Success Prediction in IT Studies based on Handwriting Samples

Submitted for the degree of

Bachelor of Sciences in Data Science

Astghik Kostanyan

Under the supervision of

Aleksandr Hayrapetyan
&
Suren Khachatryan

College of Science and Engineering
American University of Armenia

May 2024

Abstract

This research investigates methods of handwriting isolation, recognition, and classification to predict student performance in IT domain. Utilizing a dataset of 200 anonymized examination samples from Object-Oriented Programming and Data Structures courses at the American University of Armenia, the study developed a complex image processing and machine learning pipeline. The pipeline, which initially employed histogram matching that was subsequently replaced with advanced contour detection, effectively separates handwritten text from printed material. For handwriting recognition, the Google Cloud Vision API was employed, outperforming other OCR tools by smoothly handling diverse handwriting styles and extracting character-level confidence scores. These scores were essential in identifying patterns that correlate handwriting clarity with academic performance, revealing that higher confidence scores of character predictions correlate with better student performance. Differences in confidence scores between various characters also suggested that context, shape and student performance influence prediction accuracy. Progressing to handwriting classification, the research initially utilized a pre-trained CLIP model from OpenAI, which faced challenges with accuracy and model bias. This led to the development of a custom hybrid model combining a Convolutional Neural Network (CNN) with a Transformer, significantly improving classification accuracy to 78%. This model used image-based and text-based data, providing a robust tool for predicting student performance.

*The data and all implementations including algorithms, data preprocessing steps, and analysis scripts are available for academic use in GitHub.

Contents

Introduction

In modern days of educational technologies, the analysis of the academic handwritten material remains to be one of the main pillars for its educational impact and assessment. Handwriting is more than a way of communication; it carries a window into the cognitive processes and the academic performance of a student(McCarroll and Fletcher, 2017). With the fast advances in Optical Character Recognition (OCR) technologies, accurate extraction of academic materials became widely accessible, thereby opening huge opportunities and challenges for possible research.

Most work in this area has usually focused on the extraction of interpretation handwritten text—a crucial aspect for educational achievement evaluation. Few studies have, however, reported a relationship between legibility and quality of handwriting with academic performance in examination papers. Most traditional OCR technologies and methodologies are simply not able to cope with many challenges posed by handwriting, especially if they have to deal with diverse handwriting or low-quality scan styles.

The complexity of interpreting educational content, where text extraction accuracy is directly linked with reliability of analysis adds yet another layer of difficulty. Most current OCR systems, designed predominantly for well-structured, high-quality printed text, often fall short when applied to handwritten notes, especially those that include symbols, diagrams, Greek or Latin characters or variations in ink intensity and color.

Despite all these researches, there is a significant gap in the development of methods strong enough to distinguish accurately the printed from the handwritten text for student handwriting analysis. This paper, therefore, intends to fulfill the gap by developing an improved method that makes use of powerful image processing techniques with powerful machine learning models to enhance handwriting isolation, recognition and classification.

One of the main objectives of the study is to improve the process of extracting handwritten content from printed material with new methodologies by using simple image processing operations. The other objective is to obtain enhanced, accurate data with the use of modern

OCR handwriting recognition technologies. Moreover, the final motive is to develop a predictive model that classifies student performance on the basis of characteristics of handwriting. The research should shed new light on the relationship that might exist between clarity and consistency of handwriting and academic achievement—ideally, by providing a pipeline that can effectively process images, perform handwriting separation, recognition and classification.

The paper is organized as follows: after the Introduction, the Methodology section will outline the data collection process, the specific image processing techniques employed, and the details of the machine learning models employment. The Results section will then present a quantitative analysis of the performance of techniques and models, followed by a Discussion that interprets the findings and challenges. The paper concludes with a summary of the research contributions and suggestions for future research directions.

Literature Review

The literature in the area of handwriting recognition and analysis is extensive, covering a wide of methodologies and applications. The review in this paper will focus on the basic operations and methods in image processing, models in generative AI and optical character recognition (OCR) for text recognition and image classification.

1 Image Processing Techniques

The image processing techniques used in this study, primarily histogram matching, and ellipse major axis aligned bounding box algorithm, are well-defined in Burger and Burge's Digital Image Processing: An Algorithmic Introduction using Java(Burger and Burge, 2022). The book elaborates many of the underlying basic techniques related to digital image manipulation and analysis that range from point operation to morphological filters, contour shaping, and corner detection. These techniques are essential for effectively extracting valuable information from images. Histogram matching algorithm pseudocode and description, in this regard, have been of very great help in this research, since it facilitated the alignment of color intensity across different segments of an image, ensuring consistent quality. Additionally, the ellipse major axis aligned bounding box algorithm was very useful to the methodology and was chosen due to its excellent ability to isolate and define shapes properly in the image segments. Both techniques were very useful for handwriting isolation algorithm creation and development.

