Neural Network-Based System for Early Detection & Intervention of Dysgraphia in Children

Final Year Project (FYP) Report

- Final Year Thesis -

FACULTY OF COMPUTING AND
INFORMATION & TECHNOLOGY

SRI LANKA TECHNOLOGY CAMPUS

FINAL YEAR PROJECT THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
B.SC. HONS. IN DATA SCIENCE

October 2025

Supervisor: Dr. M.W.P. Maduranga

(Senior Lecturer Grade I, Faculty of Computing and IT, SLTC)

Dedication

This thesis is dedicated to the children who face learning challenges with resilience and determination, inspiring us to create a more inclusive future.

To our parents, teachers, and mentors for their constant guidance and encouragement, and to the professionals and institutions contributing toward a more inclusive learning environment in Sri Lanka.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is our own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgments.

Sahan Chamara Silva

Chathuranga Bandara
October 2025

Acknowledgements

And I would like to acknowledge our deepest gratitude to Brigadier Upendra Karunarathne and Col. Dilan Wickramasinghe at the Senehasa Education Resource Research and Information Centre (SERRIC), as well as to Dr. G. Wijesuriya and Dr. Swarna Wijeatunge at the Lady Ridgeway Hospital for Children (LRH), for granting permission and providing valuable support for data collection in this study.

I am immensely grateful to the occupational therapists, psychologists, child psychologists, special education teachers, school principals, and teachers who contributed their time and expertise to facilitate data collection and knowledge sharing. Special appreciation is extended to the students and parents whose participation and cooperation were invaluable to this research.

I would also like to convey our heartfelt thanks to the SLTC Ethical Committee for granting the ethical clearance required to collect data from children, ensuring this research adhered to proper ethical standards.

Finally, I extend our sincere appreciation to our supervisor, mentors, and peers for their continuous guidance, encouragement, and constructive feedback throughout the academic journey.

Abstract

This study has developed an AI driven system for the early detection and intervention of dysgraphia among children aged six to ten years in Sri Lanka. The research was motivated by a significant lack of awareness and diagnostic tools for learning disabilities within local educational settings. A hybrid methodology combining quantitative survey analysis, observational interviews, and advanced deep learning techniques was employed. A new numeric handwriting dataset was created locally, complementing an existing letter dysgraphia dataset to train and evaluate multiple models. Comparative experiments between machine learning and deep learning approaches demonstrated the superior performance of an Advanced Convolutional Neural Network (CNN), achieving accuracies of 99.2% and 97.4% for letter and numeric dysgraphia detection, respectively. The system architecture integrated detection, rehabilitation, and community support through gamified exercises and an interactive AI avatar. Survey results revealed that over 92% of participants were unaware of learning disorders, yet all expressed openness to adopting digital support tools. This research has successfully produced a culturally relevant, technically robust platform that not only identifies handwriting difficulties but also promotes early educational intervention and awareness. The proposed system establishes a foundation for inclusive, AI assisted learning in Sri Lanka and offers a scalable model for broader regional adoption.

Table of Contents

List of Figures

List of Tables

Nomenclature

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
ML	Machine Learning
SLD	Specific Learning Disorder
LRH	Lady Ridgeway Hospital
SVM	Support Vector Machine
RF	Random Forest
KNN	K-Nearest Neighbors
XGBoost	Extreme Gradient Boosting
VGG16	Visual Geometry Group 16-layer CNN
ResNet50	Residual Network with 50 Layers
BiLSTM	Bidirectional Long Short-Term Memory
ViT	Vision Transformer
Swin	Shifted Window Transformer
AUC	Area Under Curve
GUI	Graphical User Interface
UI/UX	User Interface / User Experience
API	Application Programming Interface
SQL	Structured Query Language
CSV	Comma-Separated Values
PNG	Portable Network Graphics
RGB	Red, Green, Blue (Color Model)
TTS	Text-to-Speech
STT	Speech-to-Text
RAG	Retrieval-Augmented Generation
JSON	JavaScript Object Notation
SDK	Software Development Kit
PII	Personally Identifiable Information

LIST OF SYMBOLS

Symbol	Description
$I(x, y)$	Original image at pixel coordinates (x, y)
$I'(x', y')$	Resized or transformed image
I_{gray}	Grayscale converted image
$I_{blurred}$	Image after Gaussian blur filtering
$T(x, y)$	Adaptive threshold function
w_0, w_1	Class weights in Otsu's method
σ^2	Variance
B	Structuring element in morphological operations
$G(i, j)$	Gaussian kernel value at (i, j)
N	Number of pixels in the local block
C	Threshold adjustment constant

LIST OF TERMINOLOGY

Term	Description
Dysgraphia	A neurological learning disorder that affects handwriting and fine motor skills.
Numeric Dysgraphia	A subtype of dysgraphia that affects the ability to write or form numbers correctly.
Letter Dysgraphia	Difficulty in forming or writing alphabetic characters.
Dataset	A structured collection of handwriting samples used for training and evaluating models.
Epoch	One complete cycle through the entire training dataset in deep learning.
Preprocessing	Techniques used to clean and prepare raw handwriting images for analysis.
Gamified Intervention	Learning activities that use game-like elements to enhance engagement.
3-Tier Architecture	System structure divided into Presentation, Logic, and Data layers.

Chapter 1

Introduction

1.1 Background

Writing is a fundamental skill essential for communication, learning, and cognitive development. However, for a significant number of children, writing poses a consistent and frustrating challenge due to learning disabilities such as dysgraphia. Dysgraphia is a specific learning disorder that affects a child's handwriting ability, spelling, and fine motor coordination, often leading to poor academic performance and reduced self-esteem. Globally, studies estimate that between 10% to 30% of school-aged children experience some degree of dysgraphia, yet awareness, diagnosis, and early interventions remain limited especially in developing regions such as Sri Lanka.

In Sri Lanka, there is a growing awareness of learning disabilities such as dyslexia and dyscalculia, but dysgraphia remains poorly recognized. Many children are mistakenly perceived as lazy or careless rather than being identified as individuals with a neurological disorder that affects their writing motor control. This delay in recognition not only affects their academic progression but also has emotional and psychological consequences.

Traditional screening for dysgraphia relies heavily on manual observation and subjective assessments by teachers or occupational therapists, which are time-consuming, inconsistent, and often unavailable in rural areas. The absence of digitized diagnostic tools and datasets tailored to Sri Lankan children further limits the development of automated or AI-driven screening mechanisms.

Recent advances in Artificial Intelligence (AI) and Deep Learning (DL) particularly Convolutional Neural Networks (CNNs) have transformed handwriting analysis and pattern recognition. When applied to dysgraphia screening, neural networks can analyze the micro-level patterns in children's handwriting to detect early indicators of motor and spatial inconsistencies. However, the success of such systems depends heavily on the availability of large, diverse, and region-specific handwriting datasets something that has not yet been developed for Sri Lankan children.

This research therefore aims to bridge this gap by developing a Neural Network-Based System for Early Detection & Intervention of Dysgraphia in Children, supported by a comprehensive local dataset of handwritten numbers collected from Sri Lankan children across multiple districts. The system not only detects dysgraphia but also integrates educational and psychological intervention modules to support affected children in a holistic, accessible, and inclusive manner.

1.2 Problem Domain

Learning disabilities such as dysgraphia, dyslexia, and dyscalculia are classified as Specific Learning Disorders (SLDs), neurological conditions that affect a child's ability to acquire academic skills despite normal intelligence and adequate education. Among these, dysgraphia specifically impairs handwriting and fine motor control, often manifesting as inconsistent letter or number formation, spacing errors, and difficulties in writing fluency.
Globally, researchers estimate that between 10% to 30% of school-aged children exhibit dysgraphic symptoms to varying degrees. However, in developing nations like Sri Lanka, awareness, diagnostic resources, and targeted interventions remain critically low. Teachers frequently misinterpret poor handwriting as laziness or lack of discipline, while parents may not recognize it as a neurological issue requiring professional attention.
Dysgraphia impacts not only academic performance but also emotional well-being children develop low self-esteem, frustration, and social withdrawal due to repeated negative feedback. Unfortunately, early screening mechanisms are minimal, and there is no national digital repository or standardized dataset reflecting Sri Lankan children's handwriting patterns.
In the context of artificial intelligence, handwriting analysis has emerged as a powerful diagnostic tool. Yet, existing datasets such as MNIST, EMNIST, or IAM are based on Western handwriting, limiting their applicability to Sri Lankan educational contexts. Moreover, these datasets lack categories for numeric dysgraphia, which is especially relevant for early primary school children.
Therefore, the problem domain addressed in this research encompasses AI-driven handwriting analysis for dysgraphia detection and localized dataset development to support educational and clinical applications in Sri Lanka as well as globally.

1.3 Research Motivation & Gap

1.3.1 Research Motivation

This research is motivated by the absence of locally relevant tools and datasets to identify dysgraphia early in Sri Lankan children. While awareness of learning disabilities such as dyslexia has grown, dysgraphia remains largely ignored in both urban and rural education systems. Teachers and parents often lack the necessary knowledge or digital tools to identify and support affected children.
Interviews with psychologists and special education teachers at the Senehasa Education Resource Research & Information Institute and Lady Ridgeway Hospital (LRH) revealed that early handwriting challenges are frequently overlooked until they cause serious academic struggles.
A locally trained AI model could transform this situation by offering, Objective, data-driven diagnosis instead of subjective observation and Scalable digital intervention tools that can reach schools even in remote areas and also Interactive rehabilitation games that make therapy engaging for children. The vision is not just to detect dysgraphia but to empower teachers, parents, and psychologists with a digital ecosystem that combines assessment, monitoring, and intervention.

1.3.2 Research Gap

Despite the global advancements in artificial intelligence and handwriting recognition, several critical research gaps remain in the context of Sri Lanka. Firstly, there is a complete lack of localized handwriting datasets that accurately represent the writing patterns of Sri Lankan children. Most existing datasets, such as MNIST and EMNIST, are based on Western handwriting samples and do not reflect the variations in number and letter formation found in local educational contexts. These differences in structure, spacing, and stroke flow limit the applicability of such datasets for identifying handwriting-related learning disabilities in Sri Lankan students.

Secondly, there has been very limited focus on the area of numeric dysgraphia, which refers to the difficulty in writing numbers. While letter-based dysgraphia has received moderate attention in international research, the numeric form remains largely unexplored. This gap has significant implications for early education, as the ability to write numbers is an essential foundation for mathematical learning in early childhood. The lack of attention to numeric dysgraphia in both research and practice has delayed early detection and intervention for affected children in primary education.

Thirdly, existing automated systems for dysgraphia detection are predominantly diagnostic in nature and do not provide integrated solutions that extend into the intervention phase. Most current approaches focus only on identifying dysgraphic tendencies, neglecting the rehabilitation aspect that helps children improve their writing ability. There is also an absence of gamified and bilingual (Sinhala and English) tools that could make intervention more accessible, interactive, and culturally relevant for children in Sri Lanka.

To address these research gaps, this study focuses on three key contributions. It develops the first Sri Lankan numeric dysgraphia dataset with a high preprocessing accuracy of 97 percent noise reduction, ensuring the quality and reliability of handwriting samples. It further designs an Advanced Convolutional Neural Network (CNN) model that achieves more than 97 percent accuracy in dysgraphia detection, outperforming baseline and transformer-based deep learning models. Finally, the study introduces an AI-driven screening and intervention platform that integrates game-based learning activities, progress monitoring, and multilingual chatbot support to create a holistic ecosystem for dysgraphia screening and rehabilitation.

1.4 Problem Statement

Despite the prevalence of dysgraphia among school-aged children, early identification continues to be a major challenge due to a lack of awareness, diagnostic tools, and specialized professionals. In the Sri Lankan context, there are no publicly available datasets or AI-based systems that cater to the unique handwriting styles of local children. Consequently, teachers and parents often misinterpret handwriting difficulties as negligence, lack of discipline, or poor effort rather than recognizing them as symptoms of a neurological condition.

This misunderstanding results in a significant number of children remaining undiagnosed and without the necessary educational support. As a consequence, they experience declining academic performance in writing-related subjects, heightened frustration and anxiety, and low self-esteem. Over time, this leads to delayed educational progress and restricted future opportunities. Furthermore, the direct application of existing international tools is not feasible due to contextual differences in language, handwriting styles, and educational standards between Sri Lankan and Western children. Therefore, there is an urgent need for a locally tailored, AI-based detection system that can accurately identify early signs of dysgraphia in Sri Lankan children using real-world data and provide accessible, culturally relevant intervention mechanisms.

1.5 The Proposed Solution

This research proposes a Neural Network-Based System for the Early Detection and Intervention of Dysgraphia in Children, designed specifically to address the diagnostic and awareness gap present in Sri Lanka. Unlike conventional diagnostic approaches that rely on manual observation and subjective judgment by teachers or therapists, the proposed system employs artificial intelligence and deep learning to objectively analyze children's handwriting. The model is based on a Convolutional Neural Network (CNN) architecture that has been trained on a locally developed numeric handwriting dataset, comprising digits from 0 to 10 collected from children across multiple districts in Sri Lanka.

The proposed solution introduces several novel contributions. It includes the creation of the first labeled Sri Lankan numeric handwriting dataset, consisting of over eight thousand samples collected from 482 participants. The dataset encompasses both dysgraphic and non-dysgraphic handwriting and has undergone extensive preprocessing to enhance its quality. A custom "Advanced CNN" model was then developed, optimized for handwriting pattern recognition and shown to outperform traditional baseline CNNs and transformer models in both accuracy and robustness.

In addition to detection, the system integrates an interactive intervention platform that combines diagnostic and rehabilitative functions. The platform incorporates gamified learning activities, bilingual AI-powered mental health chatbots in Sinhala and English, and collaborative dashboards that enable real-time communication between parents, teachers, and therapists. Voice and text-based interaction features have been included to ensure accessibility for users of varying literacy and technical levels. The system also supports cultural and linguistic inclusivity by incorporating Sinhala handwriting recognition and interfaces designed specifically for local use. The platform will be validated through expert feedback and pilot testing at the Senehasa Education Resource Research & Information Institute to ensure its practical applicability and reliability in real-world educational and clinical contexts.

1.6 Aim of study

The primary aim of this research is to develop an artificial intelligence-driven neural network system capable of detecting dysgraphia at an early stage through handwriting analysis, while also providing integrated intervention tools to assist children aged six to ten, their parents, and educators. The study seeks to contribute to both academic research and social impact by creating a reliable, accessible, and contextually appropriate solution for the early detection and management of dysgraphia in Sri Lankan children.

1.7 Objectives

The objectives of this research were formulated following the S.M.A.R.T. criteria to ensure that each target is Specific, Measurable, Achievable, Realistic, and Time-Constrained.

- Specific – Build an Advanced Convolutional Neural Network (CNN) to classify dysgraphic vs. non-dysgraphic handwriting samples, integrated within an accessible web-based platform for screening and intervention.

- Measurable – Achieve a model detection accuracy of $\geq 95\%$.

- Achievable – Utilize 8,000 local handwriting samples collected ethically from 482 Sri Lankan participants aged 6 to10 years.

- Realistic - Implement using open source frameworks such as TensorFlow, PyTorch and Keras for modeling and Flask and React for full stack development within a one year academic timeframe.

- Time Constrained - Complete data collection, model training, system integration within 12 months of the research cycle.

The scope of the proposed solution focuses on numeric dysgraphia detection, early intervention modules, and multi-stakeholder collaboration involving children, teachers, parents, and psychologists while excluding complex neurological diagnostics beyond handwriting analysis.

The main objective of this study is to design and implement a neural network-based system for the early detection and intervention of dysgraphia among Sri Lankan children. To achieve this overarching aim, several specific objectives have been identified.

Firstly, the research seeks to collect and preprocess a comprehensive dataset of handwritten numbers from Sri Lankan children aged six to ten years, representing various districts and backgrounds. Secondly, it aims to design engaging educational game based activities that enhance numeric and motor coordination skills, supporting the rehabilitation of children with dysgraphia. Thirdly, it focuses on developing a CNN-based model capable of accurately distinguishing between dysgraphic and non-dysgraphic handwriting patterns.

Furthermore, the study aims to design an AI-powered screening platform that integrates both questionnaires and handwriting analysis to provide an accessible diagnostic tool for teachers and parents. It also includes the development of intervention tools, such as gamified learning environments, AI chatbots, and educational support features that provide personalized feedback and motivation. In addition, the research intends to compare multiple machine learning algorithms to determine the most suitable approach for detecting dysgraphia in handwritten numbers and, as an alternative approach, to evaluate the detection performance for handwritten letters.

Through these objectives, the research not only addresses the technical challenge of early dysgraphia detection but also contributes toward building a holistic AI based intervention ecosystem that supports children’s learning and emotional well being across Sri Lanka.

1.8 Deliverables and Milestones

1.8.1 Expected Deliverables

This research is expected to produce several key deliverables that collectively contribute to the development and evaluation of a complete dysgraphia detection and intervention framework. The first major deliverable is the creation of a Sri Lankan numeric handwriting dataset, which will be ethically collected, annotated, and pre-processed to ensure high-quality input for machine learning and deep learning analysis. This dataset will represent a unique contribution to the academic community, as it constitutes the first large-scale collection of numeric handwriting samples from Sri Lankan children, including both dysgraphic and non-dysgraphic cases.

The second deliverable is the development of an Advanced Convolutional Neural Network (CNN) model capable of classifying handwriting samples into dysgraphic and non-dysgraphic categories with high accuracy. This model will be trained and validated using the collected dataset and will be benchmarked against other baseline deep learning and machine learning models to ensure superior performance.

The third deliverable involves the implementation of an AI-based screening platform that integrates handwriting upload functionality with a structured questionnaire module. This web-based system will allow teachers, parents, and psychologists to screen children efficiently and objectively for potential signs of dysgraphia, while maintaining user-friendliness and accessibility across multiple devices.

The fourth deliverable is the development of an intervention application designed to support children diagnosed with dysgraphia. The app will incorporate gamified learning exercises, AI-driven chatbots, and progress tracking mechanisms to help children improve their handwriting and fine motor skills through engaging, interactive, and adaptive activities.

Finally, the research will produce a comprehensive evaluation report detailing system performance, accuracy metrics. This report will document the system's real-world applicability and provide insights into the educational and clinical value of the proposed framework, serving as a foundation for future research and expansion into multilingual and multi-disability screening contexts.

1.8.2 Project Milestones & Timeline

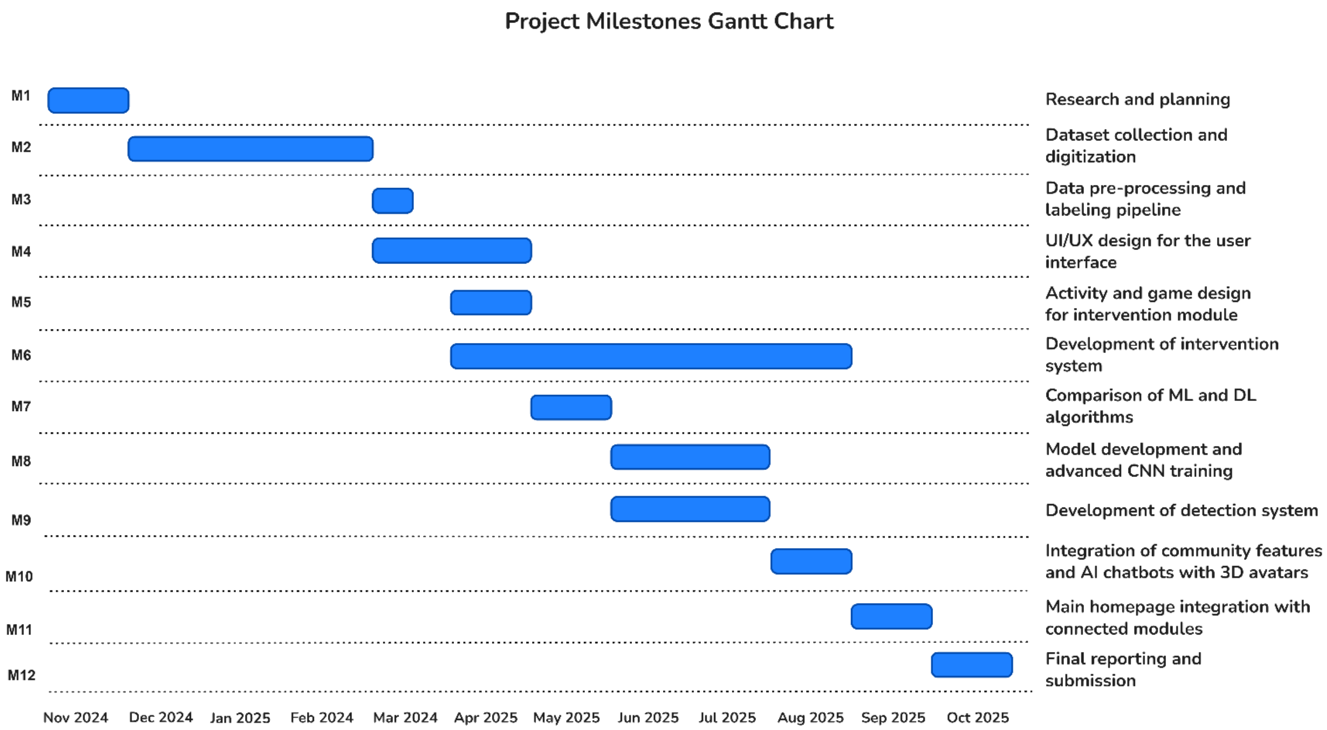


Fig. 1.1 Gantt Chart of the Project

1.9 Research Questions

This study is guided by a series of research questions aimed at exploring the role of artificial intelligence in the early detection and intervention of dysgraphia among children in Sri Lanka. The first research question (R1) investigates how neural networks can be applied to identify dysgraphia patterns from handwriting samples of children, with a focus on analyzing numeric handwriting characteristics. The second research question (R2) examines the unique handwriting features that distinguish children with dysgraphia from their non-dysgraphic peers, focusing on spatial irregularities, stroke consistency, and number formation patterns.

The third research question (R3) explores how AI-based tools can be utilized not only for detection but also for personalized educational intervention, particularly through adaptive feedback, gamified exercises, and progress monitoring. The fourth research question (R4) seeks to determine what specific features and modules should be incorporated into a digital dysgraphia support platform to effectively assist teachers, parents, and therapists in screening and intervention processes. Finally, the fifth research question (R5) addresses how awareness and early screening can be expanded across both urban and rural areas through accessible and affordable technology, ensuring inclusivity in special education support across Sri Lanka.

Together, these questions form the foundation of the research design, guiding the development, implementation, and evaluation of the proposed AI-driven dysgraphia detection and intervention system.

1.10 Significance of the Study

The significance of this study lies in its potential to make contributions across educational, clinical, technological, social, and policy domains. From an educational perspective, the proposed system provides teachers and parents with a practical and user-friendly AI-based tool for early detection of dysgraphia, enabling continuous monitoring and timely intervention. The availability of such a tool reduces reliance on subjective observation and promotes a data-driven approach to supporting children's learning needs.

In the clinical context, the system offers psychologists and occupational therapists an objective digital assessment platform that can complement traditional diagnostic methods. This tool provides quantifiable results that enhance the accuracy and consistency of dysgraphia evaluations.

From a technological standpoint, this research makes a major contribution by introducing Sri Lanka's first annotated numeric dysgraphia dataset, establishing a foundation for future research on AI-based handwriting analysis and learning disorder detection.

The study also holds strong social significance, as it aims to reduce the stigma surrounding learning disabilities by increasing awareness and promoting inclusive education practices. By engaging teachers, parents, and therapists in the diagnostic and intervention process, the platform fosters a supportive environment for children with dysgraphia.

Finally, in terms of policy impact, the findings of this research have the potential to inform the development of national guidelines and frameworks for inclusive education in Sri Lanka. The system's evidence-based outcomes may assist policymakers in designing early intervention programs and integrating AI-based tools into special education curricula.

1.11 Scope of the Study

The scope of this research focuses on the early detection and intervention of dysgraphia among Sri Lankan children aged six to ten years who are enrolled in primary education. The study includes both clinical participants children already diagnosed with dysgraphia and non-clinical participants, to ensure balanced representation across the dataset. Data were collected from over various regions in srilanka such as Anuradhapura, Polonnaruwa, Colombo, matara, Kegalle, Panadura, Nugegoda, Matale to reflecting Sri Lanka's diverse linguistic and cultural backgrounds.

The primary focus of the study is on numeric dysgraphia, specifically analyzing handwritten numbers ranging from zero to ten. The handwriting samples are processed using digital imaging techniques and analyzed through deep learning algorithms, particularly a custom-developed Convolutional Neural Network (CNN).

Beyond detection, the system integrates several intervention components to support children's handwriting improvement and emotional well-being. These include gamified handwriting exercises designed to enhance fine motor coordination, AI chatbots that provide emotional and educational support, and interfaces to ensure accessibility for local users. The platform also supports collaborative access for teachers, therapists, and parents, enabling them to monitor progress and contribute to individualized learning plans.

However, the study's scope is limited to handwriting-based analysis and does not extend to diagnosing complex neurological conditions. The focus remains on the development of a practical, affordable, and contextually appropriate tool for the Sri Lankan educational system.

1.12 Chapter Summary

This chapter introduced the background, motivation, and objectives of the study, establishing the foundation for developing a neural network-based system for the early detection and intervention of dysgraphia among children in Sri Lanka. It discussed the existing challenges in recognizing dysgraphia, the lack of localized datasets, and the potential of artificial intelligence to provide an effective solution. The chapter also outlined the problem statement, proposed solution, research questions, significance, and scope of the study, highlighting the innovative aspects of integrating detection and intervention within a single AI-based framework.

The next chapter, **Literature Review**, will present an in-depth analysis of previous research on handwriting-based learning disability detection, dataset construction methods, and AI and deep learning models used in dysgraphia and related disorders. This review will establish the theoretical and technical foundation upon which the current research is built.

Chapter 2

Related work

2.1 Chapter Overview

The purpose of this chapter is to present an overview of existing research and technological developments relevant to the early detection of dysgraphia and the use of artificial intelligence in handwriting analysis. Reviewing related work is essential to understanding the progress made by previous researchers, identifying limitations in existing methods, and highlighting the unique contributions of the present study.

Dysgraphia detection has been studied from various perspectives including psychological assessment, handwriting pattern analysis, and machine learning based classification. However, despite growing global interest in learning disabilities, research on numeric dysgraphia remains limited, particularly in the context of developing countries such as Sri Lanka. This chapter provides a critical review of literature related to data collection techniques, handwriting datasets, and AI based detection models used in dysgraphia research.

Existing studies on dysgraphia and handwriting analysis can be broadly divided into three categories

- Psychological and clinical assessment studies, which focus on behavioral and neurological indicators.
- Data collection and handwriting dataset studies, which establish the foundation for machine learning and deep learning models.
- AI-based detection and classification studies, which apply computational models to identify handwriting irregularities.

Each of these research streams has contributed valuable insights. however, their integration remains fragmented. Most importantly, the gap between diagnosis and intervention remains largely unaddressed in current literature.

2.3 Data Collection in Dysgraphia Research

2.3.1 Online Data Collection Techniques

Online data collection methods are used to capture handwriting data dynamically as it is produced. These approaches rely on digitizing tablets or stylus-enabled devices to record various real-time parameters such as writing pressure, pen-tip trajectory, pen tilt, and speed of movement. For instance, the use of smart pens such as SensoGrip has enabled researchers to capture temporal and kinematic aspects of handwriting that static images cannot represent.

Online methods are particularly valuable for studying the motor control and temporal patterns of dysgraphic writing, providing quantitative insights into gesture fluidity and rhythm. However, their effectiveness often depends on access to specialized hardware and controlled laboratory environments, which limits their practical application in school or rural contexts.

In addition, the smooth surface of digital tablets can alter natural writing friction, resulting in unnatural stroke variations. Despite these challenges, online handwriting data continues to serve as a valuable source for dynamic feature extraction in advanced neural network models.

2.3.2 Offline Data Collection Techniques

Offline data collection remains the most widely adopted method in dysgraphia research. It involves analyzing completed handwriting samples typically scanned or photographed from paper based writing tasks. This approach provides a cost effective and accessible alternative, allowing for large-scale data collection from schools and clinics without the need for specialized equipment.

Offline handwriting datasets are primarily analyzed using image processing and pattern recognition techniques. Researchers extract static spatial features, such as letter size, stroke smoothness, alignment, and spacing between characters. However, the limitations of offline data include the absence of temporal information and the dependency on consistent image quality during scanning or photography.

Nevertheless, for contexts such as Sri Lanka, where access to advanced digitizers is limited, offline data collection offers a practical and scalable approach. The present study adopts this method to ensure inclusivity and feasibility in gathering samples from both urban and rural regions.

2.4 Comparative Analysis of Existing Handwriting Datasets

A critical component of any AI-based dysgraphia detection model is the availability of high quality, annotated handwriting datasets. Table 2.1 summarizes existing datasets used in previous research and compares them with the proposed Sri Lankan dataset.

Table 2.1 A Summary of Handwriting Data Collection type and Methods used in Literature

Research Study	**Data Collection type and Methods**			**Data Collection Process**
	Country	**Dataset**	**Data Collection method**	
[7]	Malaysia	Sentence	Offline	Handwritten text was manually converted from physical paper to digital images by scanning. The dataset includes sentences written by primary school students and children under intervention.
[8]	Sri Lanka	City names	Offline	Images of handwritten city names were obtained from live mail addresses, around 500 city names
[9]	Malaysia	Letters and numbers	Offline	Scanned handwriting images were collected from dyslexic children. The dataset consists of 8 selected lowercase letters (b, c, f, p) and numbers (2, 5, 6, 7)
[10]	Malaysia	Letters, Words, Sentence	Online	used smart pen called SensoGrip, a pen equipped with sensors to capture handwriting dynamics
[11]	Sri Lanka	Letters and numbers, audio clips	offline	primary age children between the age of 6 to 7 and **Dataset development unspecified.**
Proposed System	Sri Lanka	Numbers	Offline	Numeric Handwriting from 482 children (numbers 0 to 10) age 6 to 10 children in **diverse regions in Sri Lanka**

Existing datasets such as MNIST and EMNIST provide foundational resources for generic handwriting recognition but are not designed for dysgraphia detection or localized writing patterns. In contrast, the proposed Sri Lankan dataset introduces culturally and linguistically relevant handwriting samples, representing both clinical and non-clinical participants.

Moreover, while prior studies focused predominantly on alphabetic letters or sentence writing, this research emphasizes numeric handwriting, addressing an overlooked domain numeric dysgraphia. The dataset also incorporates a carefully designed preprocessing pipeline, including resizing (300×300 → 32×32), grayscale conversion, Gaussian blurring, adaptive thresholding, morphological operations, and noise removal, achieving an estimated 97% noise reduction rate.

2.5 AI and Machine Learning Approaches in Dysgraphia Detection

Artificial intelligence and machine learning have been widely used in recent years for handwriting analysis and dysgraphia detection. Techniques such as Support Vector Machines (SVM), Random Forests (RF), K-Nearest Neighbors (KNN), and Logistic Regression have been applied to classify handwriting features extracted from digital images.

However, traditional machine learning approaches often rely on manual feature extraction and limited generalization capability. As a result, researchers have turned toward deep learning techniques, especially Convolutional Neural Networks (CNNs), which automatically learn spatial hierarchies of handwriting patterns.

Several studies have demonstrated the potential of CNNs in distinguishing dysgraphic handwriting from normal samples with high accuracy. Nevertheless, most of these models were trained on Western datasets, making them less effective when applied to culturally and linguistically distinct handwriting styles.

Recent developments in Transfer Learning (e.g., using VGG16, ResNet50) and Transformer architectures (e.g., Vision Transformer and Swin Transformer) have improved feature learning capabilities, but their performance depends heavily on dataset scale and quality. In low-resource settings, these models often underperform due to limited data diversity and overfitting. The present study addresses these limitations by designing an Advanced CNN architecture specifically optimized for small, noise-reduced datasets.

2.6 Gaps Identified in Previous Research

Despite significant advances in AI-based handwriting analysis, several gaps persist in the literature.

First, there is a lack of localized handwriting datasets that represent Sri Lankan children's numeric and alphabetic writing patterns. Most existing resources are based on Western or East Asian samples, which differ in script formation and pen-handling behavior.
Second, research has primarily focused on alphabetic dysgraphia, leaving numeric dysgraphia largely unexplored, even though number writing is equally essential in early education.

Third, while several studies have proposed detection systems, very few have integrated intervention tools. The absence of bilingual, gamified, and child-centered applications limits the practical usability of such systems for parents and teachers in real world educational contexts.

The proposed research addresses these gaps by creating a comprehensive numeric dysgraphia dataset, developing an Advanced CNN detection model, and integrating a multi-platform intervention system that includes gamified exercises and an AI chatbot.

2.7 Chapter Summary

This chapter reviewed the key literature and existing works related to dysgraphia detection, handwriting datasets, and AI-based classification systems. It highlighted the differences between online and offline data collection techniques, compared existing datasets across different countries, and discussed the role of machine learning and deep learning methods in identifying handwriting irregularities.

The review revealed that current approaches are limited by the lack of localized datasets, minimal focus on numeric dysgraphia, and inadequate integration of screening and intervention tools. Building on these findings, the next chapter outlines the research methodology, detailing the data collection procedures, preprocessing pipeline, model architecture, and system design employed in this study.

Chapter 3

Methodology

3.1 Chapter Overview

This chapter presents the detailed methodology used to design, develop, and validate the proposed Neural Network-Based System for Early Detection and Intervention of Dysgraphia in Children. The methodology forms the foundation of the research, outlining each technical and procedural step that connects the research objectives to tangible outcomes.

The approach follows a structured engineering oriented framework encompassing data collection, dataset development, image preprocessing, model training, and web-based platform implementation. Each phase of the methodology was guided by the principles of reproducibility, ethical compliance, and scientific rigor. The overall research design integrates machine learning experimentation with human-centered system development to produce both a robust AI model and a usable intervention tool.

3.2 Research Design

The research adopted a quantitative experimental design supported by comparative model evaluation and system prototyping. Figure 3.1 illustrates the high-level flow of the study, beginning with stakeholder surveys, followed by dataset creation, model training, and culminating in application deployment.

At the initial stage, surveys were conducted among parents and teachers from 20 districts to assess awareness of dysgraphia and determine the real-world need for early-screening solutions. Insights from this stage informed the dataset requirements and the design of user-centric application modules.

The second stage involved data collection, where handwriting samples of numbers (0–10) were obtained from Sri Lankan children aged 6 to 10 years, including both dysgraphic and non-dysgraphic participants. The dataset construction phase required adherence to strict ethical guidelines and ensured demographic diversity across geographical regions.

Subsequent stages focused on image preprocessing and AI model development. The collected handwriting images were digitized, cleaned, and standardized using computer-vision techniques before being fed into multiple machine-learning and deep-learning models. The models were evaluated comparatively to identify the most accurate architecture.

Finally, the trained neural-network model was embedded within a web-based dysgraphia intervention platform, featuring bilingual interaction, gamified learning, and progress-tracking dashboards. This multi-layered workflow ensured that the research addressed not only detection accuracy but also educational usability and accessibility.

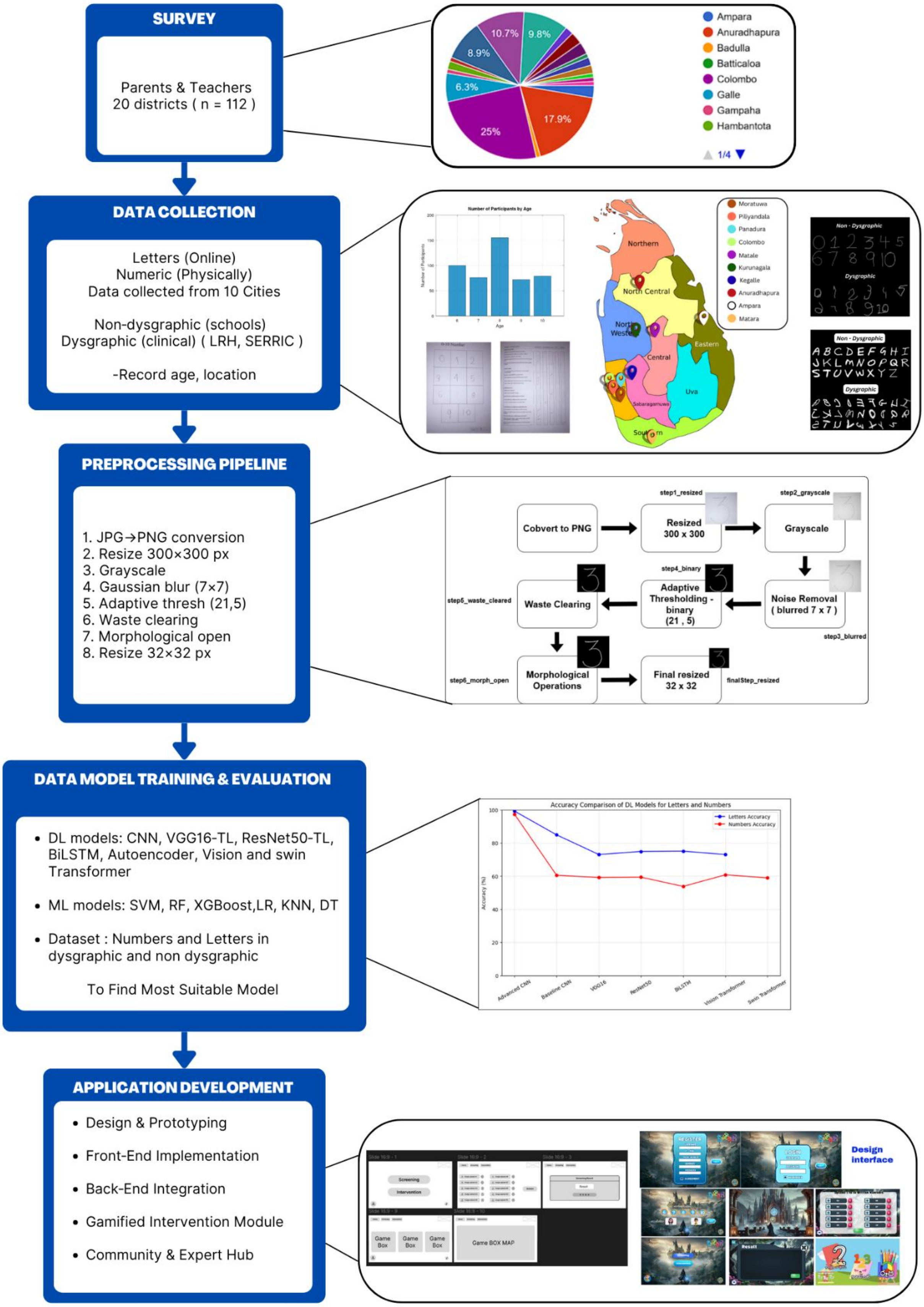


Fig. 3.1 Research Methodology Flow Diagram

3.3 Survey Design

The survey component of this research was designed with the primary purpose of assessing the level of awareness, understanding, and perceptions regarding dysgraphia and other specific learning disabilities among parents and teachers across Sri Lanka. Since dysgraphia remains a lesser-known condition in the country, the survey aimed to identify the extent to which key stakeholders in children's education particularly primary school teachers and parents recognize its symptoms, causes, and impact on children's learning and emotional well-being. The data collected through this survey provided essential contextual insights that informed the development of the proposed AI-based screening and intervention platform, ensuring that it addresses the real educational and social gaps identified in the local environment.

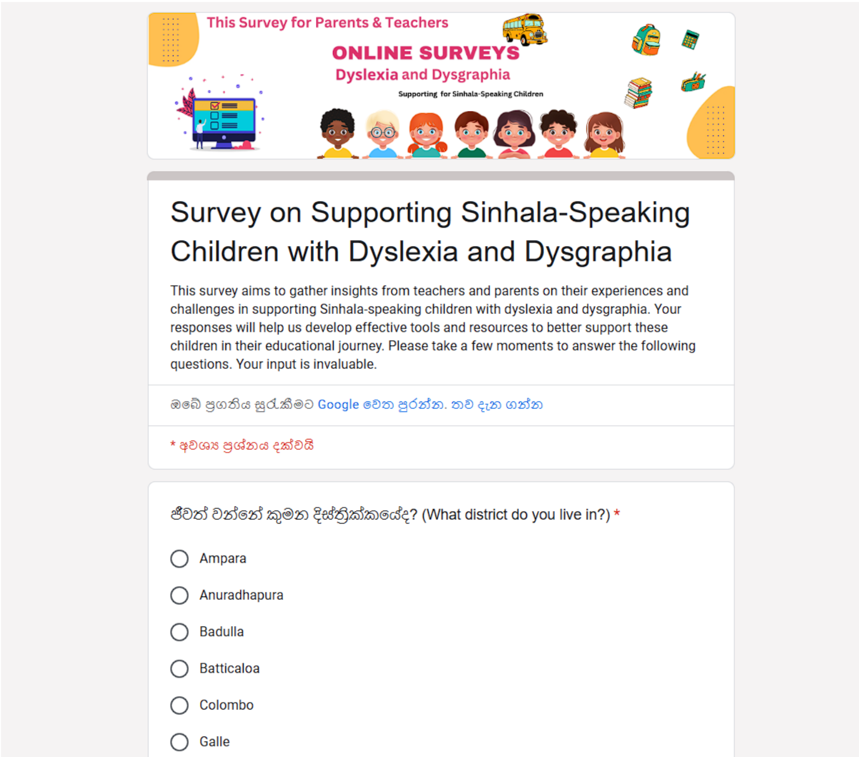


Fig. 3.2 Shared Survey

The survey was conducted online using structured questionnaires distributed through Google Forms to participants representing 20 districts across Sri Lanka. It included both quantitative and qualitative questions, covering areas such as general awareness of learning disabilities, familiarity with dysgraphia-specific indicators, access to professional diagnosis, and the perceived need for digital tools in early screening. Responses were collected from teachers and parents of children aged between 6 and 12 years, ensuring diversity in educational backgrounds and geographical representation. Ethical guidelines were strictly followed, including voluntary participation, anonymity of responses, and informed consent. The survey findings revealed significant variability in awareness levels, highlighting a general lack of knowledge about dysgraphia compared to other learning disorders such as dyslexia or ADHD. These findings underscored the necessity of developing localized digital solutions that not only detect dysgraphia but also raise awareness and provide ongoing support to children, parents, and educators across Sri Lanka.

3.4 Data Collection Process

The dataset creation phase represented one of the most critical components of this research, serving as the foundation upon which the entire model training and evaluation process was built. The primary target group consisted of children between the ages of six and ten, a period recognized as crucial for developing fine motor skills and handwriting proficiency. Data collection was carried out using a mixed-method approach that incorporated both clinical and educational environments to ensure that the dataset was diverse and representative.

Ethical clearance was obtained from relevant educational and medical authorities before data collection commenced, ensuring that all research activities adhered to ethical guidelines for working with minors. Informed consent was obtained from parents or guardians, and assent was acquired from each participating child after they were clearly informed about the purpose and procedures of the study. The participant pool was divided into two distinct categories: children who had been clinically diagnosed with numeric dysgraphia, recruited from the Senehasa Education Resource Research and Information Centre (SERRIC) and Lady Ridgeway Hospital, and children without dysgraphia, sampled from various primary schools across both urban and rural regions.

Each participant was asked to write the numbers zero to ten on standardized handwriting sheets under supervised conditions to maintain uniformity across samples. To achieve regional balance, samples were collected from multiple districts across Sri Lanka, including Ampara, Anuradhapura, Colombo, Panadura, Piliyandala, Moratuwa, Kurunegala, Matara, Matale, and Kegalle. In total, 482 participants contributed approximately 8,000 handwritten samples, consisting of 122 children with numeric dysgraphia and 360 children without the condition. These collected handwriting sheets, later digitized for analysis, formed the first comprehensive Sri Lankan numeric dysgraphia dataset.

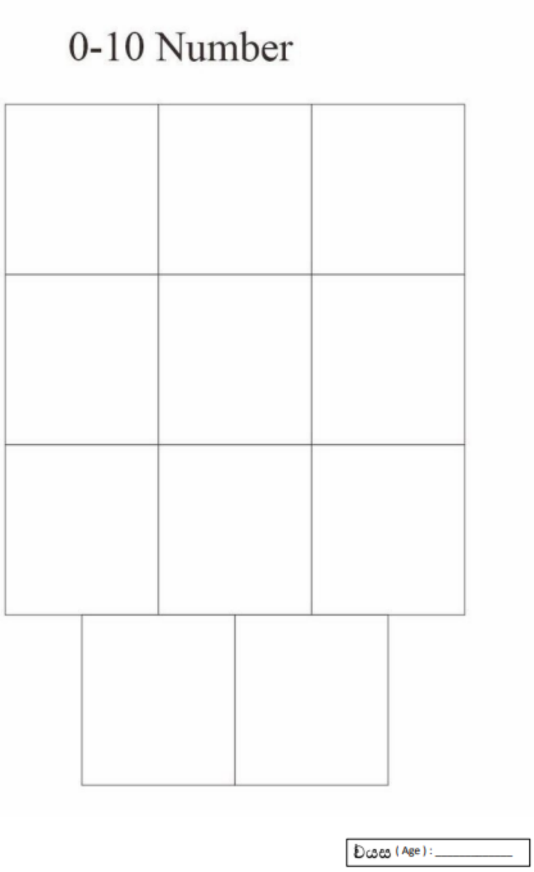


Fig. 3.3 Handwriting Collected Sheet

කොටුවේ "✓" හරි ලකුණ යොදන්න	කවදාවත් නැහැ	කලාතුරකින්	සමහර වෙලාවට	නිතර	හැම විටම
අකුරු සහ අංක ලිවීම					
අවුල් සහගත හෝ කියවීමට අපහසු අත් අකුරු තිබේද?					
සංඛ්‍යා ආපසු හරවා ලියා තිබේද (උදා: 18 වෙනුවට 81 ලිවීම)?					
අකුරු පිටුපසට (ආපසු හරවා) ලියා තිබේද?					
ඉලක්කම් පැහැදිලිව ලිවීමට සහ ඒවා අකුරුවලින් වෙනස් කිරීමට අපහසු වේද?					
සටහන් පොතක හෝ ඉරි ඇඳ ඇති කඩදාසියක ඔබට රේඛා තුළ ලිවීමට අපහසුද?					
අක්ෂර සැකසීම					
වචන සහ/හෝ වචන අතර අකුරු නිවැරදිව පරතරයක් තැබීමේ ගැටලුවක් තිබේද?					
ඔබ අකුරු හෝ ඉලක්කම් විශාල ලෙස ලියනවාද?					
විරාම ලකුණු සමඟ වාක්‍ය අවසන් කිරීමේ ගැටලුවක් තිබේද?					
ගණිත ගැටලුවක් කරන විට තීරු වල පහලට පහලට ලිවීමට අපහසුද?					
වමේ සිට දකුණට අකුරු ලිවීමේ අපහසුවක් තිබේද?					
ලිවීමේ ප්‍රවීණතාවය					
ලිවීමේ කාර්යයට විරුද්ධද?					
සිතුවිලි කඩදාසි මත ලිවීමට අපහසුද?					
අකුරු නිවැරදිව බලා ලිවීමේ අපහසුතාවයක් තිබේද?					
තනිවම ලිවීමේදී වැඩ සම්පූර්ණ කිරීමේ ගැටලුවක් තිබේද?					
ගෙදර වැඩ පැවරුම්වල අක්ෂර වින්‍යාස දෝෂ තිබේද ?					
චාලක කුසලතා සහ සියුම් චාලක සම්බන්ධීකරණය					
ඔබට පැන්සලක් හෝ පෑනක් අල්ලාගෙන සිටීමට අපහසුද ?					
ලිවීමෙන් හෝ ඇඳීමෙන් ඔබේ අත රිදෙනවාද නැත්නම් වෙහෙසට පත් වෙනවාද?					
ඔබට කුඩා වස්තු අහුලා ගැනීමේ අපහසුතාවයක් තිබේද ?					
ඔබට කතුර භාවිතා කිරීම දී , ඇඳුම් බොත්තම් දැමීම දී හෝ සිපර් භාවිතා කිරීමේ දී ගැටලුවක් තිබේද?					
ඔබේ අත අනෙක් අත සමග එකිනෙකට එක වගේ වැඩ කිරීමෙන් සිදෙනවාද ?					

Fig. 3.4 Assessment identifies Not have dysgraphia or numeric dysgraphia

The selection of numbers was based on their relevance to assessing handwriting difficulties specific to numeric dysgraphia. This Dataset Collect sheets shows in Figure 3.3 and Figure 3.4 that sheet we used to collect handwriting. when we collected handwriting in Dysgraphia diagnosed and who have numeric dysgraphia, we use only Figure 3.3 sheet. in other hand when non numeric dysgraphia collected including assessment (Figure 3.4), we collecting using Figure 3.3 and Figure 3.4 sheet when collecting non numeric dysgraphia to identify if they do not have dysgraphia or numeric symptoms, then specialist (occupational therapist, psychologist, special education teachers) overlook assessment and handwriting numbers categorized as a non-numeric dysgraphia. This dataset aims to provide a comprehensive resource for understanding and addressing numeric dysgraphia in Sri Lanka.

