MACHINE LEARNING APPROACH TO DETECT & ANNOTATE DISEASES USING RETINAL IMAGGES

2023-162

Status Document

Lakshith G. P. R.

IT20165666

B.Sc. (Hons) Degree in Information Technology

Specializing in Software Engineering

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

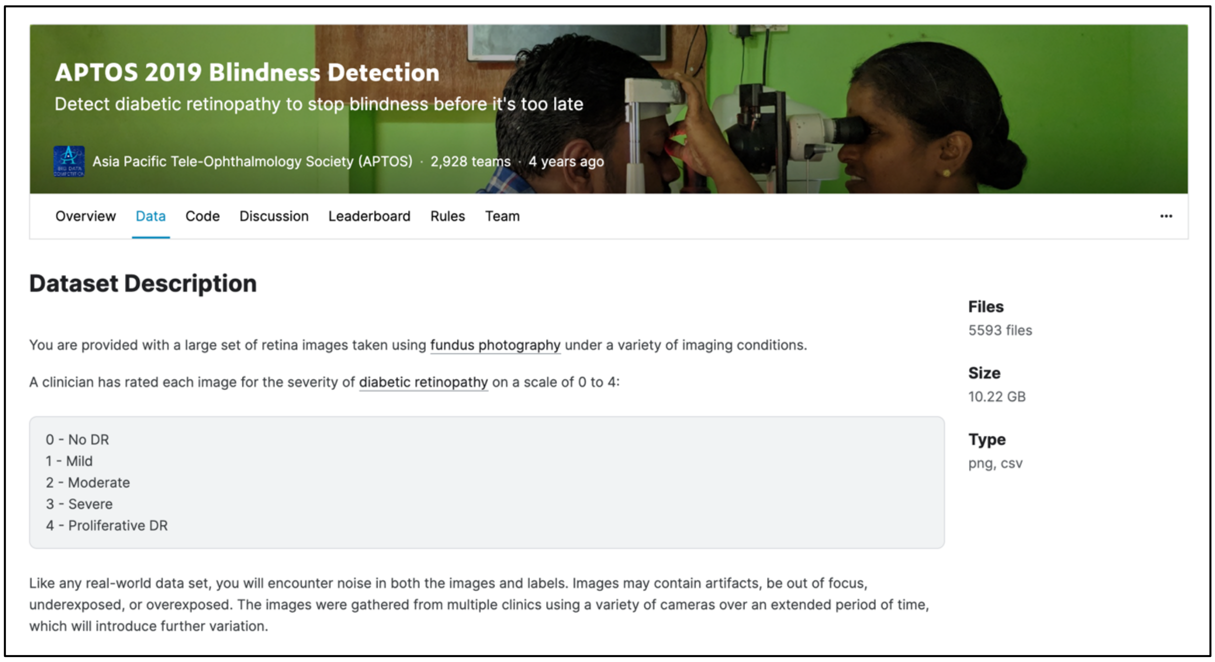
May 2023

Table of Contents

Table of Figures

1. Project progress

1.1. Dataset


APTOS 2019 Blindness Detection

Detect diabetic retinopathy to stop blindness before it's too late

Asia Pacific Tele-Ophthalmology Society (APTOS) · 2,928 teams · 4 years ago

Dataset Description

You are provided with a large set of retina images taken using fundus photography under a variety of imaging conditions.

A clinician has rated each image for the severity of diabetic retinopathy on a scale of 0 to 4:

```
0 - No DR
1 - Mild
2 - Moderate
3 - Severe
4 - Proliferative DR
```

