Optical Character Recognition based approach for automatic Image Marking Process

Aeman I. G. Masbah

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

UNIVERSITY OF HUDDERSFIELD

School of Computing and Engineering

Centre for Precision Technologies

SEPTEMBER- 2018

Copyright statement

Abstract

In today's world, programming teachers perform tedious tasks, which are time consuming; for instance, preparing and marking daily assignments, preparing and marking programming projects, preparing and marking short exams, etc. These tasks distract programming teachers from fulfilling their key role – teaching. Therefore, using automated marking approach with ability to communicate with students is highly desirable. Despite the existing approaches for automated student marking, there is still a need for more improvement. An automated program marking approach is proposed in this study based on a proposal by iMarking®. This approach automates the process of marking and assignments submission and facilitates the communication between teachers and students by designing and implementing a web-based application. In addition, the proposed approach adopts Optical Character Recognition (OCR) to extract the text from images to be evaluated using novel evaluation metrics. The novel evaluation metrics are formulated based on observation and experiment and aim to calculate the matching similarity and mismatching percentage of the submitted student answers when compared with the optimal answers. Evaluation results from a sample of 100 different programming questions show that the proposed approach is efficient in automatically marking the student answers with 100% accuracy. Furthermore, it is found to be time saving – approximately 197 seconds for marking ten questions – which is in line with the objective of creating a more efficient system for teachers.

Acknowledgement

The present research work could have not been completed if was not for the constructive guidelines, motivation and enthusiasm of my supervisory team. I would like to express my sincere gratitude to both my first supervisor **Professor Joan Lu** and my second supervisor **Dr Qiang Xu** for their outstanding guidance and invaluable feedback and support throughout my PhD.

I would like to thank every member of staff at the University of Huddersfield and particularly the school of computing and engineering for their professional approach since day one and providing the research atmosphere that every research would dream to have. PhD can be a stressful stage of student life, but always my friends and colleagues were there to help and support me whenever needed.

The family was a major part of my journey towards my dream of getting the highest achievement possible. I would like to thank my father and mother who supported me all way with their kindness, love and appreciation. My brother and sisters who were my backbone when I needed someone to support me and they were very happy to help support and motivate me during my study, from the bottom of my heart BIG THANK YOU. I would extend my thanks to my beautiful wife who shared with me all sad and happy moments in the UK, she was always my supporting me, taking care of everything I could not thank her enough.

Finally, I would like to thank the Libyan government for giving me the excellent opportunity to pursue my PhD degree and they have provided me with both academic and financial support.

Table of Contents

Abstract III

Acknowledgement IV

Table of Contents V

List of Figures VIII

List of Tables XI

List of Abbreviation XII

List of Publications XIII

Chapter 1 Introduction 14

1.1. Research Background 14

1.2. The Motivation of the Research 15

1.3. Research Problem 15

1.4. The Significance of the Study 17

1.5. Research Questions 17

1.6. Research Objective 18

1.7. Research Scope and Limitation 18

1.8. Research Contribution 18

1.9. Thesis Structure 19

Chapter 2 Literature Review 20

2.1. Introduction 20

2.2. Research background: 20

2.2.1. Automatic Marking System 20

2.2.2. Automated Grading and Assessment Systems 22

2.2.2.1. Generations of Assessment Systems 22

2.2.2.2. Automatic Grading of Programming Assignments (Criteria Characterisation) 25

2.2.3. The Critical Challenges Influencing Programming Courses. 26

2.2.4. Challenges to Embracing Programming Testing Rehearses in Assignments 28

2.2.5. Automatic Assessment: Advantages and Limitations 29

2.2.6. Optical Character Recognition 31
2.2.6.1. Techniques of Optical Character Recognition 36
2.2.6.2. Extracting text from text image using OCR 48
2.3. Related Works 51
2.3.1. The Approach Towards an Automated Marking 51
2.3.2. Automated Marking Systems Based on OCR 54
2.4. Critical Analysis 57
2.5. Summary 58
Chapter 3 Research Methodology 63
3.1. Introduction 63
3.2. Phases Description 64
3.2.1. Student Interface 65
3.2.2. Image Processing with OCR 71
3.2.2.1. Submission Process 71
3.2.2.2. Marking Process 73
3.2.3. Teacher Interface 74
3.2.3.1. Scenario One 78
3.2.3.2. Scenario Two 79
3.2.3.3. Scenario Three 80
3.3. Summary 81
Chapter 4 Design and Implementation 83
4.1. Introduction 83
4.2. Design and Implementation Description 83
4.2.1. General System Design 84
4.2.2. Back-End Design 85
4.2.3. Front-End Design 89
4.2.3.1: Benefits of Web Interface in the Design 90
4.3. Summary 92
Chapter 5 Results and Discussion 93
5.1. Overview 93

5.2. Results representation 93

5.3. Results discussion 112

5.4. Evaluation 117

5.5. Summary 126

Chapter 6 Conclusion and Future Works 128

Reference 131

Appendix 143

Appendix A: Results of the Percentage of Match and Mismatch 143

Appendix B: Questionnaire 160

List of Figures

Figure 2-1: Example of Robotised Architecture (source: Reek, 1989). 23

Figure 2-2: OCR Fonts (source: Chaudhuri et al., 2017). 32

Figure 2-3: Techniques of Optical Character Recognition. 37

Figure 2-4: Normalisation of Characters (source: Chaudhuri et al., 2017). 38

Figure 2-5: Noise Reduction Methods (source: Chaudhuri et al., 2017). 39

Figure 2-6: Normalisation Techniques (source: Chaudhuri et al., 2017). 40

Figure 2-7: The baseline extraction (source: Chaudhuri et al., 2017). 40

Figure 2-8: (a) Near vertical elements. (b) Average slant angle (Chaudhuri et al., 2017). 41

Figure 2-9: Segmentation Methods (source: Chaudhuri et al.,2017). 42

Figure 2-10: The Contour Direction (source: Chaudhuri et al., 2017). 44

Figure 2-11: The topological features (source: Chaudhuri et al., 2017). 44

Figure 2-12: Arabic character and its chain codes (source: Chaudhuri et al., 2017). 44

Figure 2-13: Matching Approaches (source: Chaudhuri et al., 2017). 46

Figure 2-14: Structural Methods (source: Chaudhuri et al., 2017). 46

Figure 2-15: (a) Deformations of a sample digit. (b) Deformed template on the target image with dissimilarity measures (Chaudhuri, et al., 2017). 48

Figure 2-16: Original Text Image and Binary Text Image (source: Jana, Chowdhury & Islam, 2014). 49

Figure 2-17: Line Extraction (source: Jana et al., 2014). 49

Figure 2-18: Input and Output of OCR (source: Jana et al., 2014). 50

Figure 2-19: Extracted Characters (source: Jana et al., 2014). 50

Figure 2-20: Extracted Charter (Source: Jana et al., 2014). 50

Figure 2-21: Naming convention for storing the scanned images (source: Bautista & Comendador, 2016). 55

Figure 2-22: Worksheets Evaluation Process (source: Uthman et al., 2012). 56

Figure 2-23: Worksheets Evaluation Process (source: Mangano, 2006)................................ 57

Figure 3-1: Main homepage. 67

Figure 3-2: Student login page. 68

Figure 3-3: Student homepage... 69

Figure 3-4: Student file upload dialog box. 70

Figure 3-5: Page with a dialog box for attaching a file. 70

Figure 3-6: Main homepage. 75

Figure 3-7: Teacher dashboard. 76

Figure 3-8: Send Question page. 76

Figure 3-9: View Solution page. 77

Figure 3-10: Dialog box for Uploading Original File (teacher answer). 77

Figure 3-11: Marking Assignment Page. 78

Figure 3-12: Report on the comparison between two similar images in Scenario One. 79

Figure 3-13: A report page showing a comparison between two different images. 80

Figure 3-14: A report page, the comparison between two images with similar lines. 80

Figure 3-15: Page for 'adding new student'. 81

Figure 4-1: System architecture (Masbah and Lu, 2018). 85

Figure 5-1: Age distribution of the respondents 94

Figure 5-2: Gender distribution of the respondents. 95

Figure 5-3: Satisfaction level with the Automatic Image Marking system. 96

Figure 5-4: Impact of automatic image marking system (iMarking®) on improving learners' abilities and skills. 96

Figure 5-5: Effectiveness of the automatic image marking system. 98

Figure 5-6: Consistency of automatic image marking system in giving accurate feedback. ... 98

Figure 5-7: Efficiency of automatic image marking system. 99

Figure 5-8: Significant matching similarity produces high marks. 100

Figure 5-9: Image quality determines the quality of recognition. 101

Figure 5-10: Significant features in essay transcriptions considered during image marking. 101

Figure 5-11: Images from students' submitted answers and the text extracted.....................102

Figure 5-12: Sample of student answers that consist of text and numbers.............................109

Figure 5-13: Sample of student answers that consist of text and numbers.............................109

Figure 5-14: Student final marks. 111

Figure 5-15: The original file. 119

Figure 5-16: The target file (submitted image T1). 119

Figure 5-17: Target file (submitted image T2). 119

Figure 5-18: The target file (submitted image T3). 120

Figure 5-19: The original file (optimal answer). 122

Figure 5-20: Target file (submitted answer). 122

List of Tables

Table 2-1: Comparison between Robotised and OCR systems (source: Reek,1989). 23

Table 2-2: Grading criteria characterisation for programming assignments. 25

Table 4-1: Hardware and Software Specifications. 87

Table 5-1: Descriptive Statistics. 95

Table 5-2: Gender Distribution. 97

Table 5-3: Pearson Chi-Square. 97

Table 5-4: Approval rating of the image marking system features. 99

Table 5-5: Calculation of matching similarity for category 1. 104

Table 5-6: Calculation of matching similarity for category two. 105

Table 5-7: An illustrative example on the evaluation metrics. 107

Table 5-8: Sample of the question and its associated student. 108

Table 5-9: The comparison between the proposed system and the manual method in terms of the processing times. 110

Table 5-10: Comparison of three main systems. 117

List of Abbreviation

A.E.S	Automatic Essay Scoring
AMS	Automated Marking system
ANNS	Artificial Neural Networks
ANNS	Artificial Neural Networks
ASSYST	Automatic Assessment System
BDT	behaviour-driven testing
CBA	Computer Based Assessment
ICR	Intelligent Character Recognition
IMARKing@	Automatic Image Marking Process System
JSP	Java Server Pages
LSA	Latent Semantic Analysis
MOOCS	Massive Open Online Course
MOSS	Measure Of Software Similarity
OCR	Optical Character Recognition
OHR	Optical Handwriting Recognition
OJS	Online Judge System
SPR	Structural Pattern Recognition
TDD	Test-Driven Development
WEB-CAT	Web-Based Centre For Automated testing

List of Publications

- Masbah, Aeman and Lu, Joan (2016) Automatic Image Marking Process. In: INFOCOMP 2016: The Sixth International Conference on Advanced Communications and Computation. IARIA: Valencia, Spain, 2016), pp. 7-11., pp. 7-11. ISBN 978-1-61208-478-7.
- Aeman I.G. Masbah, Joan Lu, Qiang Xu (2018): The Development of Automatic Image Marking System towards Better Assessment Monitoring for Computer Science Programming Students. Egyptian Computer Science Journal 42, 2, p. 44-54 11 p., 4.

Chapter 1 Introduction

1.1. Research Background

The development of an improved automatic image marking system combines theory and techniques from learning technology, structural representation and computing engineering. From a learning technology perspective, this research focuses on Optical Character Recognition (OCR). Specifically, it concentrates on the full development of the Automatic Image Marking Process System (iMarking®). The full life cycle of iMarking® incorporates the stages of development, running, testing and administering. Additionally, constructing a system of high quality is further focus of this work. Rangarajan et al. (2001) mention six major principles for any software testing: functionality, reliability, usability, efficiency, maintainability and portability. In this research, beyond the six merits discussed, software quality is also evaluated based on its fulfilment of the required specifications. Besides compliance with these performance measures, software quality further enhances the longevity of the product.

The notion of education management refers to organising and handling resources in such a way that the mission is accomplished within a well-defined range, value, time and cost boundaries (Laurillard, 2012). In this respect, handling students' academic tasks, such as examination records, plays a major role in the development and achievement of the goals of educational landscape. Accordingly, the implementation of technology to support instructors in their teaching and students in their learning enables both to perform better. Many researchers (Hetzel et al., 1991; Wilson, 1995; Jones, 2001; Wang et al., 2007; and Myers et al., 2011) have addressed the important role of information technology in an educational context. For example, numerous software and online applications have been developed to foster the educational principles and process. Therefore, information technology has played a major role as a promoter in this field.

In addressing the physical and theoretical transformation of education, the adoption of computer programming in teaching has improved impressively during the last decade. That is to say, computers are no longer used basically as techniques for precise tasks (Bull and McKenna, 2004). The spread of computer technology, along with the transformation of the educational environment, has raised interest in more helpful tools and automation of routine responsibilities (Valenti, Neri and Cucchiarelli, 2003). Computer Based Assessment (CBA) systems offer the educators a mechanism that can automate the assessment of a student's work. Charman and Elmes (1998) report on the four stages of a CBA system design. These are development, deployment, testing and administration.

However, the automatic image marking process for students' images is a significant topic at all academic levels. In the article titled "Analytical Assessment Rubrics to Facilitate Semi-Automated Essay Grading and Feedback Provision," Weinberger et al. (2011) concluded that the increasing number of educational assignments in one form or another raises the overall need for students' grading process. Hence, the amount of work and time spent by educators on assessment is increased, while the time available for feedback is reduced (Weinberger et al., 2011).

1.2. The Motivation of the Research

Technology in education has mostly been used to help access information and to make work easier in preparing notes or formulating templates for lesson plans. These strategies are beneficial especially because teaching is a profession that involves a large amount of visualisation work; however, more needs to be done. A teacher is required to have a copy of the syllabus and a lesson plan, to make notes, set examinations and assessment platforms and to mark the entire test. So much is required of one person, and he or she is still expected to deliver content in class and maintain a certain level of performance. Even though there are templates to help with lesson plans and curriculums, the templates must be customised to fit the needs of a class and the notes regularly updated. Thus, there is a need for an efficient approach that will reduce the paper workload on teachers. Thankfully, technologies such as OCR have been developed to reduce the headroom in testing, marking and grading. This forms the backbone of the present research.

1.3. Research Problem

Education policy-makers classify academic assignments and assessments as the most beneficial instrument to evaluate students' learning outcomes, implying their ability to memorise, form and integrate thoughts. Consequently, the importance of assessment feedback at initial level cannot be overlooked. For this reason, Carless et al. (2011) describe feedback as a key ingredient of quality student programming, which enhances the student's ability to improve. Feedback on an assignment allows students to revise their submissions to understand clearly where their program failed and, therefore, they can use that information to learn from their mistakes.

It should, however, be noted that one of the impeding factors of academic assignments is represented by the perceived objectivity of the marking procedure. From this perspective, many scholars have studied the critical elements influencing the reliability of academic assignments marking procedure (Sommerville 2007, Prados et al., 2011; Boud and Molloy 2012). These scholars reached the conclusion that, due to the objective nature of the human factor present within the academic marking procedure, there will always be inconsistencies in the grades given, which students can view as unfair.

One could advance a similar argument for students' stress derived from manual assessment marking. In a study carried out by Choy et al. (2008), on the impact of manual assessment marking on students' performance, it was observed that the development of an automated program marking tool for the marking process would provide feedback to orientate the students' learning and liberate creativity within teaching monitoring. However, from the results obtained by Choy et al. (2008), it can be safely concluded that an automated program marking tool provides an alternative support for securing the test environment (Choy et al., 2008).

It has also been identified that, during the initial coding process, the amount of programming carried out by the students was the main barrier to solving complex programming challenges in later years, as they did not gain enough experience. The majority of students enjoy programming; however, it can be frustrating for them when it comes to a complex situation where "standards" have to be maintained in terms of accuracy. OCR based approaches to automatic image marking are some of the most common. However, the existing OCR based approaches to marking are either not computer-focused or not fully automotive. In addition, some of the approaches adopt OCR simply to extract the hard-coded marks on assignment papers and they do not utilise it in the marking process as in Bautista and Comendador (2016). Moreover, these approaches do not use sophisticated evaluation metrics that give a comprehensive and detailed marking for the submitted assignments. To this end, automatic image marking tools have been developed for automating the assessment process; this makes them not only more consistent, but also more time and cost efficient when compared with a manual marking process.

1.4. The Significance of the Study

Lecturers and education decision-makers are always overburdened with workload. An automatic marking system can decrease this overhead and help lecturers to focus on other important issues of program tasks design. According to the study of Isong, (2001), the role of the automated marking system can deliver additional aid in terms of reliability, thoroughness and effectiveness. Such a system can also be highly beneficial to academics by providing a performance feedback to every student; this will no longer be a burdensome task.

To fill the gap in this field of research, the proposed design emerges from the proliferation of techniques, using Optimal Character Recognition (OCR) for extracting relevant coding features from the submitted image. This entire extracting image process has led to a content-based image retrieval system (Eakins, 1999; Niblack et al. 1993; Pentland, et al. 1993; Rui, Hang and Chang 1999). These systems have been changed and enormously facilitated to make it possible to index a large collection of images of coding data provided by the students.

Despite advances in the facilitation of OCR in such a manner, there are still many challenges to overcome in the development of iMarking® as a more user-oriented system. This is because the standard output of images provided by the students continues to consist of low features such as colour textures and quality. The software allows for a real-time examination platform; thus, by compiling and implementing the exam protocols set by the examiner, the system can automatically run the assessment and grade every student in line with predefined marking criteria.

