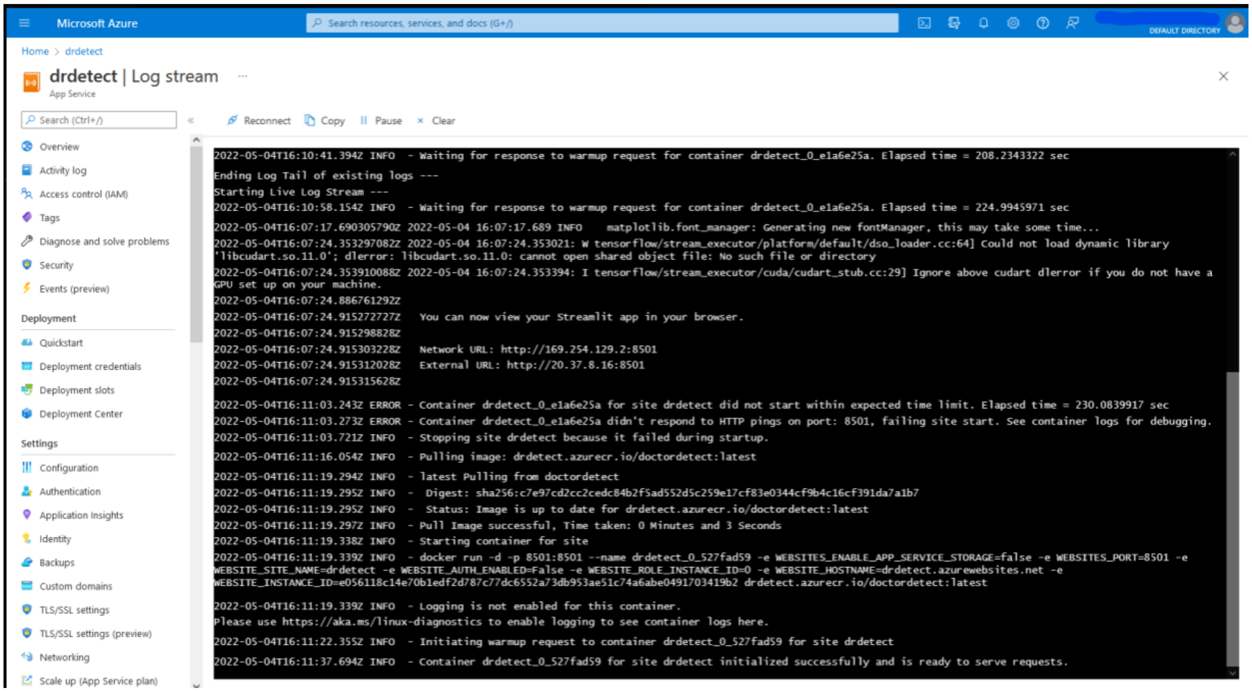

Fig 7.8 : Azure WebApp Log Stream

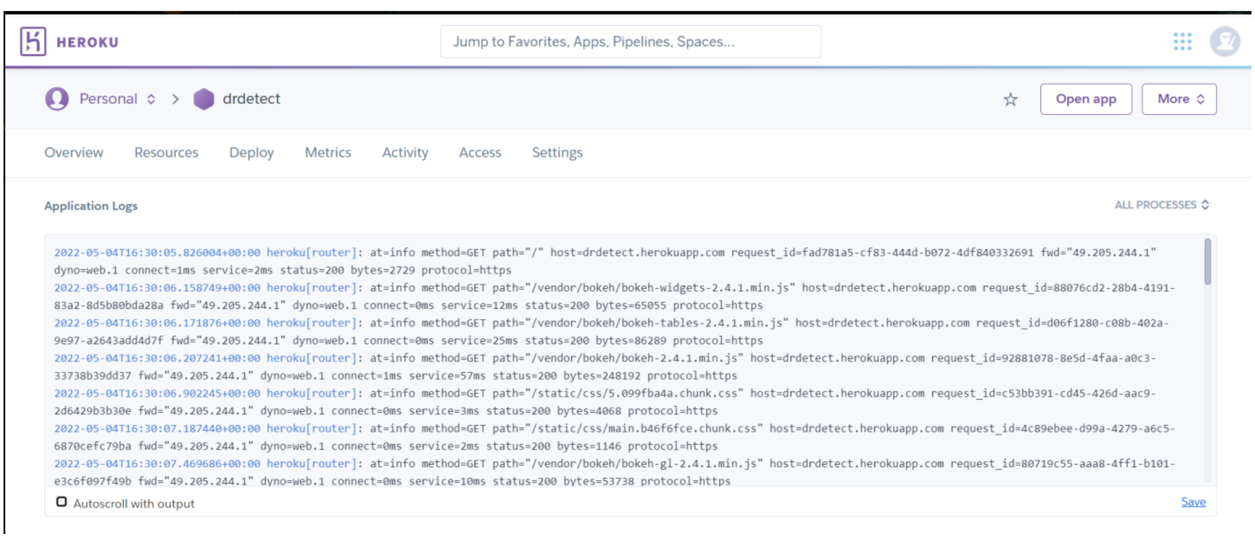

Fig 7.9 : Heroku WebApp Log Stream

8. CONCLUSION

Implemented a software system as a Web Application available to the paramedic/Jr Doctor which in the absence of the specialist helps understand the result in the state of an emergency which can help in Early Detection of medical reports and it is very crucial to help in preventing risky complications (even death).

Deep learning is assisting medical professionals and researchers to discover the hidden opportunities in data and to serve the healthcare industry better. Deep learning in healthcare provides doctors the analysis of any disease accurately and helps them treat them better, thus resulting in better medical decisions.

This technology can only benefit from intense collaboration with industry and specialist organizations. It needs to remain agile and able to adapt to ensure that it always remains relevant to the profession.

This targeted form of AI and deep learning helps the overburdened specialists by flagging items that are of concern and thereby allows the healthcare professional to direct patients with greater control and efficiency. It also reduces admin by integrating into workflows and improving access to relevant patient information. Once alerted, clinicians can take appropriate decisions to avoid life threatening complications

Successfully implemented an early diagnosis report analysis of 6 major diseases along with a brief overview of every disease. Made use of Deep Learning architectures to train, identify and diagnose medical reports like XRays, Imagery. The web application created using streamlit to serve in emergency conditions was containerized using Docker and pushed to public cloud services.

As intriguing as this pilot project can be, it represents only the very beginning of deep learning's role in healthcare analytics.

9. FUTURE PROSPECTS

1. A severity based appointment system :
    a. This type of application might help the reception team to assign appointments based on the severity of the patients reports.
    b. This will make sure there is no delay in access to healthcare.
2. Collaboration with Hospitals :
    a. For collection of vast annotated data for improved models