2 Optical Character Recognition Technologies

The field of Optical Character Recognition (OCR) has achieved significant transformations with the advent of deep learning with advanced neural networks, completely transforming text extraction from images. EasyOCR, a Python computer language Optical Character Recognition (OCR) model is especially tailored for data entry automation and image analysis. Unlike traditional OCR systems that primarily focus on printed text, EasyOCR employs a combination of Convolutional Neural Networks (CNNs) and character feature extraction techniques to improve the accuracy of recognizing handwritten characters (Sisodia and Rizvi, 2023). It works by segmenting the image into several boxes and then detecting characters in each box independently. Another powerful OCR tool, developed on existing Python package Tesseract, is PyTesseract. It is used for converting both the hard copy documents and digital images into an editable text format (Sisodia and Rizvi, 2023). PyTesseract is known for its power in the extraction of text against complex image backgrounds by the performing template matching and feature extraction. The library not only supports basic extraction of text but also includes functionality for the encryption and decryption of OCR-parsed files. Therefore, PyTesseract has great applicability in the fields of data extraction and information retrieval. Finally, the Google Cloud Vision API represents a cutting-edge solution in OCR technology, famous for being capable of covering a broad scope of handwriting styles. According to Google's documentation, the system uses deep learning models and provides a significant improvement in the accuracy and efficiency of text extraction from noisy environment such as handwritten papers. The Google Cloud Vision API excels in capturing details in handwritten texts, capturing a wide range of languages and script types. This capability is very important for applications requiring high levels of precision in text recognition and hence demonstrating the advancement of OCR technology with the use of deep learning.

3 Machine Learning Models for Handwriting Classification

The recent developments in Generative AI, particularly the introduction of models like CLIP (Contrastive Language–Image Pre-training) by OpenAI, have significantly expanded the scope of image-based applications. CLIP's design to understand and categorize images based on textual descriptions makes it highly effective for tasks requiring nuanced image classification. As detailed in the foundational paper by Radford et al., CLIP excels in learning image representations directly from raw text descriptions about images— a method proving to be scalable and efficient in matching current state-of-the-art capabilities (Hafner et al., 2021). This model has been trained on an extensive dataset of 400 million (image, text) pairs, allowing to transfer learned visual concepts for a broad set of downstream tasks without the need for task-specific training.

Additionally, the exploration of models that take the advantages from spatial pattern recognition with CNNs and the semantics of context from Transformers marks a promising area of research. These kind of hybrid models benefit from the integration of image and text data, enhancing accuracy and adaptability in complex classification tasks. For instance, CodeBERT, a bimodal pre-trained model developed by Microsoft, is tailored for both programming languages and natural languages (Feng et al., 2020). It was developed with a Transformer-based architecture and trained on a hybrid objective function attaining a relatively strong transferability and could serve various NL-PL applications like code search, documentation generation, and so forth. Such advances emphasize a very important trend towards more flexible, multimodal learning frameworks in machine learning.

Methodology

4 Data Collection

The foundational phase of this study involved an extensive collection of examination samples from Object-Oriented Programming and Data Structures courses at the American University of Armenia. Specifically, 200 exam samples were gathered, anonymized, classified and subsequently digitized from their original paper format to PDF files. These files were named and classified to reflect significant attributes such as the course name, exam type, exam date, and the performance classification of the student. For instance, the filename "OOP.MT2.240315.M110" specifically denotes the OOP course's midterm 2 exam dated March 24, 2015, taken by the 110th student whose performance is categorized as medium. The conversion of these PDF files to JPEG format was the first step facilitating easier manipulation and analysis in the next stages of the research.

5 Handwriting Isolation

The next step involved separation of handwritten content from printed text. An initial method chosen for this task involved histogram matching, a discrete mapping function that, when applied to the target image cumulative distribution function (CDF) of pixels, produced a new image with a distribution function similar to the reference image CDF (Burger and Burge, 2022). This technique was favored because of the fact that printed text, mostly in Times New Roman font and with denser ink distribution, would exhibit a higher average color intensity than handwritten text. The process entailed horizontally splitting the images into two halves, assuming that most handwritten content was located on the lower segment of the page.

Cumulative distribution functions were computed for each half, with input image being the handwritten half as shown in Figure 1. The obtained CDFs were then matched and the corresponding mapping was applied to the original image, aiming to get a uniform color intensity throughout the image The only step remaining was to subtract the original image from adjusted image, which would result in removal of printed text.

