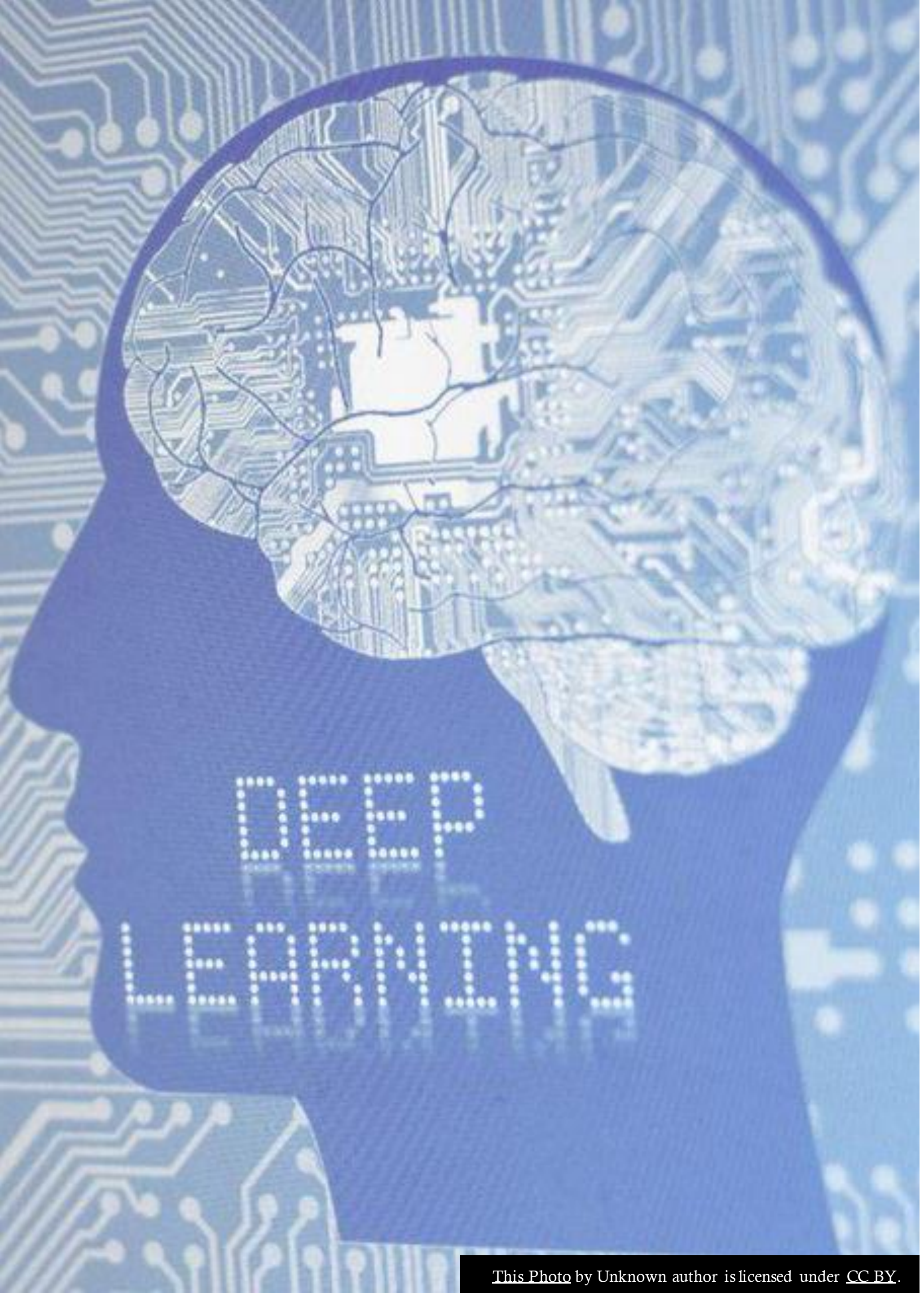


POTATO CROP DISEASE DETECTION WEBSITE USING DL AND REACTJS

TEAM B 08

IT – A

NGIT

NEIL GOGTE INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

POTATO DISEASE DETECTION WEBSITE USING DL AND REACTJS
INTERNAL GUIDE : MR. M MANOHAR RAO

C. Srividya R.	- 245319737009
M. Hari Priya	- 245319737035
P. Samhitha	- 245319737047

CONTENTS

- Introduction
- Existing System
- Proposed System
- Operating Environment
- Requirements
- System Design

INTRODUCTION

- In this project we try to detect different potato crop diseases.
- The detection of the disease is done by training the CNN model on the potato disease crop dataset that is made available by Plant Village.
- The trained CNN model is saved and is available through Fast Api server which is the backend.
- The website where the potato leaves are dropped in the drop-box for detection is created using ReactJs.
- Once the potato leaf image is placed in the drop-box , the disease is detected.

