Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)

Student’s Id and Name	21225103304-Saba-E-Zannat
Capstone Project Title	Automated Handwritten Grading System
Supervisor Name & Designation	Ahmed Shafkat, Assistant Professor
Course Teacher’s Name & Designation	Dr. Md shafiqul Islam Professor

Aspects	Paper#1 **A Novel Handwriting Grading System Using Gurmukhi Characters**
Problem Statement	The paper addresses the challenge of objectively grading handwriting in competitions, which traditionally relies on subjective human judgment. It highlights the lack of automated systems for evaluating handwriting quality, particularly for Gurmukhi script, and the need for a reliable method to compare handwritten characters against standardized printed fonts to assess writing superiority.
Key Contributions	• The study introduces a novel handwriting grading system for Gurmukhi characters, utilizing peak extent and modified division points-based features. It achieves automated grading of 100 writers using a Nearest Neighbor (NN) classifier, with the printed font Anandpur Sahib scoring highest, validating the system's effectiveness. The approach can be extended to other scripts with appropriate datasets, offering a decision support system for

	handwriting competitions.
Objectives/Goal	The objective is to develop an automated system to grade writers based on their handwriting quality in Gurmukhi script, comparing handwritten characters to printed fonts. The goal is to establish a reliable, objective method for ranking participants in handwriting competitions using classification scores.
Methodology/Theory	• The methodology involves digitizing and preprocessing handwritten Gurmukhi characters, followed by feature extraction using peak extent (horizontal and vertical pixel extents) and modified division points (balanced projection-based subdivisions). Images are resized to 88x88 pixels, divided into zones (4^L for peak extent, 100 for division points), and features are normalized. The NN classifier computes Euclidean distances between candidate and stored feature vectors to assign classification scores, which are used for grading.
Software Tools/Setup Details	The system uses Nearest Neighborhood Interpolation (NNI) for resizing images to 88x88 pixels. No specific software is mentioned for digitization or preprocessing, but the NN classifier is implemented for classification. The authors do not detail additional tools for feature extraction or data processing.
Test/Experiment	The experiment involves grading 100 writers and one printed font (Anandpur Sahib) using peak extent and modified division points features. Parameters include classification scores normalized to [0,100], derived from NN classifier Euclidean distances. Four printed

	Gurmukhi fonts (LMP_Taran, Maharaja, Granthi, Gurmukhi_Lys) form the training dataset, and testing compares writers' scores against the Anandpur Sahib font.
Test Data/Dataset Source	The testing dataset comprises handwritten Gurmukhi characters from 100 writers and the printed Anandpur Sahib font. The training dataset includes characters from four printed Gurmukhi fonts: LMP_Taran, Maharaja, Granthi, and Gurmukhi_Lys, collected specifically for this study.
Result Analysis	The Anandpur Sahib font scored 100 in all grading scenarios, confirming its superiority. For peak extent features, writer W64 scored highest (41.84); for modified division points, W53 scored 57.82; and for averaged features, W30 scored 41.57. These results demonstrate the system's ability to differentiate handwriting quality, with printed fonts outperforming human writers as expected.
Obstacles/Challenges	• The paper does not explicitly mention obstacles, but implicitly, challenges include collecting a diverse dataset from 100 writers and ensuring feature robustness across varied handwriting styles. Extending the system to other scripts requires building new datasets, which could be resource-intensive.
Terminology/Keywords	Handwriting grading, Gurmukhi script, feature extraction, peak extent features, modified division points, Nearest Neighbor classifier, character recognition, classification score, Euclidean distance, normalization.
Final Summary	The paper presents a novel automated handwriting grading system for Gurmukhi script, addressing the problem of subjective grading in competitions by using peak extent and modified division points-based features. It

	employs a Nearest Neighbor classifier to grade 100 writers against four printed fonts, with the Anandpur Sahib font scoring highest (100), and top writers achieving scores like 41.84 (W64) and 57.82 (W53). The methodology involves resizing images to 88x88 pixels, extracting features from zoned images, and normalizing scores, with no specific software detailed beyond NNI. The system, tested on a custom dataset, proves effective but faces challenges in dataset diversity and script extensibility, offering a scalable solution for objective handwriting evaluation

Department of Computer Science and Engineering Bangladesh University of Business and Technology (BUBT)

Student’s Id and Name	21225103304-Saba-E-Zannat
Capstone Project Title	Automated Handwritten Grading System
Supervisor Name & Designation	Ahmed Shafkat, Assistant Professor
Course Teacher’s Name & Designation	Dr. Md shafiqul Islam Professor

