Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

3.1.1 Developmental surveillance

Developmental surveillance is a flexible process whereby knowledgeable clinicians gather relevant information over time from multiple sources (including parents and by direct observation) toward the goal of identifying and addressing developmental concerns, including those related to ASD. Developmental surveillance involves integrating information obtained from inquiry around parental concerns, clinical observations and, possibly, also incorporating standardized measures (e.g., parent questionnaires) to inform clinical impressions and decision making.

3.1.2 Developmental screening

Developmental screening involves a brief assessment using a standardized measure to identify children at increased risk for delay or disorder. Screens vary by format (e.g., parent report versus direct assessment of the child) and scope ('broadband' screens cover multiple developmental domains versus those specific to a particular domain or disorder). Clinicians may use clinical judgement when considering referral for further assessment, even when a child has a 'screen-negative' score. Considerations include clinical observations, parental concerns, and other suggestive factors, such as a positive family history.

3.1.3 Developmental diagnosis

A brief test using a screening tool does not provide a diagnosis, but it can indicate whether a child is on the right development track or if a specialist should take a closer look. If the screening tool identifies an area of concern, a formal developmental evaluation may be needed. This formal evaluation is a more in-depth look at a child's development and is usually done by a trained specialist such as a developmental pe-

diatrician, child psychologist, speech-language pathologist, occupational therapist, or other specialist. The specialist may observe the child give the child a structured test, ask the parents or caregivers questions, or ask them to fill out questionnaires. The results of this formal evaluation highlight your child's strengths and challenges and can inform whether they meet criteria for a developmental diagnosis.

3.2 Proposed System

The idea of our proposed system is to detect markers of Autism Spectrum Disorder (ASD) in young children without the use of heavy machinery and medical expertise. The only machinery that are used in this project are eye tracker and a computer system. Because of the ease of usage, our system can be put to use in any environment. As our system does not include medical machinery, children will feel safe and are thus cooperative and less reluctant with the examination . The spontaneity and speed in which our system produces results make it easier and less time consuming to decide if a child possesses the symptoms of autism. If that is the case , then the child can be put under further examination or tests to confirm the disorder. If a child does not show markers of autism then no further tests are required which saves a lot of time, energy and money.

3.3 Feasibility Study

The feasibility study helps with the project's analysis. This project's feasibility study is divided into three categories.

3.3.1 Economic Feasibility

As our project includes less machinery and decreases the requirement of high expertise, it is safe to say that our project is economically feasible. Since most of the equipment is already used in the beginning to record the raw data of the patient it prevents any additional equipment even if the project is developed further in the future.

3.3.2 Technical Feasibility

Our project requires simple hardware equipment i.e, the eye tracker and a computer system, the rest of the equipment is software based. The software equipment involves usage of classification techniques of deep learning such as 1NN and CNN , which produce accurate results. All the programming is done in python hence it is very simple and easy to understand even if any changes or developments are required in the future it is easy to apply them in python.

3.3.3 Social Feasibility

Since our project does not involve heavy and complicated medical equipment it is easy to collect data from a child at the comfort of their own home, school etc,. Children tend to be more comfortable around people that they know or in an environment familier to them rather than a hospital. This makes it easy to collect their eye tracking data, thus making it socially feasible.

3.4 System Specification

3.4.1 Hardware Specification

• Processor: Minimum 1 GHz; Recommended 2GHz or more.
• Ethernet connection: (LAN) OR a wireless adapter (Wi-Fi).
• Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
• Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.
• FaceLAB head-free binocular eye-tracking system (60 Hz)

3.4.2 Software Specification

• IDE : Anaconda
• Programming languages:HTML,Python 3.9
• Keras
• Tensorflow
• Numpy
• seaborn
• matplotliob

• pandas
• jupyter

3.5 Data Set

Data is present in multiple excel files. Each file is for each patient and is present with the name format: Class-Patient-ID-Age-Gender-Initials Each file consists of various number of sheets for different images the patient is looking at. Some of the images are: Buffer slide, Black slide, White slide and image slides. An example of few of the images is given in the next slide. Each patient file contains the following data: x, y, Left Diameter, Right Diameter, Time, Look Zone (eyes, nose, head, screen, mouth, object, etc). 'x' and 'y' are the pixels of the image (with dimensions AKA resolution 1024*768). The data is 'x' and 'y' columns represents on which pixel the patient gazed his look on. Left diameter and right diameter represent the pupillometric measurements of the patient. They represent the diameters of the left eye pupil and right eye pupil respectively for that particular pixel that is being stared upon. Time – this column represents the amount of time the patient stared at a particular pixel LookZone – this represents the area of an image where the patient looked at. The images consist of either faces of humans (blurred and normal) or objects (blurred and normal). Therefore, the LookZones either represent the facial features of the image (human face) or of the objects.

Figure 3.1: **An Example of image shown to the patient, Stim - visob04.jpg**

Chapter 4

METHODOLOGY

4.1 General Architecture

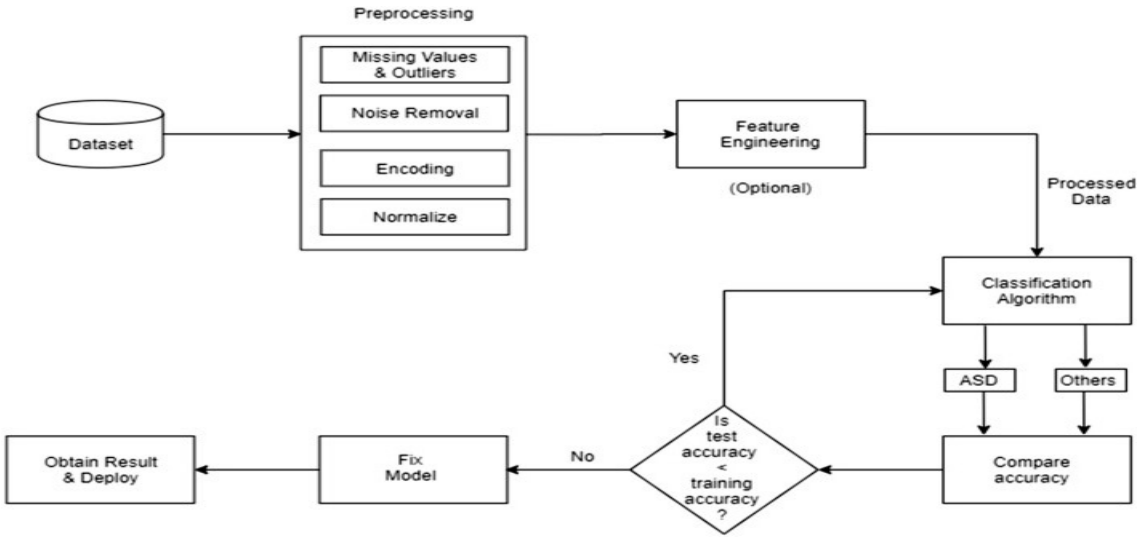


Figure 4.1: **General Block Diagram of Machine Learning**

The architecture diagram provides the outline of the project.The raw data or dataset of the patients collected has to to be preprocessed by including the missing values like patient id ,age, gender,lz zone,class ,etc. and merging all the files into one large all data file for further processes. All data file file is further undergo feature engineering and then the processed data is classified.

4.2 Design Phase

4.2.1 Data Flow Diagram

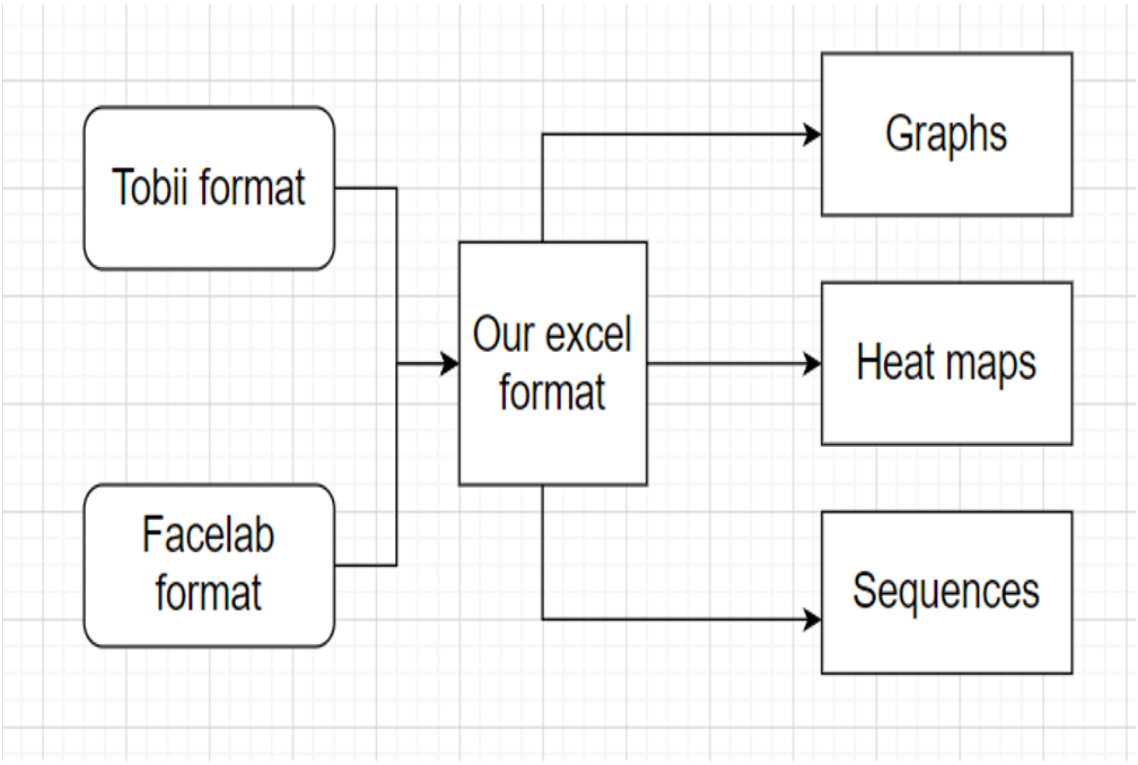


Figure 4.2: **Data Flow Block Diagram**

Data flow diagram is a pictorial or graphical representation. It can be applied to represent the input data to a system and multiple functions carried out on the data and the output is generated by the system. The collected data is pre-processed and the duplicate values are deleted. The raw data collected by using facelab eye tracker is preprocessed and converted into excel format which can be accessed easily. Using that excel data file we have generate matrices and heatmaps for further classification.