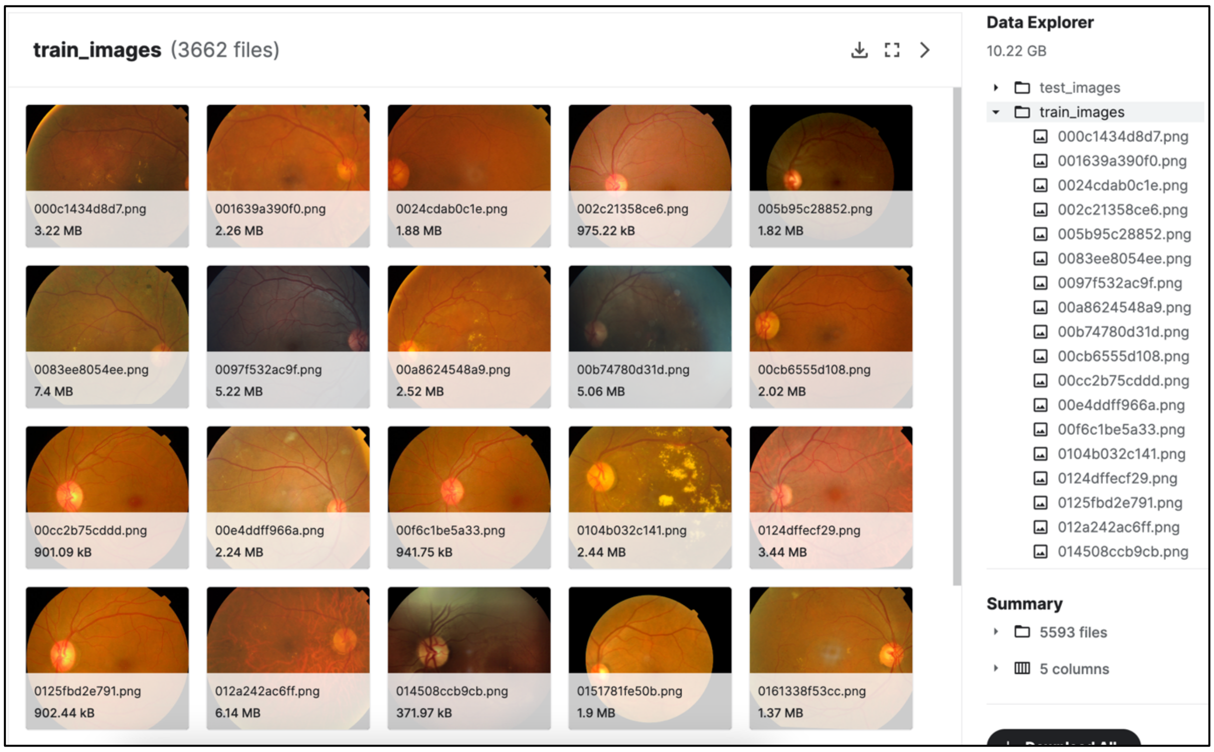
Like any real-world data set, you will encounter noise in both the images and labels. Images may contain artifacts, be out of focus, underexposed, or overexposed. The images were gathered from multiple clinics using a variety of cameras over an extended period of time, which will introduce further variation.

Files
5593 files

Size
10.22 GB

Type
png, csv

Figure 2: Kaggle APTOS dataset


train_images (3662 files)

000c1434d8d7.png 3.22 MB	001639a390f0.png 2.26 MB	0024cdab0c1e.png 1.88 MB	002c21358ce6.png 975.22 kB	005b95c28852.png 1.82 MB
0083ee8054ee.png 7.4 MB	0097f532ac9f.png 5.22 MB	00a8624548a9.png 2.52 MB	00b74780d31d.png 5.06 MB	00cb6555d108.png 2.02 MB
00cc2b75cddd.png 901.09 kB	00e4ddff966a.png 2.24 MB	00f6c1be5a33.png 941.75 kB	0104b032c141.png 2.44 MB	0124dffecf29.png 3.44 MB
0125fbd2e791.png 902.44 kB	012a242ac6ff.png 6.14 MB	014508ccb9cb.png 371.97 kB	0151781fe50b.png 1.9 MB	0161338f53cc.png 1.37 MB

Data Explorer
10.22 GB

- test_images
- train_images
  - 000c1434d8d7.png
  - 001639a390f0.png
  - 0024cdab0c1e.png
  - 002c21358ce6.png
  - 005b95c28852.png
  - 0083ee8054ee.png
  - 0097f532ac9f.png
  - 00a8624548a9.png
  - 00b74780d31d.png
  - 00cb6555d108.png
  - 00cc2b75cddd.png
  - 00e4ddff966a.png
  - 00f6c1be5a33.png
  - 0104b032c141.png
  - 0124dffecf29.png
  - 0125fbd2e791.png
  - 012a242ac6ff.png
  - 014508ccb9cb.png

