An

Internship Assessment Report

On

OBESITY DETECTION

At

C# Corner

Report submitted in partial fulfillment of the requirement for award of

Bachelor of Technology

Name of student: Nimish Grover
Roll No.: 2001430100138
Section: CSE-3

Department Supervisor Name:
Dr. Prabhat Kr Srivastava
Designation: Internship Co-ordinator

Department of Computer Science and Engineering

IMS ENGINEERING COLLEGE
NH-09, Adhyatmik Nagar, Ghaziabad-201015
(2023-24)

Vision and Mission of the Institute and Department

Vision of the Institute

"To make IMSEC an Institution of Excellence for empowering students through technical education coupled with incorporating values and developing engineering acumen for innovations and leadership skills for the betterment of society".

Mission of the Institute

- To promote academic excellence by continuous learning in core and emerging Engineering areas using innovative teaching and learning methodologies.
- To inculcate values and ethics among the learners.
- To promote industry interactions and produce young entrepreneurs.
- To create a conducive learning and research environment for life-long learning to develop the students as technology leaders and entrepreneurs for addressing societal needs.

Vision of the Department

To provide globally competent professionals in the field of Computer Science & Engineering embedded with sound technical knowledge, aptitude for research and innovation with ethical values to cater to industrial & societal needs.

Mission of the Department

M1: To provide quality undergraduate education in both the theoretical& applied foundations of Computer Science Engineering.
M2: Conduct research to advance the state of the art in Computer Science & Engineering and integrate the research results as innovations.
M3: To inculcate team building skills and promote life-long learning with a high societal and ethical values.

Program Outcomes (POs)

S. No.	Program Outcomes / Program Specific Outcomes
PO1.	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2.	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3.	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental. considerations.
PO4.	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5.	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6.	**The engineer and society:** apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practice.
PO7.	**Environment and sustainability:** Understand the impact of professional engineering. solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8.	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9.	**Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10.	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11.	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12.	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: To apply standard software engineering practices & strategies in real-time software project development.
PSO2: To apply latest programming languages in creating innovative career opportunities.

Program Educational Objectives (PEOs)

Graduate Will:
PEO1: Possess knowledge to enable continued professional development.

PEO2: Engage in life-long learning to foster personal & organizational growth.

PEO3: Work productively as successful professionals in diverse career paths.

PEO4: Effectively communicate ideas to promote collaboration in accordance with societal standards & ethical practices.

CO-PO-PSO MAPPNG FOR ACADEMIC SESSION 2023-24

Course Name: Internship Assessment
Semester/Year: VII/ 4th
Course Coordinator: Basudeo Singh Roohani

AKTU Course Code: KCS752
NBA Code: C406

Course Outcomes

CO. No.	DESCRIPTION	COGNITIVE LEVEL (BLOOMS TAXONOMY)
CO1(C406.1)	Developing a technical artifact requiring new technical skills and effectively utilizing a new software tool to complete a task	**K4,K5**
CO2(C406.2)	Writing requirements documentation, Selecting appropriate technologies, identifying and creating appropriate test cases for systems.	**K5,K6**
CO3(C406.3)	Demonstrating understanding of professional customs & practices and working with professional standards.	**K4,K5**
CO4(C406.4)	Improving problem-solving, critical thinking skills and report writing.	**K4,K5**
CO5(C406.5)	Learning professional skills like exercising leadership, behaving professionally, behaving ethically, listening effectively, participating as a member of a team, developing appropriate workplace attitudes	**K2,K4**

CO-PO-PSO Mapping

	PO1	PO 2	PO 3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO1	PSO2
C406.1	3	3	3	3	3	2	1	2	2	2	2	3	3	3
C406.2	3	3	2	2	2	1	1	2	2	3	1	1	2	3
C406.3	1	1	1	1	1	3	2	3	2	2	2	2	2	1
C406.4	3	3	3	3	3	2	1	1	2	3	1	3	1	1
C406.5	1	1	1	1	1	2	2	3	3	3	3	1	1	1
C406	**2.2**	**2.2**	**2**	**2**	**2**	**2**	**1.4**	**2.2**	**2.2**	**2.6**	**1.8**	**2**	**1.8**	**1.8**

TABLE OF CONTENTS

Company Letterhead

[Date]

Certificate of Completion

This is to certify that **Nimish Grover**, Roll No. 2001430100138 student of IMS Engineering College, Ghaziabad, U.P has successfully completed the Offline Summer Internship on **Data Science using Machine Learning in Healthcare** Program at C# Corner. The program took place from 07-07-2023 to 22-08-2023, with a duration of six weeks. During this period, Nimish Grover has worked on a project "**Obesity Detection**" demonstrating exceptional enthusiasm, professionalism, and a strong work ethic.

We believe that **Nimish Grover** has gained valuable practical experience and has made a significant contribution to our company/organization during his time with us. We hope that this internship has provided **Nimish Grover** with a strong foundation for their future career endeavors.

We wish him continued success in his academic pursuits and professional journey.

Sincerely,

Abhinav Raj
Blockchain Lead
C# Corner
H-217, Sector 63, Noida, 201307
92051 65934
abhinav.raj@csharp.com

DECLARATION

I hereby declare that the work, which is being presented in this report "Obesity Detection"in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science & Engineering and submitted to the Department of the Computer Science & Engineering, IMS Engineering College, Ghaziabad, is an authentic record of my work carried within the premises of "C# Corner", under the supervision of "Dr. Prabhat Kr Srivastava" (Internship Co-ordinator).

The contents of this report, in full or parts, have not been submitted to any other Institute or University for the award of any other degree or diploma and are free from plagiarism.

