DETECTION OF SLD IN KANNADA VOCABULARY AMONGST UNDERGRADUATE STUDENTS

PROJECT REPORT
Submitted by

Tushar B T (1BM20CS174)
Umang Goel (1BM20CS176)
Vinay Kulkarni (1BM20CS188)
Varun Urs M S (1BM20CS182)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of
Dr. Kavitha Sooda
Professor, Department of CSE, BMSCE

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project entitled “DETECTION OF SLD IN KANNADA VOCABULARY AMONGST UNDERGRADUATE STUDENTS” is a bonafide work carried out by Tushar B T (1BM20CS174), Umang Goel (1BM20CS176), Vinay Kulkarni (1BM20CS188), Varun Urs M S (1BM20CS182) in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2023-24. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Guide	**Head of Department**
Dr. Kavitha Sooda, Professor, Dept of CSE, B.M.S. College of Engineering	**Dr. JYOTHI S. NAYAK**, Professor and HOD, Dept of CSE, B.M.S. College of Engineering

Principal

Dr. S. Muralidhara
B.M.S. College of Engineering

External Viva

Name of the Examiners Signature with Date

1.

2.

Table of Contents

Abstract

This study introduces an innovative method for identifying Slow Learning Disability (SLD) in Kannada Vocabulary among undergraduate students, utilizing Convolutional Neural Networks (CNN) and Random Forest classifiers. By digitizing handwritten words from sheets, analyzing features such as writing pressure, bounding box dimensions, and character spacing, our approach achieves a 61.80% accuracy with CNN and a 68.09% accuracy with Random Forest. This method surpasses previous techniques, which were cumbersome for students, costly, required physical intervention and specific hardware. Our technique simplifies the process by scanning handwritten sheets, which are then processed to classify them as Normal or Abnormal. Additionally, it has been observed that students with low writing pressure showed lesser desire to learn, and students with higher writing pressure showed more desire to learn. Furthermore, it has been observed that students with larger handwriting were mostly classified as normal, and students with smaller handwriting were mostly classified as abnormal. This research offers a more user-friendly and affordable solution for SLD detection, potentially enhancing early support for at-risk students.

DECLARATION

We, hereby declare that the Major Project Phase-2 work entitled **"Detection of SLD in Kannada Vocabulary Amongst Undergraduate Students"** is a bonafide work and has been carried out by us under the guidance of Dr. Kavitha Sooda**,** Professor, Department of Computer Science and Engineering, B.M.S. College of Engineering, Bengaluru**,** in partial fulfilment of the requirements of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi.
I further declare that, to the best of my knowledge and belief, this project has not been submitted either in part or in full to any other university for the award of any degree.

Candidate details:

SL. NO.	Student Name	USN	Student's Signature
1	Tushar B T	1BM20CS174	
2	Umang Goel	1BM20CS176	
3	Vinay Kulkarni	1BM20CS188	
4	Varun Urs M S	1BM20CS182	

Place: Bengaluru

Date:

Certified that these candidates are students of Computer Science and Engineering Department of B.M.S. College of Engineering. They have carried out the project work titled "Detection of SLD in Kannada Vocabulary amongst Undergraduate Students" as Major Project Phase-2 work. It is in partial fulfilment for completing the requirement for the award of B.E. degree by VTU. The works are original and duly certify the same.

Guide Name Signature

Dr. Kavitha Sooda

Date:

Acknowledgment

We extend our heartfelt gratitude to all those who supported us on this journey. We are particularly grateful to the administration of BMSCE for providing essential resources and a supportive environment, and to Dr. Jyothi S. Nayak, Head of Department of Computer Science and Engineering, for generously providing both resources and the necessary framework to develop this project.

We are deeply indebted to our guide, Dr. Kavitha Sooda for her expert guidance, unwavering support, and insightful suggestions. Her constant motivation and mentorship were instrumental in shaping this project.

We also express our deepest gratitude to Mrs. Gayathri, Principal of Jaigopal Garodia Rashtrotthana Vidyalaya, Bengaluru, for allowing us to conduct our research and gather data from her students.

We extend our sincere appreciation to graphologists Dr. Noothan Rao and Dr. Shivanand Nayak for generously dedicating their time and sharing invaluable insights into the handwriting aspect from a graphologist's perspective.

Finally, we offer our profound thanks to our families and friends for their unwavering support and encouragement. Their understanding and belief in us helped us stay motivated throughout this process.

We are deeply grateful to each and every individual who contributed to the successful completion of this project. We are confident that our research will contribute significantly to the field of early dysgraphia detection and pave the way for further advancements in this critical area.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Even with technology getting better, writing by hand is still really important. Talking with others, and showing how you feel. But, some people have a hard time writing neatly and easily due to slow learning disabilities. This makes it tough for them to communicate, express themselves, and capture their ideas. This problem affects 12-33% of kids in school [1]. It can make it hard for them to do well in school, feel good about themselves. It's really important to find out about this problem and get help.

1.2 Motivation

Witnessing the struggles faced by young children with slow learning disabilities, particularly within the Kannada-speaking community, inspired us to undertake this research. Existing solutions are often time-consuming, expensive, and lack specific support for diverse languages and cultural backgrounds. By developing a novel, cost-effective, and user-friendly technology, we aim to make detection of slow learning disabilities more accessible and effective for all. Our hope is to remove barriers to learning.

1.3 Objective

This initiative addresses the lack of existing solutions specifically tailored to the Kannada language and leverages machine learning techniques to improve the accuracy and efficiency of slow learning disability assessments.

Key objectives include:

Developing an automated system: This system will utilize machine learning algorithms to analyze handwriting samples and identify potential slow learning disability symptoms in Kannada-speaking children.

Handwriting feature identification: Key handwriting features essential for identifying slow learning disability are size, pressure and letter characteristics, will be identified and analyzed.

1.4 Scope

A Machine learning model which is trained on a dataset collected from undergraduate students, capable of assessing the probability of a given person having a slow learning disability based on the features detected in their handwriting.

To complete the project within a timeframe of nine months, including four months for research and algorithm development, two months for system implementation and testing, and three months for analysis, optimization, and documentation.

1.5 Existing System

Despite the increasing prevalence of slow learning disability, particularly in developing countries like India, existing systems for its detection face significant limitations. Notably, most research and development efforts have focused on Western languages, neglecting languages like Kannada.

Furthermore, existing systems often rely on expensive technologies like iPads [3] and digital tablets [4], [5] with styluses for handwriting analysis. This creates accessibility barriers for resource-constrained families and limits the scalability of such solutions. Additionally, these systems often use proprietary apps that are not open-source or publicly accessible, raising concerns about transparency and applicability.

While some researchers have explored camera-based methods for detection, these approaches have not yet achieved the high accuracy levels which are often due to limitations in camera resolution, lighting conditions, and the ability to capture subtle handwriting details necessary for accurate analysis. making them unsuitable for classification.

1.6 Proposed System

To address the limitations of existing slow learning disability detection systems, we've built a novel approach specifically for Kannada handwriting. Unlike systems reliant on expensive tablets, our system will be accessible and affordable. We've also acquired Kannada-specific data from undergraduate students of BMSCE. These students wrote 21 kannada words on an A4 sheet divided into grid boxes for consistent data capture.

After collecting the dataset or the assessment sheet from the student, the sheet must immediately undergo a pressure test. In this pressure test we feel the back of the handwritten words and check the pressure put on by the person writing the words. By assessing the pressure we put it into one of the three categories,which is High pressure, Medium pressure and Low pressure. After classifying the assessment based on the pressure, the assessment sheet is scanned for the next step in the assessment process. The scanned sheet is fed into an opencv function which will crop out all the 21 words written by the person. All the 21 words were chosen such that they cover all the words in kannada language.

For analysis, we propose a CNN and random forest model. This model, trained on our pre-processed data, will predict a score for each writing sample. We'll then categorise these scores into two levels: Normal, and Abnormal. This classification system provides a comprehensive assessment of handwriting fluency and legibility, making slow learning disability identification easier and more reliable.

Chapter 2

Literature Survey

Dysgraphia is a handwriting disorder and early detection of dysgraphia helps to treat the patient effectively. Few researches involve in the classification of dysgraphia using the devices like digitized writing tablets or iPads along with specialized styluses and proprietary apps to capture the handwriting. While these methods offer advantages, they fail to address the needs of Kannadiga communities.