The prototype has been designed based on a web interface that enables students to submit their assignments in the form of an image. These can then be marked automatically and receive feedback accordingly. It is hoped that, by sharing the web-based interface, a way of coding data can be constructed to examine a range of programming problems. However, there is a significant and practical pedagogical value in computer-based automatic assessment system (Charman and Elmas, 1998) for such tests.

1.5. Research Questions

1. How can the text be rapidly and effectively extracted from the submitted-images of student answers?

2. What are the effective evaluation metrics for the marking and grading of submitted student answers?

3. How can the robustness and effectiveness of the proposed approach be evaluated?

1.6. Research Objective

An OCR based approach is proposed for an automatic image marking process to reduce a teacher's workload, facilitate the process of assignment submission, enable communication with students and introduce automatic grading of the submitted assignments using novel evaluation metrics. The following are the objectives of this research:

1. To adopt OCR to extract the text from the submitted-images of student answers.
2. To propose a novel evaluation metric to provide effective marking and grading for submitted student answers.
3. To evaluate the proposed approach in term of the accuracy of marking and grading for submitted student answers.

1.7. Research Scope and Limitation

It is only a real-time examination system that provides an environment for the compilation and execution of a designed approach, which automatically marks the student's image as per the criteria set by the examiner.

The task of sorting the image, coding results and comparing them with prescribed answers in an accurate manner can be a difficult task. Image quality can also be a challenge for reading and scoring the contents and requires improvement if such automatic marking systems are to be deployed in large scale assessments.

1.8. Research Contribution

The main contribution of this research is an OCR based approach for an automatic image marking process. The additional contributions, which are directly related to the previously stated objectives, are summarised as follows:

1. Adoption of OCR to extract the text from the submitted-images of student answers.
2. A novel evaluation metric to provide effective marking and grading for submitted student answers.
3. An evaluation of the proposed approach in term of the accuracy of marking and grading for submitted student answers.

1.9. Thesis Structure

The research is divided into six chapters, references and appendices, which are the following:

Chapter One contains the introduction to the research: a general background, the motivation of the research, research problem, research questions, research objective, research scope & limitation and research contribution.

Chapter Two reviews the existing literature on automated tools.

Chapter Three discusses the research methodology, which includes description of the phases.

Chapter Four examines the design of the automated tools and how the design is implemented.

Chapter Five result and discussion.

Chapter Six conclusions and provides recommendations for future work.

Chapter 2 Literature Review

2.1. Introduction

The introductory chapter presented an overview of the research gap and associated issues, which will be considered in the present study. Chapter 2 provides a critical discussion and evaluation of previous studies on the state of the art of the Automated Marking system (AMS). The chapter is divided into sections, and the first, section 2.2, examines the automatic marking system, automated grading and assessment systems, generation of assessment systems, automatic grading of programming assignments, criteria characterisation, the critical challenges influencing programming courses, challenges to embracing programming testing rehearses in assignments and automatic assessment (advantages and limitations). After that, a comprehensive review of Optical Character Recognition (OCR), including the techniques of OCR and extracting text from a text image using OCR, is provided. In Section 2.3, the current approaches towards an automated marking and automated marking systems based on OCR are considered.

2.2. Research background:

2.2.1. Automatic Marking System

Automated testing in an educational context has been receiving broad attention in the computing engineering and educational context literature. Therefore, there are several definitions of the automated marking system concept owing to the recent advancements in software-based assessment tools. Student evaluation through testing has gained more attention, but it is most commonly defined as "a computer assisted assessment that allows instructors to quickly grade multiple and complex computer literacy assignments while providing meaningful feedback to students in order to stimulate an efficient learning process" (Conole & Warburton, 2005; Matthews, Janicki, He & Patterson, 2012). However, due to the complexity of these existing software products, a better understanding of the methods needed to develop them is imperative. To this end, a review of the automatic assessment tools used in assessing students' programs is provided, focusing on the benefits in terms of objectivity and consistency.

Srivastava (2002) considers software assessment as an independent member of software quality assurance, where the software specialists involved follow an iterative process of development, verification, evaluation and modification, in order to maximise delivery time and/or minimise the number of errors in the code.

With the fast development of PCs and data innovation, software engineering plays a critical part in the innovation training. The rudiments of software engineering are required in a few educational programs. Douce (2005) demonstrates that programming issues and assignments are basic components of program design and software engineering training, which is normally consolidated through basic examination. Programming assignments can enable understudies to become comfortable with the properties of present-day programming dialects, to gain an understanding of basic instruments, and to see how the standards of programming advancement and configuration can be connected.

The evaluation of these assignments places critical requests on teachers and different assets. Computerised devices and utilities can be used to rearrange the errands that both the educator and the understudy need to do, with the goal of surveying the goal consequently. Snyder (2004) suggests one approach, based on utilisation of test cases, to enhance the probability that a program does what it should do, both from the understudy's and the instructor's perspective. In this case, in the preparation of understudies for a test procedure, the strategy must be straightforward, starting with one program then moving on to the next. In spite of the fact that the programming model utilised has an input originating from the console or an info content record, there is another wellspring of information, which is implanted in the program itself.

Shepard (2001) demonstrates that less than half of undergraduate educational modules are concerned with testing issues. This rate, and the resulting knowledge, should increase by providing the understudies with a comprehension of:

- the expensive issues of testing,
- the correct spots for testing exercises in programming forms,
- how to plan and outline great test methodologies, and
- how to limit testing.

Edwards (2004) shows that many software engineering teachers have been searching for an approach to enhance the scope of programming testing aptitudes that students receive. Thus, the time spent on testing exercises will be decreased because of the better plans and testing rehearses that the students will have.

2.2.2. Automated Grading and Assessment Systems

Unfortunately, educators and instructors are usually overburdened with large workloads, and they have a very small amount of time available for training on software engineering courses or extra appraisal exercises. Subsequently, a mechanised instrument for evaluating understudy programs becomes an attractive solution. Numerous instructors have utilised computerised frameworks to survey and give fast input on extensive volumes of understudy programming assignments.

2.2.2.1. Generations of Assessment Systems

The current literature review examines the history of automated assessment as this may shape or influence the requirement of developing an automatic image marking process. In their historical overview, Douce et al. (2005) separate assessment tools into three generations.

i. Original Early Assessment Systems

The earliest cases of computerised testing of programming assignments are found in Hollingsworth (1960). Understudies submitted programs written in low level computing constructed on punched cards, as opposed to utilising compilers and content tools. A grader program is kept running against an understudy program and two unique outcomes are returned, either wrong answer and/or program finish. Additionally, the productive utilisation of registering assets enabled a more prominent number of understudies to pick up programming.

ii. Second Generation Tool-Oriented Systems

The second generation of appraisal frameworks was created utilising prior instrument sets and utilities provided with the working framework or programming condition. The testing motors and frameworks are frequently utilised and initiated as an order line or graphical user interface (GUI) programming apparatuses. A case of a moment age evaluation device can be found in Isaacson (1989). Command file is used as a Unix solution; this file is written using a C shell script. In such frameworks, programming assignments evaluation is performed using two exercises: 1) checking the program to see if it is working effectively, and 2) checking the program to see if the programming style has been connected sensibly. Reek (1989) presents the

TRY framework, as shown in Figure 2.1; the understudy, acquainted with robotised testing, is able to test projects utilising an analyser program. Table 2.1 shows the main differences between the robotised system and OCR system for marking students' assignments.

Table 2-1: Comparison between Robotised and OCR systems (source: Reek,1989).

Robotised System	OCR System
Students can run the task with the supervisors' test data.	No need to run, only scan the written task.
Provides a log of each student's attempt.	No need for student execution log.
Flexible with any programming language.	No need for installing or supporting programming languages.
Specific configurations are required to be installed on the students' account.	No specific configurations are required.
Requires more than one directory such as project directories.	No need for many directories.

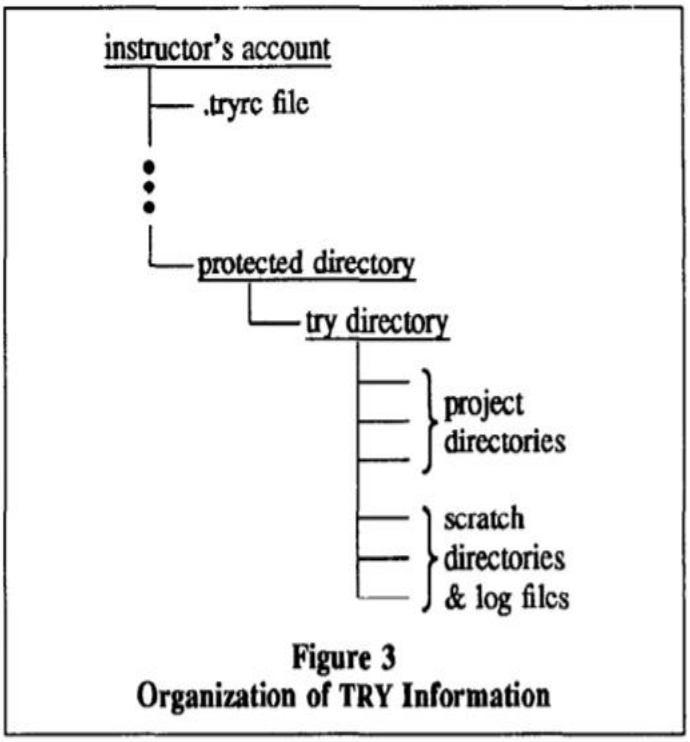


Figure 2-1: Example of Robotised Architecture (source: Reek, 1989).

At the point when the analyser program is executed, the understudy is given an arrangement of results and the test endeavour is recorded. Like other frameworks of the same period, testing is performed by using a straightforward character-by-character examination of results created against those expected. The Automatic Assessment System (ASSYST) framework (Jackson, 1997) presents a plan that investigates entries over various criteria. ASSYST dissects whether entries are right, regardless of whether they are proficient in their utilisation of CPU time and whether they have sensible metric scores that compare with many-sided quality and style. Third generation web-oriented systems (or third-age appraisal frameworks) make utilisation of improvements in web innovation and embrace, progressively, refined testing approaches. Some of these works will be exhibited in related work areas.

iii. Web-CAT: A Web-based Centre for Automated Testing

Edwards (2004) considers Web-Based Centre for Automated testing (Web-CAT), a progressed mechanised reviewing framework intended to process PC programming assignments, which was produced at Virginia Polytechnic Institute and State University. Web-CAT can review understudies as regards how well they test their own particular code and backings for all intents and purposes using any model of the program for evaluation, appraisal and criticism age. Web-CAT runs on a server and offers the majority of its capacities by means of a web interface. All of the activities such as accommodation movement, input, review of results and evaluating exercises happen through the web program.

The development of an automated marking system is inevitable and works based on a proposal by iMarking®. These systems provide markers, as well as instructors, to set up evaluation (assessments) and allow students to submit their initial tasks, including homework and assignment repeatedly in a predefined manner, with their source code to be run against the optimal results. However, these types of systems will terminate the task in case of an infinite recursion in the code and no hints will be provided to the student.

iv. Integration of Software Testing Throughout the Curriculum

Goldwasser (2002) includes software testing in the curriculum. This is done by integrating software testing throughout the curriculum, where the programming students are requested to submit both an execution and test set. Reviews are then conducted on both the legitimacy of an understudy's program on others' test sets and on how that understudy's test set performs in revealing defects in others' projects. The benefits of this approach are the following:

- Aggressive scoring gives a touch of inspiration to coursework.
- Understudies feel completely engaged in building up their own test sets.
- This offers a very different condition for programming testing.

- The scoring framework gives a quantitative assessment of both program legitimacy and test set quality, which can be incorporated as a major aspect of the general review.

Despite the previous advantages, this system suffers from the following shortcomings:

- The scoring process should fully be automated.
- It does not support an interactive environment for students and markers.
- It does not deal with the programs that include unexpected exceptions.

As it is a batch marking system, the student should wait to get his or her mark.

2.2.2.2. Automatic Grading of Programming Assignments (Criteria Characterisation)

Automatic grading of programming assignments plays a pivotal role in the enhancement of the students' programming capabilities, while making optimal use of the teaching staff's time. Similarly, Rodriguez et al. (2007) argue that having a robust grading model is a necessity for an efficient assessment process. Almost all institutions and teachers have their own assessing and grading criteria. However, if defining a common model for assessing and grading is not possible, an alternative solution would be to design and implement a flexible architecture that supports different ways of assessing and grading programming assignments (Rodriguez et al., 2007).

The study of Sommerville (2005) confirms the effectiveness of scheme architecture that aims to stimulate compliance with regulations and requirements of grading of programming assignments. For instance, Yelmo (2012) reports that maintainability is a quality attribute that can be determined using some criteria, one of which is complexity; this criterion can make use of some metrics such as the number of flow control structures used in a program. Table 2.2 below presents the grading criteria characterisation for programming assignments.

Table 2-2: Grading criteria characterisation for programming assignments.

QUALITY ATTRIBUTES		CRITERIA
Execution		**Compilation**
		Execution
Functional Testing	**Functionality**	**Correctness (system or method level)**

	Specific requirements	Specific requirement for an exercise
	Maintainability	Design
		Style
		Complexity
Non Functional Testing	**Efficiency**	Use of physical resources
		Execution time
		Processes load
		Code weight

2.2.3. The Critical Challenges Influencing Programming Courses.

Many researchers have studied the critical challenges of programming courses (Sheared, 2002; Ala-Mutka, 2004; Karen et al., 2012; Kay et al., 2012). Hence, a number of critical challenges, which have been identified in the literature as influencing programming courses, are discussed below:

- Building a mental model of the program is often a burdensome task for students. This is caused by structural differences when compared with natural languages. Although the students might demonstrate that they have learnt the fundamentals of programming languages, they may still fail to use their knowledge to create new programs. Therefore, “hands-on” practices are required for the students to strengthen their creativity with real-world programming exercises.

- Students usually perceive voluntary requests as unnecessary tasks with a complicated set of requirements. One possible solution would be to engage the students more in practical online-based exercises where they can submit their laboratory reports or answer weekly quizzes on assignment-related subject matters.

- Due to the complexities and sheer size of real-world software projects, it is not possible to incorporate the students in practical “hands-on” experience. However, they should still be able to learn the basic skill sets required for working in a professional environment. Therefore, it is imperative to introduce the students to simplified real-world programming problems, both in theory and practice, through assignments carefully planned by teachers. This means that the students are required to obey a number of basic rules. Nevertheless, it is not always possible to understand the effects of students neglecting such rules in practice.

- Beginner programmers often have a low set of standards of practice. For example, a program that functions just as desired might fail every other aspect of a correct programming framework. This is due to the complexity of the real-world requirements that makes the assessment of these simplified assignments a challenging task.

- Finally, cheating in programming assignments is another challenge to overcome. Since these assignments are marked in an electronic form, it is usually difficult to identify those who have copied from one another. For example, Sheared et al. (2002) raise concerns about the results of an aptitude test taken by their IT students, where 34% of the participants admitted to major cheating by copying from a friend and 53% completed the assignment by collaborating with one another when they were supposed to work individually.

Additionally, Ala-Mutka (2004) presents some common issues faced by understudies and instructors and audits existing evaluation rehearses for programming. The author finds that the understudies regularly face challenges in building mental models of PC programs, since this contrasts with the structure of a characteristic dialect. When the understudies have understood the programming ideas and dialects, they may in any case still lack the right skills to engage in the creation of new PC programs. Subsequently, if the understudies wish to figure out how to create PC programs, "hands-on" involvements with pragmatic programming must be practised. Understudies often deal with assignments against their will. A conceivable reason is that they see programming assignments as a particular errand with pointless and complex appraisal prerequisites.

One recommendation is to encourage the understudies to participate in regular (on the web) assessments on relatively easier segments of the course, and to instruct them either to submit their research facility assignments or answer a weekly test about the substance of the assignments. Certifiable applications and programming ventures are large meaning that they cannot be secured on one or even a few courses. Yet, at the same time, the understudies need to learn aptitudes for working under such circumstances. Along these lines, the complexities and practices of expert work must be presented partly in theory and partly by assignments that are disassociated from real-world frameworks. For educators, this implies that they have to design the assignments painstakingly. For understudies, this means that they are required to learn a few fundamental tenets, despite the fact that the impact of their negligence cannot generally be identified. Learner software engineers are generally not good at assessing their own work, as even off base projects can appear to have been completed as wanted, in light of the fact that the prerequisites of good and right programming rehearse make the assignments difficult to survey.

In addition, in the event that the understudies have not yet understood the issues of good programming skills, they cannot be surveyed fairly. Therefore, evaluation and input by a specialist are constantly required. The work needed to give good input creates substantial workloads for instructors. The issues of evaluating objectively, consistently and at reasonable speed are difficult metrics to deal with. The issues are clear to see on substantial courses, where a few mentors are required for the evaluation work. Another broadly perceived issue is cheating; since PC programs are in electronic form, they can be easily duplicated. Sheared (2002) investigates IT understudies' views about cheating at two colleges. Thirty-four percent of the respondents concede that they had replicated the majority of a task from a companion, while 53% had teamed up on a task that was intended to be completed separately. These figures should be considered with regard to understudies' learning.

2.2.4. Challenges to Embracing Programming Testing Rehearses in Assignments

Edwards (2004) presents five barriers to receiving programming tests in assignments:

- Programming testing requires involvement with basic programming, but the understudies are not prepared for it until they have acquired other essential aptitudes.

- Educators simply do not have sufficient energy (as far as time is concerned) to instruct another theme-like programming testing in an intense course.

- To gain from this action, understudies require a visit and advice on the best way to improve their answer, instead of only once towards the end of a task. The assets for fast, intensive criticism at various points of a program are simply not accessible in many courses.