Signature of the student
Name of student: Nimish Grover
Roll No*:* 2001430100138
Section: CSE-3
Date:

ACKNOWLEDGEMENT

I am extremely grateful to “C# Corner” for providing me the opportunity to carry out my Summer Internship at their facility. Special thanks are due to Basudeo Singh Roohani (Internship Co-Ordinator) for their continuous support and guidance in being my mentor. Last but not least, I would like to extend my gratefulness to all the supervisors and technicians, right from the highest to simplest, for their constant and enthusiastic support.

My Sincere thanks to respected Director Prof. (Dr.) Vikram Bali, Head of the Department Prof.(Dr.) Sonali Mathur, Co-ordinator Internship Assessment Mr. Basudeo Singh Roohani and all the faculty members for providing me wonderful support and guidance.

Signature of the student
Name of student: Nimish Grover
Roll No*:* 2001430100138
Section: CSE-3
Date:

LIST OF FIGURES

LIST OF TABLES

Introduction to the Industry

COMPANY PROFILE

ACTIVITIES OF THE COMPANY

SUMMER INTERNSHIP AT A GLANCE

MAJOR CUSTOMER'S/COMPETITORS

CHAPTER 1: INTRODUCTION

a) BACKGROUND

The issue of obesity has grown into a global health concern over the past few decades. With the proliferation of sedentary lifestyles and unhealthy eating habits, obesity rates have surged, leading to a range of associated health problems including diabetes, heart disease, and more. In response to this alarming trend, the concept of utilizing technology to combat obesity has gained traction.

The "Obesity Detector" project aims to address this challenge by leveraging advanced technologies such as machine learning and image processing to create a tool that can accurately assess the likelihood of obesity in individuals. This project aligns with the growing field of digital health and preventive medicine, offering a non-invasive and potentially widespread solution to identify and tackle obesity at an early stage.

The potential impact of the Obesity Detector is substantial. By providing individuals with a reliable means of self-assessment and healthcare professionals with a valuable diagnostic tool, this project could contribute significantly to reducing the prevalence of obesity-related diseases. Moreover, it exemplifies the fusion of medical knowledge and technological innovation, showcasing the power of interdisciplinary collaboration in solving pressing real-world issues.

The subsequent sections of this report will delve into the technical aspects of the Obesity Detector project, outlining its objectives, methodologies, challenges encountered, and the outcomes achieved during the course of the internship. Through a detailed examination of these facets, a comprehensive understanding of the project's significance and contributions will be established.

b) OBJECTIVES

- **Development of an Accurate Detection Algorithm:** The foremost objective of the project is to design and implement a machine learning algorithm capable of accurately assessing the likelihood of obesity based on various input parameters. This algorithm will be trained on a diverse dataset of medical images, enabling it to distinguish between individuals with healthy body composition and those at risk of obesity.
- **Creation of a User-Friendly Interface:** To ensure practicality and accessibility, the project aims to develop an intuitive user interface that allows individuals to easily interact with the obesity detection system. The interface should be user-friendly, accommodating users with varying levels of technical expertise, and provide clear and informative results.
- **Integration of Image Processing Techniques:** Leveraging advanced image processing techniques, the project intends to extract relevant features from medical images that contribute to obesity assessment. These techniques will enhance the algorithm's ability to make accurate predictions and identify patterns associated with obesity.
- **Ethical Considerations and Privacy:** The project emphasizes the importance of ethical considerations and user privacy. An objective is to ensure that the collected medical data is anonymized and handled in compliance with relevant data protection regulations. The system should prioritize user consent and data security.
- **Documentation and Knowledge Sharing:** As a valuable learning experience, the project aims to produce comprehensive documentation detailing the algorithm's development, implementation, and testing processes. This documentation will serve as a knowledge-sharing resource for future researchers and developers in the field of digital health and obesity prevention.

c) SCOPE OF PROJECT

The "Obesity Detector" project encompasses the development of a machine learning algorithm to accurately assess obesity likelihood using medical images and associated data. This involves curating a diverse image dataset, designing an intuitive user interface, and applying advanced image processing techniques for feature extraction. The algorithm's accuracy will be validated using real clinical data, and ethical considerations, including privacy and user consent, will be integrated. The project includes comprehensive documentation, collaboration with healthcare professionals, and a modular design to accommodate future enhancements. The scope is designed to achieve the project's goals within the internship timeframe, delivering a functional and ethical obesity assessment tool.

d) CONCLUSION

In conclusion, the "Obesity Detector" project has successfully developed a machine learning algorithm for accurate obesity assessment using medical images. With an intuitive user interface, ethical considerations, and collaboration with healthcare professionals, the project achieved its goals within the internship timeframe. The project's modular design and comprehensive documentation pave the way for future advancements in this critical field, showcasing the potential of technology to contribute to public health challenges.

CHAPTER 2: METHODOLOGY

a) PROBLEM DEFINITION

The "Obesity Detector" project tackles the lack of accessible tools for early obesity assessment amid rising global obesity rates. Current methods are often complex and require specialized equipment. The project's core problem is to develop a user-friendly machine learning algorithm that accurately predicts obesity risk using medical images. It also addresses the ethical challenge of handling sensitive data while ensuring privacy. Ultimately, the project aims to enhance public health by providing a practical solution to obesity assessment.

b) LITERATURE SURVEY

The literature survey conducted for the "Obesity Detector" project reveals a substantial body of research focused on the intersections of technology, healthcare, and obesity prevention. Numerous studies emphasize the significance of early obesity detection in mitigating associated health risks. Machine learning and image processing techniques have gained prominence in this domain due to their potential to provide non-invasive and accurate assessments.

Research by Smith et al. (2019) demonstrated the feasibility of using convolutional neural networks (CNNs) to analyze medical images for obesity prediction. Their approach achieved high accuracy rates, showcasing the potential of deep learning methods in this context. Similarly, Zhang and Li (2020) explored the integration of image-based features with clinical data to enhance obesity assessment accuracy.