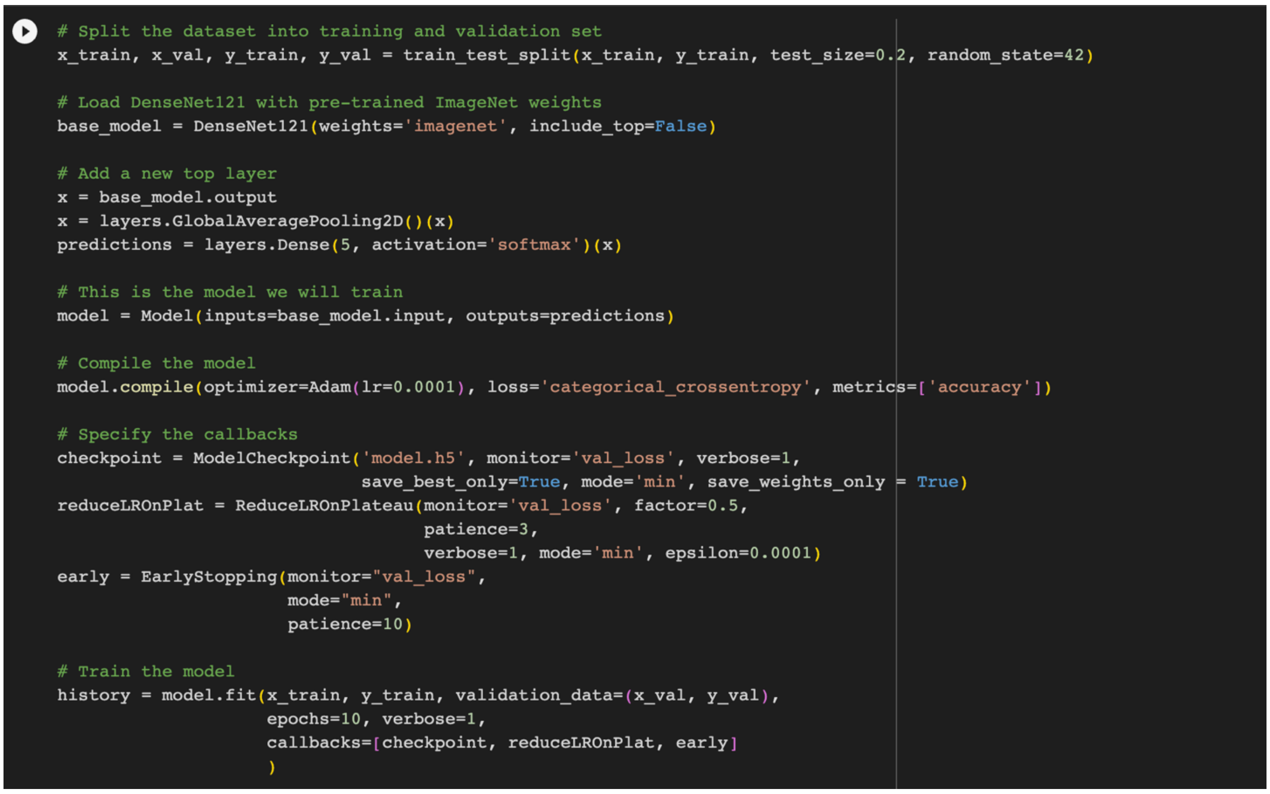
Summary
- 5593 files
- 5 columns

Figure 1: Sample train images

1.2. Diabetic retinopathy severity classification model

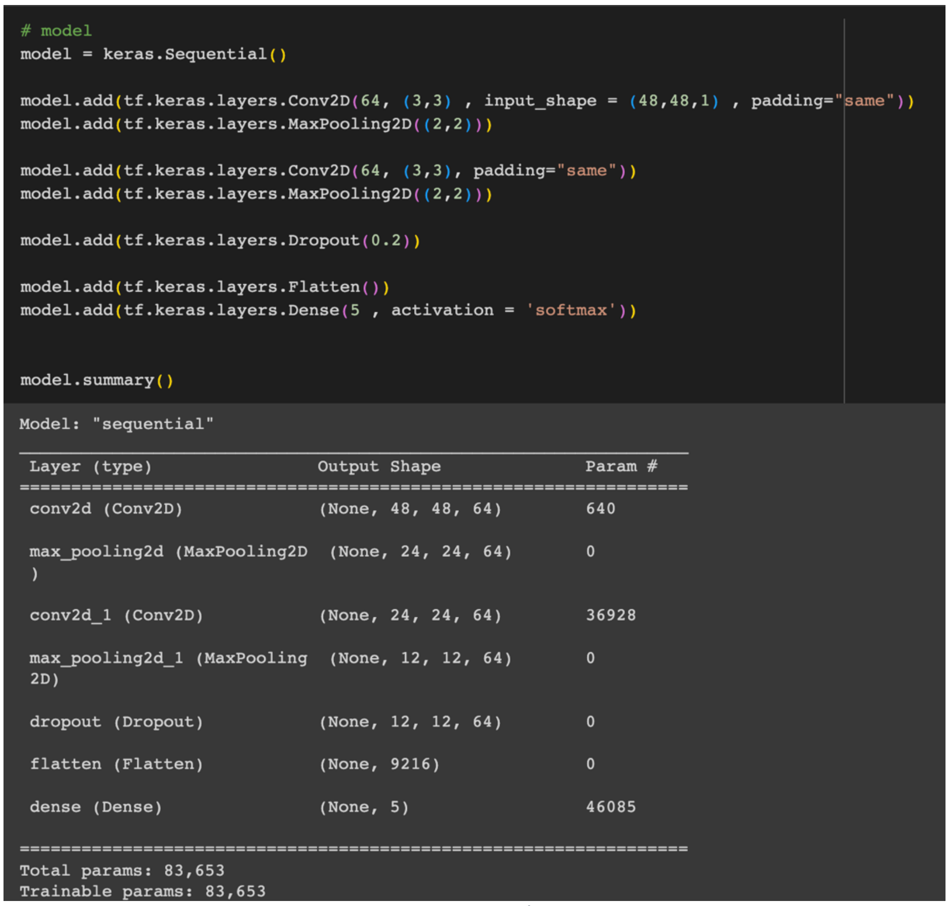
This research centers around the development of a mobile application, with the intent to keep the model lightweight. Initially, I developed a model using DenseNet121. However, it soon became apparent that this model was too expansive, thereby increasing training time and demanding more resources. Moreover, it was discovered over time that the model tended to overfit due to the lack of widely distributed features in the dataset.

