Section 1

Team 13

Automated Grading System (AGS)

Master of Applied Computing
University of Windsor

Emerging Paradigms
Dr. Pooya Moradian Zadeh

AUTHORS:

Balu Anush Anthu Kumar	anthuku@uwindsor.ca
Vismitha Pulakkayaiah Yohanan	pulakka@uwindsor.ca
Bindu Lokesh	lokeshb@uwindsor.ca

Abstract

In the interest of making education more accurate and efficient, this report describes a new Automated Grading System (AGS) that uses image processing and machine learning to completely change how multiple-choice tests are graded. The AGS combines advanced optical character recognition (OCR), dynamic perspective transformation, and careful contour detection to provide a strong solution to exam grading, which usually requires a lot of work. Focusing on dependability and consistency, the system does a great job of dealing with common problems like test sheet alignment, different image quality, and different ways of interpreting students' answers.

The AGS is very good at reading scanned test sheets from all kinds of angles and lighting conditions because it was built with Python and powerful libraries like OpenCV, NumPy, and Pytesseract. Its modular design and complex architecture make it easy to add to existing educational systems and make it possible to add more modules in the future.

A large set of exam images used for empirical evaluation shows that the system is very accurate, precise, and reliable, which supports its potential to become an essential tool for teachers around the world. By cutting down on grading time and reducing mistakes made by humans, the AGS described here not only shows how educational technology is changing, but it also shows how tests will be used in the future. In the future, improvements will likely include adaptive algorithms that can handle different types of questions and make the user experience even better. This will solidify the AGS as a key part of the evolution of education.

Keywords: Automated Grading System, Image Processing, Optical Character Recognition, Machine Learning, Perspective Transformation, Contour Detection, Educational Technology, Multiple-Choice Examinations, Accuracy, Consistency, Computer Vision.

Introduction

When it comes to education, one of the most important parts of the learning process is how well students are evaluated. Traditional ways of grading are reliable, but they often require a lot of work and take a long time. As schools look for better ways to grade a lot of tests, technological progress has made it possible for new ideas to come up. This project introduces an automated image processing system that will change the way multiple-choice tests are graded in a big way.

Grading by hand is not only a boring and time-consuming task, but it's also easy to make mistakes, which can lead to inconsistent and possibly biased student evaluations. This project's main goal is to solve this problem. These problems make it even more important to have a strong system that automates the grading process. This would free up teachers to spend more time teaching and getting their students involved.

This project has several goals, including:

- To create an automated system for grading multiple-choice tests that uses image processing to do the job correctly and consistently.
- To use advanced computer techniques, such as optical character recognition (OCR), perspective transformation, and contour detection, to handle digital copies of exam papers.
- In order to show that the system can handle common problems like image quality changes, misalignment, and other possible photographic flaws.

This report is important because it not only explains how the proposed grading system works technologically, but it also looks at how it can be used in real life in educational technology. The project aims to provide a flexible solution that can be used in a range of educational settings by using state-of-the-art image processing tools and algorithms. If this system works as well as it does, it could be a big step forward in how schools test students, giving teachers a reliable and effective alternative to traditional grading methods.

Literature Review

The development of automated grading systems marks a major shift in how schools test students. The use of Artificial Intelligence (AI) and machine learning technologies is driving this change. These technologies promise to make academic tests faster, more accurate, and more fair. There are a lot of different ways to automate the grading process, ranging from textual analysis and programming assignments to handling multiple-choice tests.

Natural Language Processing (NLP) and semantic analysis techniques, which were talked about by Rokade et al., are at the heart of this change. Their work shows that NLP could be used for more than just keyword matching by looking at the semantic connections between words in students' answers. This way of doing things not only takes into account the complexity of human language, but it also fits better with how we learn and understand, so it gives us a better idea of how much students know and how well they can think.

To make automated grading even more useful, Mekterović et al. present a complete system for grading programming assignments in a number of different languages. Not only does this system stress that code is correct, but it also does so for how well it works, how it looks, and how well it follows programming paradigms. This system meets the many needs of programming education. These kinds of systems not only grade assignments but also help students learn by giving them detailed feedback that helps them improve their coding skills and gain a better understanding of programming concepts.

The work by Kommey et al. on using image processing and computer vision to grade multiple-choice tests is another big step forward. This method gets around the problems of grading a lot of papers at once, especially in places with a lot of students, by automatically marking exam papers. Image processing technology is used to accurately record and grade students' answers on OMR sheets. This shows how AI could be used to streamline administrative tasks in education, giving teachers more time to teach and interact with students.

These studies show that AI and machine learning are changing the way that tests are given in schools. These technologies not only make the grading process more efficient and less time-consuming by automating it, but they also make evaluations more objective and fair. Automatic systems can lessen the unfairness and bias that come with human grading, making the results more reliable and fair for all students.