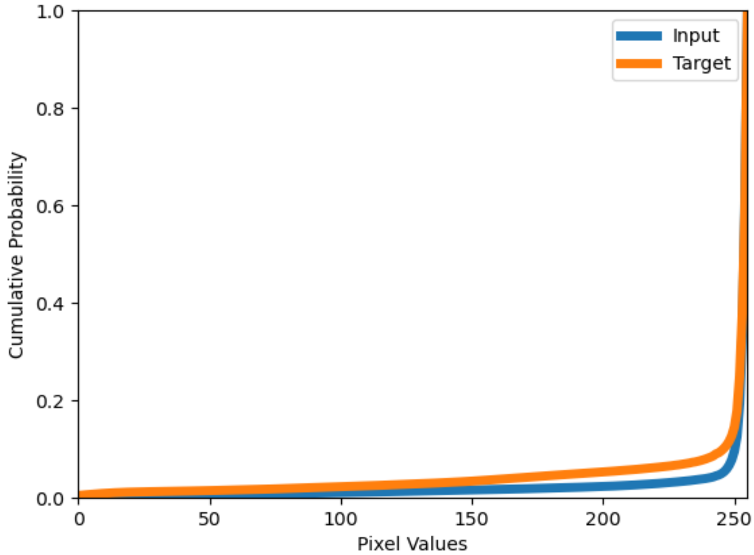


Figure 1: Histogram Matching

Moreover, another effective technique was adopted for handwriting isolation task involving contour detection and analysis. This method started with converting the image files to into binary images by applying Otsu's thresholding to clearly segment text from the white background. Once the images were binarized, a Gaussian filter was used to smooth the contours of the text and to reduce the noise that could interfere with contour detection. Contour detection process started by blurring the images, applying a second application of Otsu's thresholding to enhance the clarity of the text chunks, applying morphological operations, specifically closing and erosion, to improve the connectivity and visibility of text contours and making them more distinguishable. The next step involved adaptive thresholding to single out the largest contours, which were assumed to predominantly represent printed text, distinguishing them due to their greater size compared to typical handwriting. Ellipses were fitted into each large contour, and the largest ellipse major axis (diameter) helped to identify the printed text borders along the width of the image. Finally, a mask was generated from these identified large contours and dilated to include adjacent pixels, ensuring a comprehensive coverage of the printed sections. This mask recognized the lower most contour's coordinates which would be the lower boundary of printed text. The final step was cropping the original image using the abovementioned border coordinates to two parts and saving the handwritten part only for the next phases of research.

As the project progressed, refining the handwriting became imperative, particularly to address the challenge of removing highlighter marks from the images. These marks were initially applied to the annotation task to classify the samples as described in the Dataset Collection Section. However, their presence posed a significant risk of providing the classification models with direct indicators of their target outputs, thereby skewing the results. So, it was crucial to develop a method to remove these highlighted segments effectively. The process began with the conversion of color values from RGB (Red, Green, Blue) to HSV (Hue, Saturation, Value). The HSV color model is particularly beneficial for this type of work because it separates color (hue) from lightness (value), making it easier to isolate specific color ranges accurately. After converting to HSV, specific color range pixels corresponding to the highlighter colors in the data (yellow, blue and orange) were targeted and masked using a morphological 'close' operation, employing a large kernel. This step was crucial as it helped to bridge small gaps within the masks, resulting in unified areas or contours that were easy to identify and process. The image was then processed for contours within these masked regions. Each contour that was sufficiently large, indicating a significant area of highlighted text, was transformed to white, thereby successfully converting highlighted parts to white.

6 Handwriting Recognition

The next phase of the project involved exploring various Optical Character Recognition (OCR) technologies to identify the most effective tool for handwriting recognition. Early attempts with EasyOCR and PyTesseract revealed significant limitations in accurately detecting handwritten text from the dataset. These tools, although powerful for certain applications, proved insufficient for handling the nuanced variations in handwritten text.

Faced with these challenges, comprehensive research was conducted to find a more robust solution. This exploration led to the Google Cloud Vision API, renowned for its advanced image recognition capabilities and particularly effective in handling complex text recognition tasks. The decision to utilize Google Cloud Vision was based on its superior performance in preliminary tests, where it demonstrated a significant improvement in accuracy over the initially tried tools. Upon selecting Google Cloud Vision API, the setup involved authenticating API client that processed the images, which were enhanced with contextual hints to the API, specifying the likely languages contained within the text – English and Greek, to increase the detection accuracy. Additionally, for each character recognized by the API, a confidence score of prediction was obtained. To understand handwriting character patterns better, physical feature extraction of characters was done. As a result, groups of visually similar characters were created, shown in Table 1. This unique methodology of involving visually similar character groups would verify how uncertain the model is against the detection of visually similar characters.

Table 1: Table of Visually Similar Character Groups

Group	**Characters**
Group 1	(, c
Group 2	c, ¡
Group 3	c, e
Group 4	e, l
Group 5	e, o
Group 6	o, 0
Group 7	=, z
Group 8	z, r
Group 9	r, 2
Group 10	z, 2
Group 11	v, r
Group 12	r, n
Group 13	t, +
Group 14	y, g
Group 15	j, i
Group 16	i, ;
Group 17	j, ;
Group 18	{, (
Group 19	O, D

This comprehensive methodology for handwriting recognition using Google Cloud Vision API set the foundation for accurate and efficient extraction and interpretation of handwritten data from the collected examination scripts. The enhanced OCR capabilities facilitated by this approach were critical to achieving the objectives of the study, laying the groundwork for deeper insights into the correlation between handwriting characteristics and student performance levels.

7 Handwriting Classification

The final step in the research methodology focused on classifying student performance by their handwriting images. The main goal was to use all data available to fine-tune a pre-trained model such that, given a handwriting sample, the model would predict the student's performance. All models were trained on Google Collaboratory with a GPU to leverage accelerated computing.

7.1 Initial Approach with CLIP Model

The initial model used for classifying student handwriting was the CLIP (Contrastive Language–Image Pre-training) model (openai/clip-vit-large-patch14-336), developed by OpenAI. This model performs a variety of image-based classification and recognition tasks by learning visual concepts from natural language descriptions. The dataset was adjusted to meet the input requirements of the model: student performance levels were mapped to numerical values for model processing ('H': 2, 'M': 1, 'L': 0), and the images were resized to 336x336 pixels and converted into tensors. The model was trained over 15 epochs using the Adam optimizer with a learning rate of 5e-5 and a batch size of 4. The loss function used was Cross-Entropy Loss, which is suitable for multi-class classification tasks. To align image features with class labels, a linear projection layer with dimensions from 768 to 3 was integrated.

7.2 Patch-Based Approach for CLIP Model

The dataset's image resolution varied from 1700x200 to 1700x2200 pixels. As resizing the images to 336x336 would result in losing valuable handwriting feature information, another approach was adopted: decomposing the high-resolution images into 224x224 pixel patches. As the image sizes varied, it was crucial to address the remaining set of pixels after decomposition. A relatively trivial solution was upscaling the remaining set of pixels into 224x224 to meet the input requirement of the model. Each patch was labeled according to the label of the original image.

