### An

**Internship Assessment Report**

On

### Early Detection of Dysgraphia using YOLOv8

At

### C# Corner

**Report submitted in partial fulfillment of the requirement for award of Bachelor of Technology**



### Name of student: Victor Shah Department Supervisor Name: Roll No.: 2001430100232 Dr. Prabhat Kr Srivastava

**Section:** CSE4 **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 **AKTU Course Code**: KCS752

**Semester/Year**: VII/ 4th **NBA Code**: C406

**Course Coordinator**: Basudeo Singh Roohani

### 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**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Content** | **Page No** |
| 1. | Company Letterhead | 2 |
| 2. | Declaration | 3 |
| 3. | Acknowledgement | 4 |
| 4. | List of Figures | 5 |
| 5. | List of Tables | 6 |
| 6. | Introduction to the Industry/Institution | 7 |
| 7. | **Company Profile:**   1. Company name with full address 2. Establishment Year, Owner Name, Annual Turnover etc. 3. Name of business of the Industry/Institute (Training Institute/Software Company etc.) 4. Add organizational Chart | 8 |
| 8. | Activities of the Company | 9 |
| 9. | Summer internship at a Glance (brief introduction) | 10 |
| 10. | Major Customers/Competitors | 11 |
| 11. | **CHAPTER 1: INTRODUCTION**   1. Background 2. Objectives 3. Scope of Project 4. Conclusion | 12 |
| 12. | **CHAPTER 2: METHODOLOGY**   1. Problem Definition 2. Literature Survey 3. Tools and Technology used | 15 |
| 13. | **CHAPTER 3: SYSTEM DESIGN**   1. Basic Modules 2. Data Flow Diagram 3. User interface design 4. Security Issues | 19 |
| 14. | **CHAPTER 4: IMPLEMENTATION AND TESTING**   1. Coding 2. Testing | 24 |
| 15. | **CHAPTER 5: RESULTS AND DISCUSSION**  a) Test Reports | 29 |
| 16. | **CHAPTER 6: CONCLUSIONS**   1. Limitations of the System 2. Future Scope of the Project 3. References | 30 |

#### Company Letterhead

[Date]

**Certificate of Completion**

This is to certify that **Victor Shah**, Roll No. 2001430100232 Bonafede 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, Ashok Kumar has worked on a project “**Early Detection of Dysgraphia using YOLOv8**” demonstrating exceptional enthusiasm, professionalism, and a strong work ethic.

We believe that **Victor Shah** 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 **Victor Shah** 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” Early Detection of Dysgraphia using YOLOv8” 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: Victor Shah

#### Roll No*:* 2001430100232

**Section:** CSE4

**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: Victor Shah Roll No: 2001430100232

**Section:** CSE4

#### LIST OF FIGURES

#### Data Flow Diagram

#### Dashboard for User

#### Handwriting Detection

#### Results (Train and Metrics)

#### LIST OF TABLES

#### Epochs Training

#### Introduction to the Industry

#### COMPANY PROFILE

#### ACTIVITIES OF THE COMPANY

#### SUMMER INTERNSHIP AT A GLANCE

#### MAJOR CUSTOMER’S/COMPETITORS

#### CHAPTER 1: INTRODUCTION

#### BACKGROUND

#### The issue of dysgraphia, a neurological condition characterized by difficulties in writing, drawing, or other fine motor skills, has been recognized for many years. However, advancements in technology, particularly in machine learning and artificial intelligence, have opened up new possibilities for early detection and intervention. Dysgraphia has likely been present throughout human history, but its formal recognition and understanding as a distinct learning disability began to emerge in the mid-20th century. Early assessments were largely subjective and relied on the observations of educators and healthcare professionals. Diagnosis typically occurred after a child had already encountered significant academic challenges. Despite progress in understanding dysgraphia, early detection remains a significant challenge. Diagnosis often occurs late, leading to academic and emotional struggles for affected individuals. Current assessment methods involve subjective evaluations, handwriting samples, and standardized tests. These methods are limited by their subjectivity, reliance on observable symptoms, and the fact that dysgraphia can vary widely in its presentation. Machine learning algorithms can detect subtle patterns and indicators of dysgraphia far earlier than traditional methods. This early detection allows for timely interventions and support, reducing the impact of dysgraphia on a child's education.