- Understudies should seek any practices required for programming exercises. An understudy must perceive any additional work as support for completing projects, instead of a requirement imposed by an educator, who wishes for understudies to keep utilising a method reliably. By joining a reasonable testing system with the correct appraisal technique, and supporting them with the correct apparatuses, including a robotised evaluation motor, it is possible to overcome these obstacles.
- The evaluation of a programming task is an approach where the understudies are expected to test their own code in programming assignments, and, after that, survey them on this errand and the accuracy of their code arrangement. The approach must provide pragmatic advantages that understudies can see, but remain sufficiently basic to apply over the educational modules. Specifically, if understudies must test their own particular code, and afterwards be evaluated on both their code and their testing, how might one refrain from multiplying the reviewing workload of the workforce and showing associates while giving criticism? On expansive courses, giving input on a few programming assignments needs programmed help. The conspicuous advantages of utilising programmed appraisal apparatus to survey the understudy programs are the objectivity, consistency and speed of evaluation.

Likewise, the task of depictions and estimation of the criteria is painstakingly composed, since they must be customised to the robot, which will improve the nature of the task and make them more goal-oriented. This means that the understudies have the capacity to see precisely the desired outcomes, and when they are provided with unmistakably expressed targets and appraisal criteria, they can control their learning procedure and become more self-coordinated students. In the accompanying areas, some methodologies, programmed appraisal apparatuses and evaluation techniques have been reported in the literature.

2.2.5. Automatic Assessment: Advantages and Limitations

Automating the assessment process at all levels of Bloom's taxonomy (Bloom, 1964) can be a complex and time-consuming procedure. However, the assessment of knowledge can be automated, as it relies on a simple matching algorithm (e.g. multiple-choice equations). In contrast, the assessment of synthesis, analysis and evaluation skills is a more demanding procedure as they are domain specific (Bloom, 1964).

Brown et al. (2004) argue that the validity of the assessment is directly related to the experience of the assessment designer. By automating the assessment process, reliability and fairness are both enhanced as a result of using an identical marking procedure on every piece of work. Moreover, there is no possibility of discrimination as the students' assignments are all treated equally and owing to the formativeness of the automated process, they receive feedback on their assignments in a timely manner. Both students and educators are also compelled to respect deadlines (Brow et al., 2004).

Assessment automation is an incremental style of assessment. In their work, Benford et al. (1993) propose that weekly assessments in programming courses can increase the consistency of student learning. It is further argued that small and frequent coursework, in contrast to large pieces of work, allows the students to grasp better the taught material, especially in programming assignments. Moreover, by providing immediate feedback upon submission of a coursework and the ability to upload more than once, CBA motivates students to improve and avoid repeating the same mistakes in future assignments. Another advantage of the CBA systems is that, during the lifetime of a course, the educators are able to monitor the progress of the students and fine-tune their method or content of delivery, from both a theoretical and practical perspective. This helps to create a more tailored course (Bendford et al., 1993).

In comparison, Charman and Elmes (1998) argue that CBA cannot be applied to the assessment of all learning types, i.e. there is a range of skills that CBA is not able to address. For example, oral presentations and interpersonal skills are difficult to assess using current technologies. Furthermore, assessment of practical soft skills acquired during laboratory sessions requires CBA techniques with far greater complexity than what is currently available (Charman and Elmes, 1998).

A common misconception of CBA implementation is the amount of effort required to author exercises, which is a limiting factor in the large-scale uptake of CBA in the education system. This is in agreement with a claim made by Doube (2000), who emphasises that authoring learning material should not form an additional workload for the educators. Doube (2000) further states that the authoring of teaching and assessment material for online deployment is a comparatively more tedious task than traditional on-campus teaching.

The drive to automate the process of a learner's academic appraisal is primarily:

I. To ease the problems presented by large class size, and
II. To harness potential educational benefits.

It is worth noting that both of the aims listed above can easily be integrated. However, it is also pertinent to mention that advances in technique and tools for automation of a learner's academic evaluation process indicate that, in some cases, every level of Bloom's learning model can be assessed automatically.

2.2.6. Optical Character Recognition

One of the objectives of this study is to propose a system that will allow the submission of programming assignments in an image format, and through the use of OCR, to extract the text from the submitted assignments and save the text file. In the following, a collection of the basic ideas of OCR will be presented for a better understanding of the concept from a historical point of view.

Traditionally, OCR is defined as a process involving the classification of the optical patterns in a digital image with respect to alphanumeric and/or any other characters (Bunke and Wang, 1997). Francis and Jutamulia (1998) describe character recognition as consisting of three major steps of segmentation, feature extraction and classification. OCR technology is very useful in converting different types of documents (e.g. scanned paper documents, PDF files or images captured using digital cameras) into editable and searchable data. Over recent years, OCR's success as an enabling technology has gained much attention from both academic and industry experts of computing engineering (Bunke and Wang, 1997; Francis and Jutamulia, 1998).

OCR belongs to the family of machine recognition techniques that can perform automatic identification – a process where the recognition system identifies objects, automatically collects the data and feeds it directly into computer systems without any human interaction (Qadri and Asif, 2009). Note that any external data are captured and analysed through one of the forms of image, sound or video. The process of data capture relies on a transducer, which converts the actual image, sound or video into a digital file that can be stored for future analysis (Mithe et al., 2013; Chandarana and Kapadia, 2014). Figure 2.2 shows two examples of OCR fonts.

Figure 2-2: OCR Fonts (source: Chaudhuri et al., 2017).

2.2.6.1 OCR systems

In order to understand how OCR works and why it is incorporated in automated tools, it is necessary to explore the basic aspects of OCR, as well as provide insight into its background and history. OCR is a software which uses principles of optics to text characters in the organisation of words on scanning a page by making the contents of the paging machine readable. Once the content is machine readable, it becomes possible for it to be used in an electronic format, which can then be used for many other documents, as well as making functions to be mechanised.

OCR was initially limited to photo scanning of items and later translating it to codes. However, recently, it has grown to include a digitally generated image that may not involve scanning, but rather using stored items to generate images. Due to the development of OCR, its rate of efficiency and accuracy has been improved.

However, several concerns have been raised about how effective OCR is in photo scanning documents and presenting clear images and texts. As a result, several types of studies have been carried out and OCR has always been found to produce items with at least 99% resemblance to the real item.

According to Colaco and Borkar (2016), a recent study reveals that OCR software has a 99% accuracy rate. A similar study involving 45 pages of newspapers and articles published between 1803 and 1954 is reported by Breuel, Ul-Hasan, Al-Azawi, and Shafait (2013). The results of OCR accuracy across the news articles ranged from 71% to 98%. The recorded incorrect translation was 145 out of the 500 scanned items. Items, in this case, do not mean words but refer to general elements found within a text. The figure translates to 29% of public inconsistency in the translation of an item in all the 45 articles. Nevertheless, there is no established way of knowing the accuracy of OCR because there is no clear guidance whether the inconsistency should be assessed from the article level or page level. There is also debate on whether the items being assessed should be viewed from the character level or the word confidence level. Despite the lack of common ground for assessment, most scholars agree that modern OCR tools and software have an accuracy of around 99%.

Every invention has its pros and cons. The advantages of OCR are as follows:

1. It provides a platform for searching documents; this is done using the text-searchable platform;

2. It is time efficient when using digital files rather than physical paper documents;

3. It provides the means to change a document to other forms, for example, a written document can be turned into braille, and a Word document can be changed into a PDF document;

4. It can be used to improve one's business. OCR is helpful in improving customer service. For example, it makes it easier for clients to access whatever information they want from a directory then change it to suit their needs;

5. It is a major contributor to boosting work morale in staff. This is because most staff enjoy an environment where they can work easily and not be overburdened by extensive paperwork;

6. It increases the efficiency of office work. Most office tasks involve searching for information in a high-volume document. OCR makes it easier to search for the required documents; and

7. It is quick and accurate. Although some spelling errors may occur, the content is not interfered with and it works to generate the desired output.

The drawbacks of OCR are as follow:

1. The most common disadvantage is the lack of 100% accuracy with OCR. It is common for font size and type to be changed during the translation of items. There are also

recorded cases of the misspelling and emitting of letters that OCR considers unreadable. This leads to a text that loses the original meaning;

2. Work-around. This involves the addition of special features to help OCR in certain functions. For example, OCR has difficulties differentiating between zero and O. In such an event, the zero has to be written;

3. Additional work. Despite the help it provides, OCR, in its current state, is not able to perform some functions; it, therefore, requires additional work after the final output is presented. The work has to be proofread and any mistakes corrected. After OCR is used, the paper has to be read and the misspellings or any other mistakes corrected;

4. OCR machines and software are quite expensive; therefore, its access is limited to those with the financial power to acquire them;

5. The images produced by scanners in most cases consume much space; therefore, it requires an individual to ensure the systems and machines have a large storage space for the images produced; and

6. OCR does work well with handwritten text since the machine needs time to learn the handwriting. Although the developed version of OCR will be able to read them, it takes time, especially with poor handwritten texts.

Despite the many advantages of OCR, several other technologies are used to translate scanned objects. The alternative technologies focus mostly on the use of data science machines, which are aided by machine learning technologies (Springmann, Najock, Morgenroth, Schmid, Gotscharek, and Fink, 2014, May). Data science and machine learning technologies have been used in netting personal data details from driving licenses and passports. Furthermore, it has also been used in mobile receipt scanning.

Some proponents have argued that OCR operations can be done with technologies when it comes to accessing information online or digitally. However, it should be considered that the process involved in presenting the information is different; therefore, OCR cannot be compared with other technologies despite the real similarities. OCR focuses on photo scanning items and later on replicating the photo scanned items, while data science deals with accessing data that is already present (Springmann, Najock, Morgenroth, Schmid, Gotscharek, and Fink, 2014).

Data science deals with accessing information that is present in the digital space to produce a certain data, while OCR focuses on generating new items that are not present in the digital form. Moreover, OCR provides a platform for translating the scanned item into various forms; for example, an image can be converted into text. On the other hand, other technologies, such as machine learning technologies, cannot perform this function. They only allow access to the information as it does not alter its form or content (Breuel, Ul-Hasan, Al-Azawi, and Shafait, 2013). The ability of OCR to allow a change in the form of an item is what distinguishes it from the rest of the technologies and makes it unique.

Moreover, if OCR's functionality was limited to scanning documents and then simply changing the form, then, one could argue, why not simply scan the documents in normal format and then use scanners to translate and save them as JPEG and TIFF, among other formats? However, there is more to OCR than simply the scanning of items. It provides a platform for complex processes to be undertaken. It allows for many operations to be carried out in one space. A text can be converted to an image in one process without involving several professionals or software; the same platform can then change the image to various forms such as braille for a wider audience (Breuel, Ul-Hasan, Al-Azawi, and Shafait, 2013). It is only OCR that allows for a complex process to be carried out to a single item, within a short time and in a simple way.

On the other hand, OCR is not perfect as several errors are usually made. For instance, OCR's ability to format characters is limited to font size (Breuel, Ul-Hasan, Al-Azawi, and Shafait, 2013). Characters with a font size of 12 and below are more likely to have poor results compared with characters with larger fonts. Moreover, the output of the translated items will always be in a single column and full of spelling errors; therefore, the work must be edited and proofread even after it has been deployed to ensure the output is correct and in the desired style (Breuel, Ul-Hasan, Al-Azawi, and Shafait, 2013). The quality of output is limited to factors such as quality of the picture, font size and unusual script font sizes. In the end, the format of the original document is lost and the output has to be formatted to achieve the desired result.

OCR works by undergoing several processes to make sense of what has been presented to it. During scanning, the documents are simply seen as one object, and any material in it cannot be deciphered unless broken down into several characters that OCR software and machines can understand. According to Springmann et al. (2014), the first step of OCR is pattern recognition, because different fonts have different patterns leading to different handwritings. It is, therefore, necessary to recognise the pattern before other functions are performed. Afterwards, Intelligent Character Recognition (ICR) is used to enable identification of individual characters, no matter the style of writing; for example, after OCR (Springmann et al., 2014). The individual characters are then translated according to how they have been deciphered.

In short, OCR has greatly improved in terms of functionality and speed. It has developed from use in documentation to being incorporated in features such as mobile phones and other software that allows face recognition and identity recognition through IDs and fingerprints. Initially, OCR was very slow, but it has now become faster and almost instantaneous in its translation (Colaco and Borkar, 2016). There is no need for more developments in OCR so that its effectiveness is not limited by any factor. It should be developed to translate items of different nature, size and clarity of quality output.

2.2.6.1. Techniques of Optical Character Recognition

Automatic pattern recognition revolves around the idea of teaching the machine which class of patterns to expect most and what they would look like (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). In the case of OCR, these patterns appear in the form of letters, numbers and certain special characters, e.g. commas and question marks. In order to teach the machine these patterns, a large number of examples, containing characters from a variety of different classes, is required (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). Using these examples, the machine creates a framework for each class of characters, where unknown characters are compared with those previously recognised and the identified characters assigned to a class that best describes them (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

Digitising the analogue documents by the mean of an optical scanner is the first step in automatic pattern recognition. The segmentation process acts to extract every character when regions that contain text are located. Then, the characters are pre-processed in order to remove noise and, therefore, enhance the extraction features (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). Each character is then identified by comparing the extracted features with pre-defined symbol classes generated in an earlier learning phase. Finally, contextual information is used to reform the words and numbers in the original text. Figure 2.3 illustrates the process steps.

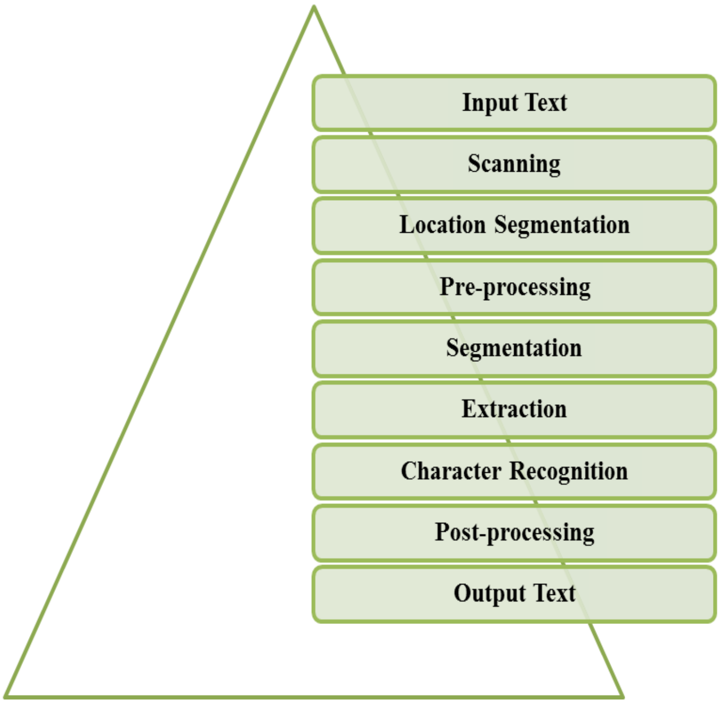


Figure 2-3: Techniques of Optical Character Recognition.

(a) Optical Scanning

Optical scanning is the first component of OCR. This is simply a practice in which the original document is converted into a digital image format. Optical scanners, which consist of a transport mechanism and sensing device, and convert light intensity into grey levels, are normally used in OCR. Printed documents are comprised of black print on white background. During extraction in (optical scanning) OCR procedures, a process simply referred to as, thresholding, is performed on the scanner to save memory space and computational effort (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). During this process of thresholding, a multilevel image is transformed into a two-level black and white image.

(b) Location Segmentation

After scanning, the next component of OCR is known as location segmentation. This module decodes the elements of an image.

The regions of the document covered in printed data are located and isolated from figures and graphics (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). For example, before the recognition process starts automatic mail sorting through envelopes, addresses have to be located and then separated from other prints including both company stamps and logo. For text recognition, separation includes isolation of each character or word.

Most OCR algorithms work by slicing words into isolated characters, which are then recognised separately (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

(c) Pre-Processing

Pre-processing is the third function of OCR, which involves subjecting the acquired raw data to a number of processing stages before making it suitable for character analysis. It is possible for the images attained from the scanning process to be contaminated with some noise in the form of smeared or broken images, which largely depends on the resolution of the scanner and the inherent thresholding. Through the pre-processing stage, some of the few defects, which might cause poor recognition rates, are eliminated, by smoothing digitised characters. Smoothing in this case refers to both thinning and filling. While filling repairs small breaks, gaps and holes in digitised characters, thinning supports the reduction of the width of line components (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

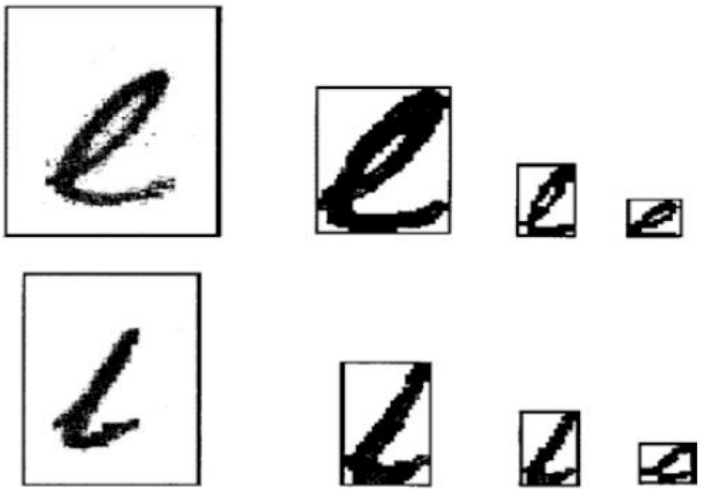


Figure 2-4: Normalisation of Characters (source: Chaudhuri et al., 2017).