7.3 Development of a Hybrid Model

To avoid possible limitations by solely relying on image data and their classes, a hybrid model was developed to integrate both image and text data for more accurate student performance classification. This model combined the predictions of the handwriting recognition model from the previous stage with the classified image data, aiming to enhance the understanding and accuracy of the model. The hybrid model utilized a dual approach, combining a Convolutional Neural Network (CNN) for image feature extraction with a Transformer model for text analysis. Specifically, the CNN employed was ResNet50, pre-trained on ImageNet, providing robust feature extraction capabilities. The Transformer model used was Microsoft's CodeBERT, optimized for code-related textual data. The dataset comprised labeled images merged with corresponding textual descriptions of the handwriting images. The input images were resized to 224x224 pixels and underwent augmentations such as random rotation (up to

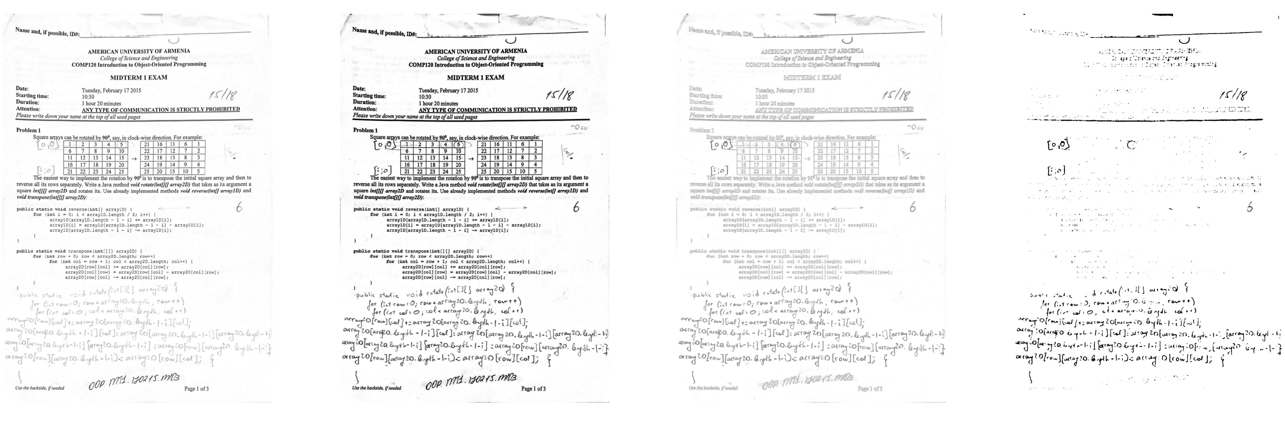
15 degrees) and color jittering to increase the model's robustness to variations in input data. The model was trained over 20 epochs using a batch size of 4. The Adam optimizer was used with a learning rate of 1e-5, chosen for its effectiveness in converging over fine-tuning tasks involving pre-trained models.

Results

8 Histogram Matching & Contour Detection

In this study, the primary objective was to isolate handwritten content from images, prioritizing clear handwriting extraction over completeness if it meant avoiding printed text inclusion. The effectiveness of each technique was therefore measured by the degree of clarity in the extracted handwritten segments.

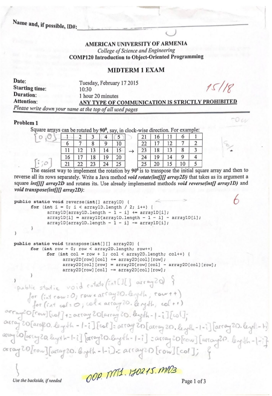
Histogram matching demonstrated initial promise by identifying the handwritten areas (Figure 2). However, the process often left the borders of the printed text visible, compromising the cleanliness of the extraction (Figure 2c). This issue was addressed by implementing binary thresholding and median filtering, which effectively erased the noisy residual borders, resulting in a clearer separation of the handwritten content, as shown in Figure 2d. Despite its efficacy in removing printed text from samples using light blue pen or pencil, this approach struggled with samples written in black or dark blue pens due to the close pixel intensity values between the handwriting and printed text, often producing nearly blank and noisy images.



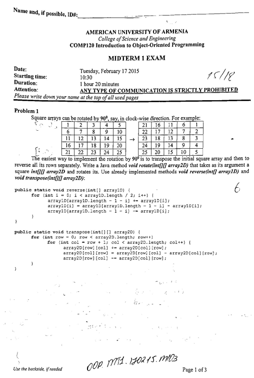
(a) Input Image (b) Matched Image (c) Subtracted Image (d) Output Image

Figure 2: Handwriting Isolation by Histogram Matching

Contour detection technique offered a more robust solution by focusing on the lowermost coordinates of printed text (Figure 3). By identifying the largest lowermost contour and cropping based on it (Figure 3f), the original image could be segmented effectively (Figure 3g).