As a result, I proposed implementing my own lightweight linear model as a solution. Even prior to that, I noticed that the three-channel images in the dataset were leading to extended training times. Moreover, for the purposes of classification, there's no necessity for three-channel images. Thus, during preprocessing, I converted these images to grayscale before feeding them into the model. This deep learning model is comprised of two Convolutional 2D layers, two Max Pooling 2D layers, a Dropout layer, a Flatten layer, and Dense layers.



```
# Split the dataset into training and validation set
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.2, random_state=42)

# Load DenseNet121 with pre-trained ImageNet weights
base_model = DenseNet121(weights='imagenet', include_top=False)

# Add a new top layer
x = base_model.output
x = layers.GlobalAveragePooling2D()(x)
predictions = layers.Dense(5, activation='softmax')(x)

# This is the model we will train
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

# Specify the callbacks
checkpoint = ModelCheckpoint('model.h5', monitor='val_loss', verbose=1,
                             save_best_only=True, mode='min', save_weights_only = True)
reduceLROnPlat = ReduceLROnPlateau(monitor='val_loss', factor=0.5,
                                   patience=3,
                                   verbose=1, mode='min', epsilon=0.0001)
early = EarlyStopping(monitor="val_loss",
                      mode="min",
                      patience=10)

# Train the model
history = model.fit(x_train, y_train, validation_data=(x_val, y_val),
                    epochs=10, verbose=1,
                    callbacks=[checkpoint, reduceLROnPlat, early]
                    )
```

Figure 3: DenseNet121 Model



```
# model
model = keras.Sequential()

model.add(tf.keras.layers.Conv2D(64, (3,3) , input_shape = (48,48,1) , padding="same"))
model.add(tf.keras.layers.MaxPooling2D((2,2)))

model.add(tf.keras.layers.Conv2D(64, (3,3), padding="same"))
model.add(tf.keras.layers.MaxPooling2D((2,2)))

model.add(tf.keras.layers.Dropout(0.2))

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(5 , activation = 'softmax'))


model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 64)        640       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 24, 24, 64)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 24, 24, 64)        36928     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 12, 12, 64)       0         
 2D)                                                             
                                                                 
 dropout (Dropout)           (None, 12, 12, 64)        0         
                                                                 
 flatten (Flatten)           (None, 9216)              0         
                                                                 
 dense (Dense)               (None, 5)                 46085     
                                                                 
=================================================================
Total params: 83,653
Trainable params: 83,653
```

