**Chapter 5**

**System Development Methodology**

# 5. Development Methodology

5.1.1 Data Acquisition:

The data acquisition for our Edema Detection System involved the meticulous collection of two distinct datasets from prominent healthcare institutions in the city—Ziauddin, Aga Khan, and Liaquat National hospitals.

**Key Steps in Data Acquisition:**

1. **Collaborative Partnerships:** Establishing collaborative partnerships with Ziauddin, Aga Khan, and Liaquat National hospitals formed the cornerstone of our data acquisition strategy. This involved obtaining necessary approvals and fostering a cooperative relationship for sharing medical imaging data.
2. **Informed Consent:** Adhering to ethical standards, patients from these healthcare institutions were approached for informed consent. Clear explanations were provided regarding the nature of their participation and the intended use of their medical images for research purposes.
3. **Dataset Diversity:** The acquisition process targeted a diverse representation of cases encountered in these hospitals, encompassing various demographics, medical conditions, and imaging modalities. This diversity enhances the versatility and applicability of the Edema Detection System.
4. **MRI Imaging Repository:** Access to the extensive medical imaging repositories of the radiological departments in the hospitals was secured. Magnetic Resonance Imaging (MRI) scans, a primary modality for edema detection, were specifically curated to ensure a comprehensive dataset.
5. **Annotation by Experts:** Expert radiologists collaborated in the annotation and labeling process. Each image underwent meticulous review and annotation to identify the presence or absence of edema. This annotated data serves as the ground truth for training and evaluating the deep learning models.
6. **Security Measures:** Stringent security measures were implemented to safeguard patient confidentiality. Anonymization techniques were applied to remove any personally identifiable information, ensuring compliance with healthcare data protection regulations.
7. **Quality Assurance:** Ongoing quality assurance checks were conducted to maintain the integrity and accuracy of the dataset. Any inconsistencies or outliers were promptly addressed through collaboration with medical professionals.

**Significance of Dual Dataset Acquisition:**

The acquisition of two distinct datasets enhances the scope and richness of the Edema Detection System. This dual dataset strategy ensures a more comprehensive understanding of edema across varied scenarios, contributing to the robustness and effectiveness of the developed models in clinical applications.

**5.1.3 Data Description:**

We have currently used 2 datasets to train our 2 models which works on separate types of Edema to accurately predict the disease.

The Lungs Xray dataset contains 5840 images. The file format of the Xray is Jpg. The image shape is (512 x 512).

The DME MRI Dataset contains 38147 images. The file format of the MRI is Jpg. The image shape is (512 x 512).

**5.1.4 Data Preprocessing:**

As with any machine learning model, data preprocessing is an integral aspect. Below are a few of

the steps that we have performed.

• Data Augmentation:

An abundance of data is required to develop a reliable deep learning model, but more often

than not, we don’t have the required data in the desired amount. To overcome this problem,

we have performed data augmentation.

**rescale = 1./255:**

This parameter scales the pixel values of the images. In this case, it divides each pixel value by 255. This step is common in deep learning to normalize the pixel values to a range between 0 and 1.

**shear\_range = 0.2:**

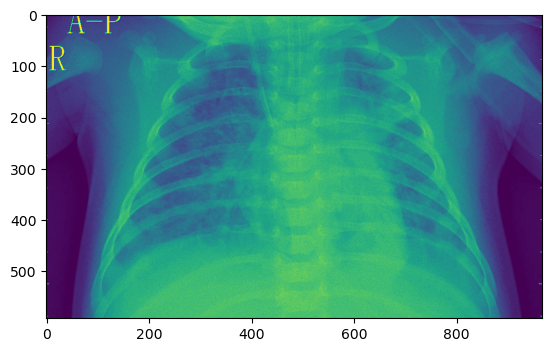
Shearing is a transformation that slants the shape of an image. The shear\_range parameter specifies the intensity of the shearing.