(A) Noise Reduction: The scanning device or writing device originates noise. Line segments that are disconnected, line bumps, gaps, loops that are filled, etc. are caused by the originated noise. Therefore, a possible problem of distortion, which includes local variations is expected to arise in addition to potential problems of rounding of corners, dilation and erosion. These imperfections should be addressed and removed before the real data processing. Methods of noise reduction are classified into three main classes as shown below in Figure 2.5.

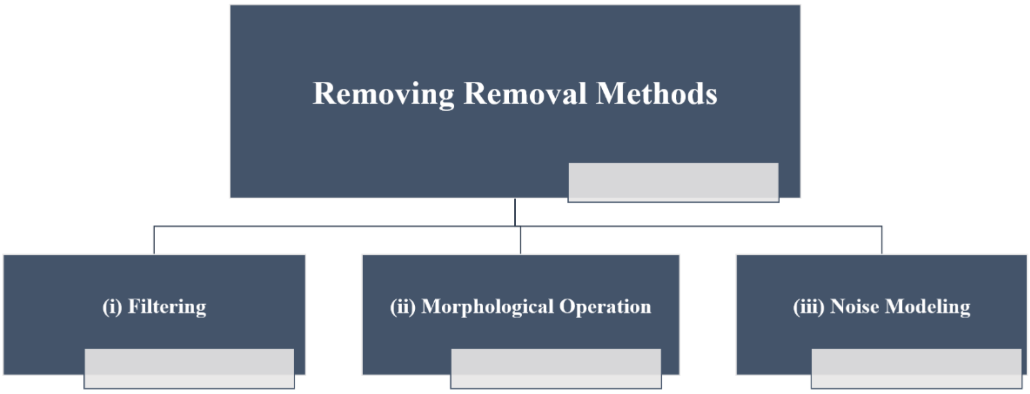


Figure 2-5: Noise Reduction Methods (source: Chaudhuri et al., 2017).

(i) *The filtering method* works on removing noise and lessening imperfect points. These points are often originated by a writing surface and an imperfect sampling ratio of the data gathering device. Varied spatial and frequency domain filters are purposively implemented. They aim to convolute a mask that is predefined with an image. Filters are modelled for smoothing and thresholding purposes, to some extent eliminating background and contrasting adjustment aspects (Chaudhuri, 2010; Arica & Yarman-Vural, 2001).

(ii) *The method of morphological operations* basically aims to filter each character by replacing the convolution by a logical operation. Varied morphological techniques are proposed aiming at connecting the broken strokes, decomposing the connected strokes, smoothing the contours, reducing the wild points and extracting the boundaries (Chaudhuri, 2010). This technique can be effective in removing noise from character images because of paper and ink imperfect condition, as well as the hand movement.

(iii) *Calibration techniques* are generally capable of removing noise. This can be done if it is possible to model noise. However, many applications cannot achieve noise modelling. Based on the existing literature, noise modelling is originated from optical distortion, i.e., speckle, skew and blur. In addition, assessing the character images' quality and decreasing the noise to some extent can be carried out (Chaudhuri, 2010; Chaudhuri et al., 2017).

(B) Normalisation techniques aim to remove writing variations. Common normalisation techniques are summarised in Figure 2.6.

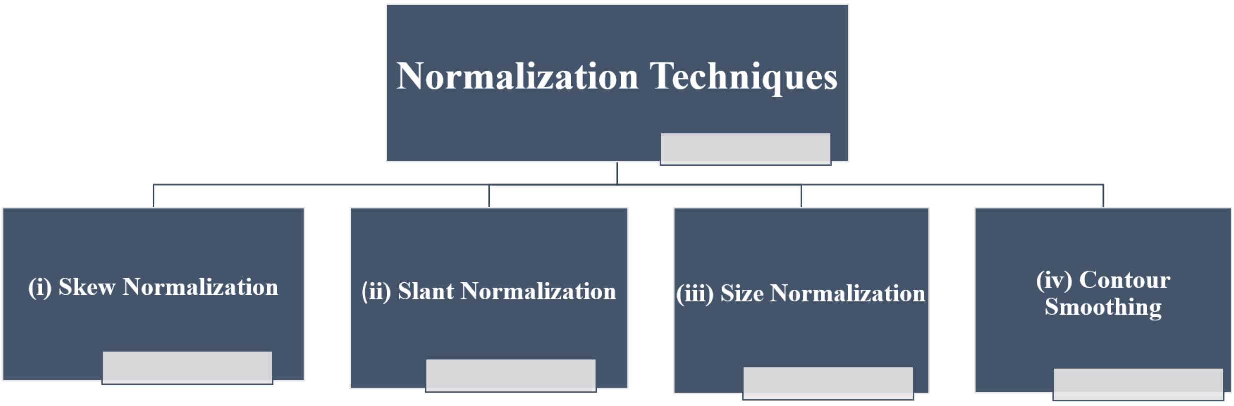


Figure 2-6: Normalisation Techniques (source: Chaudhuri et al., 2017).

(i) *Skew Normalisation*: The process of scanning and the writing style have imperfections. Therefore, within the image, writing is curved a little. As a result, the algorithms' efficiency can be affected. Hence, it should be fixed. Moreover, some characters can be recognised in conformity with the relative position regarding the baseline such as the number (9) and the character (g). To extract the baseline form image with heavy noise, a repulsive nearest neighbour is used (Reshma, James, Kavya & Saravanan, 2016) as shown in Figure 2.7.

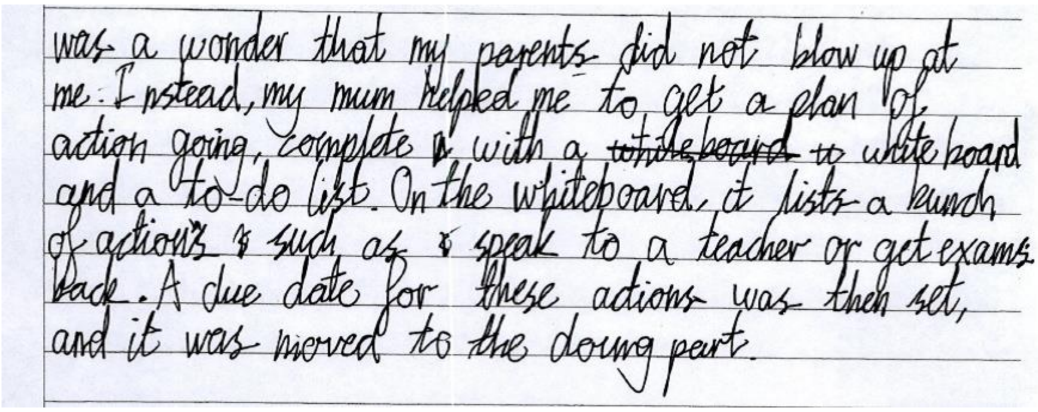


Figure 2-7: The baseline extraction (source: Chaudhuri et al., 2017).

(ii) *Slant Normalisation:* Several factors can be measured pertaining to various handwriting styles. One of these factors is the slant angle between the vertical direction and the largest stroke. This type of normalisation works on normalising all characters into a normal form. Slant estimation technique, which is commonly implemented, is carried out by calculating the near vertical elements' average angle as shown in Figure 2.8 (a). Tracing chain code components through single dimensional filters extracts the vertical line elements from contours. The slant angle is provided by the beginning and ending

coordinates' points of every line element. Predicted profiles are calculated for quite a few angles away from the vertical direction. The angle (corresponding to the top positive derivative projection) is used to detect the smallest overlap amount between vertical strokes and the leading slant angle. By segmenting the image into vertical and horizontal windows, slant detection is carried out (Reshma et al., 2016; Chaudhuri et al., 2017).

(iii) *Size Normalisation:* Size normalisation aims at adjusting the size of the character to some standard. The character is categorised into zones, and each zone is distinctly scaled. In addition, this type of normalisation can be carried out as part of the training phase and size variables are approximately calculated in a separate manner for each training data (Reshma et al., 2016; Chaudhuri et al., 2017). Based on Figure 2.8 (b), characters are steadily contracted to an optimal size maximising the recognition rate.

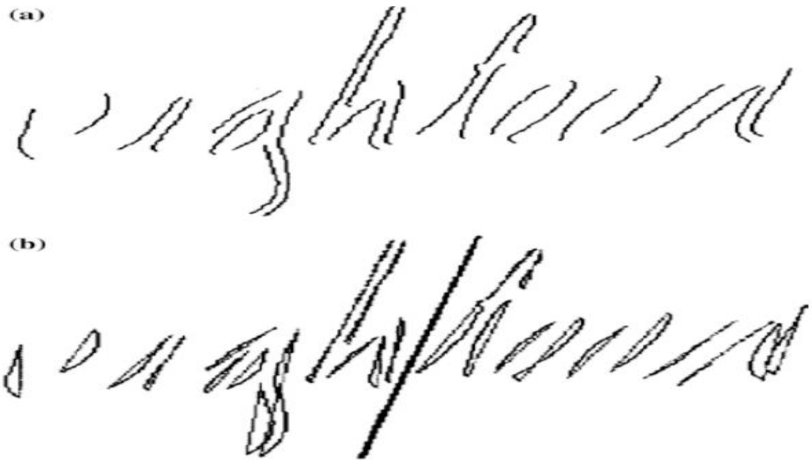


Figure 2-8: (a) Near vertical elements. (b) Average slant angle (Chaudhuri et al., 2017).

(iv) *Contour Smoothing:* Contour smoothing aims to remove errors because of the erratic hand movement throughout writing. It normally works on decreasing the number of sample points that are necessary to represent the script. Therefore, it boosts efficiency in the pre-processing steps in existence (Reshma et al., 2016; Chaudhuri et al., 2017)

(C) Compression

The conventional image compression mechanism transforms the image from the space domain into unsuitable domains for recognition. Thresholding and thinning are two familiar compression techniques. Thresholding is utilised to minimise space requirements and enhance the processing capabilities. Usually, it is advantageous that images are represented as binary images by selecting a predefined threshold. This threshold can be (i) global (i.e. use of a single threshold for the whole character image) and (ii) local (i.e. use of many thresholds' values for the whole character image). Established global and local thresholding techniques are compared through implementing a goal-directed evaluation criterion keeping in view the desired accuracy of the OCR system (Chaudhuri, 2010). The characters' singular points such as endpoints, cross points and loops are identified by some thinning methods. In a non-pixel wise thinning, global approaches handle the points. Thinning repetitions can take place. The parallel algorithms, on the other hand, surpass sequential algorithms because all pixels are simultaneously examined using similar conditions for removing (Reshma et al., 2016). It can be applied in parallel hardware in an efficient manner. Notably, compression techniques have an impact on data as unpredicted imperfections might arise, which distort the character image. As a result, significant writing data loss can happen when using these techniques. Therefore, they should be carefully applied.

(D) Segmentation

The initial stage of pre-processing produces a clean character image such that a reasonable amount of shape information is obtained, while providing the image in a normalised high compression and low noise format ready for the next OCR stage, namely, the segmentation. During this stage, a segmentation of the character image into its sub-components is done. It is worth noting that the recognition rates are directly affected by the extent of the separation of the various lines in the characters. Therefore, an internal segmentation is used, which separates the lines and curves in the characters that are cursively written (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). The character segmentation methods are categorised into three types (Casey & Lecolinet, 1996; Arica & Yarman-Vural, 2002; Chaudhuri et al., 2017):

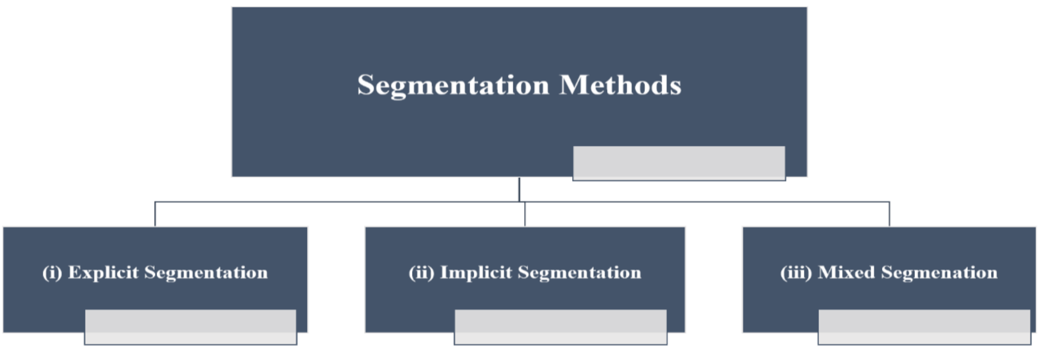


Figure 2-9: Segmentation Methods (source: Chaudhuri et al.,2017).

(a) In *the explicit method*, the segments are recognised depending on a character such as properties. The character image cutting-up process into relevant components is carried out through dissection. The character image is analysed through dissection without a certain class of shape information. The agreement of segments' general features constitutes a good segmentation criterion with those that are anticipated for valid characters. The existing character image dissection techniques utilise white space and component analysis.

(b) *The implicit method* depends on recognition as it hunts the image for matching components with predefined classes. The segmentation is carried out utilising the recognition confidence, which includes syntactic or semantic precision of the total result. The implicit segmentation utilises two approaches: (i) search process methods and (ii) feature representation segmentation of the image. The image is classified into many overlapping pieces in a systematic manner regardless of content. These methods are originated from schemes, which are enhanced for machine printed words recognition. The key principle is the utilisation of a dynamic width mobile window to deliver sequences of segmentations that are established by OCR. The second type of methods aims to segment the image in an implicit manner through subsets classification of spatial characteristics that are composed of the entire image.

(c) *The mixed* method is a combination of implicit and explicit methods. A dissection method is applied to the image aiming at covering segment, cutting the image insufficiently in different regions that the cuts made include the correct segmentation boundaries.

(E) Representation

Image representation (one of the most important factors in recognition systems) is the fifth component of OCR. Basically, this module entails binary images (or grey level) being fed into a recogniser (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015). However, for most recognition systems, a more compressed and characteristic representation tool is required, to avoid extra complexity and to increase the performance and accuracy of the algorithms. A set of predefined features are extracted (for the purpose of representation) for each class. These extracted features help to distinguish one class from another. There are three major groups of character image representation: a) global transformation and series expansion,

b) statistical representation and c) geometrical and topological representation (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

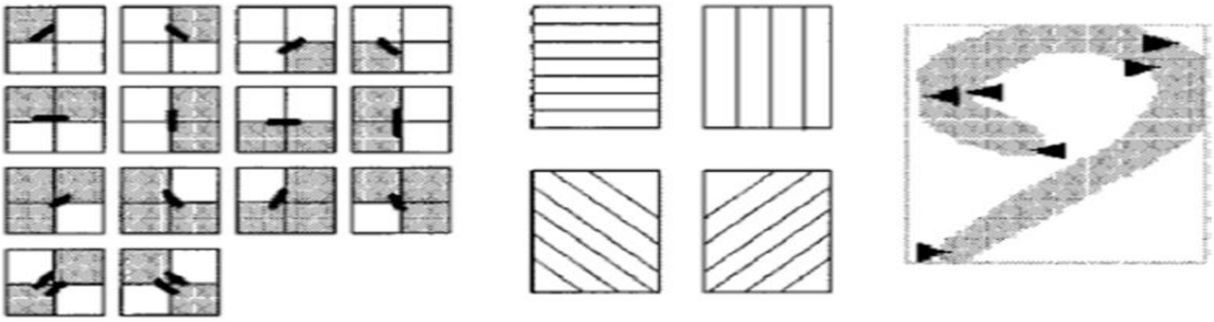


Figure 2-10: The Contour Direction (source: Chaudhuri et al., 2017).

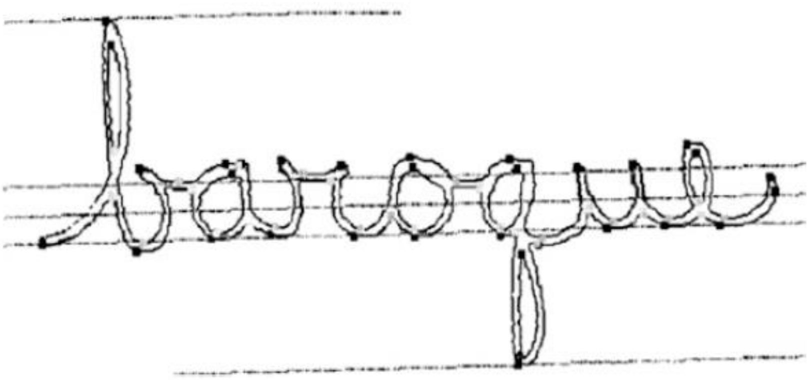


Figure 2-11: The topological features (source: Chaudhuri et al., 2017).

Figure 2-12: Arabic character and its chain codes (source: Chaudhuri et al., 2017).

(F) Feature Extraction

The sixth OCR component is feature extraction, with an objective to capture the essential characteristics of symbols. This stage of OCR often encompasses the most difficult problems in pattern recognition (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

A raster image is the simplest way to describe a character. Extracting certain features (while eliminating irrelevant attributes) that characterise symbols is another approach to OCR. The procedures for extracting these features can be categorised as follows:

- Distribution of points,
- Transformations and series expansions,
- Structural analysis.

The observed groups of features, according to Bennamoun and Mamic (2002), Qadri and Asif (2009) and Och et al. (2015), are then evaluated for noise sensitivity, deformation and ease of use and implementation.