In a study aiming to aid dysgraphia diagnosis, Dimauro et al. [6] developed TestGraphia software. This software utilizes various document and feature selection algorithms to analyze children's handwriting (grades 2-5) and identify potential dysgraphia symptoms. Their experiments employed nine algorithms to extract and analyze text features, demonstrating improved dysgraphia detection accuracy and faster processing times compared to other methods. However, a few limitations were identified: borderline cases remained ambiguous due to inherent handwriting variances, and the impact of non-discriminatory letters (those not providing significant diagnostic information) was not addressed.

Yogarajah et al. [7] applied a Convolutional Neural Network (CNN) to automate the process of detecting dysgraphia as well as dyslexia amongst Hindi writing students. The model was trained on a dataset consisting of 164 handwritten hindi words from students with ‘Strong Evidence of Risk’ on DST-J from classes 1 to 5 and 103 from normal students. This method has incorporated Optical Character Recognition (OCR) on manually cropped images which still requires further processing such as skewing and cursive methods since OCR is not found to be stable. This study has reported an average accuracy of $(85.12 \pm 2.0)\%$. The lack of a large dataset as well as lack of phonic aspects hinder the study.

Ghouse, Paranjothi, and Vaithiyanathan [8] proposed a novel method called Non-Discrimination Regularization in the Rotational Region Convolutional Neural Network (NDR-R2CNN). This model introduces balancing parameters in the loss function to achieve balanced class representation during training and eliminate irrelevant features to mitigate overfitting. This approach enables effective feature analysis and non-discriminatory word analysis. Their results demonstrate that NDR-R2CNN achieves an impressive accuracy of 98.2%, significantly outperforming the existing CNN model's 94.2% accuracy. However, one limitation noted is the lack of an attention layer, which could potentially further improve classification accuracy.

Kumoth et al. [9] explored various methods for dyslexia detection using deep convolutional neural networks (CNNs) and transfer learning. They employed several techniques, including fine-tuning pre-trained models like DenseNet201, feature extraction from CNNs, and ensemble learning with soft and hard voting strategies. These models were trained on separate datasets of word, pseudoword, difficult word, and sentence images. To evaluate their performance, the authors compared the CNNs to three traditional machine learning algorithms: SVM, AdaBoost, and Random Forest. Ultimately, the SVM model trained on word data achieved the highest accuracy of 91.7%.

Mor et al. [10] adopted a transfer learning approach by utilising the pre-trained MobileNetV2 architecture for dyslexia detection. This choice was motivated by MobileNetV2's suitability for visual tasks, which aligned well with the study's needs. To adapt the model for dyslexia classification, the authors removed the final SoftMax layer designed for ImageNet's 1,000 classes and added three hidden layers of ReLU neurons. While their approach achieved promising results, it had limitations. Firstly, the model's generalizability to diverse populations remains unproven. Additionally, the training dataset was relatively small, potentially impacting the model's robustness. Looking ahead, Mor et al. suggest exploring the assessment of handwriting across multiple languages from different populations to address these limitations and enhance the model's applicability.

While several studies like [11 - 15] have shown promise in dysgraphia detection using technology, their reliance on iPads, digitised tablets, and specialised styluses raises significant barriers to widespread adoption. These expensive devices, often paired with proprietary software, create an economic hurdle, particularly in developing countries or low-income communities. Dimauro et al.'s TestGraphia software, for instance, requires specific digitizers and proprietary apps, limiting its reach. Similarly, Ghouse et al.'s NDR-R2CNN model, while demonstrating high accuracy, relies on costly specialised hardware for data capture. This dependence on expensive technology not only restricts accessibility but also raises concerns about data privacy and software licensing costs. Furthermore, our use of open-source tools and methodologies ensures transparency and fosters further research and development tailored to specific languages like Kannada.

Chapter 3

Requirement Analysis and Specifications

3.1 Functional requirements

Inputs : The inputs must be Kannada words only.

Feature extraction: The system should extract relevant features from the handwriting samples, such as size, and letter characteristics.

Prediction: The system should be able to predict the likelihood of having slow learning disability based on the extracted features from the handwriting samples.

Classification: The system should be able to classify the student as Normal or Abnormal.

Reliability : The system should be able to handle all the types of inputs its model has been trained on.

Data management: The system should be able to store and manage large datasets of handwriting samples and associated annotations.

Security: The system should ensure the security and privacy of the person without disclosing the person's identity.

3.2 Non-Functional Requirements

Accuracy: The system should be able to accurately produce the probability of having slow learning disability with high sensitivity and specificity.

Efficiency: The system should be able to analyze handwriting samples and generate predictions quickly.

Scalability: The system should be able to handle large datasets and be scalable to accommodate future growth.

Robustness: The system should be robust to noise and variations in handwriting style.

Portability: The system should be portable and run on various operating systems.

Usability: The system should be easy to use and learn for users with varying levels of technical expertise.

Cost-effectiveness: The system should be cost-effective to develop and implement.

3.3 Hardware Requirements

- Minimum processor: Intel Core i5 or equivalent.
- Minimum RAM: 8GB.
- Minimum storage: 1TB.
- Graphics card (optional): NVIDIA GeForce GTX 1060 or equivalent for faster training with larger datasets.
- Flatbed scanner: Minimum resolution of 120 dpi for capturing high-quality handwriting samples.

3.4 Software Requirements

- Operating system : Ubuntu 20.04 LTS (Linux) or Windows 10 or MacOS.
- Python 3.6
- Google Colab to visualise and train the model on the dataset collected.
- Visual Studio
- MatLab
- TensorFlow or PyTorch: Deep learning frameworks for building and training the model.
- Keras (recommended): for building and training neural networks with TensorFlow or PyTorch.
- OpenCV: for image processing tasks.
- NumPy and Pandas: for scientific computing and data analysis.

Chapter 4

Design

4.1 High level design

Figure 4.1 briefly mentions the high-level design and methodology for an automated dysgraphia detection system in primary and middle school children. It utilizes various image processing techniques and feature extraction methods to analyze handwritten Kannada characters and identify potential probability of having slow learning disability. The detailed explanation of each process is explained in Chapter 4.2.