#### A project focused on early detection of dysgraphia using machine learning offers an ideal solution by addressing the limitations of current assessment methods. It enables: Timely Identification: Dysgraphia can be identified early in a child's academic journey, facilitating early intervention and support. Objective and Accurate Assessment: Machine learning models can provide objective and accurate assessments of dysgraphia, reducing the risk of misdiagnosis or delayed diagnosis. Tailored Interventions: Personalized interventions can be designed based on the specific needs and challenges of everyone, leading to more effective support.

#### OBJECTIVES

#### Develop a Machine Learning Model: Train a machine learning model capable of accurately detecting dysgraphia in individuals based on various input data sources (handwriting samples, keyboard inputs, eye-tracking, etc.).

#### Early Detection Capability: Achieve a high level of accuracy in identifying dysgraphia at an early stage, ideally before significant academic or emotional challenges arise.

#### Objective Assessment: Ensure that the machine learning model provides objective and consistent assessments, reducing the subjectivity associated with traditional diagnostic methods.

#### Scalability: Design the model and system to be scalable, capable of handling a large volume of assessments efficiently, making it applicable in educational and healthcare settings.

#### Integration with Educational Institution: Ensure that the system can seamlessly integrate with existing educational platforms and tools used in schools to facilitate the early identification of dysgraphia.

#### Feedback Mechanism: Establish a feedback mechanism to gather input from users, including educators, parents, and individuals with dysgraphia, for continuous improvement of the system.

#### SCOPE OF PROJECT

#### This project encompasses the development of a robust and scalable machine learning system. This system will analyze diverse datasets comprising handwriting samples, keyboard inputs, and eye-tracking data to accurately identify dysgraphia at an early stage, enabling timely interventions. The project includes the creation of a user-friendly interface for data input and results visualization, as well as the development of algorithms for personalized intervention recommendations based on individual dysgraphia profiles. Ethical considerations regarding data privacy and security will be addressed, and the system's accessibility and inclusivity for diverse users will be ensured. Extensive research and validation studies will be conducted to assess the model's effectiveness, and educational materials and outreach programs will be designed to raise awareness. The project will also evaluate cost-effectiveness and incorporate continuous monitoring and feedback mechanisms to facilitate ongoing improvement and user engagement.

#### CONCLUSION

#### In conclusion, a project focused on early detection of dysgraphia using machine learning has the potential to significantly improve the lives of individuals with dysgraphia by providing early and accurate assessments, personalized interventions, and scalable solutions. This technology-driven approach aligns with the evolving landscape of education and healthcare, where data-driven insights and early interventions are increasingly valuable.

#### CHAPTER 2: METHODOLOGY

#### PROBLEM DEFINITION

#### Handwriting is a complex ability to get, and it demands lot of preparation and effort to be learnt. Children having dysgraphia exhibit trouble writing. This can cause tension and can otherwise impact instruction. It requires the close connection and arrangement of psychomotor and biomechanical processes. Dysgraphia is a difficulty in writing that straightforwardly impacts a student's talent to perform capably. Although dysgraphia is exactly elucidated as “distressing novel”, it too influences a person’s capability to envision and draw lines and shapes. There may lie several students that would benefit from proper understanding of this disorder by educators. This record of what happened is created to comprehend the condition of dysgraphia.

#### LITERATURE SURVEY

#### In accordance with the paper Labelling Developmental Dysgraphia Traits Applying Script Classification Designs (2017), Dysgraphia is a commotion or trouble because of written language having connection with the mechanical details of the letter. The difficulties manifest as incompetent acting of longhand among youth the one argues least an average intellect level and the one destitute existed identified as bearing some understandable neurological or concerning feelings and intuition-engine questions. The prevalence of scrawl troubles with elementary school scholars ranges from 10% to 34%. Dysgraphia can have deep associations for the individual in conditions of compromised self-figure and accomplishment in school. The aim of this study was to cultivate and test a mathematical model for changing between dysgraphia and able calligraphy to establish their performance traits. This study attracted labelling and typifying dysgraphia among Israelite literature teenagers. The SVM classifier shows accurate prophecies for 89 exhausted 99 calligraphy products, which leads to a veracity of 89.9%.

#### In accordance with the paper Dysgraphia detection through machine intelligence (2020), a new script dataset was composed that consisted of various writing activities and a wide range of faces were extracted to get various facets of scrawl. Those were given to a machine intelligence treasure to envision if the manuscript was affected by dysgraphia.