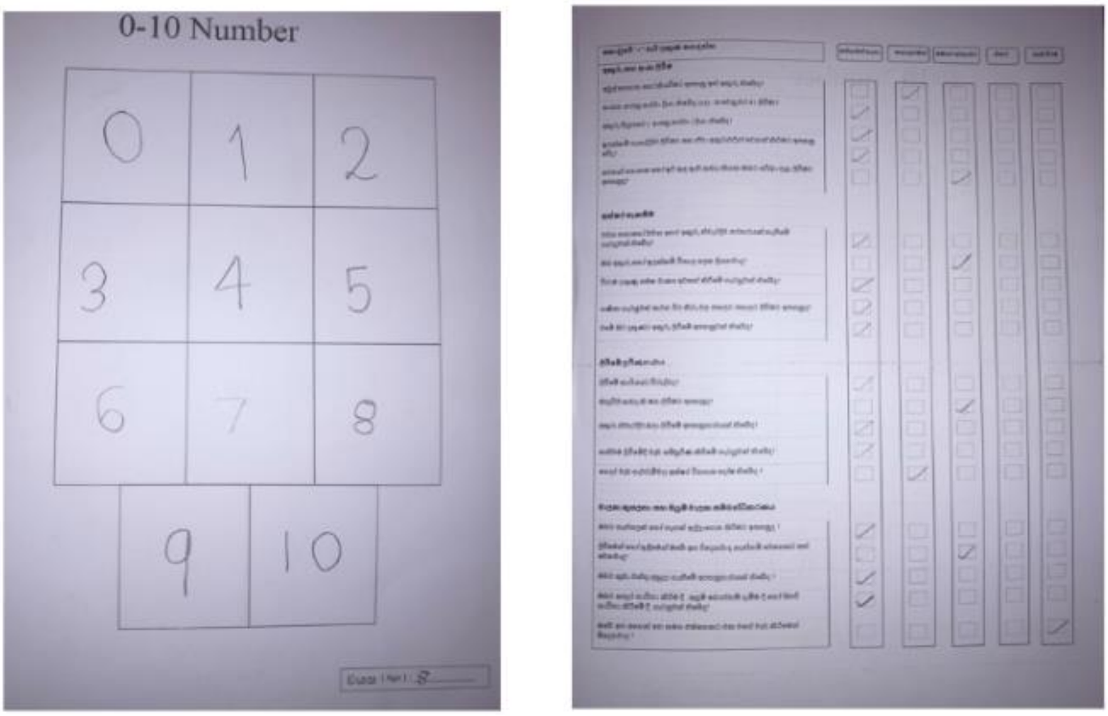


Fig. 3.5 Handwriting numbers sheet and assessment

This Figure 3.5 show collected Number sheet and assessment as a example of how we collected in dataset, when collecting dysgraphia dataset, we only collected numbers sheet because of they already diagnosed.

3.5 Image Digitization and Preprocessing

In the digitization process for the numeric dysgraphia dataset, cameras were used to capture images of handwritten numbers (0-10), ensuring consistent lighting and camera angles to minimize variations and shadows that could compromise image quality. The camera's high resolution effectively captured intricate details such as stroke width, pressure variations, with precise focus adjustments to produce sharp and legible images.

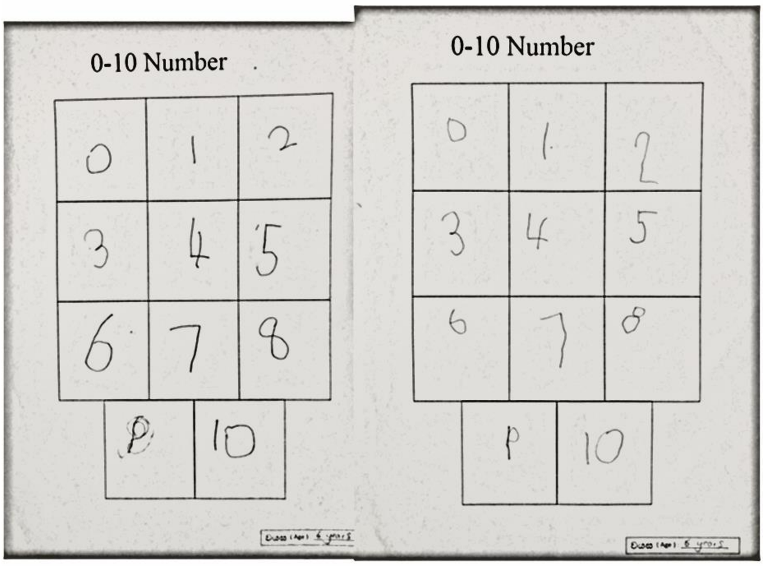


Fig. 3.6 Numeric Dysgraphia Handwriting collected Sheets

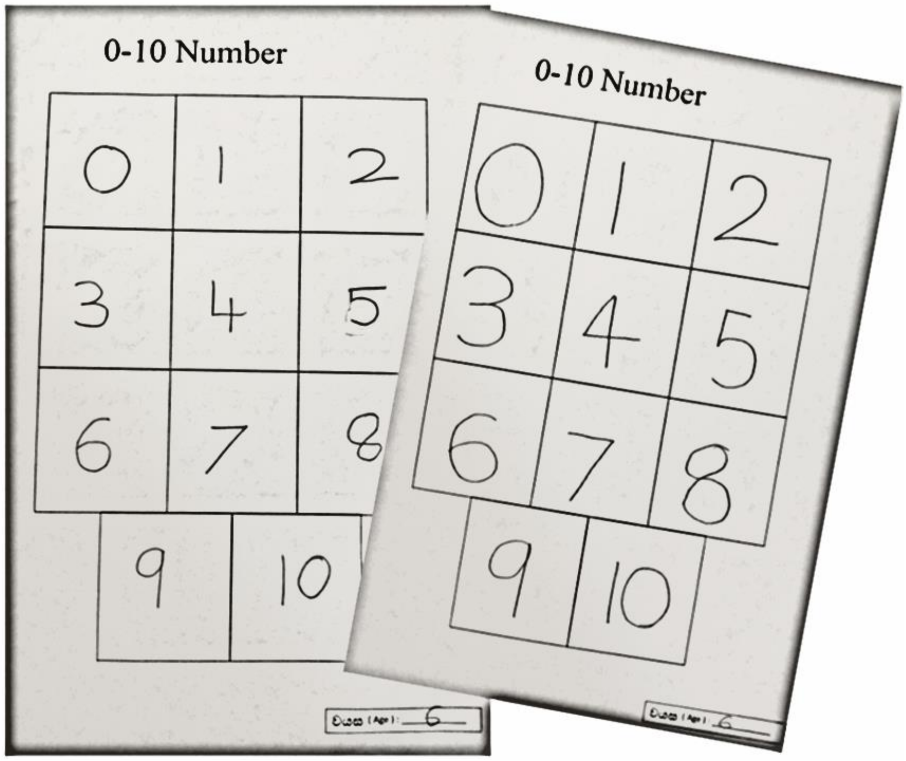


Fig. 3.7 Non-Numeric Dysgraphia Handwriting collected Sheets

The images were then systematically stored, ensuring organized access and retrieval for subsequent preprocessing and analysis stages. Unlike word-based datasets, this approach focused on capturing numbers, as the goal was to analyze the overall handwriting number writing patterns indicative of numeric dysgraphia. This method avoided potential complication.

To prepare handwriting data for neural-network training, a multi-stage preprocessing pipeline was developed. Figure 3.8 illustrates this process.

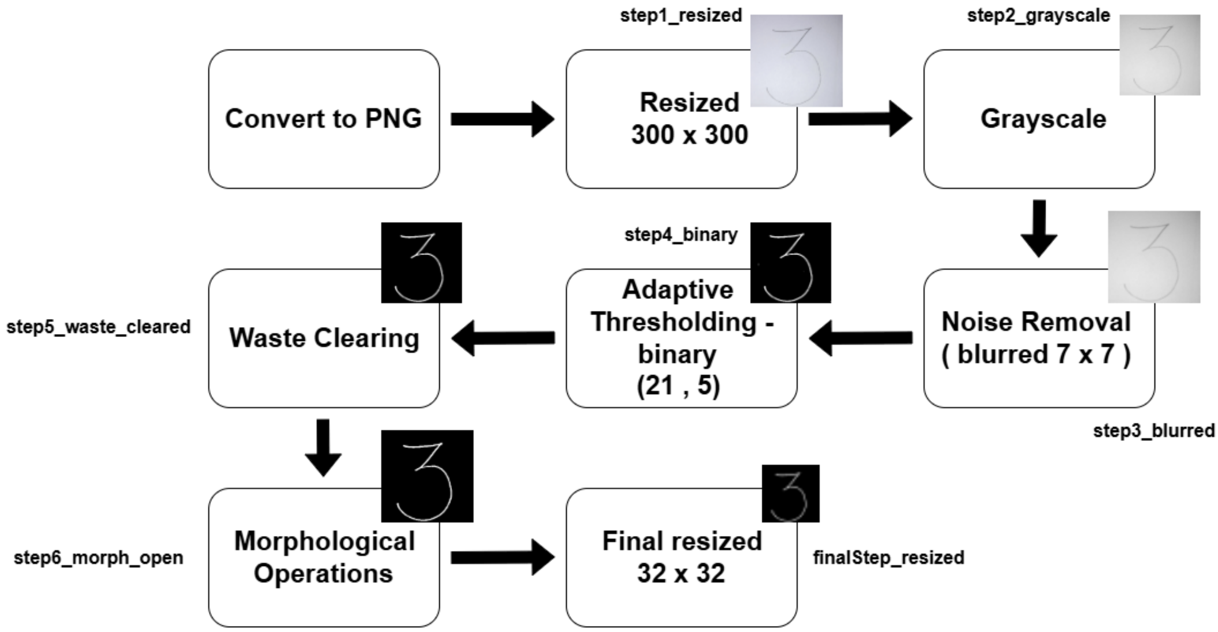


Fig. 3.8 Image Preprocessing Pipeline (Self-composed)

The digitization and preprocessing stage played a vital role in ensuring that the handwritten numeric samples were clean, consistent, and suitable for deep learning model training. The process began with the digitization of physical handwriting sheets using a high-resolution DSLR camera under controlled lighting conditions to minimize shadows, glare, and other visual distortions. Once captured, the images were converted from JPG to PNG format to preserve fine details and avoid compression artifacts that could distort handwriting patterns.

Each image was then resized to a standardized dimension of 300×300 pixels to maintain uniformity across all samples and later reduced to 32×32 pixels for efficient neural network processing. To simplify the data while retaining essential handwriting features, images were converted to grayscale using a luminance-based transformation that enhanced structural contrasts between strokes and the background.

The next step involved noise reduction using a Gaussian Blur filter with a 7×7 kernel, effectively smoothing intensity variations and removing unwanted scanning or digitization artifacts. Following this, adaptive thresholding was applied to separate the foreground handwriting from the background dynamically, compensating for uneven lighting or shading differences across samples. To further refine image quality, morphological operations such as erosion and dilation were used to enhance stroke continuity and eliminate residual specks or irregularities. Finally, waste clearing was performed using connected-component analysis to remove any remaining small artifacts or irrelevant noise elements below a certain area threshold.

Through this structured preprocessing pipeline, the images were transformed into clean, uniform inputs, achieving a 97% noise reduction rate and ensuring that the model learned from precise and high-quality handwriting data.

The image preprocessing process involved a series of steps designed to optimize the quality and consistency of handwritten numeric images. Let I be an input image. Initially, all images were converted to PNG format to ensure high-quality preservation. The images were then resized to a uniform dimension of 300x300 pixels to standardize images size [12].

$$I'(x',y') = \sum_{x=0}^{N=1}\sum_{y=0}^{M=1} I(x,y)\cdot W(x,x')\cdot W(y,y') \qquad (1)$$

Where Eq. (1), I (x, y) is the original image and $I'(x',y')$ is the resized image, $W(x,x')$ *and* $W(y,y')$ are interpolation weights.

Next, they were converted to grayscale, which simplified the images by removing color information while retaining essential structural details. Conversion to grayscale uses the luminance formula,

$$I_{gray}(\mathrm{x},\mathrm{y}) = 0.2989\cdot R(\mathrm{x},\mathrm{y}) + 0.5870\cdot G(\mathrm{x},\mathrm{y}) + 0.1140\cdot B(\mathrm{x},\mathrm{y}) \quad (2)$$

where Eq. (2), R,G,B are the red, green, and blue pixel values of the original image at coordinates (x, y).

To reduce noise and enhance clarity, Gaussian Blur was applied [12].

$$I_{blurred}(x,y) = \sum_{i=-k}^{k}\sum_{j=-k}^{k} G(i,j)\,.\,I_{gray}(x+i,y+j) \qquad (3)$$

Convolution of the image with the Gaussian kernel smooths the image, Eq. (3), $I_{Blurred} = G(i,j) * I_{Gray}$, where G(i, j) is the Gaussian kernel.
Adaptive thresholding was used to convert the grayscale images into binary format, effectively isolating the handwriting against a black background and adjusting for lighting variations.

$$T(x,y) = \frac{1}{N}\sum_{(i,j)\in block} I(i,j) - C \qquad (4)$$

Adaptive Mean Thresholding computes a local threshold for a pixel. Where, Eq. (4), Threshold Calculation, N is the number of pixels in the block, C is a constant to fine-tune the threshold.
Connected components analysis was performed to remove small, irrelevant objects, Removes small components based on size. Identify connected components,

(num_labels,labels,stats,centroids)=cv2.connectedComponentsWithStats(I_{binary},connectivity=8)
(5)

Where , Eq. (5), num_labels is the number of connected components found, labels is the label matrix for each component, and stats contains statistics for each component, then Clear small components,

$$I_{\text{cleaned}}(x,y) \begin{cases} 255 & if\ stats[l, cv2.CC\_STAT\_AREA] \geq min\_size \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

Where, Eq. (6), min_size is a predefined threshold for the minimum area of components to retain, Waste clearing typically involves connected component analysis to remove small noise components from a binary image [13].
And morphological operations were applied to refine the binary images.

$$I_{Opened} = (I_{binary} \oplus B) \ominus B \qquad (7)$$

Where, Eq. (7), B is a structuring element, $\oplus$ denotes dilation, and $\ominus$ denotes erosion. Removing missing small noise from waste clearing and irrelevant white regions, Keeping the structure of larger objects intact [12].
If necessary, global thresholding was used as a fallback to ensure proper binarization. Global Threshold with Otsu's Method,

$$\sigma^2(T) = w_0(T) \cdot \sigma_0^2(T) + w_1(T) \cdot \sigma_1^2(T) \qquad (8)$$

Where, Eq. (7), w_0, w_1 are weights of the two classes and σ_0^2, σ_1^2 are variances of the two classes, this minimizes intra-class variance, If the processed binary image is blank (all black), global thresholding is applied. global thresholding using Otsu's method ensures meaningful binarization by analyzing the overall intensity distribution.

Finally, the pre-processed images were resized to 32x32 pixels, ensuring consistency for model input. These steps ensured that the images were clean, consistent, and focused on the features most relevant for numeric dysgraphia detection.

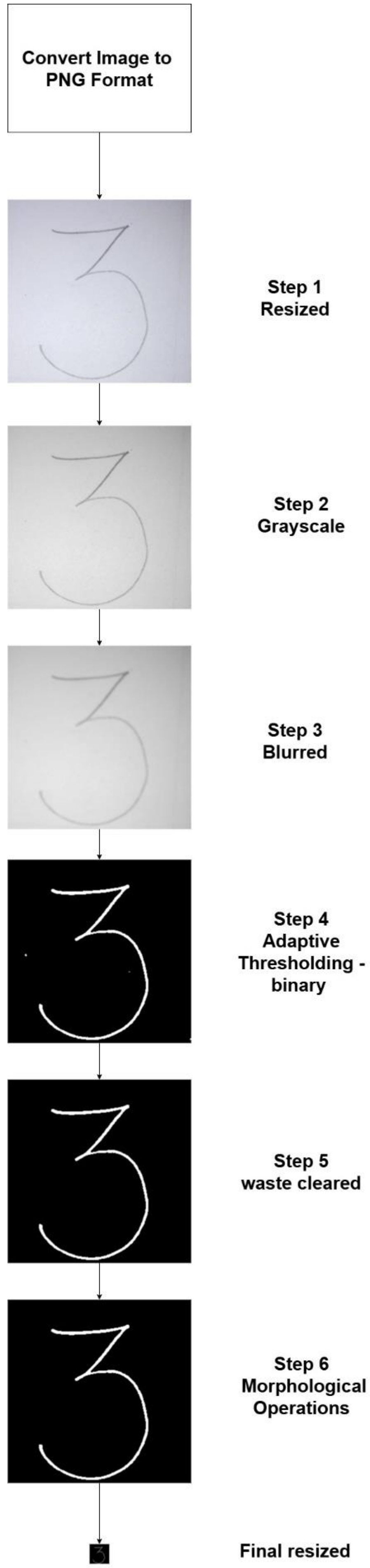


Fig. 3.9 A Sample Image Preprocessing Steps (Self-composed)

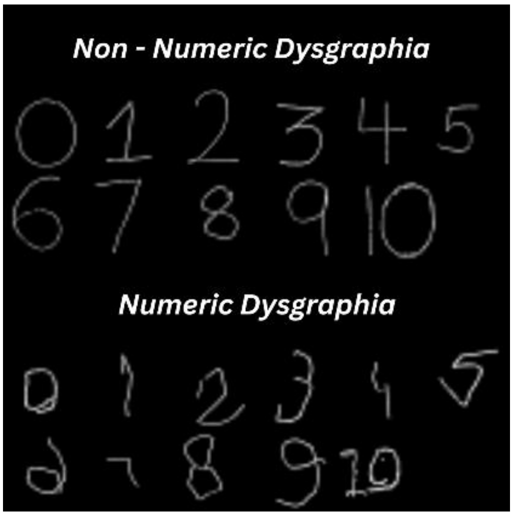


Fig. 3.10 Using a Sample Images showing difference in Numeric and Non numeric dysgraphia

3.6 Data Labeling and Organization

Handwriting images of numbers (0–10) were labelled as 1 (numeric dysgraphia) or 0 (non-numeric dysgraphia). Images were retrieved from folders, paired with labels, and organized into a shuffled data frame to avoid bias. The dataset was saved as a CSV file containing image paths and labels, enabling effective mapping and training of models to distinguish between the two classes. Python automation scripts indexed file paths, assigned corresponding labels, and compiled them into a CSV metadata file.

Figure 3.11 shows the participant distribution by age, representation across age groups.

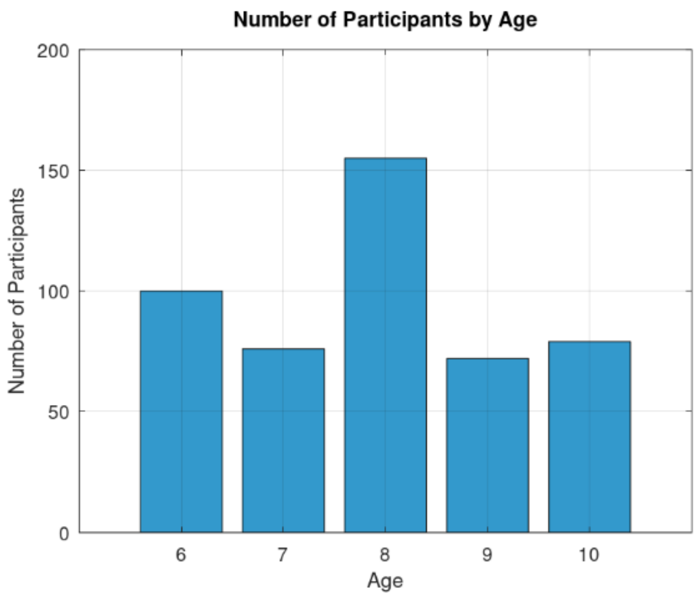


Fig. 3.11 Distribution of Participants by Age

This study looked at demographic variables including age and region in addition to collecting samples of numeric handwriting. This study participants count by their shown in Figure 3.11 Children between the ages of 6 and 10 made up the participants [14] and This labeled dataset became the foundational input for all subsequent machine-learning and deep-learning experiments.

3.7 Model Development and Training

3.7.1 Overview

The model development process formed the technical foundation of this research, aiming to design a neural network capable of identifying dysgraphic handwriting patterns with high accuracy. This stage began with the implementation of baseline machine learning and deep learning models to evaluate their suitability for numeric dysgraphia detection. A series of experiments were conducted to identify the architecture that would perform best in classifying dysgraphic and non-dysgraphic samples based on structural and spatial handwriting features. Both classical approaches and state-of-the-art deep learning methods were compared to ensure a robust and fair evaluation process.

3.7.2 Baseline Models

To establish performance benchmarks, several traditional machine learning models were first developed and evaluated. These included Support Vector Machines (SVM), Random Forest (RF), XGBoost, Logistic Regression (LR), Decision Trees (DT), and K-Nearest Neighbor (KNN). Each of these models relied on manually extracted features such as stroke density, curvature, aspect ratio, and pixel distribution, which were derived from the preprocessed numeric handwriting images. While these algorithms provided useful baseline results, they were limited by their dependence on handcrafted features that could not capture the subtle visual and spatial variations typically observed in dysgraphic handwriting. Consequently, a transition to deep learning approaches was necessary to leverage automatic feature extraction and hierarchical representation learning.

3.7.3 Deep Learning Architectures

In the next phase, several deep learning architectures were explored to assess their ability to generalize on the Sri Lankan numeric dysgraphia dataset. Transfer learning models such as VGG16 and ResNet50 were fine-tuned to learn from local handwriting patterns, while recurrent models like BiLSTM were used to capture sequential dependencies in the writing strokes. Additionally, Autoencoders were implemented for unsupervised representation learning, aiming to identify latent features in dysgraphic writing styles. Transformer-based architectures, including the Vision Transformer (ViT) and Swin Transformer, were also tested to evaluate their ability to model long-range dependencies and global spatial relationships within the handwriting images. Despite their theoretical advantages, transformer models showed limited performance on the relatively small dataset due to their higher parameter count and data requirements.

3.7.4 Proposed Advanced CNN Architecture

After evaluating multiple architectures, the final model selected for deployment was the Advanced Convolutional Neural Network (CNN), which was custom-designed to address the challenges of small-sample handwriting classification. The architecture comprised multiple convolutional layers with ReLU activation and 3×3 kernels to capture fine-grained spatial details, followed by max-pooling layers for down-sampling and batch normalization for stability. Dropout layers with rates between 0.3 and 0.5 were strategically included to prevent overfitting. The flattened output was passed through fully connected dense layers with 128 neurons before reaching a softmax output layer for binary classification. The model was trained using the Adam optimizer with a learning rate of 0.0005, a batch size of 32, and categorical cross-entropy as the loss function over 100 epochs in numeric dysgraphia and 30 epochs in letter dysgraphia . Data augmentation techniques such as rotation, zoom, and horizontal shifting were applied to improve generalization and prevent bias toward particular handwriting styles. The model was validated using an 80–20 train-test split, complemented by five-fold cross-validation to ensure reliability.

Performance metrics such as accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) were employed to evaluate model performance. These measures allowed for a comprehensive assessment of both the detection accuracy and the discriminative power of the models. The proposed Advanced CNN demonstrated superior learning efficiency and robustness compared to all baseline models, establishing itself as the optimal architecture for numeric dysgraphia detection. The detailed performance results, comparative evaluation, and graphical analysis of accuracy trends are discussed in Chapter 5.

3.8 Application Development and Integration

Following the successful development and training of the Advanced CNN model, the next step was the integration of this model into a comprehensive web-based dysgraphia screening and intervention platform. This application was designed to transform the trained AI model into a practical, user-accessible tool that could be deployed in real educational and clinical settings. The goal was to create an ecosystem that not only identifies dysgraphia but also provides continuous support through personalized learning interventions, gamified exercises, and bilingual interaction features.

The system architecture was structured into two major components: the frontend and the backend. The frontend was developed using the React.js framework to ensure an interactive and dynamic user experience. It provides separate access portals for children, parents, teachers, and psychologists, allowing each stakeholder to perform their respective roles effectively. The child interface includes modules for handwriting sample uploads, interactive learning games, and progress tracking, while the teacher and therapist dashboards provide analytical insights into student performance. State management was handled through Redux, ensuring smooth synchronization of real-time updates across all interface components.

The backend, built on the Django REST Framework, serves as the core of the system by managing user authentication, data exchange, and model inference. When a handwriting image is uploaded, it is processed through an API Gateway that communicates with the AI inference engine. The pre-trained CNN model is then loaded, and the system classifies the handwriting as dysgraphic or non-dysgraphic within milliseconds. The results are stored in a PostgreSQL database, along with anonymized metadata such as user age, region, and session statistics. This backend design ensures data privacy, scalability, and efficient communication between the model and the user interfaces.

An additional feature of the platform is the Gamified Learning Engine, which dynamically adjusts difficulty levels based on the child's progress. The system provides writing exercises and visual-motor coordination games designed to strengthen the skills of children diagnosed with dysgraphia. Furthermore, an integrated AI-powered Chatbot, utilizing natural language processing, provides bilingual (Sinhala and English) guidance, emotional support, and learning feedback to children and parents.

This application functions not only as a diagnostic system but also as a digital rehabilitation environment, fostering collaboration among children, parents, teachers, and psychologists. The implementation of such an inclusive framework aligns with the goal of creating a sustainable, accessible, and community-driven solution for addressing dysgraphia in Sri Lanka.

Figure 3.12 presents snapshots of the user interface and gamified learning modules.

Fig. 3.12 Application Interface Screens

This application not only supports early screening but also fosters ongoing learning and community engagement among educators, therapists, and families.

3.9 Ethical Considerations

Given that the study involved minors, ethical integrity was a top priority. All procedures complied with the National Ethics Guidelines of Sri Lanka. Informed consent was obtained from parents, and assent from children. Personal identifiers were removed from datasets and Data storage followed institutional data-protection policies, all digital files were stored on encrypted drives.

These safeguards ensured participant welfare, data confidentiality, and compliance with both educational and clinical research standards.

3.10 Chapter Summary

This chapter described, in detail, the methodological framework of the research. It outlined each procedural stage from survey and data collection to image preprocessing, model training, and system deployment. The methodology integrated empirical experimentation with ethical practice to ensure validity and societal relevance.

The next chapter, Experimental Setup and Implementation, elaborates on the technical environment, model deployment, system interface development, and integration workflow.

Chapter 4

Experimental Setup and Implementation

4.1 Chapter Overview

This chapter describes in detail the experimental setup and implementation process of the proposed system a Neural Network-Based System for Early Detection and Intervention of Dysgraphia in Children. While Chapter 3 explained the conceptual methodology and data collection framework, this chapter focuses on the technical realization of the system, model implementation, and overall application development process. The primary goal of this chapter is to present how the system's components, architectures, models, and datasets were integrated to form a cohesive end-to-end solution capable of both detecting dysgraphia and delivering interactive interventions.

This chapter begins with an overview of the system's goals and design philosophy, followed by a detailed explanation of the architectural framework that underpins the solution. It also outlines the technologies employed in various tiers, the dataset selection and preprocessing methods used for both numeric and letter dysgraphia detection, and the development of the core deep learning models. The chapter concludes with the implementation of the core functionalities, describing how different services ranging from AI detection to educational gamification and community interaction were realized within the platform.

4.2 System Goals

The design goals of the proposed system were formulated to ensure that the solution effectively bridges the gap between early diagnosis, personalized intervention, and awareness in the context of dysgraphia. The system was conceptualized as an end to end platform that integrates artificial intelligence, deep learning, and human centered design to create a holistic approach toward identifying and supporting children with handwriting difficulties. Each design goal focuses on a distinct but interdependent aspect of the solution from the development of robust neural network models for dysgraphia detection to the creation of an immersive and emotionally engaging intervention environment. The design also prioritizes accessibility, inclusivity, and ethical handling of sensitive child data, ensuring that the system remains both technologically advanced and socially responsible.

These design goals collectively guide the system's architecture, development, and implementation, as summarized in Table 4.1.

Table 4.1 Design Goals of the Proposed System

Design Goal	Description
AI-Based Numeric Dysgraphia Detection	Develop an intelligent neural network model capable of distinguishing between dysgraphic and non-dysgraphic handwriting patterns using a locally collected numeric dataset from Sri Lankan children aged 6 to 10.
Alternative Letter Dysgraphia Detection Model	Create an additional letter dysgraphia model using publicly available datasets to benchmark performance and validate the effectiveness of the numeric model.
Integrated Intervention Platform	Combine early detection with intervention modules such as gamified activities, educational tools, and AI-driven chatbots with 3D avatars to provide continuous learning and emotional support.
Awareness and Community Engagement	Promote awareness about dysgraphia and learning disabilities through digital tools, providing real-time insights, community chat features, and educational resources for parents and educators.
Data Ethics and Privacy	Ensure that all data collection and storage processes adhere to ethical standards, protecting the privacy and confidentiality of children's handwriting and assessment data.

4.3 System Design

The system was designed using a modular, service-oriented architecture to ensure scalability, maintainability, and flexibility. The solution follows a three-tier architecture presentation, application, and data layers which together form a robust structure capable of managing real-time data flows, AI inference, and user interaction seamlessly.

The design emphasizes usability and accessibility, integrating both web based and mobile friendly interfaces. It connects children, parents, teachers, and psychologists in a collaborative digital environment where detection, progress tracking, and educational activities are unified under one ecosystem.

4.3.1 System Architecture

The complete system architecture of the proposed solution is illustrated in Figure 4.1, which shows the relationship between frontend modules, backend services, and shared data repositories.

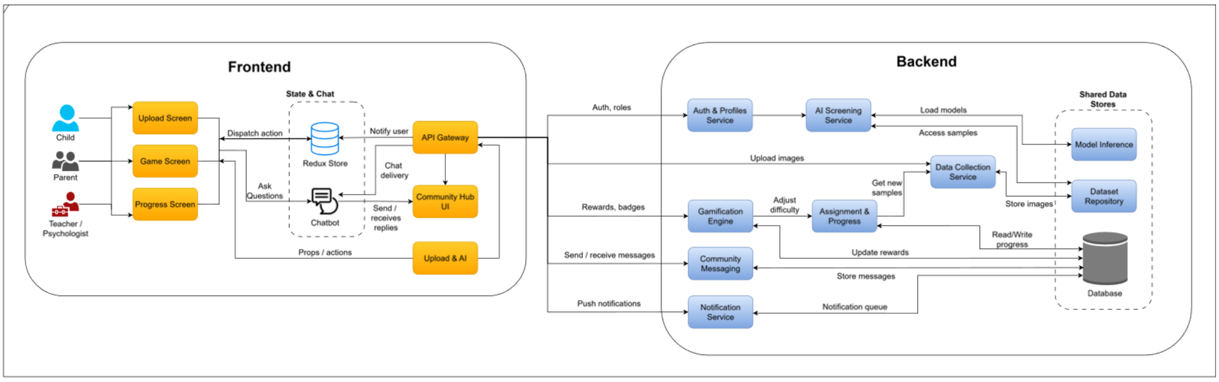


Fig. 4.1 System Architecture Diagram (self-composed)

The architecture of the proposed system is composed of two main layers, the Frontend Interface Layer and the Backend Intelligence Layer, which work together through a centralized data repository. This structure ensures seamless data exchange, system scalability, and efficient performance across both user-facing and server-side components.

The Frontend Interface Layer serves as the primary interaction point for users, enabling engagement among children, parents, teachers, and psychologists. It is designed with accessibility, simplicity, and interactivity in mind to encourage continuous use and learning. The Upload Screen is the first point of interaction, allowing users particularly children or their parents to upload images of handwriting samples or numeric worksheets for analysis. This module connects directly to the AI inference engine in the backend to facilitate real-time dysgraphia detection. The Game Screen is designed to make learning enjoyable and therapeutic. It presents gamified activities that focus on improving fine-motor control, handwriting accuracy, and numeric recognition, thus helping children strengthen their writing skills while remaining motivated. The Progress Screen functions as a performance tracker, enabling parents, teachers, and psychologists to visualize a child's development over time. It displays AI-generated reports, intervention outcomes, and personalized recommendations for further improvement.

State management within the frontend is maintained through the Redux Store, which ensures that data flow between user actions, game responses, and chatbot interactions remains consistent and synchronized. This structure allows dynamic updates, such as progress tracking or chatbot conversations, to occur in real-time without performance degradation. A key component of the frontend is the AI Chatbot, which supports bilingual communication in Sinhala and English. The chatbot offers immediate feedback, motivational messages, and tailored guidance to children during exercises, while also providing technical or behavioral insights to parents and educators. Its natural language processing capability ensures smooth and context-aware interaction, bridging the gap between AI-based detection and human-centered intervention.

The Backend Intelligence Layer is responsible for the computational and data-driven aspects of the system. It comprises multiple microservices that operate collaboratively to deliver intelligence, security, and analytics. The Authentication and Profile Management Service oversees user identity verification and role-based access control, ensuring that each user child, parent, teacher, or therapist interacts with the system according to predefined privileges. The AI Screening Service constitutes the core analytical component, loading the pre-trained convolutional neural network (CNN) model and executing handwriting analysis to determine whether a submitted sample indicates dysgraphic tendencies.

The Gamification Engine is designed to maintain engagement by dynamically adjusting game difficulty based on a child's performance and progress. It manages the allocation of virtual rewards, badges, and progression levels, encouraging sustained participation. The Assignment and Progress Service functions as the operational backbone of the intervention framework, responsible for generating personalized learning tasks, recording user performance, and updating progress metrics in real-time. In addition, the Community Messaging and Notification Services facilitate asynchronous communication among stakeholders allowing parents, teachers, and psychologists to exchange feedback, share observations, and receive automated notifications about performance milestones or upcoming interventions.

Data exchange between the frontend and backend occurs through a secure API Gateway, which acts as an intermediary channel to manage requests, enforce access policies, and optimize response time. This ensures that every transaction such as uploading handwriting images, fetching model results, or updating progress data is processed efficiently and securely. Once analyzed, the processed data and model predictions are stored within two core storage units: the Dataset Repository and the Model Inference Store. The Dataset Repository maintains structured collections of handwriting samples and user metadata, while the Model Inference Store houses prediction logs and statistical insights generated by the AI model.

This modular architecture supports long-term scalability, maintainability, and integration potential. By separating functional components into microservices, the system allows future developers to upgrade individual modules such as the AI model, chatbot intelligence, or gamification engine without affecting the overall workflow. Furthermore, the architecture is designed to support multilingual and cross platform extensions, ensuring adaptability for diverse linguistic and educational contexts. This flexible design guarantees that the system can evolve to accommodate a broader range of languages, age groups, and assistive technologies, ultimately contributing to inclusive and sustainable digital health and education initiatives.

4.3.2 Architecture Diagram - Three Tier Architecture

The proposed system follows a three-tier architecture model,

- Presentation Tier (Client Layer) - Handles all user interactions through the web interface. It is developed using React, Tailwind CSS, and Framer Motion for animations. It integrates Firebase for authentication and speech functionalities via Microsoft Cognitive Services SDK.
- Logic Tier (Application Layer) - Implements business logic using Flask based microservices. Each backend service (Detection, Chatbot, Gamification, Math learning) runs on independent Flask servers, managing inference, user communication, and scoring.
- Data Tier (Database Layer) - Manages data persistence using Firebase Firestore and local storage. It stores handwriting images, user progress, and trained models.

Figure 4.2 shows the three-tier architecture and the interconnections between its layers.

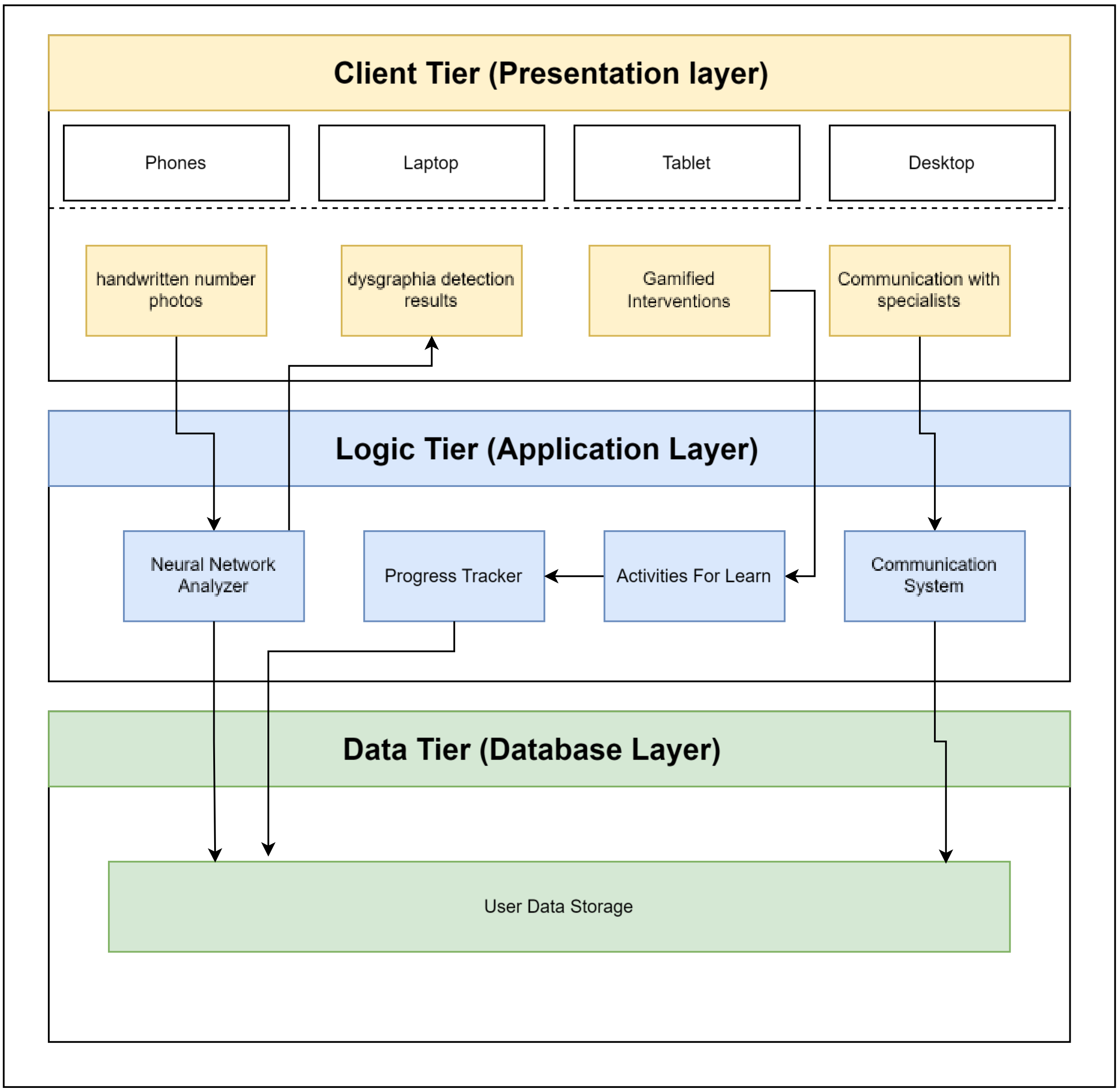


Fig. 4.2 Three-Tier Architecture (self-composed)

4.4 Design Diagrams

4.4.1 Context Diagram

The context diagram illustrates the system's interaction with external entities such as children, parents, and educators. It defines data flow between the user groups and system modules through the web application interface.

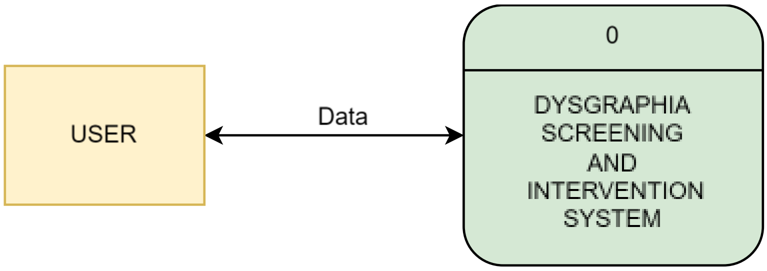


Fig. 4.3 Context Diagram (self-composed)

4.4.2 Component Diagram

The component diagram represents the major modules AI Detection, Gamification, Community Chat, and Database and their interdependencies. Each module interacts with others through REST APIs, ensuring modular integration.

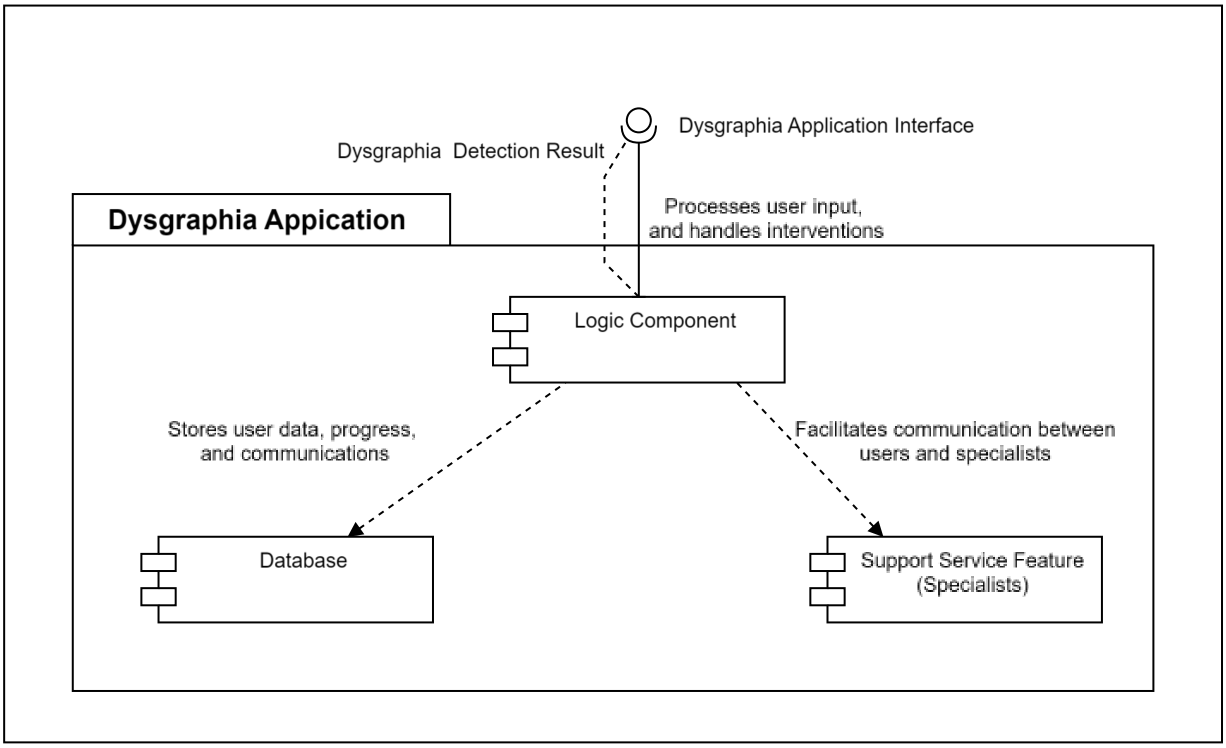


Fig. 4.4 Component Diagram (self-composed)

4.4.3 Data Flow Diagrams

The Data Flow Diagrams (DFDs) illustrate the logical flow of information within the proposed dysgraphia detection and intervention system. These diagrams provide a hierarchical breakdown of how data is captured, processed, analyzed, and presented across different system components. Each level offers an increasingly detailed view of the system's internal processes, ensuring that data movement and transformations are clearly defined from input to output.

Diagram 0 DFD

Represents the system as a single high-level process that interacts with external entities such as the user (student), teacher, parent, and administrator. At this level, the primary data exchanges include handwriting samples, detection results, and progress reports. The user uploads a handwriting sample or performs an assessment, which is processed by the system to generate diagnostic results that are returned to the respective stakeholder.

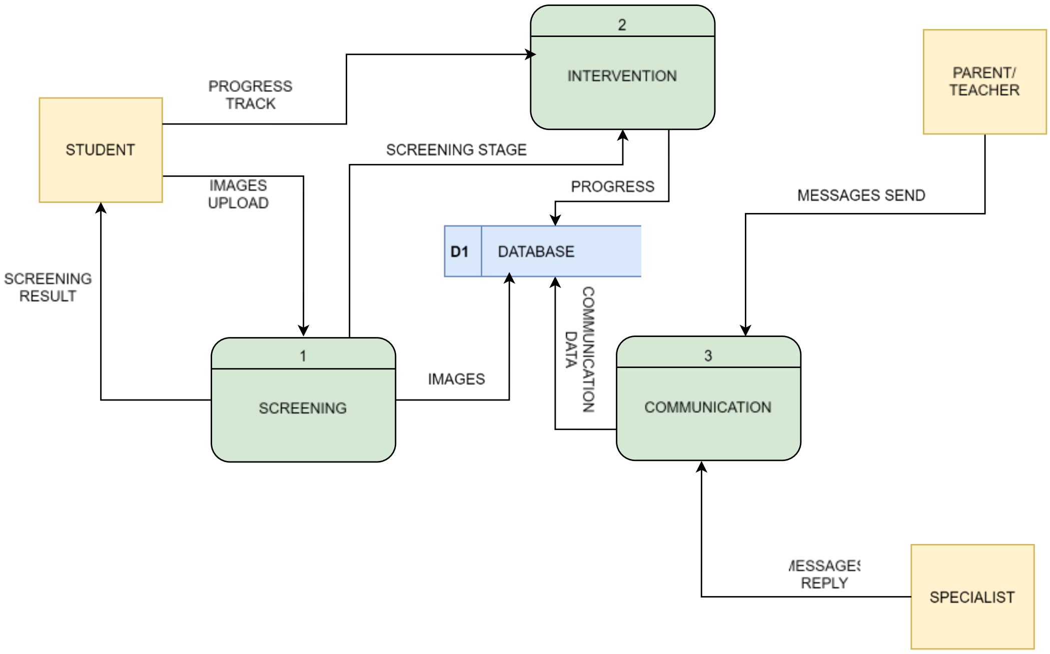


Fig. 4.5 Diagram 0 DFD for the system (self-composed)

DFD Level 1

The Level 1 DFD expands on the internal functions of the system by decomposing the single high-level process into multiple sub-processes. These include data acquisition, preprocessing, CNN-based analysis, feedback generation, and storage management. The diagram shows how raw handwriting data is first preprocessed (e.g., resized, thresholded, and denoised), then classified by the neural network to determine whether it corresponds to dysgraphic or non-dysgraphic handwriting. The classification results are logged into the database, and a personalized report is generated for teachers, parents, and therapists to view through the front-end application.

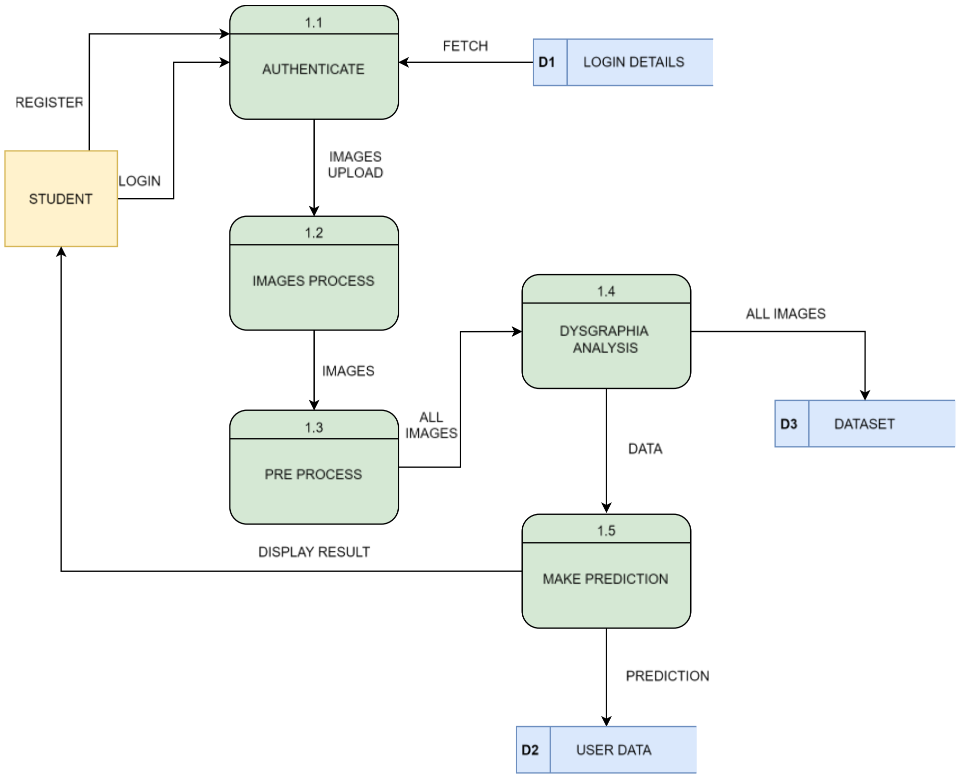


Fig. 4.6 Diagram 1 DFD (self-composed)

DFD Level 2

The Level 2 DFD provides a more granular representation of the system's internal operations within each major component. This level specifically details processes such as image preprocessing (binarization, morphological filtering, noise removal), model inference (feature extraction, classification using the Advanced CNN), result handling (accuracy evaluation, confidence thresholding), and feedback delivery (visual reporting, chatbot-guided recommendations). The data movement between the AI detection module, database layer, and user interface is clearly represented, emphasizing how processed outputs are transformed into actionable insights for intervention and tracking.