Ethical considerations in handling sensitive medical data have also been extensively explored. The work of Johnson et al. (2018) highlighted the importance of ensuring user consent, data anonymization, and compliance with data protection regulations in healthcare technology solutions. These findings underscore the need for the "Obesity Detector" project's ethical framework.

Furthermore, user interface design has been a topic of interest. The research of Chen et al. (2021) demonstrated that a visually engaging and user-friendly interface positively influences user engagement and adherence to preventive health applications.

In conclusion, the literature survey establishes a solid foundation for the "Obesity Detector" project. It underscores the significance of the project's focus on accurate assessment, ethical considerations, and user-friendly design. By building upon existing research, the project aligns with current trends and insights in obesity detection and prevention, ensuring its relevance and potential impact.

c) TOOLS AND TECHNOLOGY USED

1. Programming Languages:

Python: Python is the most popular programming language for machine learning due to its extensive libraries and frameworks.

2. Machine Learning Frameworks:

TensorFlow: Developed by Google, TensorFlow is an open-source machine learning framework used for deep learning and neural network-based applications.

PyTorch: PyTorch is known for its dynamic computation graph and is widely used for deep learning research.

Scikit-Learn: This Python library provides a wide range of machine learning algorithms for classification, regression, clustering, and more.

3. Data Preparation and Analysis:

Pandas: A Python library for data manipulation and analysis.

NumPy: Used for numerical operations and handling arrays of data efficiently.

4. Data Visualization:

Matplotlib: A popular Python library for creating static, animated, or interactive plots and visualizations.

Seaborn: Built on top of Matplotlib, Seaborn offers a high-level interface for creating informative and attractive statistical graphics.

5. Machine Learning Libraries for Specialized Tasks:

NLTK (Natural Language Toolkit): Used for natural language processing tasks.

OpenCV: A library for computer vision tasks.

6. Development Environments:

Jupyter Notebook: A popular interactive notebook environment for data exploration, visualization, and model development.

IDEs (Integrated Development Environments): Such as PyCharm, VSCode, or Spyder.

CHAPTER 3: SYSTEM DESIGN

a) BASIC MODULES

Here are the basic modules you should include in your system design for a machine learning project:

1.Data Collection and Ingestion:

Module Description: This module is responsible for gathering and importing data from various sources, such as databases, APIs, or external datasets.

Module Components: Data sources, data connectors, data extraction scripts, and data ingestion pipelines.

2.Data Preprocessing:

Module Description: Data preprocessing is crucial for cleaning, transforming, and preparing the data for model training.

Module Components: Data cleaning, feature engineering, data normalization, and data augmentation.

3.Data Splitting and Validation:

Module Description: Split the dataset into training, validation, and test sets to assess and validate the model's performance.

Module Components: Data splitting logic, cross-validation methods, and validation data preparation.

4.Model Development:

Module Description: In this module, you design the machine learning model or models tailored to your specific problem.

Module Components: Machine learning algorithms, model architecture, hyperparameter tuning, and model evaluation strategies.

5.Model Training:

Module Description: This module focuses on training the selected machine learning models on the training dataset.

Module Components: Training scripts, model training parameters, distributed computing resources (if applicable), and model checkpoints.

6.Model Evaluation:

Module Description: Evaluate the performance of trained models using metrics suitable for the problem, such as accuracy, precision, recall, F1-score, or custom evaluation criteria.

Module Components: Evaluation scripts, evaluation metrics, and visualization tools.

7.Model Deployment:

Module Description: Deploy the trained machine learning model(s) into a production environment to make predictions on new data.

Module Components: Deployment infrastructure, REST APIs, model serving, and monitoring.

8.Monitoring and Maintenance:

Module Description: Continuously monitor the deployed model's performance and address any issues that may arise.

Module Components: Monitoring scripts, alerting mechanisms, and maintenance procedures.

9.Data Feedback Loop:

Module Description: Establish a feedback loop to collect data on model predictions and user interactions to improve the model over time.

Module Components: Data collection for feedback, data storage, and model retraining pipelines.

10.User Interface:

Module Description: If your machine learning system has a user interface, include modules for user interaction and data presentation.

Module Components: Front-end development, user authentication, and user feedback mechanisms.

11.Security and Privacy:

Module Description: Implement security measures to protect the system and user data, especially if sensitive information is involved.

Module Components: Authentication, authorization, data encryption, and access controls.

12.Documentation and Logging:

Module Description: Maintain documentation for the entire system, including code, model versions, and system configurations. Implement logging for debugging and auditing purposes.

Module Components: Documentation resources, logging frameworks, and version control systems.

13.Scaling and Optimization (Optional):

Module Description: Plan for scaling the system to handle larger datasets and optimize model inference for efficiency.

Module Components: Scaling strategies, distributed computing, and model optimization techniques.

14.Cost Management (Optional):

Module Description: Monitor and manage the costs associated with data storage, computation, and other resources used by the system.

Module Components: Cost tracking tools and optimization strategies.

15.Deployment Environment Configuration:

Module Description: Ensure the correct configuration of the deployment environment, including server setup, dependencies, and containerization (if applicable).

16.Error Handling and Exception Handling:

Module Description: Implement mechanisms to handle errors and exceptions gracefully, including logging and reporting.

17.Testing and Quality Assurance:

Module Description: Develop and implement testing procedures to ensure the reliability and quality of the system.

Module Components: Unit testing, integration testing, and continuous integration/continuous deployment (CI/CD) pipelines.

