# INTELLIGENT RADIOLOGIST ASSISTANT

**An optimized web application designed specifically for classifying tears in knee MRI images.**

## LOW LEVEL DESIGN


**Domain:** Healthcare

**Technologies:** Deep Computer Vision

**Creator:** Arup Sankar Roy

**Date:** 20-05-2024

**Email:**asrl20221078.cse.uit@gmail.com

# DOCUMENT VERSION CONTROL

| Date issued | Version | Description | Author |
|---|---|---|---|
| February 29, 2024 | 1.1 | First Draft | Arup Sankar Roy |
| April 04, 2024 | 1.2 | Architecture & Architecture Description added and updated. | Arup Sankar Roy |
| May 07,2024 | 1.3 | Unit Test Cases defined and appended. | Arup Sankar Roy |

# CONTENTS

# Introduction

## What is Low-Level Design Document?

The goal of LLD or a low-level design document is to give the internal logical of the actual program code for Metro Interstate Traffic Volume Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.
The main objective of the project is to predict if traffic volume is in high or low on particular date. Weather circumstance, special days like holidays, daytime (morning, afternoon, night and etc.), a temperature, a weekday, a numeric percentage of cloud cover are vital attributes for predicting traffic volume.

## Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

# Project Summary

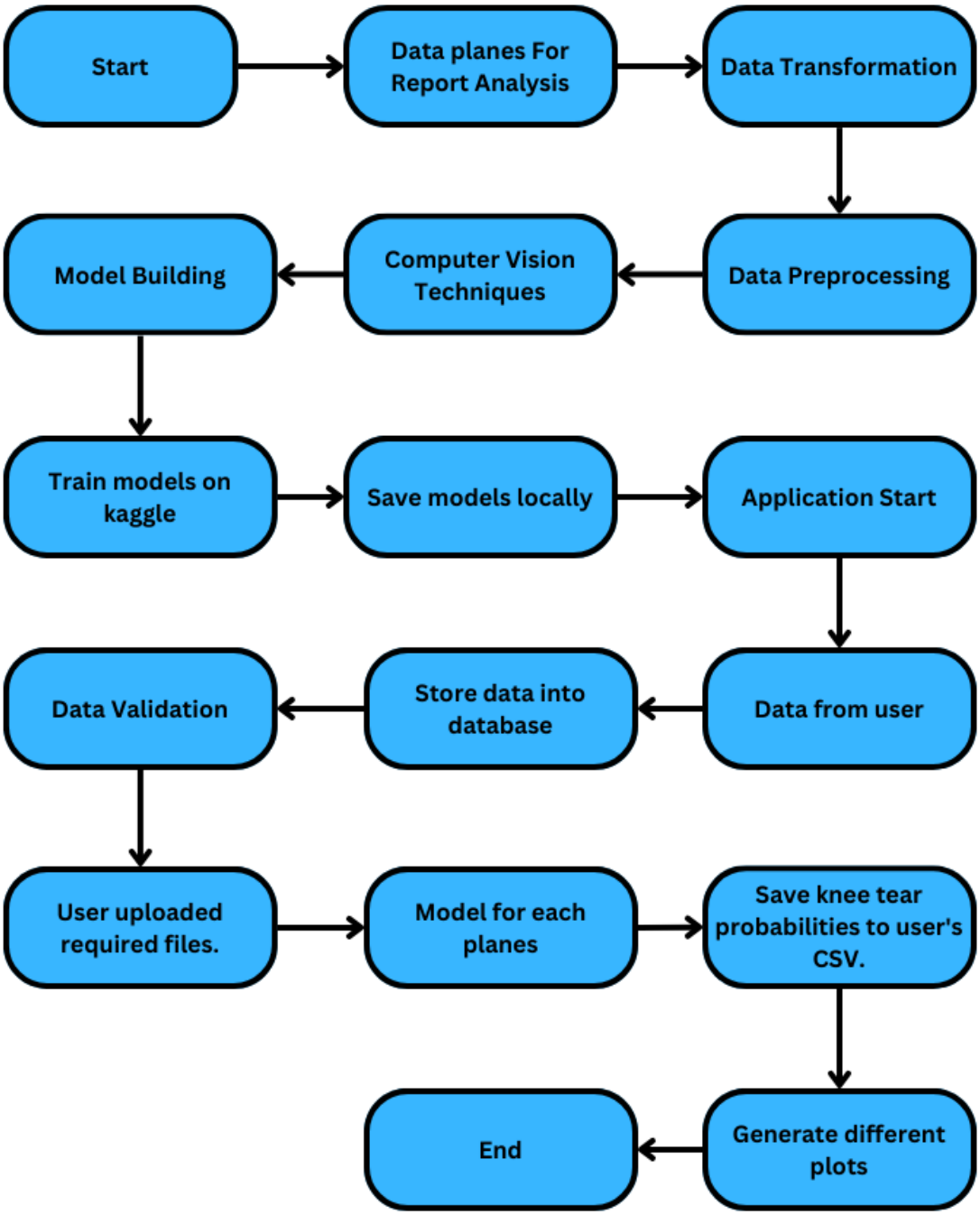
This project involves analyzing knee MRI planes using the MRNet dataset, which includes 1,370 exams with labels for abnormalities, ACL tears, and meniscal tears. The data is processed through axial, coronal, and sagittal planes, undergoing transformations such as reshaping and normalization.