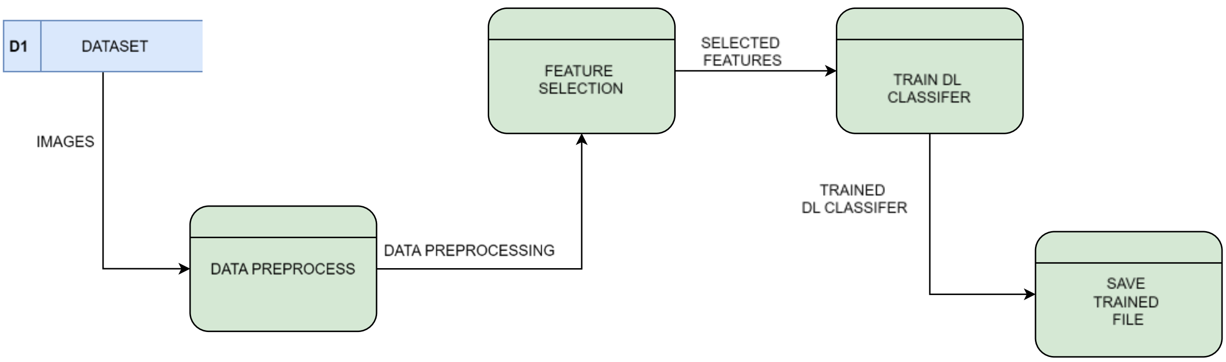


Fig. 4.7 DFD Level 2 Diagram (self-composed)

Together, these three levels of the DFD comprehensively illustrate how the system captures, processes, and interprets handwriting data to deliver accurate dysgraphia screening and meaningful intervention support. The hierarchical structure ensures traceability of data from user input to final output, aligning with the system's objectives of transparency, modularity, and scalability.

4.5 User Interface Design

4.5.1 Wireframe Design (Low Fidelity Prototype)

The low-fidelity wireframes were created to plan the layout and functional flow of the system interfaces. They focus on core features like handwriting upload, progress tracking, and game dashboards without detailed visuals.

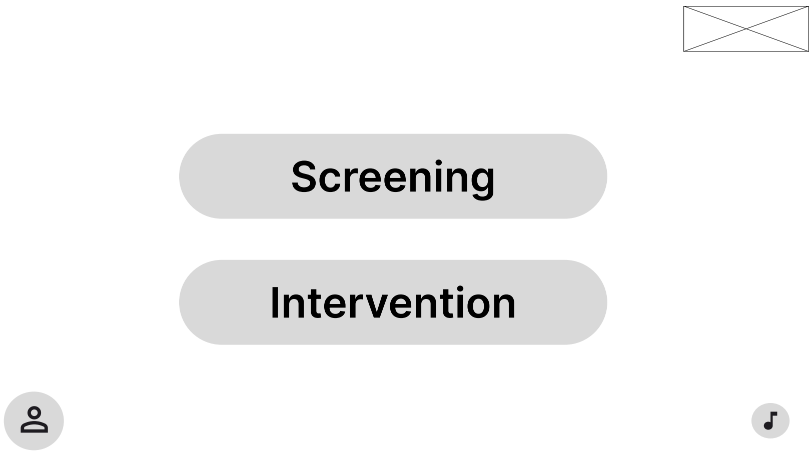


Fig. 4.8 Low Level Wireframe 1 (self-composed)

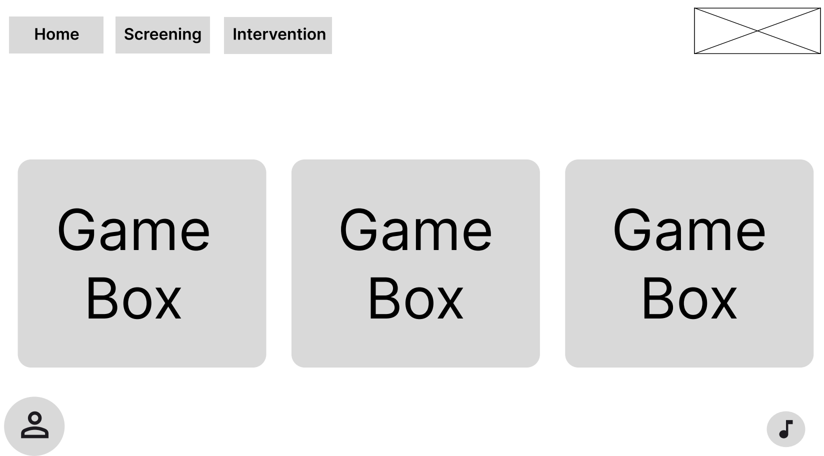


Fig. 4.9 Low Level Wireframe 2 (self-composed)

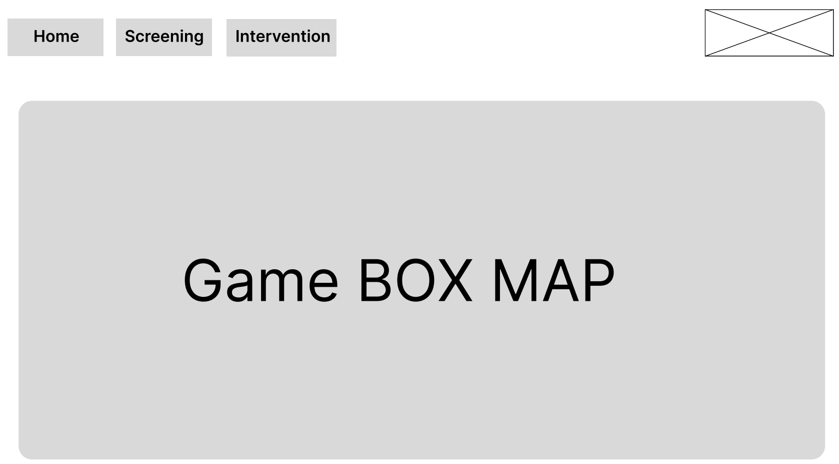


Fig. 4.10 Low Level Wireframe 3 (self-composed)

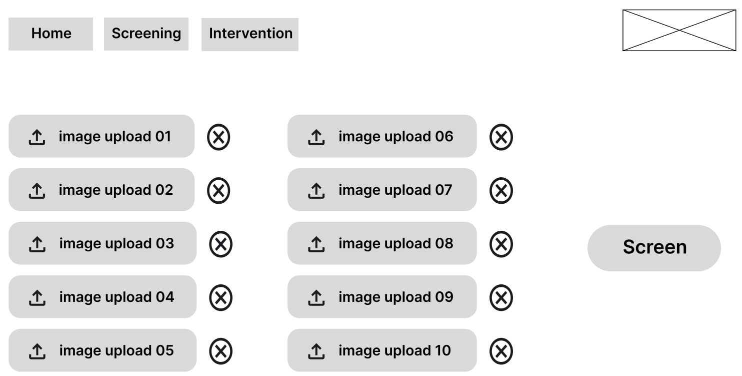


Fig. 4.11 Low Level Wireframe 4 (self-composed)

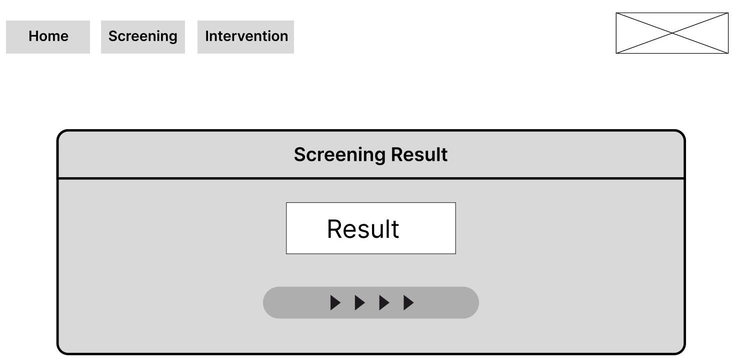


Fig. 4.12 Low Level Wireframe 5 (self-composed)

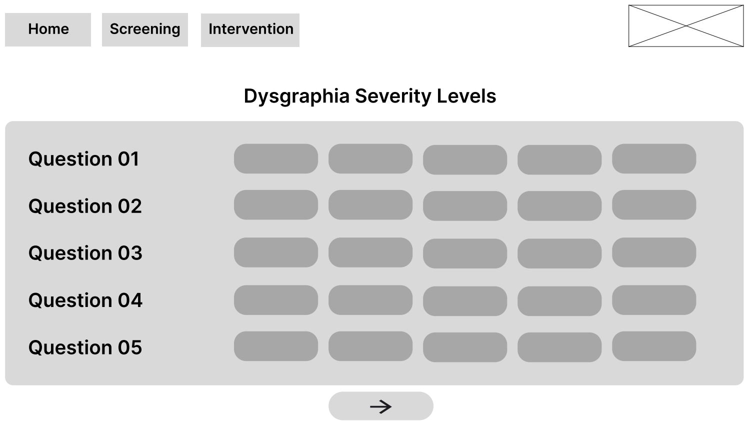


Fig. 4.13 Low Level Wireframe 6 (self-composed)

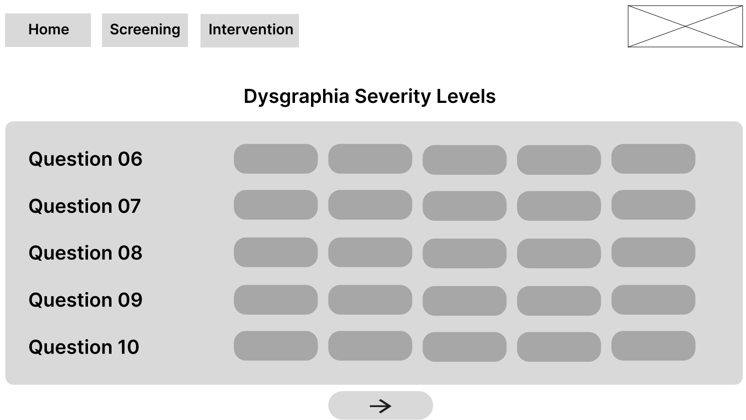


Fig. 4.14 Low Level Wireframe 7 (self-composed)

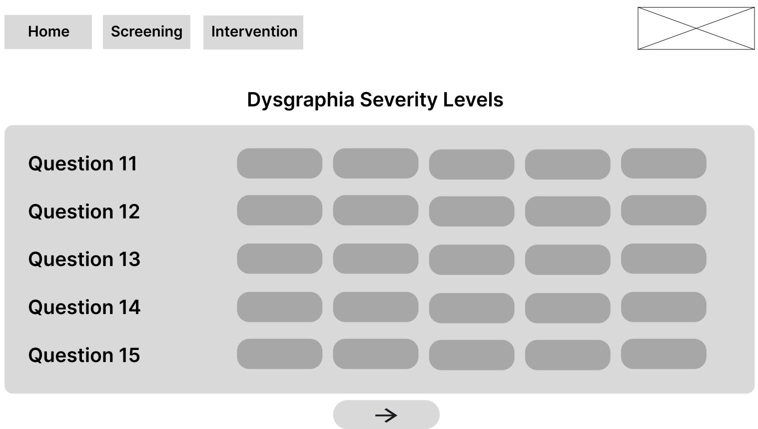


Fig. 4.15 Low Level Wireframe 8 (self-composed)

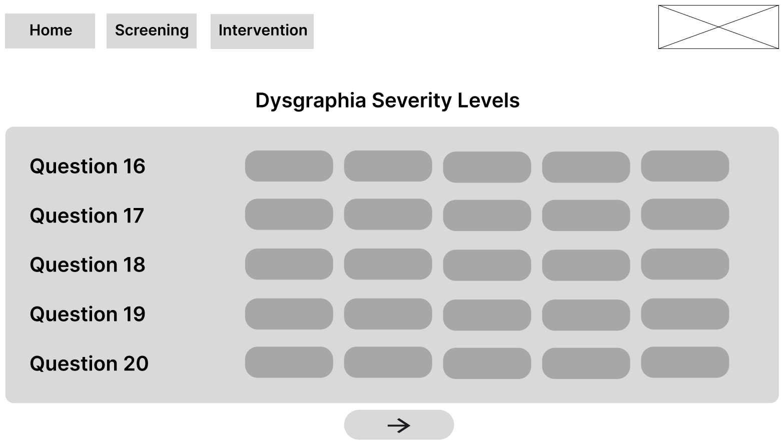


Fig. 4.16 Low Level Wireframe 9 (self-composed)

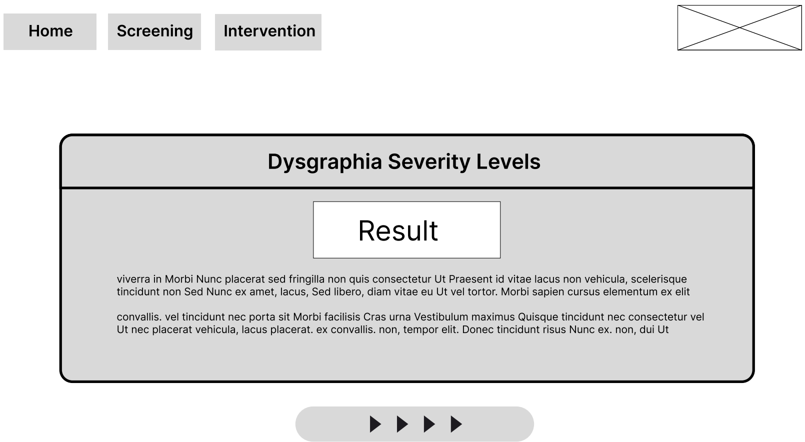


Fig. 4.17 Low Level Wireframe 10 (self-composed)

4.5.2 UI/UX Design (High Fidelity Prototype)

High-fidelity UI/UX prototypes were developed using Figma to demonstrate the final user interface, including colors, typography, animations, and accessibility features. Tailwind CSS and Framer Motion were used for responsive and animated transitions in the actual system.

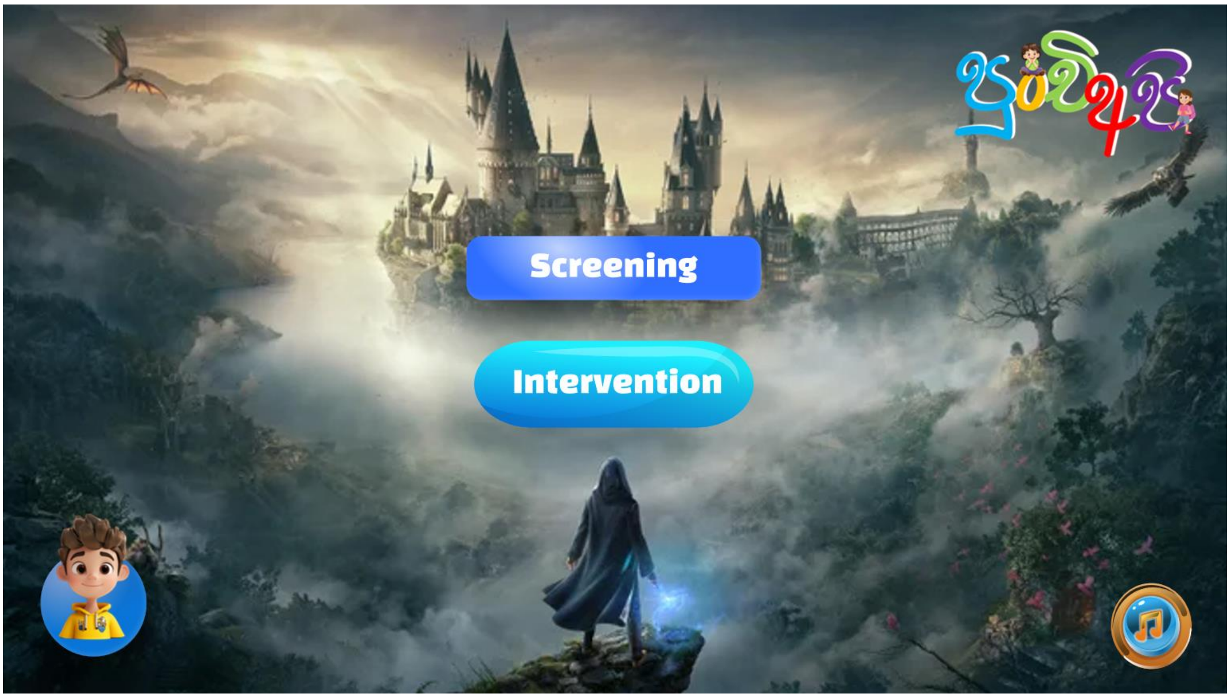


Fig. 4.18 High Level Wireframe 1 (self-composed)

Fig. 4.19 High Level Wireframe 2 (self-composed)

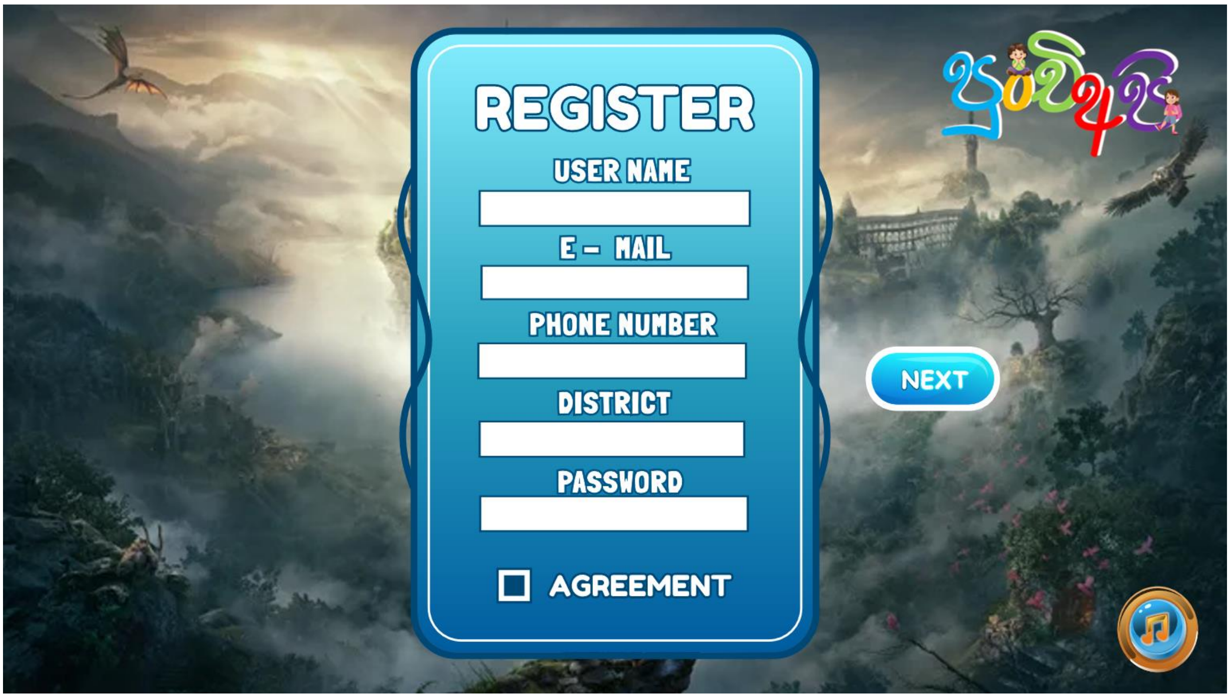


Fig. 4.20 High Level Wireframe 3 (self-composed)

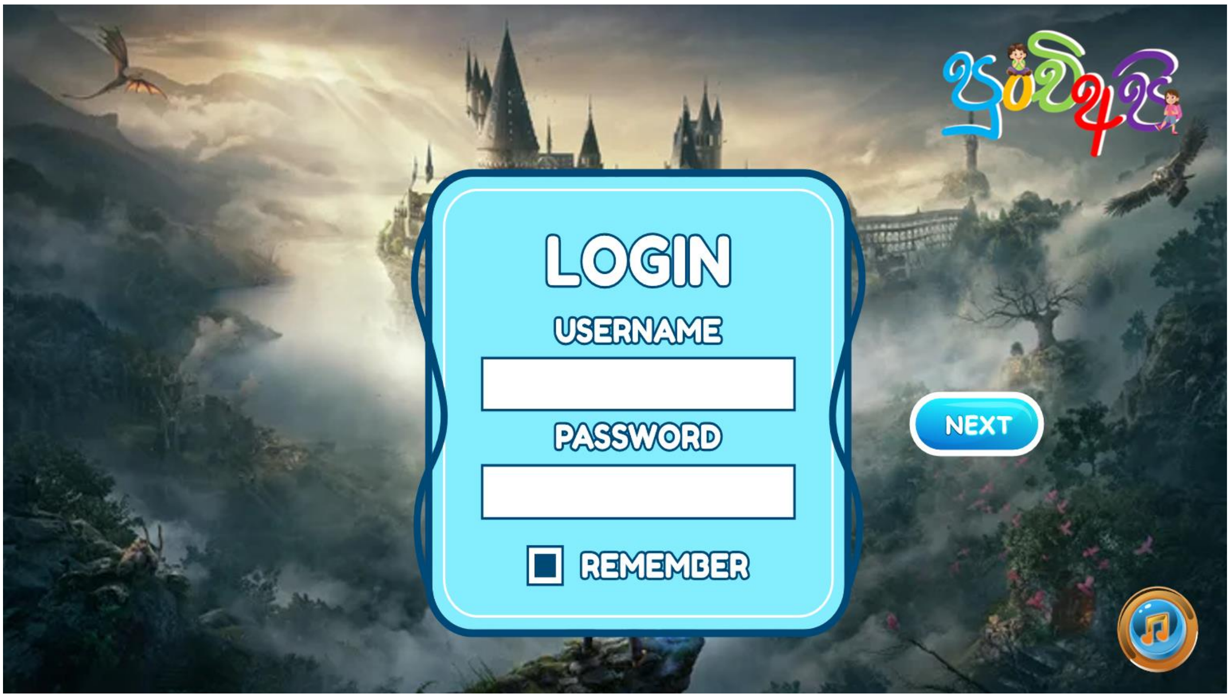


Fig. 4.21 High Level Wireframe 4 (self-composed)

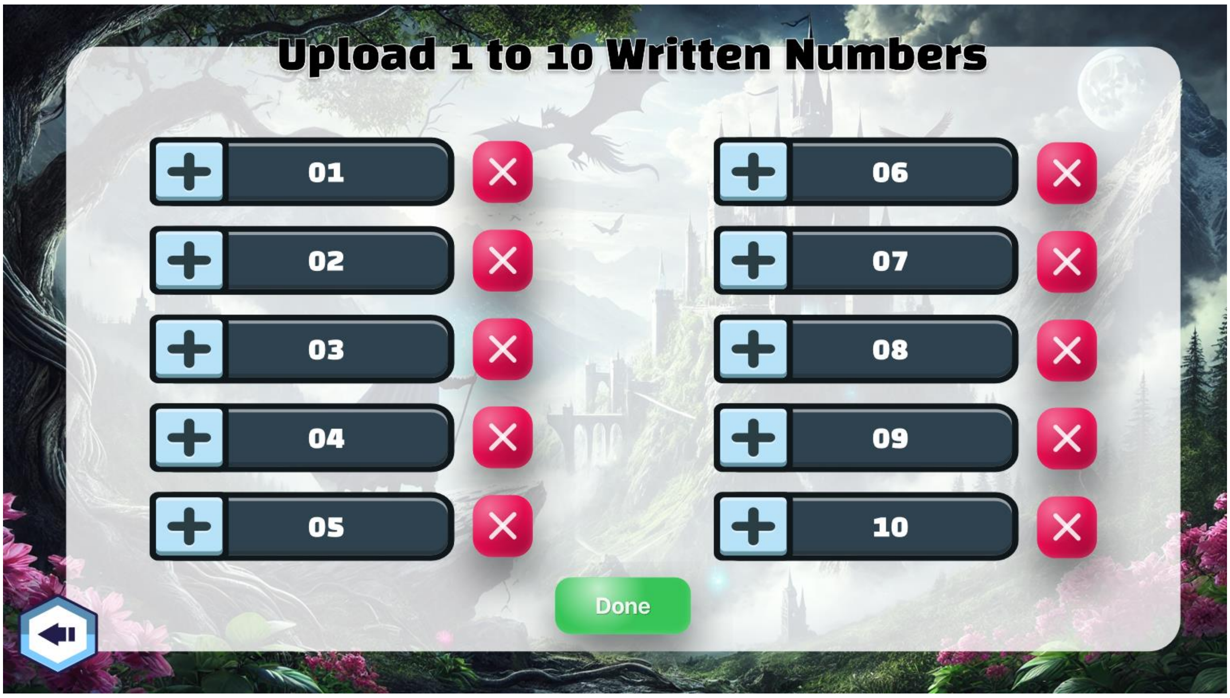


Fig. 4.22 High Level Wireframe 5 (self-composed)

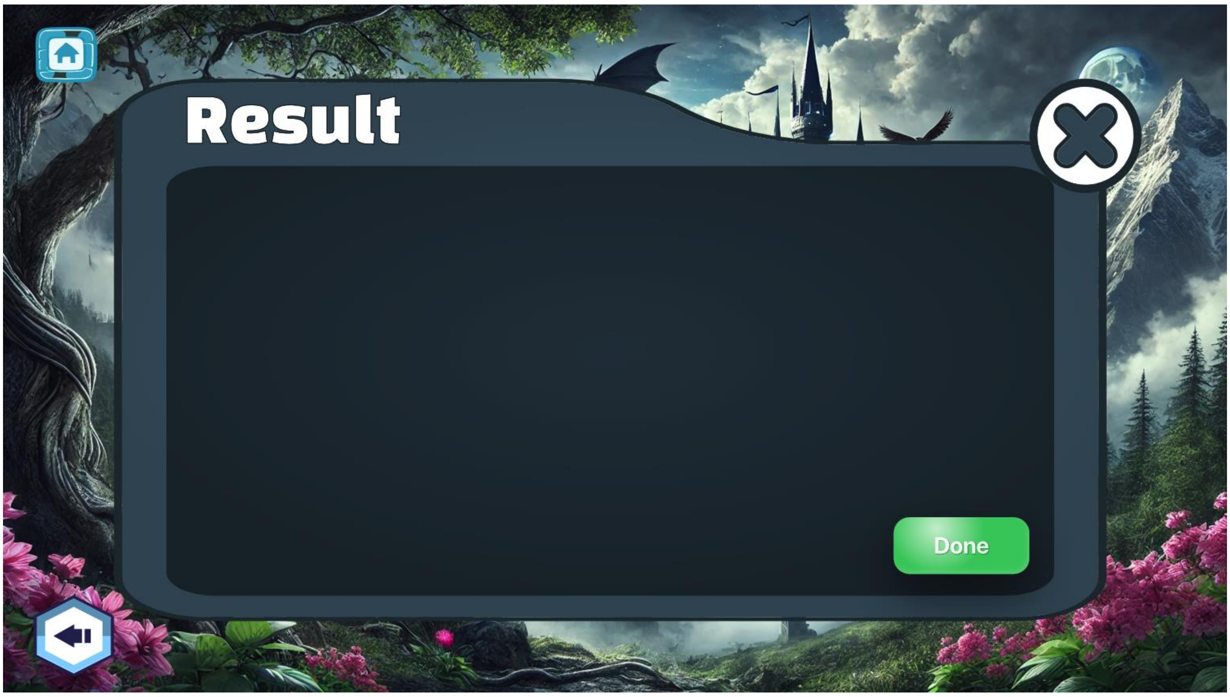


Fig. 4.23 High Level Wireframe 6 (self-composed)

4.6 Technology Selection

The technology stack was carefully chosen to ensure scalability, performance, and accessibility across devices.

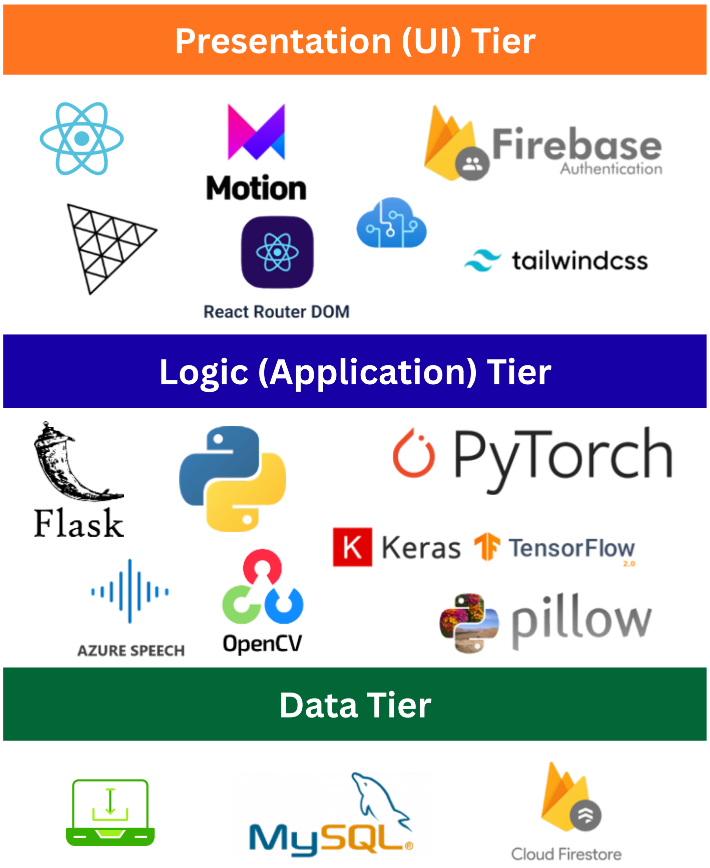


Fig. 4.24 Technology Stack (Self-composed)

This architectural approach separates the system into three distinct layers the Presentation Tier, Logic Tier, and Data Tier each responsible for a specific set of operations that collectively support the functionality of the entire platform. This separation of concerns enables independent development, maintenance, and future scalability of each layer without affecting other components.

The Presentation Tier, also referred to as the user interface (UI) layer, serves as the interaction point between users and the system. This layer was developed using React.js with Create React App, providing a dynamic, responsive, and component-based front-end environment. The design was further enhanced through the use of Tailwind CSS for styling and Framer Motion for smooth transitions and animations, offering an intuitive and visually engaging experience for children, parents, and educators. The interface incorporates React-Router-DOM for seamless navigation across modules such as the dysgraphia assessment page, progress dashboard, community chat, and AI-based interactive avatar interface. Additionally, this tier integrates Microsoft Cognitive Services Speech SDK to support real-time speech recognition and text-to-speech functionalities, enabling voice-based interaction and accessibility for users with learning difficulties. The 3D avatar communication and lip-sync features were implemented using React-Three-Fiber and Drei libraries, creating an immersive and child-friendly virtual environment. Authentication for all users, including students, parents, and teachers, was handled securely through Firebase Authentication, ensuring role-based access control and data privacy.

The Logic Tier, also known as the application or service layer, acts as the central processing hub of the system. It was built using multiple Flask microservices implemented in Python, each handling a specific functional domain such as dysgraphia detection, chatbot communication, and gamified learning interventions. The Detection Service processes handwriting images using libraries like OpenCV and Pillow for preprocessing, while PyTorch is used to load and run the trained Advanced CNN model for dysgraphia classification. The Backend microservice manages conversational AI tasks through integration with Azure Speech Services and OpenAI-based NLP frameworks, supporting intelligent, context-aware chatbot interactions. The immersive intervention microservice facilitates gamified and educational interventions, delivering skill-building exercises designed to improve fine motor and numeric-writing abilities. ensuring smooth communication between modules. The logic layer also supports real-time updates and synchronization between users via Firebase SDK listeners, particularly within the community chat component.

The Data Tier is responsible for managing persistent data storage, retrieval, and security. The system uses a combination of Firebase Firestore, MySQL, and local file storage to accommodate different data requirements efficiently. Firebase Firestore handles authentication, user profiles, and real-time community interactions such as messages and forum threads, ensuring consistent performance and cross-device synchronization. Meanwhile, MySQL is employed to store intervention analytics, user progress data, and activity outcomes from gamified learning modules. Locally, the system stores essential artifacts such as uploaded handwriting images, processed datasets, and trained model weights. This hybrid storage approach balances scalability and security, allowing sensitive datasets to remain local while user interactions remain cloud based. Configuration files, API keys, and model environment settings are securely maintained through .env and config.json files to ensure safe access to third-party APIs like Azure and Firebase.

4.7 Dataset Selection

For this research, two primary datasets were used

Numeric Dysgraphia Dataset (Proposed)

Since no public dataset existed for numeric dysgraphia, a new dataset was developed across Sri Lankan districts. It includes 8,000 images collected from 482 participants, aged 6 to 10, comprising both dysgraphic and non-dysgraphic samples. The dataset underwent preprocessing involving resizing, grayscale conversion, Gaussian blur, and adaptive thresholding before being used for model training.

Letter Dysgraphia Dataset (Alternative Approach)

To complement numeric data and evaluate model generalization, a publicly available handwriting dataset from Kaggle was used for letter dysgraphia detection. This approach enabled comparative testing between numeric and letter dysgraphia models, validating that both subsets can be analyzed within a unified deep learning framework.

4.8 Implementation of the Core Functionality

4.8.1 Dysgraphia Detection Model Implementation

The preprocessing pipeline included oversampling to balance classes and train-test splitting (80:20). The proposed Advanced CNN model was implemented in PyTorch with optimized convolutional layers and dropout regularization. The model was trained separately on both the numeric and letter datasets, then compared with standard architectures such as VGG16, ResNet50, BiLSTM, and Swin Transformer.

```python
img_path = os.path.join(input_folder, filename)
img = cv2.imread(img_path)

# Step 1: Resize to 300x300
resized_img = cv2.resize(img, (300, 300), interpolation=cv2.INTER_AREA)

# Step 2: Convert to Grayscale
gray_img = cv2.cvtColor(resized_img, cv2.COLOR_BGR2GRAY)

# Step 3: Noise Removal (Gaussian Blur)
blurred_img = cv2.GaussianBlur(gray_img, (7, 7), 0)

# Step 4: Adaptive Thresholding (Fine-tuned)
binary_img = cv2.adaptiveThreshold(
    blurred_img,
    255,
    cv2.ADAPTIVE_THRESH_MEAN_C,
    cv2.THRESH_BINARY_INV,
    21,
    5
)

# Step 5: Waste Clearing
cleaned_img = clear_waste(binary_img)

# Step 6: Morphological Operations (Optional)
kernel = np.ones((3, 3), np.uint8)  # Smaller kernel
opened_img = cv2.morphologyEx(cleaned_img, cv2.MORPH_OPEN, kernel)

# Step 7: Fallback to Global Thresholding if Binary is Blank
if cv2.countNonZero(opened_img) == 0:  # Check if the image is all black
    _, opened_img = cv2.threshold(blurred_img, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

# Step 8: Resize to 32x32 for Uniformity
final_img = cv2.resize(opened_img, (32, 32), interpolation=cv2.INTER_AREA)
```

Fig. 4.25 Pre-processing Data

```python
    image_paths = []
    labels = []
    handwriting_ids = []
    for filename in os.listdir(folder_path):
        if filename.lower().endswith((".png", ".jpg", ".jpeg", ".bmp", ".tif", ".tiff", ".gif")):
            image_paths.append(os.path.join(folder_path, filename))
            labels.append(label)
            handwriting_id = os.path.splitext(filename)[0]  # Extract the filename without the extension
            handwriting_ids.append(handwriting_id)
            print(f"Processed filename: {filename} -> Handwriting ID: {handwriting_id}")  # Debugging print stater
    return image_paths, labels, handwriting_ids

# Get dysgraphic image paths, labels, and handwriting IDs
dysgraphic_image_paths, dysgraphic_labels, dysgraphic_handwriting_ids = get_image_paths_labels_and_ids(
    dysgraphic_folder, label=1)

# Get non-dysgraphic image paths, labels, and handwriting IDs
non_dysgraphic_image_paths, non_dysgraphic_labels, non_dysgraphic_handwriting_ids = get_image_paths_labels_and_id
    non_dysgraphic_folder, label=0)

# Combine the paths, labels, and handwriting IDs
image_paths = dysgraphic_image_paths + non_dysgraphic_image_paths
labels = dysgraphic_labels + non_dysgraphic_labels
handwriting_ids = dysgraphic_handwriting_ids + non_dysgraphic_handwriting_ids

# Create a DataFrame to store the image paths, labels, and handwriting IDs
data = {'Image_Path': image_paths, 'Label': labels, 'Handwriting_ID': handwriting_ids}
df = pd.DataFrame(data)

# Shuffle the DataFrame (optional but recommended)
df = df.sample(frac=1, random_state=42).reset_index(drop=True)

# Split the data into training and testing sets (80% training, 20% testing)
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42, stratify=df['Label'])
```

Fig. 4.26 Oversampling Dataset and splitting

Please refer to **APPENDIX 8** for the screenshots of the other models Machine Leaning and Deep Learning Comparison.

The CNN architecture was fine-tuned using learning rate scheduling and batch normalization to enhance convergence stability. The output layer used a sigmoid activation for binary classification, producing predictions on whether a given handwriting sample belonged to a dysgraphic or non-dysgraphic participant.

4.8.2 Immersive Intervention Implementation

The immersive intervention component of the system was designed with the primary objective of delivering an engaging, skill-building experience for children with dysgraphia through interactive and multi-sensory activities. This section focuses on the development and integration of an intelligent, responsive, and child-centered intervention environment that complements the handwriting detection process. The intervention framework integrates both AI-driven conversational agents with 3D avatars for emotional and cognitive engagement, as well as interactive math and motor coordination games aimed at improving handwriting fluency and spatial awareness.

The AI chatbot system forms the core of the immersive environment, implemented using a combination of Flask, React, and Azure Cognitive Services Speech SDK. The chatbot allows real-time interaction with the user through both voice and text, supporting bilingual communication in Sinhala and English. A key feature of this component is the use of 3D avatars, which include both male and female models capable of realistic lip-sync and facial animations synchronized with speech synthesis. These avatars were implemented using React Three Fiber and react-three, drei libraries, enabling natural conversational flow between the child and the virtual companion. The chatbot supports various modes such as emotional support, educational guidance, and motivational reinforcement, thus functioning as both a virtual therapist and learning assistant.

Complementing the conversational module, the immersive intervention service was implemented to target fine-motor control and numeric fluency through short, repetitive, and engaging activities. Each activity was designed around visual-motor interaction tasks such as on-screen tracing, pattern drawing, and number formation, using guided animations and progressive feedback mechanisms. The sessions were intentionally structured to last between three to seven minutes to maintain the child's attention and prevent cognitive fatigue. Activities progress through gradual scaffolding, starting from guided tracing and advancing to freehand repetition and timed challenges, thereby allowing children to develop writing confidence and precision incrementally.

Each session within the intervention framework follows a clearly defined workflow. The user begins by launching the immersive intervention module through the main application interface, selecting from several activities such as number formation, pattern tracing, or simple calculations involving visual-motor coordination. During guided practice, the system provides visual cues such as ghost lines and directional arrows. In independent attempts, performance metrics such as accuracy, timing, and stroke deviation are automatically recorded. Immediate feedback is delivered through animations and corrective suggestions, encouraging continuous improvement while maintaining motivation through badges and positive reinforcement.

Personalization plays a central role in maintaining engagement and ensuring meaningful learning outcomes. The system dynamically adjusts activity difficulty based on the child's previous performance, tuning parameters such as line sensitivity, stroke width, and path tolerance. Reinforcement styles are adapted to the user's age and temperament, offering options like celebratory animations, stickers, or calm auditory confirmations. Data logs record metrics such as task completion rate, average deviation from intended paths, and response time per session. Although currently stored locally, these logs are designed to be extended into centralized databases for aggregated analysis by educators and therapists.

The user experience and accessibility aspects of the intervention system were carefully considered. The interface includes large icons, high-contrast elements, and support for both left- and right-handed modes. The low-latency rendering ensures a smooth interaction experience, particularly in activities requiring precise motor control. In addition to usability, ethical and safety considerations were prioritized throughout the design. The system does not collect any personally identifiable information (PII) and adheres to child-safety and privacy regulations. The tone and feedback mechanisms are framed to be supportive and non-diagnostic, focusing on growth and encouragement rather than evaluation or criticism.

The evaluation of this intervention system considered both quantitative and qualitative metrics. Quantitatively, improvements were measured through reductions in tracing deviations, decreased completion times, and more consistent spacing in written outputs. Qualitatively, engagement was assessed through session length, frequency of voluntary participation, and feedback from caregivers and teachers. Early observational findings indicated that children demonstrated noticeable improvement in number formation consistency and writing confidence after repeated sessions.

Although the current version logs data locally, future iterations will integrate cloud-based synchronization for centralized monitoring. Planned improvements include adaptive reinforcement loops powered by AI, gesture-based bilateral coordination tasks, and therapist dashboards for personalized learning management. Overall, the immersive intervention component provides a holistic, engaging, and safe digital environment that complements the AI-driven detection system, enabling children with dysgraphia to practice, improve, and gain confidence in their writing abilities through playful yet scientifically informed interaction.

4.9 User Interface

Please refer to **APPENDIX 10** for the screenshots of the User Interface.

4.10 Chapter Summary

This chapter elaborated on the full technical implementation of the proposed dysgraphia detection and intervention system, including its architectural design, dataset preparation, model implementation, and system integration. It demonstrated how modern deep learning techniques, coupled with a modular three-tier architecture and bilingual interfaces, were used to build a scalable and context-aware educational support system.

The next chapter, **Results and Analysis**, will present the experimental outcomes of model performance, comparative analysis between machine learning and deep learning techniques, and usability feedback derived from user testing.

Chapter 5

Results and Analysis

5.1 Chapter Overview

This chapter presents the comprehensive results and analysis of the study, encompassing findings from the survey, observational data from parents and teachers, and performance evaluations of the implemented machine learning (ML) and deep learning (DL) models for both letter and numeric dysgraphia detection. The purpose of this chapter is to assess whether the proposed system achieves the research objectives defined in earlier chapters, namely accurate early detection and support for children with dysgraphia using AI-based techniques. The results are organized logically starting with human-centered survey findings, followed by model performance evaluations, and concluding with a discussion on their implications for educational and clinical practice.

5.2 Survey Findings and Observations

5.2.1 Survey Questionnaire

The questionnaire consisted of multiple choice and short answer questions aimed at assessing participants' awareness, perceptions, and readiness to engage with AI based tools for dysgraphia detection and intervention. Table 5.1 presents a summary of each survey question, its objective, and the key findings derived from participant responses.

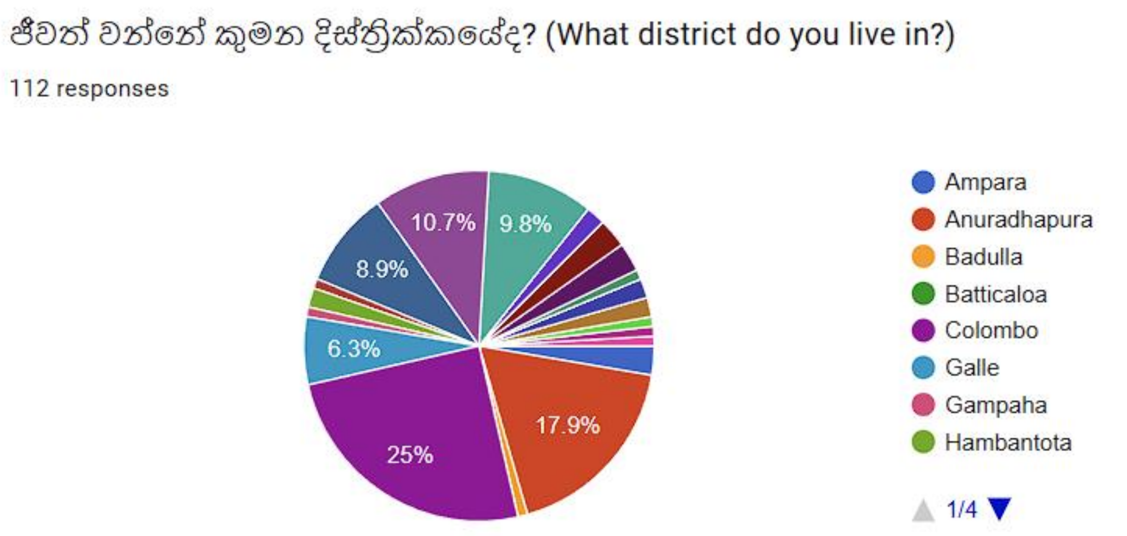
To gain contextual understanding of the awareness of dysgraphia and learning disabilities among Sri Lankan stakeholders, a survey was conducted among 112 participants, including parents and teachers from 20 districts. The purpose was to assess awareness, perception, and willingness to adopt technology-based interventions. The results indicated that 92.9% of respondents were unaware of learning disorders such as dyslexia and dysgraphia, while only 7.1% had limited prior knowledge. This underscores a substantial awareness gap regarding learning disabilities in Sri Lanka.

Furthermore, 98.2% of participants were unfamiliar with the term dysgraphia, revealing a lack of early identification practices in schools. Despite this, a positive finding was that 100% of respondents expressed openness toward digital tools for supporting children with learning disabilities. When asked about preferred activity durations for children, 55.6% selected sessions lasting 30 to 45 minutes, while 33.3% preferred 15 to 30 minutes, reinforcing that short, gamified interventions are most effective for maintaining engagement and attention.

Table 5.1 Survey Questionnaire Reply Analysis

Question	Current living area? (District)
Aim of Question	To get a better understanding of the district of the participants (parents and teachers) in the survey questionnaire.

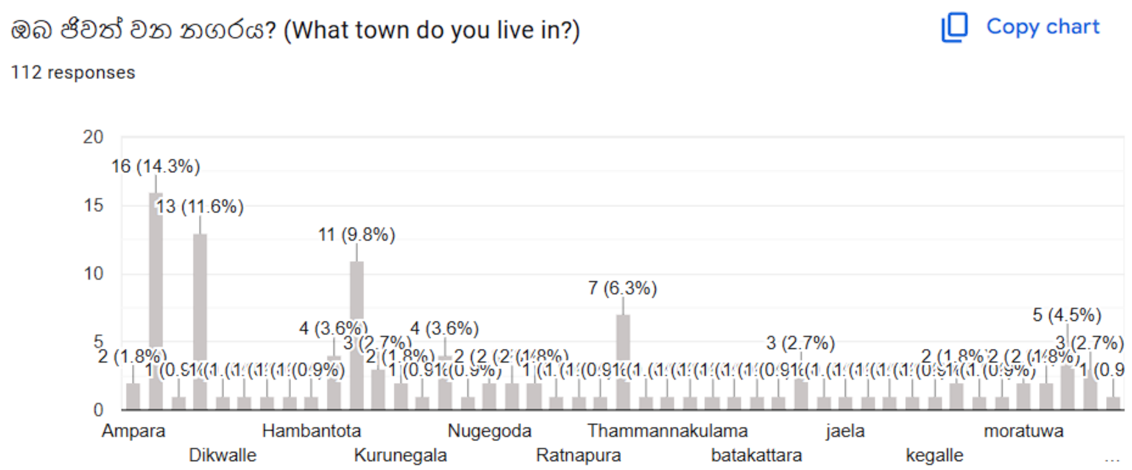
Findings and Conclusion



The survey questionnaire was filled out by participants from 20 different districts. It clearly shows that the survey questionnaire participants are spread across the country.

Question	What town do you live in?
Aim of Question	To get a better understanding in depth of the towns of the participants lives (parents and teachers) in the survey questionnaire.

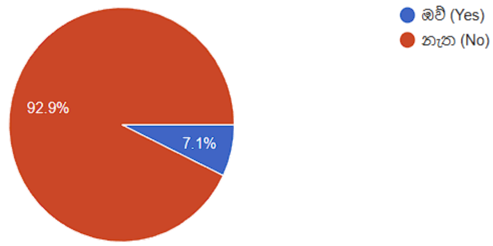
Findings and Conclusion



The survey questionnaire was filled out by participants from different towns in Colombo district. It clearly shows that the survey questionnaire participants are spread across the Colombo area.

Question	Are you aware of Specific Learning Disorders?
Aim of Question	To find out whether the participants have awareness of specific learning disorders.

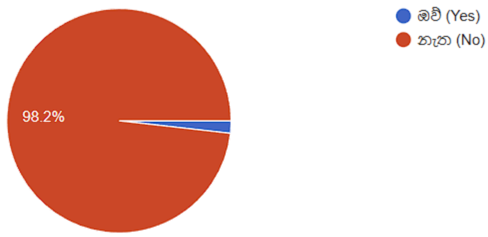
Findings and Conclusion



According to these responses, the majority of 92.9% of participants are unaware of these specific learning disorders and only 7.1% of participants have an idea about specific learning disorders. This clearly reflects the gap between awareness and lack of awareness about specific learning disorders in society. This indicates the requirement for a system capable of providing an understanding of specific learning disorders and identifying their risks.

Question	**Do you know about Dysgraphia disorder?**
Aim of Question	To determine whether the participants have awareness of Dysgraphia specific learning disorder.

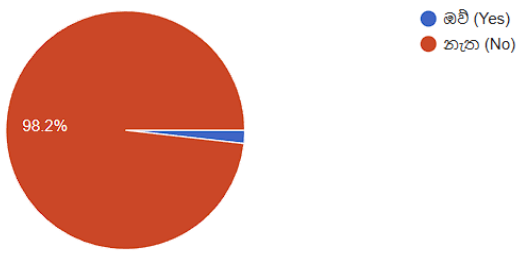
Findings and Conclusion



According to these responses, the majority of 98.2% of participants are unaware of this Dysgraphia specific learning disorder and only 1.8% of participants have an idea about Dysgraphia disorder. Furthermore, the percentage of participants who are aware of specific learning disorders is 7.1 %, but the percentage of participants who are aware of Dysgraphia is considerably low at 1.8%. This clearly shows the lack of awareness about Dysgraphia in the society. These facts can be considered as evidence that clearly reflects the need for a detection system for Dysgraphia disorder.