(G) Training and Recognition

Training and recognition is the seventh component of OCR. Techniques for pattern recognition are extensively utilised in OCR systems, where unknown samples are assigned to predefined classes. According to Bennamoun and Mamic (2002), Qadri and Asif (2009) and Och et al. (2015), the four general approaches of pattern recognition are a) template matching, b) statistical techniques, c) structural techniques and d) artificial neural networks (ANNs). These approaches are neither dependent on each other, nor disjointed from each other. Sometimes, the OCR technique applied in one approach might most likely be considered a combined member of other approaches. In all the OCR techniques discussed thus far, either a holistic or analytic strategy has been employed as part of the training and recognition stages (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

The recognition accuracy is reduced because of each character representation complexity or stroke. The analytic mechanisms, on the other hand, implement the bottom-up strategy commencing from the character level and ending with an expressive result. This phase can be done by one of the following techniques:

(*a) Template matching:* The OCR mechanisms are wide-ranging depending on the feature set that is chosen from the features' long list for the image representation. Simple features involve the grey-level image frames with distinct characters, which are complicated such as the graph representation of character primitives. The OCR simplest way depends on matching the stored prototypes against the character to be identified. The matching operation generally governs the similarity levels between two vectors, which means a set of pixels in the space that is related to the features.

(b) Statistical Techniques: Typically, statistical techniques depend on three key assumptions:

1. Gaussian distribution of features
2. Using statistics for every category.
3. Given set of character images {X} one is able to extract a group of image features {Tx} _ T; x = {1,2,3 . . . , n} that indicates each single category of patterns.

The statistics, which are obtained from i features of each character, are believed to resemble specific vector space. The major statistical approaches are shown below in Figure 2.13:

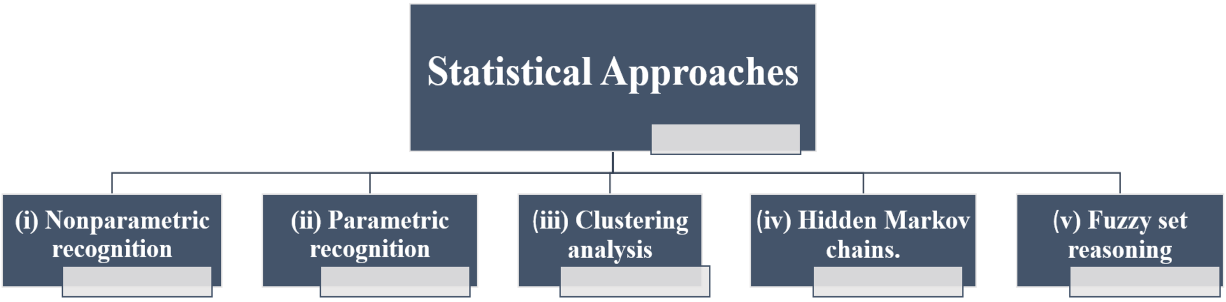


Figure 2-13: Matching Approaches (source: Chaudhuri et al., 2017).

(d) **Structural Pattern Recognition (SPR):** The description of a complicated pattern based on less complicated patterns depending on the object shape has basically formulated the idea of creating the SPR. These patterns effectively describe and categorise the characters in the OCR systems. The characters are represented by using the union of structural primitives. These primitives, which are extracted from writing, are expected to be quantifiable; relations can be found among them. The structural methods that are applied to the OCR problems are shown in Figure 2.14.

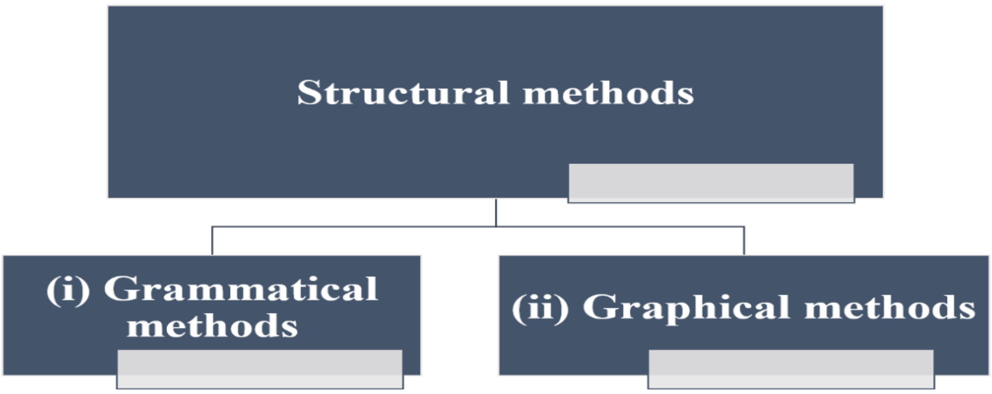


Figure 2-14: Structural Methods (source: Chaudhuri et al., 2017).

(i) Grammatical Methods: These methods take the rules of linguistics into consideration to analyse written characters. Different rules regarding linguistics are later added to the recognition model as well as considering orthographical and lexicographical rules. Some production rules are created by grammatical methods to crate from that represent each character using a group of primitives based on formal language grammars. Different characteristics including topological statistical characteristics' types must consider standardised syntactic and semantic rules. These rules might be combined by such methods. Language theory, being a formal tool, describes allowable constructions and extracts contextual information on writing through different kinds of grammar rules such as string grammars, image grammars and image description language (Makhoul & Schwartz, 1999; Chaudhuri et al., 2017). In the context of utilisation of these methods, training is carried out by describing each character by a grammar Gi. In the recognition stage, each character representation is analysed to determine to which pattern grammar the character belongs. The grammatical mechanism in OCR is applied at various character levels.

(ii) Graphical Methods: These methods present characters in the form of trees, graphs, etc. The character image primitives, like strokes, are chosen by a specific structural mechanism regardless of how the final decision about the recognition is taken (Chaudhuri et al., 2017). For each category, different techniques are used (such as tree or graphs) in the training phase in order to represent the image character strokes. The recognition phase allocates the unidentified image to one of the categories using a specific image similarity measure. It is worth mentioning that there is a range of approaches, which implement graphical methods. Figure 2.15 illustrates examples of deformed templates.

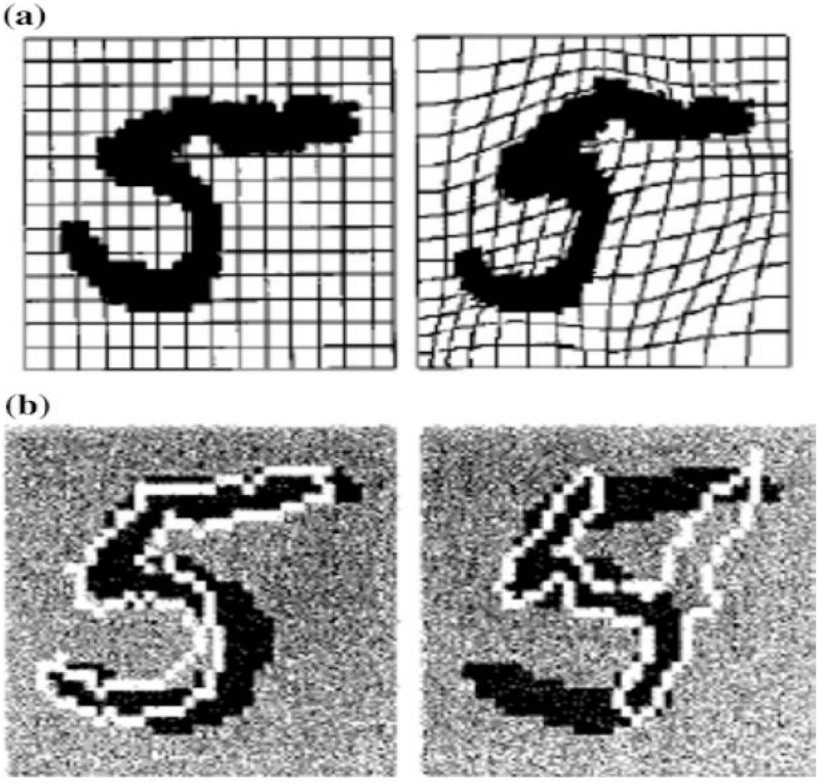


Figure 2-15: (a) Deformations of a sample digit. (b) Deformed template on the target image with dissimilarity measures (Chaudhuri, et al., 2017).

(H) Post-Processing

The final component of OCR is post-processing. Grouping, error detection and correction are some of the commonly used post-processing activities. Symbols in the text are associated with strings for the purpose of grouping. A set of individual symbols is the result of plain symbol recognition in the text. However, the symbols do not generally deliver adequate information. Based on their location in the document, the individual symbols are grouped together (based on symbols which are close in likeness) to form words and numbers. For fonts, the grouping process is easy as the position of each character is known (if fonts have fixed pitch), but for typesets, the distances between characters vary. Because the distance between words is meaningfully larger than the distance between characters grouping is, therefore, possible (Bennamoun and Mamic, 2002; Qadri and Asif, 2009; Och et al., 2015).

2.2.6.2. Extracting text from text image using OCR

In this section an example is presented showing how to extract meaningful text from a text image using object character recognition. Figure 2.16 represents the original image and binary form of the original image. The original image is directly converted to binary form. Thus, converting the original image to binary form makes it more easy to work with because it represents the characters in (0) and (1) pixel values. Therefore, it is easy to differentiate between the image background and image characters, as the background is constituted by (0) pixel values and each image character is constituted by (1) pixel value.

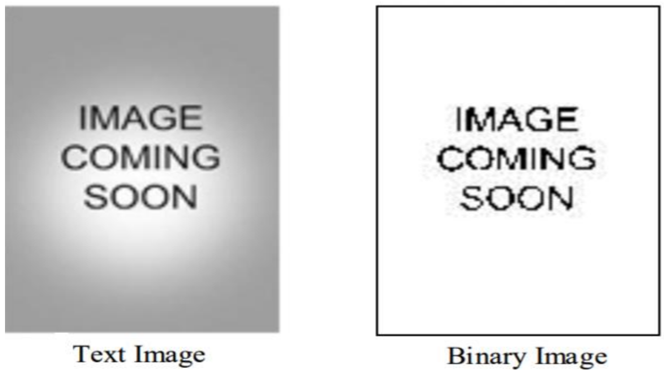


Figure 2-16: Original Text Image and Binary Text Image (source: Jana, Chowdhury & Islam, 2014).

Once the binary image is ready, the text line included in the binary image is separated from the image. The separation process is done by computing the summation value of each row in the binary image. In case the row summation is equal to zero, the separation must be done and the new line is identified. Hence, the overall summation value between two lines must be equal to (0). As a result, each binary image is divided into a set of lines, where the resulting lines are extracted gradually (i.e. one by one) as illustrated in Figure 2.17. This process is stopped when all lines in the binary image is completely extracted.

Figure 2-17: Line Extraction (source: Jana et al., 2014).

Because working with a single line is easier than working with a group of lines, extracting a single line should be performed. For every single line, each letter is extracted as shown in Figure 2.18 and Figure 2.19. This process is performed by computing the summation value for each column in the input text image. In case the summation value is equal to (0), the character should be separated. Upon finishing the separation process, all characters should be completely separated, including alphabets and punctuations, etc.

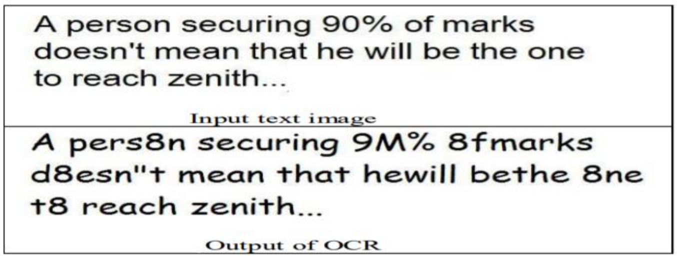


Figure 2-18: Input and Output of OCR (source: Jana et al., 2014).

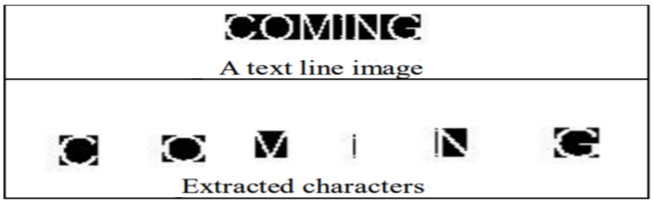


Figure 2-19: Extracted Characters (source: Jana et al., 2014).

Each character included in the image should be divided into four main regions, as shown in Figure 2.20. After that, feature extraction method is applied for every character (i.e. extracted image character).

Figure 2-20: Extracted Charter (Source: Jana et al., 2014).

The summation of the overall pixels' values (i.e. for the four regions together) and the summation of each region are then computed. The ratios for features are computed as the value of the feature of X1, X2, X3 and X4 respectively. Where:

- X1= (Summation of the pixels value of first region / Summation of the pixels value of the completed image (i.e. for the four regions together).

- X2= (Summation of the pixels value of second region / Summation of the pixels value of the completed image).
- X3= (Summation of the pixels value of third region / Summation of the pixels value of the completed image).
- X4= (Summation of the pixels value of fourth region / Summation of the pixels value of the completed image).

In order to get a high level of accuracy, features X5, X6, X7, X8, X9 and X10 are computed based on X1, X2, X3 and X4, where:

- X5= X1+ X2.
- X6= X2+ X3.
- X7= X3+ X4.
- X8= X1+ X4.
- X9= X2+ X4.
- X10= X1+ X3.

2.3. Related Works

2.3.1. The Approach Towards an Automated Marking

Automated systems for grading programming codes have indisputably a long history. For instance, ASSYST is an early-automated system, developed by Jackson and Usher (1997) for assessing/evaluating programming assignments. ASSYST despite being an assessment tool, it also assists in general housekeeping procedures including assignments submitted by students checking both the efficiency and correctness. The authors of ASSYST claim, "The system has proven to be a very useful tool that improves the consistency and efficiency of grading". In line with this development, Kurina et al. (2001) introduced the Online Judge System (OJS), which examines students' programs submitted through either a webpage or an email. The principle of OJS is a precursor of the Web-CAT system, which was discussed earlier. It is capable of collecting and then running the program against a set of pre-defined test cases. The results obtained from the testing procedure are known as "judgment", hence the name of the system, "Online Judge".

In an approach proposed by Isong (2001), a checking system was developed to evaluate the correctness of students' results on different tasks. In the program, the lecturer is responsible for the assignment specification and creation of a concrete plan to mark and evaluate student tasks. The task description is such that the lecturer can develop a wide collection of test cases from the task requirement that is used to evaluate the submissions of tasks given by each student. The key results of the developed checking program were the elimination of documentation problems in programs and better reliability for task correctness. Effective implementation of the software program must be the focus during the evaluation phase. Hence, by establishing evaluation procedures and a process for enhancing students' performance, organisations can periodically assess the effectiveness of effort and ensure continuous improvement of software programs (Isong, 2001).

Similarly, in their work titled "Production Programming in the Classroom", Allen et al. (2003) elaborate on the outstanding success of employing Test-Driven Development (TDD) in courses. In using this approach, the students are bound to pay more attention to the testing activities. This is achieved as they become more aware of the processes involved in testing. Moreover, they are required to complete the tests independently and to be motivated by comparing their test cases to other students. On the downside, the software developed by Allen et al. (2003) imposes a large amount of overhead on both the course staff and the students, warranting a new course.

Web-CAT is another tool for automated marking of programming assignments, developed by Edwards and Perez-Quinones (2008). Web-CAT is an open-source platform that allows for a web-based submission of student program codes to the Web-CAT server, which then runs the submitted code against a set of prescribed test cases and provides an online feedback to the student. The Web-CAT approach to automated testing provides both scalability and consistency through the implemented test cases. Using this tool, the assessors are able to set different criteria for the grading and feedback. For example, they can choose between having the computer grade the assignments based on passing or failing test cases, or simply using the information for student feedback and improvement purposes. Despite the well-proven performance of the Web-CAT tool, it is largely dependent on the quality of the test cases.

In another recent trend proposed by Dasarathy (2014), a massive open online course (MOOC) emerged as a result of exploiting technology advancements in the education system. MOOCs owe their rising popularity to providing fast and flexible feedback. For programming courses deployed on MOOC, students can enter their code into a browser window, where it is automatically evaluated and the results reported back to the student almost immediately. In many MOOCs, the ratio of student-to-teacher is usually high, thus demanding an automated grading and feedback system.

Zlatko and Green (2010) have recently developed an AGAF-based system, which can automate the assessment of Office applications. This tool, as described, is only deployable on Excel assignments, which is one of the many AGAF components. The authors make use of the online quiz feature of a learning management system, such as Moodle, to achieve automation of the grading and feedback processes. In this technique, an Excel spreadsheet with hidden formulas and macro functions is distributed to the students, where they can provide their answers for automated grading and feedback. To answer each question in the quiz, students are required to apply a variety of techniques in Excel; first, importing their code by copying and pasting it into the Excel workbook. The imported code is then used along with the students' response to quiz questions to compute the values of those hidden cells. In contrast, the AGAF approach allows the students to submit their work in an ordinary Excel workbook for assessment.

Plagiarism is an important element of grading any programming assignment. One of the areas prone to plagiarism is online courses. This is due to the large number of submissions involved and a lack of witnessed assurance to certify that the submitted assignment is as a result of solely the student's effort. One of the most powerful plagiarism detection systems developed to date is the Measure of Software Similarity (MOSS) (Aiken, 2005). In MOSS, plagiarism is detected by comparing the submitted work against previous submissions for similarities in grammatical structure. The tool then reports back a measure of similarity, leaving it to the discretion of the assessor to take any disciplinary actions.