EXISTING SYSTEM

- In most developing countries, the existing method for detection of blight is performed manually where the parts of a potato plant have to undergo a sequence of several examinations under a controlled environment.
- This requires the establishment of laboratories in remote areas where access to various lab equipment is difficult and not environmentally friendly .
- The main aim is to address the above issues and to improve the quality and quantity of potato production by using advanced technologies.
- Digitalizing the identification of disease helps farmers to examine their crops without much effort and saves time.

PROPOSED YSTEM

- It is statistically observed that image processing is the best solution for detecting and analyzing potato crop diseases.
- This project aims to speed up the process of potato crop disease detection using advanced Deep Learning techniques.
- It also provides a user-friendly UI through which a potato leaf image is processed and the prediction is displayed.

OPERATING ENVIRONMENT

SOFTWARE REQUIREMENTS

- Operating System: Any updated Operating System
- Frontend: React JS, React Native
- Languages used: Python
- Libraries: TensorFlow, Fastapi

HARDWARE REQUIREMENTS

- Processor: Intel Pentium Dual Core Processor
- Hard Disk: 5GB
- RAM: 4GB
- Speed : 3 GHz(min) //Has to be checked

REQUIREMENTS

Functional
And
Non – Functional
Requirements

FUNCTIONAL REQUIREMENTS

- The input for the website shall be a jpg, png or jpeg image of a potato leaf.
- The input must be processed and disease should be detected by the saved DL model through FastApi server.
- The output should be the predicted result of whether the potato crop has:
  - Early blight disease
  - Late blight disease
  - Or is Healthy
- The output along with the disease should also provide accuracy of the prediction.