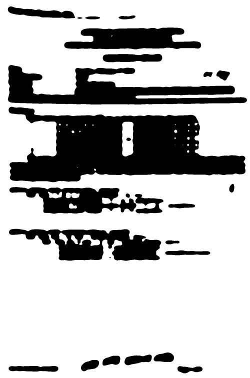


(a) Input

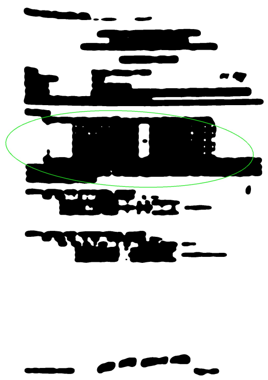


(b) Binary Threshold

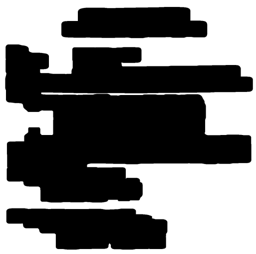
(c) Blurred



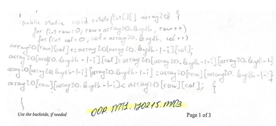
(d) Eroded



(e) Fitting ellipse



(f) Filled Contours



(g) Output

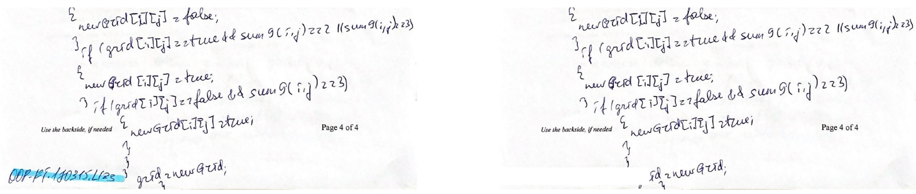
Figure 3: Handwriting Isolation by Contour Detection

Nevertheless, this method faced challenges with scratched handwriting or densely written styles, which could be misidentified as large contours and inadvertently cropped out (Figure 4). The resulting images often appeared as small, blank areas which were not useful for analysis. To maintain the integrity of the dataset, these smaller images were automatically removed in further processing steps. This selective exclusion ensured that mostly images with clear, visible handwritten content were analyzed, enhancing the overall quality of the data.

(a) Input (b) Filled Contours (c) Output

Figure 4: Failed Case of Contour Detection

The method to remove highlighter marks from images was very effective, as shown in Figure 5. The main challenge was removing only the blue highlighter marks without affecting the blue ink handwriting. To solve this, the 'closing' morphological operation was used to create contours around the marked areas. These contour areas were then compared to a threshold size. The areas of blue ink handwriting were always small, which helped distinguish them from the large, highlighted area. The method worked great for all images, it successfully removed the color distractions and prepared the images for detailed analysis, establishing optimal conditions for accurate handwriting recognition and classification.



(a) Before removal of highlighter (b) After removal of highlighter

Figure 5: Highlighter Removal

9 Handwriting Recognition with OCR

Handwriting recognition was tested using three OCR models. Initial trials with EasyOCR showed that it had trouble accurately recognizing handwritten text from the dataset, as shown in Table 2. PyTesseract was faster and more accurate at recognizing characters, but it still struggled with the unique variations in handwritten text, leading to many mistakes and unreliable data. Google Cloud Vision API demonstrated a significant improvement in accuracy over the initially tried models due to the model complexity and contextual understanding.

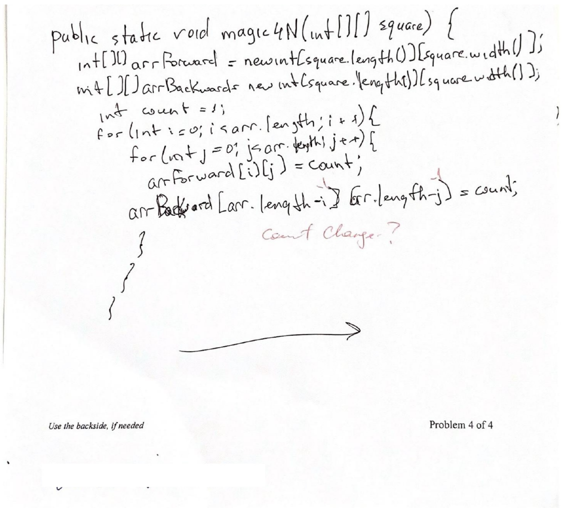


Figure 6: Input Image

Table 2: Detailed OCR Results Comparison

OCR Engine	Extracted Text
EasyOCR	”V oicl magic 4N(tlHI) 2quGre , arrRocnr 5 newintlsquare l +h 0Jlsuarc: m4[J JarcBack_ardt- 1&) twllsquare HkJ)lsg v() J; @uh F = J) For (Int 2 0, i Sar [ensth;i+4[foc (tjed i85-0jt?) [i)l;) = arRakari Lorr: lerq4-] 6r/yyf-} ¿ Csunl; Coccf Use the backside, if needed Problem 4 of 4 stac Publc widk J; In[Jl) (enq Ien u6re Int couh+ GnForwarc clarj¡”
PyTesseract	”public state votol mage4N (ut lf sq.ucce) f 1); atl JO arr ee ee newintlsquare.teng thO)Lequare.w ath! ss m&L JL) arrBackuacde ney ck Coquare Nevo Ht))E sq core w dh() J; coun =f} ie cor (Sart. [en sth, ne a foe bak) 22% feos HHS IL gor Pore ard rots) oe) ane Padee Looe lene Vhs) Gir Jength-j) = oank Cat itead i ee maar. Use the backside, if needed Problem 4 of 4”