Question	**Are you familiar with the symptoms of Dysgraphia?**
Aim of Question	To get knowledge about the participants who have awareness of Dysgraphia disorder symptoms

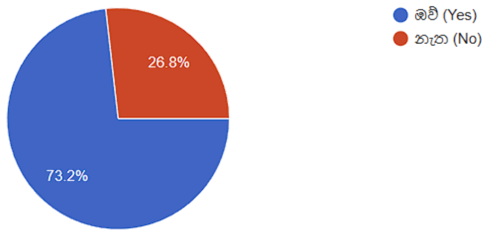
Findings and Conclusion



This survey question results indicate, the majority of 98.2% of participants are unaware of this Dysgraphia disorder symptoms and only 1.8% of participants have an idea about Dysgraphia disorder symptoms. This is clear evidence for the lack of awareness about Dysgraphia symptoms in the society.

Question	**At your school-age/ school have you met children who behave normally and understand many things well but are unwilling or unable to write?**
Aim of Question	In order to know whether the participants of the survey knowingly or unknowingly have worked with someone who has Dysgraphia disorder.

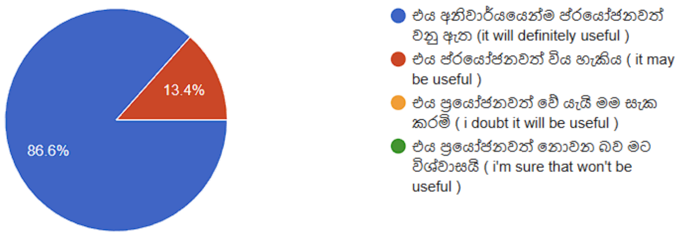
Findings and Conclusion



According to this survey question responses, 91% of the survey participants thought they were, knowingly or unknowingly have worked together with someone who has Dyslexia or Dysgraphia disorder during their school days or at school. Furthermore, 73.2% of the survey participants thought they were, knowingly or unknowingly haven't worked together with someone who has Dysgraphia disorder during their school days or at school. Thus, the majority of the participants in the survey think that they have worked with children who have Dysgraphia symptoms during their school days or at school. This implies that there are a large number of children at risk of Dysgraphia disorders in society.

Question	**If there is an application that can detect the risk of Dysgraphia in Sinhala language-speaking children, how much would you think that it will benefit society?**
Aim of Question	To find out whether this Dysgraphia risk detection system will benefit children, parents and teachers.

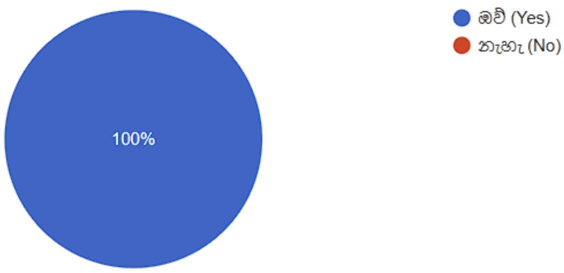
Findings and Conclusion



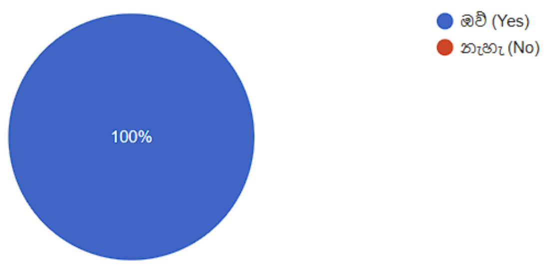
According to this survey question responses, the majority of 86.6% of participants thought this risk detection system will definitely be useful and 13.4% of participants thought this risk detection system may be useful. Finally, survey participants believe that this system will be an effective risk detection system for society.

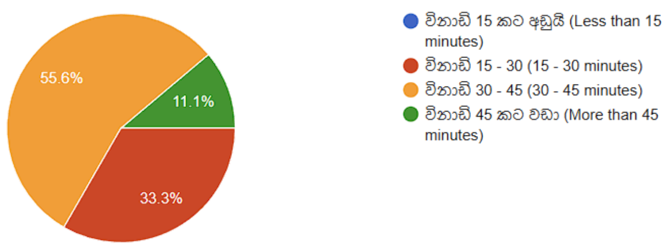
Question	**Would you find it helpful to access a community or support group within the app for parents and teachers of children with dysgraphia?**
Aim of Question	To know parents and teachers of children with dysgraphia who like to include in app to community feature to discuss with them to get better understanding of their children's behaviors and abilities

Findings and Conclusion



100% of positive finding was that respondents expressed openness toward to include community feature in application then community can grow together

<table>
<tr><td>Question</td><td>Would you like to give your mobile phone or computer for educational games ?</td></tr>
<tr><td>Aim of Question</td><td>To find parents would like to give their child to their person computer or mobile phone to play educational games</td></tr>
<tr><td colspan="2">Findings and Conclusion


A positive finding was that 100% of respondents expressed openness toward to give their mobile phones or computer for supporting children with learning disabilities.</td></tr>
<tr><td>Question</td><td>How much time do you think your child would like to spend on educational games per day?</td></tr>
<tr><td>Aim of Question</td><td>To find children playable time to set games time durations</td></tr>
<tr><td colspan="2">Findings and Conclusion


When asked about preferred activity durations for children, 55.6% selected sessions lasting 30 to 45 minutes, while 33.3% preferred 15 to 30 minutes, and further more 11.1% think more than 45 minutes, reinforcing that short, gamified interventions are most effective for maintaining engagement and attention. parents also know children would like to play educational games with gamified.</td></tr>
</table>

Overall the survey findings confirmed a substantial lack of awareness of dysgraphia among Sri Lankan parents and educators, yet revealed strong interest in digital and AI based solutions. These findings provided essential guidance for shaping the system's design priorities particularly its accessibility, gamified learning modules, and community driven support features.

5.2.2 Observations

In addition to the survey, observational sessions and structured interviews were conducted with parents of children diagnosed with dysgraphia at Lady Ridgeway Hospital (LRH) and Senehasa Education Resource and Research Centre. Parents highlighted major challenges, such as delayed recognition of handwriting difficulties and limited accessibility to specialist support outside urban areas. Teachers noted the difficulty of differentiating dysgraphia from general handwriting issues in large classroom settings. These real world insights directly informed the development of the proposed AI powered screening and intervention system, ensuring cultural relevance and educational feasibility.

5.3 Deep Learning Model Evaluation

5.3.1 Letter Dysgraphia Detection

Letter dysgraphia experiments were conducted using the publicly available Kaggle handwriting dataset. Multiple DL architectures were tested and compared, including Baseline CNN, VGG16, ResNet50, BiLSTM, Vision Transformer, and the proposed Advanced CNN.

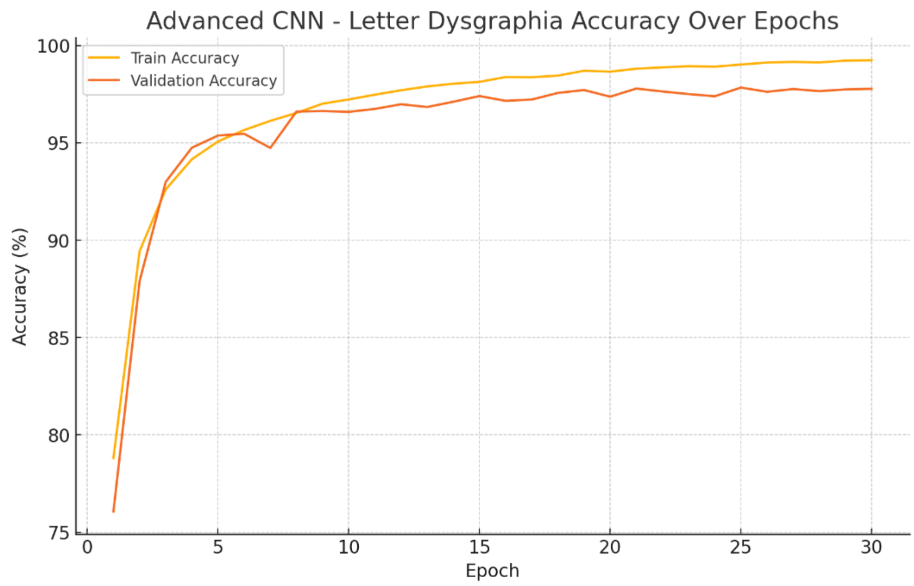


Fig. 5.1 Advanced CNN for Letters

As shown in Figure 5.1, the training and validation accuracy curves for the Advanced CNN demonstrated rapid convergence, achieving 99.2% training accuracy and 97.8% validation accuracy after approximately 30 epochs. The smooth curve indicates strong learning stability and absence of overfitting.

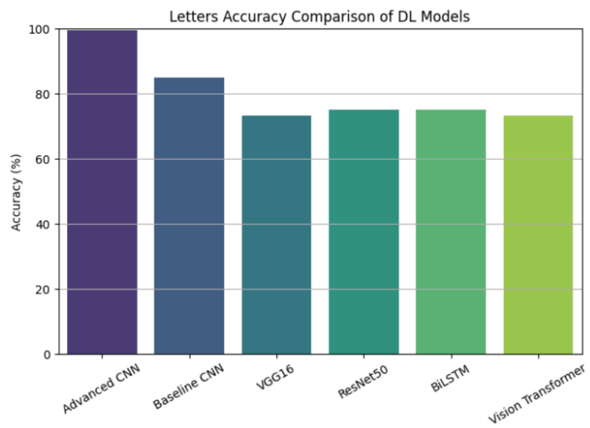


Fig. 5.2 Comparison of Deep Learning Models for Letters

The accuracy comparison bar chart (Figure 5.2) reveals that the Advanced CNN outperformed all other models, achieving the highest accuracy, followed by Baseline CNN (85%), while transfer learning models such as VGG16 and ResNet50 achieved approximately 73 to 75%. BiLSTM and Vision Transformer performed moderately well but required more computational resources.

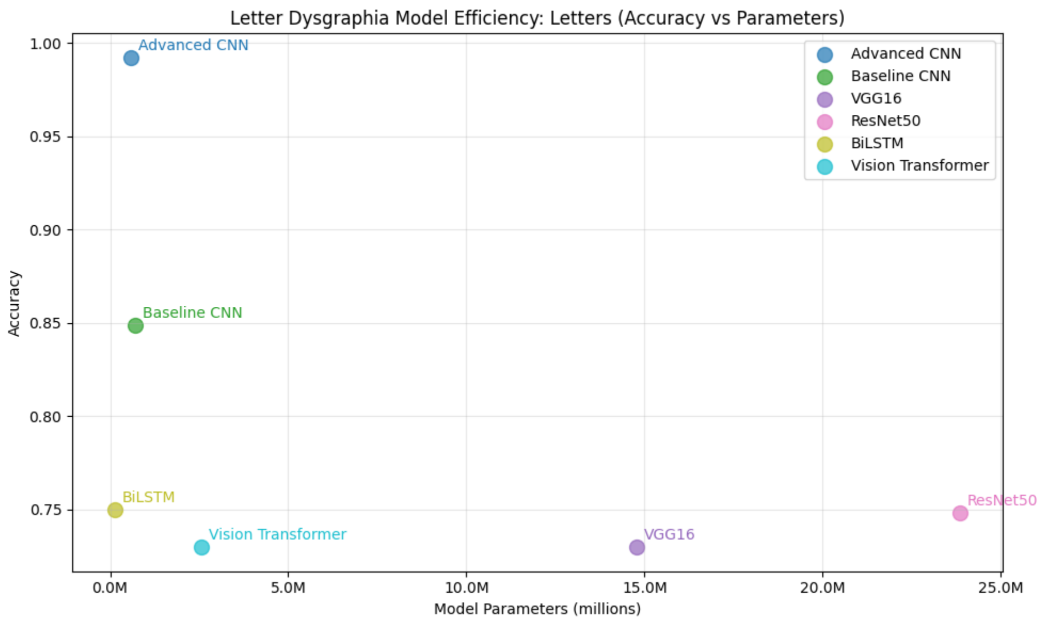


Fig. 5.3 Letter Dysgraphia Model Efficiency

The efficiency plot (Figure 5.3) illustrates the relationship between model complexity (parameters) and accuracy, showing that the Advanced CNN achieved near perfect accuracy with significantly fewer parameters, demonstrating superior efficiency.

Table 5.2 Letter Dysgraphia Performance Summary

Model	**Accuracy**
Advanced CNN	0.992
Baseline CNN	0.849
VGG16	0.730
ResNet50	0.748
BiLSTM	0.750
Vision Transformer	0.730

These findings validate that the proposed Advanced CNN delivers exceptional performance in letter dysgraphia classification, offering a robust, efficient, and deployable architecture suitable for integration into educational and clinical AI systems.

Please refer to **APPENDIX 9** for the screenshots of the deep learning model results comparison in letters and numbers.

5.3.2 Numeric Dysgraphia Detection

For numeric dysgraphia detection, a custom dataset was created by collecting 8,000 handwritten samples from 482 Sri Lankan children aged 6 to 10 years, encompassing diverse regional backgrounds. The same DL models were evaluated on this dataset to maintain consistency.

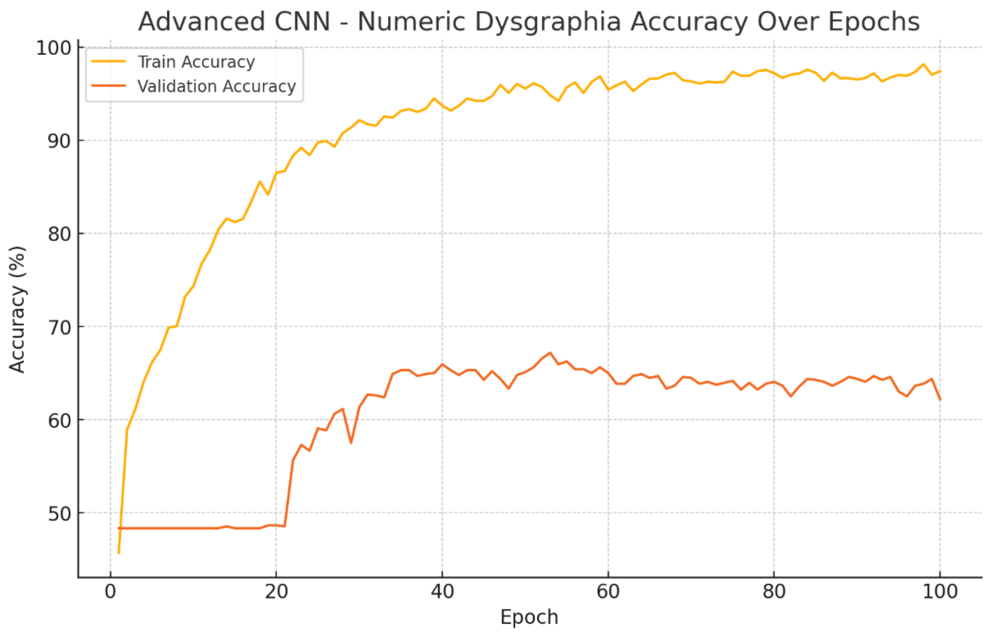


Fig. 5.4 Advanced CNN for Numbers

As illustrated in Figure 5.4, the training accuracy for the Advanced CNN steadily increased, reaching 97%, while validation accuracy stabilized around 64% after 100 epochs. Despite being lower than the letter based dataset, this is attributed to the greater variability and uneven stroke patterns present in children's numeric handwriting.

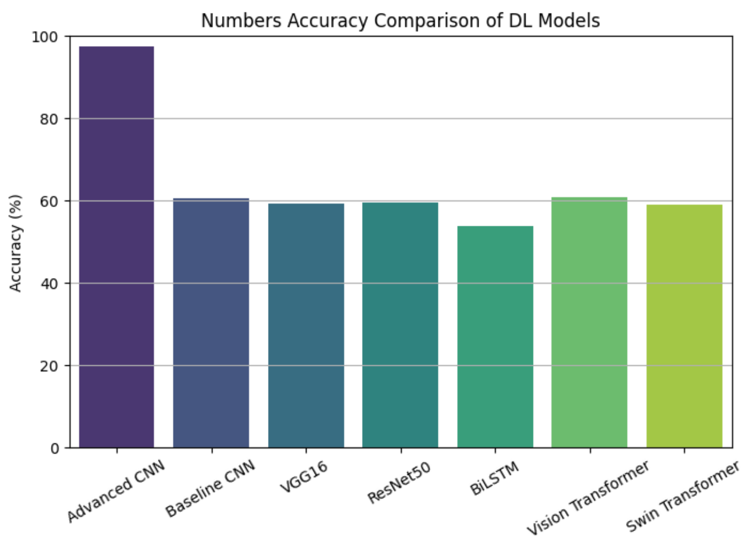


Fig. 5.5 Comparison of Deep Learning Models for Numbers

The bar graph (Figure 5.5) shows the model comparison, where the Advanced CNN achieved the highest accuracy (97.4%), followed by Baseline CNN (60.5%), Vision Transformer (60.8%), and Swin Transformer (58.9%). ResNet50 and VGG16 performed slightly better than BiLSTM but still lagged behind CNN-based models in precision and stability.

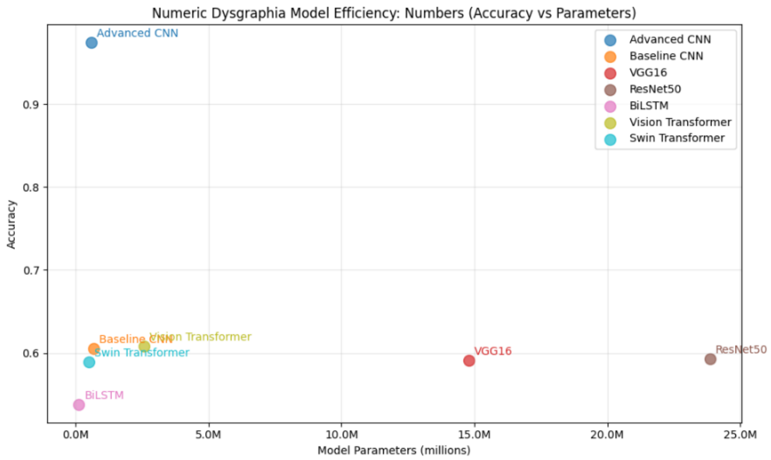


Fig. 5.6 Letter Dysgraphia Model Efficiency

The efficiency plot (Figure 5.6) demonstrates that the Advanced CNN achieved high accuracy with relatively fewer parameters, indicating a balanced tradeoff between computational efficiency and predictive power.

Table 5.3 Numeric Dysgraphia Performance Summary

Model	Accuracy
Advanced CNN	0.974
Baseline CNN	0.605
VGG16	0.591
ResNet50	0.593
BiLSTM	0.538
Vision Transformer	0.608
Swin Transformer	0.589

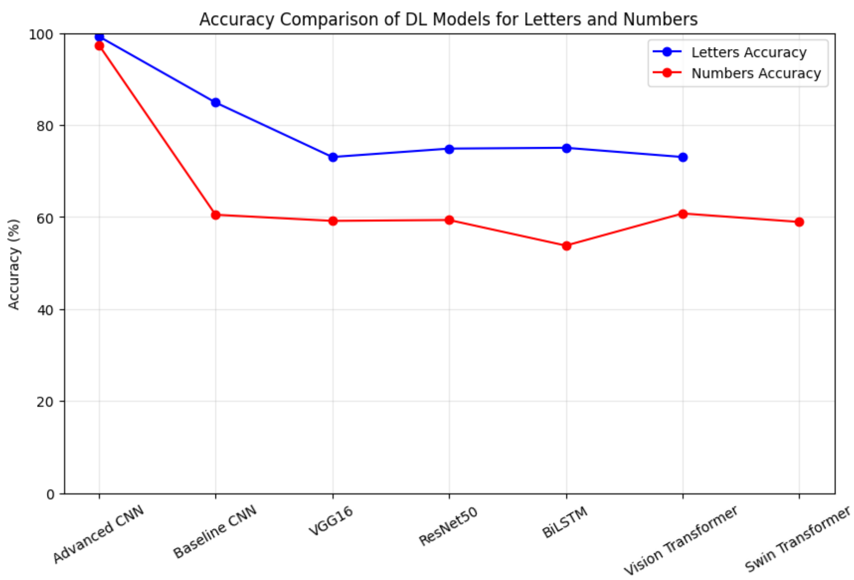


Fig. 5.7 Compares letter and numeric accuracies across all Deep Learning models

Finally, Figure 5.7 compares letter and numeric accuracies across all DL models, clearly demonstrating that while all architectures perform better on letters, the Advanced CNN maintains the most consistent advantage across both modalities.

Please refer to **APPENDIX 9** for the screenshots of the machine learning and deep learning models results comparison in letters and numbers.

5.4 Machine Learning Model Evaluation

In parallel, classical ML models were tested for baseline comparison. The algorithms included XGBoost, Random Forest, Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree.

5.4.1 Letter Dysgraphia Results

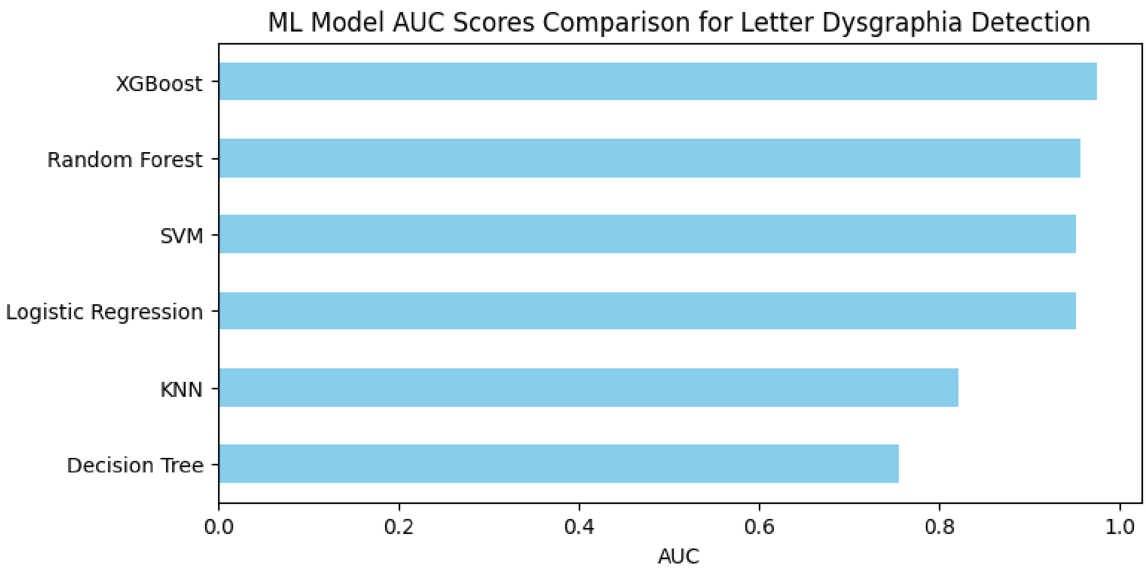


Fig. 5.8 ML Models AUC Comparison for Letters

As shown in Figure 5.8, the AUC comparison of ML models demonstrates that XGBoost (AUC = 0.975) achieved the highest performance, followed by Random Forest (0.957) and SVM (0.951). The Decision Tree and KNN models displayed comparatively lower AUCs. The detailed metrics table (**APPENDIX 9**) indicates that XGBoost achieved 92.91% accuracy, outperforming all other models. SVM and Random Forest closely followed with accuracies of 90.38% and 90.10%, respectively.

These results highlight that ensemble methods (XGBoost, Random Forest) effectively capture non-linear handwriting features, though their interpretability and scalability lag behind CNN architectures.

Please refer to **APPENDIX 9** for the screenshots of the machine learning model results in letters.

5.4.2 Numeric Dysgraphia Results

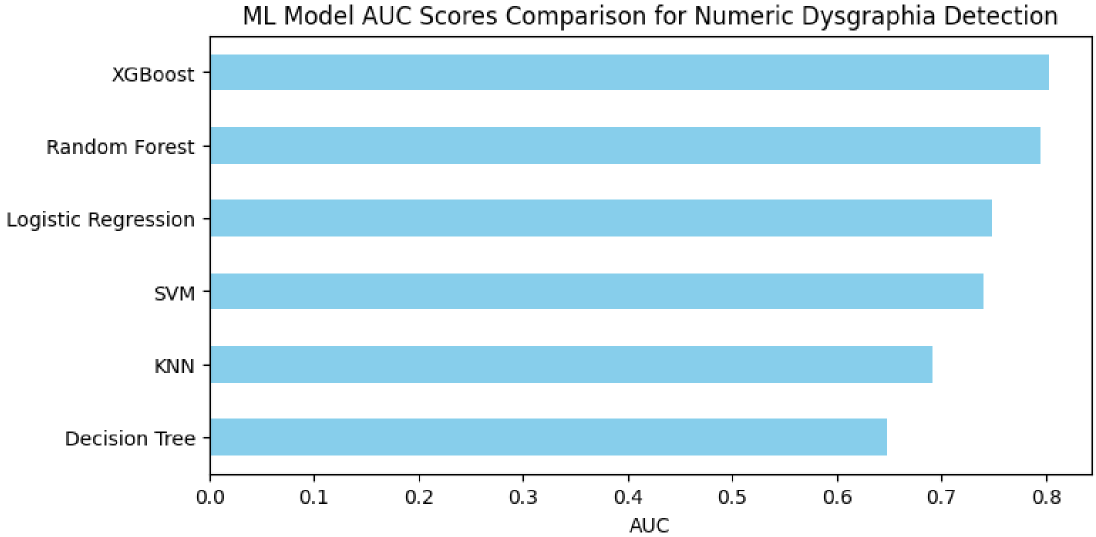


Fig. 5.9 ML Models AUC Comparison for Numbers

The numeric dysgraphia ML experiments followed the same evaluation procedure. As illustrated in Figure 5.9, XGBoost again performed best with AUC = 0.803, followed by Random Forest (0.794) and Logistic Regression (0.748). The detailed performance table (Figure 5.15) shows that XGBoost achieved the highest accuracy (71.94%), confirming its strength in handling moderately complex image derived numeric features.

However, traditional ML models generally underperformed compared to CNN based architectures, reaffirming that deep learning provides superior feature extraction and classification power for handwriting related tasks.

Please refer to **APPENDIX 9** for the screenshots of the machine learning model results in numbers.

5.5 Chapter Summary

Across all experiments, the proposed **Advanced CNN model** consistently achieved the highest accuracy, lowest test loss, and strongest AUC scores in both letter and numeric dysgraphia detection. It significantly outperformed baseline CNNs, transfer learning models, and traditional ML classifiers, demonstrating its robustness and adaptability to heterogeneous handwriting data.

The locally collected numeric dysgraphia dataset introduced in this study contributed a unique and valuable benchmark for future research in the Sri Lankan context. Additionally, the survey and field observations validated the system's societal need, confirming that most parents and educators lack access to early screening tools but show readiness to adopt AI based platforms.

Together, these results demonstrate that the proposed approach effectively bridges the gap between education and technology, offering a reliable, ethical, and accessible solution for early dysgraphia identification and intervention in Sri Lanka.

Chapter 6

Conclusions and Future Works

6.1 Chapter Overview

This chapter presents the concluding remarks of the research and outlines possible directions for future development. It revisits the study's main objectives, summarizes the key findings and achievements, and reflects on the contribution of this research to the field of AI driven learning disability detection and intervention. The chapter concludes with recommendations for future improvements, ensuring that the system remains adaptable and scalable in addressing dysgraphia within diverse learning contexts.

6.2 Conclusions

This study has presented an AI-driven framework for the early detection and intervention of dysgraphia among children aged six to ten years in Sri Lanka. The research has been motivated by the lack of awareness, datasets, and specialized educational technologies in the local context. Through the combination of data driven analysis, deep learning models, and interactive intervention design, this project has contributed a comprehensive solution that integrates detection, awareness, and rehabilitation in a unified platform.

A major achievement of this research has been the development of a localized numeric handwriting dataset, as no public dataset existed for numeric dysgraphia detection in Sri Lanka. The dataset has been collected, preprocessed, and annotated with high-quality standards to ensure representativeness and usability for future AI research. Alongside this, an Advanced Convolutional Neural Network (CNN) model has been implemented and optimized to classify dysgraphic handwriting patterns with exceptional accuracy, outperforming traditional CNNs, transfer learning models (VGG16, ResNet50), and other architectures such as BiLSTM and Vision Transformers. The model achieved 99.2% accuracy for letter dysgraphia and 97.4% for numeric dysgraphia, thereby validating its robustness and generalization across handwriting variations.

The study has also compared **machine learning** and **deep learning** approaches for both letter and numeric datasets. Classical ML models such as XGBoost, Random Forest, and SVM have shown moderate accuracy, while deep learning models significantly outperformed them in learning spatial and contextual handwriting features. This comparative analysis has demonstrated the superiority of CNN based architectures for real time dysgraphia detection systems.

From the human centered perspective, a nationwide survey and parent-teacher interviews have revealed an extensive lack of awareness regarding learning disorders. Over 92% of participants were unaware of specific learning disabilities, and 98% lacked knowledge of dysgraphia. Nevertheless, the findings have shown that parents and educators are open to adopting digital and AI based solutions, reinforcing the relevance and social demand for this system. The survey and observational data have been instrumental in shaping the platform's usability and intervention design, ensuring that it aligns with cultural and educational realities in Sri Lanka.

Another notable outcome of this study has been the implementation of an immersive AI based intervention system, featuring gamified learning modules and real-time voice interactive avatars. The system integrates a 3D avatar chatbot capable of lip synced speech and emotional engagement, enhancing user motivation and inclusivity. This intervention layer complements the detection model by providing personalized exercises for children to improve fine motor control, handwriting fluency, and numerical cognition. Furthermore, the inclusion of parent and teacher community features has created a holistic ecosystem that fosters collaboration and shared learning experiences.

Technically, the research has successfully integrated multiple technologies under a three-tier architecture React (frontend), Flask (application layer), and Firebase/MySQL (data layer) providing modularity, scalability, and real-time synchronization. The end-to-end system demonstrates practical feasibility for deployment in schools, clinics, and home environments.

In conclusion, this research has achieved its primary aim of developing a comprehensive AI based dysgraphia detection and intervention platform, contributing new datasets, models, and system designs to the field. It has not only advanced the technical understanding of handwriting disorder detection using deep learning but has also addressed critical social and educational gaps in Sri Lanka's context.

6.3 Limitations

Although the study has produced significant outcomes, certain limitations have been identified. The numeric handwriting dataset, while comprehensive, could be expanded to include more regional and age diversity to enhance generalizability. Additionally, due to resource constraints. The real time model inference and AI chatbot modules also require optimization for deployment on low end devices commonly used in schools. Despite these limitations, the research outcomes remain a strong foundation for future exploration and scaling.

6.4 Future Works

While the current system has achieved substantial success in detecting and supporting dysgraphia affected children, several promising directions for future work have been identified,

1. **Clinical Validation and Pilot Studies**

Future research should conduct large-scale pilot testing in collaboration with educational and healthcare institutions to clinically validate the detection accuracy and therapeutic effectiveness of the system.

2. **Dataset Expansion and Diversity**

The numeric and letter dysgraphia datasets can be expanded with more samples from different age groups, ethnic backgrounds, and writing instruments. This will enable more robust generalization of the AI models.

3. **Integration with Multilingual Support**

Extending the system to support Tamil and Sinhala languages will make it accessible to a wider audience across Sri Lanka and India.

4. **Adaptive Learning and Personalization**

Future versions can incorporate reinforcement learning or user feedback loops to automatically adjust activity difficulty and therapy recommendations based on child performance.

5. **Cloud and Mobile Optimization**

Deploying the system on cloud platforms or mobile applications will enhance accessibility for rural areas and low-resource educational settings.

6. **Teacher Dashboard and Analytics**

Introducing advanced dashboards with analytics, progress visualization, and individualized recommendations will empower teachers and therapists to monitor students' improvement effectively.

7. **Enhanced 3D Avatars and Emotion Recognition**

The AI chatbot could be enhanced with facial expression recognition and adaptive emotional responses to create a more engaging, empathetic interaction environment for children.

6.5 Concluding Remarks

This research has successfully demonstrated the potential of artificial intelligence to support early detection and intervention for dysgraphia among children in Sri Lanka and world. Through the creation of a locally relevant numeric handwriting dataset, the design of an advanced CNN based detection model, and the integration of immersive gamified interventions, the study has contributed meaningfully to both the technological and educational domains.

The system has been designed not merely as a diagnostic tool but as a holistic support platform empowering children, parents, and educators through awareness, community collaboration, and engaging learning experiences. The comparative evaluation of multiple deep learning and machine learning models has established the proposed Advanced CNN as an efficient and accurate approach for both letter and numeric dysgraphia detection.

Beyond its technical achievements, the study has addressed a pressing social issue the lack of awareness and early identification mechanisms for learning disabilities in Sri Lanka. By bridging this gap through an accessible AI driven system, the research has laid the groundwork for inclusive, technology enabled educational support.

In essence, this work represents an important step toward integrating machine intelligence with human centered design to enhance learning outcomes for children with special educational needs. With continued development, validation, and expansion, the system has the potential to become a valuable national asset, contributing to equitable access to education and early childhood intervention in Sri Lanka.

References

[1] I. Avishka, K. Kumarawadu, A. Kudagama, M. Weerathunga and S. Thelijjagoda, "Mobile App to Support People with Dyslexia and Dysgraphia," 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 2018, pp. 1-6, doi: 10.1109/ICIAFS.2018.8913335.

[2] J. Mekyska, M. Faundez-Zanuy, Z. Mzourek, Z. Galaz, Z. Smekal, and S. Rosenblum, "Identification and Rating of Developmental Dysgraphia by Handwriting Analysis," IEEE Trans Hum Mach Syst, vol. 47, no. 2, pp. 235–248, Apr. 2017, doi: 10.1109/THMS.2016.2586605.

[3] J. Kunhoth, S. Al-Maadeed, S. Kunhoth, and Y. Akbari, "Automated Systems For Diagnosis of Dysgraphia in Children: A Survey and Novel Framework," Jun. 2022, [Online]. Available: http://arxiv.org/abs/2206.13043

[4] I. Sazanita Isa, M. Aiman Zahir, S. Azura Ramlan, W. Li-Chih, and S. Noraini Sulaiman, "CNN Comparisons Models on Dyslexia Handwriting Classification," ESTEEM Academic Journal, vol. 17, pp. 12–25, 2021.

[5] J. Kunhoth, S. Al Maadeed, M. Saleh, and Y. Akbari, "CNN feature and classifier fusion on novel transformed image dataset for dysgraphia diagnosis in children," Expert Syst Appl, vol. 231, Nov. 2023, doi: 10.1016/j.eswa.2023.120740.

[6] P. Yogarajah and P. Braj, "Deep Learning Approach to Automated Detection of Dyslexia-Dysgraphia."

[7] S. A. Ramlan, I. S. Isa, A. P. Ismail, M. K. Osman, and Z. H. C. Soh, "Development of potential dysgraphia handwriting dataset," Data Brief, vol. 54, Jun. 2024, doi: 10.1016/j.dib.2024.110534.

[8] N. Kodikara, H. C. Fernando, sliitlk N. Kodikara, and S. Hewavitharana, "A Database for Handwriting Recognition Research in Sinhala Language," 2003. [Online]. Available: https://www.researchgate.net/publication/2939080

[9] I. S. Isa, W. N. Syazwani Rahimi, S. A. Ramlan, and S. N. Sulaiman, "Automated Detection of Dyslexia Symptom Based on Handwriting Image for Primary School Children," in Procedia Computer Science, Elsevier B.V., 2019, pp. 440–449. doi: 10.1016/j.procs.2019.12.127.

[10] M. Bublin et al., "Automated dysgraphia detection by deep learning with SensoGrip."

[11] R. Kariyawasam, M. Nadeeshani, T. Hamid, I. Subasinghe, P. Samarasinghe and P. Ratnayake, "Pubudu: Deep Learning Based Screening And Intervention of Dyslexia, Dysgraphia And Dyscalculia," 2019 14th Conference on Industrial and Information

Systems (ICIIS), Kandy, Sri Lanka, 2019, pp. 476-481, doi: 10.1109/ICIIS47346.2019.9063301.

[12] R. C. . Gonzalez and R. E. . Woods, Digital image processing. Pearson, 2018.

[13] L. Shapiro and G. Stockman, Computer Vision. 2000. Available: https://theswissbay.ch/pdf/Gentoomen%20Library/Artificial%20Intelligence/Computer%20Vision/Computer%20Vision%20-%20Linda%20Shapiro.pdf

[14] Silva, S., Bandara, C., Maduranga, P., & Uvais, M. I. (2025, September). *Developing a comprehensive handwriting dataset for early detection of numeric dysgraphia across Sri Lanka*.researchgate.net. https://www.researchgate.net/publication/395246401_Developing_a_Comprehensive_Handwriting_Dataset_for_Early_Detection_of_Numeric_Dysgraphia_Across_Sri_Lanka

[15] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: https://doi.org/10.1109/tsmc.1979.4310076

[16] G. Richard and M. Serrurier, "Dyslexia and Dysgraphia prediction: A new machine learning approach."

[17] L. Bartha-Doering et al., "Developmental surface dyslexia and dysgraphia in a child with corpus callosum agenesis: an approach to diagnosis and treatment," Cogn Neuropsychol, pp. 1–23, Jun. 2024, doi: 10.1080/02643294.2024.2368876.

[18] M. Ikermane and A. El Mouatasim, "Digital handwriting characteristics for dysgraphia detection using artificial neural network," Bulletin of Electrical Engineering and Informatics, vol. 12, no. 3, pp. 1693–1699, 2023, doi: 10.11591/eei.v12i3.4571.

[19] K. Zolna et al., "The Dynamics of Handwriting Improves the Automated Diagnosis of Dysgraphia," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.07576

[20] D. Vydeki, D. Bhandari, P. P. Patil, and A. A. Kulkarni, "Towards Accessible Learning: Deep Learning-Based Potential Dysgraphia Detection and OCR for Potentially Dysgraphic Handwriting."

APPENDIX 1 : ETHICAL CLEARANCE CERTIFICATE AND ETHICAL APPROVAL LETTER

ETHICAL CLEARANCE CERTIFICATE

SLTC Research University

Ethical Clearance Certificate

Certificate Reference Number: ERC-2024-8-RS-01

Date of the Certificate: 26.11.2024

Project Title: AI - Powered Screening and Intervention for Dysgraphia in Sinhala-Speaking Children Using Neural Network-Based Multi-Platform Application

Nature of Project: Undergraduate Research

Principal Researcher / Supervisor: Dr.Pasan Maduranga

Investigators: Sahan Silva, Chathuranga Bandara,Yasara Fernando

Validity Period: 26.11.2024 to 25.11.2025

On behalf of the Ethics Review Committee (ERC) of Sri Lanka Technology Campus, I hereby grant ethical clearance approval in respect of the undertakings contained in the above-mentioned project and research instrument(s). Should any other instruments be used, these require separate authorization. The researchers may therefore commence with the research as from the date of this certificate, using the reference number indicated above.

Please note that the ERC must be informed immediately of;

- Any material changes in the conditions or undertakings mentioned in the document.
- Any material breaches of ethical undertakings or events that impact upon the ethical conduct of the research.
- Any unanticipated events involving potential risks to study subjects.

The ERC retains the right to

- Withdraw or amend this Ethical Clearance Certificate if
  - ✓ Any unethical principal or practices are revealed or suspected
  - ✓ Relevant information has been withheld or misrepresented
  - ✓ Regulatory changes of whatsoever nature so require
  - ✓ The conditions contained in the Certificate have not been adhered to
- Request access to any information or data at any time during the course or after the completion of the project.

The Principal Researcher must report to the ERC, where applicable in respect of ethical compliance. Final report of the study should be submitted within three months of completion of the study.

Thank you.
Yours Sincerely,

....................................,

Dr. Sampath Deegalla
Chairperson

ETHICAL APPROVAL LETTER

SLTC Research University, Ingiriya Road, Padukka, Sri Lanka.

+94 112 100 500 | info@sltc.ac.lk
www.sltc.ac.lk

Ref. No: ERC-2024-8-RS-01

26.11.2024

Dr.Pasan Maduranga,
Faculty of Computing & IT,
SLTC Research University,
Padukka

Ref: Approval for ethical clearance for a study titled AI - Powered Screening and Intervention for Dysgraphia in Sinhala-Speaking Children Using Neural Network-Based Multi-Platform Application

Reference is made to the above heading.

I am pleased to inform you that the ethical clearance committee of SLTC Research University, approved the ethical clearance of the above-mentioned study.

The validity of this ethical clearance is one year effective from 25/11/2025. Please find the the clearance certificate.

You will be required to apply for renewal of ethical clearance on a yearly basis if the study is not completed at the end of this clearance. You will be expected to provide a progress report in the mid of the study and the Final report of the study should be submitted within three months of completion of the study.

The Ethics Committee wish you well in your research.

Yours Sincerely,

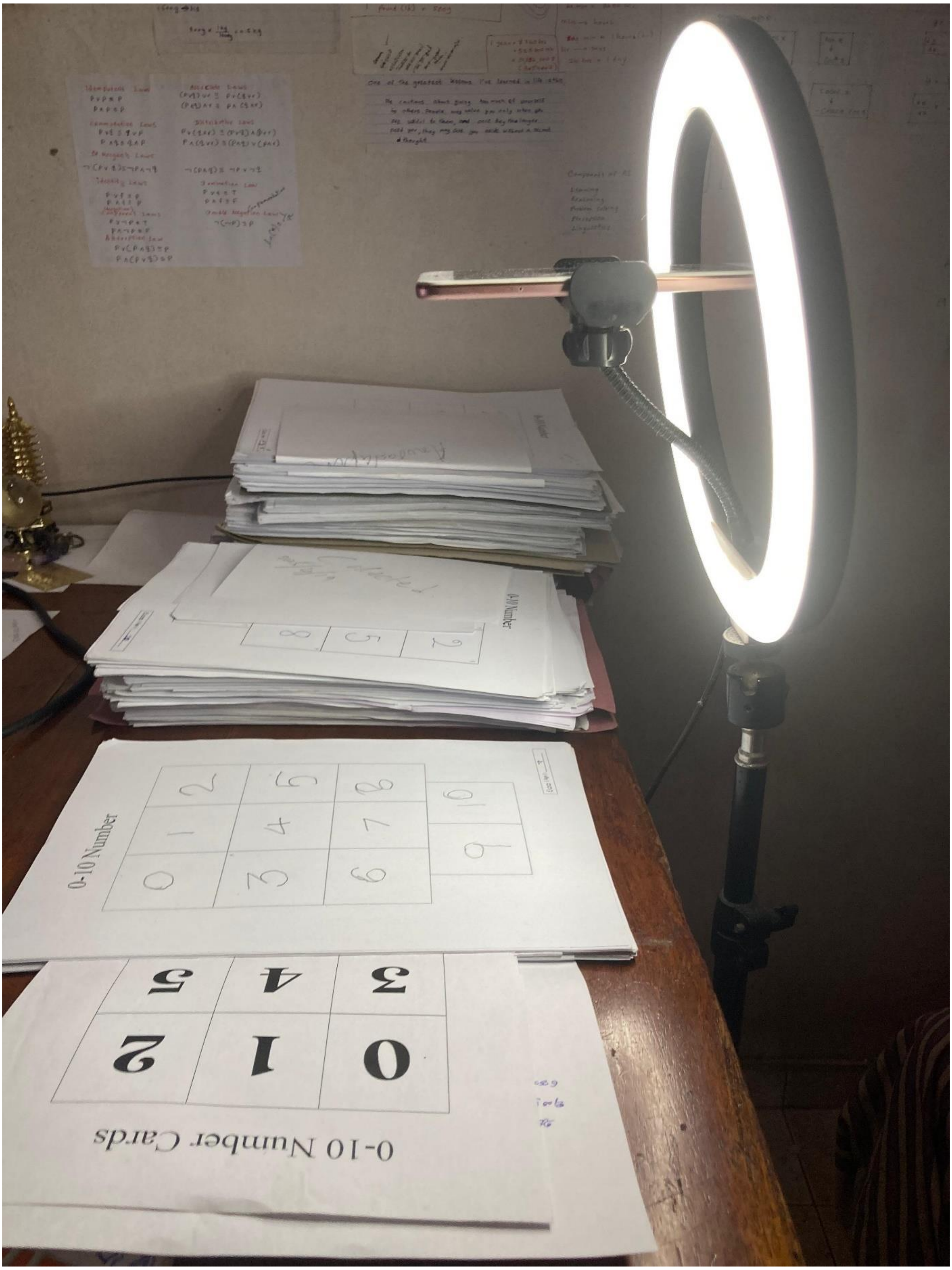
....

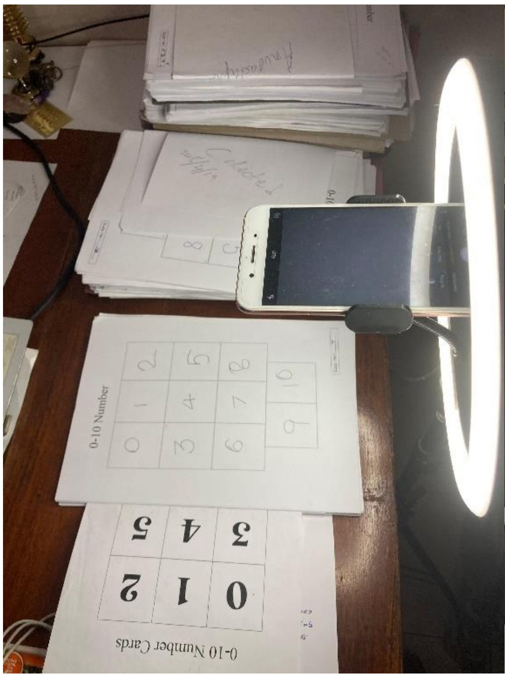
Dr. Sampath Deegalla,
Chairperson, ERC

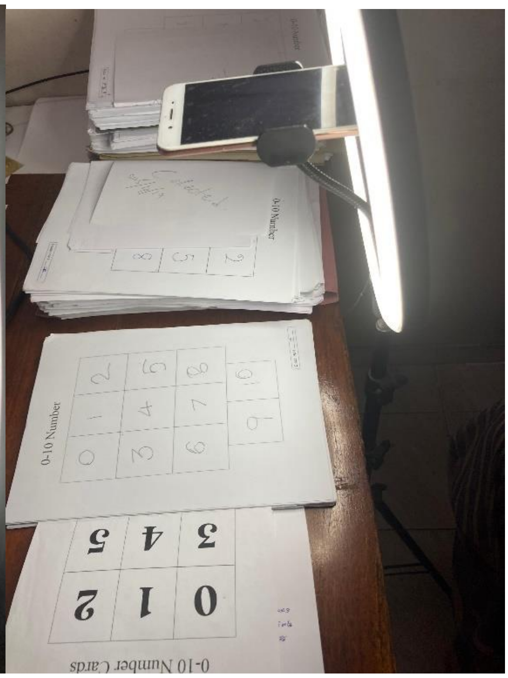
Copy to:

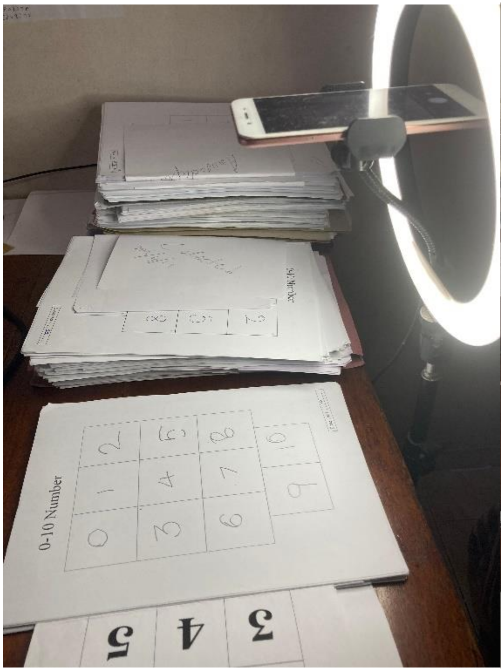
- Vice Chancellors Office, SLTC Research University
- Director, Academic Affairs, SLTC Research University
- Dean, Faculty of Postgraduate & Research
- Lead, Office of Research and Innovation Services, SLTC

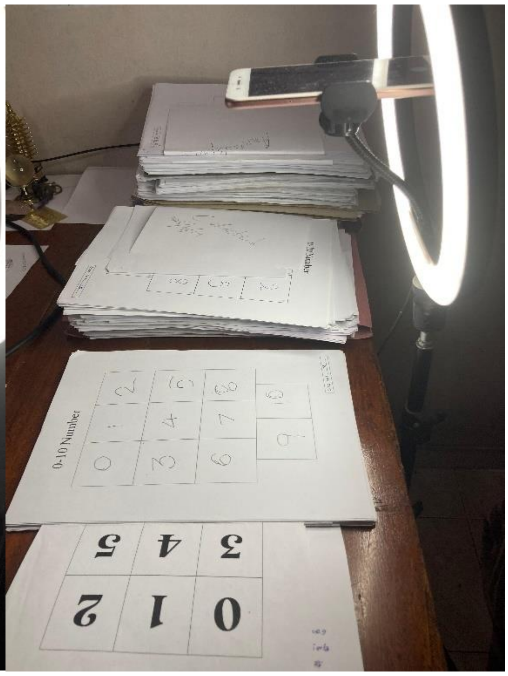
APPENDIX 2 : DATA COLLECTION

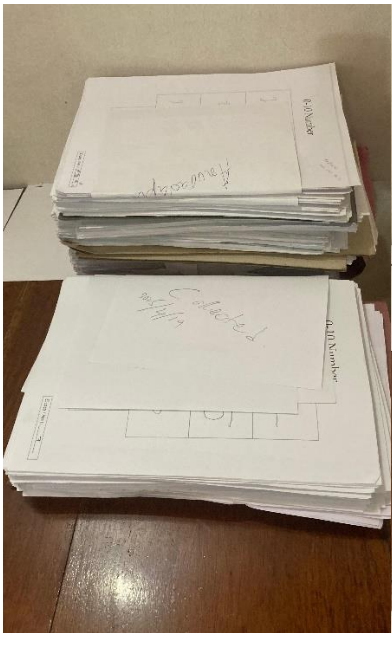


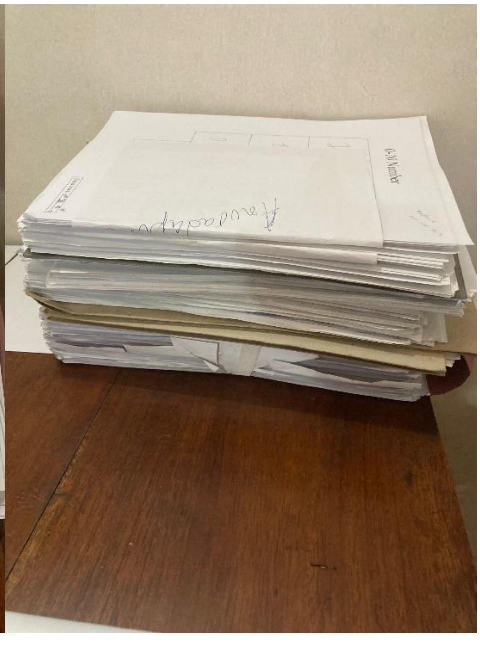

0-10 Number
0-10 Number Cards


0-10 Number
0-10 Number Cards


0-10 Number


0-10 Number





APPENDIX 3 : CONSENT FORM AND ASSENT FORM

CONSENT FORM FOR PARENTS

SLTC Research University

Consent Form - Ethical Clearance Committee (ECC)

1. Ethical Clearance Ref No:	**ERC-2024-8-NS-01**
2. Title of the project:	AI - Powered Screening and Intervention for Dysgraphia in Sinhala-Speaking Children Using Neural Network-Based Multi-Platform Application

3. **Consent of the Participant** (To be completed by the participant (Please tick the appropriate box))

Description	Yes	No
Have you read the information sheet which is in your mother language?		
Have you had an opportunity to discuss this study and ask questions?		
Have you received enough information about the study?		
Have you had satisfactory answers to all your questions?		
Sections of your medical notes, including those held by the investigators relating to your participation in this study may be examined by other research assistants. All personal details will be treated as TRICTLY CONFIDENTIAL. Do you give your permission for these individuals to have access to your records?		
Do you understand that you are free to withdraw from the study at any time?		
Have you had sufficient time to come to your decision regarding participation?		
Do you agree to take part in this study?		