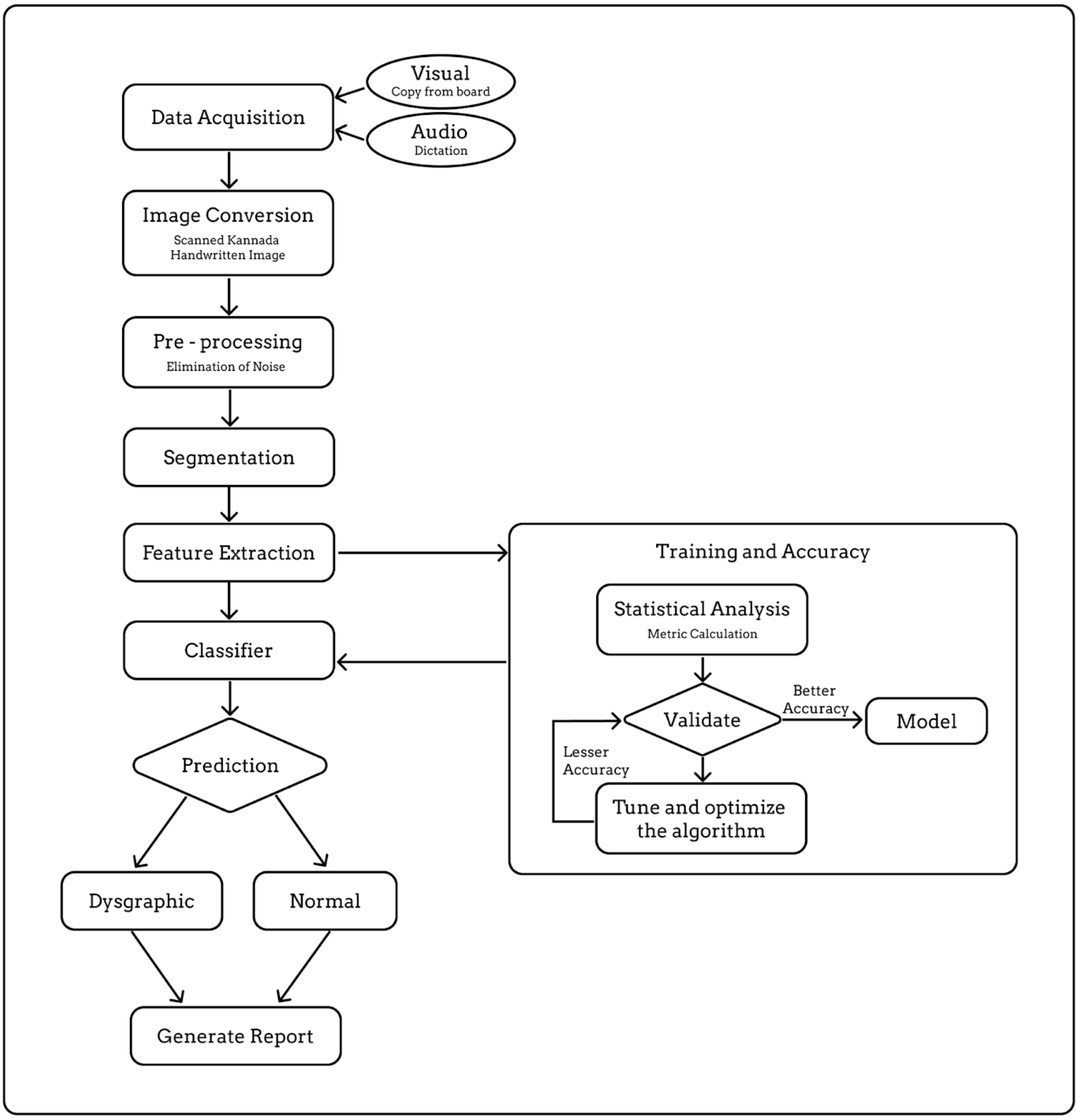


Figure 4.1 - High level design

4.2 System Architecture

Each sub-system within the architecture performs a specific task with defined inputs and outputs. Their detailed specifications can be found in the methodology.

1. **Data Acquisition**:
    - A dataset specific to Kannada is compiled from undergraduate students of BMSCE. These students copied 21 words on an A4 sheet, systematically divided into grid boxes to ensure consistent data capture.
2. **Data preprocessing:**
    - Canny edge detection algorithm is applied to grayscale images followed by contour analysis as canny edge detection effectively captures sharp transitions between character strokes, common in Kannada writing.
    - Compared to Watershed segmentation directly applied to the entire image, Canny edges provide cleaner initial boundaries for individual cells, reducing segmentation errors.
    - Grid based image cropping is implemented to extract pertinent writing regions from the dataset from A4 sheet.
    - Denoising techniques such as grey scaling is applied as it simplifies the image processing by removing irrelevant colour information while retaining essential stroke and shading details needed for further analysis.
    - Gaussian blur which smoothens the image, reducing isolated pixel variations and improving edge consistency.
    - Thresholding is used to eliminate undesired artifacts from the images as it binarizes the images (black and white) to clearly differentiate strokes from background.
    - Image enhancement procedures such as elevation maps are applied to standardise contrast and brightness. Segmentation techniques to isolate individual characters from the whole word.

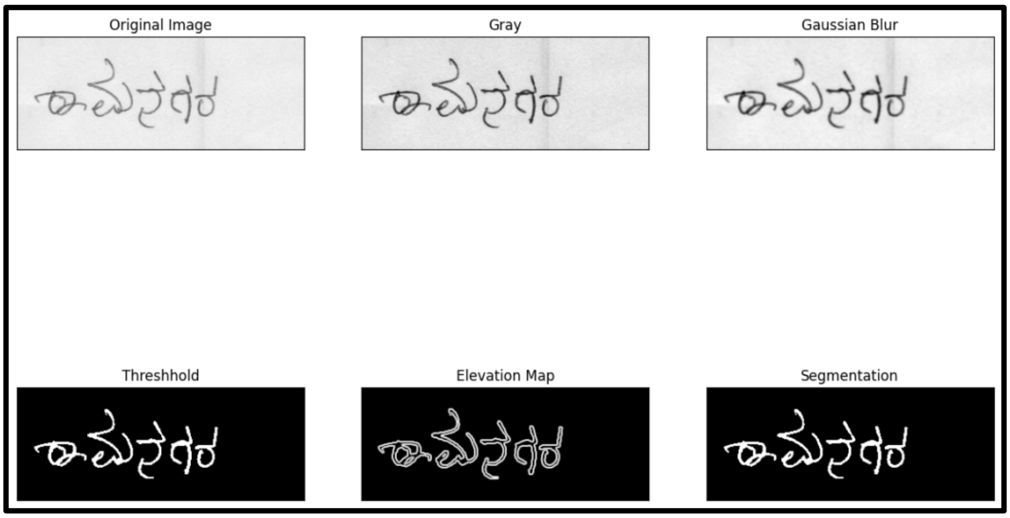


Figure 4.2- Filters applied on the cropped image

3. **Feature identification:**
    - The pressure analysis done in the previous stage acts as one of the features for statistical analysis and classifies the dysgraphia symptoms. Height and length of the word, skew of the written word, height, length and distance between each character in a word.
    - These six features are chosen as they can be easily identified and analysed by the handwriting analysis.
    - Watershed algorithm is applied to thresholded grayscale images with predefined markers to extract each character from the whole word.
    - Watershed effectively handles touching or overlapping characters by identifying valleys between strokes as segmentation points.
    - Compared to fixed grid-based methods, Watershed adapts to individual writing styles and variations in character spacing.
    - Predefined markers (areas with low and high intensity) guide the segmentation process, improving accuracy for challenging cases.
    - These features are extracted by drawing the bounding boxes and contours around each character in the word and taking the highest and lowest points of each character.
    - The highest and lowest points allow us to calculate the width and height of each character in the word.
    - By drawing a horizontal line by taking the highest and lowest point of the first character in the word we get the skew angle of the written word.

- These features are then used to do the statistical analysis.
- Table 4.1 shows the sample extracted features from the dataset collected.