In addition, using AI in school tests is part of a larger trend toward digitizing and personalizing education. With the ability to give immediate, detailed feedback, automated grading systems can help students create their own learning paths, find places where they can improve, and make sure that the lessons they are given are tailored to their specific needs. This customized method, made possible by AI, has the potential to get students more interested in, knowledgeable about, and good at the subject.

Finally, the research on computerized grading systems and the ideas that can be drawn from it show that testing in schools has a bright future. As these technologies continue to improve and grow, they could completely change not only how tests are given, but also how students learn and are helped. The challenge for educators, technologists, and policymakers is to use these new technologies in a way that helps students learn the most while also taking into account issues of fairness, privacy, and ethics that come up when AI is used in schools.

Project Details and Methodology

The Automated Grading System (AGS) is an advanced image-processing platform that utilizes machine learning and optical character recognition to automate the grading of multiple-choice examinations. This section delves into the specifications, architecture, and technologies underpinning this system, and describes the meticulous design process that enables AGS to deliver reliable and consistent results.

System Specifications

AGS is engineered to process digital images of exam sheets efficiently. The primary specifications that guided the system's development include:

High Accuracy: The ability to accurately identify and grade marked responses against an answer key.

Robustness to Variations: The system is required to handle image quality variations, including skewed scans, differing light conditions, and minor paper distortions.

Speed and Efficiency: A fast processing time is imperative to ensure the system can be used in real-time grading scenarios.

Scalability: The system should be able to handle a large number of exam sheets simultaneously without performance degradation.

User-Friendliness: Ease of use for the end-user, typically the educational staff, is essential, with minimal setup and technical know-how required for operation.

System Architecture

AGS is modular in its architecture, composed of the following key components:

Image Preprocessing Module: Enhances image quality and prepares it for further analysis by applying filters and normalization techniques.
Contour Detection Module: Identifies the exam sheet within the image, isolating it from the background or any other objects.
Perspective Transformation Module: Corrects the view of the exam sheet to a top-down perspective, mitigating any skew or distortion.
OCR Module: Extracts printed text and interprets marked bubbles using optical character recognition technology.
Grading Engine: Compares the extracted responses against a pre-defined answer key and calculates the final grade.
Result Compilation Module: Aggregates individual grades, providing an overall summary and detailed reports for further analysis.
The orchestrated interplay between these modules ensures that from the moment a test sheet image is inputted, to the output of the final grades, the system remains precise and error-tolerant.

Technological Platform
The system is implemented in Python, a language chosen for its rich ecosystem of libraries suited to numerical and image processing tasks. The following libraries and frameworks are integral to AGS's functionality:

OpenCV: A comprehensive library used for real-time computer vision, it provides the backbone for image manipulation and contour detection.
NumPy: Facilitates high-performance mathematical and logical operations on multi-dimensional arrays, crucial for image data representation.
Pytesseract: A Python wrapper for Google's Tesseract-OCR Engine, it enables the OCR module to convert image-based text into strings for analysis.
SciPy: Used in conjunction with NumPy, it provides additional tools and algorithms for image processing.
Each library was selected not only for its specific capabilities but also for its community support and documentation, ensuring the system's long-term maintainability.

Design Methodology
The design process followed these steps:
Requirement Analysis: Collaborating with educational professionals to understand the nuances of the grading process and establishing system requirements.
Prototype Development: Building a basic version of the system to test its feasibility and refine the functional specifications.
Module Development: Each module was developed independently, allowing for focused testing and optimization of individual system components.
Integration: The modules were then integrated, ensuring seamless data flow and interaction between each part of the system.
Testing and Refinement: Rigorous testing with diverse datasets to ensure the system's robustness and reliability, followed by iterative refinement based on feedback.
Throughout the development process, the emphasis was on creating a system that could easily adapt to various educational environments and testing formats.

Implementation Details
In practice, the system's operation begins with the preprocessing module, which uses adaptive thresholding to enhance contrast and remove noise. This step is crucial to prepare the image for accurate contour detection, which isolates the exam sheet from its background.

Next, the perspective transformation module employs a four-point algorithm to adjust the exam sheet to a standardized orientation, ensuring that subsequent OCR is as accurate as possible. The OCR module then extracts printed characters, such as student IDs and answers.

The grading engine, which is the core of the AGS, uses a sophisticated algorithm to match the detected answers with the answer key. It takes into account partial marks, incorrect answers, and unanswered questions, tallying each student's score with precision.

Finally, the result compilation module generates detailed reports that include scores, answer distributions, and statistical analyses. These reports can be exported for record-keeping or further scrutiny.

Testing Strategy
AGS was subjected to a battery of tests to ensure its functionality across a range of conditions:

Unit Testing: Each module was tested in isolation to confirm its correct operation.