4.2.2 Activity Diagram

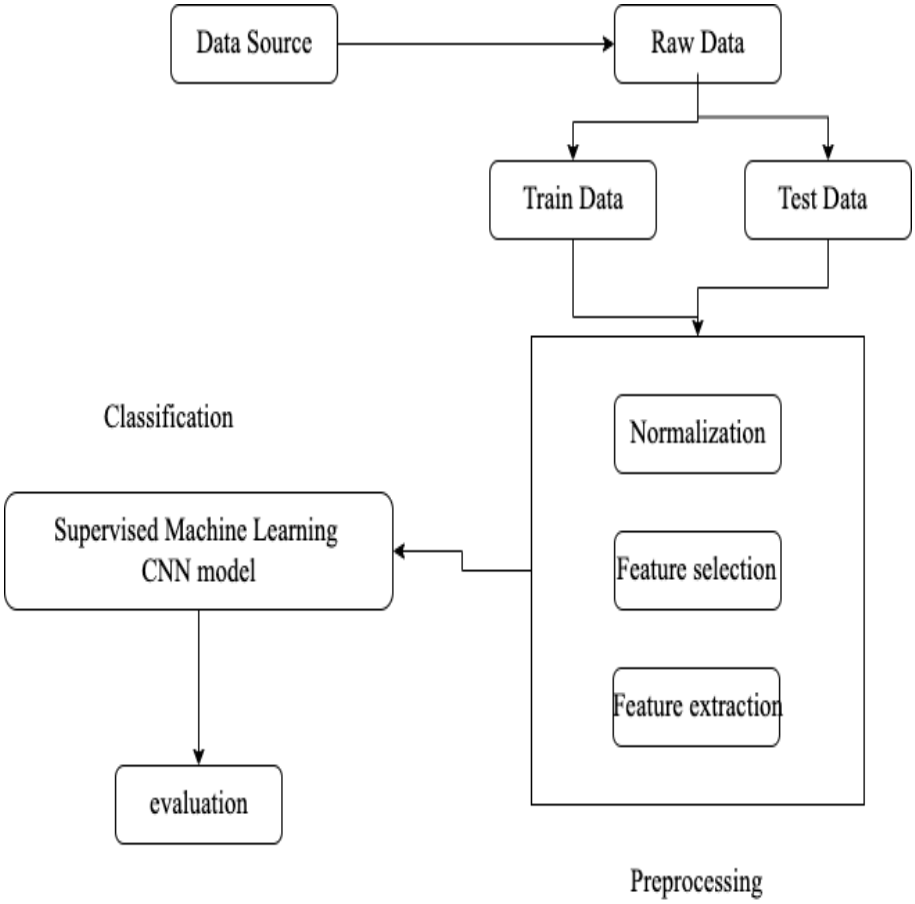


Figure 4.3: **Activity Diagram**

Activity diagram is made to understand the flow of activities in the project. In the above diagram there are 3 major activities they are data collection, feature extraction and prediction . After collecting the data the raw data is preprocessed and the duplicate values are removed from the data set .if there are no duplicate values then the data is sent to feature extraction where the features of data are extracted and the data with similar features are assigned to near. then the data is sent for ML training and the Autism markers are predicted . Figure 4.6 shows about the Activity diagram.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

Algorithm For Adding Sheet Name Column :

Step 1 : Import requuired libraries
Step 2 : Read the folder of data set
Step 3 : Read each file in the folder
Step 4 : Read file as pandas dataframe
Step 5 : Get the sheetnames in each file
Step 6 : Adding sheet name column to the file
Step 7 : Save the files with sheetname column in excel format

In [1]:
```
#import requuired libraries

import pandas as pd
import os
```

In [ ]:
```
# read the folder of data set

folder = "/Users/dhanu/Desktop/Patients/"
```

In [ ]:
```
#read each file in the folder

for file in listdir.os(folder):
    df = pd.read_excel(file)  #read file as pandas dataframe
    sheets = df.sheet_names   #get the sheetnames in each file
    #adding sheet name column to the file
    df1 = pd.concat([pd.read_excel(df, sheet_name=s).assign(sheet_name=s) for s in sheets])
    # save the files with sheetname column in excel format
    df1.to_excel("/Users/dhanu/Desktop/c_out/"+ file + ".xlsx")
```

Figure 4.4: **code for adding sheet name column in excel file**

Algorithm For Adding File Name Column :

Step 1 : Import requuired libraries
Step 2 : Read the folder of files with sheetnames added

Step 3 : Create an empty dataframe
Step 4 : Read each file in folder and add filename column
Step 5 : Append all the files with filename column to the empty dataframe
Step 6 : Save the dataframe as excel file

```
In [4]:
#import required libraries

import pandas as pd
import os
import numpy as np
```

```
In [24]:
# read the folder
path = r'/Users/dhanu/Desktop/c2/'
output_loc = "/Users/dhanu/Desktop/cp all/"
files = os.listdir(path)
```

```
In [27]:
#create an empty dataframe
df_total = pd.DataFrame()

for file in files:
    #read each file in folder and add filename column
    df_temp = pd.read_excel(path + "" + file)
    df_temp['filename'] = file
    #append all the files with filename column to the empty dataframe
    df_total = df_total.append(df_temp)
#save the dataframe as excel file
df_total.to_excel("/Users/dhanu/Desktop/cp all/c2final.xlsx")
```

```
In [ ]:
```

Figure 4.5: **code for adding filename column in excel file**

Algorithm for removing unwanted data and nodata updation:

Step 1 : Import required libraries
Step 2 : Read the sheetname column and filename column added file
Step 3 : Remove the unwanted data by column filtering
Step 4 : Dealing with the nodata cells
Step 5 : Save the dataframe as excel file

In [57]:

```
#import required libraries

import pandas as pd
import numpy as np
import os
import openpyxl
```

In [58]:

```
#read the excel file with both sheetname column and filename column
DF = pd.read_excel('/Users/dhanu/Desktop/final/finalp.xlsx')
```

In [59]:

```
#remove unwanted data
f1 = DF[(DF['sheet_name']!= 'Black Slide - F')
& (DF['sheet_name']!= 'Black Slide - STAT')
& (DF['sheet_name']!= 'Black Slide - F 2')
& (DF['sheet_name']!= 'White Slide - F')
& (DF['sheet_name']!= 'White Slide - STAT')
& (DF['sheet_name']!= 'Buffer Slide - F')
& (DF['sheet_name']!= 'Buffer Slide - F 2')
& (DF['sheet_name']!= 'Buffer Slide - F 3')
& (DF['sheet_name']!= 'Buffer Slide - F 4')
& (DF['sheet_name']!= 'Buffer Slide - F 5')
& (DF['sheet_name']!= 'Buffer Slide - F 6')
& (DF['sheet_name']!= 'Buffer Slide - F 7')
& (DF['sheet_name']!= 'Buffer Slide - F 8')
& (DF['sheet_name']!= 'Buffer Slide - F 9')
& (DF['sheet_name']!= 'Buffer Slide - F 10')
& (DF['sheet_name']!= 'Buffer Slide - F 11')
& (DF['sheet_name']!= 'Buffer Slide - F 12')
& (DF['sheet_name']!= 'Buffer Slide - F 13')
& (DF['sheet_name']!= 'Buffer Slide - F 14')
& (DF['sheet_name']!= 'Buffer Slide - F 15')
& (DF['sheet_name']!= 'Buffer Slide - F 16')
& (DF['sheet_name']!= 'Buffer Slide - F 17')
& (DF['sheet_name']!= 'Buffer Slide - F 18')
& (DF['sheet_name']!= 'Buffer Slide - F 19')
& (DF['sheet_name']!= 'Buffer Slide - F 20')
& (DF['sheet_name']!= 'Buffer Slide - F 21')
& (DF['sheet_name']!= 'Buffer Slide - F 22')
& (DF['sheet_name']!= 'Buffer Slide - F 23')
& (DF['sheet_name']!= 'Buffer Slide - F 24')
```

```
& (DF['sheet_name']!= 'mosob03.jpg - F')
& (DF['sheet_name']!= 'mosob03.jpg - STAT')
& (DF['sheet_name']!= 'mosob04.jpg - F')
& (DF['sheet_name']!= 'mosob04.jpg - STAT')
& (DF['sheet_name']!= 'ob01.jpg - F')
& (DF['sheet_name']!= 'ob01.jpg - STAT')
& (DF['sheet_name']!= 'ob02.jpg - F')
& (DF['sheet_name']!= 'ob02.jpg - STAT')
& (DF['sheet_name']!= 'ob03.jpg - F')
& (DF['sheet_name']!= 'ob03.jpg - STAT')
& (DF['sheet_name']!= 'ob04.jpg - F')
& (DF['sheet_name']!= 'ob04.jpg - STAT')
& (DF['sheet_name']!= 'vis01.jpg - F')
& (DF['sheet_name']!= 'vis01.jpg - STAT')
& (DF['sheet_name']!= 'vis02.jpg - F')
& (DF['sheet_name']!= 'vis02.jpg - STAT')
& (DF['sheet_name']!= 'vis03.jpg - F')
& (DF['sheet_name']!= 'vis03.jpg - STAT')
& (DF['sheet_name']!= 'vis04.jpg - F')
& (DF['sheet_name']!= 'vis04.jpg - STAT')
& (DF['sheet_name']!= 'vismos01.jpg - F')
& (DF['sheet_name']!= 'vismos01.jpg - STAT')
& (DF['sheet_name']!= 'vismos02.jpg - F')
& (DF['sheet_name']!= 'vismos02.jpg - STAT')
& (DF['sheet_name']!= 'vismos03.jpg - F')
& (DF['sheet_name']!= 'vismos03.jpg - STAT')
& (DF['sheet_name']!= 'vismos04.jpg - F')
& (DF['sheet_name']!= 'vismos04.jpg - STAT')
& (DF['sheet_name']!= 'visob01.jpg - F')
& (DF['sheet_name']!= 'visob01.jpg - STAT')
& (DF['sheet_name']!= 'visob02.jpg - F')
& (DF['sheet_name']!= 'visob02.jpg - STAT')
& (DF['sheet_name']!= 'visob03.jpg - F')
& (DF['sheet_name']!= 'visob03.jpg - STAT')
& (DF['sheet_name']!= 'visob04.jpg - F')
& (DF['sheet_name']!= 'visob04.jpg - STAT')
]
#save it as the excel file
f1.to_excel("/Users/dhanu/Desktop/final1/p.xlsx",index = False)
```

In [ ]:

Figure 4.6: **code for removing unwanted data**

```
In [30]:
import pandas as pd
import os
import numpy as np

In [31]:
df = pd.read_excel("/Users/dhanu/Desktop/final1/f1.xlsx")

In [33]:
new_df = df['X'].fillna('nodata',inplace = True)

In [35]:
new_df = df['Y'].fillna('nodata',inplace = True)

In [37]:
new_df = df['Left Diam'].fillna('nodata',inplace = True)

In [38]:
new_df = df['Right Diam'].fillna('nodata',inplace = True)

In [40]:
new_df = df['TIme'].fillna('nodata',inplace = True)

In [ ]:
df.to_excel("/Users/dhanu/Desktop/final1/f2.xlsx",index = False)
```

Figure 4.7: **code for updating nodata cells**

Algorithm for Splitting Filename Column:

Step 1 : import required libraries.
Step 2 : read the excelfile as dataframe.
Step 3 : remove the extension in filename column.
Step 4 : split the filename column into sepretate columns.
Step 5 : give names to the split columns.
Step 6 : save the dataframe as an excel file.

```
In [2]:
#import the required libraries
import pandas as pd
import os

In [3]:
#read the excel file as dataframe
df = pd.read_excel("/Users/dhanu/Desktop/final1/g3.xlsx")

In [4]:
#remove the extension in filename column
df['File Name'] =df['File Name'].str.replace('.xlsx','')

/var/folders/02/k754q_hn7kvdlvjf1_vyr79m0000gn/T/ipykernel_8695/591001716.py:2: FutureWarning: The default value of r
sion.
  df['File Name'] =df['File Name'].str.replace('.xlsx','')

In [11]:
#split the filename column into sepretate columns
split_d = df['File Name'].str.split('_',n = -1,expand = True)

In [41]:
#give names to the split columns
df[['Class','Patient ID','Age(in months)','Gender','Initials']]= df['File Name'].str.split('_',expand = True)

In [43]:
#save the dataframe as excel file
df.to_excel("/Users/dhanu/Desktop/final1/f1.xlsx",index = False)
```

Figure 4.8: **code for splitting filename column**

Algorithm for Splitting LZ Name Column:

Step 1 : import required libraries.
Step 2 : read the excelfile as dataframe.
Step 3 : split the LZ name column by reading the column from right to left.
Step 4 : give names to the split columns.
Step 5 : save the updated dataframe as an excel file.

```
In [90]:
#import the required libraries
import pandas as pd
import os
import openpyxl
```

```
In [92]:
#read the excel file as dataframe
df = pd.read_excel("/Users/dhanu/Desktop/final1/dhanu.xlsx")
```

```
In [103…
#split the LZ name column by reading the column from right to left
df["Lz3"] = df["LZ Name"].str.split().str[-1]
df["Lz2"] = df["LZ Name"].str.split().str[-2]
df["Lz1"] = df["LZ Name"].str.split().str[-3]
```

```
In [102…
#save the updated dataframe as excel file
df.to_excel("/Users/dhanu/Desktop/final1/dhanu2.xlsx")
```

Figure 4.9: **code for splitting LZ Name column**

Algorithm for Splitting excelfile into multiple excelfiles based on Column values:

Step 1 : Import required libraries.
Step 2 : Read the excelfile as dataframe.
Step 3 : Split the file into multiple files based on column value using unique() function.
Step 4 : Repeat the same code for multiiple columns until we get the desired files.
Step 5 : Save the dataframes as excel files with names as unique column values.

In [1]:
```
#import required libraries
import pandas as pd
import numpy as np
import openpyxl
import os
```

In [14]:
```
#read the final all data excel file as dataframe
df = pd.read_excel("/Users/dhanu/Desktop/split/p/P.xlsx")
```

In [16]:
```
#give output location
output = "/Users/dhanu/Desktop/split/p"
```

In [18]:
```
# split the all data file into different excel files based on column values
pid = df['Patient ID'].unique()
for ids in pid:
    df1 = df[df['Patient ID'] == ids]
    path = os.path.join(output,str(ids)+".xlsx")
    #read the dataframes into different excel files with names as unique column values
    df1.to_excel(path,index = False)
```

Figure 4.10: **code for splitting file into multiple files based on column values**

Algorithm For Generating HeatMmaps :

Step 1 : Import required libraries.
Step 2 : Read the folder of matrices.
Step 3 : Read each file from the folder using for loop.
Step 4 : Read that file as dataframe matrix.
Step 5 : Create a zero numpy matrix.
Step 6 : If the cell value in matrix is not zero then increase the cell value by 5.
Step 7 : Convert that updated numpy matrix to datframe matrix.
Step 8 : If the cell value is not equal to zero then covert it into numpy matrix and add ones 7*7 matrix.
Step 9 : Convert that numpy matrix to dataframe matrix by retreiving previous value in non zero cell.
Step 10 : Convert the dataframe matrix to seaborn matrix.
Step 11 : Save the heatmap as jpg file.

In [2]:
```
#import required libraries
import pandas as pd
import numpy as np
import os
from scipy.ndimage import gaussian_filter
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.ndimage.morphology import binary_dilation
from scipy.signal import convolve2d
```

In [3]:
```
#read the folder of matrices
folder = "/Users/dhanu/heatmaps from matrix"
plt.rcParams.update({'figure.max_open_warning': 0})
```

In [4]:
```
#read eachfile from the folder
for file in os.listdir(folder):
    if file.startswith("mos01"):
        #read each file as dataframe matrix
        df = pd.read_excel(file)
        df = df.to_numpy()
        #create a zero numpy matrix
        df_inc = np.zeros((1023, 768),dtype=int)
        for i in range(1023):
            for j in range(768):
                if df[i][j] != 0:
                    #if the cell value in matrix is not zero then increase the cell value by 5
                    df_inc = np.where(df,df+5,df)
        #convert that updated numpy matrix to datframe matrix
        df1 = pd.DataFrame(df_inc)
        #if the cell value is not equal to zero then covert it into numpy matrix and add ones 7*7 matrix
        a = df1.ne(0).to_numpy()
        b = np.ones((7,7))
        #convert that numpy matrix to dataframe matrix by retreiving previous value in non zero cell.
        df1 = pd.DataFrame(np.where(binary_dilation(a,b)^a,3, df1))
        plt.figure(figsize=(14.236, 10.67))
        #convert the dataframe matrix to seaborn matrix
        ax = sns.heatmap(df1,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        #save the heatmap as jpg file
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mos01", file + '.jpg')
        plt.savefig(output)
```

Figure 4.11: **code for generating heatmaps**

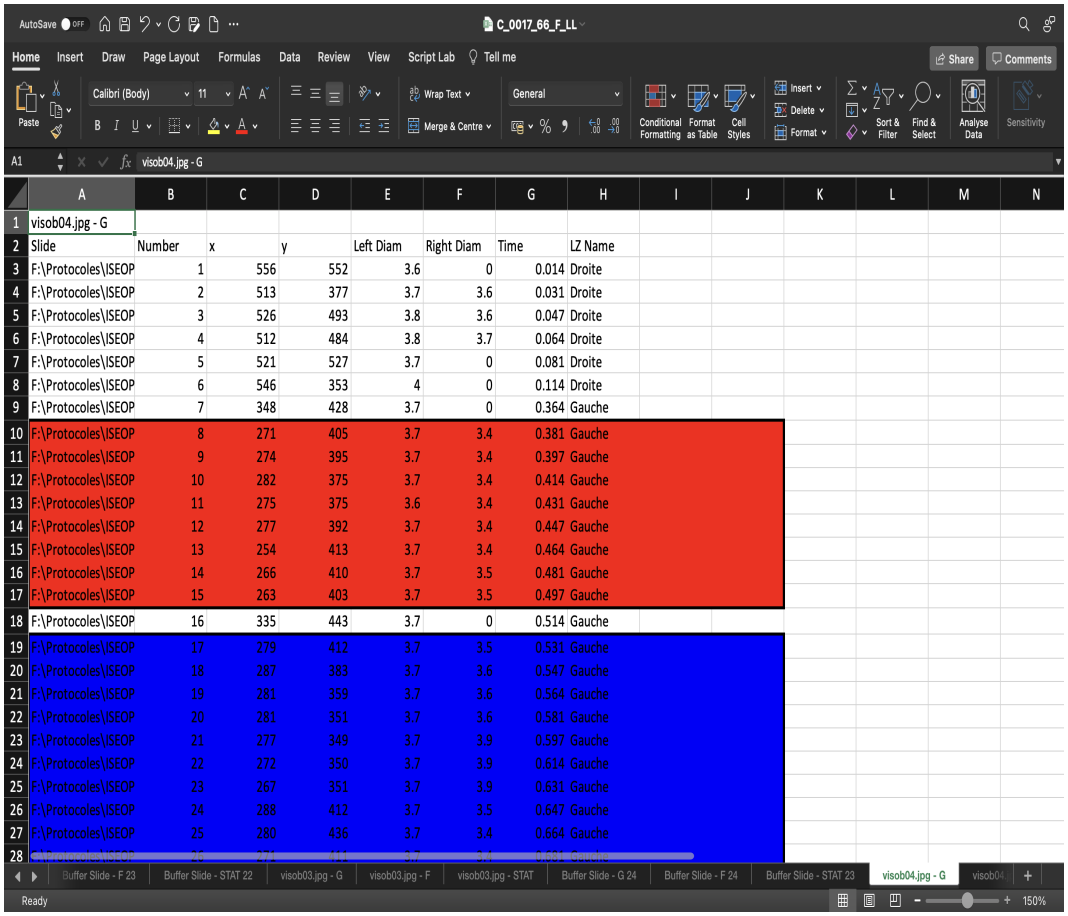
4.3.2 Pseudo Code

4.4 Module Description

4.4.1 Module1: Data Preprocessing

Data is present in multiple excel files. Each file is for each patient and is present with the name format: Class-Patient-ID-Age-Gender-Initials Each file consists of various number of sheets for different images the patient is looking at. Some of the images are: Buffer slide, Black slide, White slide and image slides. An example of few of the images is given in the next slide. Each patient file contains the following data: x, y, Left Diameter, Right Diameter, Time, Look Zone (eyes, nose, head, screen, mouth, object, etc). 'x' and 'y' are the pixels of the image (with dimensions AKA resolution 1024*768). The data is 'x' and 'y' columns represents on which pixel the patient gazed his look on. Left diameter and right diameter represent the pupillometric measurements of the patient. They represent the diameters of the left eye pupil

and right eye pupil respectively for that particular pixel that is being stared upon. Time – this column represents the amount of time the patient stared at a particular pixel LookZone – this represents the area of an image where the patient looked at. The images consist of either faces of humans (blurred and normal) or objects (blurred and normal). Therefore, the LookZones either represent the facial features of the image (human face) or of the objects.