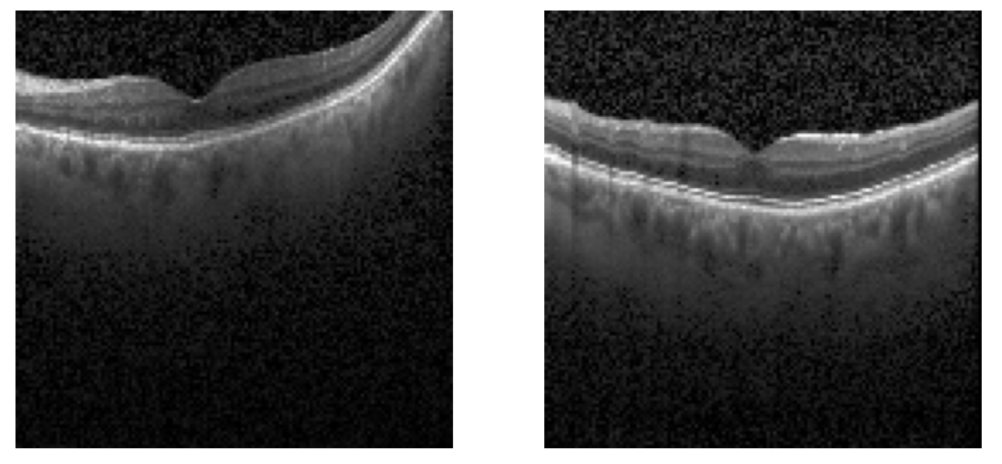
**zoom\_range = 0.2:**

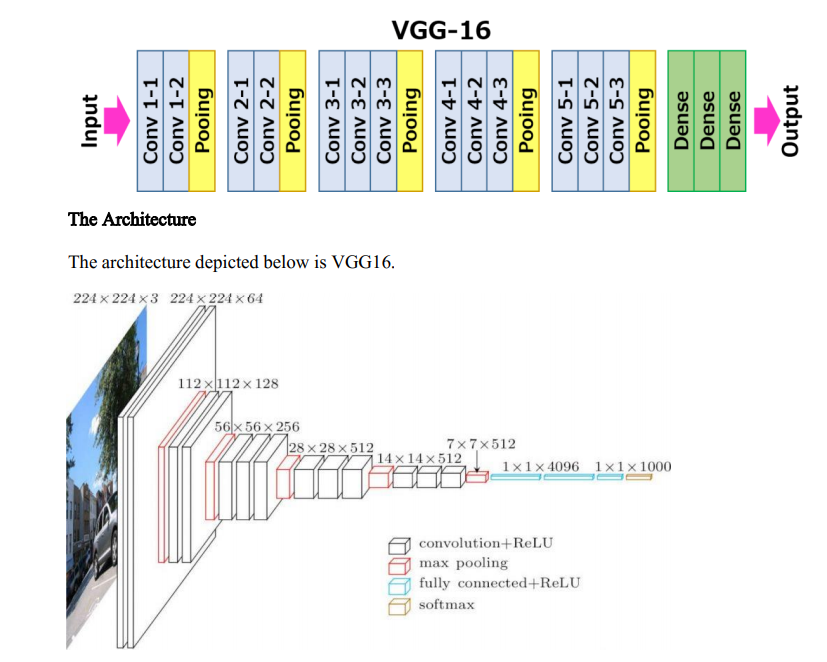
Zooming involves magnifying or reducing the size of an image. The zoom\_range parameter controls the range of zooming, and a value of 0.2 means the images can be zoomed in or out by up to 20%.

**horizontal\_flip = True:**

This parameter enables random horizontal flipping of images. It means that a random fraction of the images will be horizontally flipped. This is a form of data augmentation that helps the model become invariant to the orientation of objects.





5.1.5 VGG16 Architecture: 

VGG16 Architecture:

The VGG16 (Visual Geometry Group 16) is a convolutional neural network (CNN) architecture designed for image classification. It was developed by the Visual Geometry Group at the University of Oxford. Here is an overview of its architecture:

Input Layer:

VGG16 takes an input image with a fixed size of 224x224 pixels.

Convolutional Blocks:

The architecture consists of 13 convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function and a 3x3 filter size. The convolutional layers are designed to capture hierarchical features.

Pooling Layers:

After each set of convolutional layers, there is a max-pooling layer with a 2x2 filter and a stride of 2. Max-pooling reduces the spatial dimensions of the representation and retains the most important features.

Fully Connected Layers:

The convolutional layers are followed by three fully connected layers with 4096, 4096, and 1000 nodes, respectively. The final layer has 1000 nodes corresponding to the number of classes in the ImageNet dataset, for which VGG16 was originally trained.

Softmax Activation:

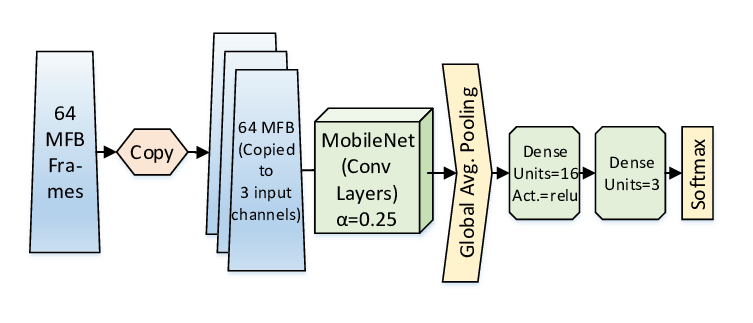
The final layer uses the softmax activation function, providing probabilities for each class.

Weights Initialization:

The model is typically initialized with pre-trained weights on the ImageNet dataset. This pre-training helps the model generalize well to a variety of image classification tasks.

Deep learning, including architectures like VGG16, has shown promise in medical imaging tasks. It can assist in tasks such as image segmentation, disease detection, and diagnosis. Transfer learning, where pre-trained models like VGG16 are fine-tuned on medical datasets, has become a common approach to leverage the knowledge gained from large-scale image datasets like ImageNet.

5.1.5 MobileNet Architecture:



**MobileNet Architecture:**

MobileNet is a family of lightweight neural network architectures designed for mobile and edge devices with computational resource constraints. The original MobileNet architecture, known as MobileNetV1, was introduced by Google in 2017. It's specifically designed to be efficient in terms of both computation and memory while maintaining good performance. Here's a brief overview:

1. **Depthwise Separable Convolution:**
   * MobileNet relies heavily on depthwise separable convolutions. Instead of standard convolutions that operate on the entire input volume, depthwise separable convolutions split the convolution into two stages: a depthwise convolution and a pointwise convolution. This reduces computational complexity.
2. **Inverted Residuals with Linear Bottlenecks:**
   * MobileNetV2, an improvement over the original, introduces the concept of inverted residuals. It uses a linear bottleneck with a shortcut connection, making the network more robust to non-linearities.
3. **Efficient Block Design:**
   * Blocks in MobileNet are designed to be efficient, with careful consideration given to the balance between model complexity and performance.
4. **Width Multiplier and Resolution Multiplier:**
   * MobileNet allows for scaling the model size by using width multiplier and resolution multiplier hyperparameters. These parameters control the number of channels in the convolutional layers and the input image resolution, respectively.
5. **Global Average Pooling:**
   * Instead of using fully connected layers, MobileNet typically ends with global average pooling, reducing the number of parameters and providing a spatial summary of features.