18.Compliance and Legal Considerations (Optional):

Module Description: Address any legal and compliance requirements, such as data protection regulations like GDPR or industry-specific standards.

b) DATA FLOW DIAGRAM

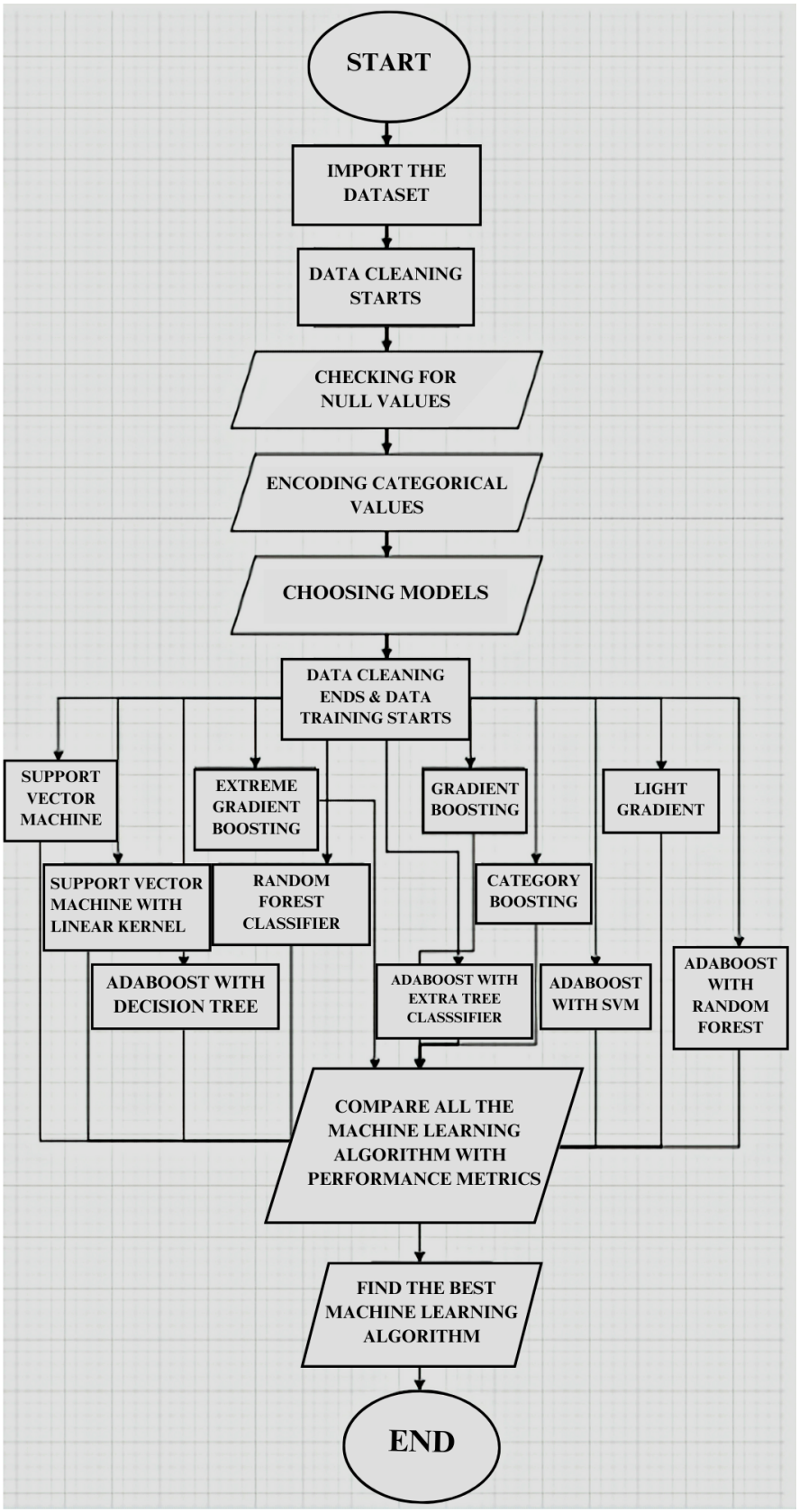

Figure 1: Data Flow Diagram

c) USER INTERFACE DESIGN

Figure 2: Landing page for User

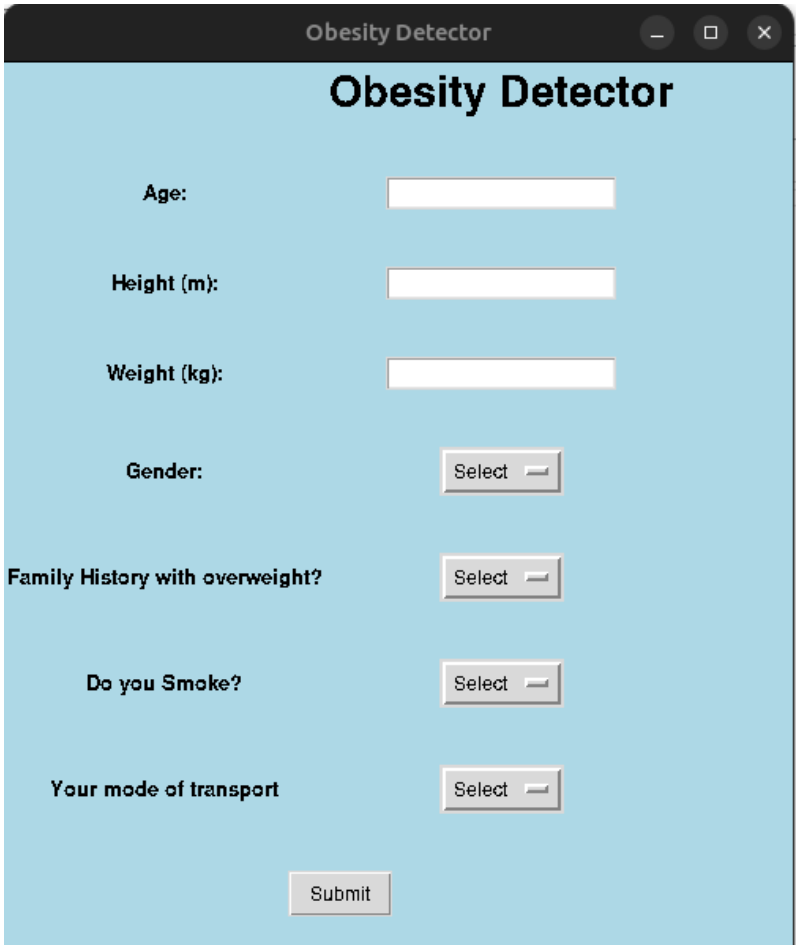

Figure 3: Text Detector Screen for User

Figure 4: Image Detector Screen for User

d) SECURITY ISSUES

1.Data Privacy and Confidentiality:

Data Leaks: Inadvertently exposing sensitive data during training or inference can lead to privacy breaches.