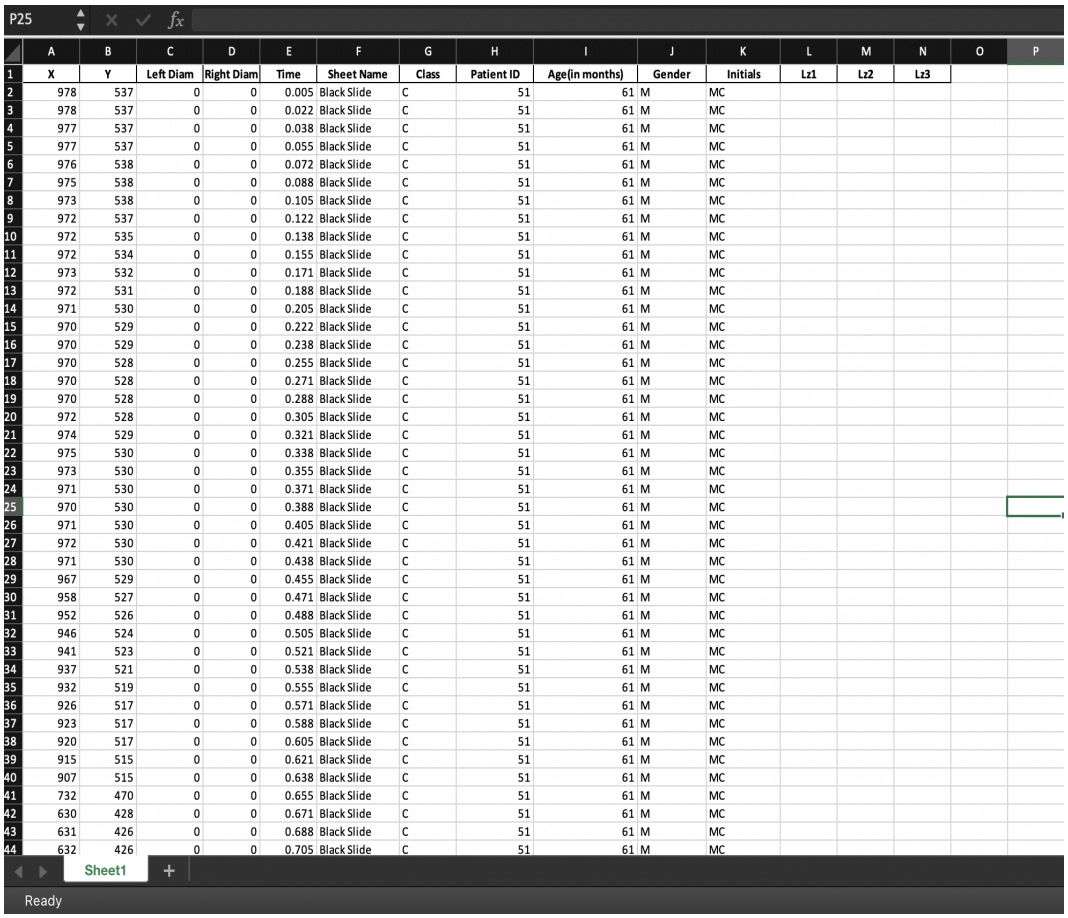


	A	B	C	D	E	F	G	H
1	visob04.jpg - G							
2	Slide	Number	x	y	Left Diam	Right Diam	Time	LZ Name
3	F:\Protocoles\ISEOP	1	556	552	3.6	0	0.014	Droite
4	F:\Protocoles\ISEOP	2	513	377	3.7	3.6	0.031	Droite
5	F:\Protocoles\ISEOP	3	526	493	3.8	3.6	0.047	Droite
6	F:\Protocoles\ISEOP	4	512	484	3.8	3.7	0.064	Droite
7	F:\Protocoles\ISEOP	5	521	527	3.7	0	0.081	Droite
8	F:\Protocoles\ISEOP	6	546	353	4	0	0.114	Droite
9	F:\Protocoles\ISEOP	7	348	428	3.7	0	0.364	Gauche
10	F:\Protocoles\ISEOP	8	271	405	3.7	3.4	0.381	Gauche
11	F:\Protocoles\ISEOP	9	274	395	3.7	3.4	0.397	Gauche
12	F:\Protocoles\ISEOP	10	282	375	3.7	3.4	0.414	Gauche
13	F:\Protocoles\ISEOP	11	275	375	3.6	3.4	0.431	Gauche
14	F:\Protocoles\ISEOP	12	277	392	3.7	3.4	0.447	Gauche
15	F:\Protocoles\ISEOP	13	254	413	3.7	3.4	0.464	Gauche
16	F:\Protocoles\ISEOP	14	266	410	3.7	3.5	0.481	Gauche
17	F:\Protocoles\ISEOP	15	263	403	3.7	3.5	0.497	Gauche
18	F:\Protocoles\ISEOP	16	335	443	3.7	0	0.514	Gauche
19	F:\Protocoles\ISEOP	17	279	412	3.7	3.5	0.531	Gauche
20	F:\Protocoles\ISEOP	18	287	383	3.7	3.6	0.547	Gauche
21	F:\Protocoles\ISEOP	19	281	359	3.7	3.6	0.564	Gauche
22	F:\Protocoles\ISEOP	20	281	351	3.7	3.6	0.581	Gauche
23	F:\Protocoles\ISEOP	21	277	349	3.7	3.9	0.597	Gauche
24	F:\Protocoles\ISEOP	22	272	350	3.7	3.9	0.614	Gauche
25	F:\Protocoles\ISEOP	23	267	351	3.7	3.9	0.631	Gauche
26	F:\Protocoles\ISEOP	24	288	412	3.7	3.5	0.647	Gauche
27	F:\Protocoles\ISEOP	25	280	436	3.7	3.4	0.664	Gauche

Figure 4.12: **An Example Raw Data File of Patient 17, Class - C**

We have to combine all the sheets inside each file and then combine all the files to make one large all data file which is easy to access for further processes using python. I have to combine the files by adding columns with patient id ,gender ,class and age and have to remove irrelevant data like sheets which ends with F. there is this no data case in some patients for some images ,it happens when that child is not looking at the image or screen. In that case that sheets has only one no data cell. So while combing all sheets it will give empty cells in remaining columns so to deal with it it should be filled with “nodata” string. And also have to split the LZ column into 3.LZ is look zone at which the patient is seeing it is in French words like tete(head),ecran(screen),bouche(mouth).I have to split it into 3 columns by reading

it from right to left to get the required order.

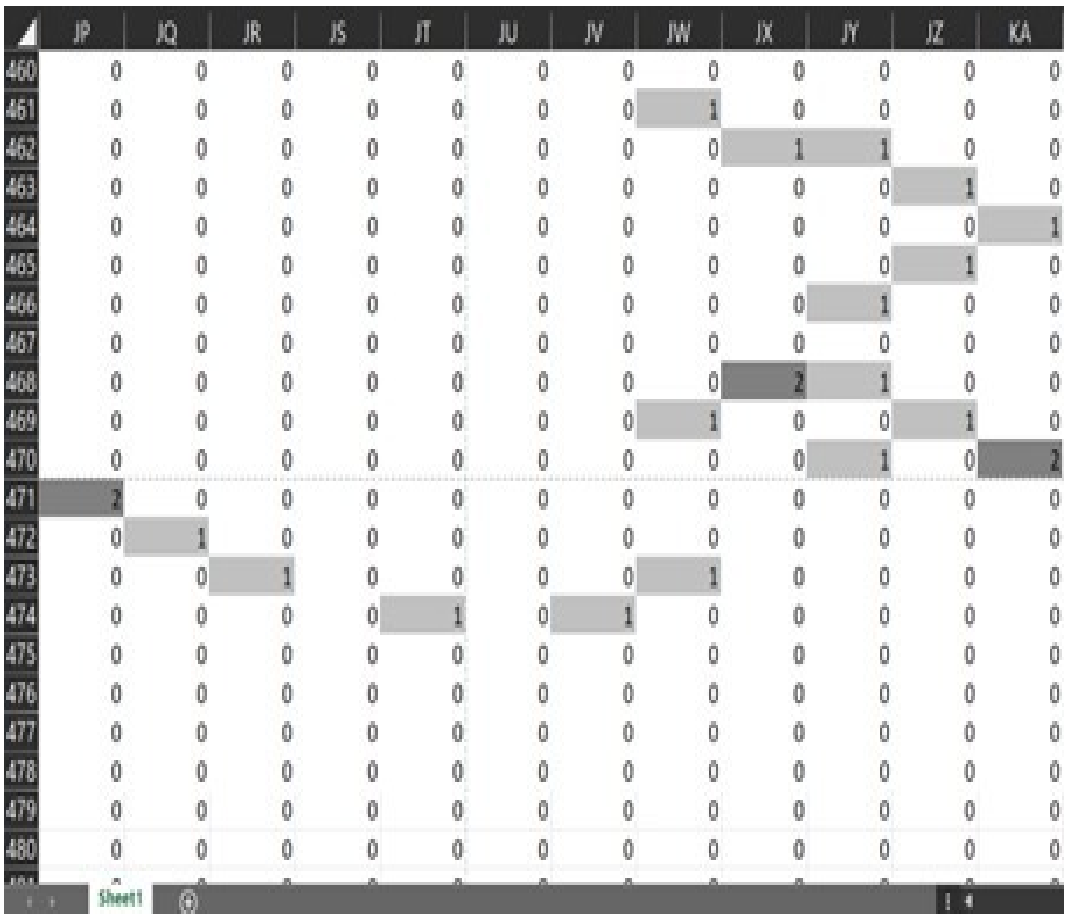


	X	Y	Left Diam	Right Diam	Time	Sheet Name	Class	Patient ID	Age(in months)	Gender	Initials	Lz1	Lz2	Lz3
2	978	537	0	0	0.005	Black Slide	C	51	61	M	MC			
3	978	537	0	0	0.022	Black Slide	C	51	61	M	MC			
4	977	537	0	0	0.038	Black Slide	C	51	61	M	MC			
5	977	537	0	0	0.055	Black Slide	C	51	61	M	MC			
6	976	538	0	0	0.072	Black Slide	C	51	61	M	MC			
7	975	538	0	0	0.088	Black Slide	C	51	61	M	MC			
8	973	538	0	0	0.105	Black Slide	C	51	61	M	MC			
9	972	537	0	0	0.122	Black Slide	C	51	61	M	MC			
10	972	535	0	0	0.138	Black Slide	C	51	61	M	MC			
11	972	534	0	0	0.155	Black Slide	C	51	61	M	MC			
12	973	532	0	0	0.171	Black Slide	C	51	61	M	MC			
13	972	531	0	0	0.188	Black Slide	C	51	61	M	MC			
14	971	530	0	0	0.205	Black Slide	C	51	61	M	MC			
15	970	529	0	0	0.222	Black Slide	C	51	61	M	MC			
16	970	529	0	0	0.238	Black Slide	C	51	61	M	MC			
17	970	528	0	0	0.255	Black Slide	C	51	61	M	MC			
18	970	528	0	0	0.271	Black Slide	C	51	61	M	MC			
19	970	528	0	0	0.288	Black Slide	C	51	61	M	MC			
20	972	528	0	0	0.305	Black Slide	C	51	61	M	MC			
21	974	529	0	0	0.321	Black Slide	C	51	61	M	MC			
22	975	530	0	0	0.338	Black Slide	C	51	61	M	MC			
23	973	530	0	0	0.355	Black Slide	C	51	61	M	MC			
24	971	530	0	0	0.371	Black Slide	C	51	61	M	MC			
25	970	530	0	0	0.388	Black Slide	C	51	61	M	MC			
26	971	530	0	0	0.405	Black Slide	C	51	61	M	MC			
27	972	530	0	0	0.421	Black Slide	C	51	61	M	MC			
28	971	530	0	0	0.438	Black Slide	C	51	61	M	MC			
29	967	529	0	0	0.455	Black Slide	C	51	61	M	MC			
30	958	527	0	0	0.471	Black Slide	C	51	61	M	MC			
31	952	526	0	0	0.488	Black Slide	C	51	61	M	MC			
32	946	524	0	0	0.505	Black Slide	C	51	61	M	MC			
33	941	523	0	0	0.521	Black Slide	C	51	61	M	MC			
34	937	521	0	0	0.538	Black Slide	C	51	61	M	MC			
35	932	519	0	0	0.555	Black Slide	C	51	61	M	MC			
36	926	517	0	0	0.571	Black Slide	C	51	61	M	MC			
37	923	517	0	0	0.588	Black Slide	C	51	61	M	MC			
38	920	517	0	0	0.605	Black Slide	C	51	61	M	MC			
39	915	515	0	0	0.621	Black Slide	C	51	61	M	MC			
40	907	515	0	0	0.638	Black Slide	C	51	61	M	MC			
41	732	470	0	0	0.655	Black Slide	C	51	61	M	MC			
42	630	428	0	0	0.671	Black Slide	C	51	61	M	MC			
43	631	426	0	0	0.688	Black Slide	C	51	61	M	MC			
44	632	426	0	0	0.705	Black Slide	C	51	61	M	MC			

Figure 4.13: **All Data File after Preprocessing**

4.4.2 Module2:Generation of Matrices

To generate the matrix we need to split that all data file into different files based on image name, patient id, class. After splitting I got around 4455 excel files and have to generate matrices (using x and y values) for all those files separately. A matrix was created for each file (for X and Y values) using 'Crosstab' function [6]. The values in the matrix shows the number of occurrences of intersection of a particular X Y value. The dimensions of each matrix were 1024*768.