Continued on next page

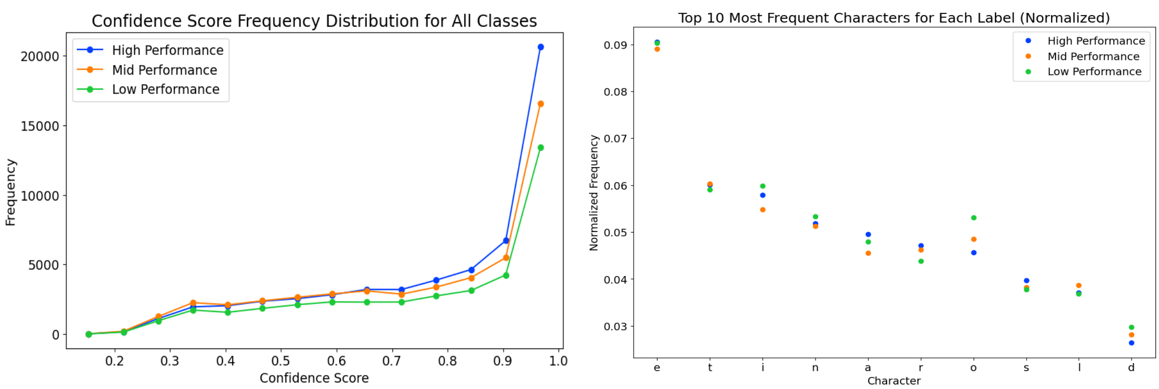
Table 2 – continued from previous page

OCR Engine	Extracted Text
Google Cloud Vision API	"public static void magic 4N (int[][] square) int[][] arr Forward = new int[square.length()][square.width()); mit [ ][ ] arr Backwards new int (square. lengthi)] [square. int count = 1; for (int i=0; is arr. length; i++); for (int j = 0; jsam. dength; j++) am Forward [i][j] = count; width()); arr Backward [arr.length-1] bar.length-j] = count; Count Change? Use the backside, if needed Problem 4 of 4"

The full predicted texts and character confidence scores by Google Cloud Vision API was used to find out how accurately characters were predicted in various contexts. The analysis led to the following interesting findings:

a. Confidence score frequencies across the three classes revealed that their distributions follow similar trend, as shown in Figure 8a. However, frequency gaps start to appear between the classes starting from 70% confidence score, with high-performance students' handwriting samples exhibiting the highest frequency of confidence scores, followed by medium and low performance student samples. The most significant rise in frequency occurs after reaching a 90% confidence level, where the frequencies become significantly different. This pattern indicates a clear correlation between student performance and the accuracy of character prediction.

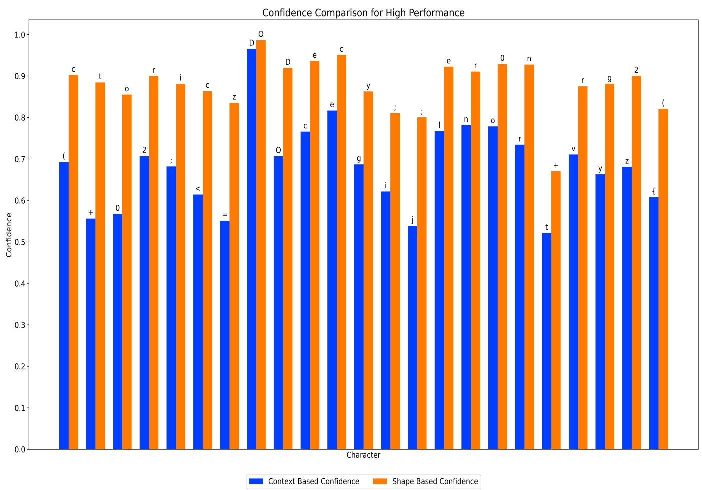
b. Across all classes, the letter 'e' emerged as the most common one, closely followed by 't' and 'i', as illustrated in Figure 8b.