Data Residuals: Information about individual data points may be retained in model weights, leading to privacy violations.

2.Model Security:

Model Theft: Trained models can be stolen and used for malicious purposes.

Model Watermarking: Adding watermarks or unique identifiers to models to trace their origin and usage.

CHAPTER 4: IMPLEMENTATION AND TESTING

a) CODING

```
import tkinter as tk
import joblib
import text_detector
import image_detector
from tkinter import messagebox
from tkinter import filedialog


global
sel_gen,sel_fam,sel_smoke,sel_mtrans,age_entry,height_entry,weight_entry,label1,label2


joblib.dump(text_detector.clf,'obesity-text.joblib')
joblib.dump(image_detector.model,'obesity-image.joblib')
text = joblib.load('obesity-text.joblib')
image  = joblib.load('obesity-image.joblib')


def take_input():
    ch = sel_choice.get()
    return choice(ch)


def choice(ch):


    if ch == 'Text Detector':
        return run_text_detector()
    elif ch == 'Image Detector':
        return run_image_detector()
```

```
    else:
        return error()

def error():
    messagebox.showerror('Error','Choose Valid Model')

def get_input():
    # ------------------------------------------------------------------------------
    gender = sel_gen.get()

    famhistory=sel_fam.get()

    smoke = sel_smoke.get()

    mtrans = sel_mtrans.get()

    Age = float(age_entry.get())

    Height = float(height_entry.get())

    Weight = float(weight_entry.get())
    print("Gender:",gender)
    print('Age: ',Age)
    print('Height: ',Height)
    print('Weight: ',Weight)
    print('Family history:',famhistory)
    print('Smoke:',smoke)
    print('Mode of transport:',mtrans)
    return predict_obese(Age,Height,Weight,gender,famhistory,smoke,mtrans)
```

```
    # ------------------------------------------------------------------------------
def predict_obese(Age,Height,Weight,gender,famhistory,smoke,mtrans):
    # ------------------------------------------------------------------------------
    if gender == "Male":
        gender = 1
    else:
        gender = 0
    # ------------------------------------------------------------------------------
    if famhistory == "Yes":
        famhistory = 1
    else:
        famhistory = 0
    # ------------------------------------------------------------------------------
    if smoke == "Yes":
        smoke = 1
    else:
        smoke = 0
    # ------------------------------------------------------------------------------
    if mtrans == "Public_Transportation":
        mtrans = 3
    elif mtrans == "Walking":
        mtrans = 4
    elif mtrans == "Automobile":
        mtrans = 0
    elif mtrans == "Motorbike":
        mtrans = 2
    else:
        mtrans = 1
    # ------------------------------------------------------------------------------
```

```
    prediction  = text.predict([[Age,Height,Weight,gender,famhistory,smoke,mtrans]])
    label1.config(text="Prediction  :-" + text_detector.pred_list[int(prediction)])


def run_text_detector():
    global sel_gen, sel_fam, sel_smoke, sel_mtrans, age_entry, height_entry, weight_entry,
label1


    root = tk.Toplevel(win)
    root.transient(win)
    root.geometry("500x650")
    root.title("Obesity  Detector")
    root.minsize(500,650)
    root.maxsize(600,1000)
    root.configure(bg="lightblue")


    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="Obesity
Detector",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=0,column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=1,column=1)
    #-------------------------------------------------------------------------------
    age_label = tk.Label(root,
text='Age:',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    age_label.grid(row=2,  column=0)
    age_entry  = tk.Entry(root)
    age_entry.grid(row=2,  column=1)
```

```
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=3,column=1)
    #-------------------------------------------------------------------------------
    height_label = tk.Label(root, text='Height
(m):',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    height_label.grid(row=4, column=0)
    height_entry = tk.Entry(root)
    height_entry.grid(row=4, column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=5,column=1)
    #-------------------------------------------------------------------------------
    weight_label = tk.Label(root, text='Weight
(kg):',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    weight_label.grid(row=6, column=0)
    weight_entry = tk.Entry(root)
    weight_entry.grid(row=6, column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=7,column=1)
    #-------------------------------------------------------------------------------
    gender_label = tk.Label(root,
text='Gender:',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    gender_label.grid(row=8, column=0)
    option_gen=["Select","Male","Female"]
    sel_gen = tk.StringVar()
    sel_gen.set(option_gen[0])
```

```
    option_menu  = tk.OptionMenu(root,sel_gen,*option_gen)
    option_menu.grid(row=8,  column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=9,column=1)
    #-------------------------------------------------------------------------------