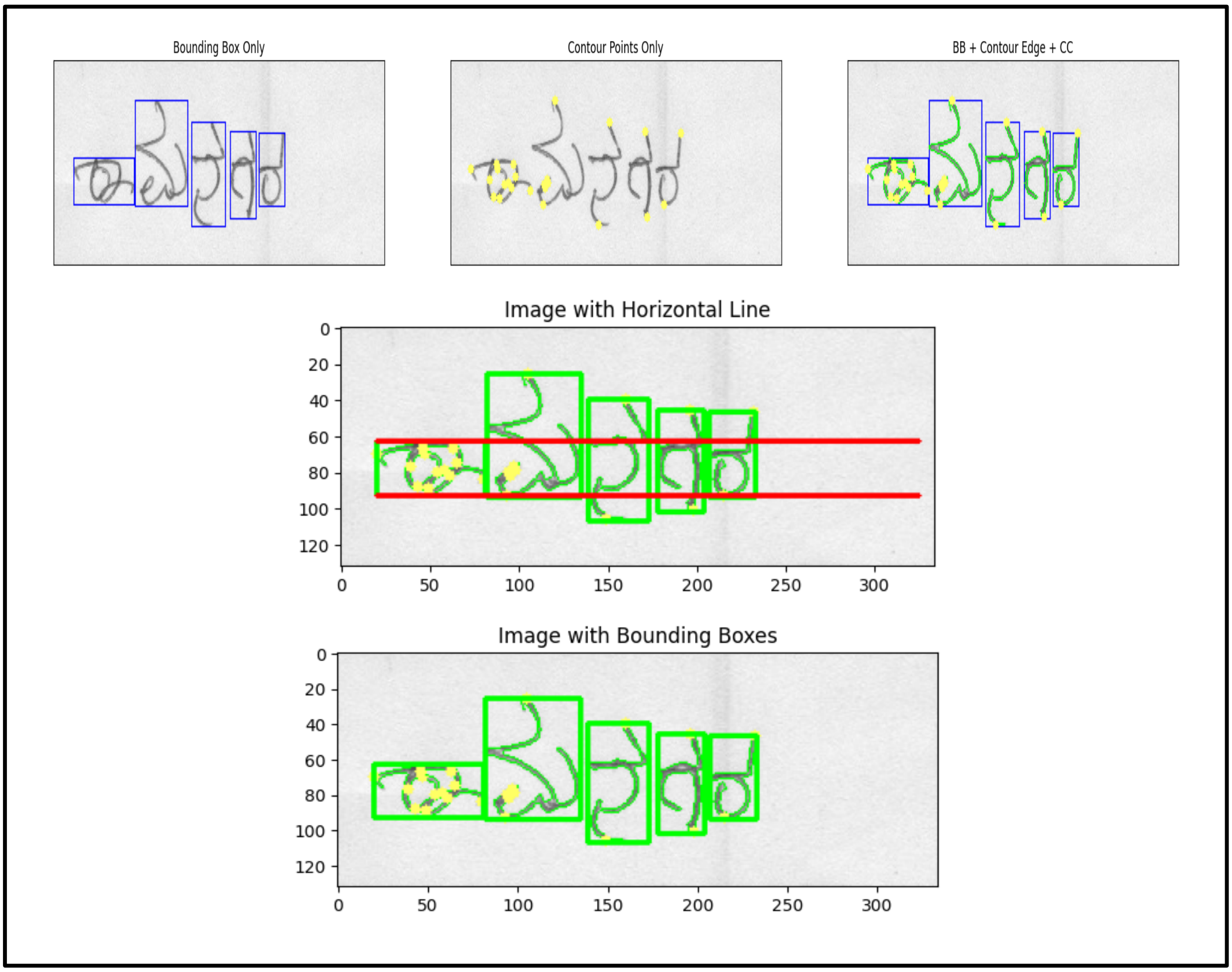


Figure 4.3 - Cropped image with contours and feature points

SI.No	Char	Bounding Box	Char Width	Character Height	Character Gaps	Word Width	Word Height	Skewed Angle	ClassId
0	Char_1	(56, 55, 39, 36)	39	36	4	87	10	[7.1134853]	0
1	Char_2	(99, 58, 42, 27)	42	27	2	87	10	[7.1134853]	0
2	Char_3	(143, 65, 20, 18)	20	18	2	87	10	[7.1134853]	0
3	Char_1	(56, 55, 39, 36)	39	36	4	87	10	[1.1134853]	0
4	Char_2	(99, 58, 42, 27)	42	27	2	87	10	[1.1134853]	0