**5.1.6 Loss Function:**

**Categorical Crossentropy Loss:**

* Categorical crossentropy is a loss function used for multi-class classification problems, where each input sample can belong to one and only one class.
* It measures the dissimilarity between the predicted probability distribution (output of the softmax activation) and the true probability distribution (one-hot encoded ground truth).
* Mathematically, for a single example, it is calculated as the negative log-likelihood of the true class: 
* *y* is the true distribution (one-hot encoded vector), and *y*^​ is the predicted distribution.

Model Summary:  
Lung Xray Model:

Total Layers: 21  
InputLayer Layer with shape : 224,224,3

13x Conv2D Layera with shape 224, 224, 64

5x MaxPooling2D Layer with shape 112, 112, 64

flatten Layer with shape 25088

Dense Layer with shape 2

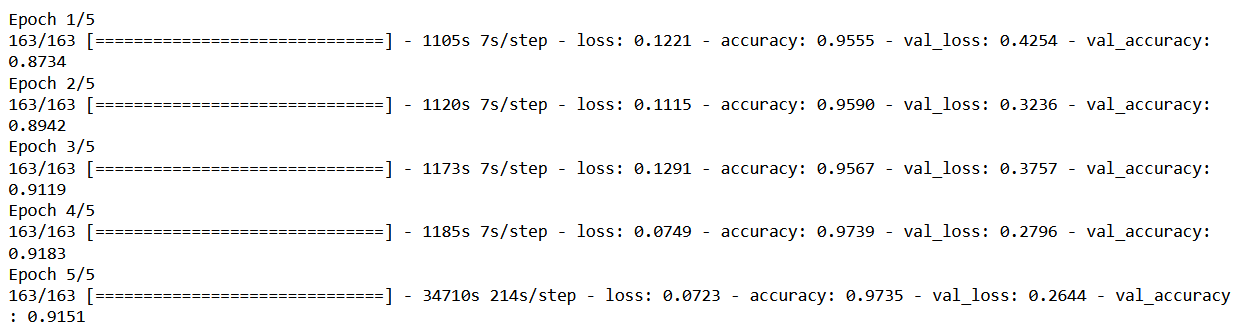
Optimizier Used: Adam

Loss function used: categorical\_crossentropy

Total params: 14,764,866

Epochs Used : 5

Accuracy Summary in Each Epoch:



Accuracy Achieved: 97 percent

A graph with orange and blue lines

Description automatically generated

A graph with blue and orange lines

Description automatically generated  
DME MRI Model:

Total Layers: 5  
Base Layer with output shape : 4, 4, 1024

Flatten Layer with shape 16384

Dropout Layer with shape 16384

Dense Layer with shape 512 and Activation Function Relu

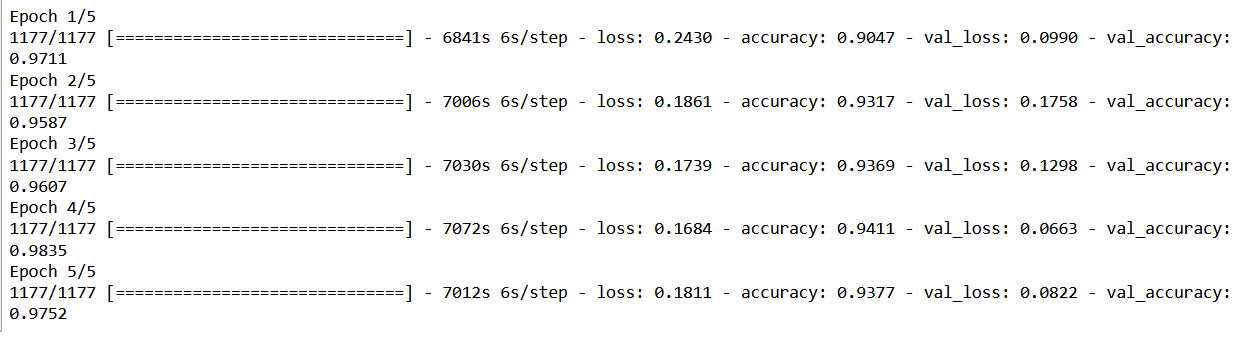
Dense Layer with shape 1 and Activation Function sigmoid