Who explained the study to you?..

Full name (In Block Capital) : ....

....

....

Address: ....

....

....

Signature of the participant:.. Date:...

4. To be completed by the investigator/ person obtaining consent

I have explained the study to the above participant and he/she has indicated his/her willingness to take part in this study.

Full name (In Block Capital):...

....

....

Signature of Investigator:.. Date:...

ASSENT FORM FOR CHILDREN

AI - Powered Screening and Intervention for Dysgraphia in Sinhala Speaking Children Using Neural Network-Based Multi-Platform Application

Child Assent Form

I have been told that my parents have said it's okay for me to participate, if I want to, in a project about finding out how children like me write numbers and how we can improve them if needed.

I know that I can stop at any time I want to and it will be okay if I want to stop.

______________________ ________________

Name Date

____

Signature
(Parents, legal guardians or a legally authorized official)

Investigators – Sahan Silva, Chathuranga Bandara

Instructions of the project

Participation

- Children will be asked to write numbers in a structured manner. (0 to 10)

Parental/Guardian Involvement

- Parents or guardians must provide consent for their child to participate in this study.
- Additionally, children must give their assent to take part in the project.

APPENDIX 4 : DATASET VALIDATION JUSTIFICATION FROM LRH AND SERRIC

SLTC
Research University

30th November 2024

The Director
Lady Ridgeway Hospital for Children,
Dr. Denister De Silva Mawatha,
Colombo 08

Approved

Dr. G. Wijesuriya
Director
Lady Ridgeway Hospital for Children
Colombo 08.

Dear Sir/Madam,

Request for Permission to Collect Numeric Handwriting Dataset

I am writing to formally request permission to collect a numeric handwriting dataset from the Lady Ridgeway Hospital for Children (LRH) for our ongoing research project, titled **"AI-Powered Screening and Intervention for Dysgraphia in Sinhala Speaking Children Using a Neural Network-Based Multi-Platform Application."**

Our research aims to develop a multi-platform application to screen for and provide intervention activities for dysgraphia, specifically focusing on Sinhala-speaking children. This innovative approach leverages artificial intelligence to assist parents, teachers, and healthcare professionals in the early identification of and targeted interventions for children with dysgraphia.

We have obtained ethical clearance for this study from the Sri Lanka Technological Campus (SLTC) and are currently in the process of building a comprehensive dataset.

We kindly request your approval for the following:

- Permission to collect numeric handwriting samples from children with your organization's support.
- Collaboration with your team to ensure that all data collection aligns with your institution's guidelines and protocols.
- We assure you that the data collection process will be conducted with the utmost confidentiality and adherence to ethical guidelines. Consent will be obtained from parents or guardians before participation, and no personal identifiers will be included in the dataset.

Further, **Mr. Sahan Chamara Silva**, a research student of our research team, will be in contact with you to coordinate the next steps.

Your support in this initiative will be instrumental in advancing research and developing a tool that could significantly benefit children in Sri Lanka.

Thank you for considering our request.

Yours sincerely,

Dr. M.W.P. Maduranga
Senior Lecturer/ Supervisor to the Project
SLTC Research University

SLTC Research University

30th November 2024

The Director
Senehasa Education Resource Research & Information Institute (SERRIC)
Matha Road, Manning Town
Colombo

Dear Sir/Madam,

Request for Permission to Collect Numeric Handwriting Dataset

I am writing to formally request permission to collect a numeric handwriting dataset from the Senehasa Education Resource Research & Information Institute (SERRIC) for our ongoing research project, titled **"AI-Powered Screening and Intervention for Dysgraphia in Sinhala-Speaking Children Using a Neural Network-Based Multi-Platform Application."**

Our research aims to develop a multi-platform application to screen for and provide intervention activities for dysgraphia, specifically focusing on Sinhala-speaking children. This innovative approach leverages artificial intelligence to assist parents, teachers, and healthcare professionals in the early identification of and targeted interventions for children with dysgraphia.

We have obtained ethical clearance for this study from the Sri Lanka Technological Campus (SLTC) and are currently in the process of building a comprehensive dataset.

We kindly request your approval for the following:

- Permission to collect numeric handwriting samples from children with your organization's support.
- Collaboration with your team to ensure that all data collection aligns with your institution's guidelines and protocols.
- We assure you that the data collection process will be conducted with the utmost confidentiality and adherence to ethical guidelines. Consent will be obtained from parents or guardians before participation, and no personal identifiers will be included in the dataset.

Further, **Mr. Sahan Chamara Silva**, a research student of our research team, will be in contact with you to coordinate the next steps.

Your support in this initiative will be instrumental in advancing research and developing a tool that could significantly benefit children in Sri Lanka.

Thank you for considering our request.

Yours sincerely,

Dr. M.W.P. Maduranga
Senior Lecturer/ Supervisor to the Project
SLTC Research University

Commandant
Brigadier
Senehasa Education Resource Research &
Information Centre
NARAHENPITA

APPENDIX 5 : DATA COLLECTION SHEET

0-10 Number

වයස (Age) : ____________

කොටුවේ "✓" හරි ලකුණ යොදන්න	කවදාවත් නැහැ	කලාතුරකින්	සමහර වෙලාවට	නිතර	සෑම විටම
අකුරු සහ අංක ලිවීම					
අවුල් සහගත හෝ කියවීමට අපහසු අත් අකුරු තිබේද?					
සංඛ්‍යා ආපසු හරවා ලියා තිබේද (උදා: 18 වෙනුවට 81 ලිවීම)?					
අකුරු පිටුපසට (ආපසු හරවා) ලියා තිබේද?					
ඉලක්කම් පැහැදිලිව ලිවීමට සහ ඒවා අකුරුවලින් වෙනස් කිරීමට අපහසු වේද?					
සටහන් පොතක හෝ ඉරි ඇඳ ඇති කඩදාසියක ඔබට රේඛා තුළ ලිවීමට අපහසුද?					
අක්ෂර සැකසීම					
වචන සහ/හෝ වචන අතර අකුරු නිවැරදිව පරතරයක් තැබීමේ ගැටලුවක් තිබේද?					
ඔබ අකුරු හෝ ඉලක්කම් විශාල ලෙස ලියනවාද?					
විරාම ලකුණු සමඟ වාක්‍ය අවසන් කිරීමේ ගැටලුවක් තිබේද?					
ගණිත ගැටලුවක් කරන විට තීරු වල පහලට පහලට ලිවීමට අපහසුද?					
වමේ සිට දකුණට අකුරු ලිවීමේ අපහසුවක් තිබේද?					
ලිවීමේ ප්‍රවීණතාවය					
ලිවීමේ කාර්යයට විරුද්ධද?					
සිතුවිලි කඩදාසි මත ලිවිමට අපහසුද?					
අකුරු නිවැරදිව බලා ලිවීමේ අපහසුතාවයක් තිබේද?					
තනිවම ලිවීමේදී වැඩ සම්පූර්ණ කිරීමේ ගැටලුවක් තිබේද?					
ගෙදර වැඩ පැවරුම්වල අක්ෂර වින්‍යාස දෝෂ තිබේද ?					
චාලක කුසලතා සහ සියුම් චාලක සම්බන්ධීකරණය					
ඔබට පැන්සලක් හෝ පෑනක් අල්ලාගෙන සිටීමට අපහසුද ?					
ලිවීමෙන් හෝ ඇඳීමෙන් ඔබේ අත රිදෙනවාද නැත්නම් වෙහෙසට පත් වෙනවාද?					
ඔබට කුඩා වස්තු අහුලා ගැනීමේ අපහසුතාවයක් තිබේද ?					
ඔබට කතුර භාවිතා කිරීම දී , ඇඳුම් බොත්තම් දැමීම දී හෝ සිපර් භාවිතා කිරීමේ දී ගැටලුවක් තිබේද?					
ඔබේ අත අනෙක් අත සමග එකිනෙකට එක වගේ වැඩ කිරීමෙන් මිදෙනවාද ?					

APPENDIX 6 : DOMAIN KNOWLEDGE IN SPECIAL NEEDS EDUCATION AND DATA COLLECTION FROM CHILDREN


Alison
EMPOWER YOURSELF

CERTIFICATE

SAHAN SILVA

has received this award for successfully completing the course:

Teaching Special Needs Education

CPD CERTIFIED
The CPD Certification Service

To verify:

https://alison.com/certification/check/61a215d16d

25th October, 2024
Date of Award

Maeve Richardson
Director of Certification


unicef for every child
Global NUTRITION CLUSTER
Global Education Cluster
WASH Cluster Water Sanitation Hygiene
Child Protection

Certificate of Completion

Sahan Silva

has successfully completed

Child participation principles, methods and tools

by

UNICEF

21 October 2024

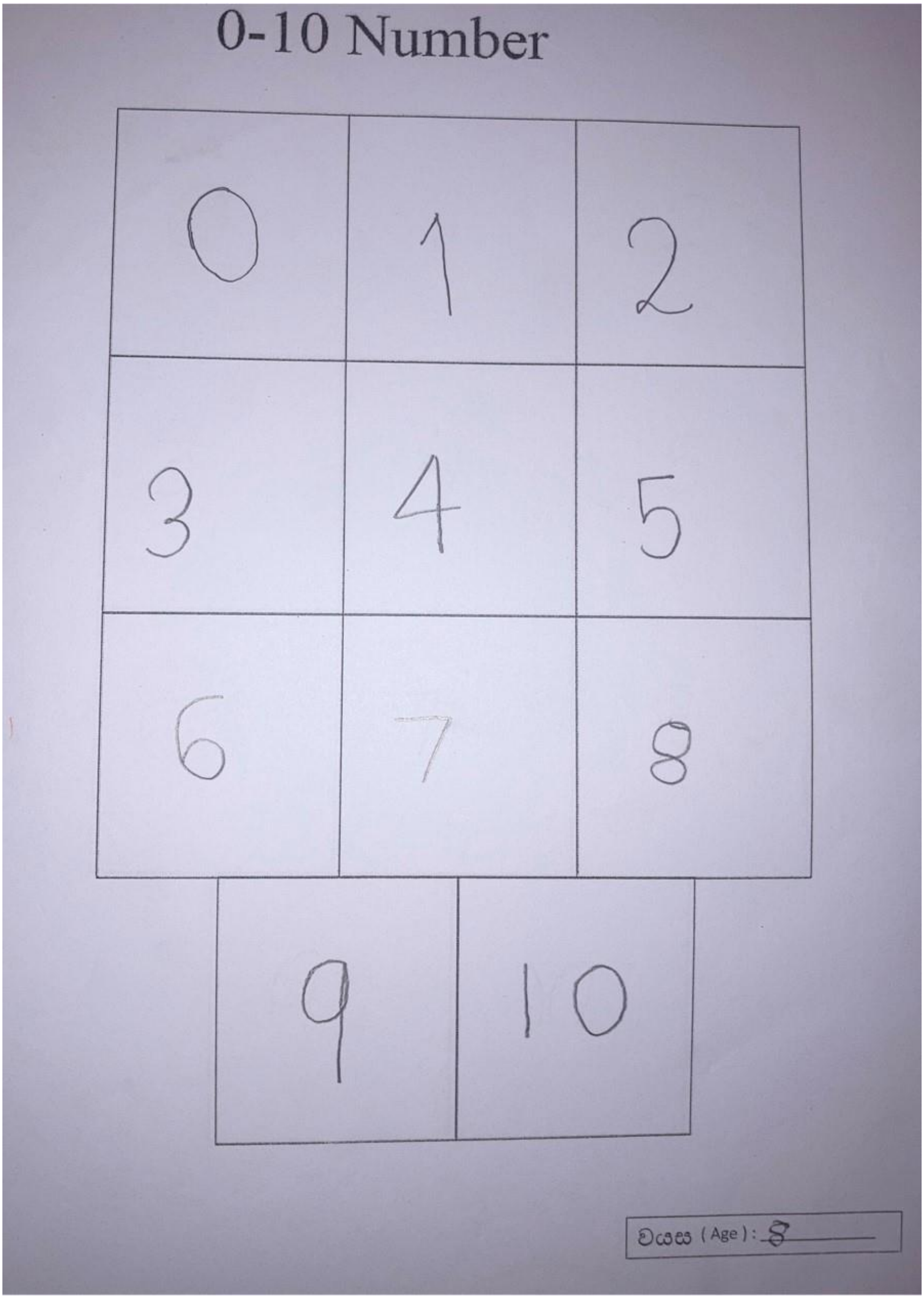
Nisar Syed
Chief, Global Cluster Coordination Unit
Office of Emergency Programmes (EMOPS), UNICEF

RuI3MPSo9G

AGORA

Note: This certificate is issued by UNICEF through the Agora platform. It may not be recognized by other institutions – including third party vendors or universities from which content may be offered in this course or programme.

APPENDIX 7 : SAMPLE IMAGE OF COLLECTED



කොටුවේ "✓" හරි ලකුණ යොදන්න	කවදාවත් නැහැ	කලාතුරකින්	සමහර වෙලාවට	නිතර	සෑම විටම
අකුරු සහ අංක ලිවීම					
අවුල් සහගත හෝ කියවීමට අපහසු අත් අකුරු තිබේද?		✓			
සංඛ්‍යා ආපසු හරවා ලියා තිබේද (උදා: 18 වෙනුවට 81 ලිවීම)?	✓				
අකුරු පිටුපසට (ආපසු හරවා) ලියා තිබේද?	✓				
ඉලක්කම් පැහැදිලිව ලිවීමට සහ ඒවා අකුරුවලින් වෙනස් කිරීමට අපහසු වේද?	✓				
සටහන් පොතක හෝ ඉරි ඇඳ ඇති කඩදාසියක ඔබට රේඛා තුළ ලිවීමට අපහසුද?			✓		
අක්ෂර සැකසීම					
වචන සහ/හෝ වචන අතර අකුරු නිවැරදිව පරතරයක් තැබීමේ ගැටලුවක් තිබේද?	✓				
ඔබ අකුරු හෝ ඉලක්කම් විශාල ලෙස ලියනවාද?			✓		
විරාම ලකුණු සමඟ වාක්‍ය අවසන් කිරීමේ ගැටලුවක් තිබේද?	✓				
ගණිත ගැටලුවක් කරන විට තීරු වල පහලට පහලට ලිවීමට අපහසුද?	✓				
වමේ සිට දකුණට අකුරු ලිවීමේ අපහසුවක් තිබේද?	✓				
ලිවීමේ ප්‍රවීණතාවය					
ලිවීමේ කාර්යයට විරුද්ධද?	✓				
සිතුවිලි කඩදාසි මත ලිවීමට අපහසුද?			✓		
අකුරු නිවැරදිව බලා ලිවීමේ අපහසුතාවයක් තිබේද?	✓				
තනිවම ලිවීමේදී වැඩ සම්පූර්ණ කිරීමේ ගැටලුවක් තිබේද?	✓				
ගෙදර වැඩ පැවරුම්වල අක්ෂර වින්‍යාස දෝෂ තිබේද ?		✓			
චාලක කුසලතා සහ සියුම් චාලක සම්බන්ධීකරණය					
ඔබට පැන්සලක් හෝ පෑනක් අල්ලාගෙන සිටීමට අපහසුද ?	✓				
ලිවීමෙන් හෝ ඇඳීමෙන් ඔබේ අත රිදෙනවාද නැත්නම් වෙහෙසට පත් වෙනවාද?			✓		
ඔබට කුඩා වස්තු අහුලා ගැනීමේ අපහසුතාවයක් තිබේද ?	✓				
ඔබට කතුර භාවිතා කිරීම දී , ඇඳුම් බොත්තම් දැමීම දී හෝ සිපර් භාවිතා කිරීමේ දී ගැටලුවක් තිබේද?	✓				
ඔබේ අත අනෙක් අත සමඟ එකිනෙකට එක වගේ වැඩ කිරීමෙන් මිදෙනවාද ?					✓

APPENDIX 8 : MODEL SCREENSHOTS

Deep Learning Model Comparison in Letter Dysgraphia

Letters_DL_COMPARISON updated.ipynb

Letters_DL_COMPARISON updated.ipynb > # Deep Learning Model Comparison - Complete Analysis

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Select Kernel

```python
        print(f"  {key}: {value}")

# Cell 3: Data Loading and Preprocessing
def load_and_prepare_data():
    """Load and prepare training and testing datasets"""
    try:
        # Load CSV files
        train_df = pd.read_csv("Letters_training_dataset.csv")
        test_df = pd.read_csv("Letters_testing_dataset.csv")

        print(f"Training data shape: {train_df.shape}")
        print(f"Testing data shape: {test_df.shape}")

        # Display basic statistics
        print("\nTraining data info:")
        print(train_df.info())
        print("\nLabel distribution in training data:")
        print(train_df['Label'].value_counts())

        print("\nLabel distribution in test data:")
        print(test_df['Label'].value_counts())

        return train_df, test_df

    except FileNotFoundError as e:
        print(f"Error loading data: {e}")
        print("Please ensure the CSV files are in the current directory")
        return None, None

# Load data
train_df, test_df = load_and_prepare_data()

# Cell 4: Data Generators Setup
def create_data_generators(train_df, test_df):
    """Create data generators for training and testing"""
```

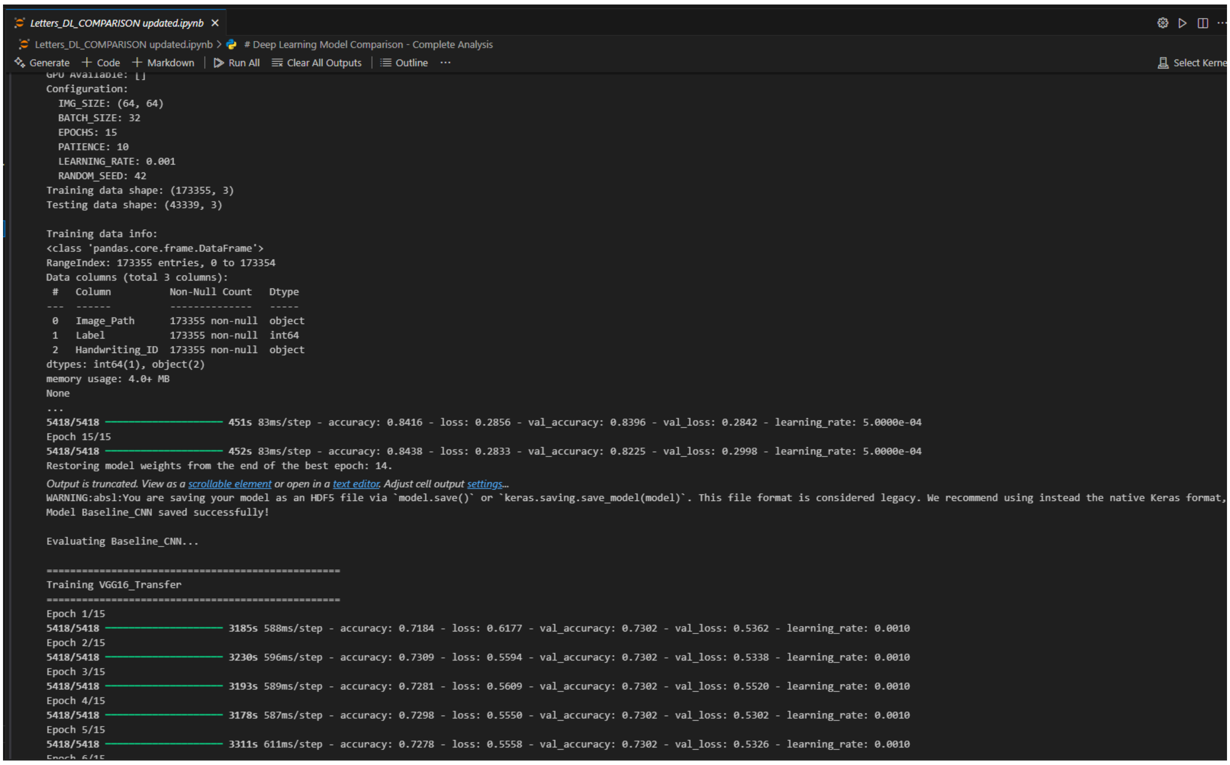
[3]

```python
def create_vgg16_transfer():
    """Create VGG16 transfer learning model"""
    base_vgg = VGG16(weights="imagenet", include_top=False, input_shape=(64,64,3))
    base_vgg.trainable = False

    inp = Input((64,64,1))
    x = tf.keras.layers.Concatenate()([inp, inp, inp])  # Convert to 3 channels
    x = vgg_preprocess(x)
    x = base_vgg(x, training=False)
    x = GlobalAveragePooling2D()(x)
    x = Dense(128, activation="relu")(x)
    x = Dropout(0.5)(x)
    x = Dense(64, activation="relu")(x)
    x = Dropout(0.3)(x)
    out = Dense(1, activation="sigmoid")(x)

    model = Model(inp, out)
    model.compile(
        optimizer=Adam(learning_rate=CONFIG['LEARNING_RATE']),
        loss="binary_crossentropy",
        metrics=["accuracy"]
    )

    return model

def create_resnet50_transfer():
    """Create ResNet50 transfer learning model"""
    base_res = ResNet50(weights="imagenet", include_top=False, input_shape=(64,64,3))
    base_res.trainable = False

    inp = Input((64,64,1))
    x = tf.keras.layers.Concatenate()([inp, inp, inp])  # Convert to 3 channels
    x = resnet_preprocess(x)
    x = base_res(x, training=False)
    x = GlobalAveragePooling2D()(x)
    x = Dense(128, activation="relu")(x)
```

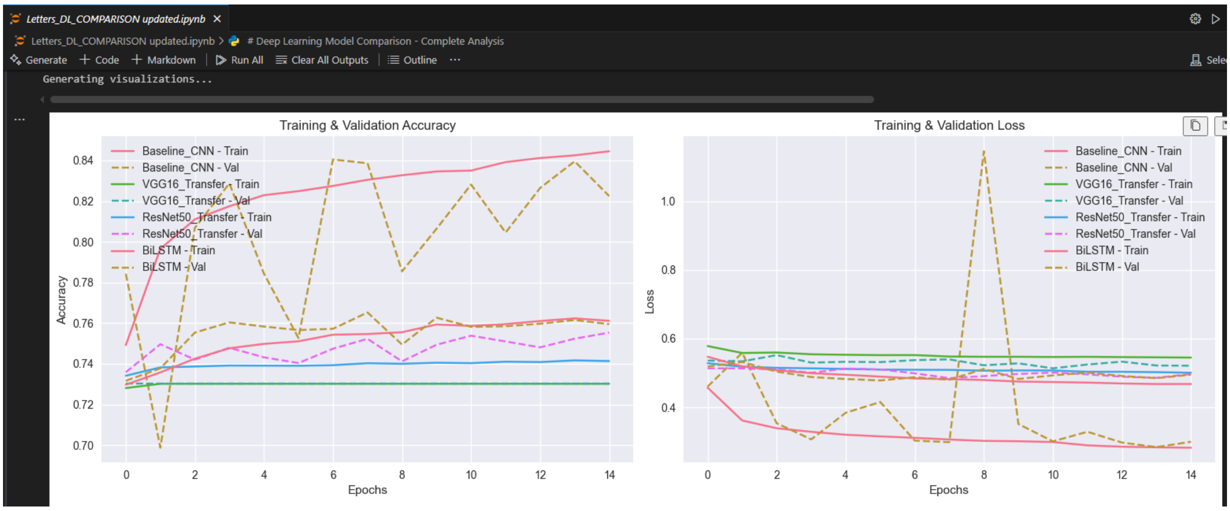
[3]

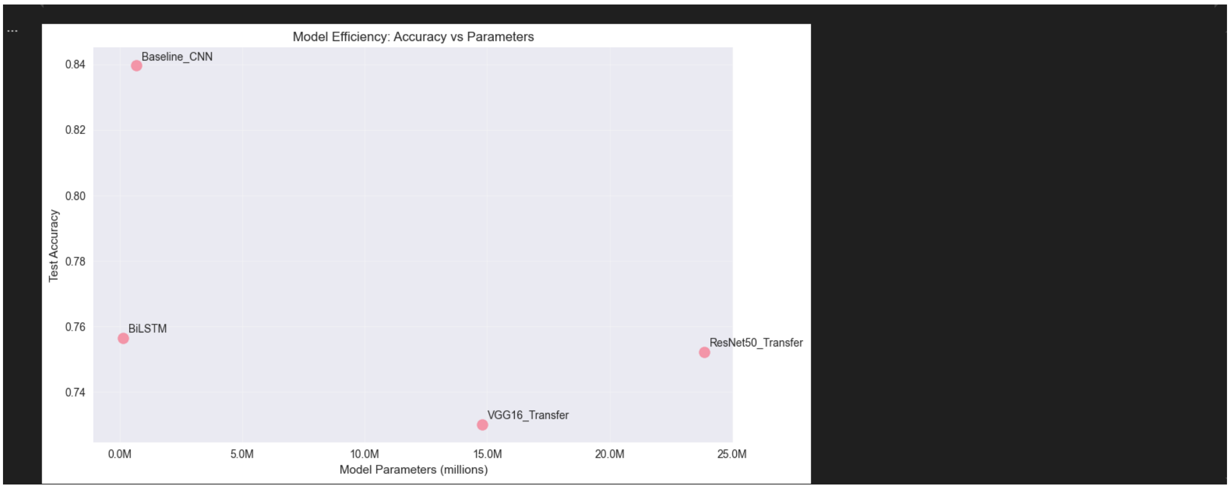
```
    return model

# Cell 6: Model Training Function
def train_model(model, model_name, train_gen, test_gen):
    """Train a model and return training history"""
    print(f"\n{'='*50}")
    print(f"Training {model_name}")
    print(f"{'='*50}")

    # Callbacks
    callbacks = [
        EarlyStopping(
            monitor='val_loss',
            patience=CONFIG['PATIENCE'],
            restore_best_weights=True,
            verbose=1
        ),
        ReduceLROnPlateau(
            monitor='val_loss',
            factor=0.5,
            patience=3,
            min_lr=1e-7,
            verbose=1
        )
    ]

    # Train the model
    history = model.fit(
        train_gen,
        validation_data=test_gen,
        epochs=CONFIG['EPOCHS'],
        callbacks=callbacks,
        verbose=1
    )

    # Save the model
    model.save(f"trained_models/{model_name}.h5")
    print(f"Model {model_name} saved successfully!")

    return history

# Cell 7: Model Evaluation Functions
def evaluate_model(model, model_name, train_gen, test_gen):
    """Evaluate model performance"""
    print(f"\nEvaluating {model_name}...")

    # Handle autoencoder differently (it only returns loss, not accuracy)
```

```
# Cell 9: Main Execution - Model Creation and Training
def main():
    """Main execution function"""
    # Initialize models
    models = {
        'Baseline_CNN': create_baseline_cnn(),
        'VGG16_Transfer': create_vgg16_transfer(),
        'ResNet50_Transfer': create_resnet50_transfer(),
        'BiLSTM': create_bilstm_model(),
        'Autoencoder': create_autoencoder()
    }

    # Print model summaries
    for name, model in models.items():
        print(f"\n{name} Architecture:")
        print(f"Total parameters: {model.count_params():,}")
        print(f"Trainable parameters: {sum([tf.keras.backend.count_params(w) for w in model.trainable_weights]):,}")

    # Train all models
    histories = []
    evaluations = []

    for name, model in models.items():
        # Train model
        history = train_model(model, name, train_gen, test_gen)
        histories.append(history)

        # Evaluate model
        evaluation = evaluate_model(model, name, train_gen, test_gen)
        evaluations.append(evaluation)

    return models, histories, evaluations

# Execute main function
if train_df is not None and test_df is not None:
    models, histories, evaluations = main()

# Cell 10: Results Analysis and Visualization
def create_results_table(evaluations):
    """Create comprehensive results table"""
    results = []

    for eval_result in evaluations:
        results.append({
            'Model': eval_result['model_name'],
            'Train Accuracy': eval_result['train_acc'],
            'Test Accuracy': eval_result['test_acc'],
```



```
Letters_DL_COMPARISON updated.ipynb
Letters_DL_COMPARISON updated.ipynb > # Deep Learning Model Comparison - Complete Analysis
Generate + Code + Markdown | Run All  Clear All Outputs | Outline ...        Select Kerne
GPU Available: []
Configuration:
   IMG_SIZE: (64, 64)
   BATCH_SIZE: 32
   EPOCHS: 15
   PATIENCE: 10
   LEARNING_RATE: 0.001
   RANDOM_SEED: 42
Training data shape: (173355, 3)
Testing data shape: (43339, 3)

Training data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 173355 entries, 0 to 173354
Data columns (total 3 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Image_Path      173355 non-null  object
 1   Label           173355 non-null  int64
 2   Handwriting_ID  173355 non-null  object
dtypes: int64(1), object(2)
memory usage: 4.0+ MB
None
...
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 451s 83ms/step - accuracy: 0.8416 - loss: 0.2856 - val_accuracy: 0.8396 - val_loss: 0.2842 - learning_rate: 5.0000e-04
Epoch 15/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 452s 83ms/step - accuracy: 0.8438 - loss: 0.2833 - val_accuracy: 0.8225 - val_loss: 0.2998 - learning_rate: 5.0000e-04
Restoring model weights from the end of the best epoch: 14.
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format,
Model Baseline_CNN saved successfully!

Evaluating Baseline_CNN...

==================================================
Training VGG16_Transfer
==================================================
Epoch 1/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 3185s 588ms/step - accuracy: 0.7184 - loss: 0.6177 - val_accuracy: 0.7302 - val_loss: 0.5362 - learning_rate: 0.0010
Epoch 2/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 3230s 596ms/step - accuracy: 0.7309 - loss: 0.5594 - val_accuracy: 0.7302 - val_loss: 0.5338 - learning_rate: 0.0010
Epoch 3/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 3193s 589ms/step - accuracy: 0.7281 - loss: 0.5609 - val_accuracy: 0.7302 - val_loss: 0.5520 - learning_rate: 0.0010
Epoch 4/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 3178s 587ms/step - accuracy: 0.7298 - loss: 0.5550 - val_accuracy: 0.7302 - val_loss: 0.5302 - learning_rate: 0.0010
Epoch 5/15
5418/5418 ━━━━━━━━━━━━━━━━━━━━ 3311s 611ms/step - accuracy: 0.7278 - loss: 0.5558 - val_accuracy: 0.7302 - val_loss: 0.5326 - learning_rate: 0.0010
```





Deep Learning Model Comparison in Numeric Dysgraphia

Numbers_DL_COMPARISON-Updated.ipynb ×

Numbers_DL_COMPARISON-Updated.ipynb > # Deep Learning Model Comparison - Complete Analysis

Generate + Code + Markdown | Run All Clear All Outputs | Outline ···

```python
# Set style for better plots
plt.style.use('seaborn-v0_8')
sns.set_palette("husl")

print("TensorFlow version:", tf.__version__)
print("GPU Available:", tf.config.list_physical_devices('GPU'))

# Cell 2: Configuration and Setup
# Configuration parameters
CONFIG = {
    'IMG_SIZE': (64, 64),
    'BATCH_SIZE': 32,
    'EPOCHS': 15,
    'PATIENCE': 10,
    'LEARNING_RATE': 5e-5,
    'RANDOM_SEED': 42
}

# Set random seeds for reproducibility
np.random.seed(CONFIG['RANDOM_SEED'])
tf.random.set_seed(CONFIG['RANDOM_SEED'])

# Create output directories
os.makedirs("trained_models", exist_ok=True)
os.makedirs("results", exist_ok=True)

print("Configuration:")
for key, value in CONFIG.items():
    print(f"  {key}: {value}")

# Cell 3: Data Loading and Preprocessing
def load_and_prepare_data():
    """Load and prepare training and testing datasets"""
    try:
        # Load CSV files
        train_df = pd.read_csv("Numbers_training_dataset.csv")
        test_df = pd.read_csv("Numbers_testing_dataset.csv")

        print(f"Training data shape: {train_df.shape}")
        print(f"Testing data shape: {test_df.shape}")

        # Display basic statistics
        print("\nTraining data info:")
        print(train_df.info())
        print("\nLabel distribution in training data:")
        print(train_df['Label'].value_counts())
```

[2]

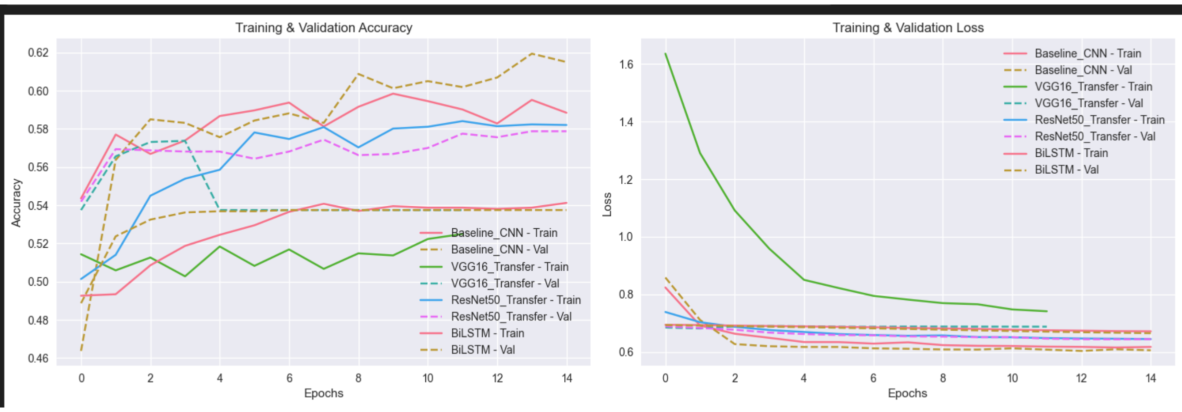
```python
    )

    return model

def create_resnet50_transfer():
    """Create ResNet50 transfer learning model"""
    base_res = ResNet50(weights="imagenet", include_top=False, input_shape=(64,64,3))
    base_res.trainable = False

    inp = Input((64,64,1))
    x = tf.keras.layers.Concatenate()([inp, inp, inp])  # Convert to 3 channels
    x = resnet_preprocess(x)
    x = base_res(x, training=False)
    x = GlobalAveragePooling2D()(x)
    x = Dense(128, activation="relu")(x)
    x = Dropout(0.5)(x)
    x = Dense(64, activation="relu")(x)
    x = Dropout(0.3)(x)
    out = Dense(1, activation="sigmoid")(x)

    model = Model(inp, out)
    model.compile(
        optimizer=Adam(learning_rate=CONFIG['LEARNING_RATE']),
        loss="binary_crossentropy",
        metrics=["accuracy"]
    )

    return model

def create_bilstm_model():
    """Create BiLSTM model for sequence processing"""
    model = Sequential([
        Reshape((64, 64), input_shape=(64,64,1)),
        Bidirectional(LSTM(64, return_sequences=True, dropout=0.3)),
        Bidirectional(LSTM(32, dropout=0.3)),
        Dense(128, activation="relu"),
        Dropout(0.5),
        Dense(64, activation="relu"),
        Dropout(0.3),
        Dense(1, activation="sigmoid")
    ])

    model.compile(
        optimizer=Adam(learning_rate=CONFIG['LEARNING_RATE']/10),
        loss="binary_crossentropy",
        metrics=["accuracy"]
    )
```

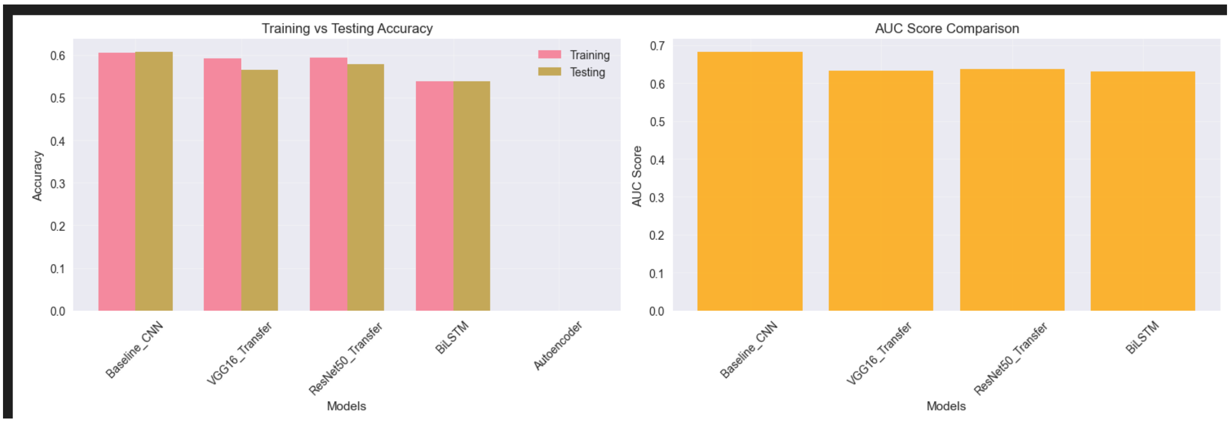
[2]

Generate + Code + Markdown | Run All Clear All Outputs | Outline ...

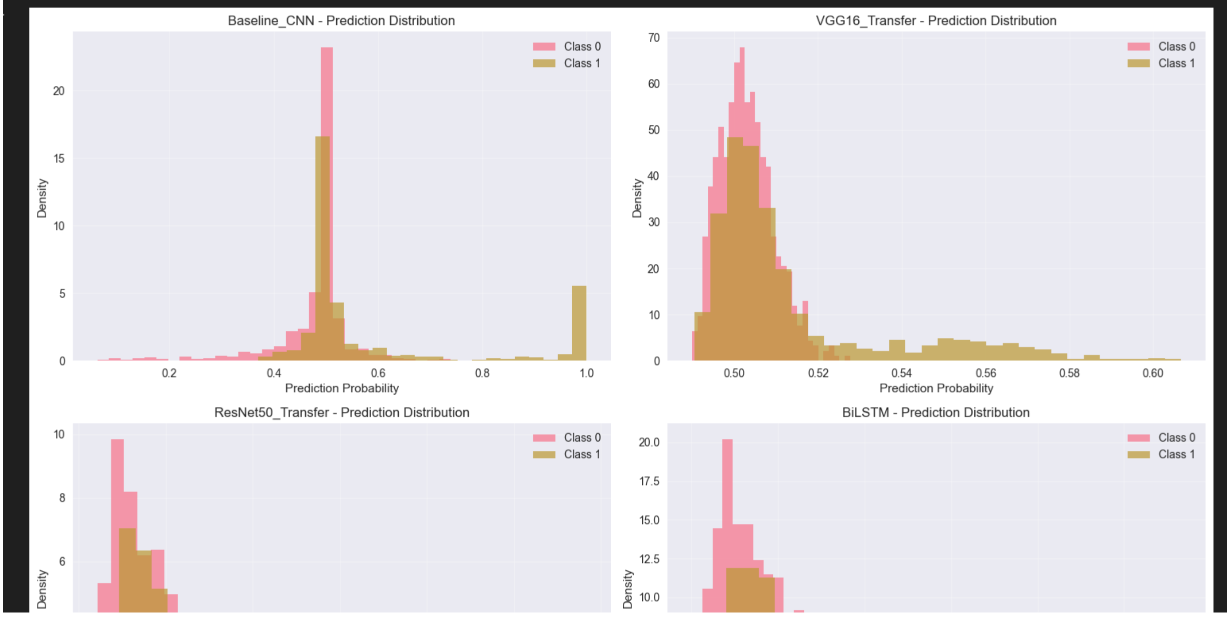
```python
            plot_idx += 1

    # Remove empty subplots
    for i in range(plot_idx, len(axes)):
        fig.delaxes(axes[i])

    plt.tight_layout()
    plt.savefig('results/confusion_matrices.png', dpi=300, bbox_inches='tight')
    plt.show()

# Cell 9: Main Execution - Model Creation and Training
def main():
    """Main execution function"""
    # Initialize models
    models = {
        'Baseline_CNN': create_baseline_cnn(),
        'VGG16_Transfer': create_vgg16_transfer(),
        'ResNet50_Transfer': create_resnet50_transfer(),
        'BiLSTM': create_bilstm_model(),
        'Autoencoder': create_autoencoder()
    }

    # Print model summaries
    for name, model in models.items():
        print(f"\n{name} Architecture:")
        print(f"Total parameters: {model.count_params():,}")
        print(f"Trainable parameters: {sum([tf.keras.backend.count_params(w) for w in model.trainable_weights]):,}")

    # Train all models
    histories = []
    evaluations = []

    for name, model in models.items():
        # Train model
        history = train_model(model, name, train_gen, test_gen)
        histories.append(history)
```

```
TensorFlow version: 2.19.0
GPU Available: []
Configuration:
  IMG_SIZE: (64, 64)
  BATCH_SIZE: 32
  EPOCHS: 15
  PATIENCE: 10
  LEARNING_RATE: 5e-05
  RANDOM_SEED: 42
Training data shape: (6400, 3)
Testing data shape: (1600, 3)

Training data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6400 entries, 0 to 6399
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Image_Path      6400 non-null   object
 1   Label           6400 non-null   int64
 2   Handwriting_ID  6400 non-null   object
dtypes: int64(1), object(2)
memory usage: 150.1+ KB
None
...
200/200 ━━━━━━━━━━━━━━━━━━━━ 24s 119ms/step - accuracy: 0.5935 - loss: 0.6155 - val_accuracy: 0.6194 - val_loss: 0.6096 - learning_rate: 5.0000e-05
Epoch 15/15
200/200 ━━━━━━━━━━━━━━━━━━━━ 24s 120ms/step - accuracy: 0.5870 - loss: 0.6224 - val_accuracy: 0.6150 - val_loss: 0.6061 - learning_rate: 5.0000e-05
Restoring model weights from the end of the best epoch: 13.
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead t
Model Baseline_CNN saved successfully!

Evaluating Baseline_CNN...

==================================================
Training VGG16_Transfer
==================================================
Epoch 1/15
200/200 ━━━━━━━━━━━━━━━━━━━━ 80s 393ms/step - accuracy: 0.5111 - loss: 1.8054 - val_accuracy: 0.5375 - val_loss: 0.6850 - learning_rate: 5.0000e-05
Epoch 2/15
200/200 ━━━━━━━━━━━━━━━━━━━━ 78s 390ms/step - accuracy: 0.5056 - loss: 1.3469 - val_accuracy: 0.5656 - val_loss: 0.6819 - learning_rate: 5.0000e-05
Epoch 3/15
200/200 ━━━━━━━━━━━━━━━━━━━━ 78s 390ms/step - accuracy: 0.5176 - loss: 1.1280 - val_accuracy: 0.5731 - val_loss: 0.6890 - learning_rate: 5.0000e-05
```






```
ResNet50_Transfer         0.0086          0.0139          15         14          0.9333
           BiLSTM         0.0123          0.0090          15          7          0.4667

================================================================================
PREDICTION ANALYSIS
================================================================================
```



Machine Learning Model Comparison in Letter Dysgraphia

DL Models results.ipynb | Dysgraphia_ML_Comparison_Letters.ipynb | Letters DL comprehensive_analysis_report.txt | Numbers DL comprehensive_analysis_report.txt | Dysgraphia

ML > Dysgraphia_ML_Comparison_Letters.ipynb > Comparative Analysis of ML Models for Dysgraphia Detection

Generate | + Code | + Markdown | Run All | Clear All Outputs | Outline | ... Select

[4]
```python
# Load training and testing CSV files
train_csv = './Letters_training_dataset.csv'
test_csv  = 'Letters_testing_dataset.csv'
train_df = pd.read_csv(train_csv)
test_df  = pd.read_csv(test_csv)
train_df.head()
```

	Image_Path	Label	Handwriting_ID
0	/Users/sahan/Documents/Research/Letters/Positi...	1	D_A_34818
1	/Users/sahan/Documents/Research/Letters/Positi...	1	D_A_6322
2	/Users/sahan/Documents/Research/Letters/Negati...	0	ND_A_181
3	/Users/sahan/Documents/Research/Letters/Positi...	1	D_P_8111
4	/Users/sahan/Documents/Research/Letters/Positi...	1	D_G_4415

[7]
```python
# Limit the dataset to 10,000 images
train_df = train_df.sample(n=10000, random_state=42).reset_index(drop=True)
test_df = test_df.sample(n=10000, random_state=42).reset_index(drop=True)

# Feature extraction using pretrained VGG16
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,224,3))
feature_model = Model(inputs=base_model.input, outputs=base_model.output)

def extract_features(df):
    features = []
    for path in tqdm(df['Image_Path'], desc='Extracting'):
        img = load_img(path, target_size=(224,224))
        arr = img_to_array(img)
        arr = np.expand_dims(arr, 0)
        arr = preprocess_input(arr)
        feat = feature_model.predict(arr, verbose=0)
        features.append(feat.flatten())
    return np.array(features)

X_train = extract_features(train_df)
X_test  = extract_features(test_df)
y_train = train_df['Label'].values
y_test  = test_df['Label'].values
```

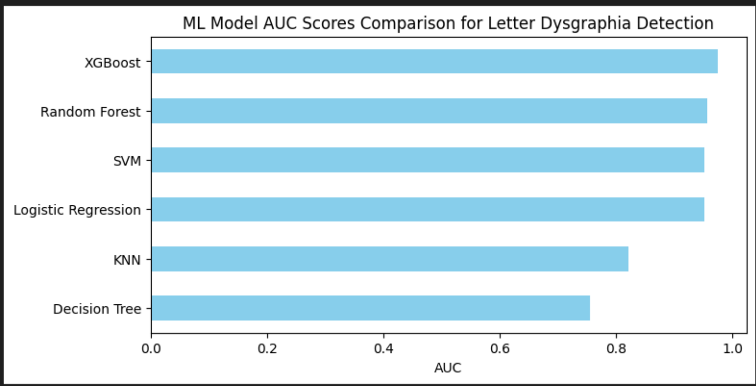
```
Extracting: 100%|██████████| 10000/10000 [29:34<00:00,  5.64it/s]
```

[8]
```python
# Scale features
scaler = StandardScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled  = scaler.transform(X_test)
```

[9]
```python
# Define ML models
models = {
    'SVM': SVC(kernel='linear', probability=True),
    'Decision Tree': DecisionTreeClassifier(max_depth=10, random_state=42),
    'Logistic Regression': LogisticRegression(max_iter=1000, random_state=42),
    'KNN': KNeighborsClassifier(n_neighbors=5),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
}
```