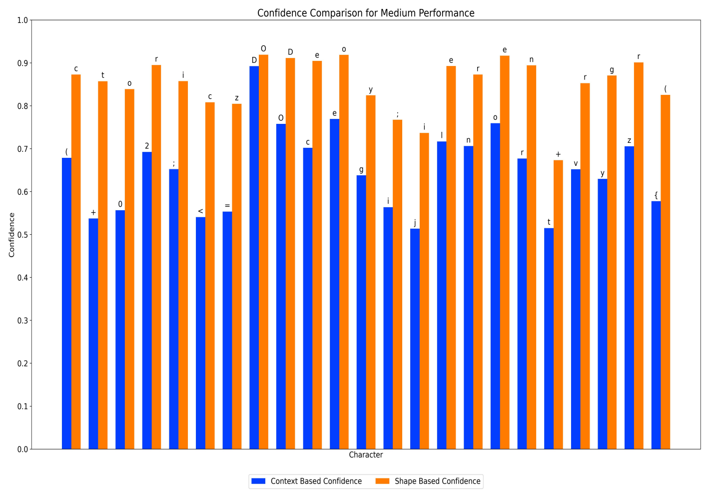
(a) Plot 1 (b) Plot 2

Figure 7

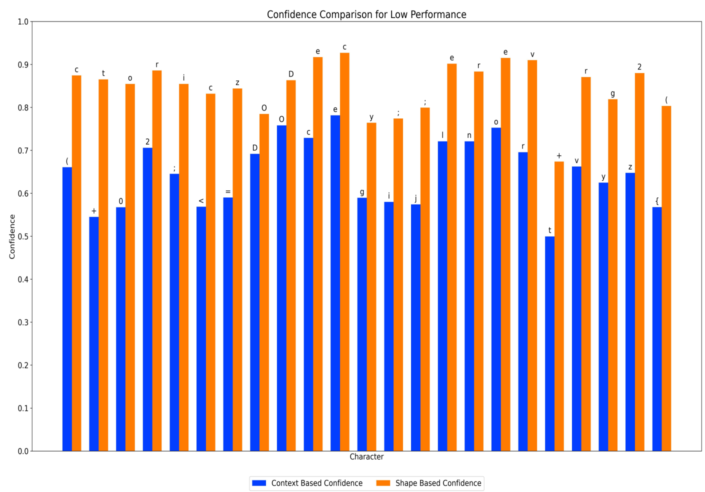
c. A significant discovery was the varying confidence scores between context-based and shape-based predictions, particularly with the characters ’D’ and ’O’. In high and medium performance student handwritings, both ’D’ and ’O’ had confidence scores of around 90%, as seen in Figure 8. However, scores dropped to about 70% for ’D’ and 80% for ’O’ among low performance students, suggesting that in the handwritings of high and medium performance students, the model is confident that the predicted character is either ’O’ or ’D’, but confused if it is ’O’ or ’D’. In case of low performance students, the model in uncertain about both ’O’ and ’D’, but is more confident in classifying ’O’.



(a) High Performance



(b) Medium Performance

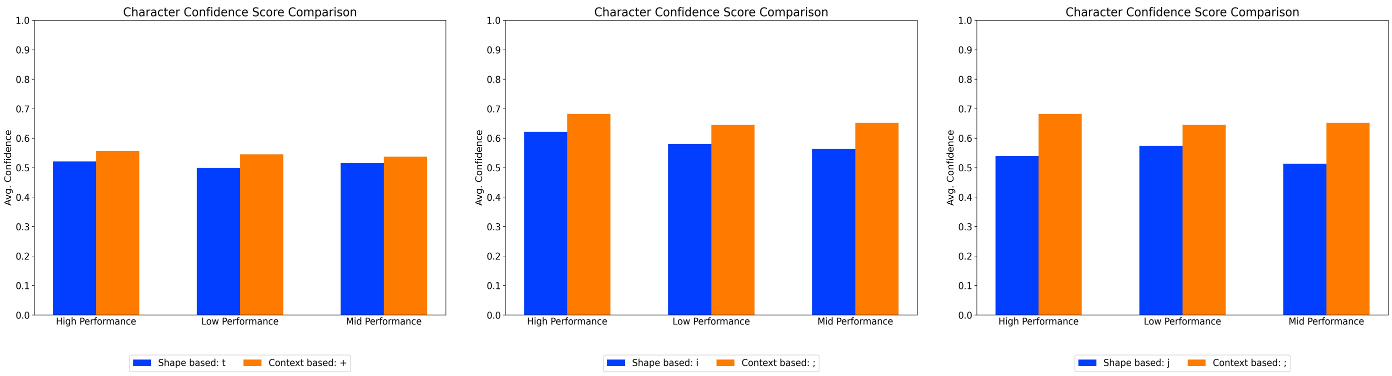


(c) Low Performance

Figure 8: Confidence Comparison across Categories

d. Interestingly, shape based predictions for ’l’ and ’o’ often resulted in a context based prediction of ’e’, suggesting potential confusion in these shapes. Moreover, ’2’, ’r’, and ’z’ all shared roughly 80% confidence, pointing to a high degree of uncertainty. Also, the pairs of ’v’ and ’r’, as well as ’z’ and ’r’, also showed model uncertainty.

e. The difference between the context-based '+' and shape-based 't' was about 30%, showing that the model was more prone to predicting 't' if context did not matter, as detailed in Figure 9a. The context based ';' and shape based 'i' pair revealed a 10% higher confidence for 'i', while for shape based 'j' the model showed a preference for ';', as shown in Figure 9b and 9c.



(a) t vs + (b) i vs ; (c) j vs ;

Figure 9

f. The 'O' and 'D' pair had the most unusual results as the predictions differed among classes, with higher confidence in 'D' for higher performing students and more confidence in 'O' for lower performing students, as depicted in Figure 10.

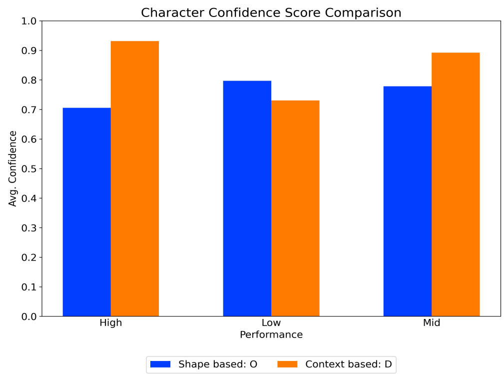


Figure 10: O vs D

10 Handwriting Classification with CLIP & Custom hybrid model

The performance of the two CLIP models was surprisingly similar, despite their significant differences in input data size. The larger CLIP model (336x336) achieved an accuracy of 27% on the test set, while the patch-based CLIP model (224x224) reached 38% accuracy. Both models tended to predict only the high-performance class according to their confusion matrices. This limitation could be attributed to the small size of the dataset or the challenging nature of the task.