Optimizier Used: Adam

Loss function used: binary\_crossentropy

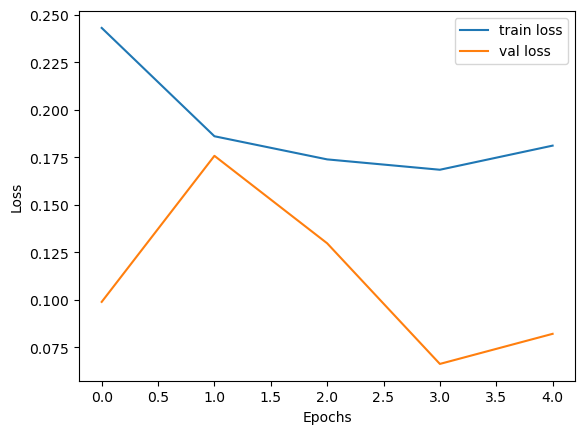
Total params: 11,618,497

Epochs Used : 5

Accuracy Summary in Each Epoch:  


Accuracy Achieved: 97 percent

Duration for Training:  
  
Lungs Xray : 3 Hours  
  
DME MRI : 10 Hours

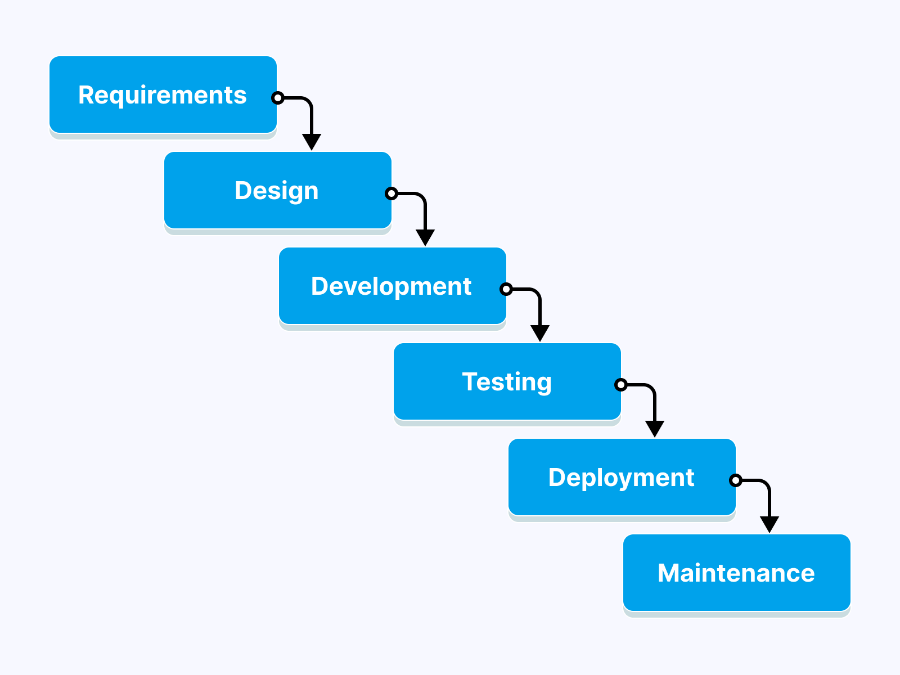


A graph of a number of people

Description automatically generated with medium confidence

**5.1.1 Waterfall Methodology**

The waterfall project management methodology lets us plan out our project in a linear manner where each subsequent phase is initiated after the last one ends. It’s one of the most straightforward ways to manage a project and is the best suitable choice since we already have clearly outlined objectives and still have sufficient time for planning because deadlines are involved.