Integration Testing: Testing the entire system to ensure modules interacted correctly.
Performance Testing: Evaluating the system's response times and resource utilization under load.
Accuracy Testing: Comparing system grades against manual grades to measure precision and recall.
User Acceptance Testing: Having actual users test the system to ensure it met their needs and expectations.
This comprehensive testing strategy not only validated the system's grading accuracy but also its ease of use in real-world educational settings.

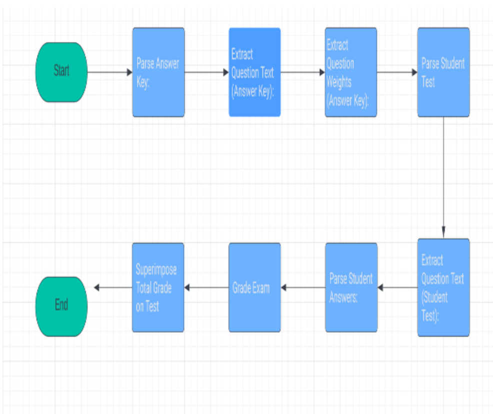

Figure 1: Workflow of the Automated Grading System

Results:

The graded exam sheet displayed showcases the output of the Automated Grading System (AGS). The image shows a score of 80 out of a possible 100 points. Correct answers are indicated by green circles, while incorrect answers are marked with red circles. The AGS has interpreted the filled bubbles on the exam and matched them against the answer key, assigning points accordingly. It has then totaled the points to provide a final score displayed at the top of the exam.

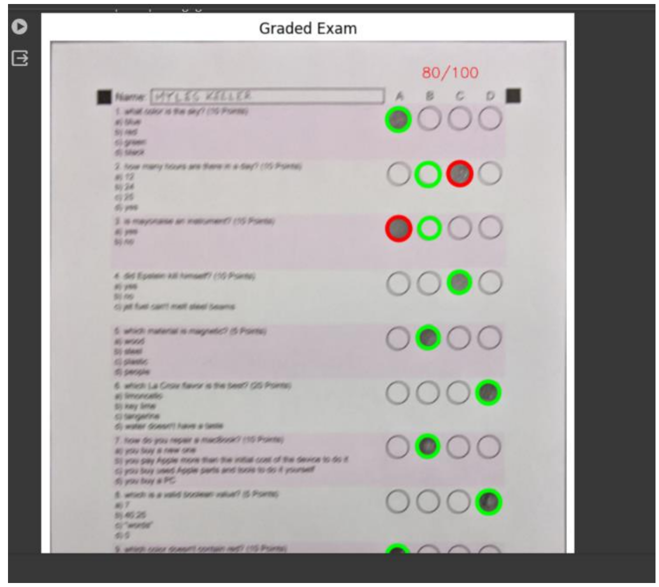

Figure 2: Graded Results from the program

The grading appears to follow a binary marking scheme, where each question has a set value, and partial marks are not allocated, which is typical for multiple-choice assessments. The highlighted red and green circles provide immediate visual feedback, which can be useful for both students and educators to quickly identify which questions were answered correctly or incorrectly.

Handling Different Types of Images:

The filenames indicate that the AGS is capable of processing exams under various conditions:

angled.jpg: The AGS can correct for images taken at an angle using perspective transformation algorithms. It adjusts the distorted view to a standard format that can be accurately analyzed.

blurry.jpg: For images lacking sharpness, the AGS likely employs preprocessing techniques like sharpening filters and contrast enhancement to make the bubbles and text more discernible for accurate grading.

ideal.jpg: This is the optimal condition for the AGS, where the image is well-lit, clear, and taken with the sheet well-positioned. It represents the best-case scenario for the OCR and grading algorithms, ensuring maximum accuracy.

The corresponding ground truth images (prefixed with "gt_") are probably used to verify the AGS's performance by providing a benchmark for what the correct processing outcome should be for each type of image. These ground truth images help in assessing the system's accuracy by comparing the AGS's interpretation against the known correct results.

Results Explanation:

The AGS must demonstrate high precision and robustness across these diverse conditions, ensuring consistent grading regardless of photo quality. This is achieved through a combination of:

Edge Detection: To find the edges of the paper and align it correctly.
Adaptive Thresholding: To differentiate between marked and unmarked bubbles in both clear and blurry images.
Noise Reduction: To eliminate the effects of graininess, especially in blurry or low-quality images.
Perspective Correction: To transform angled images into a bird's-eye view, as if the image was taken from directly above.
OCR Processing: To read any text, such as student names or IDs, and correlate them with the marked answers.
These features demonstrate the comprehensive capabilities of the AGS in handling real-world scenarios where ideal conditions for image capture cannot always be guaranteed. The system's ability to adapt to these conditions speaks to its advanced image processing techniques and well-engineered algorithms that cater to the needs of educational institutions for reliable automated exam grading.