In contrast, the custom hybrid model demonstrated greater adaptability to variations in input data. The training loss for this model decreased consistently over 20 epochs, indicating effective learning from the training data. Its performance was assessed using both accuracy and the confusion matrix, which detailed the distribution of predictions across the three performance levels (high, medium, low). The results were significantly better for the hybrid model, which achieved an accuracy of 78% on the test set, confirming its superior capability in classifying the handwriting samples more accurately across different student performance levels.

Table 3: Model Performance Comparison

Model Name	**Accuracy Score**
CLIP 336x336	27%
Patch-based CLIP	38%
Hybrid	78%

The results obtained from the custom hybrid model are comparable to those reported in a previous study, which also focused on the classification of student performance levels using handwriting analysis (Khachtryan, 2022). In that study, a Random Forest Classifier was deployed with an accuracy of 79%, achieved through the training on a reduced dataset size and a different set of features, including handwriting organization, indent quality, and readability. Despite the variance in input feature selection, model type and dataset scale, both studies underscore a consistent ability of machine learning models to accurately classify student performance levels.

Discussion & Future Work

This study addresses critical challenges in the field of educational technology, particularly the analysis of handwritten academic materials. The results indicate substantial progress in isolating and recognizing handwritten text from examination samples and classifying student performance based on those samples.

The histogram matching and contour detection techniques developed in this study have demonstrated their efficacy in isolating handwritten text from printed materials. However, they also highlight the intrinsic challenges associated with processing handwritten text, such as the handling of different ink colors and handwriting densities. These findings show the necessity for more refined research for these techniques that can handle the diverse characteristics of handwritten text.

The comparison among different OCR models showcased the superior capabilities of the Google Cloud Vision API in handling the complexity and variability of handwritten text, which is critical for ensuring the reliability of text extraction in educational assessments. This confirms the importance of selecting advanced OCR tools that can effectively manage diverse handwriting styles, a key component in the automated analysis of educational materials.

The innovative approach of using a hybrid model that combines image and text data has proven particularly effective, achieving a 78% accuracy rate in predicting student performance based on handwriting analysis. This suggests that increasing the dataset size and integrating multiple data modalities can enhance the predictive power of analytical models, providing a more accurate classification of student performance.

Future research could expand upon this work by exploring larger datasets, integrating additional linguistic and non-linguistic features, and applying these methodologies across different educational levels and disciplines. Additionally, further refinement and testing of the hybrid model would enhance its applicability and accuracy, making it a more powerful tool for educational assessments and interventions.

Conclusion & Acknowledgements

11 Conclusion

This research has effectively explored and demonstrated advanced methodologies for isolating, recognizing, and classifying handwritten text to predict student performance within IT-related educational contexts. Utilizing a robust dataset of 200 anonymized examination scripts from Object-Oriented Programming and Data Structures courses at the American University of Armenia, this study has made significant contributions to the field of educational technology, particularly in the automated analysis of student handwriting.

The development of a sophisticated image processing pipeline incorporating histogram matching and advanced contour detection has proved instrumental in distinguishing handwritten text from printed material. This has enabled more precise input for the OCR phase, enhancing the overall accuracy of text recognition. The use of Google Cloud Vision API for OCR demonstrated superior performance in handling diverse handwriting styles and extracting critical character-level confidence scores, which were key in identifying patterns that correlate handwriting clarity with academic performance.

Moreover, the initial application of OpenAI's pre-trained CLIP model highlighted some challenges with accuracy and model bias, leading to the development of a custom hybrid model. This model, combining the capabilities of Convolutional Neural Networks and Transformers, has shown remarkable effectiveness, achieving a 78% accuracy rate in classifying student performance. This success underscores the potential of integrating image-based and text-based data to enhance analytical precision and adaptability.

The findings from this study suggest that clearer and more consistent handwriting is associated with higher academic achievement, offering an innovative perspective on evaluating student performance through automated handwriting analysis. This insight could potentially guide future educational interventions aimed at improving handwriting as a facet of academic development.

In conclusion, this research not only advances our understanding of the relationship between handwriting and academic performance but also enhances the technological approaches used to study this linkage. By pushing the boundaries of what can be achieved with OCR and generative AI in educational settings, this study provides a foundation for future innovations that could transform educational assessments and student learning analytics.

12 Acknowledgements

The success of this research project owes much to the support and collaboration. I extend my gratitude to the Data Science department of the American University of Armenia (AUA) for the academic knowledge and learning opportunities that I have received. I am particularly thankful to my project supervisors, Aleksandr Hayrapetyan and Suren Khachatryan, for their expert guidance and assistance with development of interesting image processing techniques. Their expertise and practical insights have significantly enhanced the quality of the work and my professional growth. This research and its achievements would not have been possible without their support and encouragement.

Bibliography

Wilhelm Burger and Mark J Burge. *Digital image processing: An algorithmic introduction.* Springer Nature, 2022.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*, 2020.

Markus Hafner, Maria Katsantoni, Tino Köster, James Marks, Joyita Mukherjee, Dorothee Staiger, Jernej Ule, and Mihaela Zavolan. Clip and complementary methods. *Nature Reviews Methods Primers*, 1(1):1–23, 2021.

Naira Khachtryan. Using handwriting features to predict student performance in it and engineering domain. pages 1–10, 2022.

Hope McCarroll and Tina Fletcher. Does handwriting instruction have a place in the instructional day? the relationship between handwriting quality and academic success. *Cogent education*, 4(1):1386427, 2017.

Prakhar Sisodia and Syed Wajahat Abbas Rizvi. Optical character recognition development using python. *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, 4(3): 1–13, 2023.