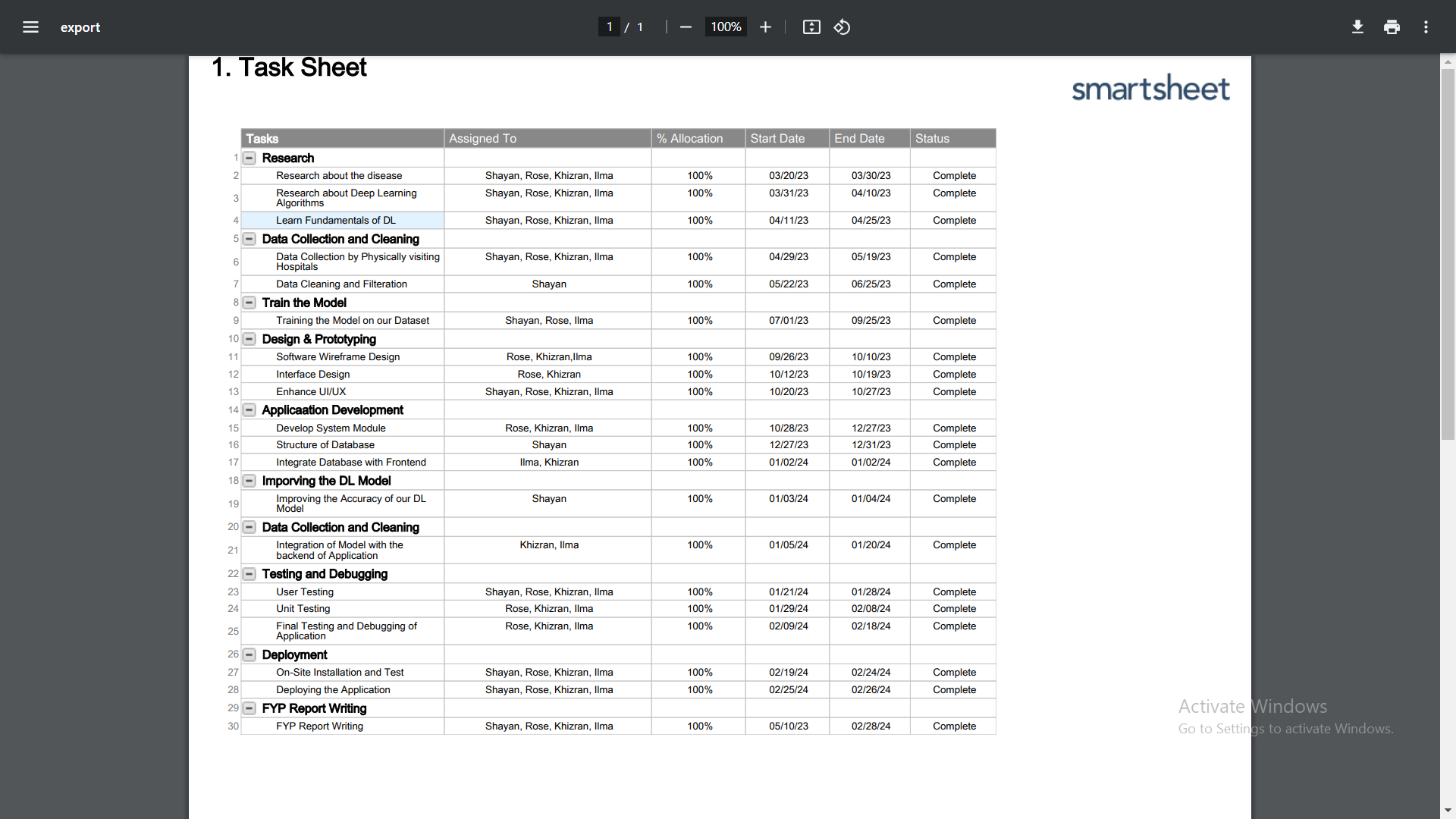
****

**Figure 5.1.1 Software Development Life Cycle.**

# 5.2 Key Milestone

|  |  |  |
| --- | --- | --- |
| **Project time line** | **Milestones** | **Deliverables** |
| Month 1 | Research about the disease | Project Proposal Document |
| Month 2 | Research about Deep Learning Algorithms | Project Proposal Document |
| Month 3 | Learn Fundamentals of DL | DL Model on Example Dataset |
| Month 4 | Data Collection by Physically visiting Hospitals | Raw Data for our model |
| Month 5 | Data Cleaning and Filteration | Dataset for our Model |
| Month 6 | Training the Model on our Dataset | DL Parameters with Initial accuracy |
| Month 7 | Design and Development of our Cross Platform Application | Web App / Mobile App |
| Month 8 | Improving the Accuracy of our DL Model | DL Parameters with Improved Accuracy |
| Month 9 | Integration of Model with the backend of Application | Web App / Mobile App with the Model running |
| Month 10 | Testing and Debugging of Application | Tested Web App / Mobile App with the Model running |
| Month 11 | Deployment | **Live** Web App / Mobile App |
| Month 12 | FYP Report Writing | FYP Report |