Table 4.1 - Sample features extracted from the dataset

4. **Model selection:**
   - CNN [19-20] model is used to train on the dataset. compare.
   - CNN is used because of its power to identify the features from images.
   - Random forest classification is used to perform classification.
   - **Threshold:** To make a definitive call, a threshold score is set. If the dysgraphia score predicted by the model exceeds this threshold, the handwriting sample is classified as dysgraphic. If it falls below, it's considered non-dysgraphic.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 112, 112]           1,792
         MaxPool2d-2           [-1, 64, 56, 56]               0
              ReLU-3           [-1, 64, 56, 56]               0
            Conv2d-4          [-1, 192, 56, 56]         110,784
         MaxPool2d-5          [-1, 192, 28, 28]               0
              ReLU-6          [-1, 192, 28, 28]               0
            Conv2d-7          [-1, 384, 28, 28]         663,936
              ReLU-8          [-1, 384, 28, 28]               0
            Conv2d-9          [-1, 256, 28, 28]         884,992
             ReLU-10          [-1, 256, 28, 28]               0
           Conv2d-11          [-1, 256, 28, 28]         590,080
        MaxPool2d-12          [-1, 256, 14, 14]               0
             ReLU-13          [-1, 256, 14, 14]               0
          Dropout-14                [-1, 25088]               0
           Linear-15                 [-1, 1000]      25,089,000
             ReLU-16                 [-1, 1000]               0
          Dropout-17                 [-1, 1000]               0
           Linear-18                  [-1, 256]         256,256
             ReLU-19                  [-1, 256]               0
           Linear-20                   [-1, 43]          11,051
================================================================
Total params: 27,607,891
Trainable params: 27,607,891
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 10.15
Params size (MB): 105.38
Estimated Total Size (MB): 116.10
----------------------------------------------------------------
```

Figure 4.4 - CNN model architectures

4.2.1 Interface Design

The proposed slow learning disability prediction system operates through a series of interconnected subsystems with defined inputs and outputs. Figure 4.5 briefly describes the proposed subsystem interface.

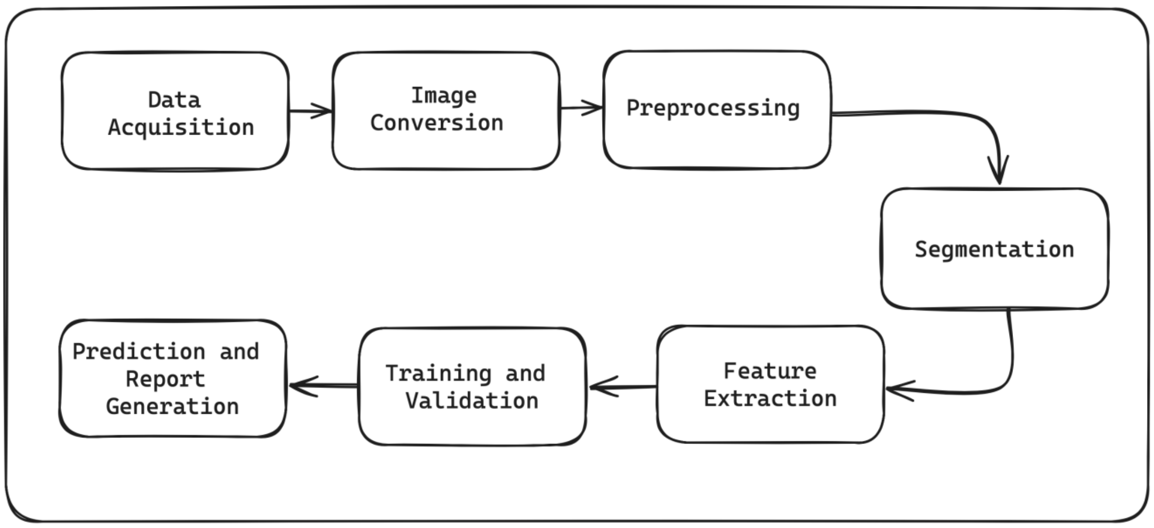


Figure 4.5 – Interface Design

4.3 Use-case diagram

Figure 4.6 depicts the use case diagram with the following entities using the project.

1. Faculty Dictates Kannada Words :

   Faculty dictates Kannada words to students during language learning sessions.

   Purpose: Facilitate language comprehension and pronunciation practice.

2. Scan and Upload the Sheet :

   Faculty scans the sheet containing their handwritten responses.

   Purpose: Digitize handwritten content for further processing and analysis.

3. Alert of Abnormal Handwriting :

   System analyzes scanned sheets to detect deviations in handwriting patterns.

   Purpose: Provide early alerts to educators or administrators regarding potential handwriting abnormalities for further evaluation.

4. Inform Concerned Parents/Doctors :

   Upon detection of abnormal handwriting, system notifies parents or healthcare professionals.

   Purpose: Enable timely intervention and support for students with handwriting-related issues.

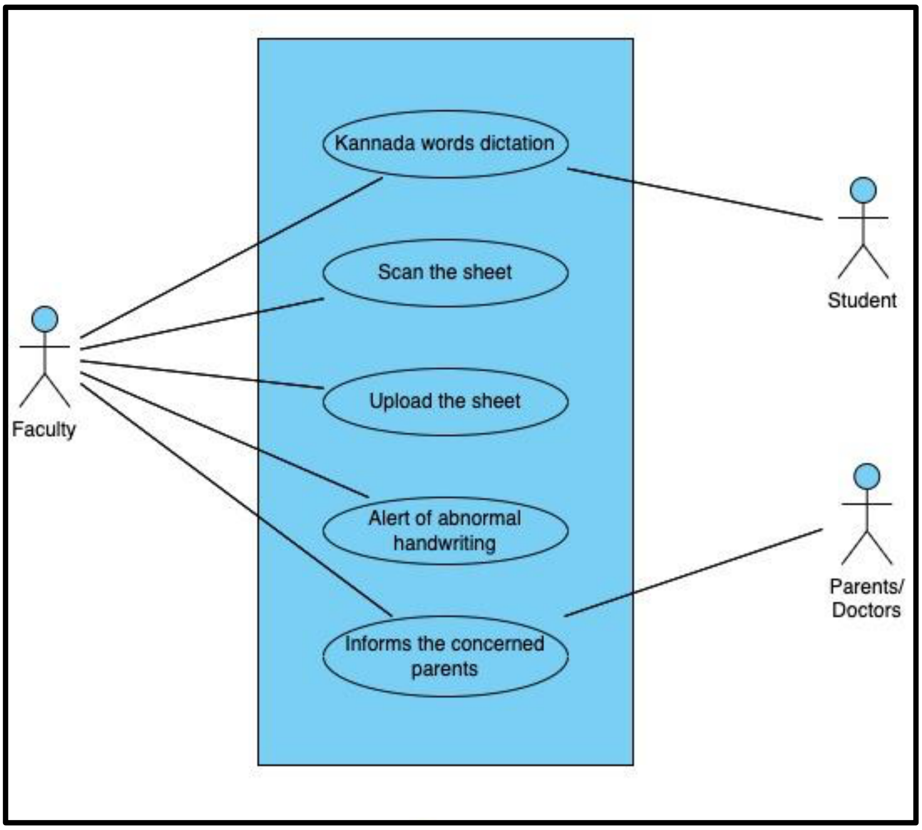


Figure 4.6 - Use Case Diagram for the concerned users

4.4 Class diagram

The class diagram shown in Figure 4.7 depicts the core components involved in the system.

Classes:

User: Represents the system user who interacts with the application to analyze handwritten data.

Predict: (Abstract Class) This abstract class defines the core functionality for making predictions on handwritten data. Concrete subclasses (CNNModel and RandomForestModel) inherit from this class and implement specific prediction algorithms.

Evaluate: This class handles the evaluation metrics used to assess the performance of the prediction models. It interacts with the Predict class to obtain predictions and calculates metrics like accuracy.

CNN Model: This concrete subclass of Predict implements a Convolutional Neural Network (CNN) model for classifying handwritten data.

RandomForest Model: This concrete subclass of Predict implements a Random Forest model for classifying handwritten data.

Handwritten Dataset: This class encapsulates the handwritten data used for training and testing the machine learning models. It provides methods for data access and preprocessing.

Relationships:

Inheritance: Predict acts as an abstract superclass for CNNModel and RandomForestModel, enforcing a common interface for prediction tasks.

Association: User interacts with the system functionalities provided by other classes.

Association: Predict and Evaluate collaborate to assess the performance of the prediction models. (Predict provides predictions, Evaluate calculates metrics)

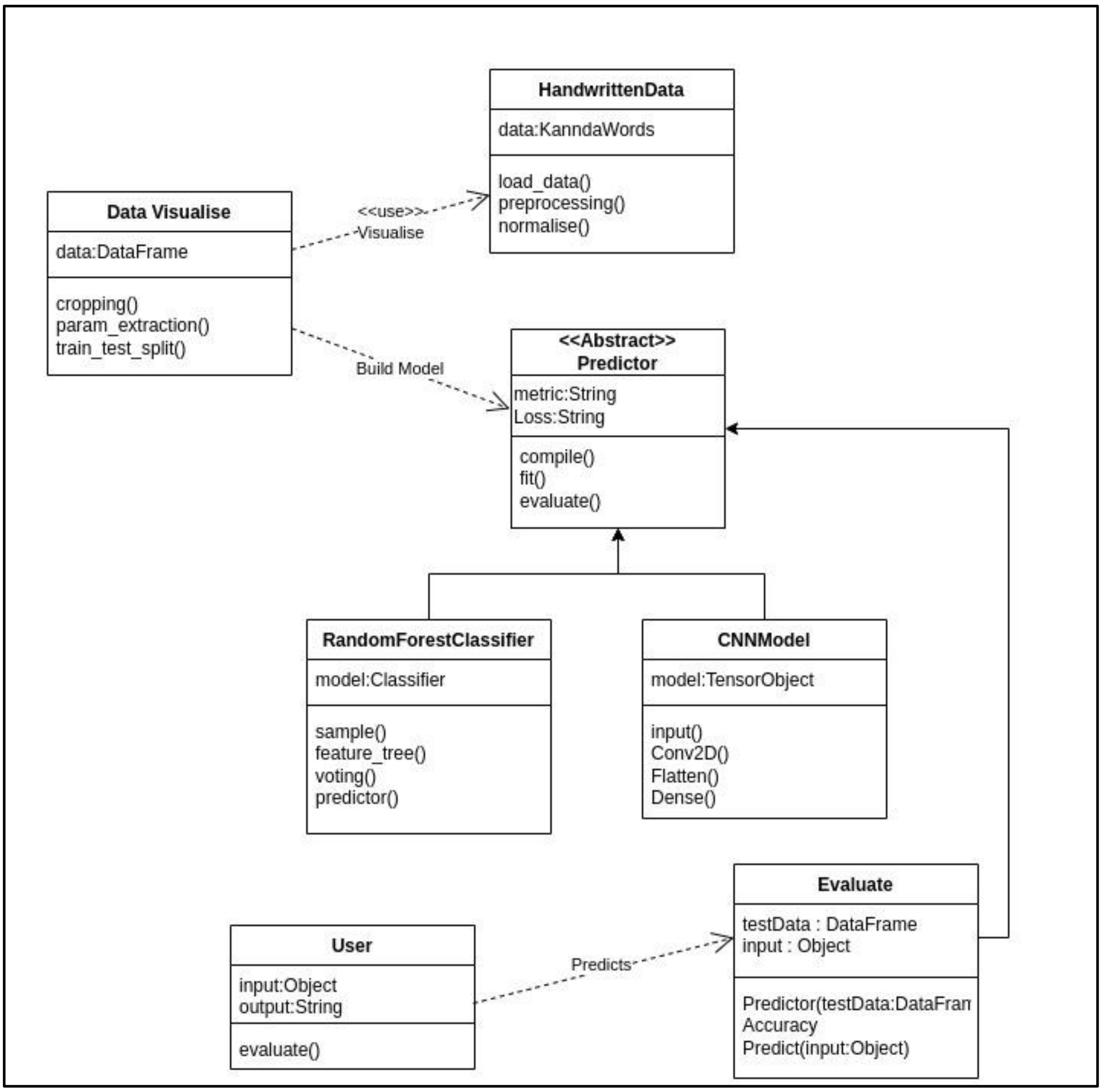


Figure 4.7 - Class Diagram depicting the different classes involved in the project

4.5 Sequence diagram

This sequence diagram in Figure 4.8 illustrates the interaction flow between a User, Application, and Model in a handwritten data analysis system.

Participants:

User: Initiates the analysis process by uploading a handwritten data sample.

Application: Acts as an intermediary between the User and the Model, handling data pre-processing and result presentation.

Model: Performs the analysis on the pre-processed data and generates a prediction output.

Sequence of Interactions:

Upload Data: The User uploads a handwritten data sample (image or document) to the Application.

Pre-process Image: The Application receives the data and performs necessary pre-processing steps. This may involve tasks like image resizing, noise reduction, or format conversion.

Send Data to Model: The pre-processed data is sent by the Application to the Model for analysis.

Analyze Data: The Model receives the data and performs its analysis using its trained algorithms (e.g., CNN, Random Forest).

Generate Prediction: Based on the analysis, the Model generates a prediction output. This could be a classification label (e.g., handwritten digit) or a more complex result depending on the application.