#### Before they distinguished several machine intelligence algorithms and erect high-quality results were achieved apiece adjusting pushing (Adaptive Boosting) algorithm. Finally, it proved that machine intelligence may be used to identify dysgraphia accompanying nearly eighty percent accuracy, while handling a heterogeneous set of matters. Cases with some harm or physical dislike to draft were expelled. Speed, jerk, acceleration, pressure, azimuth, and peak were culled in form of heading of calligraphy sample. Classifier confirmation was conducted utilizing layered having ten of something cross validation, and all processes were recurring for ten periods. Categorization accuracy, feeling, and particularity over the ten duplications were averaged. Results showed it is likely to select youngsters with disorder with accuracy of 79.5% on a sample of kids of various ages using the AdaptiveBoosting and utilizing the RF and SVM algorithms. The accuracy score of the added models were high, 72.3% for RF classifier and 72.5% for SVM.

#### In accordance with paper, TestGraphia, a Spreadsheet Whole for the Early Diagnosis of Dysgraphia (2020), this study plans an judgment agreement for BHK test by including writing face to a degree book size, impartial abandoned border, skewed lines, lacking scope 'tween words, sharp angles, defective links middle from two points reports, collision 'between two postcards, uneven height of letters, contradictory crest 'tween letters accompanying and outside enlargement, atypical messages, uncertain notes, traced memos, doubtful path. These facial characteristics should be deliberate by doctors and forms to pronounce dysgraphia. Certain countenance is approximately arithmetic based, while additional physiognomy demands a doctor’s interpretation and few visages may be automated. In accordance with the paper, it uses a reduced version of loop affecting animate nerve organs networks accompanying keras using tensor flow backend, a Scanning in of documents model was erected utilizing CNN. Accuracy of 86.14% is computed utilizing veracity principles of 85.12% and 87.18% from the data and the experiments. 5-Fold cross validation approach was second-hand. In every trial, the veracity profit was calculated every round and last veracity utilizing the mean principles of all rounds were calculated. Networks (2019), an akin approach was second-hand using CNN in addition to keras and TensorFlow outside Scanning in of documents which managed to a veracity of 55.7%. So, scanning in of documents using CNN is favorable.

#### In accordance with the paper, Concept Classification utilizing SVM and CNN (2020), SVM model that second-hand a very narrow dataset realized an accuracy of 93% and even though SVM is a very powerful method, achieving an extreme veracity was still an

#### anomaly. Utilizing dossier improving, the size of the dataset was in addition to increase and was acted on SVM repeatedly, it achieved a veracity of 82%. On favorably executing CNN, it achieved a veracity of 93.57% on the unchanging dataset. So, it decides that using CNN over an abundant improved dataset of countenances is better than using SVM because it determines larger veracity.

#### 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.

#### R: R is another language commonly used for statistical analysis and data visualization in machine learning.

#### 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. Data Storage:

#### SQL and Relational Databases: For structured data storage and retrieval.

#### NoSQL Databases: Such as MongoDB or Cassandra for unstructured or semi-structured data storage.

#### Big Data Tools: Like Hadoop and Spark for handling large-scale datasets.

#### 6. Machine Learning Libraries for Specialized Tasks:

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

#### OpenCV: A library for computer vision tasks.

#### Gensim: For topic modeling and document similarity analysis.

#### 7. Cloud Platforms:

#### AWS (Amazon Web Services): Provides cloud-based infrastructure and services for machine learning.

#### Google Cloud Platform (GCP): Offers various machine learning tools, including AI Platform and TensorFlow.

#### Azure: Microsoft's cloud platform with machine learning services like Azure Machine Learning.

#### 8. 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

#### 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 (Optional):

#### 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.

#### 19.Conclusion and Future Directions:

#### Module Description: Summarize the overall system design, its components, and mention potential future improvements or enhancements.