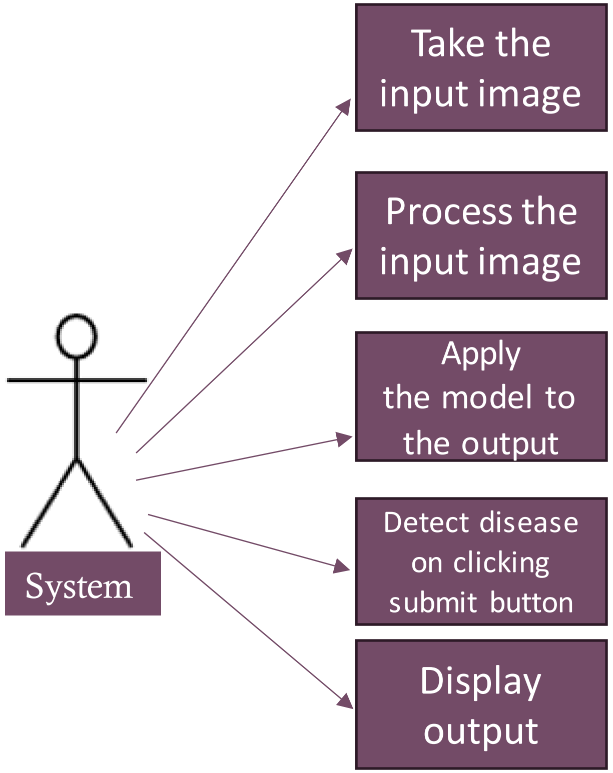
NON – FUNCTIONAL REQUIREMENTS

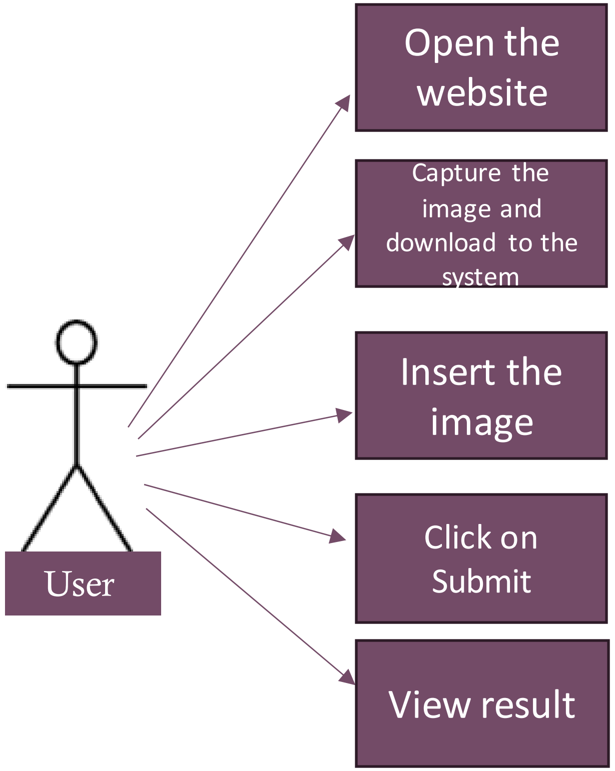
- Performance Requirements - The Disease Detection should happen upon user's action of providing an image of potato leaf.
- Response time - The average response time should be less than 3 seconds.
- Capacity - The website can only take one image at a time.
- Utilization of resources - We can have as many potato leaf images as we desire given we have enough storage capacity to handle so many images.

Software quality attributes:

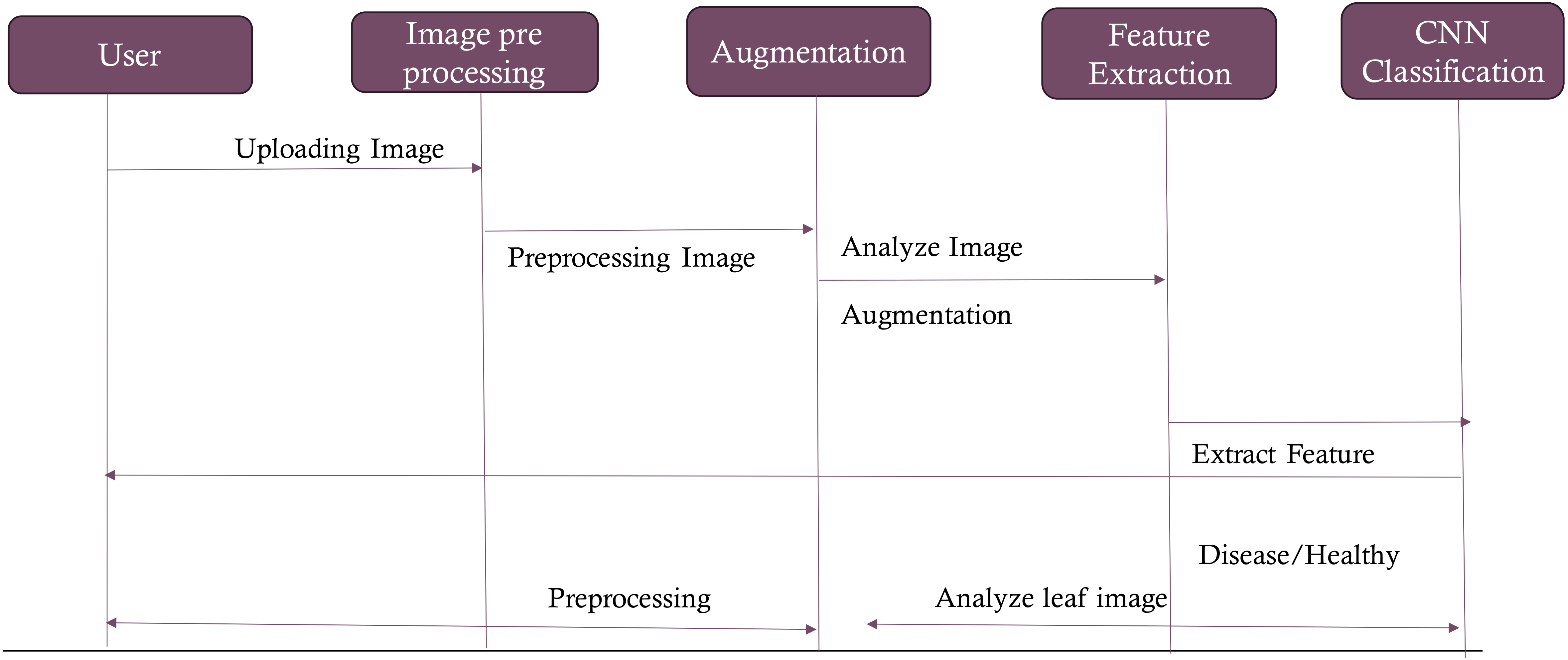
- Scalability : It is extensible to detect other crops also
- Availability : The website is available to all the users 24x7.
- Usability : The website is user-friendly which makes it easier for the users to use without the need of any technical explanation.