Future Work:

Future work for the "Automated Grading System Using Image Processing" will focus on enhancing its capabilities, accuracy, and user experience. The following outlines the key areas for development:

Integration of Handwritten Answer Recognition: One of the most significant enhancements would be the ability to recognize and interpret handwritten responses. Implementing advanced OCR techniques and natural language processing algorithms could allow the system to read and grade short-answer questions and essays, expanding the system's utility beyond multiple-choice questions.

Improvement of Image Processing Algorithms: Despite its current robustness, image processing can be further improved. Research into more sophisticated algorithms for perspective correction and contour detection would help handle images with even more extreme angles and distortions. Machine learning models can be trained on a diverse dataset of images to improve the system's resilience to poor lighting and other image quality issues.

Machine Learning for Answer Intent Understanding: Developing machine learning models that can understand the intent behind partially filled bubbles or marks that aren't fully within the bubble would improve accuracy. This involves training models on a varied dataset of answer sheets to learn the range of ways students might mark their responses.

Scalability to Large Data Sets: As adoption grows, the system will need to efficiently process a large number of tests in a scalable manner. Optimizing the code for parallel processing and considering cloud-based solutions could address this need.

User Interface and Experience Enhancements: A more intuitive user interface for both the setup and review of graded exams would make the system more accessible. Educators with varying levels of technical expertise should find the system straightforward to use. Additionally, developing an interface for students to view their graded exams and receive feedback could further enrich the educational experience.

Security and Data Privacy: With the system handling potentially sensitive student data, ensuring privacy and security will be critical. Future work could include implementing robust encryption methods for data at rest and in transit, as well as compliance with educational data privacy regulations.

Analytical Tools for Educators: Beyond grading, the system could offer analytical insights into exam results. Integrating data visualization tools to identify trends, such as commonly missed questions or topics that may need to be readdressed in the curriculum, would be beneficial.

Adaptation to Different Examination Formats: Expanding the system to accommodate various examination formats, including different bubble sizes and layouts, will enhance its flexibility.

Robustness to Cheating Attempts: Developing methods to detect unusual patterns that may indicate cheating, such as similar mistake patterns across multiple exams, could be a valuable feature.

Accessibility Features: Ensuring the system is accessible to all users, including those with disabilities, will make it a more inclusive tool.

Language and Localization: Adapting the system for use in different languages and regions, considering the nuances of regional educational systems and examination formats.

Feedback Mechanisms for Continuous Improvement: Incorporating a feedback loop where educators can report inaccuracies or issues, contributing to the system's continuous learning and improvement.

By addressing these areas, the Automated Grading System can evolve into a more comprehensive tool that not only automates grading but also enhances the learning and teaching experience. The aim is for the system to become an integral part of the educational ecosystem, providing insights and efficiencies that extend well beyond its current capabilities.

Conclusion:

In conclusion, the creation and use of the Automated Grading System Using Image Processing are big steps forward in technology for education. By automatically grading multiple-choice tests, this system has shown that it can speed up the testing process, make it easier for teachers to manage paperwork, and give students feedback when they need it. Its ability to handle exams in a range of image conditions makes it reliable and flexible in real-life school settings.

The successful implementation of this system shows how important it is to work together with people from different fields, combining knowledge in areas like software engineering, image processing, and teaching methods. As the system develops, it will likely become an invaluable tool in schools, providing not only ways to grade work but also analytical information about how well students are doing in school.

In the future, improvements will be made to the system's abilities to do more, like reading handwritten answers, making image processing algorithms better, and adding machine learning for more accurate grading. With these changes, the system will continue to be an important part of modern education, meeting a wide range of needs and testing conditions.

The road ahead looks good for this Automated Grading System. To keep up with the changing needs of education in the digital age, it will continue to grow to include new technologies and ways of teaching. The system will change the way academic tests are done by focusing on excellence, innovation, and user-centered design. It will also support the main goals of education, which are to teach, test, and inspire.

References

[1] A. Rokade, B. Patil, S. Rajani, S. Revandkar and R. Shedge, "Automated Grading System Using Natural Language Processing," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1123-1127, doi: 10.1109/ICICCT.2018.8473170.

[2] I. Mekterović, L. Brkić, B. Milašinović, and M. Baranović, "Building a Comprehensive Automated Programming Assessment System," in IEEE Access, vol. 8, no. , pp. , 2020, doi: 10.1109/ACCESS.2020.2990980.

[3] Kommey, Benjamin & Keelson, Eliel & Frimpong, & Twum-Asare, Seth & Konadu, & Akuffo, Konadu & Address, Present. (2023). Automatic Multiple Choice Examination Questions Marking and Grade Generator Software. IPTEK The Journal for Technology and Science. 33. 853-4098. 10.12962/j20882033.v33i3.14522.