Aspects	Paper#2 **Use of technological tools to evaluate handwriting production of the alphabet and pseudocharacters by Brazilian students**
Problem Statement	The study addresses the lack of Brazilian research on handwriting automation, particularly the interaction between central (cognitive) and peripheral (motor) processes in letter production. It highlights the absence of systematized teaching of letter-writing movements in Brazil, which may hinder automation, impacting academic performance. The problem is compounded by limited use of technological tools to evaluate dynamic handwriting subprocesses, such as latency, duration, and fluency, in Brazilian students
Key Contributions	• The study provides the first Brazilian analysis of handwriting using digital tablets, comparing 3rd to 5th-grade students’ performance in writing alphabet letters and pseudocharacters. It demonstrates progression in handwriting automation by 5th grade, with reduced latency, duration, and

	improved fluency. The findings emphasize the need for explicit motor instruction to enhance letter production and suggest educational interventions to support handwriting proficiency
Objectives/Goal	The objective is to characterize and compare handwriting performance of Brazilian students from 3rd to 5th grade, focusing on latency, letter production duration, and movement fluency in writing alphabet letters and pseudocharacters. The study aims to identify automation progression and assess the impact of instructional gaps on handwriting efficiency
Methodology/Theory	• The methodology involves a comparative study of 95 right-handed students (aged 8–11) from 3rd (GI), 4th (GII), and 5th (GIII) grades, using a digital tablet (Wacom Intuos 5) and Ductus software. Students performed two tasks: writing uppercase alphabet letters and copying pseudocharacters. Three variables were measured: latency (time from stimulus to writing start), letter production duration (time to write each letter), and movement fluency (velocity peaks per stroke). Statistical analysis used ANOVA and Tukey tests to compare groups, with normalization by stroke count.
Software Tools/Setup Details	The study utilized a notebook with adapted Ductus software, connected to a Wacom Intuos 5 digital tablet for stimulus presentation and movement analysis. Statistical analyses were performed using SPSS V20, Minitab 16, and Excel Office 2010, ensuring robust data processing and visualization
Test/Experiment	• The experiment involved 95 students divided into three groups (GI: 27 3rd-graders, GII: 37 4th-graders, GIII: 31

	5th-graders) performing handwriting tasks. Tasks included writing uppercase alphabet letters and copying pseudocharacters. Parameters measured were latency (ms), letter production duration (ms), and movement fluency (cm/ms, normalized by stroke count). ANOVA and Tukey tests assessed significant differences ($p<0.05$) across groups and letters
Test Data/Dataset Source	Source The dataset comprised handwriting performance data from 95 Brazilian students, collected via digital tablet tasks. No external datasets were used; data were generated through controlled experiments, with inclusion criteria ensuring satisfactory academic performance and exclusion of students with sensory, motor, or cognitive impairments.
Result Analysis	• Results showed a progression in handwriting automation from 3rd to 5th grade, with GIII exhibiting lower latency, shorter letter production duration, and improved fluency for most letters (e.g., E, H, R, S). Latency was higher for pseudocharacters, indicating incomplete motor program automation. Statistical significance ($p<0.05$) was observed for letters like E and S, suggesting 5th-graders achieved faster, more automated writing, though fluency slightly regressed due to increased precision demands
Obstacles/Challenges	• The study notes the lack of systematized calligraphy teaching in Brazil, particularly the delayed introduction of cursive writing until 3rd grade, impacting automation. Limited prior Brazilian research on computerized handwriting evaluation posed methodological challenges.

	Evaluating younger students (1st–2nd grades) was difficult due to incomplete letter trace mastery, as per the national curriculum
Terminology/Keywords	Common terms include handwriting automation, latency, letter production duration, movement fluency, motor programs, pseudocharacters, digital tablet, graphomotor gestures, cognitive load, and action grammar
Final Summary	This study addresses the gap in Brazilian research on handwriting automation by evaluating 3rd to 5th-grade students' performance using a Wacom Intuos 5 tablet and Ductus software, focusing on latency, duration, and fluency. It finds progression toward automation by 5th grade, with reduced latency and duration, though fluency slightly regresses due to precision demands. The absence of systematized calligraphy teaching and limited prior research pose challenges, highlighting the need for explicit motor instruction. Statistical analysis using SPSS, Minitab, and Excel reveals significant improvements in 5th-graders' handwriting efficiency, despite higher pseudocharacter latency indicating incomplete automation.