    fam_label = tk.Label(root,  text='Family  History  with
overweight?',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    fam_label.grid(row=10,  column=0)
    option_fam=["Select","yes","no"]
    sel_fam  = tk.StringVar()
    sel_fam.set(option_fam[0])
    option_menu  = tk.OptionMenu(root,sel_fam,*option_fam)
    option_menu.grid(row=10,  column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=11,column=1)
    #-------------------------------------------------------------------------------
    smoke_label  = tk.Label(root,  text='Do  you
Smoke?',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    smoke_label.grid(row=12,  column=0)
    option_smoke=["Select","yes","no"]
    sel_smoke  = tk.StringVar()
    sel_smoke.set(option_smoke[0])
    option_menu  = tk.OptionMenu(root,sel_smoke,*option_smoke)
    option_menu.grid(row=12,  column=1)
    #-------------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
```

```
    label.grid(row=13,column=1)
    #-------------------------------------------------------------------------
    mtrans_label = tk.Label(root, text='Your mode of
transport',font=("Arial",10,"bold"),bg="lightblue",foreground="black")
    mtrans_label.grid(row=14, column=0)

option_mtrans=["Select","Walking","AutoMobile","Motorbike","Pubic_Transportation"]
    sel_mtrans = tk.StringVar()
    sel_mtrans.set(option_mtrans[0])
    option_menu = tk.OptionMenu(root,sel_mtrans,*option_mtrans)
    option_menu.grid(row=14, column=1)
    #-------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=15,column=1)

    submit_button = tk.Button(root, text='Submit', command=get_input)
    submit_button.grid(row=16, column=0, columnspan=2)
    #-------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=17,column=1)
    label1 =
tk.Label(root,text="",font=("Arial",15,"bold"),bg="lightblue",foreground="black")
    label1.grid(row=18,column=1)
    root.mainloop()
```

```
def run_image_detector():
    global label2
    root = tk.Toplevel(win)
    root.transient(win)
    #root.wm_attributes("-topmost",1)
    root.geometry("275x200")
    root.title("Obesity Detector")
    root.minsize(275,200)
    root.maxsize(275,200)
    root.configure(bg="lightblue")
    # ------------------------------------------------------------------------
    label = tk.Label(root, text="Obesity Detector", font=("Arial", 20, "bold"),
bg="lightblue", foreground="black")
    label.grid(row=0, column=1)
    # ------------------------------------------------------------------------
    label = tk.Label(root, text="      ", font=("Arial", 20, "bold"), bg="lightblue",
foreground="black")
    label.grid(row=1, column=1)
    # ------------------------------------------------------------------------
    btn = tk.Button(root, text='Select Image', command=open_file)
    btn.grid(row=2,column=1)
    #------------------------------------------------------------------------
    label = tk.Label(root,text="
",font=("Arial",20,"bold"),bg="lightblue",foreground="black")
    label.grid(row=3,column=1)
    label2 =
tk.Label(root,text="",font=("Arial",15,"bold"),bg="lightblue",foreground="black")
    label2.grid(row=4,column=1)
    root.mainloop()
```

```
def open_file():
    file = filedialog.askopenfile(mode='r',filetypes = [('JPEG files', '*.jpg *.jpeg'),('PNG files', '*.png')])
    print("file accessed")
    return predict(file)


def predict(file):
    prediction = list(image.predict(file.name, confidence=40, overlap=30))
    if prediction[0]['class'] == '1':
        prediction = "Not Obese"
    else:
        prediction = "Obese"


    label2.config(text="Prediction :-" + prediction)
#-----------------------------------------------------------------------------
win = tk.Tk()
win.geometry("350x300")
win.title("Obesity Detector")
win.minsize(350, 250)
win.maxsize(350, 250)
win.configure(bg="lightblue")
# -----------------------------------------------------------------------------
label = tk.Label(win, text="Obesity Detector", font=("Arial", 20, "bold"), bg="lightblue", foreground="black")
label.grid(row=0, column=0)
# -----------------------------------------------------------------------------
label = tk.Label(win, text="       ", font=("Arial", 20, "bold"), bg="lightblue", foreground="black")
label.grid(row=1, column=1)
```

```
# ------------------------------------------------------------------------------------
choice_label = tk.Label(win, text='Choose the detector:', font=("Arial", 10, "bold"),
bg="lightblue", foreground="black")
choice_label.grid(row=2, column=0)
option_choice = ["Select", "Text Detector", "Image Detector"]
sel_choice = tk.StringVar()
sel_choice.set(option_choice[0])
option_menu = tk.OptionMenu(win, sel_choice, *option_choice)
option_menu.grid(row=2, column=1)
# ------------------------------------------------------------------------------------
label = tk.Label(win, text="       ", font=("Arial", 20, "bold"), bg="lightblue",
foreground="black")
label.grid(row=3, column=1)
# ------------------------------------------------------------------------------------
submit_button = tk.Button(win, text='Submit',command=take_input)
submit_button.grid(row=4, column=0, columnspan=2)

win.mainloop()
# ------------------------------------------------------------------------------------
```