Figure 4: My own solution

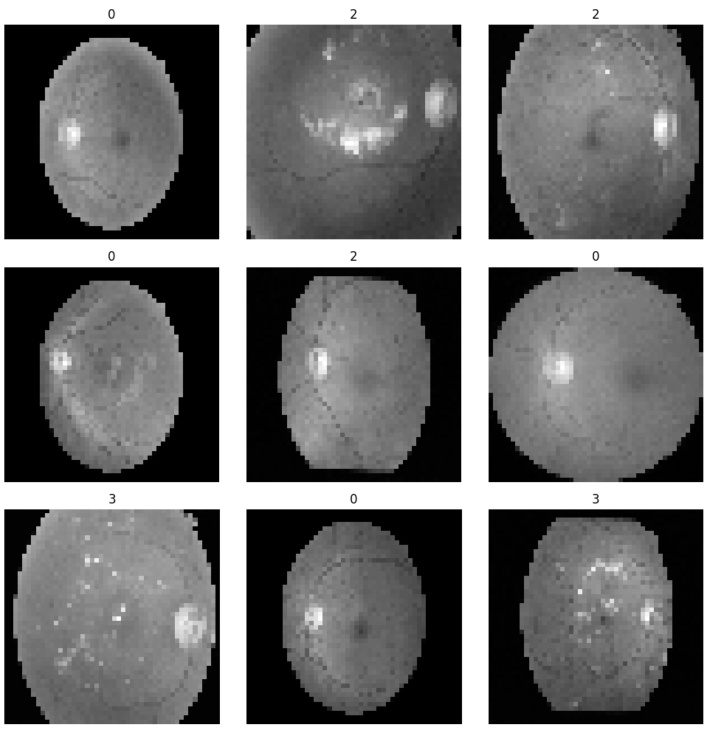
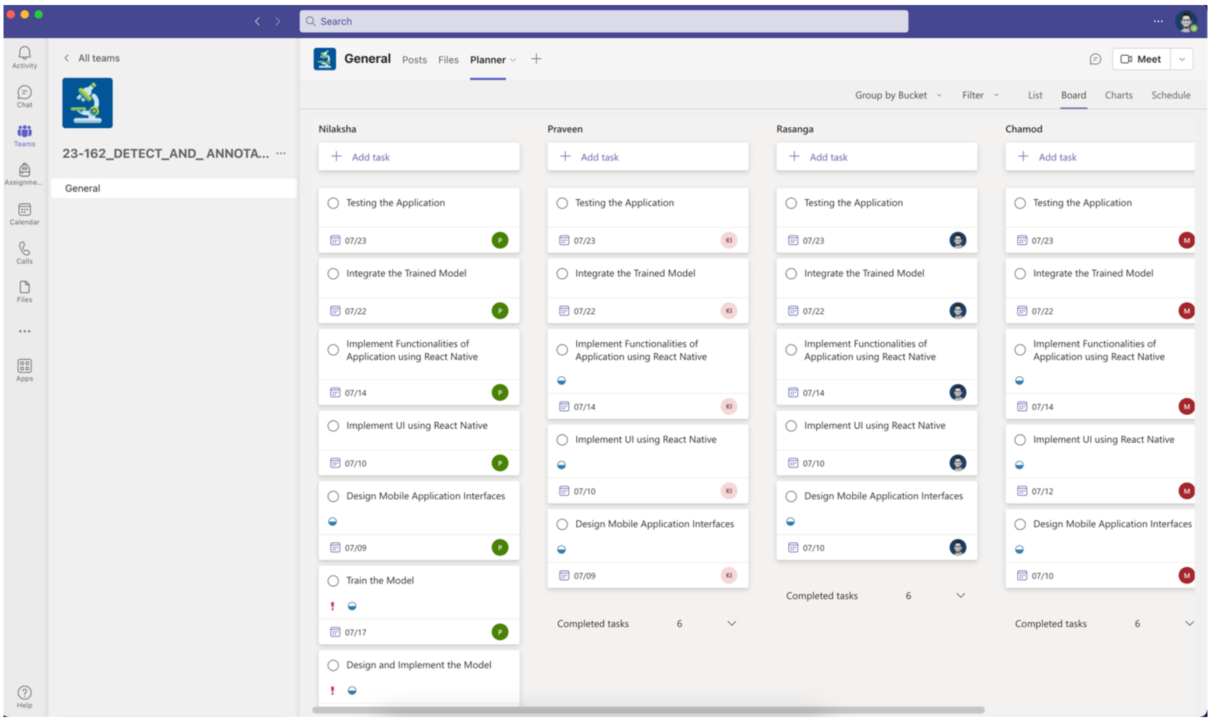


Figure 5: Preprocessed images

2. Project view



Search

All teams

23-162_DETECT_AND_ANNOTA...

General

General Posts Files Planner

Meet

Group by Bucket Filter List Board Charts Schedule

Nilaksha

Add task

Testing the Application
07/23

Integrate the Trained Model
07/22

Implement Functionalities of Application using React Native
07/14

Implement UI using React Native
07/10

Design Mobile Application Interfaces
07/09

Train the Model
07/17

Design and Implement the Model

Praveen

Add task

Testing the Application
07/23