Department of Computer Science and Engineering Bangladesh University of Business and Technology (BUBT)

Student’s Id and Name	21225103304-Saba-E-Zannat
Capstone Project Title	Automated Handwritten Grading System
Supervisor Name & Designation	Ahmed Shafkat, Assistant Professor
Course Teacher’s Name & Designation	Dr. Md shafiqul Islam Professor

Aspects	Paper#3 **Handwritten Code Recognition for Pen-and-Paper CS Education**
Problem Statement	The paper addresses the challenge of recognizing handwritten code for pen-and-paper Computer Science (CS) education, where traditional Optical Character Recognition (OCR) struggles with high error rates (30% in state-of-the-art systems) due to varied handwriting styles and inconsistent indentation, critical for languages like Python. This limits the ability to digitize and execute handwritten programs, hindering accessible CS education for students without computer access or in low-distraction learning environments.
Key Contributions	• The study introduces two novel benchmark datasets (Correct Student Dataset and Logical Error Dataset) for handwritten code recognition and a methodology to measure OCR

	accuracy and logical fix hallucinations. It proposes two indentation recognition methods (absolute and relative clustering) and a three-stage OCR pipeline (OCR, indentation recognition, LLM post-correction), reducing error rates from 30% to 5%. Additionally, it explores an end-to-end multimodal LLM (GPT-4V) approach, achieving a 6% error rate with minimal logical fixes.
Objectives/Goal	The objective is to develop an accurate handwritten code recognition system to enable CS education without computers, ensuring precise transcription of student code, including indentation, without introducing logical corrections that could undermine learning. The goal is to make CS education accessible and distraction-free, particularly for students with limited computer access.
Methodology/Theory	• The methodology comprises two approaches: a three-stage pipeline and an end-to-end multimodal method. The pipeline uses off-the-shelf OCR (e.g., Azure), followed by indentation recognition via absolute (Meanshift clustering) or relative (Gaussian Mixture Model-based delta clustering) methods, and LLM post-correction using Simple or Chain-of-Thought (CoT) prompting to fix typographical errors without logical changes. The multimodal approach uses GPT-4V to transcribe code directly from images with strict transcription prompts. Levenshtein distance measures OCR accuracy, and a Logical Fix Hallucination Test assesses unintended corrections.

Software Tools/Setup Details	The study employs Google Cloud Vision, Microsoft Azure OCR, AWS Textract, and MathPix for initial OCR, with Azure selected for its superior accuracy. GPT-4 and GPT-4-Vision-Preview are used for post-correction and multimodal transcription, respectively. Meanshift and Gaussian Mixture Model algorithms, implemented via standard libraries (not specified), handle indentation clustering. The codebase and prompts are available on GitHub.
Test/Experiment	The experiment evaluates OCR accuracy and logical fix hallucinations across 55 handwritten programs using normalized Levenshtein distance (OCR Error) and percentage of logical fixes in 11 programs. Four OCR systems, two indentation methods, and two post-correction strategies (Simple, CoT) are tested, alongside GPT-4V. A held-out set of 39 images validates indentation hyperparameters. Parameters include bounding box coordinates, delta values, and GMM hyperparameters (mean, standard deviation).
Test Data/Dataset Source	• Two datasets are used: the Correct Student Dataset (44 programs from 40 students in Stanford's Code In Place course, solving Python tasks) and the Logical Error Dataset (11 programs, mostly authored, with typical student errors like Fence Post, Arithmetic, and Control Flow). Both are open-sourced with annotations of intended digitalization and error descriptions.
Result Analysis	The three-stage pipeline with Azure OCR, relative indentation, and Simple prompting achieved the lowest OCR error (5.3±0.9%) but introduced 9% logical fixes. CoT prompting had a higher error (8.5±1.0%) but zero logical fixes. GPT-4V achieved a 6.0±0.8% error rate

	with 5% logical fixes, approaching the Pareto frontier. Relative indentation outperformed absolute, reducing errors to 20.2% before post-correction, validated on a held-out set.
Obstacles/Challenges	• Challenges include high initial OCR error rates (up to 64.6% for MathPix), inconsistent indentation in handwritten code, and LLM hallucinations introducing logical fixes. Handling crossed-out code and annotations (e.g., arrows) is difficult, and scalability to programs longer than 40 lines remains untested. Adaptive bandwidth estimation for indentation clustering and offline implementation are noted as future challenges.
Terminology/Keywords	Handwritten code recognition, Optical Character Recognition (OCR), indentation recognition, Large Language Models (LLMs), Levenshtein distance, logical fix hallucination, Python, CS education, Gaussian Mixture Model, Meanshift clustering.
Final Summary	The paper tackles handwritten code recognition for pen-and-paper CS education, addressing high OCR error rates and indentation challenges in Python by introducing two novel datasets and a three-stage pipeline (Azure OCR, relative indentation via GMM, LLM post-correction) that reduces errors from 30% to 5.3%. An end-to-end GPT-4V approach achieves 6% error with minimal logical fixes. Using tools like Azure OCR and GPT-4, the system ensures accurate transcription without altering student intent, validated on 55 student programs. Challenges include handling crossed-out code and scaling to longer programs, but the solution enhances CS accessibility for students with limited computer

	access.