[10]
```python
results = {}
for name, model in models.items():
    print(f'Training and evaluating {name}...')
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    y_prob = model.predict_proba(X_test_scaled)[:,1]
    # Metrics
    acc = classification_report(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    auc = roc_auc_score(y_test, y_prob)
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    # Store
    results[name] = {'cm': cm, 'auc': auc, 'fpr': fpr, 'tpr': tpr}
    # Confusion matrix plot
    plt.figure(figsize=(4,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.title(f'{name} Confusion Matrix')
    plt.xlabel('Predicted'); plt.ylabel('Actual')
    plt.show()
    # ROC curve
    plt.figure(figsize=(6,4))
    plt.plot(fpr, tpr, label=f'AUC = {auc:.2f}')
    plt.plot([0,1],[0,1],'--', color='gray')
    plt.title(f'{name} ROC Curve'); plt.xlabel('FPR'); plt.ylabel('TPR')
```

```python
print("=" * 60)
print("ML MODEL PERFORMANCE COMPARISON for Letter Dysgraphia Detection")
print("=" * 60)
print(performance_df.to_string(index=False))
print("=" * 60)

# Also create a more detailed table with additional metrics
print("\nDETAILED PERFORMANCE METRICS")
print("=" * 80)
print(f"{'Model':<20} {'Accuracy':<10} {'AUC':<10} {'Accuracy %':<12}")
print("-" * 80)
for _, row in performance_df.iterrows():
    print(f"{row['Model']:<20} {row['Accuracy']:<10} {row['AUC']:<10} {row['Accuracy (%)']:<12}")
print("=" * 80)
```




```
============================================================
ML MODEL PERFORMANCE COMPARISON for Letter Dysgraphia Detection
============================================================
              Model Accuracy    AUC Accuracy (%)
            XGBoost   0.9291 0.9754       92.91%
                SVM   0.9038 0.9518       90.38%
      Random Forest   0.9010 0.9570       90.10%
Logistic Regression   0.9001 0.9512       90.01%
```

```
============================================================
ML MODEL PERFORMANCE COMPARISON for Letter Dysgraphia Detection
============================================================
              Model Accuracy    AUC Accuracy (%)
            XGBoost   0.9291 0.9754       92.91%
                SVM   0.9038 0.9518       90.38%
      Random Forest   0.9010 0.9570       90.10%
Logistic Regression   0.9001 0.9512       90.01%
      Decision Tree   0.8260 0.7551       82.60%
                KNN   0.8146 0.8210       81.46%
============================================================

DETAILED PERFORMANCE METRICS
================================================================================
Model                Accuracy   AUC        Accuracy %
--------------------------------------------------------------------------------
XGBoost              0.9291     0.9754     92.91%
SVM                  0.9038     0.9518     90.38%
Random Forest        0.9010     0.9570     90.10%
Logistic Regression  0.9001     0.9512     90.01%
Decision Tree        0.8260     0.7551     82.60%
KNN                  0.8146     0.8210     81.46%
================================================================================
```

Machine Learning Model Comparison in Numeric Dysgraphia

```
# Load training and testing CSV files
train_csv = './Numbers_training_dataset.csv'
test_csv  = './Numbers_testing_dataset.csv'
train_df = pd.read_csv(train_csv)
test_df  = pd.read_csv(test_csv)
train_df.head()
```

[3]

	Image_Path	Label	Handwriting_ID
0	/Users/sahan/Documents/Research/Numbers/Number...	0	ND_7_0_46
1	/Users/sahan/Documents/Research/Numbers/Number...	0	ND_7_1_35
2	/Users/sahan/Documents/Research/Numbers/Number...	1	D_9_7_45
3	/Users/sahan/Documents/Research/Numbers/Number...	1	D_6_3_51
4	/Users/sahan/Documents/Research/Numbers/Number...	1	D_10_5_7

```
# Feature extraction using pretrained VGG16
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224,224,3))
feature_model = Model(inputs=base_model.input, outputs=base_model.output)

def extract_features(df):
    features = []
    for path in tqdm(df['Image_Path'], desc='Extracting'):
        img = load_img(path, target_size=(224,224))
        arr = img_to_array(img)
        arr = np.expand_dims(arr, 0)
        arr = preprocess_input(arr)
        feat = feature_model.predict(arr, verbose=0)
        features.append(feat.flatten())
    return np.array(features)

X_train = extract_features(train_df)
X_test  = extract_features(test_df)
y_train = train_df['Label'].values
y_test  = test_df['Label'].values
```

[4]

```
Extracting: 100%|██████████| 6400/6400 [18:23<00:00,  5.80it/s]
Extracting: 100%|██████████| 1600/1600 [05:51<00:00,  4.55it/s]
```

```
# Scale features
scaler = StandardScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled  = scaler.transform(X_test)
```

[5]

```
# Define ML models
models = {
    'SVM': SVC(kernel='linear', probability=True),
    'Decision Tree': DecisionTreeClassifier(max_depth=10, random_state=42),
    'Logistic Regression': LogisticRegression(max_iter=1000, random_state=42),
    'KNN': KNeighborsClassifier(n_neighbors=5),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'XGBoost': XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
}
```

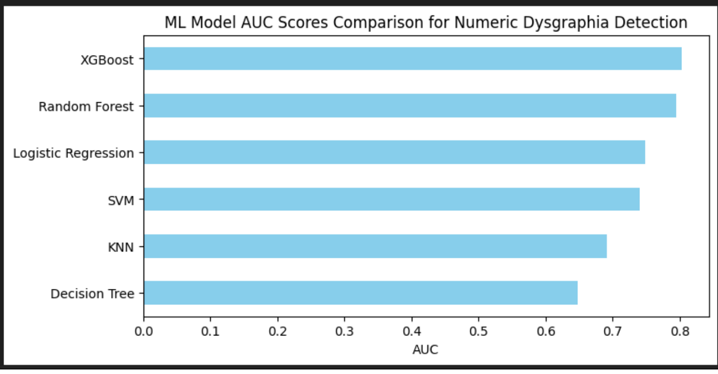
[6]

```
results = {}
for name, model in models.items():
    print(f'Training and evaluating {name}...')
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    y_prob = model.predict_proba(X_test_scaled)[:,1]
    # Metrics
    acc = classification_report(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    auc = roc_auc_score(y_test, y_prob)
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    # Store
    results[name] = {'cm': cm, 'auc': auc, 'fpr': fpr, 'tpr': tpr}
    # Confusion matrix plot
    plt.figure(figsize=(4,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.title(f'{name} Confusion Matrix')
    plt.xlabel('Predicted'); plt.ylabel('Actual')
    plt.show()
    # ROC curve
    plt.figure(figsize=(6,4))
    plt.plot(fpr, tpr, label=f'AUC = {auc:.2f}')
    plt.plot([0,1],[0,1],'--', color='gray')
    plt.title(f'{name} ROC Curve'); plt.xlabel('FPR'); plt.ylabel('TPR')
    plt.legend(); plt.show()
```

[7]

```python
        performance_data.append({
            'Model': name,
            'Accuracy': f"{accuracy:.4f}",
            'AUC': f"{info['auc']:.4f}",
            'Accuracy (%)': f"{accuracy*100:.2f}%"
        })

    # Create DataFrame for better formatting
    performance_df = pd.DataFrame(performance_data)
    performance_df = performance_df.sort_values('Accuracy', ascending=False)

    print("=" * 60)
    print("ML MODEL PERFORMANCE COMPARISON for Numeric Dysgraphia Detection")
    print("=" * 60)
    print(performance_df.to_string(index=False))
    print("=" * 60)

    # Also create a more detailed table with additional metrics
    print("\nDETAILED PERFORMANCE METRICS")
    print("=" * 80)
    print(f"{'Model':<20} {'Accuracy':<10} {'AUC':<10} {'Accuracy %':<12}")
    print("-" * 80)
    for _, row in performance_df.iterrows():
        print(f"{row['Model']:<20} {row['Accuracy']:<10} {row['AUC']:<10} {row['Accuracy (%)']:<12}")
    print("=" * 80)
```

[19] Python

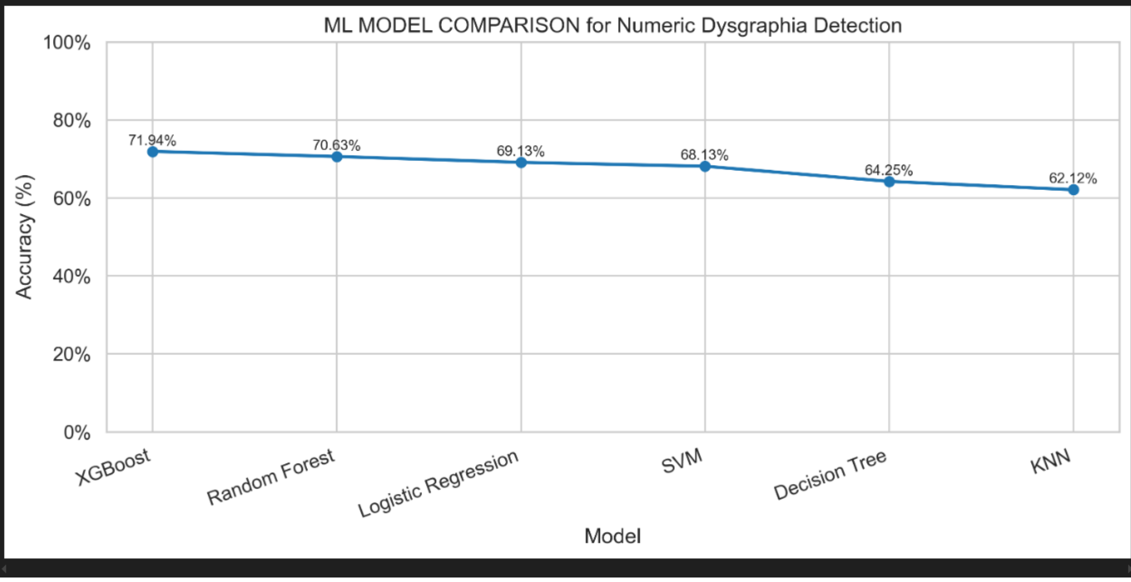



```
============================================================
ML MODEL PERFORMANCE COMPARISON for Numeric Dysgraphia Detection
============================================================
              Model Accuracy    AUC Accuracy (%)
            XGBoost   0.7194 0.8030       71.94%
      Random Forest   0.7063 0.7947       70.62%
Logistic Regression   0.6913 0.7482       69.12%
                SVM   0.6813 0.7396       68.12%
      Decision Tree   0.6425 0.6478       64.25%
                KNN   0.6212 0.6909       62.12%
============================================================

DETAILED PERFORMANCE METRICS
================================================================================
Model                Accuracy   AUC        Accuracy %
--------------------------------------------------------------------------------
XGBoost              0.7194     0.8030     71.94%
Random Forest        0.7063     0.7947     70.62%
Logistic Regression  0.6913     0.7482     69.12%
SVM                  0.6813     0.7396     68.12%
Decision Tree        0.6425     0.6478     64.25%
KNN                  0.6212     0.6909     62.12%
================================================================================
```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import PercentFormatter

# Manual accuracies from your table
manual_acc = [
    {"Model": "XGBoost",             "Accuracy": 0.7194},
    {"Model": "Random Forest",       "Accuracy": 0.7063},
    {"Model": "Logistic Regression", "Accuracy": 0.6913},
    {"Model": "SVM",                 "Accuracy": 0.6813},
    {"Model": "Decision Tree",       "Accuracy": 0.6425},
    {"Model": "KNN",                 "Accuracy": 0.6212},
]

# Build accuracy DataFrame
if 'results' in globals():
    acc_rows = []
    for name, info in results.items():
        cm = info['cm']
```

[3]

```python
for x, y in zip(acc_df['Model'], acc_df['Accuracy'] * 100):
    ax.text(x, y + 1.0, f'{y:.2f}%', ha='center', va='bottom', fontsize=9)

plt.tight_layout()
plt.savefig('ml_model_comparison_accuracy_line.png', dpi=300, bbox_inches='tight')
plt.show()
```



ADVANCED CNN FOR LETTERS

ADVANCED CNN > 30 epochs with letters.ipynb > Training > class conv(nn.Module):

Generate + Code + Markdown | Run All | Clear All Outputs | Outline ...

```python
from PIL import Image
```

```python
seed=42
random.seed(seed)
os.environ['PYTHONHASHSEED'] = str(seed)
np.random.seed(seed)
torch.manual_seed(seed)
torch.cuda.manual_seed(seed)
torch.backends.cudnn.deterministic = True
device= torch.device("cuda" if torch.cuda.is_available() else "cpu")
learning_rate=0.0001
num_epochs=30
```

DataLoader

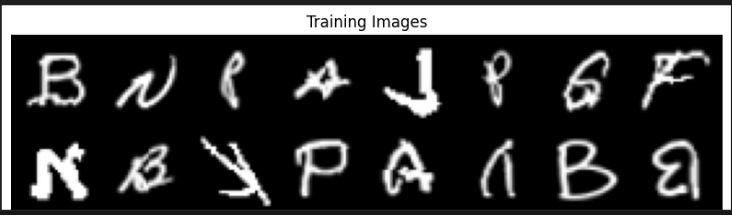
```python
# data transformer
cnn_transform = T.Compose([
    T.Grayscale(num_output_channels=1),
    T.Resize((32, 32)),
    T.ToTensor()
])

dl_transform = T.Compose([
    T.Grayscale(num_output_channels=1),
    T.Resize((224, 224)),
    T.ToTensor()
])
```

```python
class HandwritingDataset(torch.utils.data.Dataset):
    def __init__(self, df, transform=None):
        self.df = df.reset_index(drop=True)
        self.transform = transform
```

```python
train_data = train_df.iloc[:train_size].reset_index(drop=True)
val_data = train_df.iloc[train_size:].reset_index(drop=True)


# --- Create datasets and loaders ---
train_dataset_cnn = HandwritingDataset(train_data, transform=cnn_transform)
val_dataset_cnn   = HandwritingDataset(val_data, transform=cnn_transform)
test_dataset_cnn  = HandwritingDataset(test_df, transform=cnn_transform)

train_dataset_dl = HandwritingDataset(train_data, transform=dl_transform)
val_dataset_dl   = HandwritingDataset(val_data, transform=dl_transform)
test_dataset_dl  = HandwritingDataset(test_df, transform=dl_transform)

train_loader_cnn = DataLoader(train_dataset_cnn, batch_size=5000, shuffle=True)
vald_loader_cnn  = DataLoader(val_dataset_cnn, batch_size=10000, shuffle=True)
test_loader_cnn  = DataLoader(test_dataset_cnn, batch_size=1)

train_loader_dl = DataLoader(train_dataset_dl, batch_size=128, shuffle=True)
vald_loader_dl  = DataLoader(val_dataset_dl, batch_size=256, shuffle=True)
test_loader_dl  = DataLoader(test_dataset_dl, batch_size=1)

print("Dataloader Initialized")
```

```
Dataloader Initialized
```

```python
real_batch = next(iter(train_loader_cnn))
plt.figure(figsize=(40,20))
plt.axis("off")
plt.title("Training Images")
plt.imshow(np.transpose(vutils.make_grid(real_batch[0].to(device)[:128], padding=2, normalize=True).cpu(),(1,2,0)))
plt.show()
```



ADVANCED CNN > 30 epochs with letters.ipynb > Training > class conv(nn.Module):

```python
        )
        self.linearlayers=nn.Sequential(
            #input 64*6*6
            nn.Flatten(),
            nn.Linear(3136,128),
            nn.ReLU(),
            nn.BatchNorm1d(128),
            nn.Dropout(0.05),
            nn.Linear(128, classes),

            #nn.Softmax(),
        )
        self.softmax=nn.Softmax(dim=1)
    def forward(self,x):
        output=self.convlayer(x)
        output=self.linearlayers(output)
        return self.softmax(output)
```

```python
def calc_accuracy(true,pred):
    true = torch.zeros(pred.shape[0], pred.shape[1]).scatter_(1, true.unsqueeze(1), 1.)
    acc = (true.argmax(-1) == pred.argmax(-1)).float().detach().numpy()
    acc = float((100 * acc.sum()) / len(acc))
    return round(acc, 4)
```

```python
# Function to count and display model parameters
def display_model_parameters(model):
    total_params = sum(p.numel() for p in model.parameters())
    trainable_params = sum(p.numel() for p in model.parameters() if p.requires_grad)

    print(f"Total Parameters: {total_params:,}")
    print(f"Trainable Parameters: {trainable_params:,}")
    print("\nDetailed Parameters:")
    for name, param in model.named_parameters():
        print(f"{name}: {param.shape}, Trainable: {param.requires_grad}")

# Example usage for your CNN model
display_model_parameters(model)
```

```
Total Parameters: 595,843
Trainable Parameters: 595,843
```

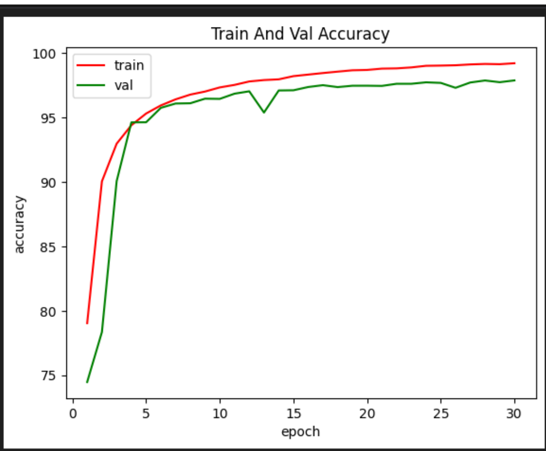
```python
    model.train()

    print("Training Started")

    for epoch in range(num_epochs):
        train_run = []
        val_run = []
        model.train()
        for i, (images, labels) in enumerate(train_loader_cnn):

            images = images.to(device)
            labels = labels.to(device)

            optimizer.zero_grad()

            output  = model.forward(images)

            labels=labels.long().squeeze()

            loss = F.cross_entropy(output,labels)
            loss.backward()
            optimizer.step()

            train_run.append(calc_accuracy(labels.to("cpu"), output.to("cpu")))

        if (epoch+1) % 1 == 0:
            with torch.no_grad():
                model.eval()
                for i, (images, labels) in enumerate(vald_loader_cnn):

                    images = images.to(device)
                    labels = labels.to(device)


                    output  = model.forward(images)

                    labels=labels.long().squeeze()

                    val_run.append(calc_accuracy(labels.to("cpu"), output.to("cpu")))
```

```
Training Started
Epoch [1/30] AC: 79.0370 , Valid_AC:74.4656
Epoch [2/30] AC: 90.0488 , Valid_AC:78.3482
Epoch [3/30] AC: 92.9722 , Valid_AC:90.06110000000001
Epoch [4/30] AC: 94.4039 , Valid_AC:94.62896666666666
Epoch [5/30] AC: 95.3145 , Valid_AC:94.6368
Epoch [6/30] AC: 95.9444 , Valid_AC:95.75540000000001
Epoch [7/30] AC: 96.4142 , Valid_AC:96.09196666666666
Epoch [8/30] AC: 96.7849 , Valid_AC:96.11186666666667
Epoch [9/30] AC: 97.0209 , Valid_AC:96.46633333333334
Epoch [10/30] AC: 97.3413 , Valid_AC:96.45190000000001
Epoch [11/30] AC: 97.5373 , Valid_AC:96.85516666666666
Epoch [12/30] AC: 97.8030 , Valid_AC:97.03949999999999
Epoch [13/30] AC: 97.9120 , Valid_AC:95.38993333333333
Epoch [14/30] AC: 97.9685 , Valid_AC:97.1028
Epoch [15/30] AC: 98.2170 , Valid_AC:97.1185
Epoch [16/30] AC: 98.3376 , Valid_AC:97.37283333333335
Epoch [17/30] AC: 98.4561 , Valid_AC:97.5161
Epoch [18/30] AC: 98.5658 , Valid_AC:97.36616666666667
Epoch [19/30] AC: 98.6707 , Valid_AC:97.47276666666666
Epoch [20/30] AC: 98.6980 , Valid_AC:97.47496666666666
Epoch [21/30] AC: 98.8008 , Valid_AC:97.45946666666667
Epoch [22/30] AC: 98.8198 , Valid_AC:97.62386666666667
Epoch [23/30] AC: 98.8947 , Valid_AC:97.6238
Epoch [24/30] AC: 99.0206 , Valid_AC:97.73610000000001
...
Epoch [27/30] AC: 99.1263 , Valid_AC:97.72269999999999
Epoch [28/30] AC: 99.1640 , Valid_AC:97.87939999999999
Epoch [29/30] AC: 99.1446 , Valid_AC:97.74603333333334
Epoch [30/30] AC: 99.2176 , Valid_AC:97.88486666666665
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

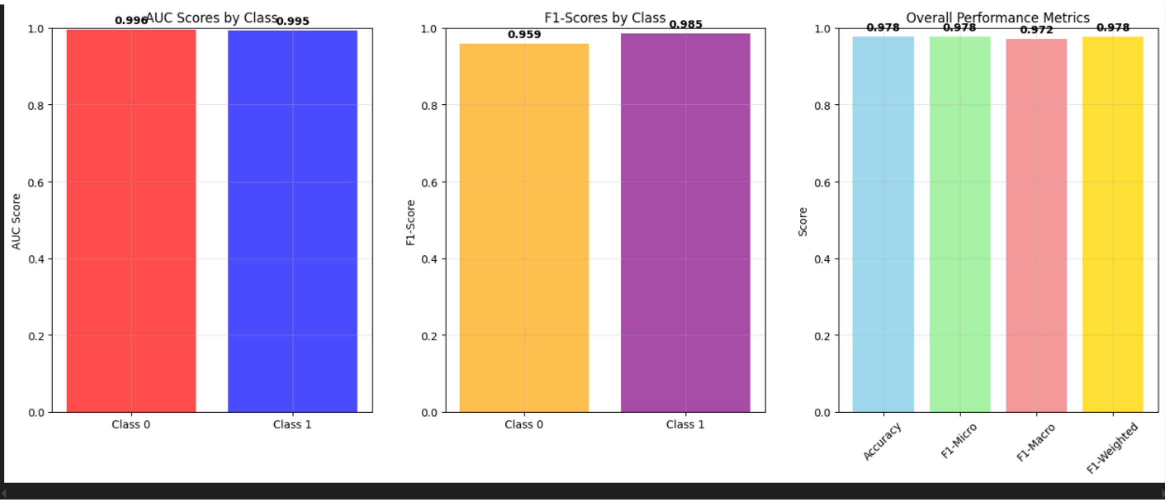
```python
x = [i+1 for i in range(num_epochs)]
plt.plot(x, train_acc, color='r', label='train')
plt.plot(x, val_acc, color='g', label='val')
plt.xlabel("epoch")
plt.ylabel("accuracy")
plt.title("Train And Val Accuracy")
plt.legend()
plt.show()
```

[22] Python




```python
import numpy as np

x = np.array([i+1 for i in range(num_epochs)])
width = 0.35  # width of the bars

plt.bar(x - width/2, train_acc, width, color='r', label='train')
plt.bar(x + width/2, val_acc, width, color='g', label='val')
plt.xlabel("epoch")
plt.ylabel("accuracy")
plt.title("Train And Val Accuracy (Bar Graph)")
plt.legend()
plt.show()
```

[23]



ADVANCED CNN FOR NUMBERS

ADVANCED CNN > Number Advanced CNN copy.ipynb > DataLoader > real_batch = next(iter(train_loader_cnn))

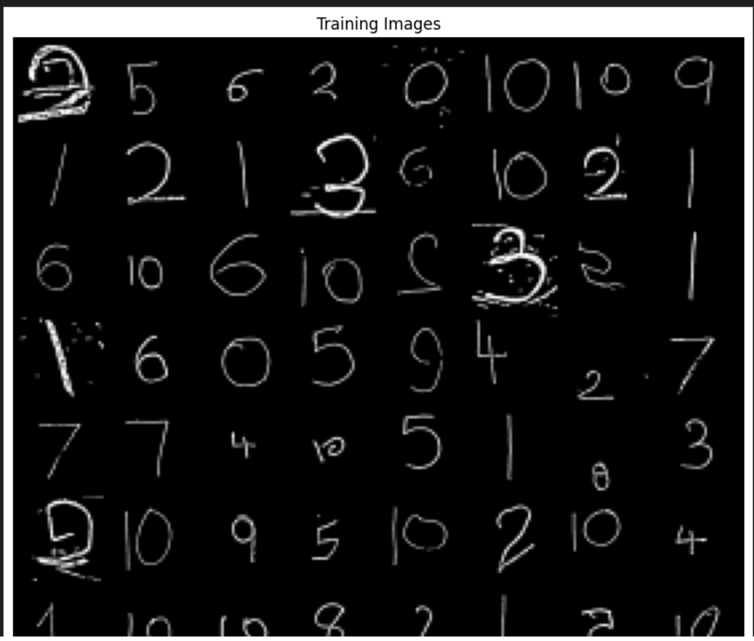
DataLoader

```python
# data transformer
cnn_transform = T.Compose([
    T.Grayscale(num_output_channels=1),
    T.Resize((32, 32)),
    T.ToTensor()
])

dl_transform = T.Compose([
    T.Grayscale(num_output_channels=1),
    T.Resize((224, 224)),
    T.ToTensor()
])
```

```python
class HandwritingDataset(torch.utils.data.Dataset):
    def __init__(self, df, transform=None):
        self.df = df.reset_index(drop=True)
        self.transform = transform

    def __len__(self):
        return len(self.df)

    def __getitem__(self, idx):
        row = self.df.iloc[idx]
        img_path = row['Image_Path']
        label = int(row['Label'])
        img = Image.open(img_path).convert('L')
        if self.transform:
            img = self.transform(img)
        return img, label

# Load CSVs
train_df = pd.read_csv("./Numbers_training_dataset.csv")
test_df  = pd.read_csv("./Numbers_testing_dataset.csv")

# Split train/val from train_df
train_size = int(0.85 * len(train_df))
val_size = len(train_df) - train_size
```

```
Dataloader Initialized
```

```python
real_batch = next(iter(train_loader_cnn))
plt.figure(figsize=(40,20))
plt.axis("off")
plt.title("Training Images")
plt.imshow(np.transpose(vutils.make_grid(real_batch[0].to(device)[:128], padding=2, normalize=True).cpu(),(1,2,0)))
plt.show()
```



```python
                nn.Conv2d(32,64,3,1),
                nn.ReLU(),
                nn.BatchNorm2d(64),
                nn.Conv2d(64,64,3,1),
                nn.ReLU(),
                nn.BatchNorm2d(64),
                nn.Conv2d(64,64,5,2,3),
                nn.ReLU(),
                nn.BatchNorm2d(64),
                nn.Dropout2d(0.1),
            )
            self.linearlayers=nn.Sequential(
                #input 64*6*6
                nn.Flatten(),
                nn.Linear(3136,128),
                nn.ReLU(),
                nn.BatchNorm1d(128),
                nn.Dropout(0.05),
                nn.Linear(128, classes),

                #nn.Softmax(),
            )
            self.softmax=nn.Softmax(dim=1)
        def forward(self,x):
            output=self.convlayer(x)
            output=self.linearlayers(output)
            return self.softmax(output)
```

[32] Python

```python
def calc_accuracy(true,pred):
    true = torch.zeros(pred.shape[0], pred.shape[1]).scatter_(1, true.unsqueeze(1), 1.)
    acc = (true.argmax(-1) == pred.argmax(-1)).float().detach().numpy()
    acc = float((100 * acc.sum()) / len(acc))
    return round(acc, 4)
```

[33] Python

```python
model = conv(3).to(device)
train_acc = []
val_acc = []

optimizer = optim.Adam(model.parameters(),0.0005)
total_step = len(train_loader_cnn)
model.train()
```

[36]

```
Epoch [17/100] AC: 83.4772 , Valid_AC:48.6458
Epoch [18/100] AC: 85.5618 , Valid_AC:48.5417
Epoch [19/100] AC: 84.1645 , Valid_AC:55.625
Epoch [20/100] AC: 86.5191 , Valid_AC:57.2917
Epoch [21/100] AC: 86.6882 , Valid_AC:56.6667
Epoch [22/100] AC: 88.3464 , Valid_AC:59.0625
Epoch [23/100] AC: 89.1900 , Valid_AC:58.8542
Epoch [24/100] AC: 88.4136 , Valid_AC:60.625
...
Epoch [97/100] AC: 97.3500 , Valid_AC:64.375
Epoch [98/100] AC: 98.1654 , Valid_AC:62.1875
Epoch [99/100] AC: 97.0391 , Valid_AC:61.9792
Epoch [100/100] AC: 97.4263 , Valid_AC:62.2917
```

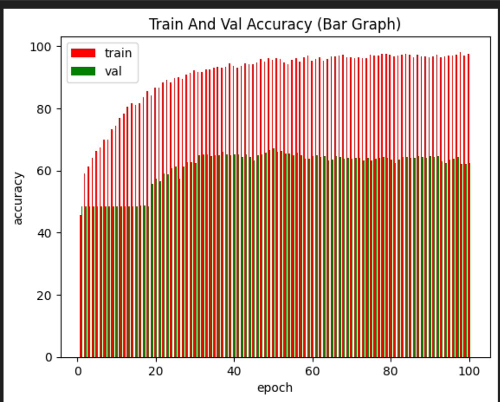
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```python
x = [i+1 for i in range(num_epochs)]
plt.plot(x, train_acc, color='r', label='train')
plt.plot(x, val_acc, color='g', label='val')
plt.xlabel("epoch")
plt.ylabel("accuracy")
plt.title("Train And Val Accuracy")
plt.legend()
plt.show()
```

[37] Python

```python
import numpy as np

x = np.array([i+1 for i in range(num_epochs)])
width = 0.35  # width of the bars

plt.bar(x - width/2, train_acc, width, color='r', label='train')
plt.bar(x + width/2, val_acc, width, color='g', label='val')
plt.xlabel("epoch")
plt.ylabel("accuracy")
plt.title("Train And Val Accuracy (Bar Graph)")
plt.legend()
plt.show()
```

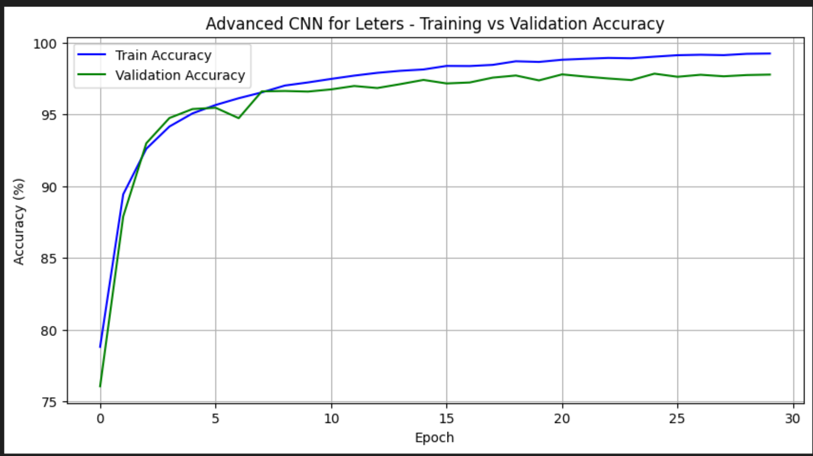
[38] Python

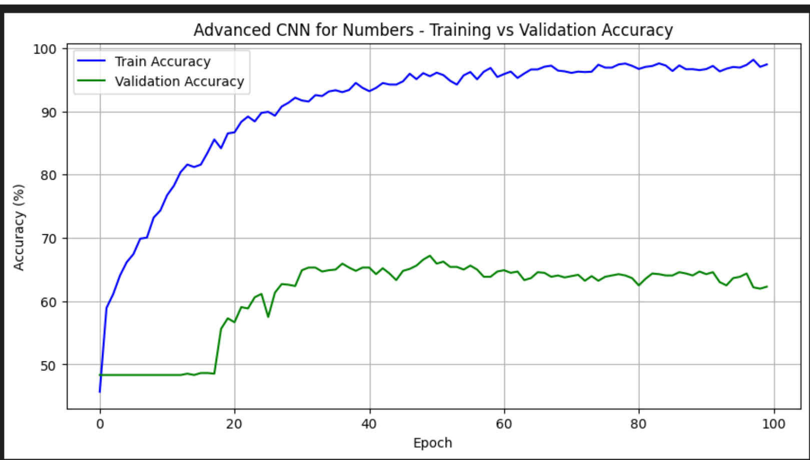


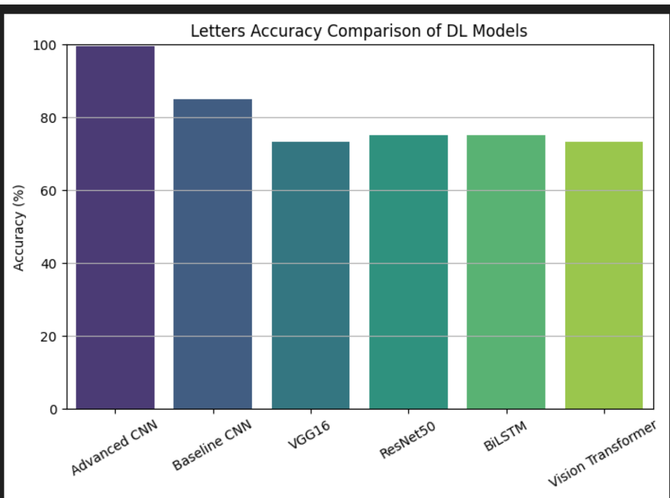
APPENDIX 9: DEEP LEARNING AND MACHINE LEARNING MODEL RESULTS COMPARISON IN LETTERS AND NUMBERS

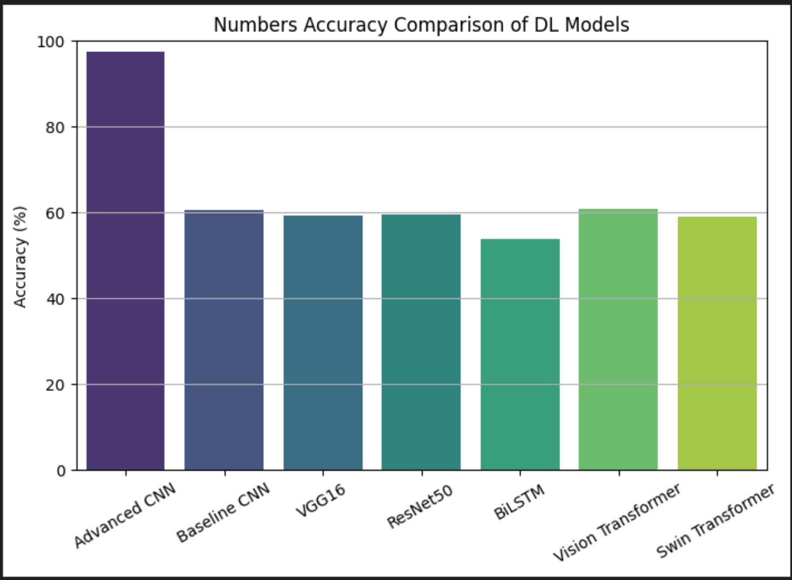
DEEP LEARNING COMAPRISON LETTERS AND NUMBERS DYSGRAPHIA

```
# Numbers Format for display
metrics_df_numbers = metrics_df_numbers.round(3)
print("Numbers Performance Summary Table:")
print(metrics_df_numbers)
```

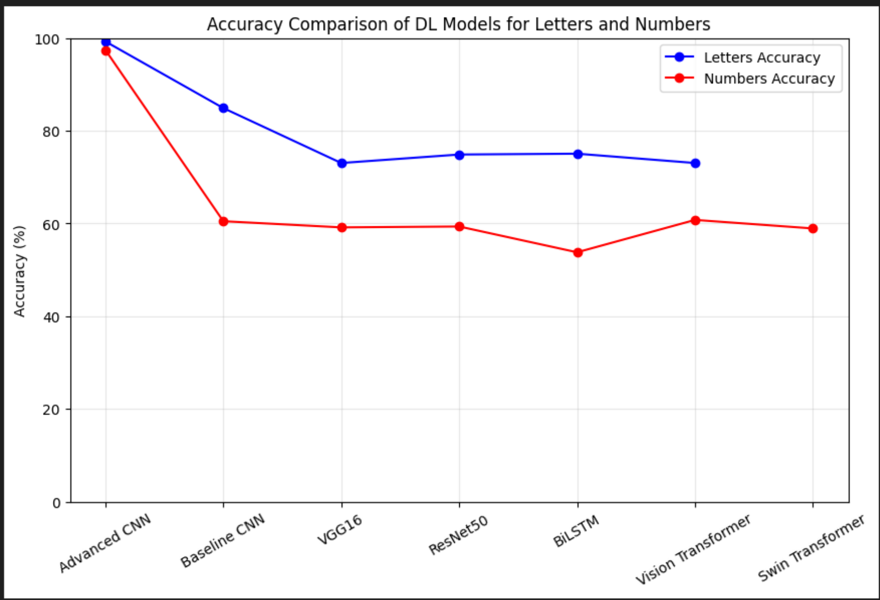


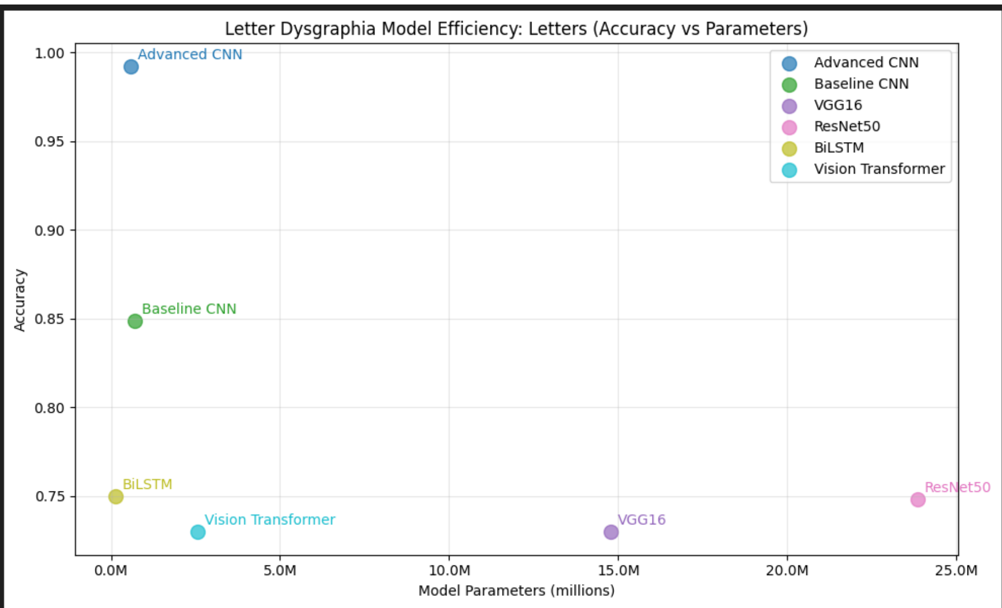


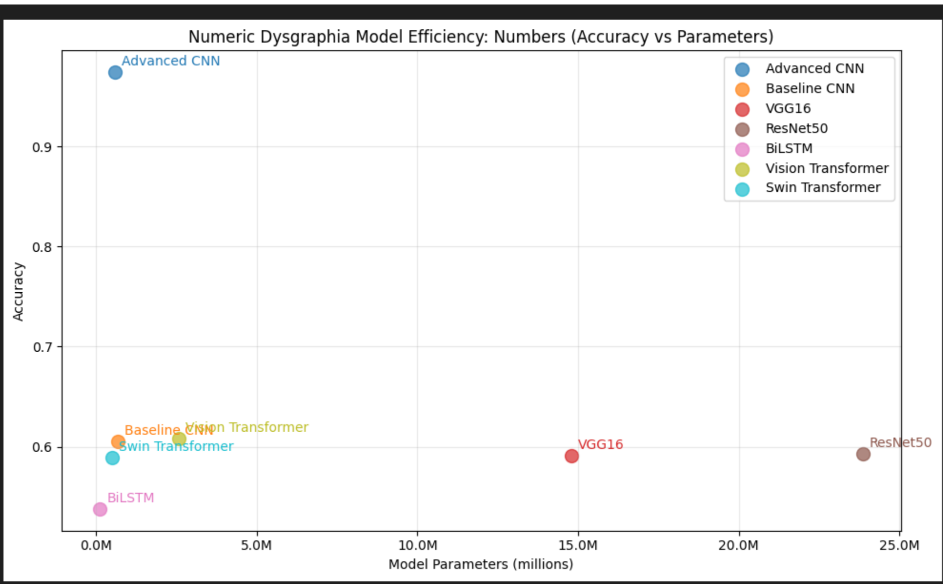




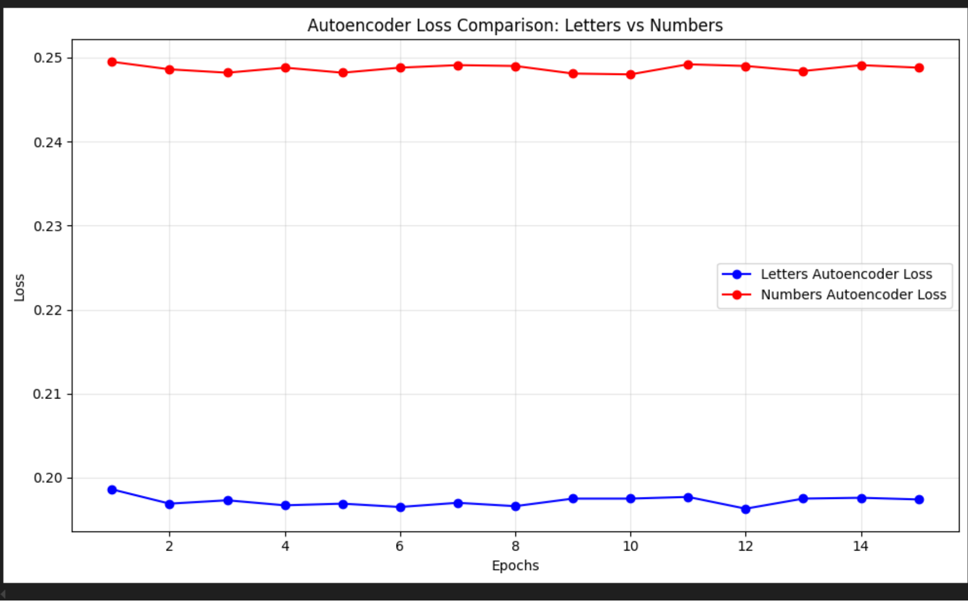

```
plt.legend()
plt.show()
```



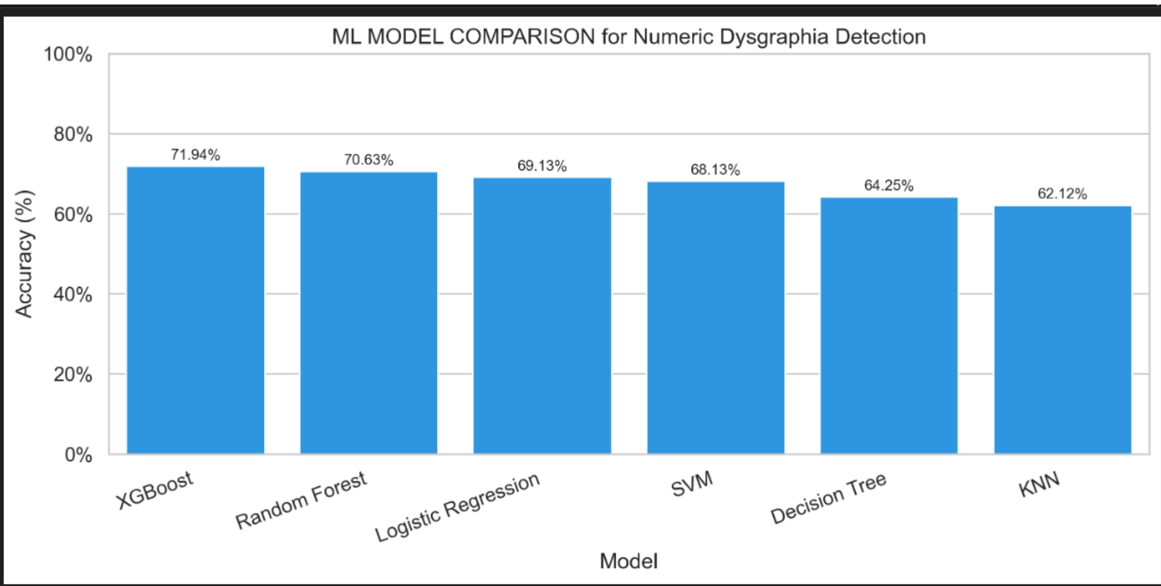


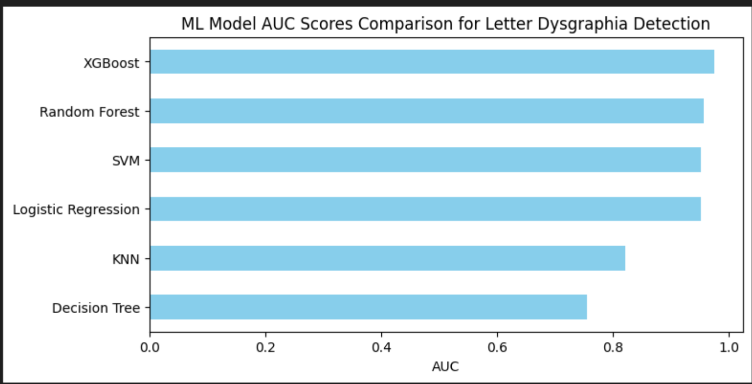



```
plt.ylabel("Loss")
plt.title("Autoencoder Loss Comparison: Letters vs Numbers")
plt.legend()
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()
```



MACHINE LEARNING COMAPRISON LETTERS AND NUMBERS DYSGRAPHIA

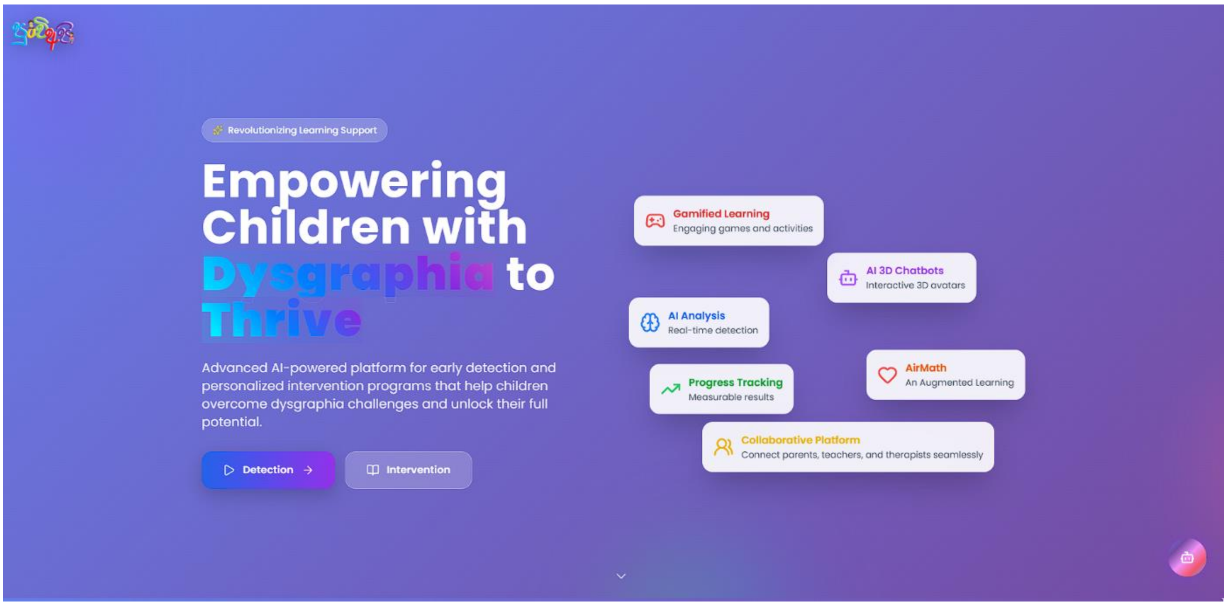


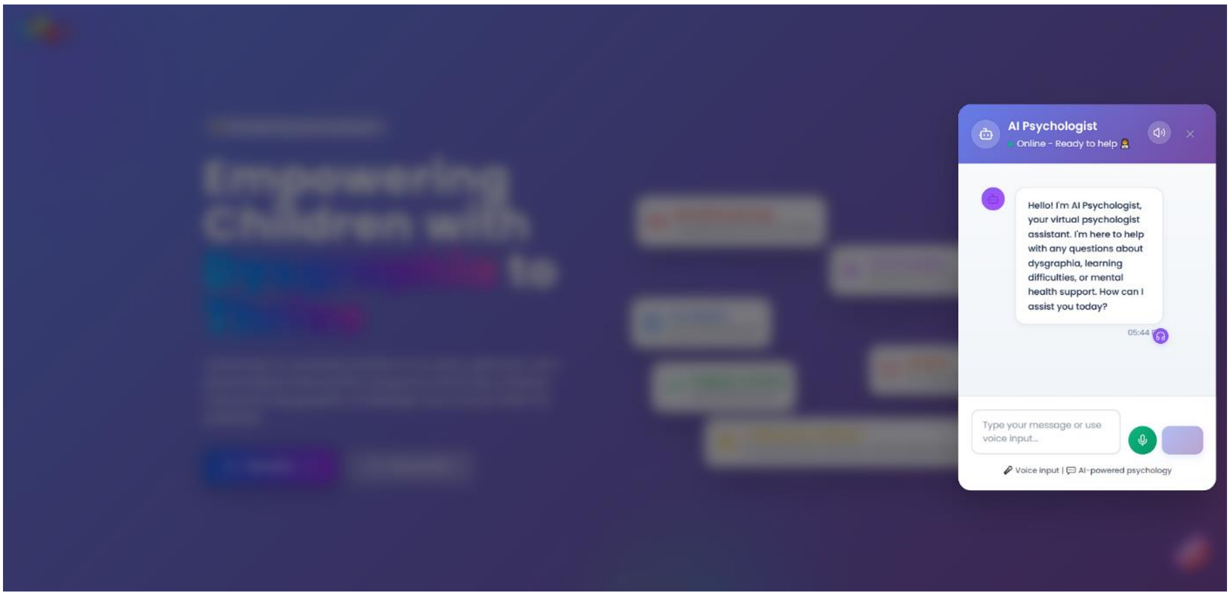


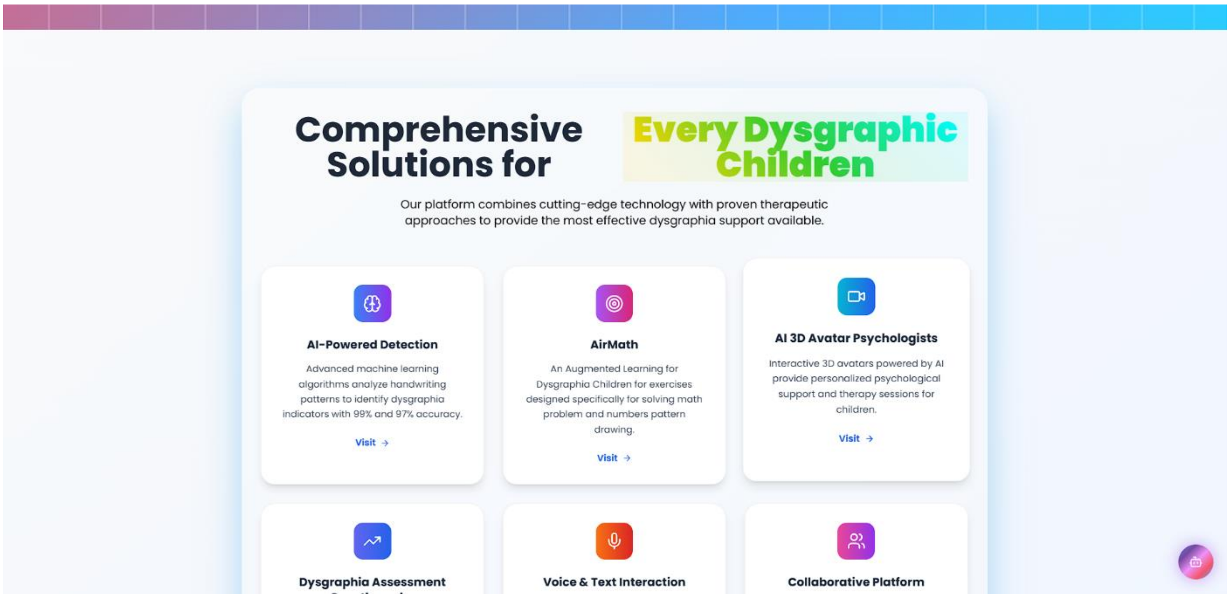