Integrate the Trained Model
07/22

Implement Functionalities of Application using React Native
07/14

Implement UI using React Native
07/10

Design Mobile Application Interfaces
07/09

Completed tasks 6

Rasanga

Add task

Testing the Application
07/23

Integrate the Trained Model
07/22

Implement Functionalities of Application using React Native
07/14

Implement UI using React Native
07/10

Design Mobile Application Interfaces
07/10

Completed tasks 6

Chamod

Add task

Testing the Application
07/23

Integrate the Trained Model
07/22

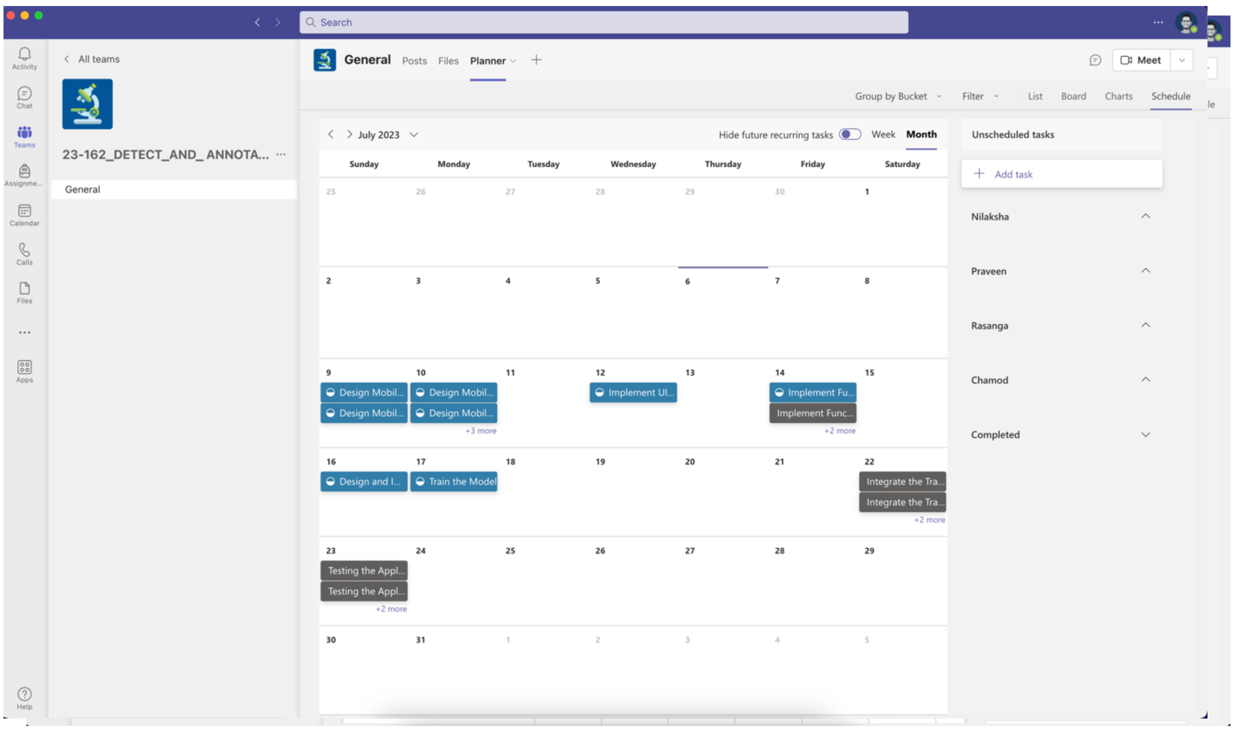
Implement Functionalities of Application using React Native
07/14