In a similar work, Warne (2014) reports on a novel approach to program grading, which encompasses a BDT based philosophy as its behavioural framework. The reported system utilises the Request Specs (RSpec) and Capybara features embedded in Ruby programming language to develop and deploy test cases for an assessment of course assignments.

As a more advanced approach, Williamson et al. (2010) propose an automated scoring for the assessment of common core standards, using machine learning technology. These systems benefit from natural language processing and image processing techniques that have become more accessible in recent years. Such new approaches to the automated grading of programming assignments can offer a precise textual analysis of the student's assignments. Currently, AGAF tools cannot grade the student's text (e.g. quality of a student blog post); however, the system proposed by Williamson et al. (2010) presents the user with the option of doing so, should the computer programming courses wish to proceed in that direction.

2.3.2. Automated Marking Systems Based on OCR

The methods employed in OCR systems have greatly developed over the years, from those primitive schemes that were only suitable for interpreting stylised printed numbers, to more advanced, sophisticated and complex techniques for identifying a great variety of typeset fonts including hand printed characters. New systems for character recognition emerge constantly with the development of very powerful computing devices. Nevertheless, the highest benefit lies in exploiting existing methods through the means of integration of different technologies and making more use of the framework. The combination of segmentation and contextual analysis advances the recognition of joined and split characters.

The proposed design has seen the proliferation of several techniques used in OCR, by extracting relevant coding features from an assignment's image. This entire image extraction process forms the basis of the content-based image retrieval system (Eakins, 1999; Niblack et al. 1993; Pentland, et al. 1993; Rui, Hang and Chang 1999), which has been facilitated to index large collections of images of coding data provided by the students, followed by the remaining procedures required for the automated process.

Despite the facilitation of OCR in such an advanced manner, there are still many challenges to overcome in the development of the iMarking®, in an effort to accomplish a more user-oriented system. One of the challenges is that the standard output of images provided by the students continues to consist of low features such as poor colour textures and quality.

Masbah and Lu (2016) propose a statistical framework based on OCR and Latent Semantic Analysis, in order to relate the keywords to the characteristics of the image. This framework compares the students' answers with the correct answer. The teacher must log in to the system first, then he or she chooses one of the submitted answers. Also, the proposed framework allows the teacher to choose the optimal answer and to give the mark. It compares the submitted answer and an optimal answer, and then the similarity measure is used to compute the similarity percentage. The experimental results show that the proposed system achieves a high level of marking efficiency and accuracy.

Bautista and Comendador (2016) propose a new automated marking tool based on OCR, which assists higher education institutions to mark and store students' mark automatically. Using this marking tool, each mark of the uploaded grade sheet would be indexed to a specific student. It provides digital storing of the scanned grade sheet. The students' marks on the digital copies will be stored in a local database. The system identifies each student by his or her number, and then searches using this number and retrieves the data that relate to the number that fits the student ID number. This tool aims to increase the matching accuracy (Cheng, Qu, Shi & Xie, 2010). To ensure uniqueness, each student number with the related student grade tagged is stored in a database individually as a database record, which can be retrieved for any future query. After the scanning, the scanned student grades can be displayed. Moreover, the user can edit the result in case there are errors in the scanning and recognition of characters (Bautista & Comendador, 2016). Figure 2.21 shows how all scanned images are stored based on the following naming convention.

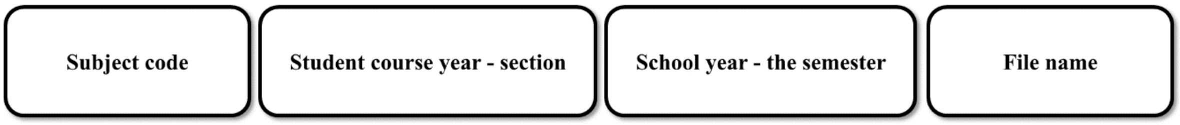


Figure 2-21: Naming convention for storing the scanned images (source: Bautista & Comendador, 2016).

Once the scanned images are stored in the system, the user can search and restore the file to view the grade sheets. The proposed system provides different functionalities such as adding new users, invoking the users to access the system, changing the user login password, generating a certification of grades for each student based on a predefined template, etc.

Uthman, Porter and Ledgerwood, (2012) provide a new system to evaluate the worksheets administered to at least one student. Each worksheet contains the presentation of one problem or more for the student to answer and each presentation contains an answer region where the student would manually, mechanically, or electronically write his or her answer using a variety of alphanumeric characters. This system works, as shown below in Figure 2.22, based on evaluating the answer sheets using a processor that analyses the digital images stored in the system's database. Each answer sheets contains the problem presentation and each presentation includes an answer region that allows the student to write the answer in different ways (i.e. manually or electronically). Then, the system generates the expected answer for each answer region, and it locates the answer region in the image. After that, the student's marks are extracted from the located answer region based on OCR. The student answers are assessed by comparing with the expected answers and generating corresponding marks.

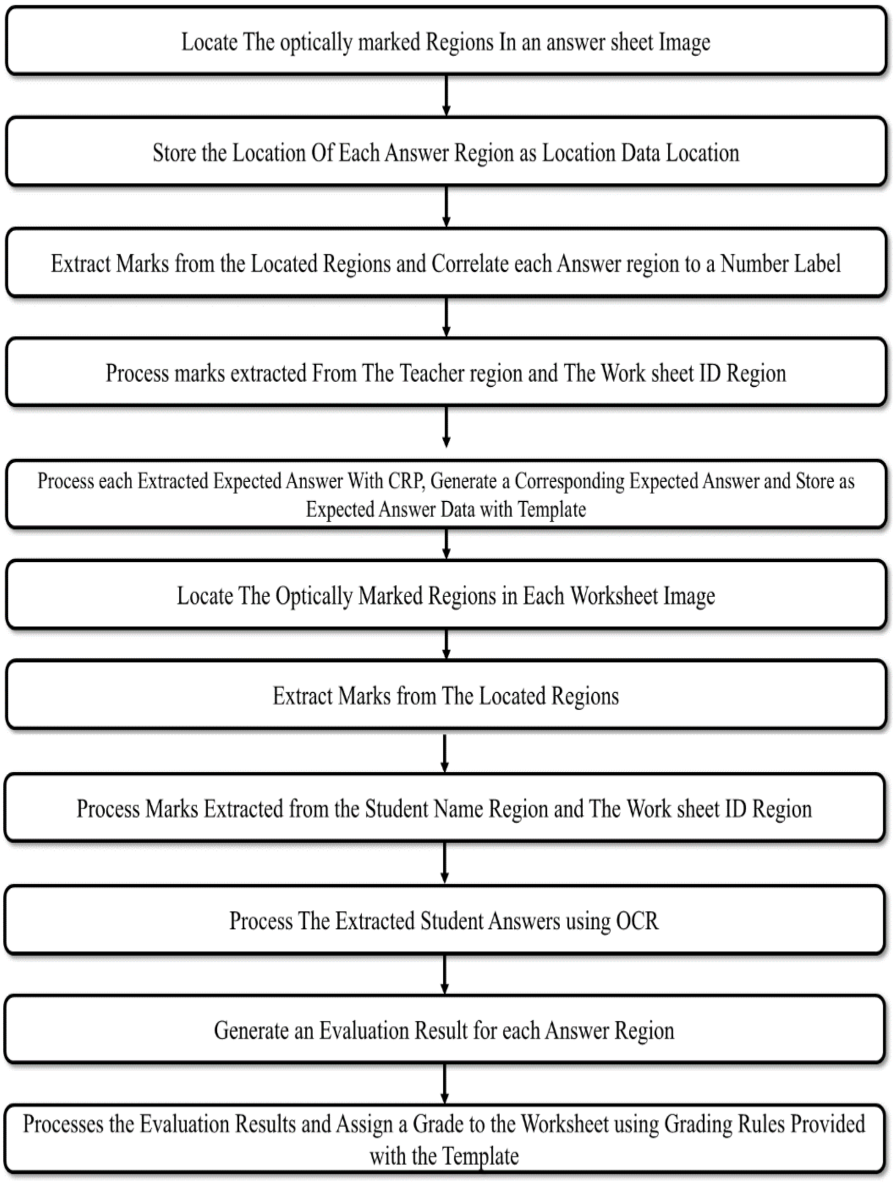


Figure 2-22: Worksheets Evaluation Process (source: Uthman et al., 2012).

Further, Mangano (2006) proposes a new system for evaluating articles and tracking student progress automatically. The workflow of this system is shown below in Figure 2.23. The grading system contains a main grader unit that can be also used for bidirectional data transmission. The teacher unit contains several components including a processor, memory, data input, a data port and a display unit. The processor is responsible for accepting any new entry of a Scoring Key based on the alphanumeric input means. This key is directly transmitted from the teacher unit to the grader unit where other components including processor, storage, a scanner with OCR functionalities and a data port are included. Inside the grader unit, assessment algorithm is used to determine incorrect answers and to provide the overall score based on a Scoring Key.

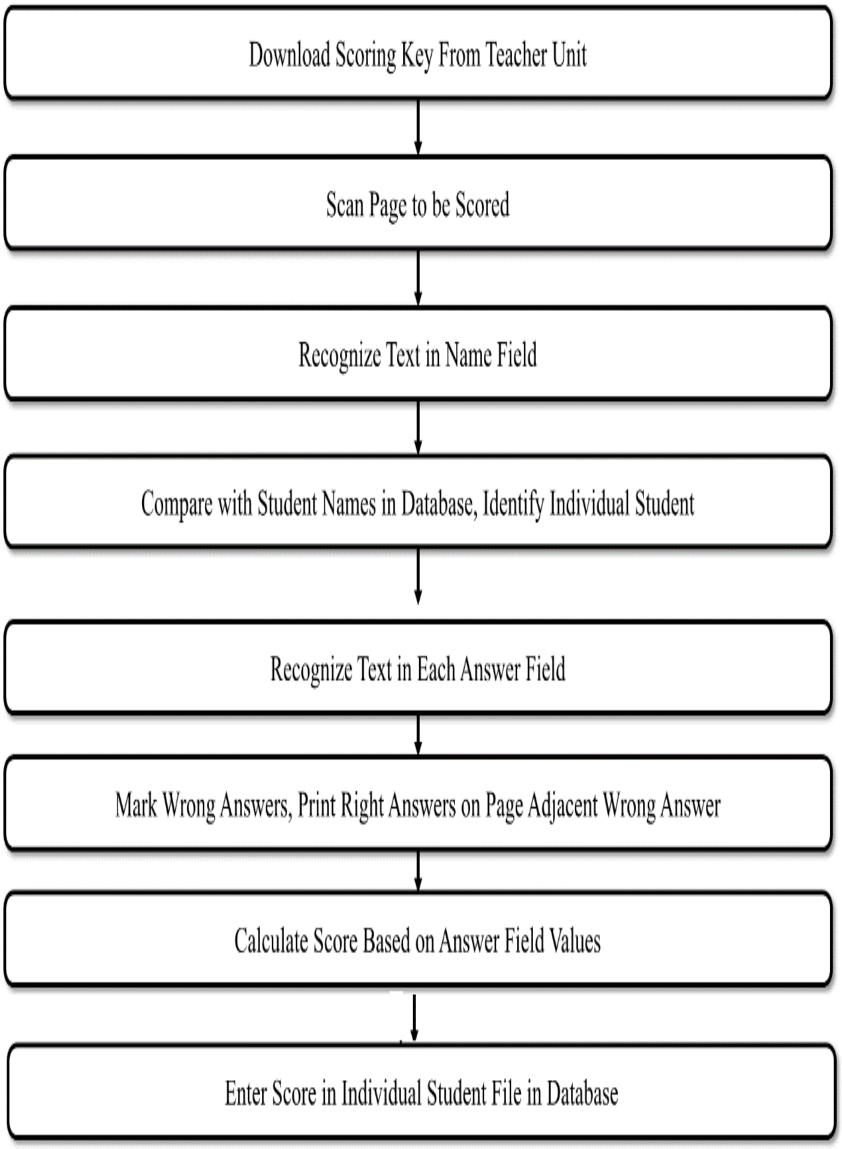


Figure 2-23: Worksheets Evaluation Process (source: Mangano, 2006).

2.4. Critical Analysis

The existing OCR based approaches for marking process are either computer-focused or do not have a fully automotive marking process. In addition, some of the approaches are adopted OCR just to extract the hard-coded marks on assignment papers, which they do not utilise in the marking process such as in the study of Bautista and Comendador (2016). Moreover, these approaches do not use sophisticated evaluation metrics that give a comprehensive and detailed marking for the submitted assignments. Unlike the proposed approach, the input is the submission of an assignment (programming codes) in an image format by the students. Then, OCR is used to extract the text from the submitted documents, which is saved as a file text. The proposed system is an integration of OCR techniques, web technology and database. The web technology is applied here for developing a user interface, which allows students to submit their assignments, and teachers to mark and evaluate the submitted answers. This process is explained in detail during a) the submission process and b) the marking process. The educators, to generate student performance reports, use the database to store student marks for later analysis. The database section also comprises two sub-sections of a) submission process and b) marking process.

In addition, the prominent feature of the proposed approach is the novel evaluation metric, which was introduced based on an experimental approach by performing a deep investigation and advanced analysis of the sample questions, to ensure robustness of the evaluation metric in calculating matching similarity and mismatching percentage. In a nutshell, the proposed approach is considered a combination of database, web expertise and OCR. Notably, web technology is used purposely to create a web interface, which allows students to submit their assignments in a central location where teachers can read and mark them and at the same time manage the scores.

2.5. Summary

This chapter provided background information on an automated marking system including generation of assessment tools, the critical challenges influencing programming courses, embracing programming testing rehearses in assignments and automatic assessment, both advantages and limitations. Optical Character Recognition was then discussed and the OCR systems and OCR techniques explored to show how text can be extracted from a text image using OCR.

Related works were examined in two main sections: an approach towards an automated marking and automated marking systems based on OCR. Additionally, this chapter focused on the most common automated marking systems on OCR. The existing automated marking systems suffer from many challenges, which need to be overcome in the development of iMarking® as a more user-oriented system. This is because the standard output of images provided by the students continues to consist of low features such as colour textures and quality. The proposed automated program marking tool in the present study seems inevitable and based on iMarking®, a web-based software that allows markers, as well as teachers, to set up assessments and enable students to submit their initial tasks and summative assignments repeatedly, with their source code to be run against the optimal result. Table 2.4 shows a summary of the existing work.

Table 2. 4: Summary of existing research.

Reference	Method	Strengths	Weaknesses
Niblack et al. (1993)	On-line image databases and query based on images' contents	It covers all image features such as colour, texture and shape of image objects and regions.	It suffers from some challenges regarding derivation and computation of attributes of images and objects that provide useful query functionality. No user interface is provided.
Pentland et al. (1994)	Direct use of the image content rather than relying on annotations, semantics-preserving image compression.	The similarity measure produced by the system corresponds to functionally similar shape. It recognises the most similar wrench or hammer from among a group of tools, even if there is no tool that is an exact match.	The average recognition accuracy over the entire database must be considered. No user interface is provided.
Benford et al. (1993)	It works based on particular messages or information in program output (to check the results of a dynamic test), or in the output of an analysis program (to check for the presence of certain messages in output from lint, for	It covers all user view, including student, tutor, teacher, course developer, system administrator.	The text being searched is not exactly specified. As it depends on regular expressions, the accuracy might be reduced. Time consuming and complex processes.

	example), or in text files (to check for particular constructions in the source of a student program).		
Jackson and Usher (1997)	The assessment process follows several steps, including answer attachment, correctness (by tutor's test data), efficiency (by tutor's test data), style, complexity and test data adequacy.	ASSYST's function is related to the housekeeping operations associated with running practical classes for a programming course. The system uses a scaling approach in awarding marks. It is simple to use and does much that a human tutor would find difficult or impossible to do without its help.	For most assessment metrics, it would be iniquitous to apply a simplistic 'right or wrong' approach, which gives either full marks or nothing at all to a student. It might be thought reasonable that a student whose code contained either no comments or only comments should be given zero marks. It would be somewhat harsh to apply the same penalty to a student whose comments accounted for eight or twelve percent of the text.
Isong (2001)	The grading plan and the test cases are the bases for coding the program checker. Checking strategy is based on presence strategy, aggregate checking and instance checking.	It is a fast and efficient way to grade a large number of student programs.	It is complex to develop a testable specification. It does not assess other aspects of the assignment solution--efficiency, style, complexity and test data adequacy. The program checker needs to consider checking strategy

			and elements of the assignment specification.
Allen et al. (2003)	It allows for a web-based submission of student program codes to a server, which then runs the submitted code against a set of prescribed test cases and provides an online feedback to the student.	It provides scalability and consistency through the implemented test cases. The assessor is able to set different criteria for the grading and feedback.	It is largely dependent on the quality of the test cases
Dasarathy (2014)	Students can enter their code into a browser window, where the code is automatically evaluated and the results are reported back to the student almost immediately.	It provides peer evaluation, because it is the best method of learning when teaching or grading.	There is no opportunity for effective assessment methods like Q&A in classroom, surprise quizzes and presentations. There are no proper evaluation methods as automatic machine assessments and evaluations are not effective. Peer evaluation sometimes leads to discouragement among students, who may drop out from the course.

Bautista and Comendador (2016)	An automated marking tool based on OCR that assists higher education institutions to mark and store students' grades automatically.	It consolidates the past grades of students faster than the usual method. The operational and error messages are easy to read and understand.	The best type of grade sheet to scan is that of the original copies of the grade sheets. An organisation should keep a standardised format, font and text layout for all grade sheets in order to increase the accuracy of scanning and reading of the OCR.
Cheng et al. (2010)	It works based on local similarity voting to recognise an ID number automatically.	It is efficient and is robust to the change of illumination and rotation. It provides a high level of recognition accuracy.	It does not consider the images with a larger angle of rotation.