```
============================================================
ML MODEL PERFORMANCE COMPARISON for Letter Dysgraphia Detection
============================================================
              Model Accuracy    AUC Accuracy (%)
            XGBoost   0.9291 0.9754       92.91%
                SVM   0.9038 0.9518       90.38%
      Random Forest   0.9010 0.9570       90.10%
Logistic Regression   0.9001 0.9512       90.01%
      Decision Tree   0.8260 0.7551       82.60%
                KNN   0.8146 0.8210       81.46%
============================================================

DETAILED PERFORMANCE METRICS
================================================================================
Model                Accuracy   AUC        Accuracy %
--------------------------------------------------------------------------------
XGBoost              0.9291     0.9754     92.91%
SVM                  0.9038     0.9518     90.38%
Random Forest        0.9010     0.9570     90.10%
Logistic Regression  0.9001     0.9512     90.01%
Decision Tree        0.8260     0.7551     82.60%
KNN                  0.8146     0.8210     81.46%
================================================================================
```

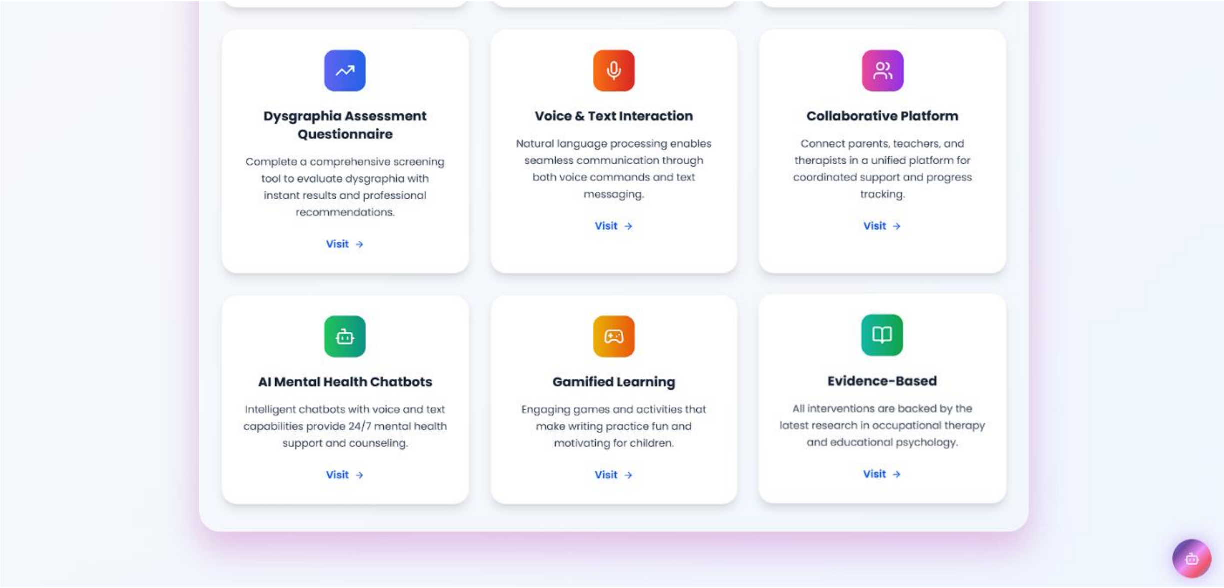
APPENDIX 10: USER INTERFACE

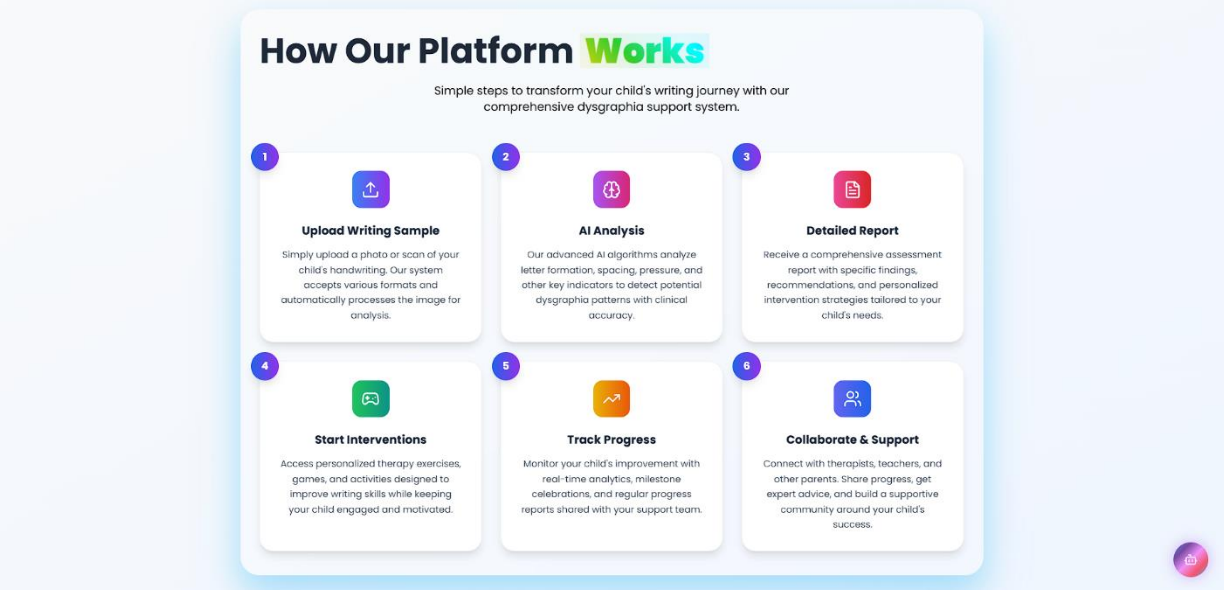
Home Site

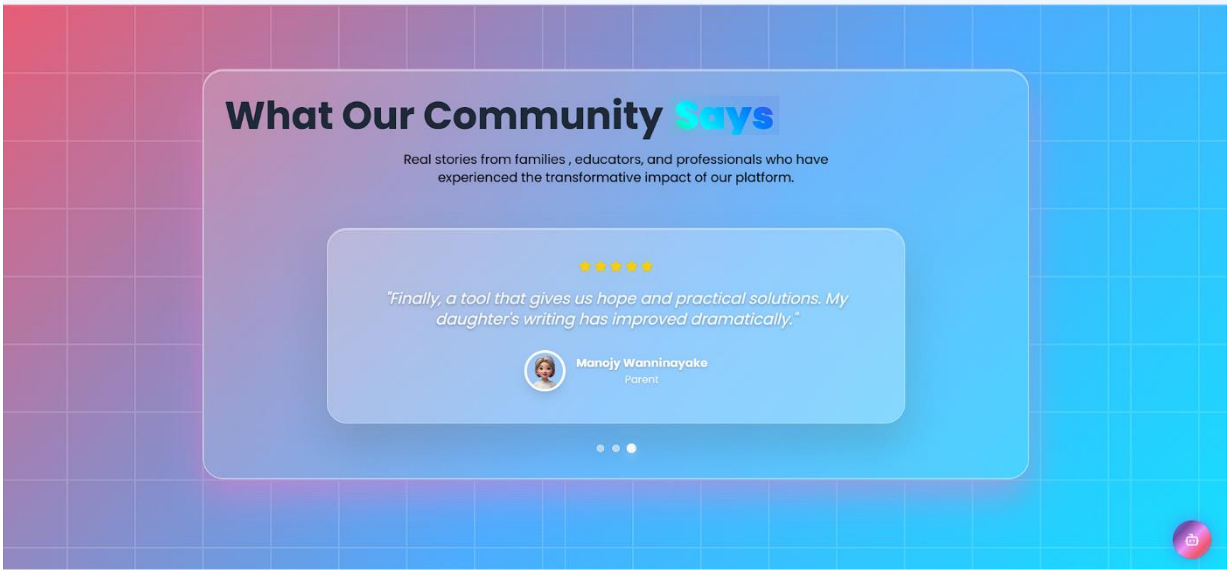


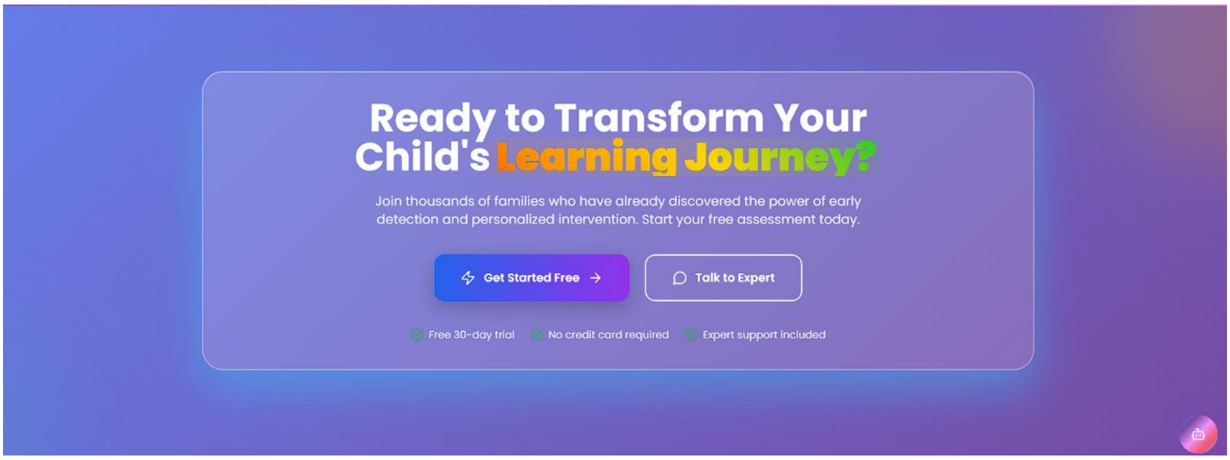


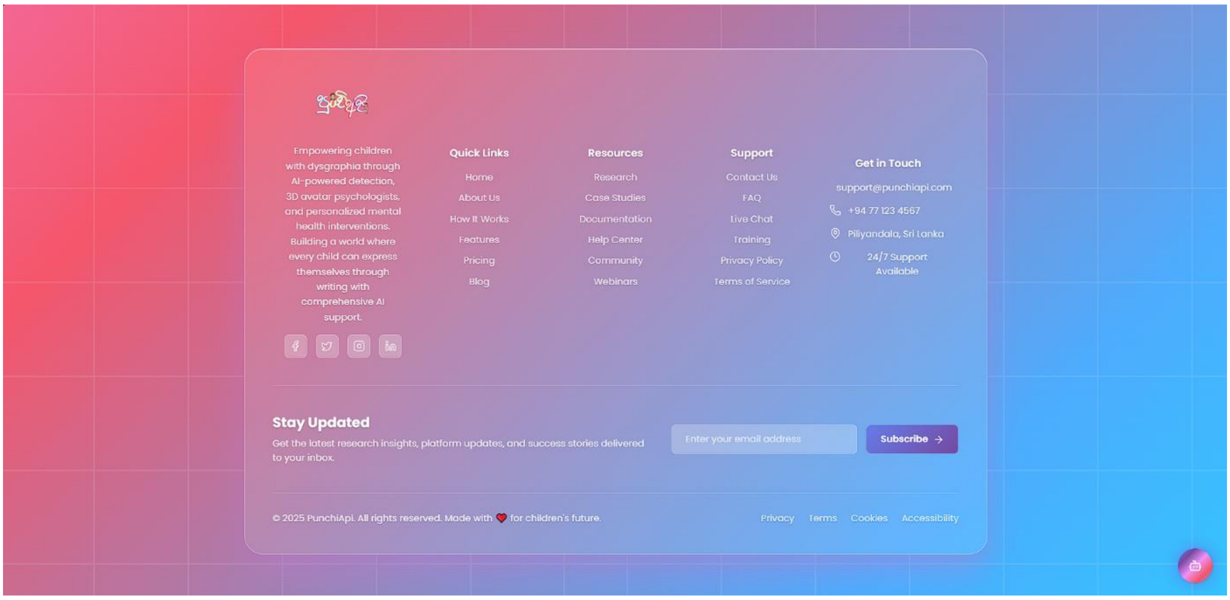



Dysgraphia Assessment Questionnaire
Complete a comprehensive screening tool to evaluate dysgraphia with instant results and professional recommendations.
Visit →
Voice & Text Interaction
Natural language processing enables seamless communication through both voice commands and text messaging.
Visit →
Collaborative Platform
Connect parents, teachers, and therapists in a unified platform for coordinated support and progress tracking.
Visit →
AI Mental Health Chatbots
Intelligent chatbots with voice and text capabilities provide 24/7 mental health support and counseling.
Visit →
Gamified Learning
Engaging games and activities that make writing practice fun and motivating for children.
Visit →
Evidence-Based
All interventions are backed by the latest research in occupational therapy and educational psychology.
Visit →


How Our Platform Works
Simple steps to transform your child's writing journey with our comprehensive dysgraphia support system.
1
Upload Writing Sample
Simply upload a photo or scan of your child's handwriting. Our system accepts various formats and automatically processes the image for analysis.
2
AI Analysis
Our advanced AI algorithms analyze letter formation, spacing, pressure, and other key indicators to detect potential dysgraphia patterns with clinical accuracy.
3
Detailed Report
Receive a comprehensive assessment report with specific findings, recommendations, and personalized intervention strategies tailored to your child's needs.
4
Start Interventions
Access personalized therapy exercises, games, and activities designed to improve writing skills while keeping your child engaged and motivated.
5
Track Progress
Monitor your child's improvement with real-time analytics, milestone celebrations, and regular progress reports shared with your support team.
6
Collaborate & Support
Connect with therapists, teachers, and other parents. Share progress, get expert advice, and build a supportive community around your child's success.

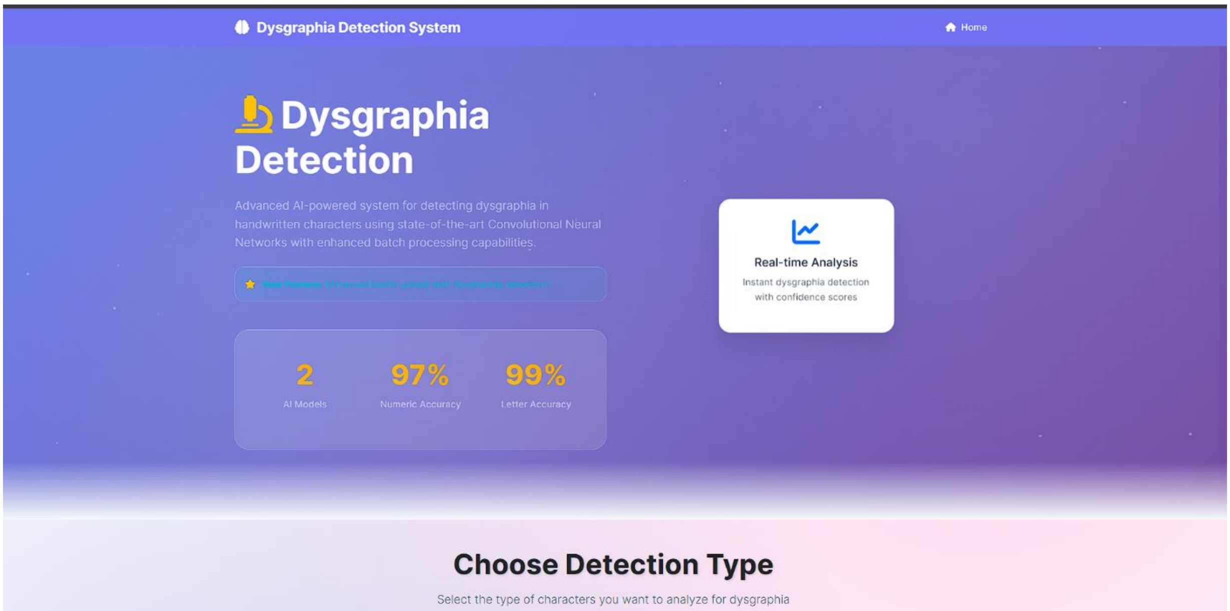

What Our Community Says
Real stories from families , educators, and professionals who have experienced the transformative impact of our platform.
"Finally, a tool that gives us hope and practical solutions. My daughter's writing has improved dramatically."
Manojy Wanninayake
Parent

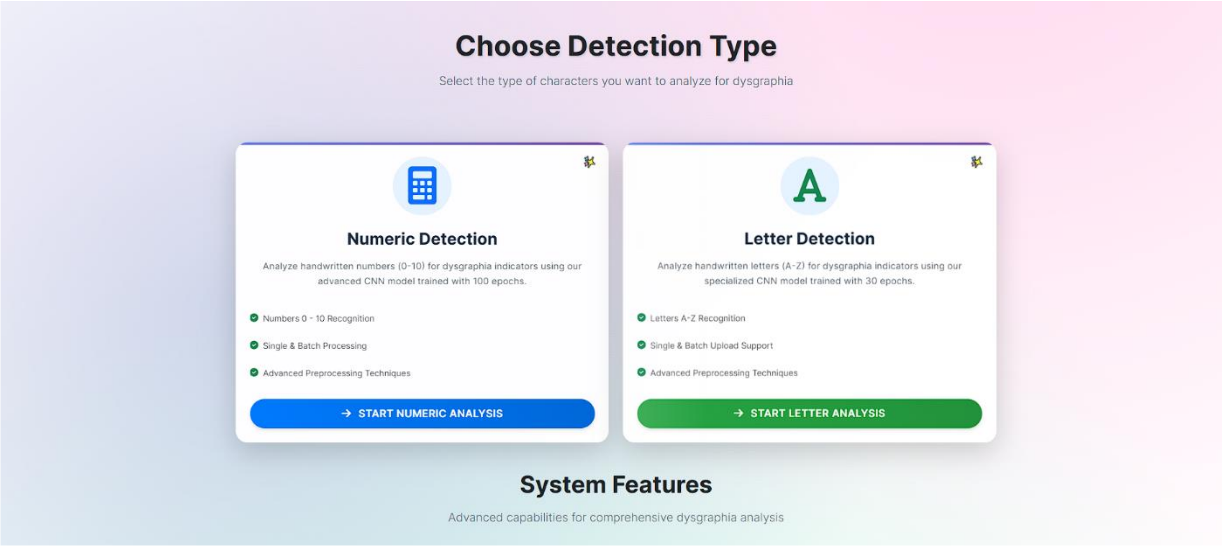


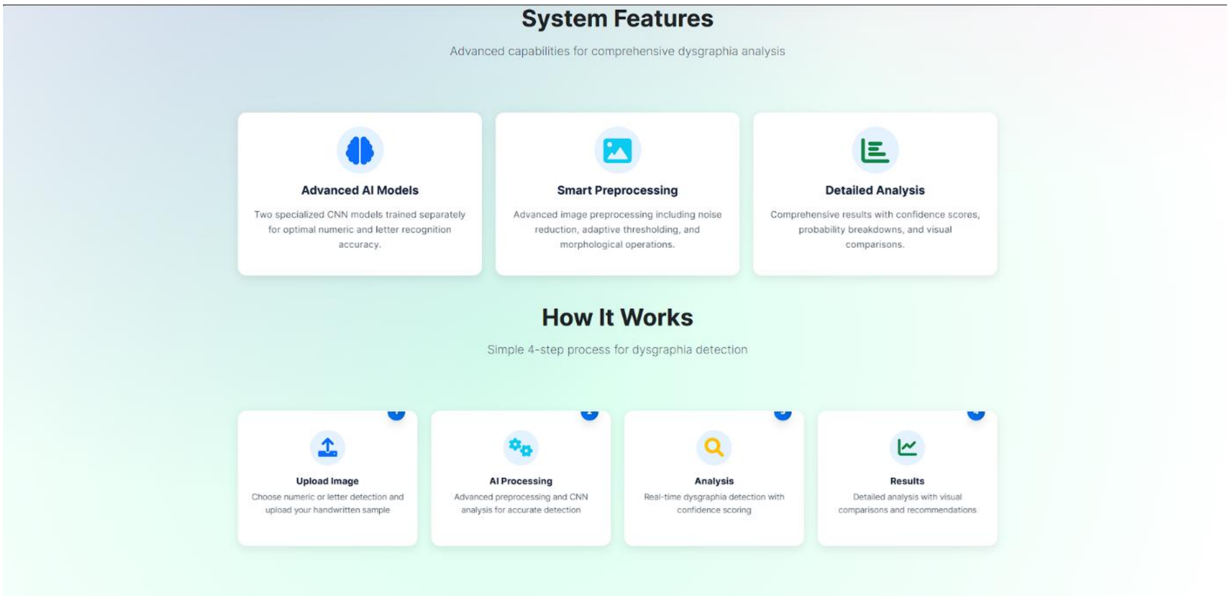


Features

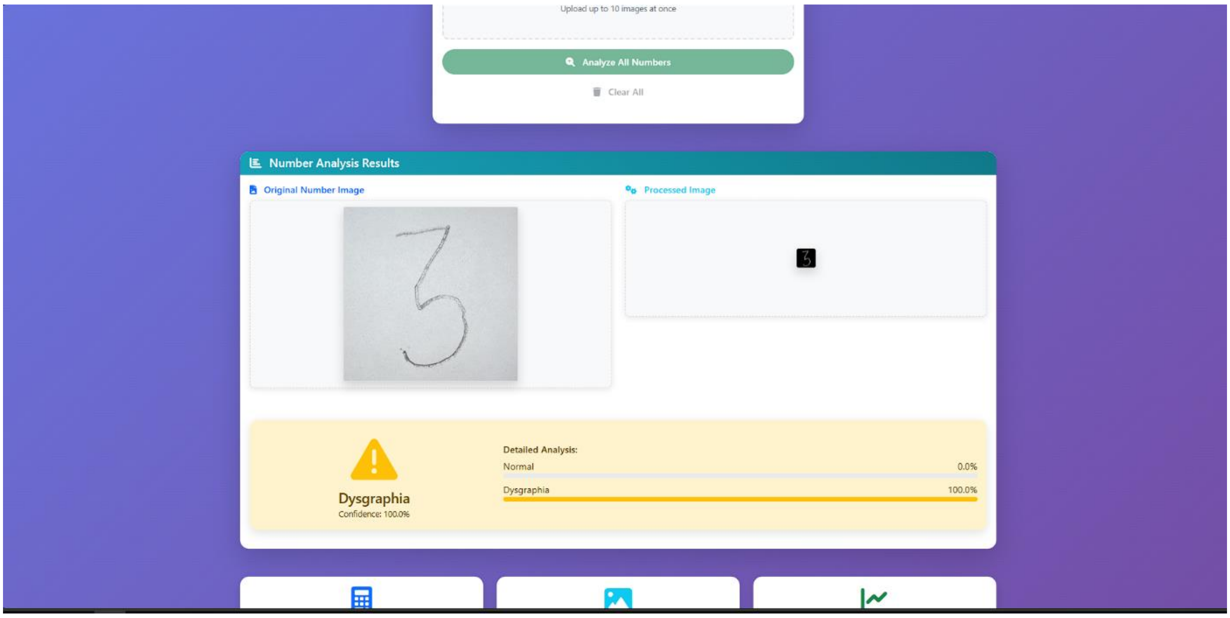
01 AI-Powered Detection



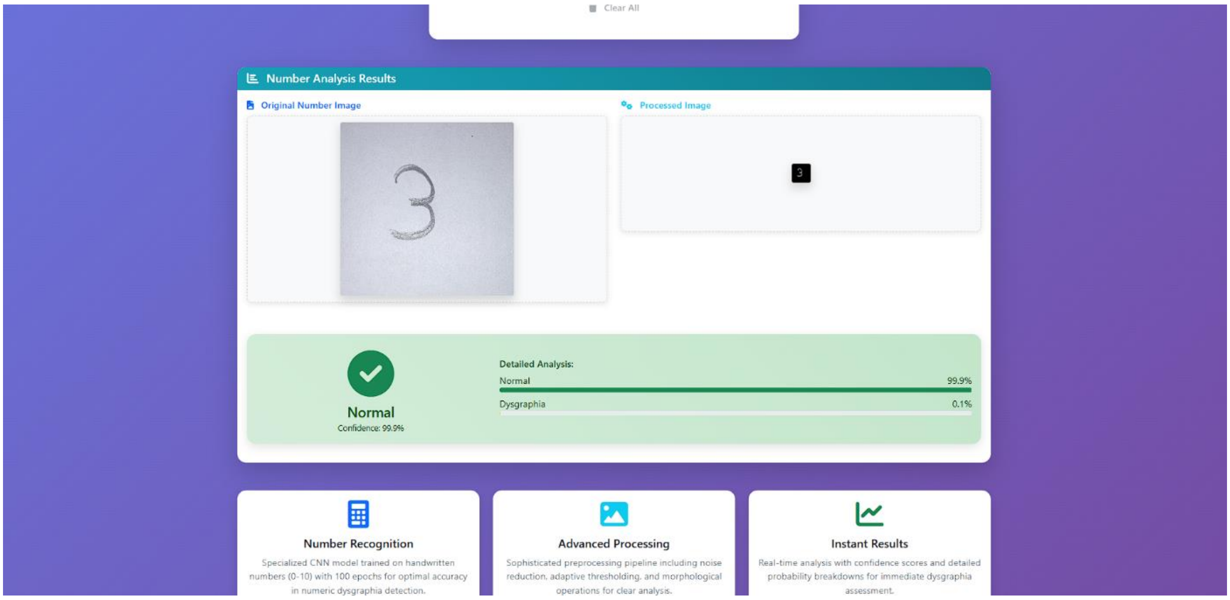




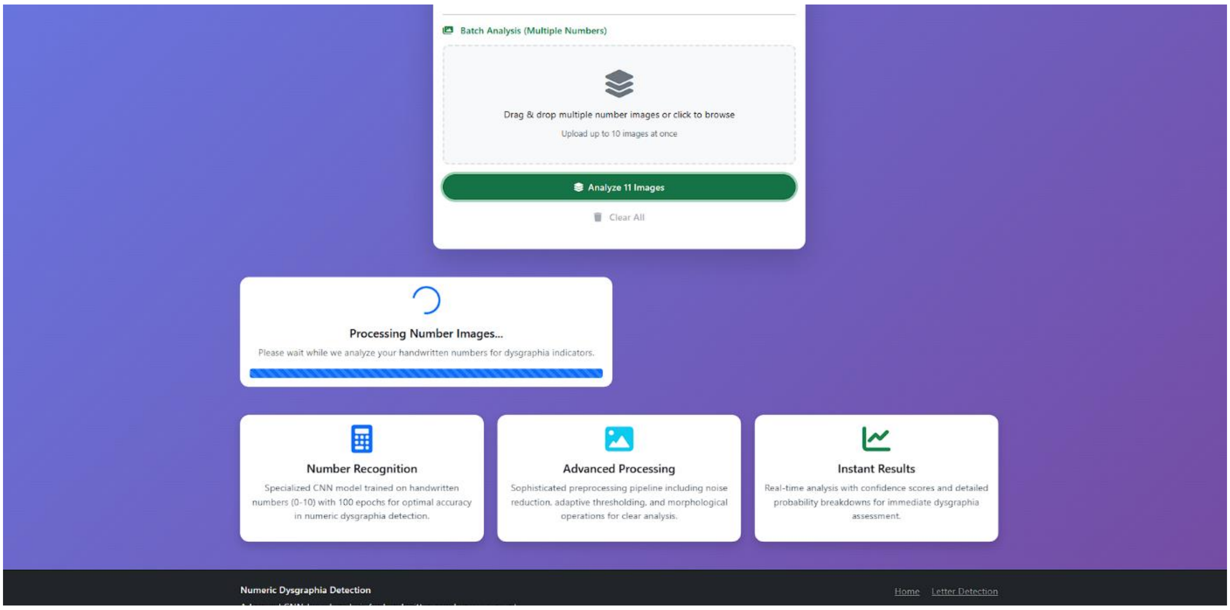
NUMERIC DYSGRAPHIA SINGLE IMAGE TEST - POSITIVE