	JP	JQ	JR	JS	JT	JU	JV	JW	JX	JY	JZ	KA
460	0	0	0	0	0	0	0	0	0	0	0	0
461	0	0	0	0	0	0	0	1	0	0	0	0
462	0	0	0	0	0	0	0	0	1	1	0	0
463	0	0	0	0	0	0	0	0	0	0	1	0
464	0	0	0	0	0	0	0	0	0	0	0	1
465	0	0	0	0	0	0	0	0	0	0	1	0
466	0	0	0	0	0	0	0	0	0	1	0	0
467	0	0	0	0	0	0	0	0	0	0	0	0
468	0	0	0	0	0	0	0	0	2	1	0	0
469	0	0	0	0	0	0	0	1	0	0	1	0
470	0	0	0	0	0	0	0	0	0	1	0	2
471	2	0	0	0	0	0	0	0	0	0	0	0
472	0	1	0	0	0	0	0	0	0	0	0	0
473	0	0	1	0	0	0	0	1	0	0	0	0
474	0	0	0	0	1	0	1	0	0	0	0	0
475	0	0	0	0	0	0	0	0	0	0	0	0
476	0	0	0	0	0	0	0	0	0	0	0	0
477	0	0	0	0	0	0	0	0	0	0	0	0
478	0	0	0	0	0	0	0	0	0	0	0	0
479	0	0	0	0	0	0	0	0	0	0	0	0
480	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.14: **Example of Matrix**

4.4.3 Module3: Generation of Heatmaps

Heatmaps are the representation of data in the form of a map in which the data values are represented in colours.Heatmaps can be easily accesed by machinelearning than matrices..heatmaps transforms the corelation matrix into color coding. the heatmaps produced with the original matrices are hardly visible and it is hard for the classification method to work on such sparse data, so to increse the visibility a gaussian filter is used. Gausiian diatribuation is a probability dist sym about mean, showing the data near the mean is occured more frequently than the data away from the mean. first i tried using the 5*5 gaussian matrix but then i found out that results are still very sparse to be well enough to be detected in the classification. so i applied the 7*7 gaussian matrix.

	JO	JP	JQ	JR	JS	JT	JU	JV	JW	JX	JY	JZ	KA	KB	KC
445	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
446	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
447	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
448	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
449	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
450	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
451	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
452	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
453	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
454	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
455	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
456	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
457	0	0	0	0	0	0	1	2	1	0	0	0	0	0	0
458	0	0	0	0	0	3	13	23	16	6	1	0	0	0	0
459	0	0	0	0	1	13	62	113	94	49	19	4	0	0	0
460	0	0	0	0	2	23	111	231	256	191	97	29	5	0	0
461	0	0	0	0	1	15	83	217	328	332	232	106	29	6	1
462	0	0	0	0	0	4	27	96	196	284	313	224	98	37	14
463	0	0	0	0	0	0	4	23	77	197	321	296	182	121	61
464	1	0	0	0	0	0	1	17	91	226	303	242	182	174	98
465	13	3	0	0	0	0	6	41	146	258	228	121	87	101	59
466	59	13	1	0	0	4	28	107	235	284	178	60	23	22	13
467	99	23	2	0	1	15	83	223	354	376	258	112	29	6	1
468	82	28	5	0	2	23	112	246	336	387	352	221	83	15	1
469	123	85	28	5	1	13	65	138	207	288	308	243	113	25	3
470	222	207	103	29	6	8	29	59	92	132	141	126	76	36	16
471	207	277	210	95	41	40	85	115	76	39	30	39	75	113	81
472	103	208	234	170	135	142	207	219	111	25	6	25	111	218	195
473	28	84	132	161	195	208	240	196	81	15	2	15	81	197	231
474	4	16	36	73	113	121	123	82	26	4	0	4	27	94	172
475	0	1	5	14	25	26	26	15	4	0	0	0	6	38	133
476	0	0	0	1	2	2	2	1	0	0	0	0	1	16	83
477	0	0	0	0	0	0	0	0	0	0	0	0	0	4	26
478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
479	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
480	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
481	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
482	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
483	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
484	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.15: **Example of Matrix after Gaussian filter**

Heatmaps are generated for all the files using these gaussian filtered matrices. All those heatmaps are then sorted to different files based on image id for all patients.i.e, all patients for one stim(image) id.

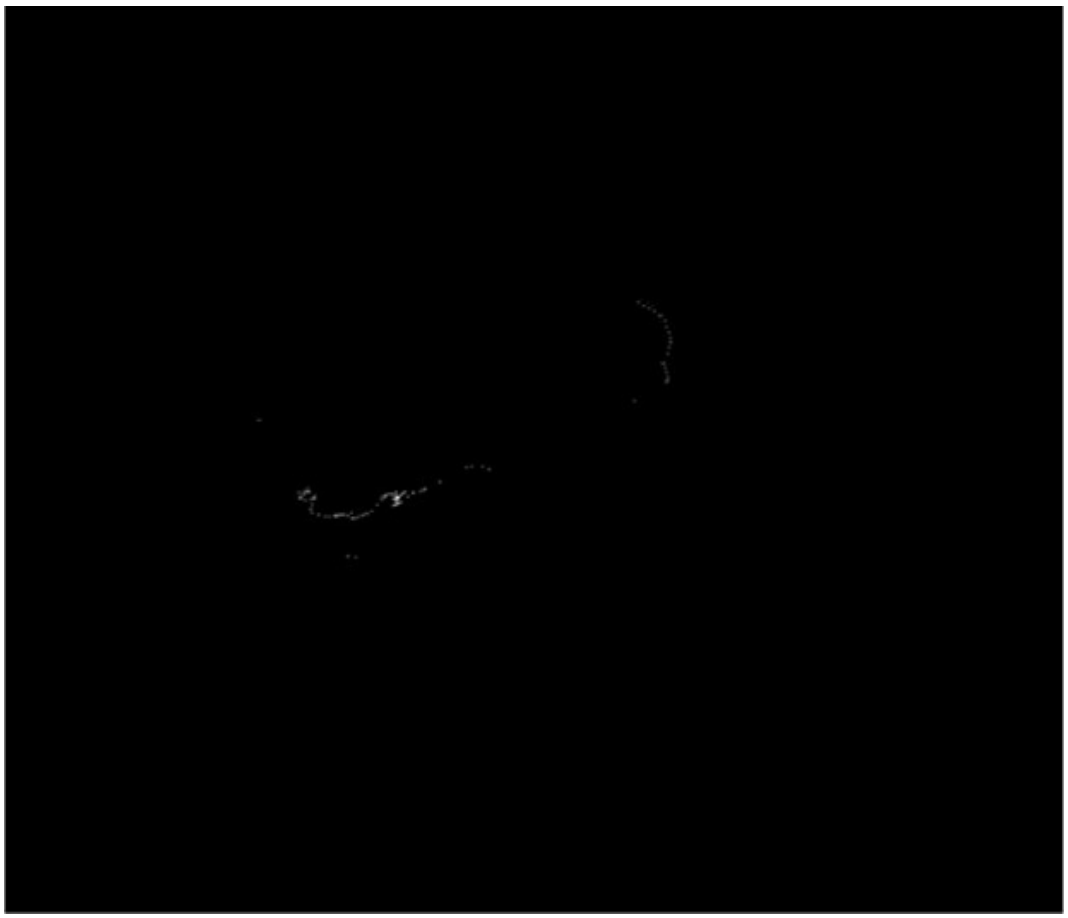


Figure 4.16: **Example of heatmap**

4.4.4 Module4: CNN classification

Convolutional neural networks (CNNs) are deep neural networks that have the capability to classify and segment images. Cnn classification is used to classify the heatmaps by training the cnn model by assigning train data set.the training dataset contains heatmaps of both children with and without autism. the model recognizes the pattern of both class c and p and that pattern is used to classify the test data. The main advantage of CNN compared to its neural networks is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs, it can learn the key features for each class by itself. the class mode illbe using is binary as there are only two classes with and without autism.

```
In [40]: validation_dataset.classes

Out[40]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1], dtype=int32)

In [41]: model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape = (200,200,3)),
                                             tf.keras.layers.MaxPool2D(2,2),
                                             #
                                             tf.keras.layers.Conv2D(32,(3,3),activation = 'relu',input_shape = (200,200,3)),
                                             tf.keras.layers.MaxPool2D(2,2),
                                             #
                                             tf.keras.layers.Conv2D(64,(3,3),activation = 'relu',input_shape = (200,200,3)),
                                             tf.keras.layers.MaxPool2D(2,2),
                                             ##
                                             tf.keras.layers.Flatten(),
                                             ##
                                             tf.keras.layers.Dense(512,activation= 'relu'),
                                             ##
                                             tf.keras.layers.Dense(1,activation= 'sigmoid')
                                             ])

In [42]: model.compile(loss = 'binary_crossentropy',
                       optimizer = RMSprop(lr=0.001),
                       metrics =['accuracy'])

In [45]: model_fit = model.fit(train_dataset,
                               steps_per_epoch = 3,
                               epochs= 20,
                               validation_data = validation_dataset)

Train for 3 steps, validate for 8 steps
Epoch 1/20
3/3 [==============================] - 2s 700ms/step - loss: 0.4524 - accuracy: 0.8889 - val_loss: 0.3664 - val_accuracy: 0.9167
Epoch 2/20
3/3 [==============================] - 2s 641ms/step - loss: 0.7067 - accuracy: 0.6667 - val_loss: 0.4593 - val_accuracy: 0.8750
Epoch 3/20
```

Figure 4.17: **CNN model**

4.5 Steps to execute/run/implement the project

4.5.1 Step1

Open Jupyter notebook in anaconda and import required libraries

4.5.2 Step2

Type code to generate heatmaps

4.5.3 Step3

Save the heatmaps in a folder which can be accesed through keras library.

4.5.4 Step4

Import the test data into jupyter notebook through keras.