Return Prediction: The Model sends the prediction output back to the Application.

Display Result: The Application receives the prediction and translates it into a user-friendly format for presentation. This may involve displaying the predicted label, confidence score, or visualization of the results.

Present to User: The Application displays the processed and analyzed data along with the prediction output to the User.

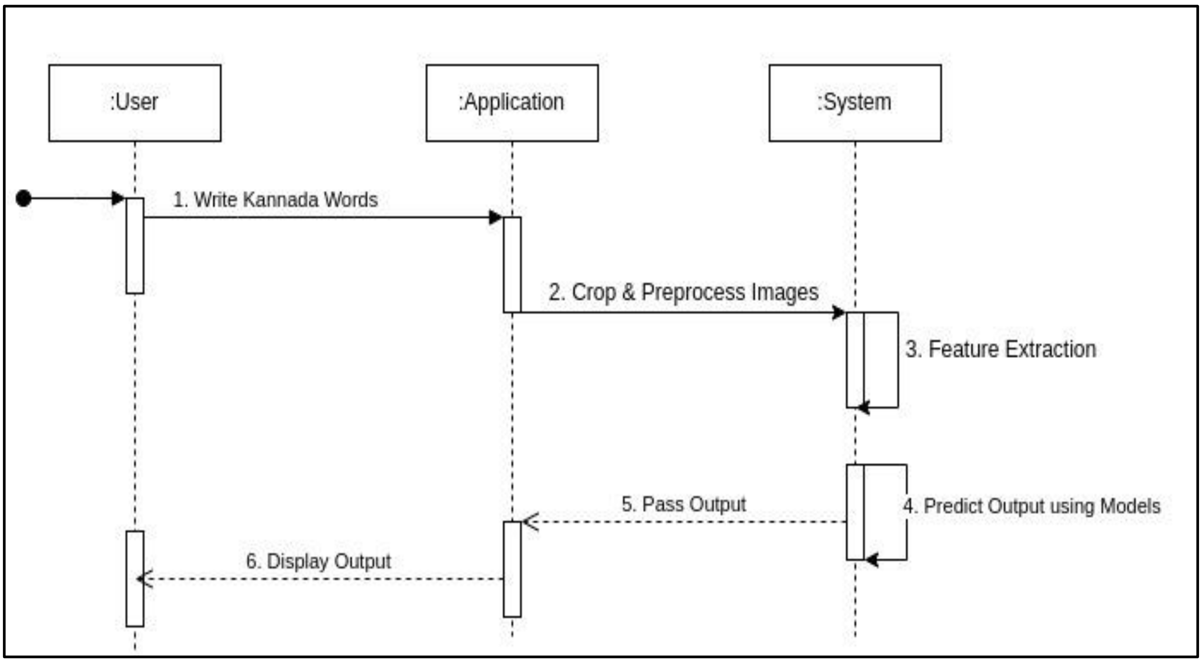


Figure 4.8 - Sequence Diagram depicting the flow of the data

Chapter 5

Implementation

5.1 Overview of the Technologies Used

5.1.1 Cloud Resources:

Google Colab: This free Jupyter notebook environment lets you train complex models in the cloud. It provides access to powerful GPUs, making tasks like training deep learning models on large datasets much faster and more accessible. We used Colab for training all our models.

Google Drive: This cloud storage platform keeps your data safe and readily available. It simplifies file sharing and collaboration, allowing teams to work together seamlessly. We used Google Drive to store our datasets and trained models.

5.1.2 Programming Languages and Libraries:

Python: As a high-level, interpreted language, Python is known for its readability and ease of use. It supports various programming paradigms like object-oriented, functional, and procedural, making it versatile for different project needs. Python's extensive ecosystem of machine learning libraries makes it a favorite for such projects.

OpenCV: This open-source computer vision library played a crucial role in our project. We used OpenCV for image processing tasks like extracting features (parameters) from individual letters in images. This prepared the data for further analysis by other libraries.

TensorFlow: This open-source library underpins various machine learning applications, including neural networks. Its use of data flow graphs and differentiable programming allows for efficient model development and training.

Keras: Built on top of TensorFlow, Keras is a high-level neural network API. It simplifies the creation and experimentation with deep learning models thanks to its modular design.

Other Libraries: Additional libraries like matplotlib, scikit-learn, pandas, and NumPy provided functionalities for data visualization, manipulation, and analysis.

5.2 Implementation details of the modules

The process begins with the extraction module, which navigates through a folder containing all the scanned sheets and cropping each grid box into individual files. This is followed by pre-processing, which involves various techniques such as Canny edge, watershed, denoising, Gaussian blur, thresholding, elevation maps, contour points, segmentation, baseline characteristics, among others.

Once the pre-processing is complete, the OpenCV2 module takes each image as an individual input and extracts the required data from it. This includes the character width and height, word width and height, inter-character spacings, and skew angle. All of these characteristics are then compiled into a dataframe, which stores all the metrics in a serial order. This dataframe is subsequently converted into a csv file, which serves as a comprehensive dataset for further analysis.

In addition to the above, the metrics from the csv file are used to train and fine-tune the Convolutional Neural Network (CNN) model and the Random Forest Classifier. Both of these models are then saved and used for future analysis.

5.3 Difficulties encountered and strategies used to tackle

One of the primary challenges was procuring the dataset, which required significant effort and resources to collect and organize. Additionally, obtaining high-quality data from scanned sheets was a significant challenge due to the need for people to write the sample texts in quiet environments without any hindrance.

Another challenge was understanding how to leverage code to capture the features from the text as advised by the graphologist. This required a deep understanding of the data extraction process and the ability to develop custom code to extract the required features from the text.

To address these challenges, a systematic approach was taken to data extraction, which involved reaching out to students and dictating the words. Secondly, to address the challenge of obtaining high-quality data from scanned sheets, we made sure all sheets were scanned using flat-bed scanners and not cameras or mobile phones to avoid loss of data. This also involved pre-processing the data to remove any contamination from the images and to extract the text features precisely. Finally, to address the challenge of understanding how to leverage our code, we consulted graphologists to understand how we can assess the pressure of the text and decide upon the parameters that can be of our use.

Chapter 6

Testing and Experimental Analysis and Results

6.1 Unit Testing

Unit tests would be created for each module involved in the data preprocessing pipeline. This includes tests for scanning the dataset, arranging the images into their respective directories based on the classes of the machine learning model, cropping the images to the desired dimensions, and storing them properly. Additionally, unit tests would be created for the feature extraction process to ensure that individual features are extracted accurately from the images, verifying that the cropping process accurately extracts the relevant portions of the images while preserving important visual information, ensuring that no essential features are lost during cropping.

Furthermore, the tests ensure that the images are stored securely and efficiently, with proper metadata and naming conventions, facilitating easy retrieval and access during subsequent stages of the pipeline.

Finally, unit tests are devised to assess the accuracy and reliability of the feature extraction process. This crucial module extracts individual features from the preprocessed images, which are used as input for the machine learning models. The tests verify that the feature extraction process accurately captures relevant image characteristics, facilitating effective model training and prediction. Additionally, these tests may involve fine-tuning the feature extraction parameters to optimize model performance based on the extracted features.

6.2 Integration Testing