#### DATA FLOW DIAGRAM

#### 

#### Figure 1: Data Flow Diagram

#### USER INTERFACE DESIGN

#### A screenshot of a webcam Description automatically generated

#### Figure 2: Dashboard for User

#### 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

#### CODING

#### import numpy as np

#### import argparse

#### import cv2 as cv

#### import subprocess

#### import time

#### import os

#### from yolo\_utils import infer\_image, show\_image

#### FLAGS = []

#### if \_\_name\_\_ == '\_\_main\_\_':

#### parser = argparse.ArgumentParser()

#### parser.add\_argument('-m', '--model-path',

#### type=str,

#### default='./yolov3-coco/',

#### help='The directory where the model weights and \

#### configuration files are.')

#### parser.add\_argument('-w', '--weights',

#### type=str,

#### default='./yolov3-coco/yolov3.weights',

#### help='Path to the file which contains the weights \

#### for YOLOv3.')

#### parser.add\_argument('-cfg', '--config',

#### type=str,

#### default='./yolov3-coco/yolov3.cfg',

#### help='Path to the configuration file for the YOLOv3 model.')

#### parser.add\_argument('-i', '--image-path',

#### type=str,

#### help='The path to the image file')

#### parser.add\_argument('-v', '--video-path',

#### type=str,

#### help='The path to the video file')

#### parser.add\_argument('-vo', '--video-output-path',

#### type=str,

#### default='./output.avi',

#### help='The path of the output video file')

#### parser.add\_argument('-l', '--labels',

#### type=str,

#### default='./yolov3-coco/coco-labels',

#### help='Path to the file having the \

#### labels in a new-line seperated way.')

#### parser.add\_argument('-c', '--confidence',

#### type=float,

#### default=0.5,

#### help='The model will reject boundaries which has a \

#### probabiity less than the confidence value. \

#### default: 0.5')

#### parser.add\_argument('-th', '--threshold',

#### type=float,

#### default=0.3,

#### help='The threshold to use when applying the \

#### Non-Max Suppresion')

#### parser.add\_argument('--download-model',

#### type=bool,

#### default=False,

#### help='Set to True, if the model weights and configurations \

#### are not present on your local machine.')

#### parser.add\_argument('-t', '--show-time',

#### type=bool,

#### default=False,

#### help='Show the time taken to infer each image.')

#### FLAGS, unparsed = parser.parse\_known\_args()

#### # Download the YOLOv3 models if needed

#### if FLAGS.download\_model:

#### subprocess.call(['./yolov3-coco/get\_model.sh'])

#### # Get the labels

#### labels = open(FLAGS.labels).read().strip().split('\n')

#### # Intializing colors to represent each label uniquely

#### colors = np.random.randint(0, 255, size=(len(labels), 3), dtype='uint8')

#### # Load the weights and configutation to form the pretrained YOLOv3 model

#### net = cv.dnn.readNetFromDarknet(FLAGS.config, FLAGS.weights)

#### # Get the output layer names of the model

#### layer\_names = net.getLayerNames()

#### layer\_names = [layer\_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]

#### # If both image and video files are given then raise error

#### if FLAGS.image\_path is None and FLAGS.video\_path is None:

#### print ('Neither path to an image or path to video provided')

#### print ('Starting Inference on Webcam')

#### # Do inference with given image

#### if FLAGS.image\_path:

#### # Read the image

#### try:

#### img = cv.imread(FLAGS.image\_path)

#### height, width = img.shape[:2]

#### except:

#### raise 'Image cannot be loaded!\n\

#### Please check the path provided!'

#### finally:

#### img, \_, \_, \_, \_ = infer\_image(net, layer\_names, height, width, img, colors, labels, FLAGS)

#### show\_image(img)

#### elif FLAGS.video\_path:

#### # Read the video

#### try:

#### vid = cv.VideoCapture(FLAGS.video\_path)

#### height, width = None, None

#### writer = None

#### except:

#### raise 'Video cannot be loaded!\n\

#### Please check the path provided!'

#### finally:

#### while True:

#### grabbed, frame = vid.read()

#### # Checking if the complete video is read

#### if not grabbed:

#### break

#### if width is None or height is None:

#### height, width = frame.shape[:2]

#### frame, \_, \_, \_, \_ = infer\_image(net, layer\_names, height,\ width, frame, colors, labels, FLAGS)

#### if writer is None:

#### # Initialize the video writer

#### fourcc = cv.VideoWriter\_fourcc(\*"MJPG")

#### writer = cv.VideoWriter(FLAGS.video\_output\_path, fourcc, 30, \

#### (frame.shape[1], frame.shape[0]), True)

#### writer.write(frame)

#### print ("[INFO] Cleaning up...")