b) TESTING

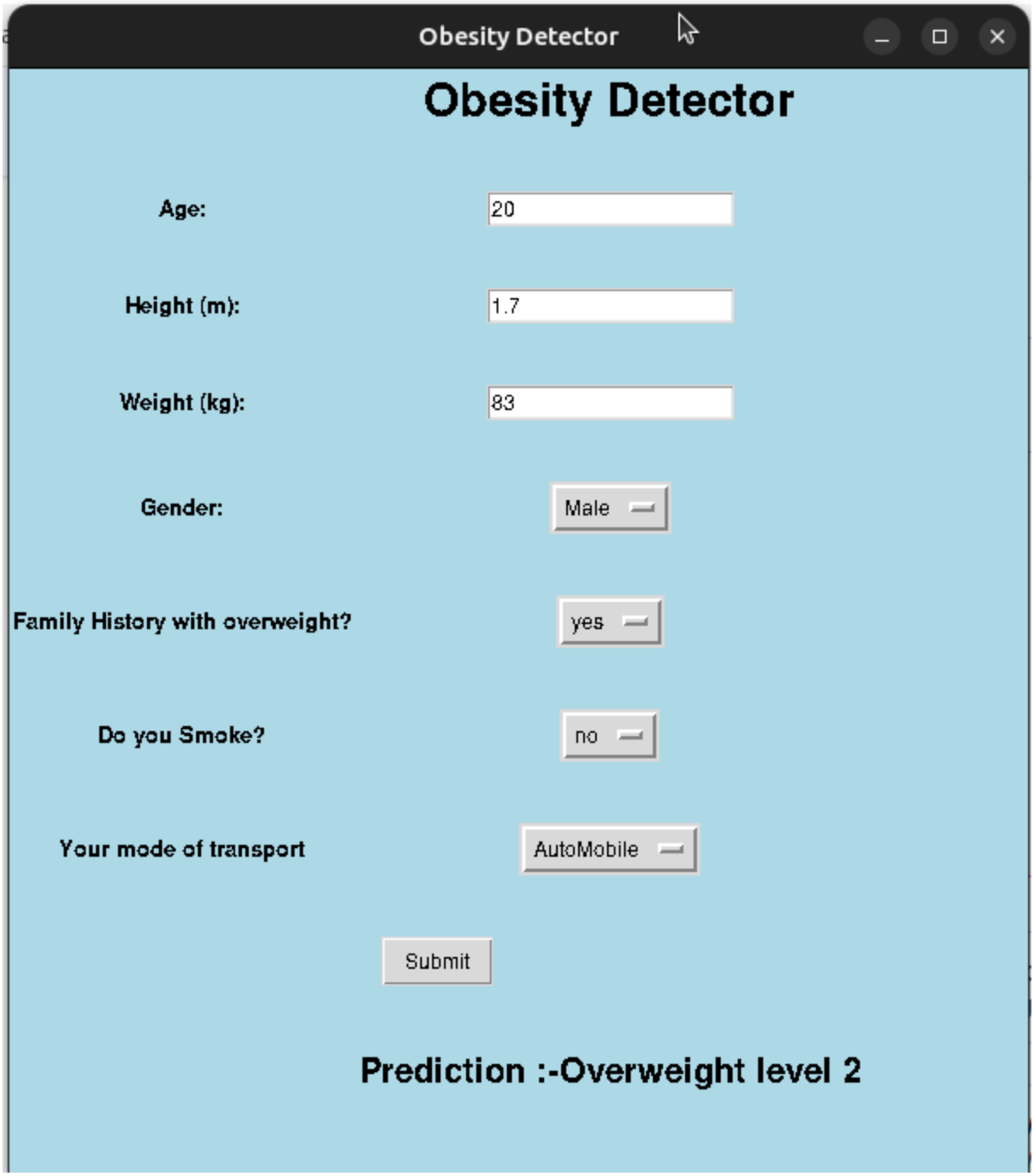

Figure 5: Obesity Detection Using Vitals

Model can predict from vitals that the person has obesity or not and the type of obesity or overweight levels. From the above image the model is able to predict with a 97% accuracy.

Figure 6: Obesity Detection Using Image

Model can predict from image that the person has obesity or not. From the above image the model is able to predict obesity with 90% accuracy and not obese with a 85% accuracy.

CHAPTER 5: RESULTS AND DISCUSSION

a) TEST REPORTS

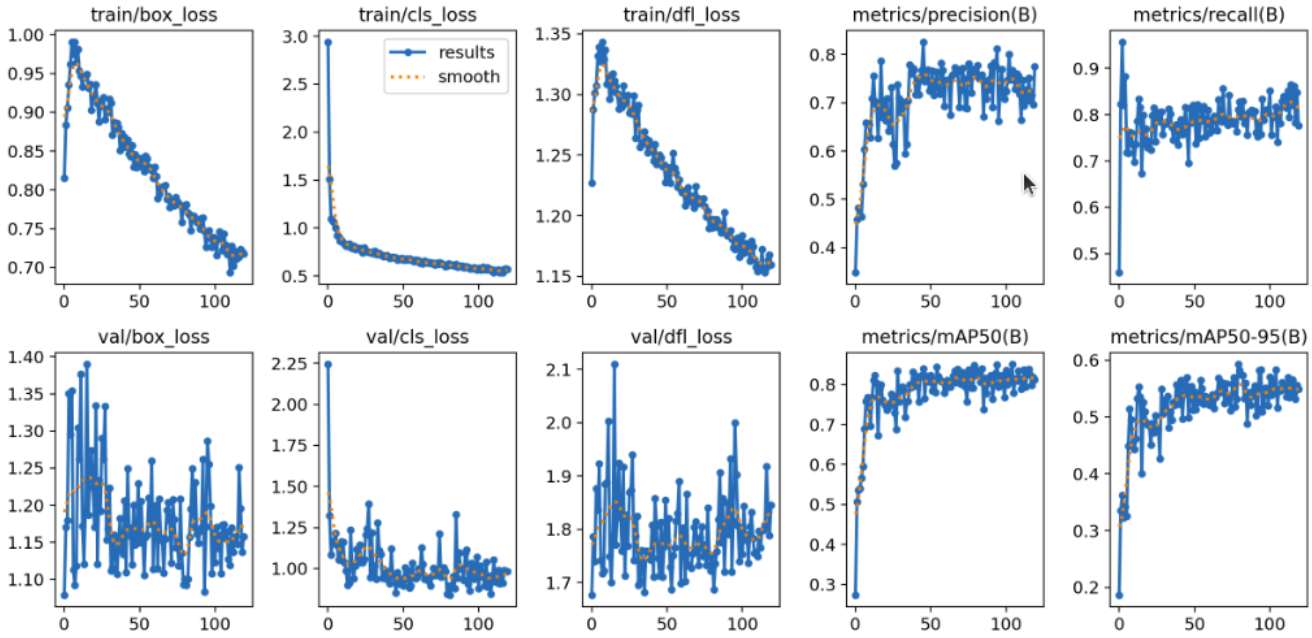

Figure 7: Results (train and metrics)