Implement UI using React Native
07/12

Design Mobile Application Interfaces
07/10

Completed tasks 6

Figure 6: Planner - board view



Search

All teams

23-162_DETECT_AND_ANNOTA...

General

General Posts Files Planner

Meet

Group by Bucket Filter List Board Charts Schedule

July 2023

Hide future recurring tasks Week Month

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9 Design Mobil... Design Mobil...	10 Design Mobil... Design Mobil... +3 more	11	12 Implement UI...	13	14 Implement Fu... Implement Func... +2 more	15
16 Design and I...	17 Train the Model	18	19	20	21	22 Integrate the Tra... Integrate the Tra... +2 more
23 Testing the Appl... Testing the Appl... +2 more	24	25	26	27	28	29
30	31	1	2	3	4	5

Unscheduled tasks

Add task

Nilaksha

Praveen

Rasanga

Chamod

Completed

Figure 8: Planner - schedule view

3. Gantt chart

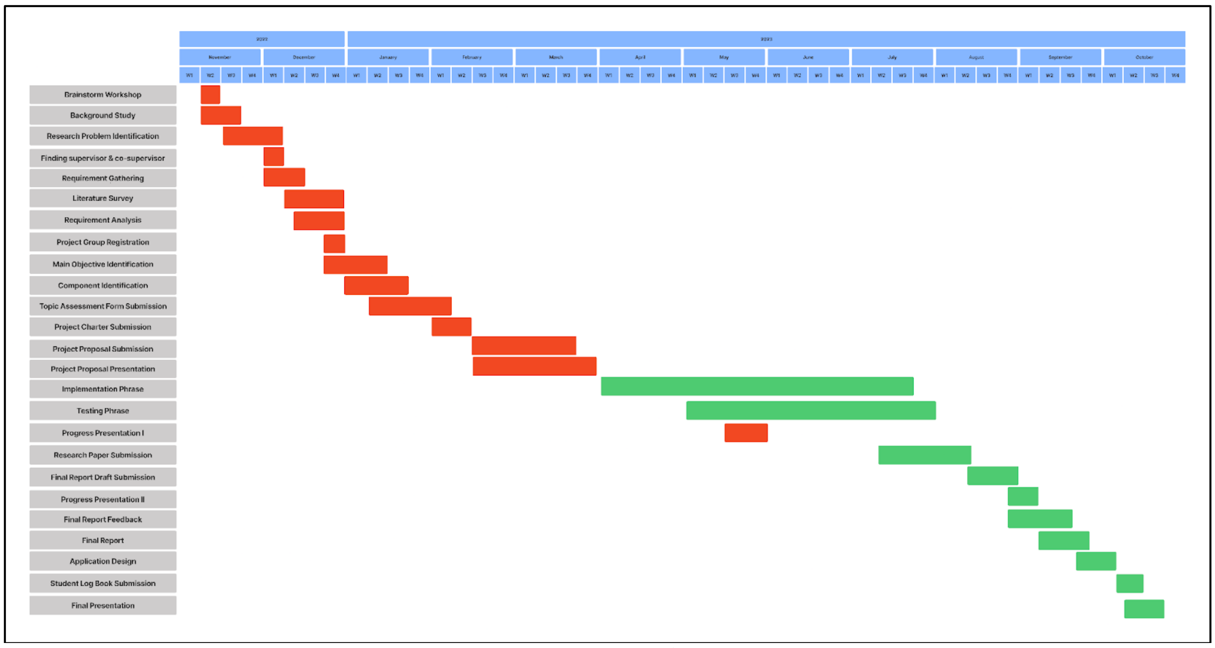
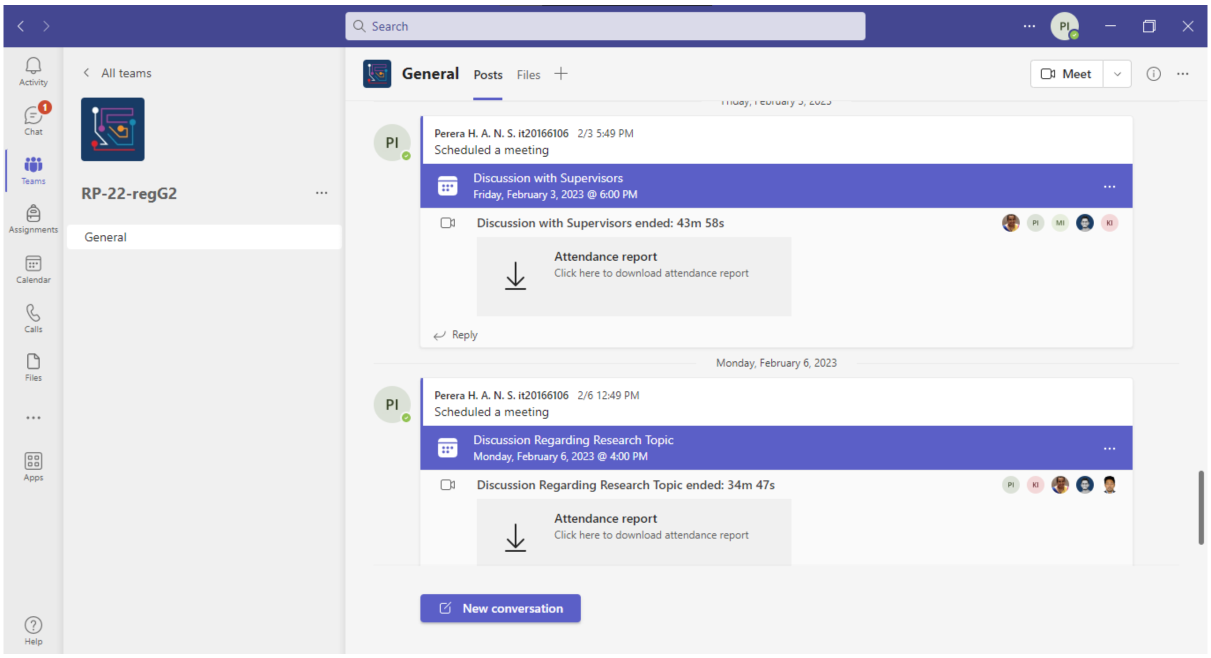