#### writer.release()

#### vid.release()

#### else:

#### # Infer real-time on webcam

#### count = 0

#### vid = cv.VideoCapture(0)

#### while True:

#### \_, frame = vid.read()

#### height, width = frame.shape[:2]

#### if count == 0:

#### frame, boxes, confidences, classids, idxs = infer\_image(net, layer\_names, \

#### height, width, frame, colors, labels, FLAGS)

#### count += 1

#### else:

#### frame, boxes, confidences, classids, idxs = infer\_image(net, layer\_names, \

#### height, width, frame, colors, labels, FLAGS, boxes, confidences, classids, idxs, infer=False)

#### count = (count + 1) % 6

#### cv.imshow('webcam', frame)

#### if cv.waitKey(1) & 0xFF == ord('q'):

#### break

#### vid.release()

#### cv.destroyAllWindows()

#### TESTING

#### 

#### Figure 3: Handwriting Detection

#### Model can categorize the handwriting into Good and Bad. From the above image the model is able to predict Bad handwriting with a 40+ accuracy and good handwriting with a 60+ accuracy.

#### CHAPTER 5: RESULTS AND DISCUSSION

#### TEST REPORTS

#### A screenshot of a graph Description automatically generated

#### Figure 4: Results (train and metrics)

#### Table 1: Epochs training

#### A table of numbers and symbols Description automatically generated

#### CHAPTER 6: CONCLUSIONS

#### 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.

#### 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.

#### REFERENCES

#### [1] G. Molina-Vargas, H. Arias-Flores, and J. Jadán-Guerrero, “Benefit of developing assistive technology for writing,” in Intelligent Human Systems Integration 2021, vol. 1322, Cham: Springer International Publishing, 2021, pp. 586–590, doi: 10.1007/978-3-030-68017-6\_86.

#### [2] C. Atanga, B. A. Jones, L. E. Krueger, and S. Lu, “Teachers of students with learning disabilities: Assistive technology knowledge, perceptions, interests, and barriers,” Journal of Special Education Technology, vol. 35, no. 4, pp. 236–248, Dec. 2020, doi: 10.1177/0162643419864858.

#### [3] F. Dcruz, V. Tiwari, and M. Soni, Using machine learning to help students with learning disabilities learn, vol. 39, Springer ISSN: 2088-8708 Int J Elec & Comp Eng, Vol. 13, No. 3, June 2023: 2999-3009 3008 International Publishing, 2020, doi: 10.1007/978-3-030-34515-0\_27.

#### [4] K. A. Al-Dababneh and E. K. Al-Zboon, “Using assistive technologies in the curriculum of children with specific learning disabilities served in inclusion settings: teachers’ beliefs and professionalism,” Disability and Rehabilitation: Assistive Technology, vol. 17, no. 1, pp. 23–33, Jan. 2022, doi: 10.1080/17483107.2020.1752824.

#### [5] M. Virnes, E. Sutinen, and E. Kärnä-Lin, “How children’s individual needs challenge the design of educational robotics,” in Proceedings of the 7th international conference on Interaction design and children - IDC ’08, 2008, pp. 274-281, doi: 10.1145/1463689.1463766.

#### [6] R. Nightingale, E. Sumner, M. Prunty, and A. L. Barnett, “Handwriting and typing: Occupational therapy practice when supporting adolescents with handwriting difficulties,” British Journal of Occupational Therapy, vol. 85, no. 11, pp. 891–899, Nov. 2022, doi: 10.1177/03080226221097314.

#### [7] A. Piller and E. Torrez, “Defining occupational therapy interventions for children with fine motor and handwriting difficulties,” Journal of Occupational Therapy, Schools, & Early Intervention, vol. 12, no. 2, pp. 210–224, Apr. 2019, doi: 10.1080/19411243.2019.1592053.

#### [8] D. Tamakloe, “Enhancing learning and development of young children with disabilities with assistive technology,” in International Perspectives on Inclusive Education, vol. 14, 2020, pp. 141–161, doi: 10.1108/S1479-363620200000014012.

#### [9] T. L. C. Asselborn, “Analysis and remediation of handwriting difficulties,” École Polytechnique Fédérale de Lausanne, 2020, doi: 10.5075/epfl-thesis-8062.

#### [10] K. Zolna et al., “The dynamics of handwriting improves the automated diagnosis of dysgraphia,” Prepr. arXiv1906.07576, Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.07576