    b. This will help in building more precise models under the guidance of healthcare specialists.
3. Blend of Academia and Industry.
    a. Future of Healthcare tech solely depends on investing time and resources on Research and Development.
    b. This will encourage researchers to collaborate with organizations and thus lead to utilizing the research study on a large scale.
4. Creating an API for the trained models.
    a. This will inspire individual developers to contribute to open source software.
    b. Small teams can make use of it in accessing and interacting with trained models and external software components as per the requirement.

REFERENCES

[1] : Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*. 2016;316(22):2402–2410. doi:10.1001/jama.2016.17216

[2] : Hassan Ali Khan, Wu Jue, Muhammad Mushtaq, Muhammad Umer Mushtaq. Brain tumor classification in MRI image using convolutional neural network[J]. Mathematical Biosciences and Engineering, 2020, 17(5): 6203-6216. doi: 10.3934/mbe.2020328

[3] : Guo Y, Duan X, Wang C, Guo H (2021) Segmentation and recognition of breast ultrasound images based on an expanded U-Net. PLoS ONE 16(6): e0253202. https://doi.org/10.1371/journal.pone.0253202

[4] : Kundu R, Das R, Geem ZW, Han G-T, Sarkar R (2021) Pneumonia detection in chest X-ray images using an ensemble of deep learning models. PLoS ONE 16(9): e0256630. https://doi.org/10.1371/journal.pone.0256630

[5] : S. Kaymak, P. Esmaili and A. Serener, "Deep Learning for Two-Step Classification of Malignant Pigmented Skin Lesions," 2018 14th Symposium on Neural Networks and Applications (NEUREL), 2018, pp. 1-6, doi: 10.1109/NEUREL.2018.8587019.

[6] : Antony, J., McGuinness, K., Moran, K., O'Connor, N.E. (2017). Automatic Detection of Knee Joints and Quantification of Knee Osteoarthritis Severity Using Convolutional Neural Networks. In: Perner, P. (eds) Machine Learning and Data Mining in Pattern Recognition. MLDM 2017. Lecture Notes in Computer Science(), vol 10358. Springer, Cham. https://doi.org/10.1007/978-3-319-62416-7_27

[7] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[9] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[10] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

[11] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520. 2018.