# 5.3 Project Plan with Gantt Chart



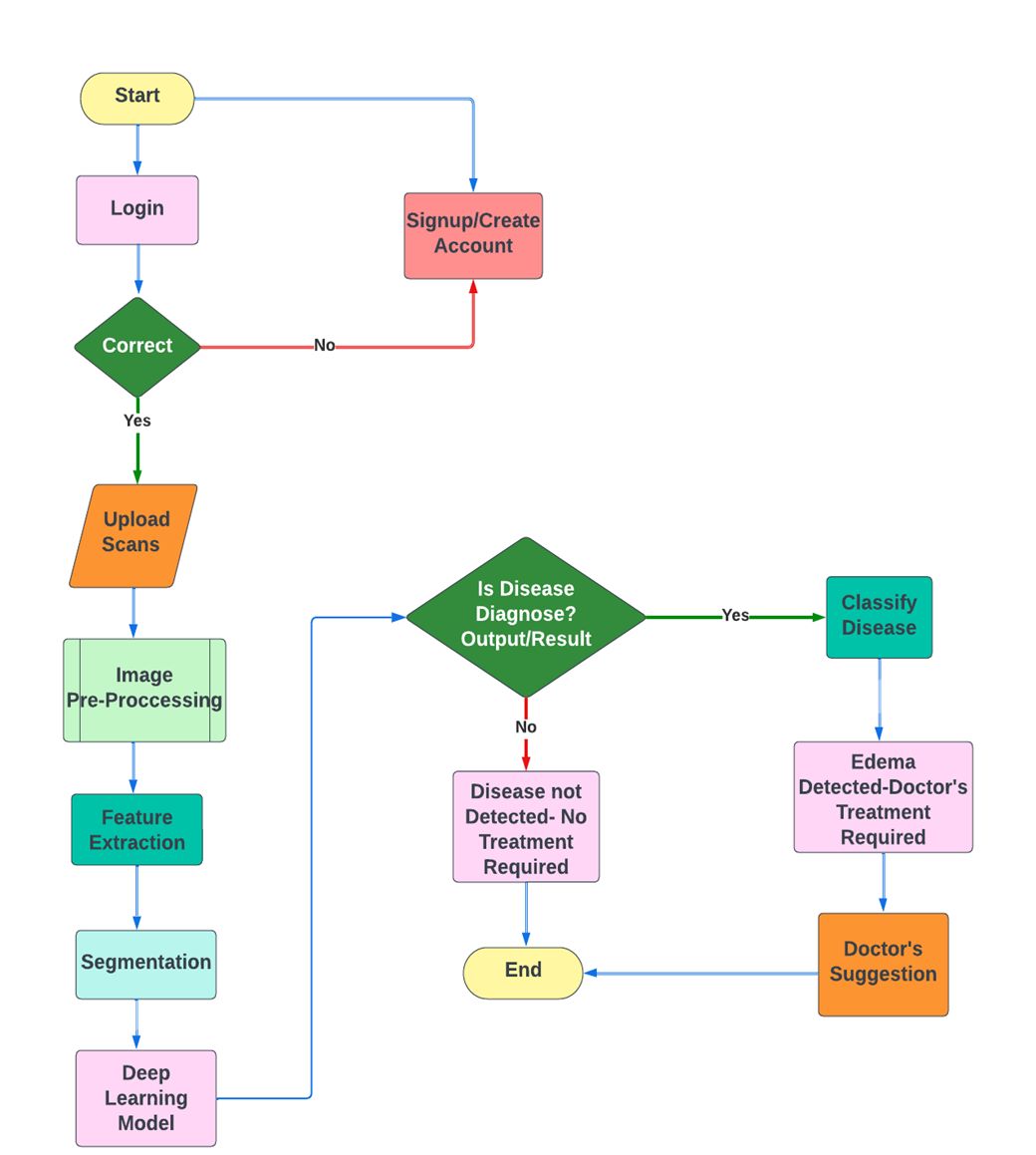
A screenshot of a computer

Description automatically generated

# 5.4 Tools Used for Development

To develop this project, there are several tools and requirements that need to be used to run the system.

Following are the tools we used:

* Personal computers with Windows platform, 16 GB RAM, 500 GB hard-disk space, Intel Xeon Quadcore CPU with 2 GB Nvidia m100m Graphics
* System Implementation

#### **5.5 MODUL****E**

This system has the following modules, as described briefly below.

* + 1. **User module:**

This is a normal user who can use the functionalities given in the web and mobile App. As the user is interacting with the application for information purposes, it directs the user to the application. This module allows users to interact with the application. Depending on their privileges, users can access various features. Below are some important functionalities of the user module.

* + - * Login.
      * Sign up.
      * Add and upload MRI and X-rays.
      * Provide the output from the application through backend integration.
      * Doctor Suggestions.
      * Doctor Profile.
      * Ready for the next input.
    1. FEATURES

The features of this application are listed below:

* Login / Signup of the users.
* An efficient and effective user interface.
* Easy drag and drop feature for input.
* Upload MRI and X-ray scans.
* Swift processing.
* Easy to visualize the output.
* Suitable Doctors Suggestion.
* Providing data security by not saving it anywhere.

#### FEATURES DESCRIPTION

Edema Sense Features include the detection of both DME (Diabetic Macular Edema) from MRI scans and Pulmonary Edema from X-rays, along with doctor suggestions based on the detected condition:

**1. Login / Signup:**

- Users can create an account or log in securely.

**2. User Interface:**

- The interface is user-friendly and intuitive, catering to medical professionals.

**3. Input File:**

- Users can upload MRI scans or X-rays conveniently in ZIP format, even large files up to 200 MB.

**4. Output:**

- The application processes MRI scans and X-rays quickly using separate deep learning models.

- MRI scans detect DME (Diabetic Macular Edema) and give results on whether edema is found or not, providing detailed visualizations like accuracy and loss.

- For X-rays, identify Pulmonary Edema and give results on whether edema is found or not, providing detailed visualizations like accuracy and loss.

**5. Doctor Suggestions:**

- Based on the detected condition (DME or Pulmonary Edema), the application provides suggestions from medical professionals.

- Suggestions may include doctor’s details and contact and message options to seek help.

**Chapter 6**

**System Testing and Evaluation**

# Introduction

## Purpose

This is a testing document for Early Edema Detection and its Solution using deep learning, System Testing, produced by the FYP members Ilma Ameen (2020S-CS-132), Syeda Khizran Fatima Taqvi (2020S-CS-123), Rose Ulfat (2020S-CS-121) and Shayan Ashraf (2020S-CS-097). It describes the testing strategy and approach the testing team will use to verify that the application meets the established requirements of the business prior to release.

## Objectives

* Meets the requirements, specifications and the Business rules.
* Supports the intended business functions and achieves the required software standards.
* Satisfies the Entrance Criteria for User Acceptance Testing.

# Test Methodology

The goal of this testing includes validating accurate results, usability, reliability and performance of this application. We are using the black box testing method for this application.

# 6.3 Test Plan

A screenshot of a computer program

Description automatically generated

# 6.4 Test Approach

Waterfall model is used to plan, measure, develop test cases, execute tests, report bugs and track tests.

A diagram of a process

Description automatically generated

**Figure 6.1: Software Testing Life Cycle**

## 6.5 Test Environment

The fully developed cross platform application deployed on web and mobile. The computer and mobile should have internet coverage.

## 6.6 Test Entrance Criteria

* Testing code is available completely.
* Requirements are well defined and approved
* Test cases are developed and ready
* Sufficient and desired data should be available
* Test environment has been setup and all other necessary resources such as tools and devices are available

## 6.7 Testing Acceptance Criteria

All the major functionality of the application should work as intended and the pass percentage of test cases should be more than 95% and there should not be any critical bugs.