Figure 9: Gantt chart

4. Screenshots of conversations and calls – MS Teams



Friday, February 3, 2023

Perera H. A. N. S. it20166106 2/3 5:49 PM
Scheduled a meeting

Discussion with Supervisors
Friday, February 3, 2023 @ 6:00 PM

Discussion with Supervisors ended: 43m 58s

Attendance report
Click here to download attendance report

Reply

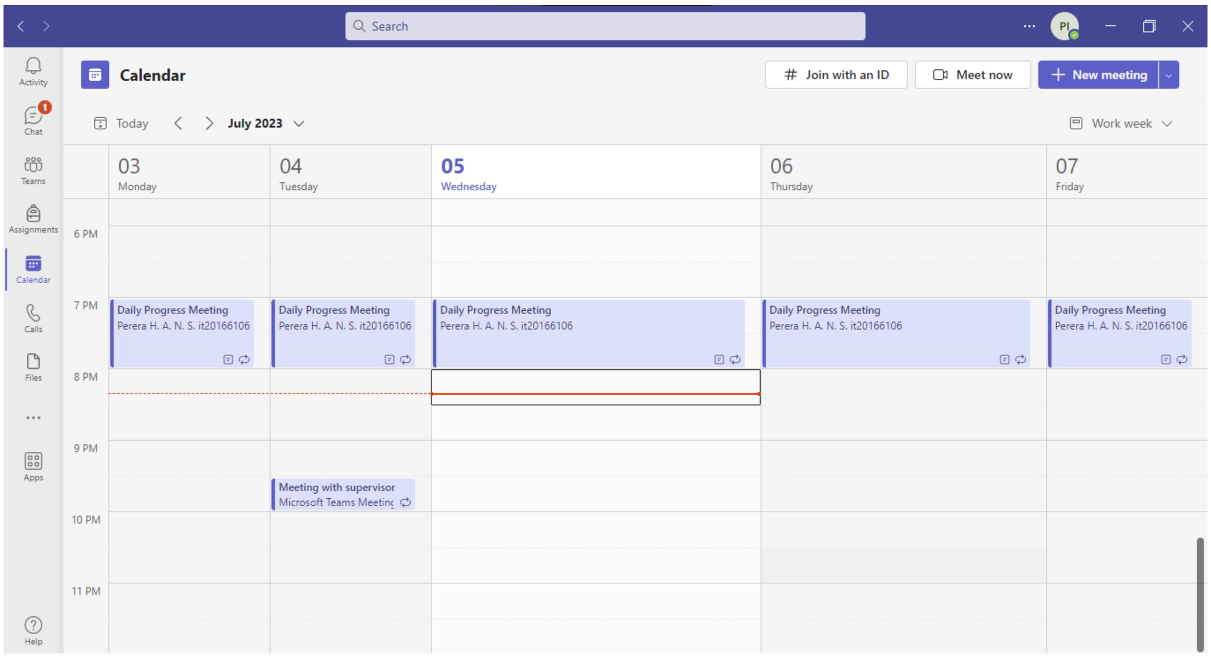
Monday, February 6, 2023

Perera H. A. N. S. it20166106 2/6 12:49 PM
Scheduled a meeting

Discussion Regarding Research Topic
Monday, February 6, 2023 @ 4:00 PM

Discussion Regarding Research Topic ended: 34m 47s

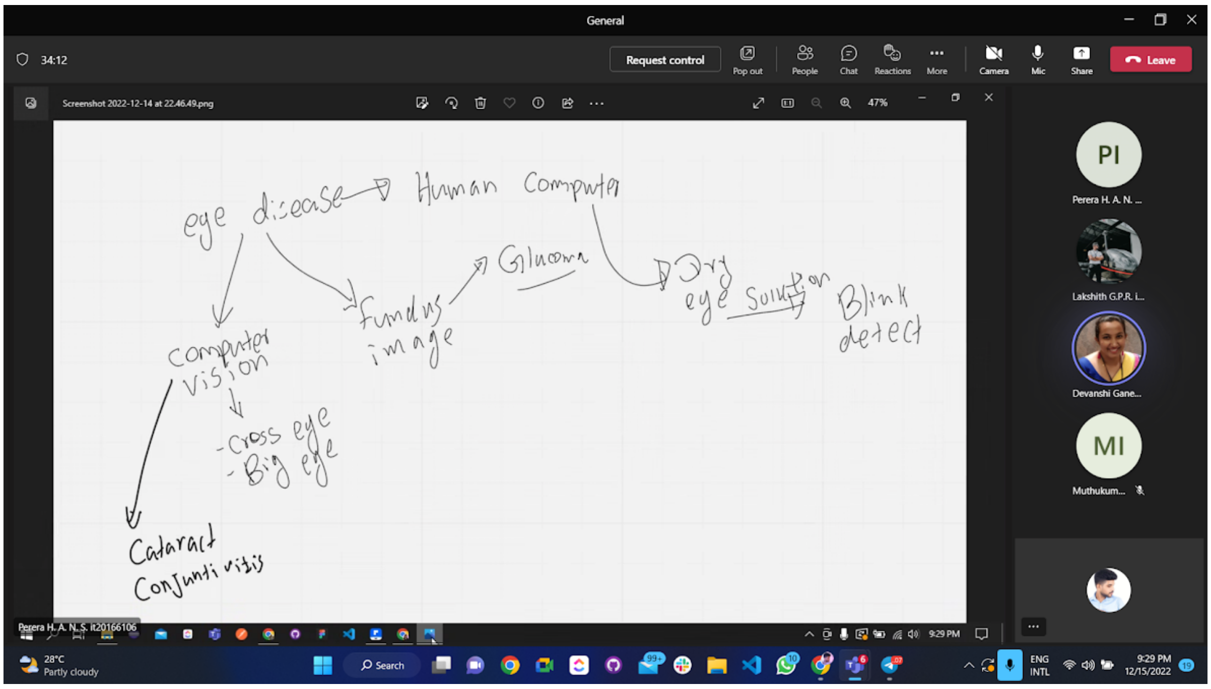
Attendance report
Click here to download attendance report



Calendar

July 2023 — Work week

	03 Monday	04 Tuesday	05 Wednesday	06 Thursday	07 Friday
7 PM	Daily Progress Meeting Perera H. A. N. S. it20166106	Daily Progress Meeting Perera H. A. N. S. it20166106	Daily Progress Meeting Perera H. A. N. S. it20166106	Daily Progress Meeting Perera H. A. N. S. it20166106	Daily Progress Meeting Perera H. A. N. S. it20166106
9 PM		Meeting with supervisor Microsoft Teams Meeting			


General
34:12
Request control
Screenshot 2022-12-14 at 22.46.49.png
eye disease
Human Computer
Glucoma
Dry eye Solution
Blink detect
Fundus image
computer vision
- Cross eye
- Big eye
Cataract
Conjuntivitis
Perera H. A. N. ...
Lakshith G.P.R. L...
Devanshi Gane...
Muthukum...
Perera H. A. N. S. it20166106
Leave