Involved testing the integration of all these individual modules or components to ensure that they work together as expected. For our project, integration tests would involve using the features extracted from the images to train the machine learning models encompassing the optimization and fine-tuning of the entire workflow to ensure cohesive functionality. This would include pipelining the entire process from scanning the dataset to classification by the convolutional neural network (CNN) and random forest models. Integration testing ensures that the entire workflow, from data preprocessing to model training and evaluation, functions seamlessly. This comprehensive testing approach entails pipelining the entire process, starting from scanning the dataset and preprocessing the images to training and evaluating the models.

6.3 Experimental Dataset

The dataset initially consisted of a small set of text samples collected from students of JGRV. Later, the dataset was expanded to include undergraduate students, likely to increase the diversity and representativeness of the dataset. Gathering data from different sources and populations is crucial for building robust machine learning models that generalize well to unseen data. The experimental dataset serves as the foundation for training and evaluating the performance of the machine learning models developed in the project.

Chapter 7

Conclusion and Future Enhancements

7.1 Conclusion

Slow learning disability detection in people helps to effectively treat them and also prevent low self-esteem. Few methods were applied to machine learning and deep learning techniques for predicting the probability of having slow learning disability. Existing methods have the limitations of lower efficiency in feature analysis and overfitting problems. By addressing the limitations of existing systems, this project offers a language-specific, accessible, and accurate solution for slow learning disability detection in Kannada.

In this project, the proposed CNN model shows 61.80% accuracy in predicting the slow learning disability and it has the advantage of effective feature analysis.

Random Forest classifier showed accuracy of 68.09%. Which needs the features extracted from the image data in the csv format.

7.2 Future Enhancements

Our research lays the groundwork for future advancements in dysgraphia diagnosis using machine learning. The future of dysgraphia diagnosis holds promise for greater accuracy and personalised support. One promising avenue is the exploration of multimodal datasets. This means incorporating not just written samples, but also data on factors like pressure applied while writing and pen lifts. These additional dimensions could provide a more comprehensive picture of writing difficulties.

Another exciting direction lies in developing novel methods to capture pressure and pen-lift data. This would allow for a more seamless and user-friendly data collection process.

Finally, researchers are delving deeper into the impact of writing systems (orthography) and specific tasks on machine learning models used for diagnosis. We can expect further studies to shed light on these aspects, leading to more robust and adaptable diagnostic tools. Additionally, with broader datasets encompassing various age groups, the technology could evolve to suggest targeted interventions for individuals with dysgraphia. This, coupled with more sophisticated models, could pave the way for not only diagnosing dysgraphia but also suggesting personalized intervention strategies.

References

[1] American Psychiatric Association (APA). (2013). Diagnostic and statistical manual of mental disorders (5th ed.). Author.

[2] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[3] Lomurno, Eugenio, Linda Greta Dui, Madhurii Gatto, Matteo Bollettino, Matteo Matteucci, and Simona Ferrante. 2023. "Deep Learning and Procrustes Analysis for Early Dysgraphia Risk Detection with a Tablet Application" Life 13, no. 3: 598. https://doi.org/10.3390/life13030598

[4] Jayakanth Kunhoth, Somaya Al-Maadeed, Suchithra Kunhoth, Younus Akbari. (in press) 2022. “Automated Systems For Diagnosis of Dysgraphia in Children: A Survey and Novel Framework” https://doi.org/10.48550/arXiv.2206.13043

[5] Drotár, P., Dobeš, M. Dysgraphia detection through machine learning. Sci Rep 10, 21541 (2020). https://doi.org/10.1038/s41598-020-78611-9

[6] G. Dimauro, V. Bevilacqua, L. Colizzi and D. Di Pierro, "TestGraphia, a Software System for the Early Diagnosis of Dysgraphia," in IEEE Access, vol. 8, pp. 19564-19575, 2020, doi: 10.1109/ACCESS.2020.2968367.

[7] Yogarajah, P., & Bhushan, B. (in press). “Deep Learning Approach to Automated Detection of Dyslexia-Dysgraphia”. Paper presented at The 25th IEEE International Conference on Pattern Recognition, Milan, Italy. 2020.

[8] Ghouse, Fathima; Paranjothi, Kavitha; Vaithiyanathan, Revathi. 2022. “Dysgraphia Classification based on the Non-Discrimination Regularization in Rotational Region Convolutional Neural Network”. International Journal of Intelligent Engineering & Systems . 2022, Vol. 15 Issue 1, p55-63. 9p.

[9] Jayakanth Kunhoth, Somaya Al Maadeed, Moutaz Saleh, Younes Akbari. 2023. “CNN feature and classifier fusion on novel transformed image dataset for dysgraphia diagnosis in children”. Expert Systems with Applications. Volume 231. 120740. https://doi.org/10.1016/j.eswa.2023.120740.

[10] Nuriel Shalom Mor, Kathryn Dardeck. 2021. “Applying a Convolutional Neural Network to Screen for Specific Learning Disorder” Learning Disabilities: A Contemporary Journal 19(2), pp. 161-169.

[11] Avishka, I., Kumarawadu, K., Kudagama, A., Weerathunga, M., & Thelijjagoda, S. (2018). Mobile App to Support People with Dyslexia and Dysgraphia. 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS). doi:10.1109/iciafs.2018.8913335

[12] Villegas-Ch, William, Isabel Urbina-Camacho, and Joselin García-Ortiz. 2023. "Detection of Abnormal Patterns in Children’s Handwriting by Using an Artificial-Intelligence-Based Method" Informatics 10, no. 2: 52. https://doi.org/10.3390/informatics10020052

[13] Iza Sazanita Isa, Wan Nurazwin Syazwani Rahimi, Siti Azura Ramlan, Siti Noraini Sulaiman. 2019. “Automated Detection of Dyslexia Symptom Based on Handwriting Image for Primary School Children” Procedia Computer Science, Volume 163, pp 440-449. https://doi.org/10.1016/j.procs.2019.12.127.