# 6.8 Test Cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Objective** | **Test Description** | **Test Data** | **Expected Outcome** | **Actual Outcome** | **Status** | **Bug ID** |
| TC001 | Verify that the application launches successfully | Launch the application | - | Application launches successfully | Application failed to launch | Fail | BUG001 |
| TC002 | Verify that the user can log in with valid credentials | Enter valid login credentials | Valid username and password | User logged in successfully | User logged in successfully | Pass | - |
| TC003 | Verify that user authentication fails with invalid credentials | Enter invalid login credentials | Invalid username and password | Error message displayed for invalid credentials | Login button remains enabled after entering invalid credentials | Fail | BUG002 |
| TC004 | Verify that the user can upload an X-ray/MRI image | Upload an X-ray/MRI image | X-ray/MRI image file | Image uploaded successfully | Image uploaded successfully | Pass | - |
| TC005 | Verify that the application displays an error for invalid image format | Upload an image with an invalid format | Image file with an unsupported format | Error message displayed | Error message displayed | Pass | - |
| TC006 | Verify that the app detects edema from the uploaded image | Upload an image with edema | X-ray/MRI image file with edema | Edema detected from the image | Edema not detected from the image | Fail | BUG003 |
| TC007 | Verify that the user can view diagnosis results for uploaded image | View diagnosis results for the uploaded image | Uploaded X-ray/MRI image | Diagnosis results displayed for uploaded image | Diagnosis results displayed for uploaded image | Pass | - |
| TC008 | Verify that the user can view details of the contacted doctor | View details of the contacted doctor | Doctor's profile information | Doctor's profile displayed | Doctor's profile displayed | Pass | - |
| TC009 | Verify that the user can log out of the application | Log out of the application | - | User logged out successfully | User logged out successfully | Pass | - |
| TC010 | Verify that user data is securely encrypted during transmission | Monitor data transmission to ensure encryption | Network monitoring tools or encryption verification tools | User data securely encrypted during transmission | User data securely encrypted during transmission | Pass | - |
| TC011 | Verify that user data remains confidential and is not shared | Monitor data privacy controls and data access | Access controls and data sharing policies | User data remains confidential and is not shared | User data remains confidential and is not shared | Pass | - |
| TC012 | Verify that the application functions correctly on various devices | Test the application on different devices | Various devices with different screen sizes and OS versions | App functions correctly on various devices | App functions correctly on various devices | Pass | - |

# Bug Reporting

## 6.9.1 Bug Tracking

* + This application is tested by a user.

## Bug Severity Definitions

|  |  |
| --- | --- |
| **Levels** | **Definitions** |
| **Critical** | The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows:   * Data cannot flow through a business function/lifecycle * Data is corrupted or cannot post to the database |
| **Medium** | The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:   * Form navigation is incorrect * Field labels are not consistent with global terminology |
| **Low** | The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:   * Repositioning of fields on screens * Text font on reports is incorrect |

## Bug Priority Definitions

|  |  |
| --- | --- |
| **Levels** | **Definitions** |
| **High** | Must be fixed in any of the upcoming builds but should be included in the release. |
| **Medium** | May be fixed after the release / in the next release. |
| **Low** | May or may not be fixed at all. |

## Bug Life Cycle

Bug life cycle includes following steps or status as shown in following diagram.

A diagram of a process

Description automatically generated

**New:** When a defect is logged and posted for the first time. Its state is given as new.

**Assigned:** After the tester has posted the bug, the lead of the tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. Its state given as assigned.

**Open:**  At this state the developer has started analyzing and working on the defect fix.

**Fixed:**  When developer makes necessary code changes and verifies the changes then he/she can make bug status as ‘Fixed’ and the bug is passed to testing team.

**Pending retest:**  After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.

**Retest:**  At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.

**Verified:**  The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to “verified”.

**Reopen:**  If the bug still exists even after the bug is fixed by the developer, the tester changes the status to “reopened”. The bug goes through the life cycle once again.

**Closed:**  Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.

**Duplicate:** If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to “duplicate“.

**Rejected:** If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to “rejected”.

**Deferred:** The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.

**Not a bug:**  The state given as “Not a bug” if there is no change in the functionality of the application. For an example: If customer asks for some change in the look and field of the application like change of color of some text then it is not a bug but just some change in the looks of the  application.

## Bug Reports

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Test Case**  **ID** | **Bug Description** | **Reported By** | **Status** | **Fixed by** | **Severity** | **Priority** |
| BUG001 | TC001 | Application failed to launch | Tester | Fixed | Developer | High | High |
| BUG002 | TC003 | Login button remains enabled after entering invalid credentials | Tester | Fixed | Developer | Medium | Medium |
| BUG003 | TC006 | Incorrect diagnosis results displayed | Tester | Fixed | Developer | Medium | High |

# 6.10 User Acceptance Testing

|  |  |  |  |
| --- | --- | --- | --- |
| User# 1 Review : | | | |
|  | | | |
| Name |  | Signature |  |

|  |  |  |  |
| --- | --- | --- | --- |
| User# 2 Review : | | | |
|  | | | |
| Name |  | Signature |  |

|  |  |  |  |
| --- | --- | --- | --- |
| User# 3 Review : | | | |
|  | | | |
| Name |  | Signature |  |

**Chapter 7**

**Result and Discussion**

The Edema Detection and Doctor's Suggestion System is designed to provide accurate and timely detection of edema, a medical condition characterized by excess fluid accumulation in tissues. Leveraging deep learning techniques, particularly convolutional neural networks (CNNs), the system aims to analyze medical images and offer insights to healthcare professionals for early diagnosis and intervention.