4.5.5 Step5

classify the test data using CNN model

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

In order to predict the presence of markers of autism spectrum disorder the data collected from the patient should be preprocessed and converted in to heatmap. Then the heatmap of the patient is imported through tensorflow library, further passed through trained CNN model.

5.1.2 Output Design

The heat map of the Patient is passed through already trained Convolutional Neural Network model and it predicts whether the child is autistic or not by using the patterns of the heatmap.

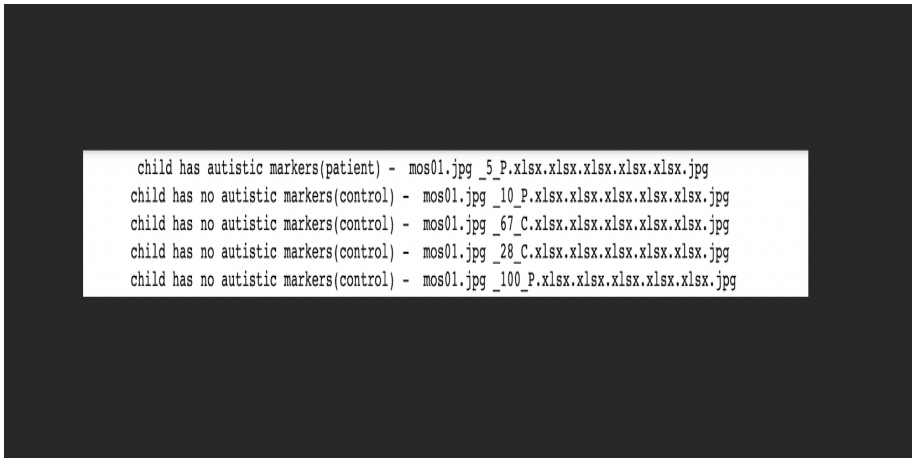


Figure 5.1: **Example of Output**

5.2 Testing

Testing is the process of assessing a system or its component(s) in order to determine whether it complies with the required specifications. It is the process of thoroughly

inspecting a framework to identify any errors, gaps, or missing pieces.

The true precondition versus necessity.

5.3 Types of Testing

5.3.1 Unit Testing

Designing test cases for unit testing ensures that the internal program logic is working correctly and that program inputs result in legitimate outputs. It is important to verify the internal code flow and all decision branches. It is the testing of the application's separate software components. Before integration, it is done following the completion of each individual unit. Unit tests carry out fundamental tests at the component level and examine a particular configuration of a system, application, or business process. As soon as the input was given to the system, it successfully completed the training. It requests that you resend the valid inputs when you send values that are outside the allowed range.

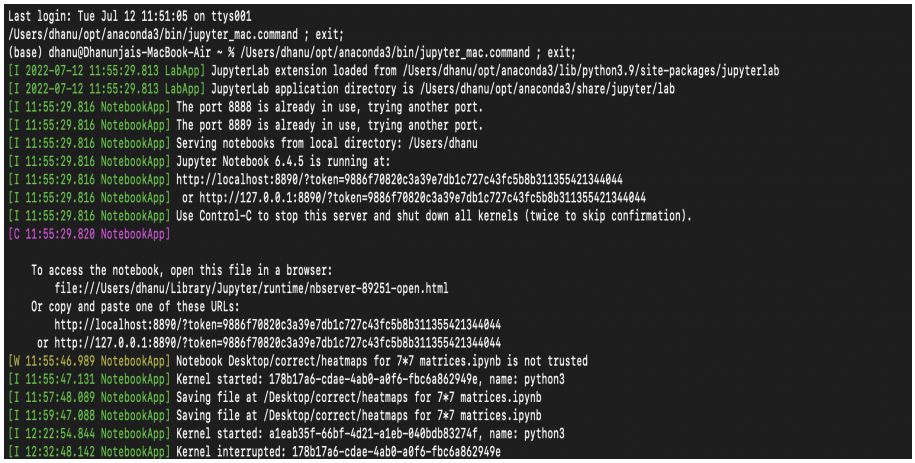


Figure 5.2: **unit testing of single class**

5.3.2 Integration Testing

Software components that have been merged are tested in integration tests to see if they genuinely operate as a single program. Testing is event-driven and focuses more on the fundamental result of screens or fields. Even though the individual components were successful in unit testing, integration tests indicate that the combination of the components is accurate and consistent. Integration testing is especially designed to highlight issues that result from combining components.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The complete system is designed using Python and the data was handled using MS Excel. For the image classification I used Convolutional Neural Network Model (CNN) as it is more accurate while recognising the data patterns than any other neural network model as it can automatically detect the important features of the image without any human intervention. The accuracy after processing the training data is around 70 percent, so we can assume that the accuracy when we process the test data will also be considerably good. The classification is done for each image for all patients to increase the accuracy even more.

6.2 Comparison of Existing and Proposed System

The diagnosis of Autism SPectrum Disorder is very cumbersome and time taking. Even with the methods that are used now, a lot of experience is required to examine a child for ASD and come to conclusion. Our project aims to solve this issue. By using machine learning and eye tracking, the children can easily be classified into two classes - with ASD symptoms and with typical growth. Hence, only the children classified under the former class need to be taken under examination to be diagnosed properly instead of examining each and every child individually in a traditional method.

6.3 Sample Code

```
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import RMSprop
import PIL

train = ImageDataGenerator(rescale = 1/255)
validation = ImageDataGenerator(rescale = 1/255)
train_dataset = train.flow_from_directory("/Users/dhanu/Desktop/cnn/train/",
                                          target_size = (200,200),
                                          batch_size = 3,
                                          class_mode = 'binary')
validation_dataset = validation.flow_from_directory("/Users/dhanu/Desktop/cnn/valid/",
                                          target_size = (200,200),
                                          batch_size = 3,
                                          class_mode = 'binary')
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape
    = (200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Conv2D(32,(3,3),activation = 'relu',input_shape =
                                        (200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Conv2D(64,(3,3),activation = 'relu',input_shape =
                                        (200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    ##
                                    tf.keras.layers.Flatten(),
                                    ##
                                    tf.keras.layers.Dense(512,activation= 'relu'),
                                    ##
                                    tf.keras.layers.Dense(1,activation= 'sigmoid')
                                    ])
model.compile(loss = 'binary_crossentropy',
              optimizer = RMSprop(lr=0.001),
              metrics =['accuracy'])
model_fit = model.fit(train_dataset,
                      steps_per_epoch = 3,
                      epochs= 20,
                      validation_data = validation_dataset)
path = '/Users/dhanu/Desktop/cnn/test'

for i in os.listdir(path):
    img = image.load_img(path +'//'+ i,target_size = (200,200))

    X = image.img_to_array(img)
    X = np.expand_dims(X,axis = 0)
    images = np.vstack([X])

    val = model.predict(images)
    if val ==0:
```

```
        print("child has no autistic markers(control) - ", i)
    else:
        print(" child has autistic markers(patient) - ", i)
```

Output

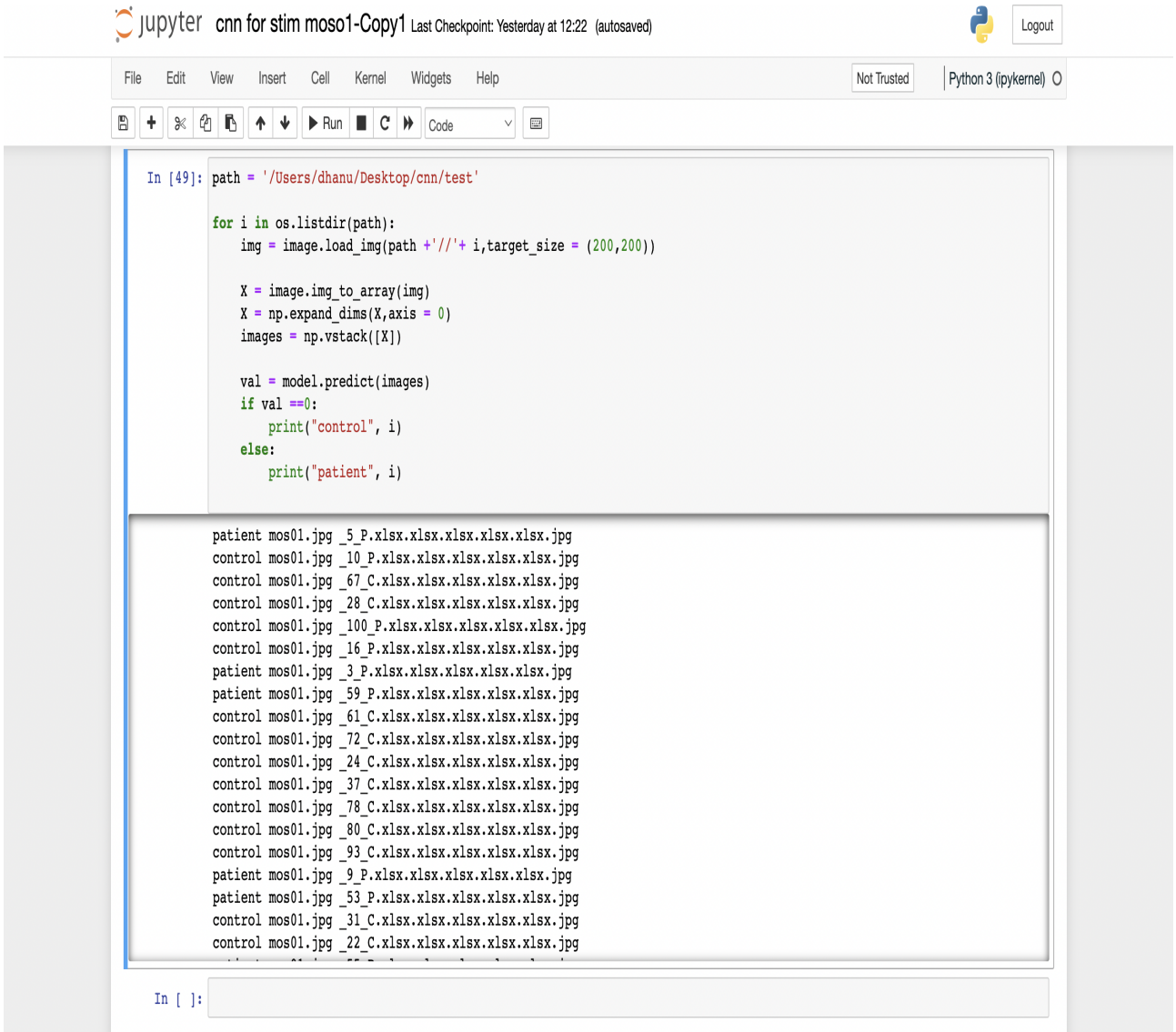


Figure 6.1: **Output 1**

Chapter 7

CONCLUSION

7.1 Conclusion

In this project, we have effectively proposed and implemented a system to detect Autism Spectrum Disorder trackers in a child through machine learning. This system helps in diagnosis of ASD in a way that is less time-taking and more effective. This project is a great addition, in many ways, to the whole process of diagnosis of ASD but is never a replacement for it. Eliminating children with no markers of autism can save up time and effort for the family, child and even the doctors since it is a very cumbersome process. With successful implementation of this project, the children with ASD and typical growth can be easily separated from each other. Hence, only the children with the ASD markers can go further for an ASD examination. After an expert examining the child, a proper diagnosis can be made.