[14] V. Vilasini, B. Banu Rekha, V. Sandeep and V. Charan Venkatesh, "Deep Learning Techniques to Detect Learning Disabilities Among children using Handwriting," 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 2022, pp. 1710-1717, doi: 10.1109/ICICICT54557.2022.9917890.

[15] N. N. Doshi, M. U. Maniyar, K. K. Shah, N. D. Sarda, M. Narvekar and D. Mukhopadhyay, "A Convolutional Recurrent Neural Network-Based Model For Handwritten Text Recognition To Predict Dysgraphia," 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 145-150, doi: 10.1109/ICISCoIS56541.2023.10100514.

[16] Devi, G. Kavya. 2023. “Dysgraphia disorder forecasting and classification technique using intelligent deep learning approaches”. Progress in Neuro-Psychopharmacology and Biological Psychiatry. Volume 120. 110647. https://doi.org/10.1016/j.pnpbp.2022.110647.

[17] Overvelde, A., & Hulstijn, W. (2011). Handwriting development in grade 2 and grade 3 primary school children with normal, at risk, or dysgraphic characteristics. Research in Developmental Disabilities, 32(2), 540–548. doi:10.1016/j.ridd.2010.12.027

[18] C. Sharmila, N. Shanthi, S. Santhiya, E. Saran, K. Sri Rakesh and R. Sruthi, "An Automated System for the Early Detection of Dysgraphia using Deep Learning Algorithms," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 251-257, doi: 10.1109/ICSCDS56580.2023.10105022.

[19] Drotár, P., Dobeš, M. Dysgraphia detection through machine learning. Sci Rep 10, 21541 (2020). https://doi.org/10.1038/s41598-020-78611-9

[20] B.R. Kavitha and C. Srimathi, “Benchmarking on offline handwritten tamil character recognition using convolutional neural networks”, Jour- nal of King Saud University - Computer and Information Sciences, 34(4):1183–1190, 2022.

APPENDIX A: Snapshots

The following snapshots depict the accuracy of the CNN Model over the different epochs. Figure A.1 shows the accuracy, that is the training accuracy and the validation accuracy shown by the blue and the orange lines respectively, where the Y-axis represents the accuracy percentage and the X-axis shows the epoch number.

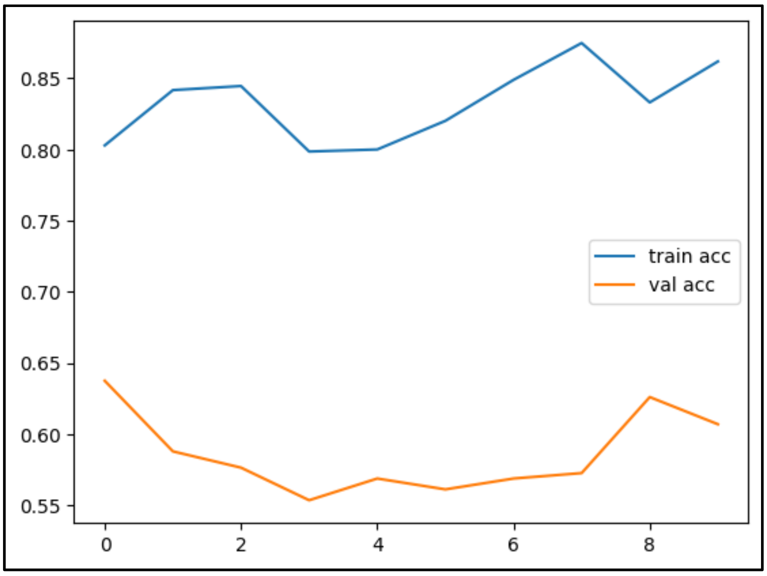


Figure A.1. Accuracy Graph

The following snapshots depict the accuracy of the CNN Model over the different epochs. Figure A.2 shows the loss, that is the training loss and the validation loss shown by the blue and the orange lines respectively, where the Y-axis represents the loss percentage and the X-axis shows the epoch number.

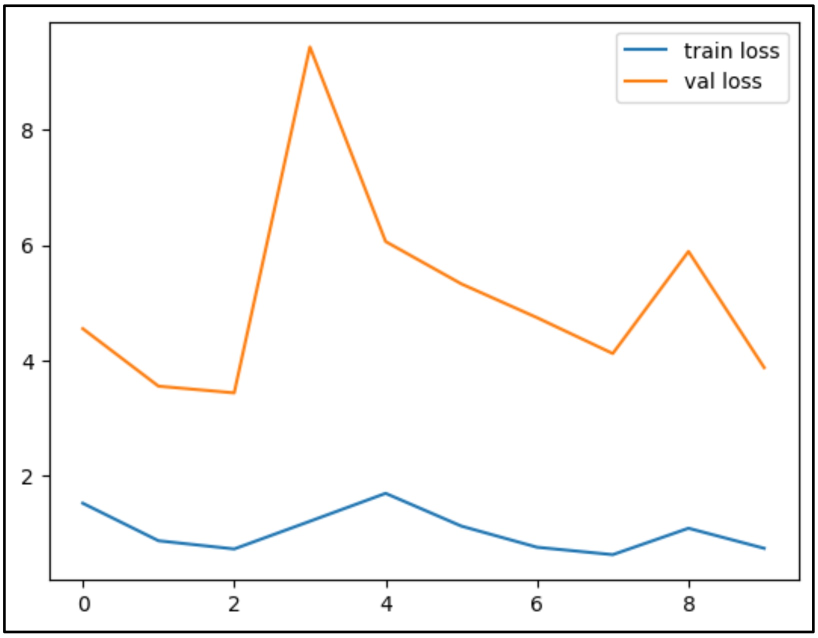


Figure A.2. Loss Graph

Figure A.3 shows the final output of the project which classifies the uploaded dataset as abnormal based on the extracted features as shown in the snapshot.

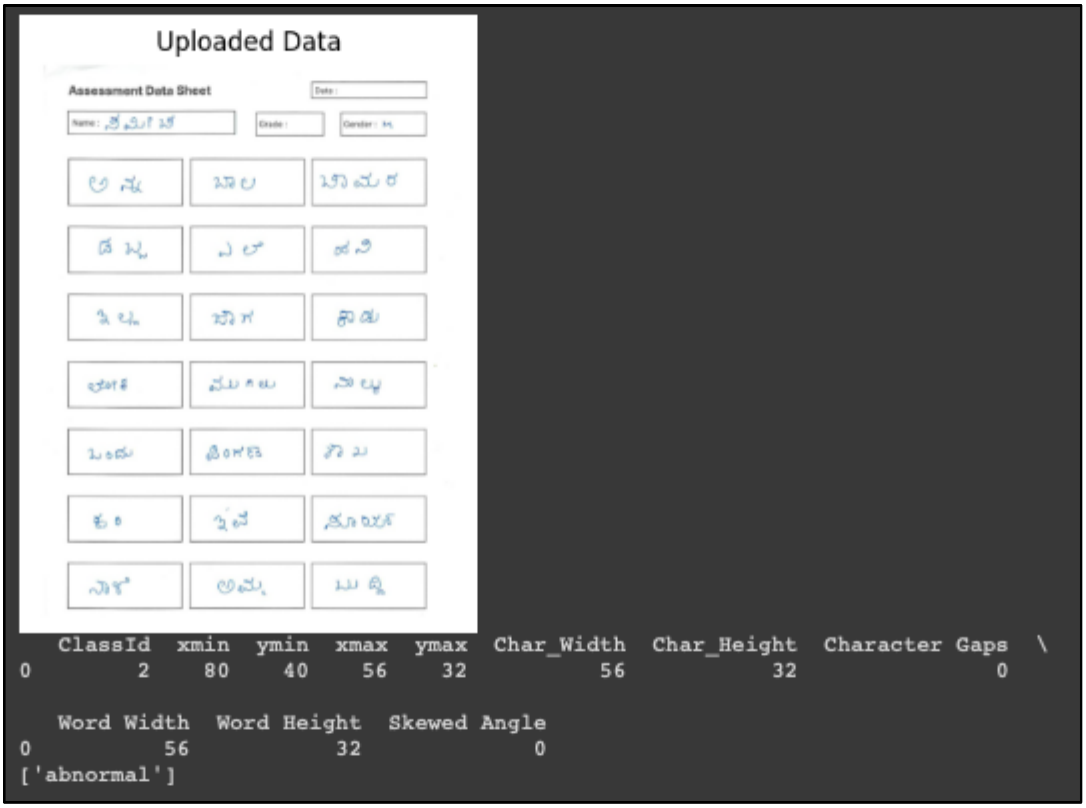


Figure A.3. Final Output

APPENDIX B: Details of publications

Author Names: Tushar B T, Umang Goel, Vinay Kulkarni, Varun Urs M S, Kavitha Sooda.

Paper Title: Automated Detection of Dysgraphia Symptoms In Primary and Middle School Children

Name of the Conference: 2024 International Conference on Emerging Smart Computing and Informatics (ESCI)

Place of the Conference: AISSMS Institute of Information Technology, Pune, India

Vol No. , Issue No. , Page No.s of Journal : 2024 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2024, pp. 1-5, doi: 10.1109/ESCI59607.2024.10497397

Date of Conference : 05-07 March 2024

Date of Publication: 17 April 2024

APPENDIX D: Programme Outcomes and Programme Specific Outcomes Mapped

Batch No. : 13 Date: 01/05/2024

Project Title: Detection of SLD in Kannada Vocabulary Amongst Undergraduate Students

PROGRAMME OUTCOMES	Level (1/2/3)	Justification if addressed
PO1	2	This research applies knowledge of engineering principles (pattern recognition, image processing, machine learning) to the complex problem of dysgraphia detection.
PO2	2	The project identifies and formulates the problem of dysgraphia detection, researches existing solutions, and analyzes data to reach substantiated conclusions.
PO3	2	A system for analyzing handwriting and classifying dysgraphia risk levels is designed, considering societal issues like affordability and language inclusivity.
PO4	2	Experiments are designed and conducted to collect handwriting data, with the data analyzed using various techniques and conclusions drawn based on the results.
PO5	2	Modern tools like open-source software and pre-trained models are employed for image processing and machine learning.
PO6	2	The project considers the societal impact of dysgraphia and the ethical implications of AI-based handwriting analysis.
PO7	1	While the project utilizes technology, the environmental impact is not directly addressed.
PO8	2	Ethical principles are followed in data collection and analysis, emphasizing anonymity and respect for participants.
PO9	3	Teamwork skills are demonstrated through coherent collaboration amongst everyone during the project.
PO10	3	The project details and findings are communicated through the research paper and potentially presentations, showcasing communication skills.
PO11	2	Project management aspects like budget and timeline are explicitly mentioned.
PO12	3	This research contributes to a growing field and encourages further development in dysgraphia detection, demonstrating a commitment to lifelong learning.

PROGRAMME SPECIFIC OUTCOMES	Level (1/2/3)	Justification if addressed
PSO1	2	Software engineering principles are applied in developing the pre-processing and analysis algorithms for the system.
PSO2	2	The system analyzes visual data and processes it using algorithms, addressing aspects of network and mobile computational systems, although it is not a web-based application.
PSO3	3	Image processing and machine learning algorithms are utilized, showcasing the ability to design efficient algorithms.

APPENDIX E: Plagiarism report

DrillBit

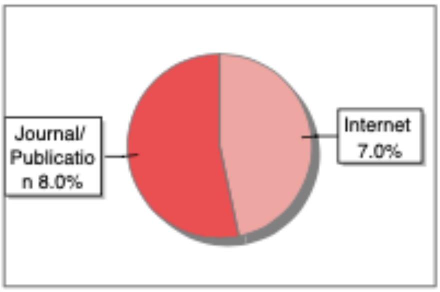
The Report is Generated by DrillBit Plagiarism Detection Software

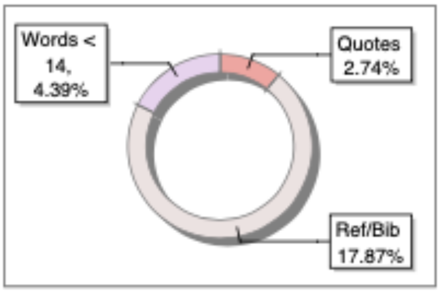
Submission Information

Author Name	TUVV
Title	TUVV
Paper/Submission ID	1771714
Submitted by	hod.cse@bmsce.ac.in
Submission Date	2024-05-09 10:07:56
Total Pages	40
Document type	Project Work

Result Information

Similarity **15 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Sources: Less than 14 Words %	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File