Chapter 3 Research Methodology

3.1. Introduction

OCR is a critical element of any image processing system. This chapter discusses the research methodology, which includes the description of the project's phases and a summary of the activities performed. The automatic image marking process system has been designed using Java NetBeans Integrated Development Environment version 8.1, a web interface and MySQL database forming the phases to be explored in this section. This research has applied a systematic mixed methods research methodology, including experimentation, surveys with a questionnaire and making comparisons with case studies of previous research. These experiments have been conducted to mine data on outcomes from real world implementations using the system's prototype.

A questionnaire was prepared and distributed to 15 respondents. The reason for selecting only 15 respondents to evaluate the functionality of the proposal is due to the lack of people who have expertise in OCR. The respondents would evaluate the OCR marking system using different attributes as captured by the individual questions (see Appendix B). The questionnaire was administered to test the impact of the designed system on the respondents and their satisfaction rates in terms of effectiveness, suitability and the role it played in improving the abilities of students. A questionnaire was preferred as the best option for performing the evaluation amongst the respondents as it permitted anonymity, uniformity and a cost effective approach to conducting a study survey. The results from the questionnaire were then collated and analysed using SPSS.

The remainder of this thesis is organised as follows. The design and implementation are discussed in Chapter 4. In Chapter 5, Result and discussion. Chapter 6 draws some conclusions and provides recommendations for future work.

According to Ross, Marrison and Lowther (2005), experimental methods have been widely used for several years to carry out research in education and psychology as applied to the systematic approach. Nevertheless, using experimental methods in investigating technology-based studies, among other instructional improvements in higher education environments, has been comparatively inadequate. This gives the current research a good opportunity to investigate approaches through which experiments conducted in system development can be useful and productive for computer science students and their lecturers in institutions of higher learning.

The experimental method also paves the way for future scientific researchers to improve on the quality and thoroughness of research studies (Long, 2014). Particular areas comprise different forms of experiments conducted, benefits of using experiments, functional procedures for designing and piloting experiment tests, and dissemination of results generated from the outcomes. This conforms to the simple random sampling approach within the mixed research methodology used in the experiments to select 100 students for the experimentation test in this study. Much attention is given to assisting future researchers in assessing the situations that motivate or discourage application of experimental designs in relation to other methodologies used in the field of computer science.

3.2. Phases Description

Following the classification of academic assessments by education decision-makers as the most beneficial instrument to evaluate students' learning outcomes, the implication of the ability to memorise, form and integrate thoughts is daunting. Lecturers and education decision-makers are also overburdened with large workloads. In computer science, "the natural programming design process has four stages: identification of the target audience and domain, understanding the target audience, designing a new system and evaluation of the system" (Dörner, 2009, p. 51). This relates to the general context of phases description in project design that include definition of the problem, the creation of an approach to tackle the problem, formulation of research design and implementation. Gibson and Baek (2009) state that education decision-makers classify academic assignments as the most beneficial instrument to evaluate learning outcomes of the student, implying the capability to memorise, form and integrate thoughts. Lecturers and education decision-makers are also overburdened with education workload. Therefore, an automatic marking system can decrease this overhead and help lecturers focus on other important matters of designing program tasks.

The introductory discussion to the presented problem statement and research objectives and the scope of the methodological framework has been defined.

The experimental approach includes the automated testing lifecycle methodology to enable software testing as required by the designed online marking system. The most interesting and important steps in establishing the research methodology in this context are the identification of some reports to illustrate the significant impact of automatic image marking programming in grading student assignments, as the influencing critical shortcomings in earlier efforts regarding automated marking software and adopted strategies are used and the successful implementation of automated marking software is also investigated and classified accordingly.

The procedure also includes the process of collecting the primary data within the system for use in testing the iMarking® system through the three main phases: student interface, image processing by OCR, and matching the results with the database as discussed previously. OCR methods, in particular, belong to the family of machine recognition processes for automatic identification of data. External data is captured through image analysis where a transducer is used to convert the real image into digital form. The user interface is one of the important parts of network connectivity enclosed within the database; each phase requires the application of systematic methodologies, and techniques in image retrieval system, which only accepts 100% similarity in every feature extraction vector, otherwise results will illustrate a mismatch. The methodology also incorporates the evaluation of the behaviour of the interface of the developed automatic image marking system, which led to the use of surveys in administering questionnaires to correspondents, in order to help with information acquisition (Bird, 2009).

A sampling technique was also used in this research, applying a judgemental sampling approach in which there is a choice of subjects who are in the best position to provide the required information. As such, 100 students were selected randomly to participate in the experimental test of the designed iMarking® system. This simple random sampling method is appropriate since the sample of the study requires specific attributes to be attached to it in order to make sure the interpretation of user interfaces is meaningful. Along with the review of the literature, this research was implemented in its entirety and the research contributed to the development of the iMarking® system, consisting of three main phases listed below:

1. Student Interface;
2. Image processing by OCR; and
3. Teacher Interface.

3.2.1. Student Interface

Going by the framework of the prototype, the students will be provided with a graphical user interface (GUI) for submission of their files using the web-based online platform. These files are the assignments completed by students of programming courses, requiring to be marked by their respective lecturers who happen to be substituted by the automated marking process. The user interface is friendly and easy to use, responsive and concise.

Oulasvirta and Abowd (2016) claim that the modern-day user interface paradigm was built in the age of the Personal Computer and its GUI, which materialised in the ’80s. The theory of direct manipulation revealed GUI's defining characteristic is that it limits the perceptual and intellectual space between a visually accessible object and its computational correspondent. The interfaces are reliant on the Object-Oriented Programming in Java, using the NetBeans IDE as cited in the introduction. The GUI concept for the students' module is critical to numerous innovative conceptions and communication approaches. Herein, the experimental research generated hundreds of outcomes that were combined into heuristics for the GUI design.

The student interface has the element of simplicity for easier use by students who are novices in programming and human-computer interaction. In keeping with Chen and Zhang (2007), the GUI necessitates considerably less time and fewer steps than Text-based User Interfaces when used by a beginner programmer. For beginners, the use of a Text-based User Interface has a substantial difference for successive use of interfaces. As a result, the Text-based User Interface does not lessen the cognitive load for either expert users or beginners. Usability engineering experience helped in coming up with the whole students' GUI for image marking needs in programming assignments. Its focus is usability testing that generated results during interactive testing exercises in the experimental session.

The system uses its student module for login and processing of submission documents. The compilation functionality is used by the instructor to generate implementation images over the various criteria to mark the image as to whether accepted or declined and subsequently passed to the second phase which is executed for use by the students through the web-based interface. Then the submission process starts in which the students convert the results into an image and upload it to the main server where a folder is created to save the image. The web interface is used to allow the students to upload their assignment answers, while the prototype provides an automated marking process for the submission answers, which is currently in progress.

It is worth noting that efforts were made to identify some reports exemplifying the significant impact of automatic image marking programming towards students' assignments grading. Falkner, N. Vivian, Piper and Falkner, K. (2014) assert that computer-based assessment is a valuable instrument in the management of extensive classes with many students and is widely applied in the automated assessment of student programming assignments in computer science. Different types of assessment criteria come along with the recognition of the valuation of the output displayed in image form for the OCR system to analyse. It gives room for adjustments of automatic feedback, which has a greater impact on students' scores and determination in the submission of their assignments to the web interface for marking. Giving automated feedback allows the evaluation of any noticeable changes in students' behaviour in the submission and performance in programming (Fangohr, O'Brien, Prabhakar and Kashyap, 2015).

Several web-based automatic marking systems, with respect to their effectiveness, efficiency and accuracy, were considered and investigated. The influencing critical shortcomings in earlier efforts regarding automated marking software and adopted strategies used to ensure the successful implementation of automated marking software were also investigated and classified accordingly. The main page of the student interface is shown in Figure 3.1 below.

Figure 3-1: Main homepage.

The prototype of the proposed system's main page for automatic image marking process system contains the heading element (Automatic Image Marking Process System), home, marking and submission buttons. It includes a file icon on the top right corner, along with two modules for both the 'student' and 'teacher', plus direct links that might be deemed helpful by the user currently accessing the system. This homepage contains information on the main operations outlining the corresponding operations that can be executed by the student, teacher and question policy. The user interface is one of the important parts of the network connectivity enclosed with the database and each phase requires an application of systematic methodologies and techniques in its image retrieval system, necessitating every feature extraction vector to have a 100% similarity value, otherwise the results will show a mismatch. This digest reveals the simplicity, interactivity and friendliness of the interface. The home button helps in navigating back to the main dashboard when users have accessed other sections of the system and would like to come back to the main page. The marking button initiates the marking process once the file to be checked is ready for the matcher. At the same time, the submission button is used by the students to send the files required for marking to the system.

Accessing the student module requires that the user, a student, clicks on the 'Student' button in order to activate the login section. This can be achieved by either the student clicking on the submission button or an image of the student representing the student module. After a successful click, the image on the following page having the student login page will be seen. It prompts the student to enter their ID number, which allows the system to authenticate the login and allow the user to proceed to the Login page shown in Figure 3.2. The student is only able to log into the system using a correct metric number. For proper database security, each user is required to first register on the system before being able to use any of the functionalities on the system. Registration ensures that a proper ID is created for every new user and assigns appropriate user access privileges to minimise the use of system resources while enhancing the network security.

Figure 3-2: Student login page.

There are different modules operating independently on this system as the students have different access privileges to teachers. Parsons (2012) states that MySQL administration has a significant number of running database queries. It is important for the designer to take close precautions with the security of the data and in the management of the system users who have access to this data. As a result, this system employed the use of MySQL's 'grant' command to assign different access privileges to the users of the iMarking® system. In larger organisations, there are several different users with different access permissions, such as in a school environment in this research context.

It is not advisable to generalise access to the system given the privacy and confidentiality requirements for accessing the results in the online image marking system. One would not like to grant root access to all users as it would allow all users to have complete control of the server. Rather, root access must be restricted to very few people, which happens to be a system developer in this context, implying that each additional user has their own unique permission. Parsons (2012) explains how different users can be created and managed on a MySQL database, connecting from Java code to MySQL by creating a user with the required access privileges to the database. The administrator has to log into MySQL as the root user and grant access privileges to students and teachers who will be identified by ID numbers as the access credentials for login.

After the student enters the ID number on the system login page, another student page will pop up, as shown in Figure 3.3. The 'Upload Solution' and 'Logout' buttons appear, indicating that the student is already logged in and can easily attach a file for the marking process to begin.

Figure 3-3: Student homepage.

At this step on the student homepage in Figure 3.3, the student has the ability to upload the assignment file and send it to the teacher page for marking. This is achieved by clicking on the 'Upload Solution' button, which activates another dialog box for attaching the file, as shown in Figure 3.4.

Figure 3-4: Student file upload dialog box.

To proceed, the student clicks on 'Upload File' in the dialog box so as to prompt the system to browse for the file on the host computer, as shown in Figure 3.5. It automatically opens File Explorer to enable the user to select the location of the image file and attach it. Once located from the computer's hard drive or removable media, the file is then clicked so as to upload it into the system, as shown in Figure 3.5. After clicking on the image file, selecting 'Open' on the dialog box and the file will successfully upload to the teacher's marking module.

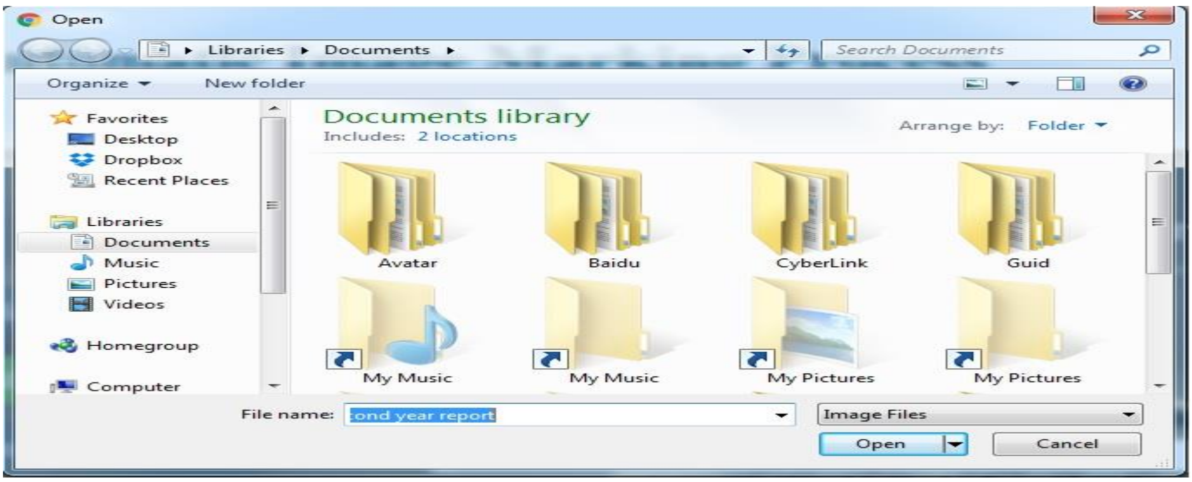


Figure 3-5: Page with a dialog box for attaching a file.

Once the initial research (described above) has been performed, the next phase, being the interactive query of evaluation and testing cycle, is carried out. There are two main features of this evaluation and testing cycle. The results of testing provided by the prototype system, in terms of accuracy with the optimal answers, show the degree of relevance and enrichment of the system's representation of the images. Characteristically, the system requires the users to indicate which results are the most relevant and to give more detailed information, such as the part of the query a given image relates to improving the quality of the results. This is high-value interactivity posed by the developed system that would serve the purpose stated in the objectives.

3.2.2. Image Processing with OCR

This phase is the main part of the system, incorporating the submission and marking processes of the uploaded images. The entry point function of the system requires students to submit their programming assignments as completed in picture form. The proposed system is comprised of a combination of OCR, web technology and database, where OCR is applied to analyse the text contained in the uploaded images and to store them as text files, while the web technology is used to provide a web-based interface that allows for the students to submit their assignments and for the teachers to mark and manage the submitted assignments by students. Finally, a database is used to store the marking results for feedback purposes and report generation by the teachers.

The image processing procedure using OCR involves the submission process and the marking process. The proposed system exploits the web services offered by OCR to extract the texts from the students' answers and from the teacher's answer models. A pre-processing step in the submitted images is performed to make sure that the images have good resolution quality. A high resolution image improves and ensures the accuracy of the extracted text. The recommended image resolution to be used as input for the image is between 200 and 400 dots per inch (DPI).

3.2.2.1. Submission Process

The submission process helps programming students in handing in their assignments electronically for purposes of marking. The computer science programming assignments follow three stages of compilation, execution and testing as the submission process begins. Islam, N., Islam, Z. and Noor (2017) explain that major phases of an OCR system consist of the image acquisition phase, pre-processing, character segmentation, feature extraction, character classification and post-processing.

The process of OCR is a combined activity containing the different phases listed with image acquisition, helping in capturing the image from an external source; in this case, the students are the source. The pre-processing phase takes on once the image has been acquired, allowing for a number of pre-processing steps to be done in order to increase the quality of the image. These include, but are not limited to, noise removal techniques, among others. The character segmentation phase allows for the separation of the characters in the image in a manner that they can be channelled to the recognition engine.

The next phase is feature extraction which is responsible for the processing of the segmented characters using a variety of features. The feature extraction phase contains the segmented characters which are further processed so as to extract different characteristics. In relation to these features, the OCR system recognises these characters as various types of characteristics that can be extracted from the images (e.g. timestamps). The extracted characteristics should be resourcefully quantifiable, minimalistic in terms of intra-class disparities and able to exploit inter-class disparities.

The next phase of the system is the character classification, which is used to map the characteristics of the segmented image to different groupings or modules. There are different forms of character classification methods, including the structural classification approaches that are centred on characteristics obtained from the structure of the image and applies different decision policies to categorise the characters (Islam, N., Islam, Z. and Noor, 2017). In the post processing, the results are not 100% after completing the classification process, particularly for complicated programming languages. In order to improve on the accuracy of OCR systems, it is advisable to perform test processing techniques which exploit natural language processing, geometric and linguistic perspectives in the conversion of errors in OCR into valuable results. However, it is important to note that the time and space complexity of a post-processor should be kept optimal and the application of a post-processor should not result in the generation of further errors. The following steps are a summarised recap of the submission process that should be observed by all students handing in their assignments electronically:

i. The student completes the programming assignment using a programming IDE.

ii. The student converts the completed data file into an image (e.g. a snapshot of the result).

iii. The student logs into the system with the correct metric number. System security requires that each user should register before operating the interface in an attempt to complete any tasks. The process of registering student users guarantees the use of right credentials such as student ID and password.

iv. The next step is for the student to attach the picture file containing the answers. A folder will be automatically generated by the system with a naming convention following the metric number assigned to each student and then the attached image will be stored in the created folder.

v. The system provides a web interface for students to upload their assignment answers.

3.2.2.2. Marking Process

The marking process incorporates the matching of results generated from the student assignment marking with optimal answers in the database. The implementation of an automatic image marking process tool promotes a system that would allow automated assessment of students' assignments by their lecturers. This approach would be consistent in the marking the process as it is computer-based, reduces costs of implementation and saves a lot of time in getting students' results ready for analysis and grading. A summary of the marking process is given as follows:

i. The teacher logs into the system using their teacher ID.

ii. The teacher selects one of the submitted assignments. The system allows the teacher to upload the model answers and adjust the marks awarded. Figure 4 shows how to upload the optimal answers.

iii. The system then uses the OCR web service to extract the text from the students' answers and the provided model answers.

iv. The system compares both texts (i.e. student and teacher submissions) with each other and produces a percentage of similarity between the two.