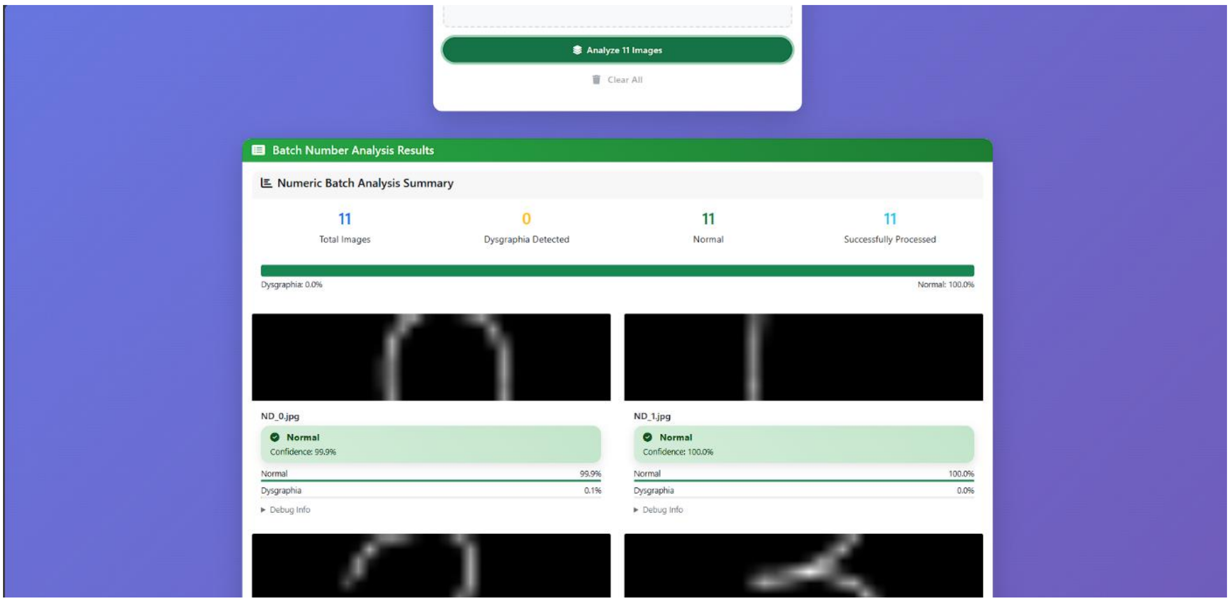


NUMERIC DYSGRAPHIA SINGLE IMAGE TEST - NEGATIVE

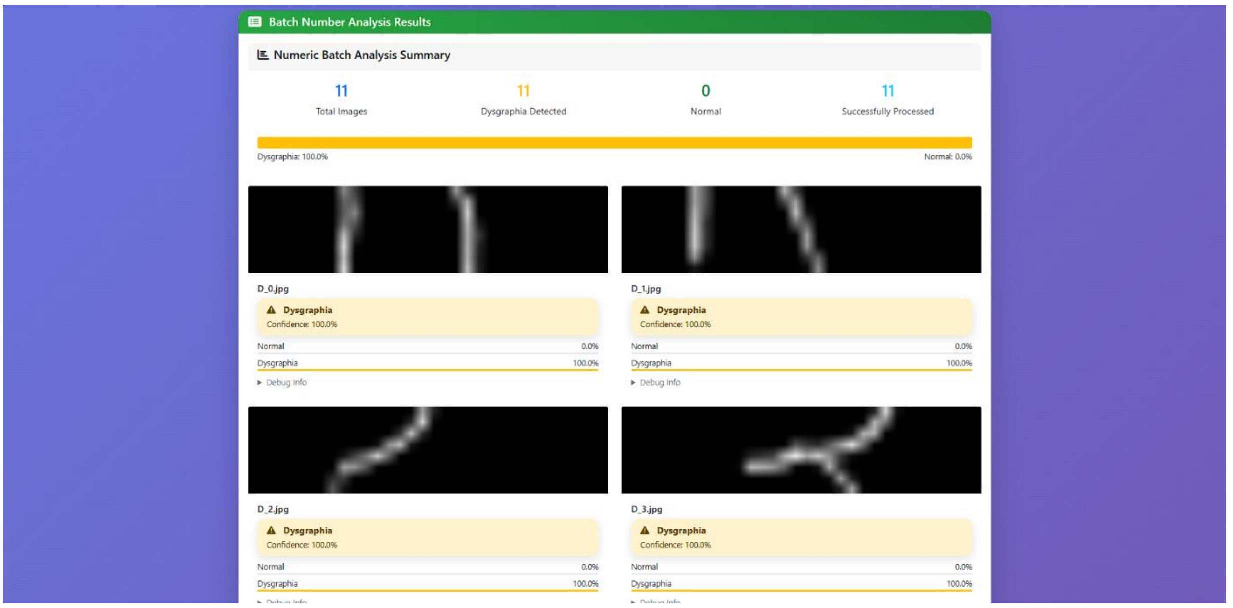


NUMERIC DYSGRAPHIA MULTIPLE IMAGE TEST - NEGATIVE

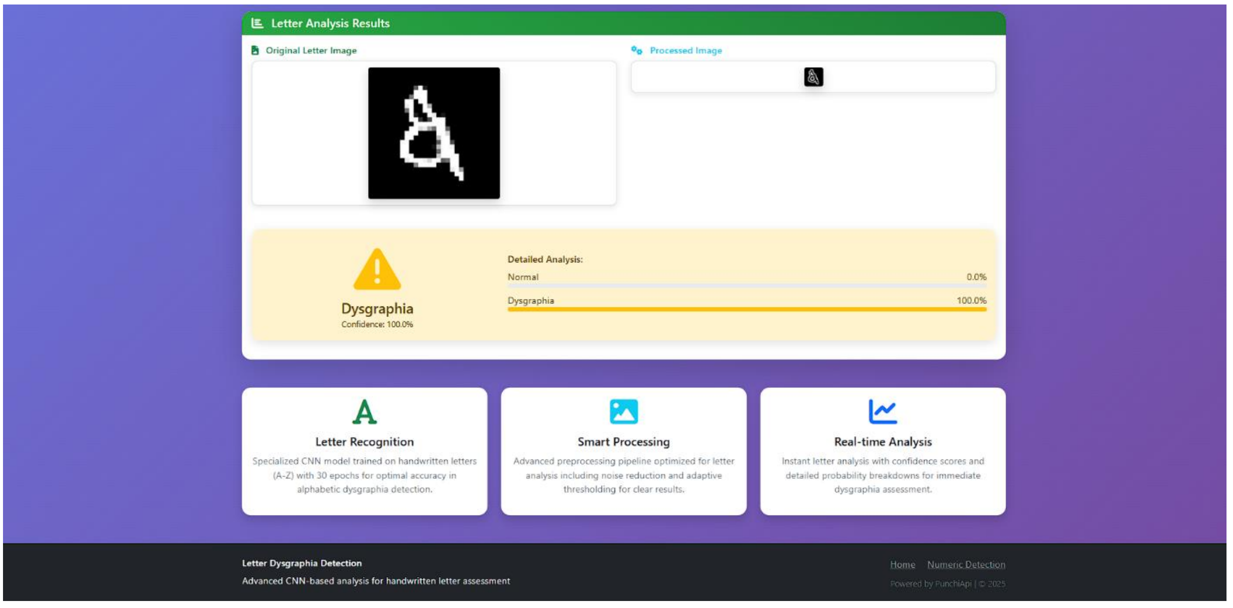




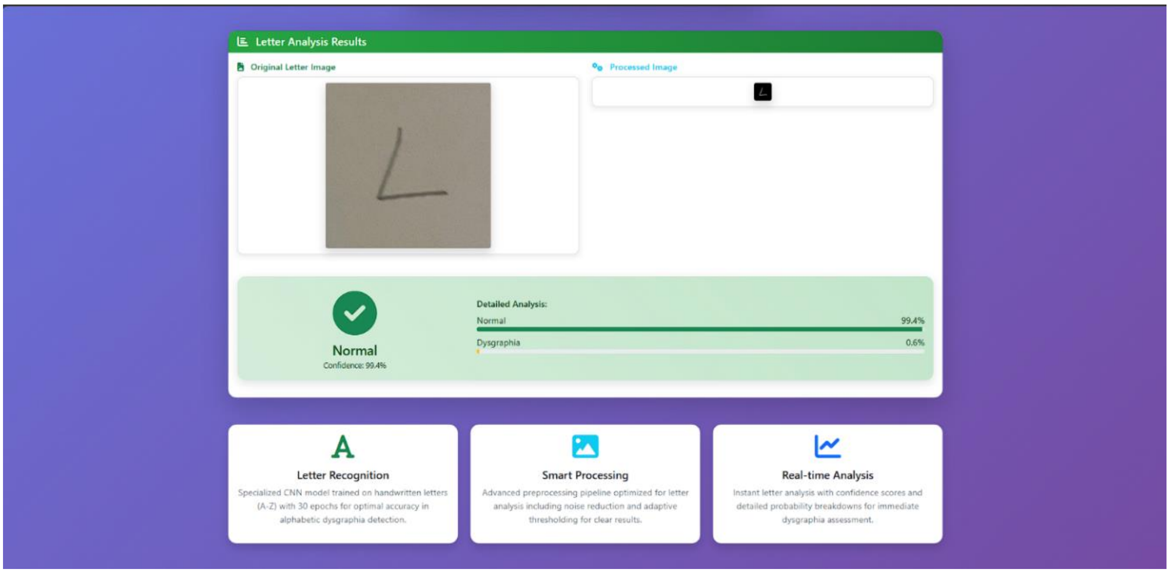
NUMERIC DYSGRAPHIA MULTIPLE IMAGE TEST - POSITIVE



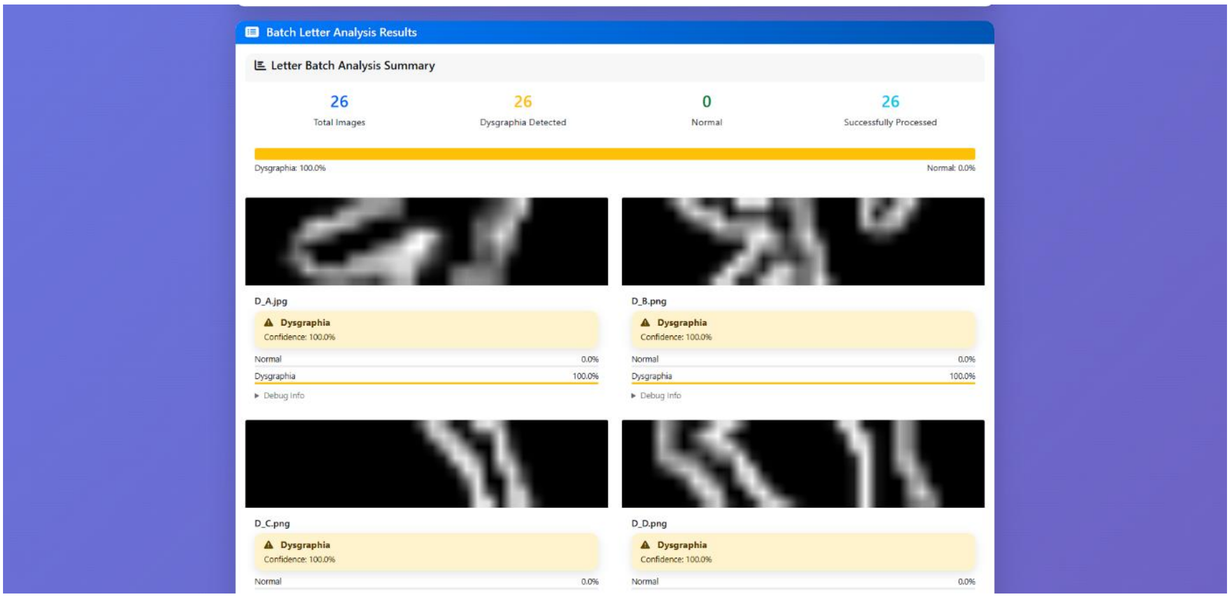
LETTER DYSGRAPHIA SINGLE IMAGE TEST - POSITIVE



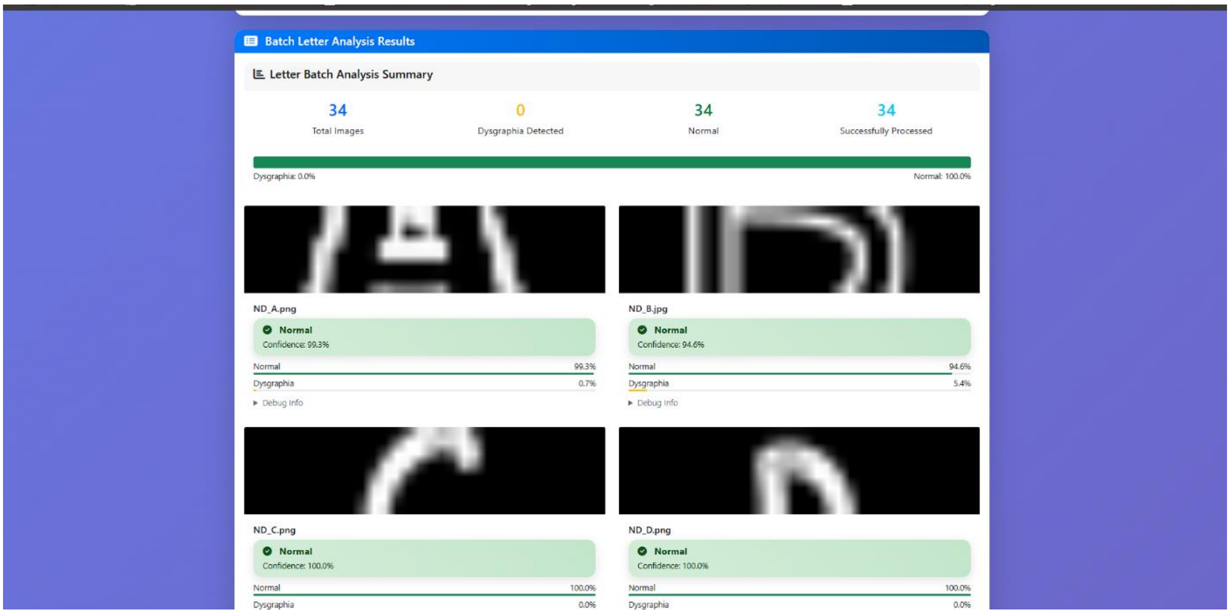
LETTER DYSGRAPHIA SINGLE IMAGE TEST - POSITIVE



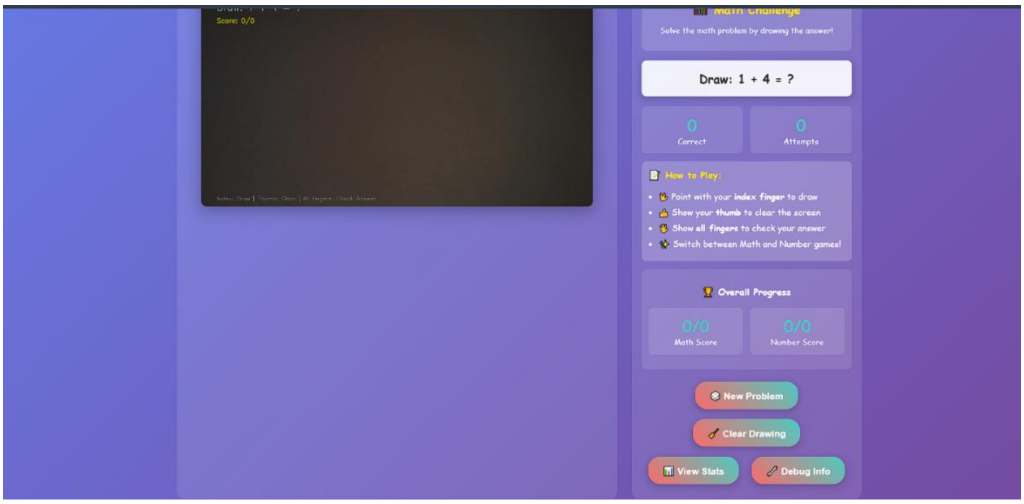
LETTER DYSGRAPHIA MULTIPLE IMAGE TEST - POSITIVE

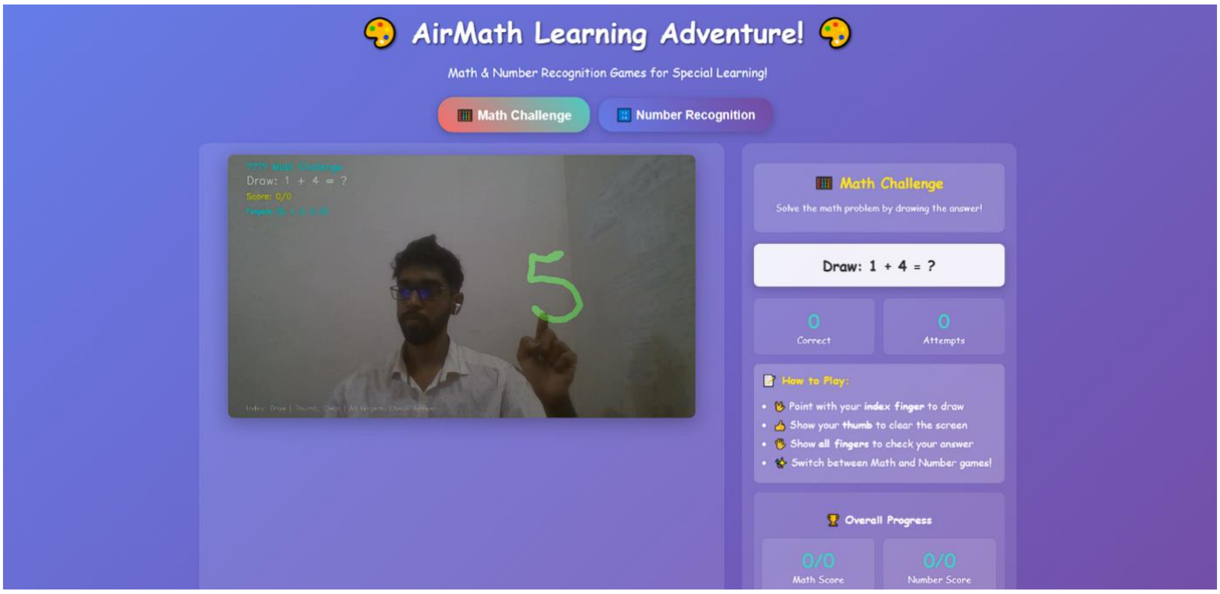


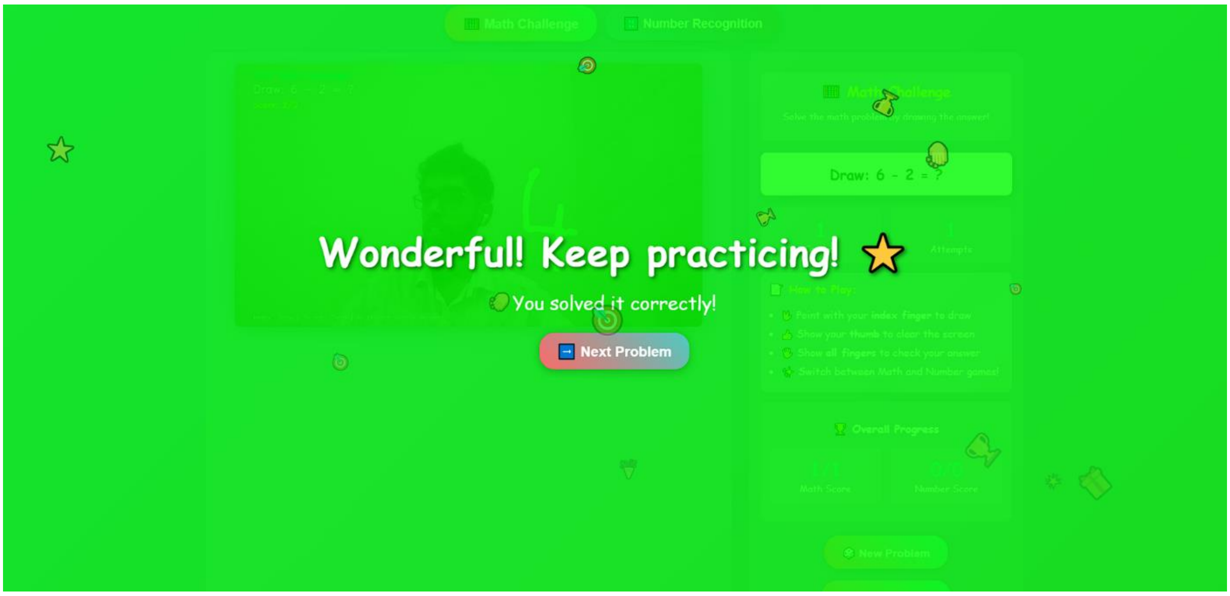
LETTER DYSGRAPHIA MULTIPLE IMAGE TEST - NEGATIVE

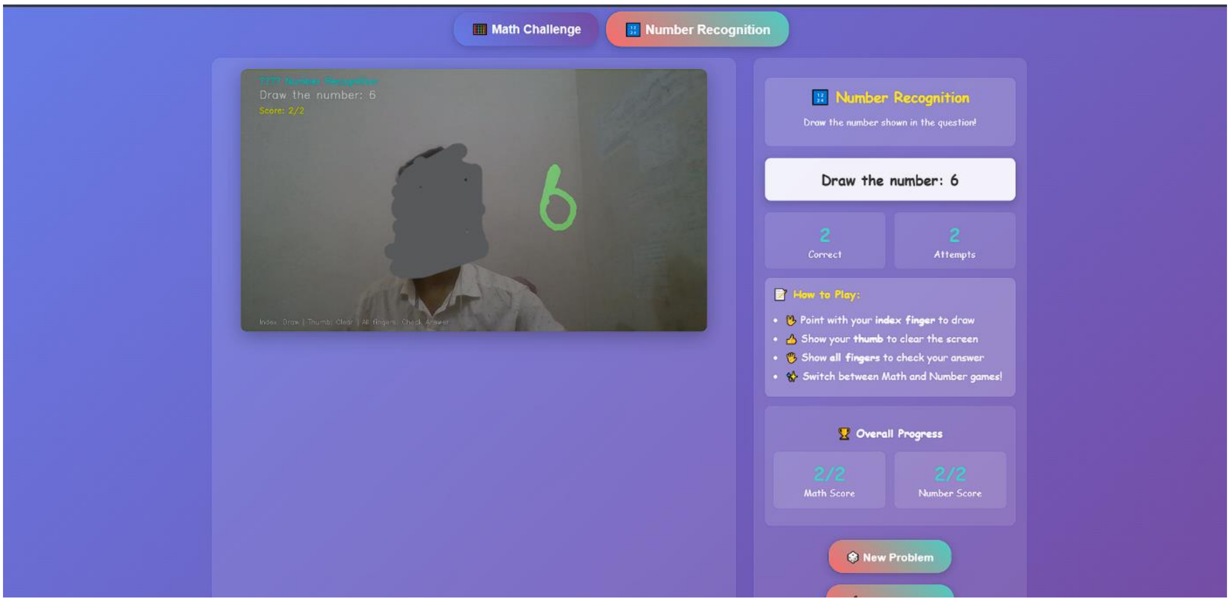


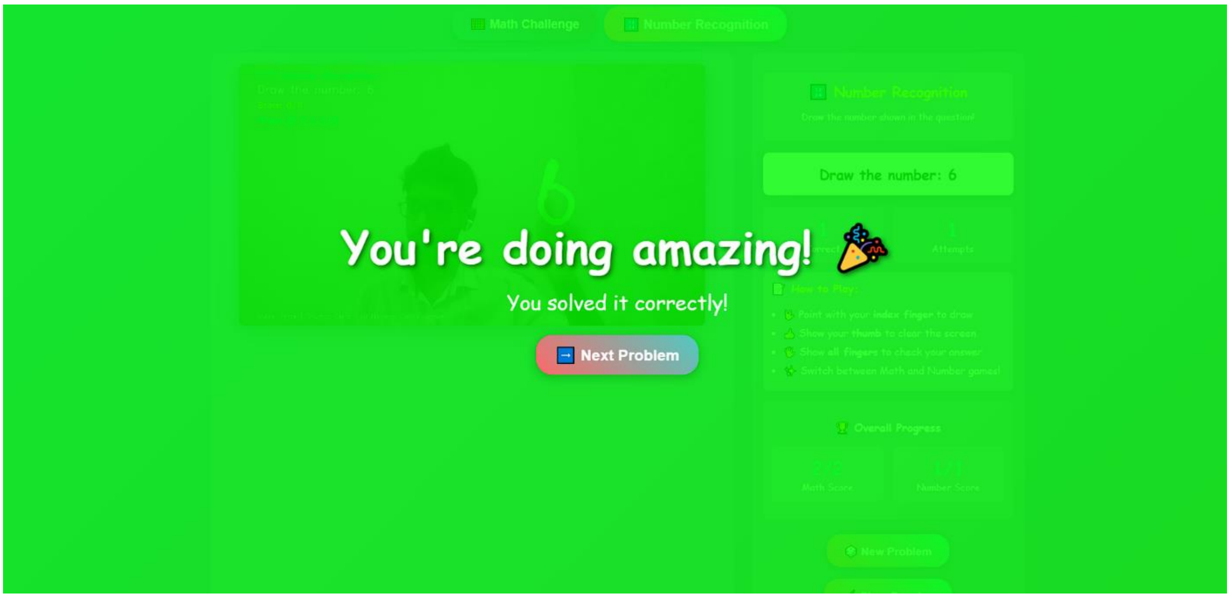
02 AirMath – Immersive Intervention



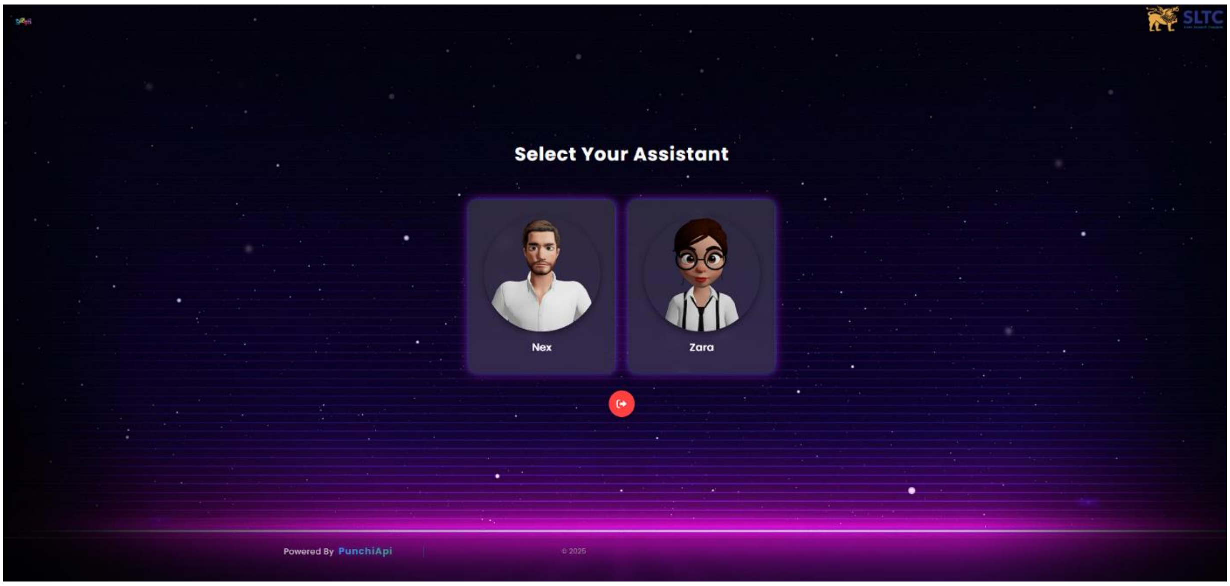

AirMath Learning Adventure!
Math & Number Recognition Games for Special Learning!
Math Challenge
Number Recognition
Draw: 1 + 4 = ?
5
Math Challenge
Solve the math problem by drawing the answer!
Draw: 1 + 4 = ?
0
Correct
0
Attempts
How to Play:
Point with your index finger to draw
Show your thumb to clear the screen
Show all fingers to check your answer
Switch between Math and Number games!
Overall Progress
0/0
Math Score
0/0
Number Score

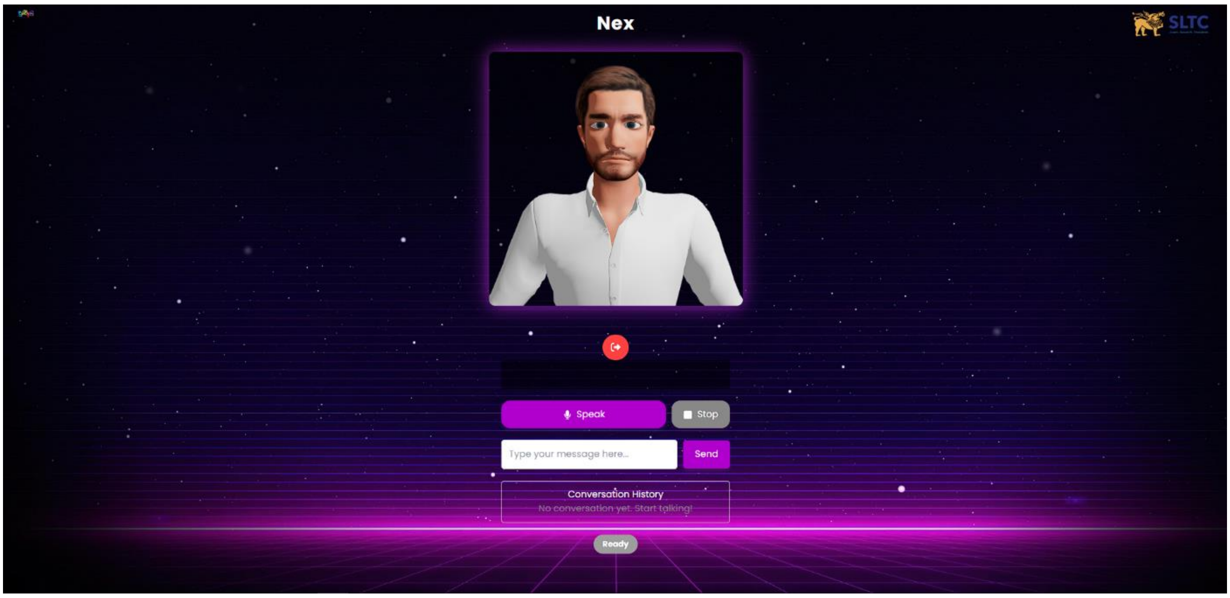

Wonderful! Keep practicing!
You solved it correctly!
Next Problem

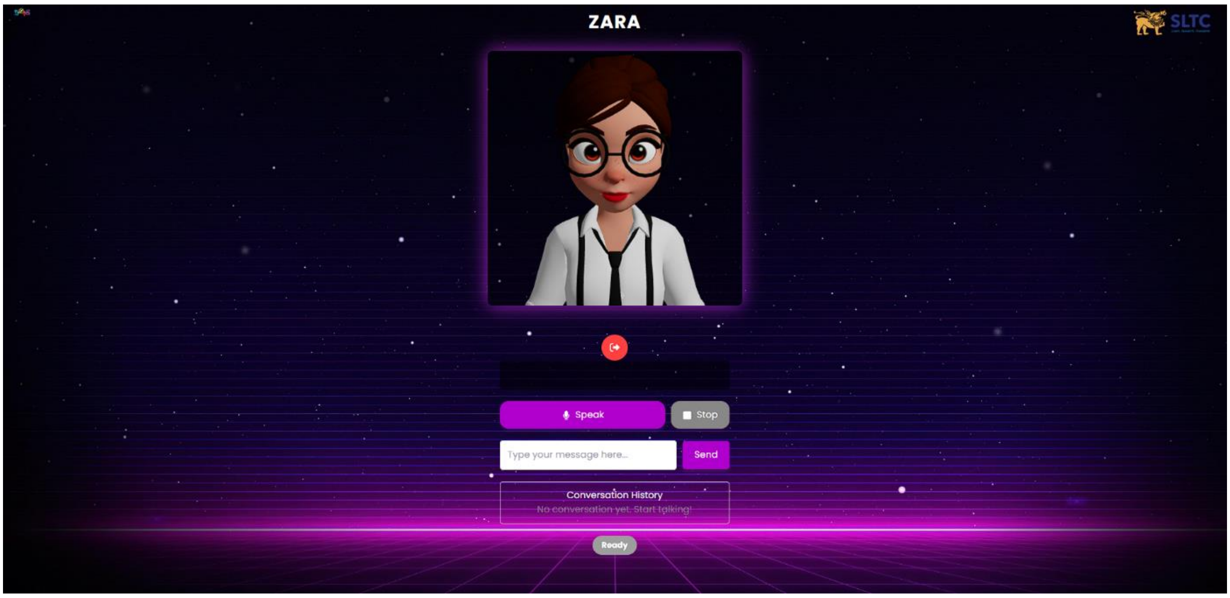

Math Challenge
Number Recognition
Draw the number: 6
Score: 2/2
6
Number Recognition
Draw the number shown in the question!
Draw the number: 6
2
Correct
2
Attempts
How to Play:
Point with your index finger to draw
Show your thumb to clear the screen
Show all fingers to check your answer
Switch between Math and Number games!
Overall Progress
2/2
Math Score
2/2
Number Score
New Problem



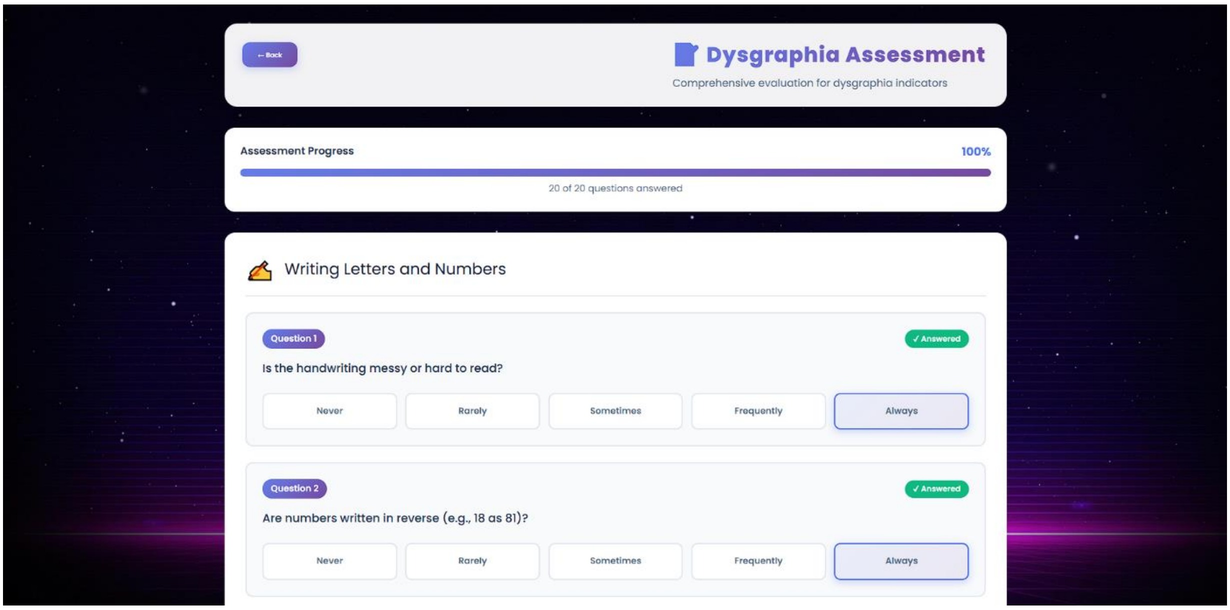
03 AI 3D Avatar Psychologists

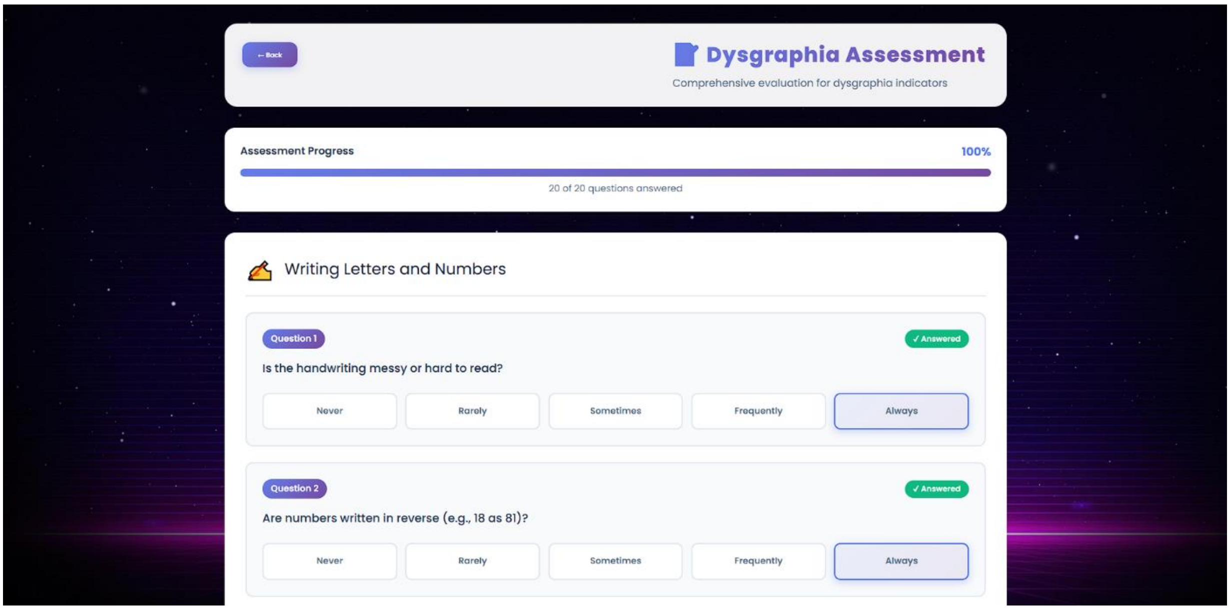


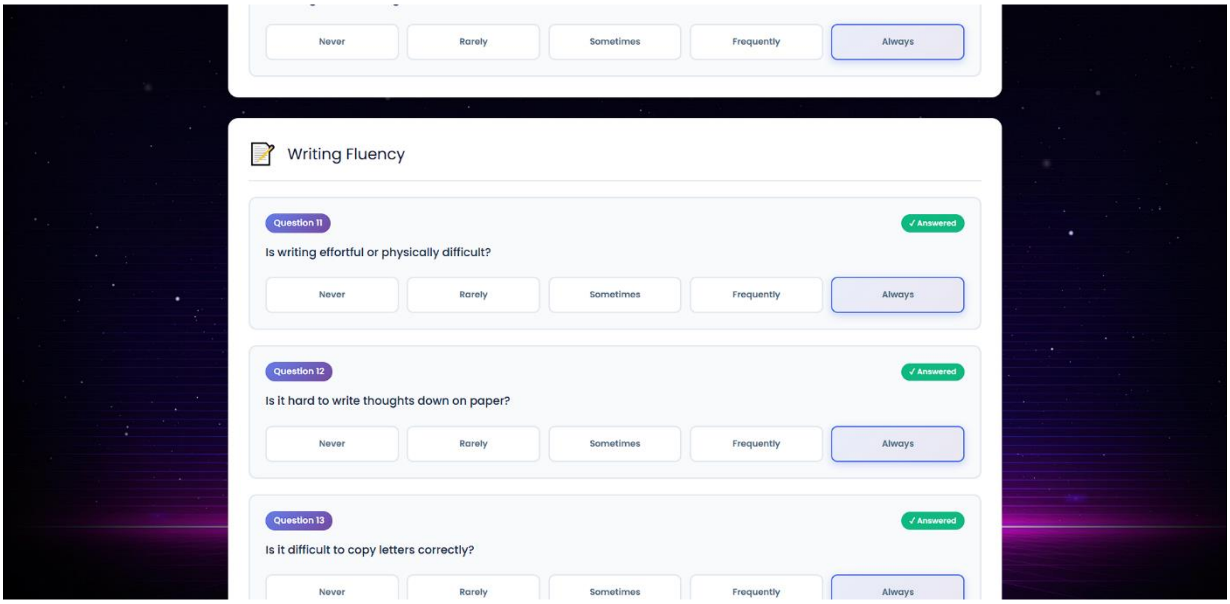


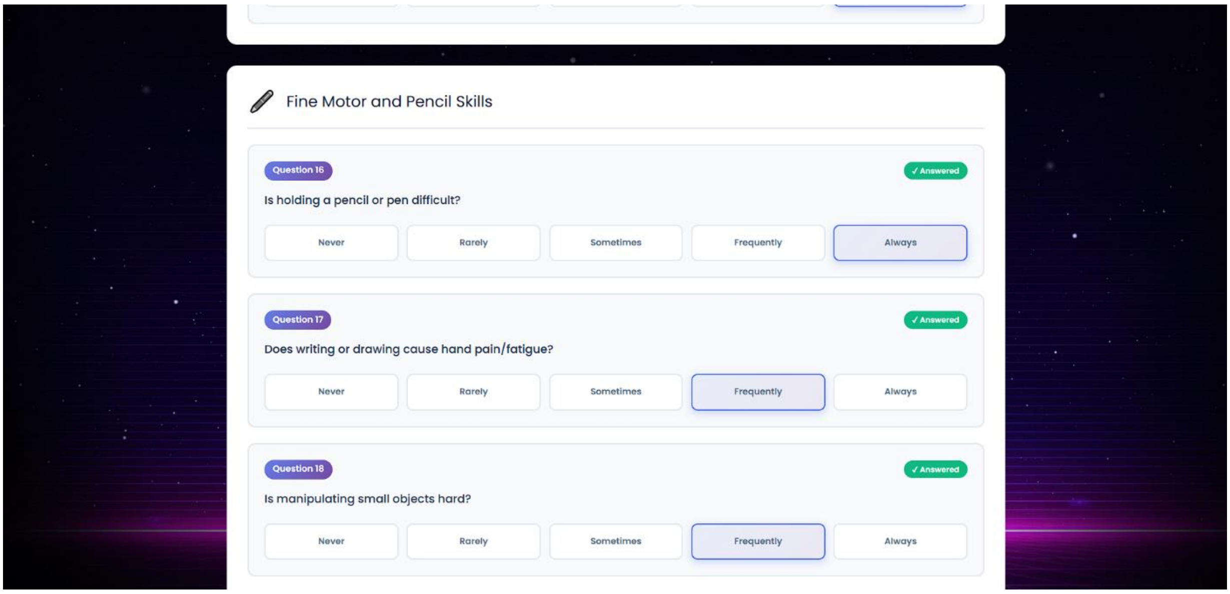


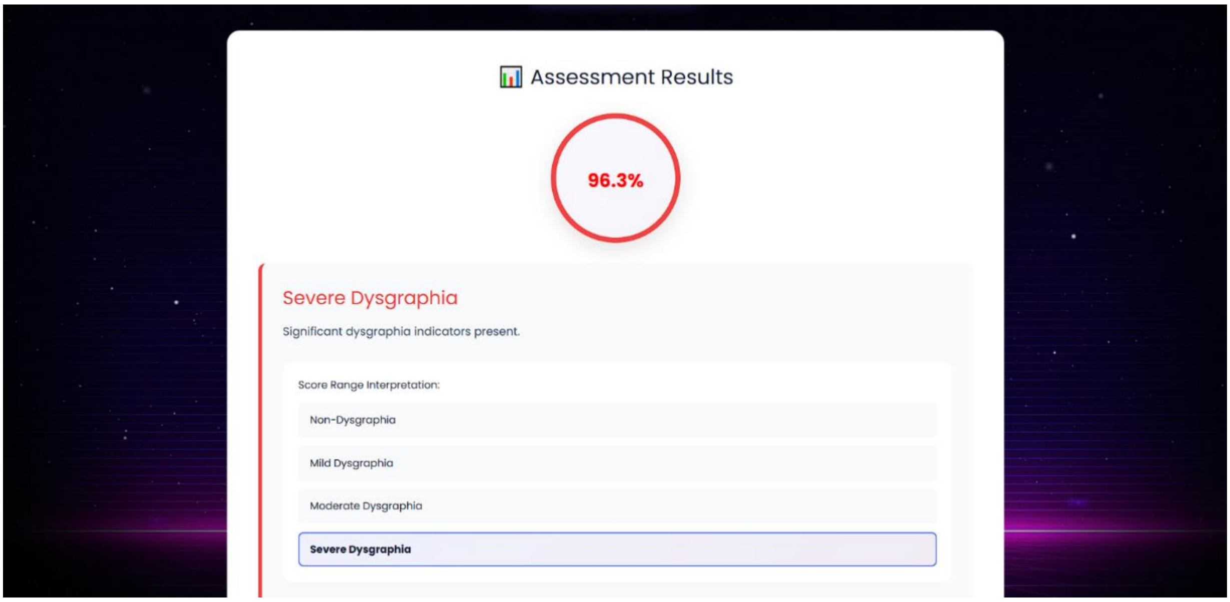
04 Dysgraphia Assessment Questionnaire

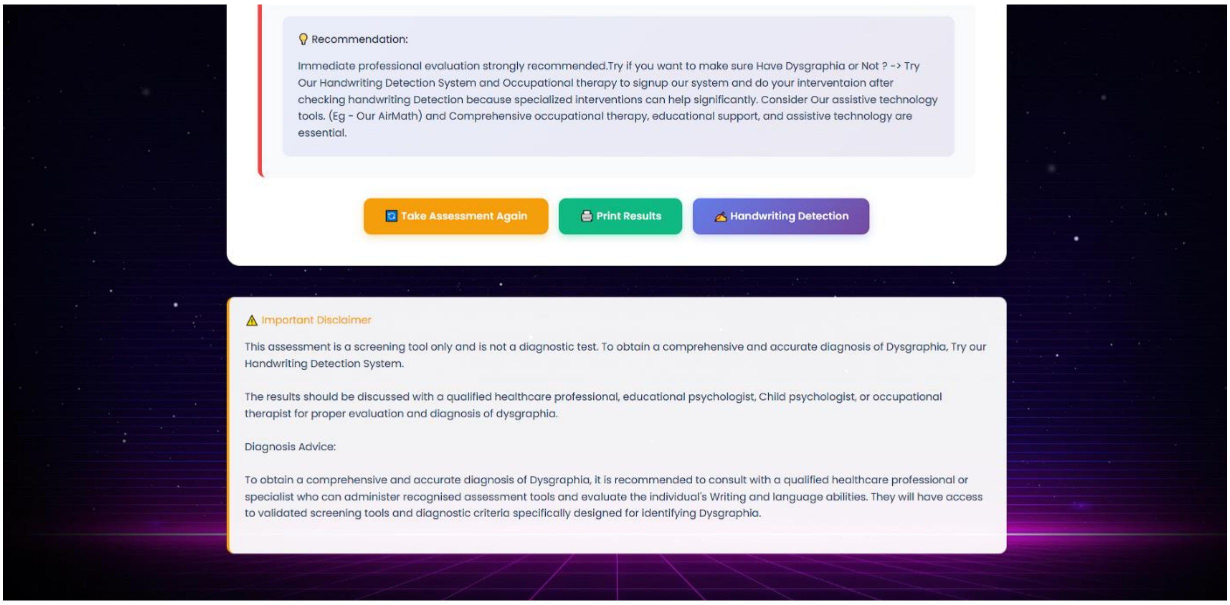




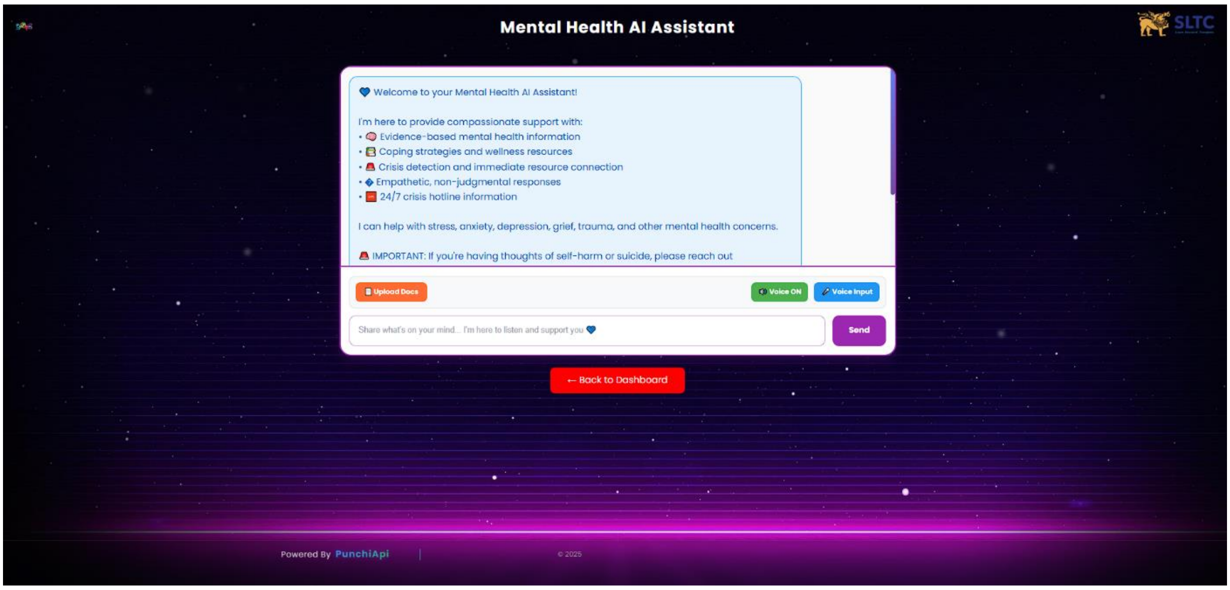

Never
Rarely
Sometimes
Frequently
Always
Writing Fluency
Question 11
✓ Answered
Is writing effortful or physically difficult?
Never
Rarely
Sometimes
Frequently
Always
Question 12
✓ Answered
Is it hard to write thoughts down on paper?
Never
Rarely
Sometimes
Frequently
Always
Question 13
✓ Answered
Is it difficult to copy letters correctly?
Never
Rarely
Sometimes
Frequently
Always


Fine Motor and Pencil Skills
Question 16
✓ Answered
Is holding a pencil or pen difficult?
Never
Rarely
Sometimes
Frequently
Always
Question 17
✓ Answered
Does writing or drawing cause hand pain/fatigue?
Never
Rarely
Sometimes
Frequently
Always
Question 18
✓ Answered
Is manipulating small objects hard?
Never
Rarely
Sometimes
Frequently
Always

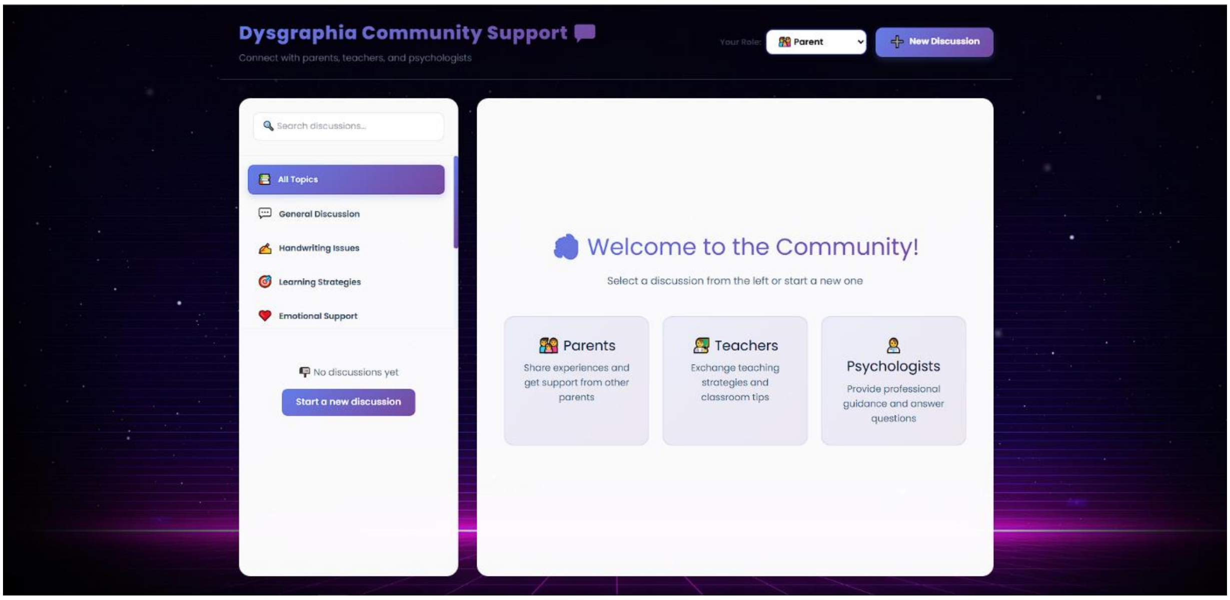

Assessment Results
96.3%
Severe Dysgraphia
Significant dysgraphia indicators present.
Score Range Interpretation:
Non-Dysgraphia
Mild Dysgraphia
Moderate Dysgraphia
Severe Dysgraphia

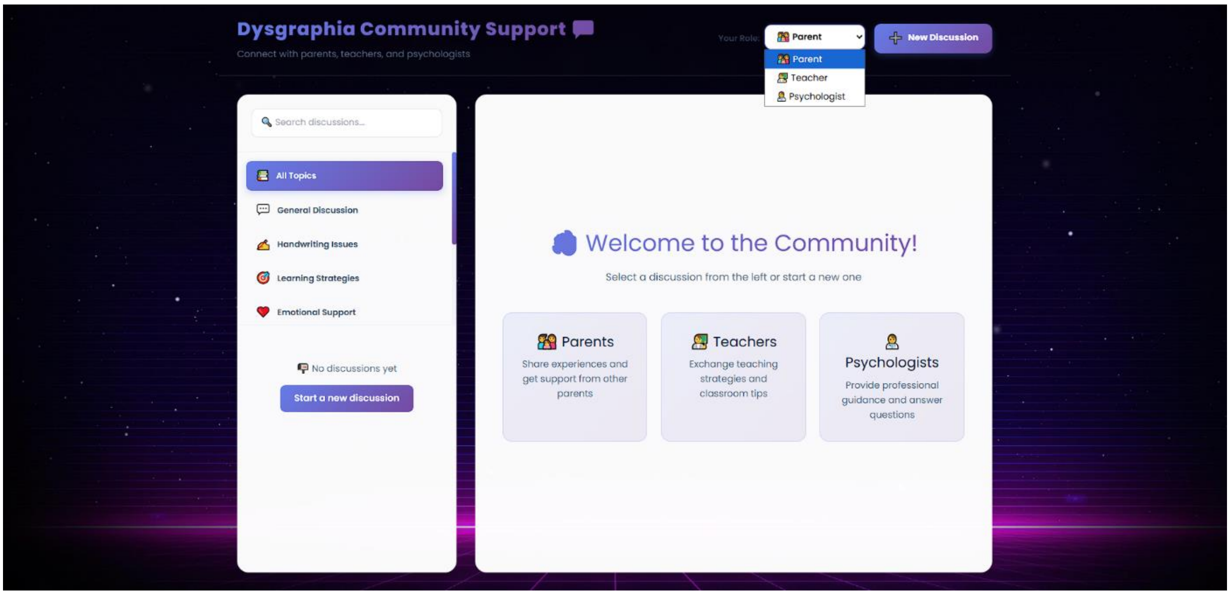


05 Voice & Text Interaction

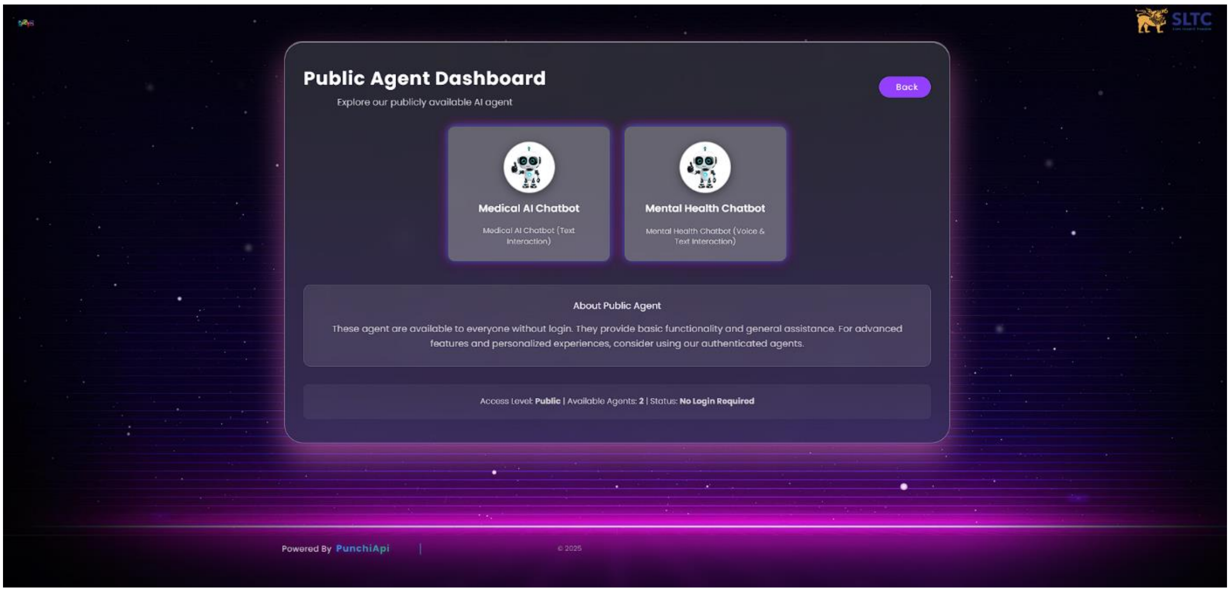


06 Collaborative Platform

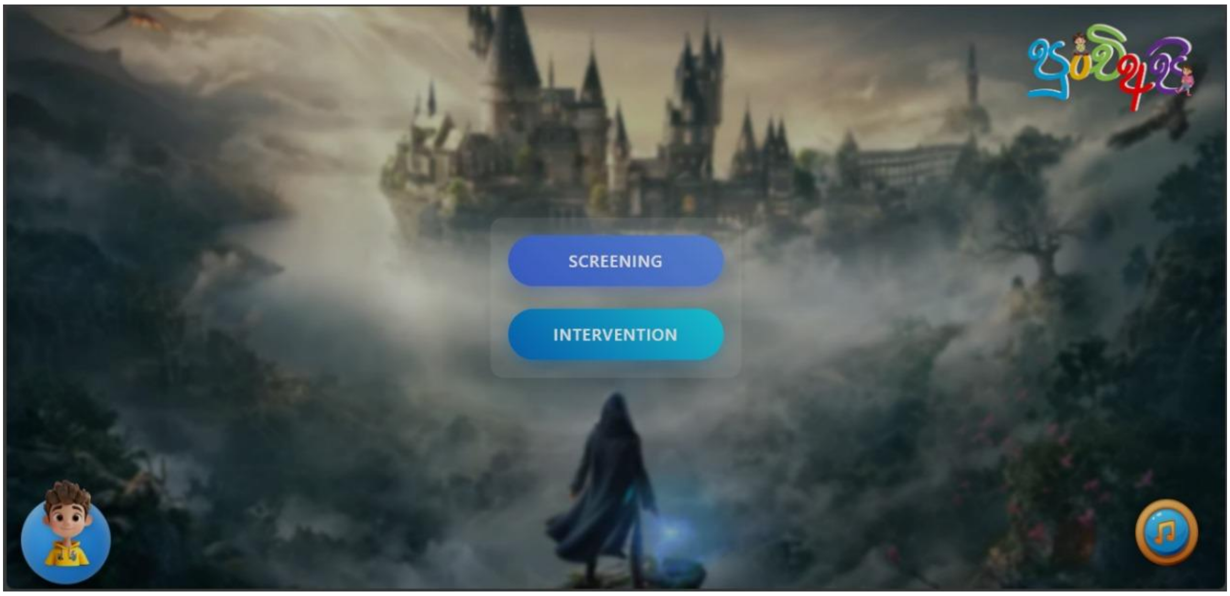




07 AI Mental Health Chatbots

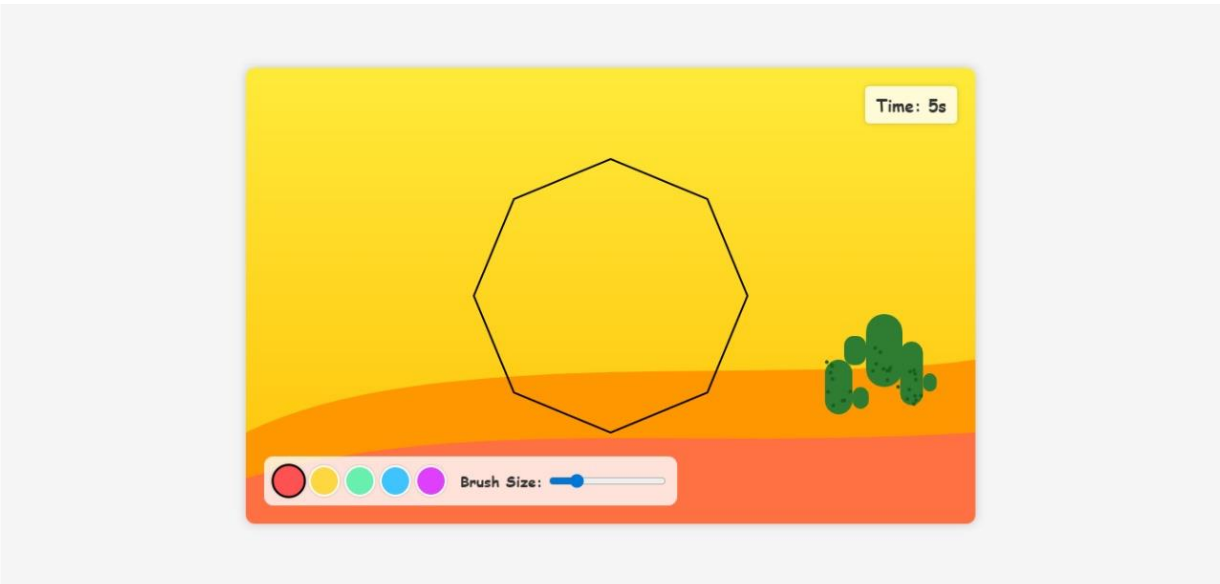


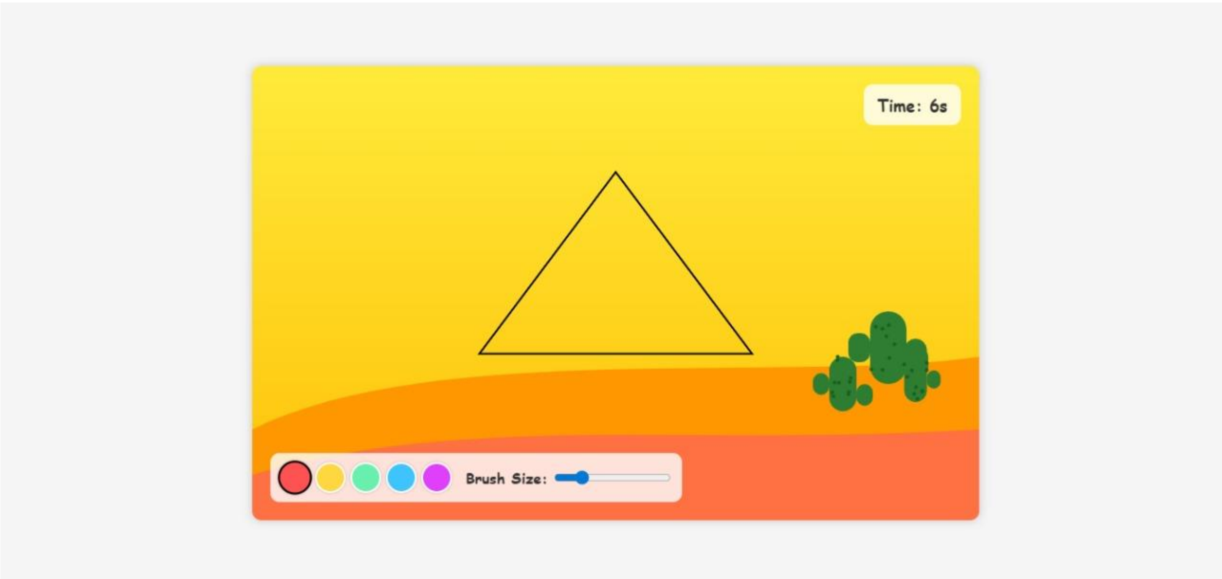
08 Gamified Learning (Gamified Intervention)




පොඩිත්තාට අවුරුදු ?
6
7
8
9
10
පොඩිත්තා
NEXT

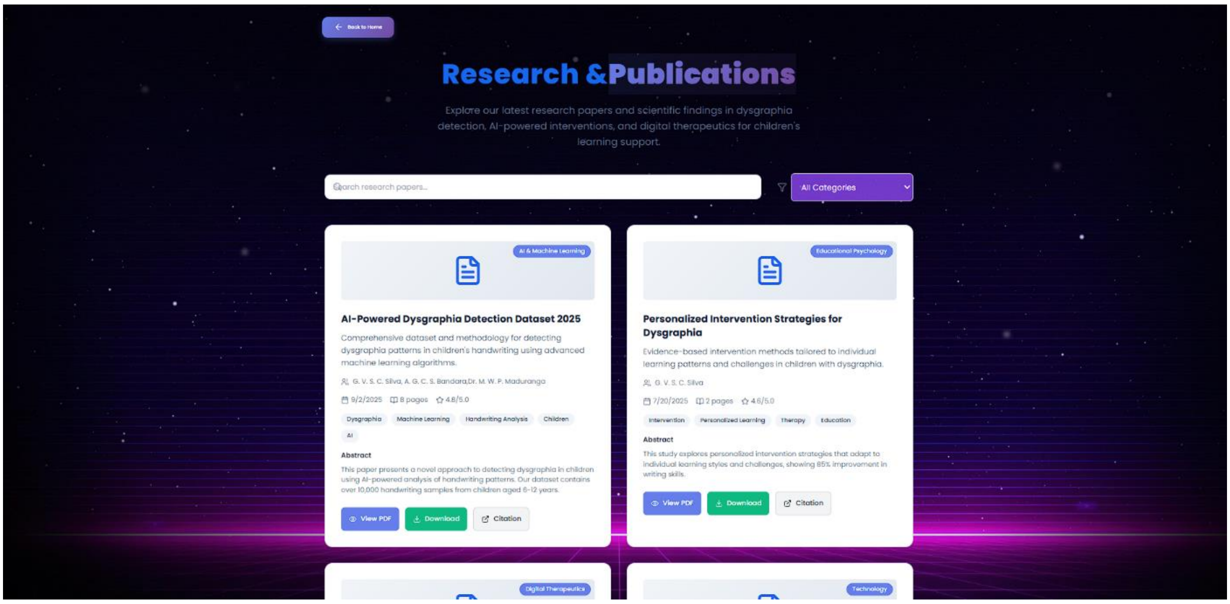

ඉලක්කම් ලිවීම
ඉලක්කම්
රූප


Time: 5s
Brush Size:



09 Evidence-Based

Listed in our published paper in 2025 about our collected dataset and other papers related to Dysgraphia




119TH ANNUAL SESSIONS
IESL ANNUAL RESEARCH SYMPOSIUM
September 02, 2025
Galadari Hotel, Colombo
The Institution of Engineers, Sri Lanka


119TH ANNUAL SESSIONS
IESL ANNUAL RESEARCH SYMPOSIUM
September 02, 2025
Galadari Hotel, Colombo
PRESENTER