Two models, MRNet and TripleMRNet, based on AlexNet and ResNet18 architectures, are used for analysis. The models are trained on Kaggle and then saved locally.

This application collects user information, validates it, and allows users to upload MRI plane files and a blank CSV. Separate models analyze each plane, and the resulting classification probabilities are saved into the user-provided CSV. The application generates various plots for detailed analysis of tear types.

3

# ARCHITECTURE



# Architecture Description

## Data Description:

The MRNet dataset, sourced from Stanford University Medical Center, includes 1,370 knee MRI exams. Among these, 1,104 exams (80.6%) are abnormal, with 319 (23.3%) ACL tears and 508 (37.1%) meniscal tears. Labels were manually extracted from clinical reports.

## Data Planes for Report Analysis:

MRI data is categorized into axial, coronal, and sagittal planes for further processing.

## Data Transformation & Preprocessing:

The transformation process involves:
Data reshaping
Normalization
Image augmentation

### Computer Vision Techniques:

Various computer vision techniques are applied for training purposes.

## Model Building:

Two primary models are used in this project:
1. MRNet: Analyzes single MRI planes based on the AlexNet architecture.
2. TripleMRNet: Processes MRI scans from three planes using AlexNet or ResNet18 as the backbone.

## Train moddels on Kaggle:

Model are trained on Kaggle. **LINK**

## Save Models Locally:

Post-training, models are downloaded and saved in the project directory, and the models folder is compressed into a .rar file.

## Application Start:

After completing the steps above, the MRI Assistant application is launched.

## Data Collection from User:

Personal information, including name, mobile number, and address, is collected from the user.

## Store Data into Database:

User data is stored locally in a database.

## Data Validation:

During login, the input data is validated against stored data for user authentication.

## File Upload by User:

Users upload required files, including a blank CSV and MRI planes (axial, coronal, sagittal).

## Model for Each Plane:

Separate models, saved earlier, are used for each plane.

## Save tear probabilities to user’s csv:

Prediction probabilities are saved into the blank CSV provided by the user.

## Generate Different Plots:

Based on the probabilities, various plots (bar, pie, line, stacked bar) are generated for a detailed analysis of tear types.

# UNIT TEST CASES

| NECESSARY FILES & TEST CASE DESCRIPTON | | PRE-REQUISITE | EXPECTED RESULT |
|---|---|---|---|
| Verify whether the Application's necessary files are Correctly pushed to github. | 1<br>2<br>3<br>4<br>5<br>6<br>7 | MODELS.rar<br>check_models.py<br>plots.py<br>model.py<br>predictions.py<br>templates folder<br>app.py | All are present and properly arranged. |
| Verify whether app download and setup instruction are correctly provided in github. | 1<br>2<br>3<br>4 | Database setup instruction<br>Cloning the repository instruction<br>Install dependencies insruction.<br>update app.py file instruction if there is any change in database by user. | All are present |
| Verify whether the User is able to sign up to the application. | 1 | Application is accessible | User is able to sign up to the application |
| Verify whether user is able to see input fields on logging in. | 1<br>2 | Application is accessible<br>User is logged in to the application | User should be able to see input fields on log in |
| Verify whether user is able to see input fields on registration page. | 1<br>2 | Application is accessible<br>User is signed up to the application | Yes user can manipulate these fields |
| Verify whether user gets Submit button to submit the inputs | 1<br>2<br>3 | Application is accessible<br>User is signed up to the application<br>User is login up to the application | yes user can submit the inputs. |
| Verify whether the Application's upload files page is properly taking the inputs. | 1<br>2 | page is accessible.<br>page is taking the inputs. | Page is accessible and able to take the inputs. |
| Verify whether user gets predict button to submit files and predict the results. | 1<br>2 | Application is accessible<br>User able to upload files to this page. | yes user can upload the files and predict button works fine. |
| Verify whether the Application's output page is correctly working. | 1<br>2 | page is accessible.<br>csv outputs are present. | yes user can access this page and see the results. |
| Verify whether the plots are present. | 1 | pie plot,bar plot,stacked bar,line plot must be present. | Yes all the plots are present. |