SYSTEM DESIGN

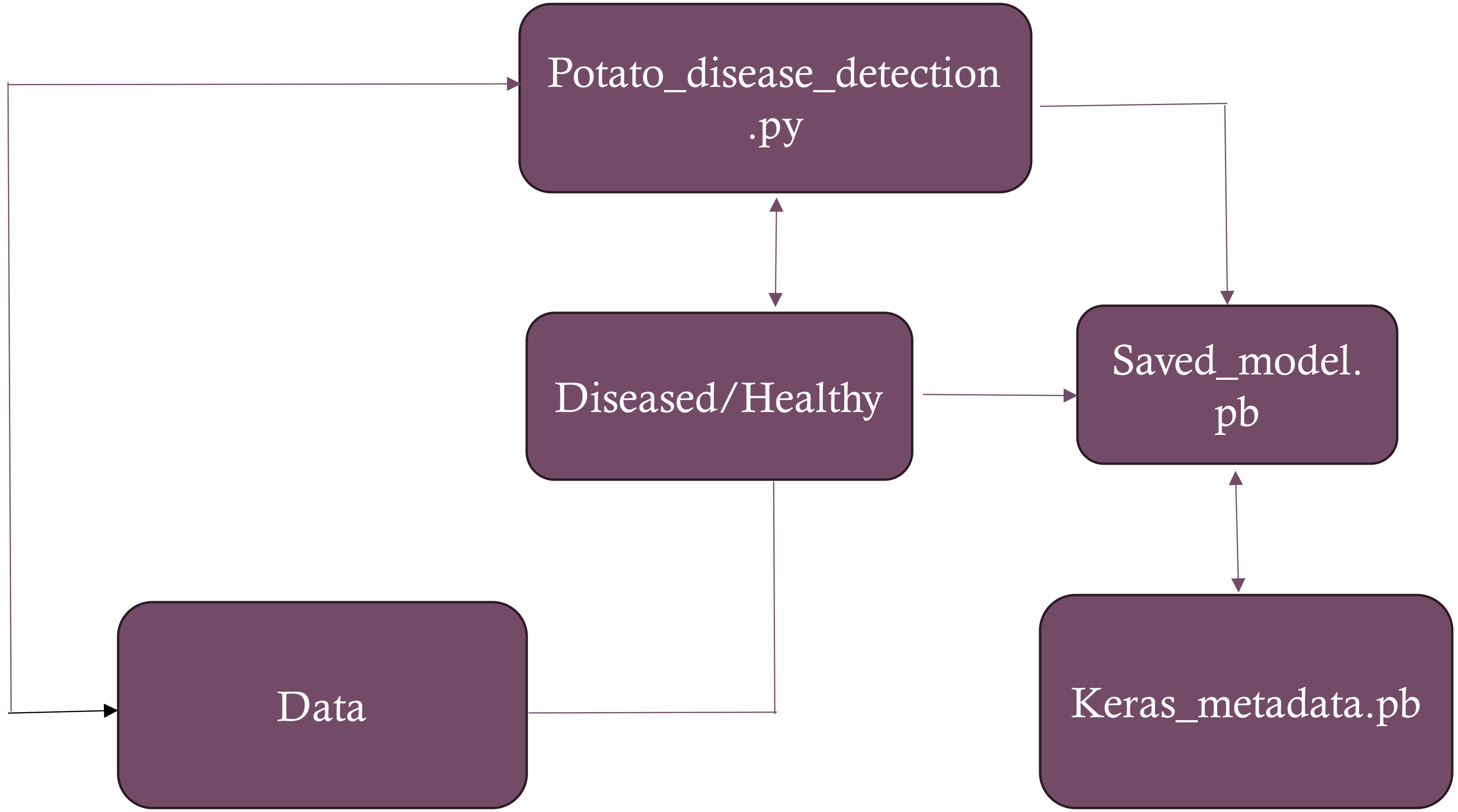

System
Take the input image
Process the input image
Apply the model to the output
Detect disease on clicking submit button
Display output