Chapter 8

EXCHANGE PROGRAM DETAILS

8.1 Industry name : Polytech Tours (University of Tours

8.1.1 Duration of Internship (April 2022 - To july 2022)

8.1.2 4 months

8.1.3 University Address : Polytech Tours, Universite de Tour ,Tours , France - 37200

8.2 Internship offer letter

POLYTECH TOURS
polytech.univ-tours.fr

université de TOURS

Tours, February 22, 2022

Object: Confirmation of admission at the Graduate School of Engineering Polytech Tours - Academic Year 2021-2022.

Dear Dhanunjai Sai Kumar ANDEY,

The file for the scholarship will be established upon the student's arrival only.

I am pleased to announce you that the committee has decided to accept you as an exchange student in our graduate school of engineering of the University of Tours for the Academic Year 2021-2022 (until the end from March until the beginning of July 2022).

The teaching program will start at **1rst of March, 2022 to beginning on July 2022**. We advise you to be in Tours at least ten days before the beginning of the classes

According to your home country, you will have to initiate in advance the visa **request** (VISA VLS-TS "Visa de long séjour pour études- Titre de Séjour") by registering via the **mandatory online Campus France procedure** (please follow the link : http://www.campusfrance.org/fr/page/a-partir-dun-pays-a-procedure-cef)

For all administrative procedures, you are invited to keep in touch with Mrs Annabelle NOUR (annabelle.nour@univ-tours.fr) who is the administrative manager for I urs you will have to proceed to the administrative registratio onfirm you participate in the program, we will inform you about this process.

http://www.campusfrance.org/fr/page/a-partir-dun-pays-a-procedure-cef

For organization and teaching details, please contact Pr. Jean-Paul Chemla (jean-paul.chemla@univ-tours.fr) who is in charge of the international relations at the engineering school. He will help you to choose your pedagogic program.

Looking forward to seeing you in Tours in March,

Best wishes,

Jean Paul Chemla

Jean Paul Chemla
Assistant Professor
Graduate School of Engineering of the University of Tours.
Vice director in charge of International Relations
7 Avenue Marcel Dassault - 37200 Tours
Tel: +33 6 82 34 59 29