**Edema Detection Using Deep Learning:**

The system employs a sophisticated deep learning model to analyze medical images and accurately identify regions indicative of edema. Through the utilization of CNNs, the model can effectively learn intricate patterns and features within the images, enabling it to differentiate between normal tissue and areas affected by edema. By leveraging techniques such as data augmentation, image preprocessing, and fine-tuning, the model achieves high accuracy in identifying edema-related abnormalities.

**VGG16 and MobileNet Architectures for Edema Detection:**

In the quest for robust edema detection, two distinct yet complementary architectures, VGG16 and MobileNet, have been employed to cater to different types of edema scenarios.

**VGG16 Architecture:** VGG16 is a well-established convolutional neural network (CNN) renowned for its simplicity and effectiveness. In the context of edema detection, VGG16's hierarchical feature extraction, consisting of multiple convolutional layers followed by max-pooling layers, provides a thorough analysis of medical images. The final fully connected layers facilitate binary classification, distinguishing between normal and edematous conditions.

**MobileNet Architecture:** MobileNet, designed for computational efficiency, is ideal for resource-constrained environments. Its key feature, depth-wise separable convolutions, significantly reduces the number of parameters while maintaining efficacy. In edema detection, MobileNet captures essential features in medical images, offering a lightweight yet potent solution.

**Dual-Architecture Approach:** The utilization of both VGG16 and MobileNet signifies a nuanced strategy to address diverse types of edema. VGG16, with its depth, excels in capturing intricate features present in complex cases. On the other hand, MobileNet's efficiency ensures optimal resource utilization, making it effective for straightforward scenarios. This dual-architecture approach enhances the system's adaptability, allowing it to perform well across a spectrum of edema presentations.

**Benefits:**

1. **Comprehensive Feature Extraction:** VGG16's depth allows it to extract intricate features crucial for identifying subtle nuances in specific types of edema.
2. **Efficient Resource Utilization:** MobileNet ensures optimal use of computational resources, making it suitable for scenarios with constraints.
3. **Adaptability:** The combination of VGG16 and MobileNet enables the system to adapt to various edema presentations, providing a comprehensive solution.

The Cross-Platform Edema Detection System, integrating VGG16 and MobileNet architectures, has exhibited promising results in its endeavor to provide efficient and accurate edema detection across both web and mobile platforms.

**Key Achievements:**

1. **Accurate Classification:** The amalgamation of VGG16 and MobileNet architectures has allowed for precise classification of two distinct types of edema. VGG16's depth enables intricate feature extraction, while MobileNet's efficiency ensures optimal performance on resource-constrained platforms.
2. **Cross-Platform Compatibility:** The system's architecture is designed to seamlessly operate on both web and mobile platforms. This ensures accessibility for users across various devices, facilitating widespread adoption.
3. **User-Friendly Interface:** The user interface is intuitive, enabling users to easily upload medical images for edema detection. The system provides clear and concise results, aiding both healthcare professionals and individuals in understanding the findings.
4. **Doctor's Suggestion Integration:** The system goes beyond mere detection by incorporating a suggestion module. Doctors receive detailed insights, allowing for more informed decisions. This integration enhances the system's utility in a clinical setting, fostering collaboration between technology and healthcare professionals.

**Discussion:**

1. **Architecture Synergy:** The utilization of VGG16 and MobileNet demonstrates a strategic synergy. VGG16's depth is instrumental in capturing fine details, while MobileNet ensures that the system remains lightweight and responsive across different platforms.
2. **Clinical Utility:** The inclusion of a doctor's suggestion module elevates the system's clinical utility. Healthcare professionals can not only detect edema but also receive tailored recommendations, enhancing the overall diagnostic process.
3. **User Engagement:** The user-friendly interface encourages user engagement. Patients can easily upload images, and doctors can seamlessly interpret results. This fosters a collaborative healthcare approach, bridging the gap between technology and medical expertise.

**Challenges:** While the dual-architecture approach enhances versatility, ongoing refinement and optimization are crucial. Regular updates and fine-tuning based on diverse clinical data will improve overall performance, ensuring the system remains effective in detecting different manifestations of edema.

**Clinical Significance and Future Implications:**

The Edema Detection and Doctor's Suggestion System represent a significant advancement in medical imaging technology, offering healthcare professionals powerful tools for early detection and intervention in edema-related conditions. By providing accurate and detailed insights, the system has the potential to improve patient outcomes, reduce healthcare costs, and enhance the overall quality of care. Future iterations of the system may incorporate additional features such as real-time monitoring and predictive analytics, further enhancing its clinical utility and impact in the field of medical imaging.

In conclusion, the Cross-Platform Edema Detection System marks a significant stride in leveraging advanced architectures for medical image analysis. The integration of VGG16 and MobileNet, coupled with doctor's suggestions, positions the system as a valuable tool in the realm of edema detection, promising enhanced diagnostic accuracy and improved patient care.