SEQUENCE DIAGRAM



CLASS DIAGRAM



Potato_disease_detection .py

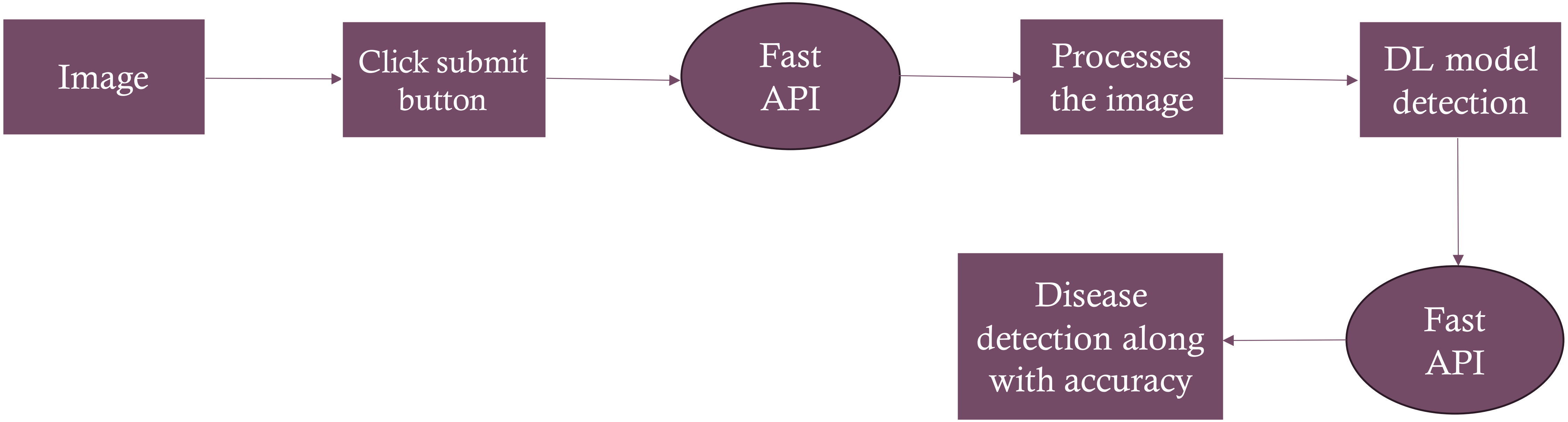
Diseased/Healthy

Saved_model. pb

Data

Keras_metadata.pb

PROCESS FLOW DIAGRAM



Image

Click submit button

Fast API

Processes the image

DL model detection

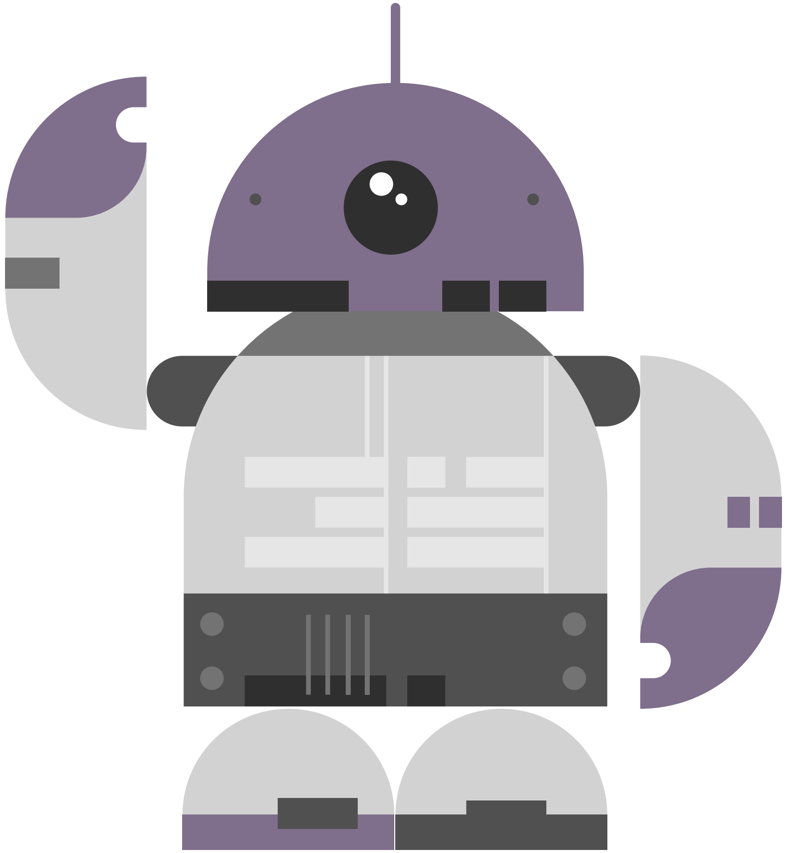
Fast API

Disease detection along with accuracy

TEAM B 08

IT – A

NGIT



THANK YOU