http://polytech.univ-tours

Figure 8.1: **Offer Letter**

8.3 Internship Completion certificate

Chapter 9

PLAGIARISM REPORT

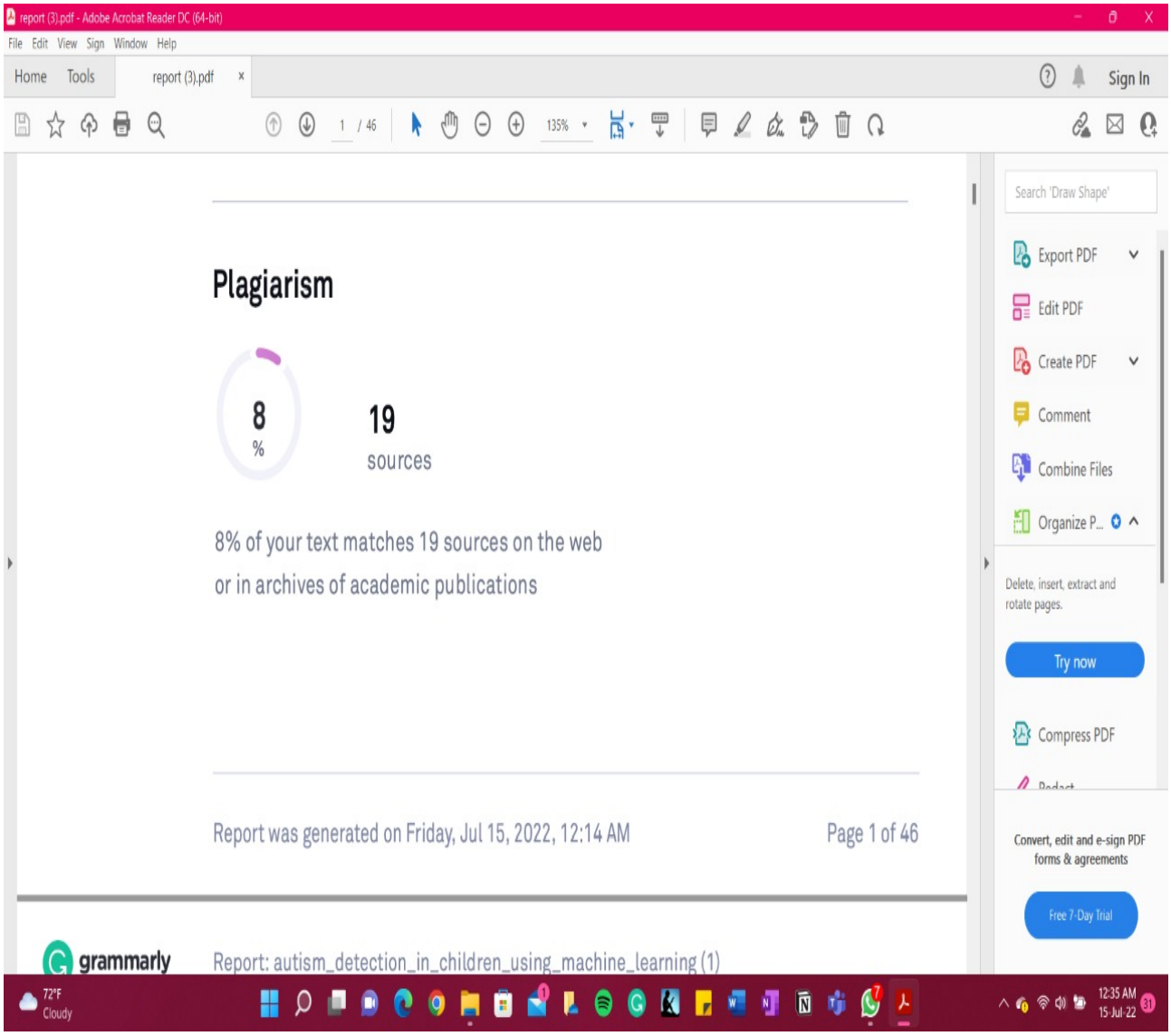


Figure 9.1: **plagarism report**

Chapter 10

SOURCE CODE & POSTER PRESENTATION

10.1 Source Code

```
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import RMSprop
import PIL
for file in os.listdir(folder):
    if file.endswith(".xlsx"):
        df = pd.read_excel(file,header = None)
        df = df.to_numpy()
        df2 = np.zeros((1024, 768),dtype=int)
        for i in range(1024):
            for j in range(768):
                if df[i][j] != 0:
                    if i > 2 and i < 1021:
                        if j > 2 and j < 765:
                            df2[i-3][j-3] = df2[i-3][j-3] + gaussian[0][0]
                            df2[i-2][j-3] = df2[i-2][j-3] + gaussian[1][0]
                            df2[i-1][j-3] = df2[i-1][j-3] + gaussian[2][0]
                            df2[i][j-3]   = df2[i][j-3]   + gaussian[3][0]
                            df2[i+1][j-3] = df2[i+1][j-3] + gaussian[4][0]
                            df2[i+2][j-3] = df2[i+2][j-3] + gaussian[5][0]
                            df2[i+3][j-3] = df2[i+3][j-3] + gaussian[6][0]
                            df2[i-3][j-2] = df2[i-3][j-2] + gaussian[0][1]
                            df2[i-2][j-2] = df2[i-2][j-2] + gaussian[1][1]
                            df2[i-1][j-2] = df2[i-1][j-2] + gaussian[2][1]
                            df2[i][j-2]   = df2[i][j-2]   + gaussian[3][1]
                            df2[i+1][j-2] = df2[i+1][j-2] + gaussian[4][1]
                            df2[i+2][j-2] = df2[i+2][j-2] + gaussian[5][1]
                            df2[i+3][j-2] = df2[i+3][j-2] + gaussian[6][1]
                            df2[i-3][j-1] = df2[i-3][j-1] + gaussian[0][2]
                            df2[i-2][j-1] = df2[i-2][j-1] + gaussian[1][2]
```

```
                            df2[i-1][j-1] = df2[i-1][j-1] + gaussian[2][2]
                            df2[i][j-1]   = df2[i][j-1]   + gaussian[3][2]
                            df2[i+1][j-1] = df2[i+1][j-1] + gaussian[4][2]
                            df2[i+2][j-1] = df2[i+2][j-1] + gaussian[5][2]
                            df2[i+3][j-1] = df2[i+3][j-1] + gaussian[6][2]
                            df2[i-3][j] = df2[i-3][j] + gaussian[0][3]
                            df2[i-2][j] = df2[i-2][j] + gaussian[1][3]
                            df2[i-1][j] = df2[i-1][j] + gaussian[2][3]
                            df2[i][j]   = df2[i][j]   + gaussian[3][3]
                            df2[i+1][j] = df2[i+1][j] + gaussian[4][3]
                            df2[i+2][j] = df2[i+2][j] + gaussian[5][3]
                            df2[i+3][j] = df2[i+3][j] + gaussian[6][3]
                            df2[i-3][j+1] = df2[i-3][j+1] + gaussian[0][4]
                            df2[i-2][j+1] = df2[i-2][j+1] + gaussian[1][4]
                            df2[i-1][j+1] = df2[i-1][j+1] + gaussian[2][4]
                            df2[i][j+1]     = df2[i][j+1]   + gaussian[3][4]
                            df2[i+1][j+1] = df2[i+1][j+1] + gaussian[4][4]
                            df2[i+2][j+1] = df2[i+2][j+1] + gaussian[5][4]
                            df2[i+3][j+1] = df2[i+3][j+1] + gaussian[6][4]
                            df2[i-3][j+2] = df2[i-3][j+2] + gaussian[0][5]
                            df2[i-2][j+2] = df2[i-2][j+2] + gaussian[1][5]
                            df2[i-1][j+2] = df2[i-1][j+2] + gaussian[2][5]
                            df2[i][j+2]     = df2[i][j+2]   + gaussian[3][5]
                            df2[i+1][j+2] = df2[i+1][j+2] + gaussian[4][5]
                            df2[i+2][j+2] = df2[i+2][j+2] + gaussian[5][5]
                            df2[i+3][j+2] = df2[i+3][j+2] + gaussian[6][5]
                            df2[i-3][j+3] = df2[i-3][j+3] + gaussian[0][6]
                            df2[i-2][j+3] = df2[i-2][j+3] + gaussian[1][6]
                            df2[i-1][j+3] = df2[i-1][j+3] + gaussian[2][6]
                            df2[i][j+3]     = df2[i][j+3]   + gaussian[3][6]
                            df2[i+1][j+3] = df2[i+1][j+3] + gaussian[4][6]
                            df2[i+2][j+3] = df2[i+2][j+3] + gaussian[5][6]
                            df2[i+3][j+3] = df2[i+3][j+3] + gaussian[6][6]
        df3 = pd.DataFrame(df2)
        output = os.path.join("/Users/dhanu/Desktop/correct 1/", file + '.xlsx')
        df3.to_excel(output ,header = None ,index = False)
        for file in os.listdir(folder):
    if file.startswith("mos01"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mos01", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mos02"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
```

```
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mos02", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mos03"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mos03", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mos04"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mos04", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mosob01"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mosob01", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mosob02"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mosob02", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mosob03"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mosob03", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("mosob04"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/mosob04", file + '.jpg')
```

```
136             plt.savefig(output)
137         elif file.startswith("ob01"):
138             df = pd.read_excel(file)
139             plt.figure(figsize=(14.236, 10.67))
140             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
141             ax.invert_yaxis()
142             plt.axis('off')
143             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/ob01", file + '.jpg')
144             plt.savefig(output)
145         elif file.startswith("ob02"):
146             df = pd.read_excel(file)
147             plt.figure(figsize=(14.236, 10.67))
148             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
149             ax.invert_yaxis()
150             plt.axis('off')
151             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/ob02", file + '.jpg')
152             plt.savefig(output)
153         elif file.startswith("ob03"):
154             df = pd.read_excel(file)
155             plt.figure(figsize=(14.236, 10.67))
156             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
157             ax.invert_yaxis()
158             plt.axis('off')
159             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/ob03", file + '.jpg')
160             plt.savefig(output)
161         elif file.startswith("ob04"):
162             df = pd.read_excel(file)
163             plt.figure(figsize=(14.236, 10.67))
164             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
165             ax.invert_yaxis()
166             plt.axis('off')
167             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/ob04", file + '.jpg')
168             plt.savefig(output)
169         elif file.startswith("visob01"):
170             df = pd.read_excel(file)
171             plt.figure(figsize=(14.236, 10.67))
172             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
173             ax.invert_yaxis()
174             plt.axis('off')
175             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/visob01", file + '.jpg')
176             plt.savefig(output)
177         elif file.startswith("visob02"):
178             df = pd.read_excel(file)
179             plt.figure(figsize=(14.236, 10.67))
180             ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
181             ax.invert_yaxis()
182             plt.axis('off')
183             output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/visob02", file + '.jpg')
184             plt.savefig(output)
185         elif file.startswith("visob03"):
```

```
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/visob03", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("visob04"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/visob04", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vis01"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vis01", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vis02"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vis02", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vis03"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vis03", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vis04"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vis04", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vismos01"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
```

```
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vismos01", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vismos02"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vismos02", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vismos03"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vismos03", file + '.jpg')
        plt.savefig(output)
    elif file.startswith("vismos04"):
        df = pd.read_excel(file)
        plt.figure(figsize=(14.236, 10.67))
        ax = sns.heatmap(df,cmap ="Greys_r",cbar = False)
        ax.invert_yaxis()
        plt.axis('off')
        output = os.path.join("/Users/dhanu/Desktop/heatmaps (mat)/vismos04", file + '.jpg')
        plt.savefig(output)

train = ImageDataGenerator(rescale = 1/255)
validation = ImageDataGenerator(rescale = 1/255)
train_dataset = train.flow_from_directory("/Users/dhanu/Desktop/cnn/train/",
                                          target_size = (200,200),
                                          batch_size = 3,
                                          class_mode = 'binary')
validation_dataset = validation.flow_from_directory("/Users/dhanu/Desktop/cnn/valid/",
                                          target_size = (200,200),
                                          batch_size = 3,
                                          class_mode = 'binary')
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape
    = (200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Conv2D(32,(3,3),activation = 'relu',input_shape =
                                        (200,200,3)),
                                    tf.keras.layers.MaxPool2D(2,2),
                                    #
                                    tf.keras.layers.Conv2D(64,(3,3),activation = 'relu',input_shape =
                                        (200,200,3)),
```

```
                                    tf.keras.layers.MaxPool2D(2,2),
                                    ##
                                    tf.keras.layers.Flatten(),
                                    ##
                                    tf.keras.layers.Dense(512,activation= 'relu'),
                                    ##
                                    tf.keras.layers.Dense(1,activation= 'sigmoid')
                                    ])
model.compile(loss = 'binary_crossentropy',
              optimizer = RMSprop(lr=0.001),
              metrics =['accuracy'])
model_fit = model.fit(train_dataset,
                      steps_per_epoch = 3,
                      epochs= 20,
                      validation_data = validation_dataset)
path = '/Users/dhanu/Desktop/cnn/test'

for i in os.listdir(path):
    img = image.load_img(path +'//'+ i,target_size = (200,200))

    X = image.img_to_array(img)
    X = np.expand_dims(X,axis = 0)
    images = np.vstack([X])

    val = model.predict(images)
    if val ==0:
        print("child has no autistic markers(control) - ", i)
    else:
        print(" child has autistic markers(patient) - ", i)
```

10.2 Poster Presentation

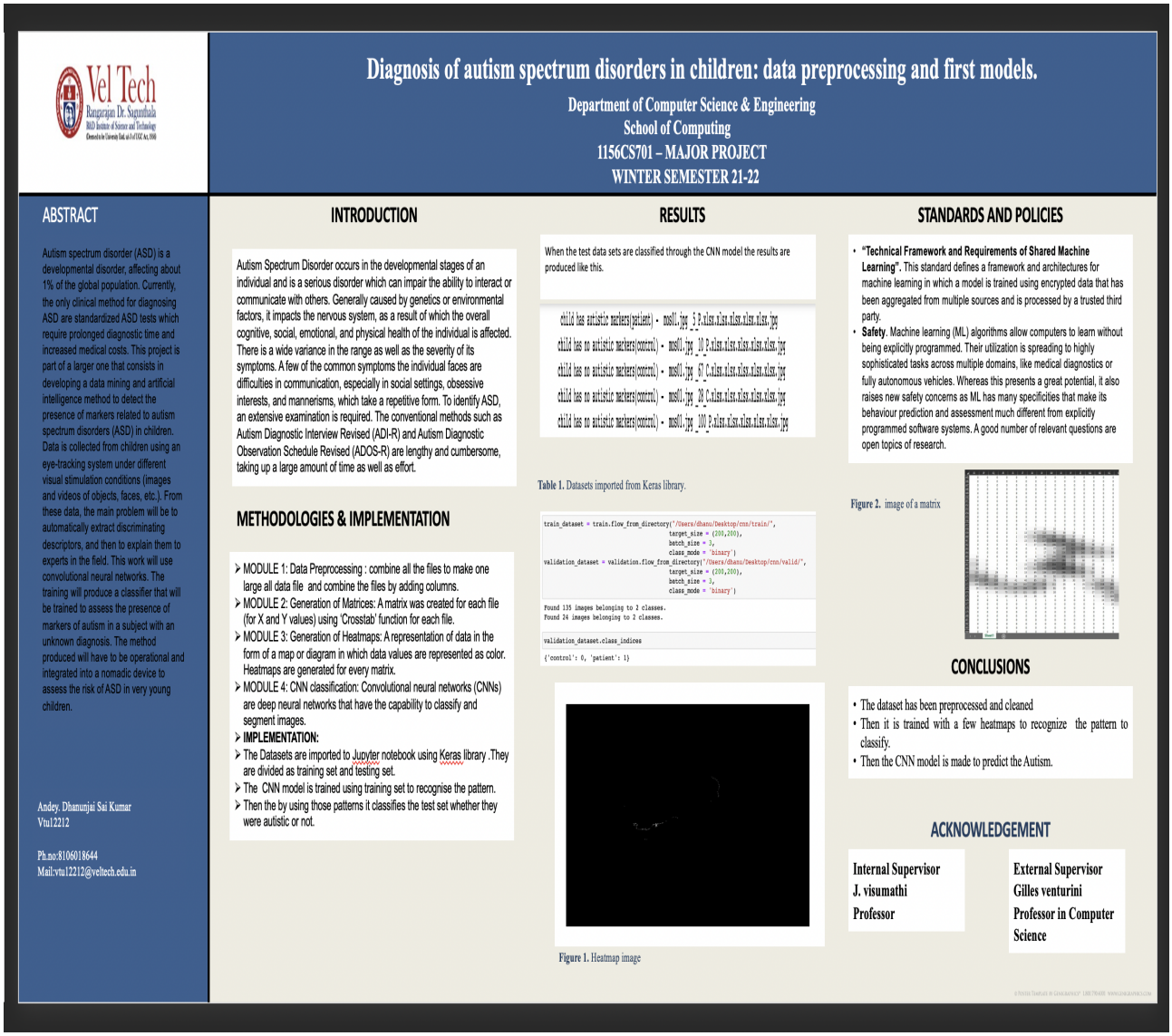


Figure 10.1: **Poster Presentation**

References

[1] Vakadkar, Kaushik, Diya Purkayastha, and Deepa Krishnan. ”Detection of Autism Spectrum Disorder in Children Using Machine Learning Techniques.” SN Computer Science 2.5 (2021): 1-9.

[2] Thabtah, Fadi. ”Machine learning in autistic spectrum disorder behavioral research: A review and ways forward.” Informatics for Health and Social Care 44.3 (2019): 278-297.

[3] Hossain, M.D., Kabir, M.A., Anwar, A. and Islam, M.Z., 2021. Detecting autism spectrum disorder using machine learning techniques. Health Information Science and Systems, 9(1), pp.1-13.

[4] Hus Y, Segal O. Challenges Surrounding the Diagnosis of Autism in Children. Neuropsychiatr Dis Treat. 2021;17:3509-3529. Published 2021 Dec 3. doi:10.2147/NDT.S282569

[5] Liu, W., Yu, X., Raj, B., Yi, L., Zou, X. and Li, M., 2015, September. Efficient autism spectrum disorder prediction with eye movement: A machine learning framework. In 2015 International conference on affective computing and intelligent interaction (ACII) (pp. 649-655). IEEE.

[6] Raschka, S., 2015. Python machine learning. Packt publishing ltd.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, Chapter 4, 12, pp.2825-2830.

[8] Permuter, H., Francos, J. and Jermyn, I., 2006. A study of Gaussian mixture models of color and texture features for image classification and segmentation. Pattern recognition, 39(4), pp.695-706.

[9] Lee, K., Xu, W., Fan, F. and Tu, Z., 2018. Wasserstein introspective neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3702-3711).

[10] Alwidian, J., Elhassan, A. and Ghnemat, R., 2020. Predicting autism spectrum disorder using machine learning technique. International Journal of Recent Technology and Engineering, 8(5), pp.4139-4143.

[11] Thabtah, F., 2019. An accessible and efficient autism screening method for behavioural data and predictive analyses. Health informatics journal, 25(4), pp.1739-1755.

[12] Heinsfeld, A.S., Franco, A.R., Craddock, R.C., Buchweitz, A. and Meneguzzi, F., 2018. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. NeuroImage: Clinical, 17, pp.16-23.