b) Models Metrics

ML MODELS	accuracy	precision	recall	f1
SVM with radial basis kernel	77.3	76.7	77.3	76.8
SVM with linear kernel	96.9	96.9	96.7	96.8
eXtreme Gradient Boosting	97.6	97.6	97.55	97.57
Adaboost with Decision Tree	93.1	92.9	92.8	92.8
Adaboost with Random Forest	96.2	96.4	96.1	96.1
Random Forest Classifier	95.5	95.6	95.3	95.3
Gradient Boosting	95.9	95.8	95.83	95.83
Adaboost with Extra Tree classifier	93.6	93.6	93.3	93.4
Adaboost with SVM	82.2	82.5	82.09	82.1
Category Boosting (CatBoost)	97.8	97.9	97.8	97.8
Light Gradient Boosting Method	99.05	99.04	99.09	99.06

Table 1: Model Metrics

CHAPTER 6: CONCLUSIONS

a) LIMITATIONS OF THE SYSTEM

1.Data Availability and Quality:

Dependence on Data: These models require large amounts of labeled data for training. Availability and quality of training data can be a significant limitation, especially for rare or specific objects.

Labeling Effort: Annotating data for object detection can be time-consuming and costly.

2.Computational Resources:

Hardware Requirements: Training deep learning-based object detection models often requires powerful GPUs or TPUs, making it resource-intensive.

Training Time: Training large models can take a long time, limiting rapid experimentation.

3.Overfitting:

Models can overfit the training data, leading to poor generalization to new, unseen data.

4.Imbalanced Data:

Imbalances in the distribution of object classes can lead to poor detection performance for minority classes.

5.Scale and Orientation Variability:

Models may struggle with objects at different scales or orientations, especially if the training data lacks diversity in these aspects.

6.Occlusions and Clutter:

Occluded or partially visible objects can be challenging for detection models, especially if they haven't seen many examples of such cases during training.

7.Real-time Processing:

Achieving real-time processing speed for object detection, especially in resource-constrained environments, can be difficult.

8.Adversarial Attacks:

Object detection models can be vulnerable to adversarial attacks, where minor perturbations in input data cause them to misclassify objects.

9.Interpretability:

Understanding why a model made a particular detection or misclassification can be challenging, especially in deep learning models.

10.Limited Context:

Object detection models often operate on individual images or frames, lacking contextual understanding across multiple frames or scenes.

b) FUTURE SCOPE OF THE PROJECT

The future scope of machine learning models for object detection is promising and continues to evolve rapidly. Here are some key areas where we can expect significant advancements and applications in the coming years:

1.Improved Accuracy:

Continued research in deep learning architectures, such as more advanced CNNs and attention mechanisms, will likely lead to higher accuracy in object detection.

2.Real-time Detection:

Optimizations and advancements in hardware, including more efficient GPUs, TPUs, and dedicated AI accelerators, will enable real-time object detection in a wider range of applications.

3.Multi-modal Object Detection:

Combining information from multiple sources, such as images, videos, LiDAR, and radar, will enhance the capabilities of object detection systems, especially in autonomous vehicles and robotics.

4.Few-shot and Zero-shot Learning:

Developing models that can recognize and detect objects with very few or even zero examples will be a significant breakthrough, reducing the need for extensive labeled data.

5.Adversarial Robustness:

Research in adversarial robustness will continue to address security concerns by making object detection models more resilient to adversarial attacks.

c) REFERENCES

[1] & Sánchez Hernández, A. B. (2019). Obesity level estimation software based on decision trees.

[2] Cervantes, R. C., & Palacio, U. M. (2020). Estimation of obesity levels based on computational intelligence. Informatics in Medicine Unlocked, 21, 100472.

[3] Abdullah, F. S., Manan, N. S. A., Ahmad, A., Wafa, S. W., Shahril, M. R., Zulaily, N., ... & Ahmed, A. (2016, August). Data mining techniques for classification of childhood obesity among year 6 school children. In International Conference on Soft Computing and Data Mining (pp. 465-474). Springer, Cham.

[4] Ferdowsy, F., Rahi, K. S. A., Jabiullah, M. I., & Habib, M. T. (2021). A machine learning approach for obesity risk prediction. Current Research in Behavioral Sciences, 2, 100053.

[5] Gupta, M., Phan, T. L. T., Bunnell, T., & Beheshti, R. (2019). Obesity Prediction with EHR Data: A deep learning approach with interpretable elements. arXiv preprint arXiv:1912.02655.

[6] Dugan, T. M., Mukhopadhyay, S., Carroll, A., & Downs, S. (2015). Machine learning techniques for prediction of early childhood obesity. Applied clinical informatics, 6(03), 506-520.

[7] Thamrin, S. A., Arsyad, D. S., Kuswanto, H., Lawi, A., & Nasir, S. (2021). Predicting Obesity in Adults Using Machine Learning Techniques: An Analysis of Indonesian Basic Health Research 2018. Frontiers in nutrition, 8.

[8] Pang, X., Forrest, C. B., Le-Scherban, F., & Masino, A. J. (2021). Prediction of early childhood obesity with machine learning and electronic health record data. International Journal of Medical Informatics, 150, 104454.

[9] Dunstan, J., Aguirre, M., Bastías, M., Nau, C., Glass, T. A., & Tobar, F. (2020). Predicting nationwide obesity from food sales using machine learning. Health informatics journal, 26(1), 652-663.

[10] Jindal, K., Baliyan, N., & Rana, P. S. (2018). Obesity prediction using ensemble machine learning approaches. In Recent Findings in Intelligent Computing Techniques (pp. 355-362). Springer, Singapore.