The marking process provides a detailed explanation of the novel measurements for computing the matching similarity of the assignments' answers that contain a mix of digits and text, as shown in Tables 5.5 and 5.6. The computation process goes through twelve steps as explained below. Note that the exactness of the text obtained from the attachment is positively influenced by high image quality and negatively influenced by low image quality. The automated marking process gives feedback to direct the students' learning, and free the lecturers on the manual marking of scripts so they can do more creative work.

Results generated from the scanning of images are cross-referenced (matched) with optimal files from the database. At this point, it is important to make use of an application programming interface called Java Database Connectivity (JDBC) for the Java programming language linking the web interface with MySQL database with the help of Automatic Image Marking Process application. The database is populated with scanned copies of image documents, as OCR in many cases employs the extraction of the text from the submitted assignments stored in the computer system as the text file.

This research provides a novel evaluation metric to calculate the matching similarity and mismatching percentage of the submitted student answers when compared to the optimal answers. The novel evaluation metric was introduced based on an experimental approach by performing a deep investigation and advanced analysis on the sample questions to ensure robustness of the evaluation metric in calculating matching similarity and mismatching percentage. The system is, therefore, considered as a combination of database, web expertise and OCR combination. Remarkably, web technology is used purposely to create a web interface which allows students to submit their assignments in a central location where teachers can read and mark them and at the same time manage the student's scores.

Borovikov (2004) found out that an OCR percentage matching algorithm results in a substantial drop in the number of false mismatches representing groupings of unprocessed spacing, AWOL characters, additional characters and character substitution inaccuracies while conducting document scan process. The modality of checking the image files involves matching similarity and mismatching percentage calculation, which are based on the content type of the submitted programming answer. The content of the answers is divided into two categories: the pure numbers or text and those having a mixture of both digits and text. High matching similarity percentage indicates a high accuracy of student answers compared to optimal answers provided by teachers, which may lead to high final marks. The mismatching percentage can also drive from the result of matching similarity. The calculation of matching similarity for each category was done by starting with the illustration of calculation of matching similarity for category 1, which includes pure numbers or text. The second illustration involved the calculation of matching similarity for category two i.e., a mixture of digits and text.

3.2.3. Teacher Interface

The web interface's main page looks like that depicted in Figure 3.6, allowing the lecturer to log in as a teacher. This profile allows for higher privileges in managing the attached images during the marking process.

Figure 3-6: Main homepage.

Similarly, the prototype of the proposed system's main page for automatic image marking process system contains the heading element (Automatic Image Marking Process System), home, marking and submission buttons. It includes a file icon on the top right corner, along with two modules for both the 'student' and 'teacher', plus direct links that might be deemed helpful by the user currently accessing the system. This homepage contains information on the main operations outlining the corresponding operations that can be executed by student, teacher and question policy.

Accessing the teacher module requires the user to click on the 'Teacher' button in order to activate the login section. This can be achieved by either the teacher clicking on the 'Marking' button or the 'Teacher' icon representing the teacher's module. After a successful click, the image on the following page showing the teacher login page will be seen. It prompts the teacher to enter their ID number, which allows the system to authenticate the login and grant access to the user.

The next page that will pop up looks like the screenshot shown in Figure 3.7, which depicts the teacher's 'Login Page'. The teacher enters the system by providing the correct metric number. As a security measure for the database access, each user must first register into the system using the 'Register' module before making use of any other system functionalities. During the registration process, a unique ID is created for every new user and appropriate access privileges are provided to them, which allows for usage minimisation of system resources and security enhancement for teachers' accounts.

Upon a successful login, the teacher is then taken to the teacher's dashboard, where all operations assigned to the teacher within the interface are accessible. These tasks can be achieved through a click on the respective buttons designed in the GUI system.

Figure 3-7: Teacher dashboard.

The allowable operations for teachers, as supported by the buttons, include 'Send Question', 'View Solutions', 'View Report', 'Add Student', 'Manage Student', and 'Logout' functions. The next step would be to click on 'Send Question', which leads to Figure 3.8.

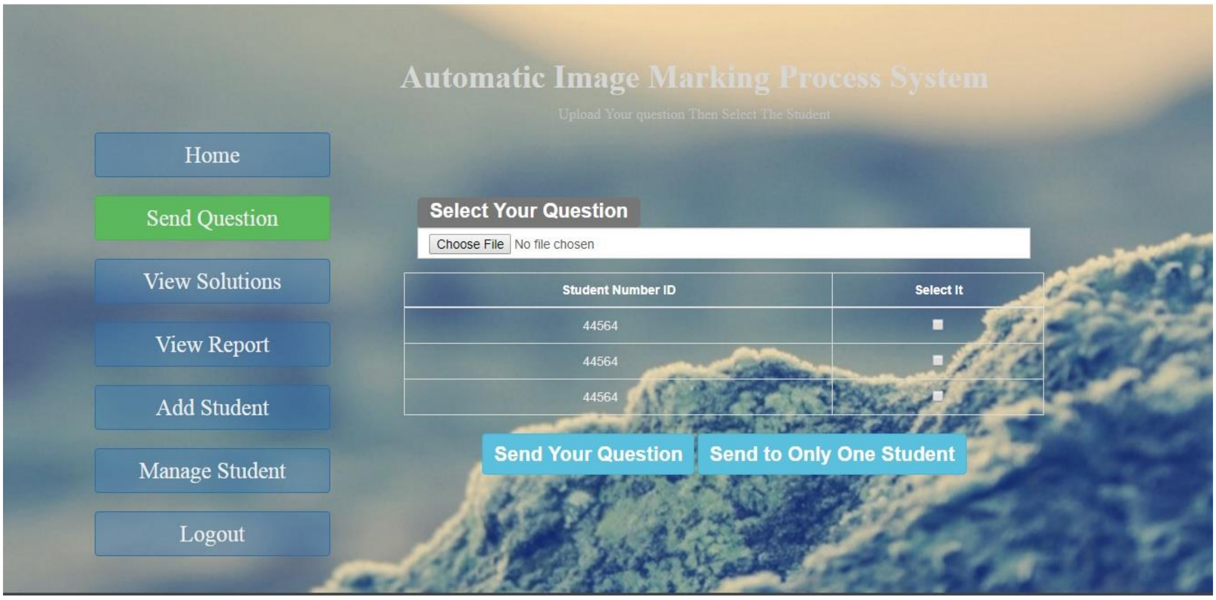


Figure 3-8: Send Question page.

At this step, the page allows the teacher to choose some students as recipients of the question or to send the question to all of the students in need of the question. The files to be selected are those uploaded by students from their module. Each file is registered with a unique code (Student ID), which serves as a naming convention to distinguish each file from another with regards to the students who attached them to the portal. To the right are radio buttons used to complete the checklist of files to attach. When the teacher clicks on the 'View Solution' button, the system will show the page as in Figure 3.9 below.

Figure 3-9: View Solution page.

This step leads to another page in which the teacher can identify those students who managed to send the assignment and can proceed with marking of the assignments by clicking on the 'Marking' Link. The use of 'Upload Original File' link and the system will appear as shown in Figure 3.10, depicting the dialog box window for 'Uploading Original File' (teacher answer).

Figure 3-10: Dialog box for Uploading Original File (teacher answer).

The teacher enters the mark and clicks on the ‘Process’ button, as shown in Figure 3.11, which will generate a report leading to the following scenarios.

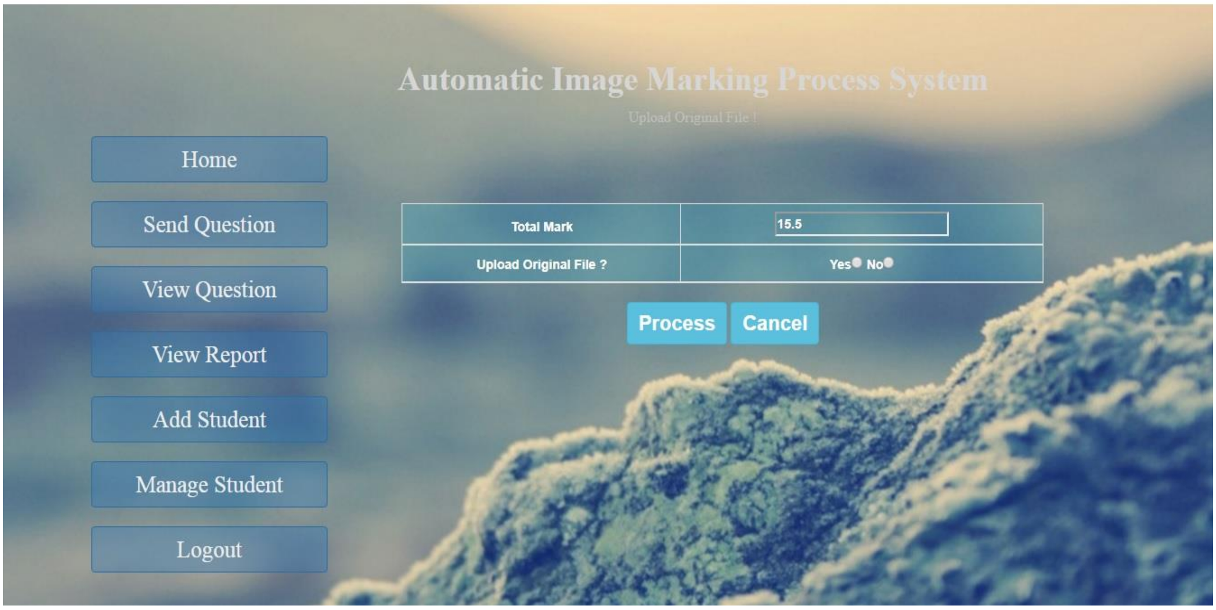


Figure 3-11: Marking Assignment Page.

3.2.3.1. Scenario One

A page showing a report will be generated for two similar images, with green ticks checking all attributes of the files under comparison. The page above shows a report on the comparison of two similar images, where the target file is compared to the original file and the results are checked with a tick or a cross. From the sample, the files have a good similarity in all the elements being compared within the files. In case all the contents are identical, the results will be ticked in all checkboxes in the right column of Figure 3.12.

Original file	Target file	Result
ENTER the voltage of collector circuit U2 in voltage	ENTER the voltage of collector circuit U2 in voltage	✓
10	10	✓
ENTER the current (I> of the circuit in Ampex.	ENTER the current (I> of the circuit in Ampex.	✓
0.845	0.845	✓
ENTER the Resister of the circuit in ohm	ENTER the Resister of the circuit in ohm	✓
10	10	✓
the collector of emitter voltage of the circuit 1.55U	the collector of emitter voltage of the circuit 1.55U	✓

THE RESULT OF FILE SCANNING	
Percentage of matching	100
Number of total lines in original	7
Number of identical lines	7
Number of none identical lines	0
Mark	20

Figure 3-12: Report on the comparison between two similar images in Scenario One.

3.2.3.2. Scenario Two

The second scenario illustrates the comparison of two dissimilar images. When the image files have different contents, it implies a mismatch that is marked with the X sign in the check boxes of the results column. The mismatch will be evident as illustrated in Figure 3.13.

Original file	Target file	Result
fl 8 £3	ENTER the voltage of collector circuit U2 in voltage	✗
student Number	10	✗
Name	ENTER the current (I> of the circuit in Ampex.	✗
1	0.845	✗
steve	ENTER the Resister of the circuit in ohm	✗
2	10	✗
Mark	the collector of emitter voltage of the circuit 1.55U	✗
THE RESULT OF FILE SCANNING		
Percentage of matching	0	
Number of total lines in original	13	
Number of identical lines	0	
Number of none identical lines	7	
Mark	0	

Figure 3-13: A report page showing a comparison between two different images.

3.2.3.3. Scenario Three

The third scenario involves the comparison of two different images with similar lines, giving a report with a mixture of ticks and crosses, as shown in the Figure 3.14 below. Here, the page shows a report on the comparison between two different images which contain similar lines.

Original file	Target file	Result
press the voltage of collector <112> in voltage;	ENTER the voltage of collector circuit U2 in voltage	✗
10	10	✓
put the current *in amper*	ENTER the current (I> of the circuit in Ampex.	✗
0.845	0.845	✓
press the resistance in ohms	ENTER the Resister of the circuit in ohm	✗
10	10	✓
the voltage will he come up1.55	the collector of emitter voltage of the circuit 1.55U	✗
THE RESULT OF FILE SCANNING		
Percentage of matching	42.9	
Number of total lines in original	7	
Number of identical lines	3	
Number of none identical lines	4	
Mark	8.6	

Figure 3-14: A report page, the comparison between two images with similar lines.

Lastly, the teacher's profile has administrative privileges for adding more students to the database and creating their profiles accordingly, as shown in Figure 3.15.

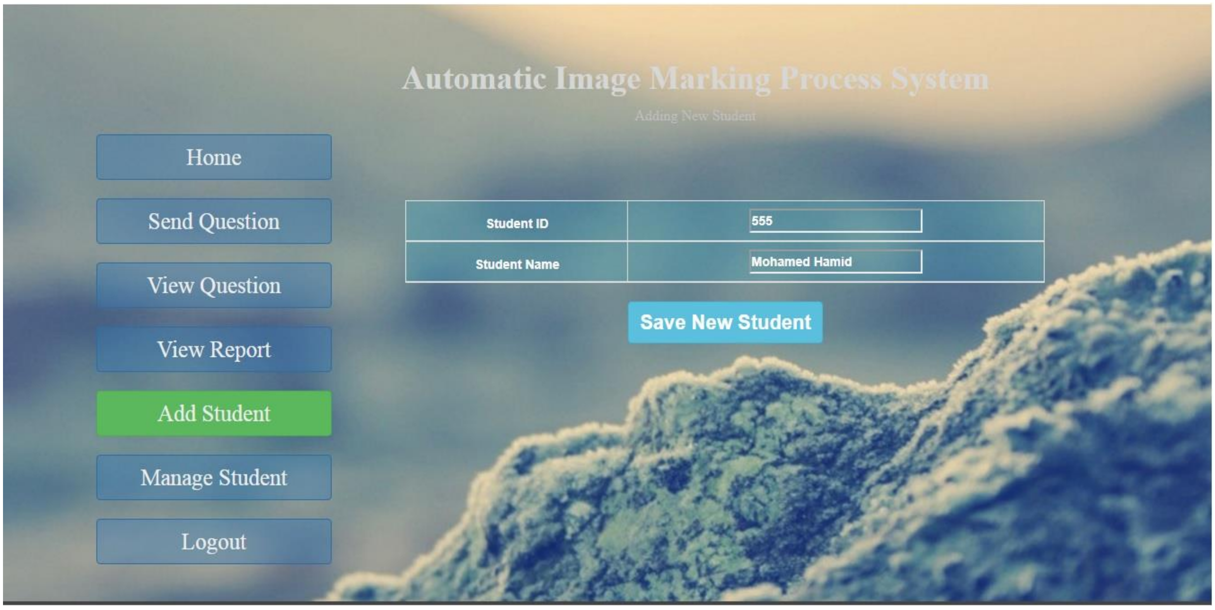


Figure 3-15: Page for 'adding new student'.

This can be done by clicking on the 'Add Student' button, which prompts a dialogue box for inputting the student's ID and name. The teacher proceeds to save the student's details by tapping on the 'Save New Student' button. Similarly, the teacher, via a click on the 'Manage Student' button, can edit student details.

3.3. Summary

This research employed a systematic mixed methods research methodology in developing a simple random sampling technique for implementation in a system that aids lecturers with marking programming assignments automatically. It investigated approaches through which the experimental approach has been adopted in the system development and testing, as this approach is deemed useful and productive to computer science students and their lecturers in institutions of higher learning. The developed iMarking® system consists of three main phases, which are the student interface, image processing by OCR and the teacher interface. The student interface provides a GUI for students to submit their image files using a web-based online platform. These files are the programming assignments in an image format requiring to be marked by their respective lecturers who happen to be substituted by the automated marking image process system.

The image processing by the OCR phase incorporates the submission and marking processes of the uploaded images. This phase is the most critical to the system as it forms the tasks of marking scripts uploaded by students, making comparisons with the teacher's reference files. Finally, the teacher interface feature enables the teacher to log into the system with the correct metric number and process the images using the automatic image processing system. Moreover, they are privileged to add students to their class using the web interface and assigning them with respective student IDs and names.

Chapter 4 Design and Implementation

4.1. Introduction

This research employed an experimental design by performing various experiments with random assignments to test the iMarking® system before the implementation. The system's architectural design and development of a user interface is deemed as a new approach towards automated image marking process, aimed at overcoming the lack of research in this area and filling this gap, the system design has been seen as the proliferation of techniques by using OCR for extraction of relevant coding features from the assignments' image. This entire image extracting process has led to the development of a content-based image retrieval system. This chapter discusses the design of the research and description of its implementation, along with a summary of the processes.

4.2. Design and Implementation Description

Specification and possible design of the prototype includes a probabilistic conceptual framework, dialogue, user interface, metadata, and definition of an interface engine. This helped with gaining a critical understanding of latent semantic analysis (LSA). The process of system design has been based on OCR, whose idea is a practical proposition since the Second World War time. Historically, wartime experience of using computers in the United States has revealed contrasts in the speed between the transcription of documents to be processed and the central processing within the computer itself.

However, after decades of improvement in computer-based technologies, nowadays, OCR allows for processing of pictures containing text and exporting of amendable files such as text files, word documents or even PDF files. Islam, N., Islam, Z. and Noor (2017) explain that OCR helps in the conversion of printed text and images into digitised form in a manner that a machine can manipulate it. The technology is a complicated problem because of the diversity of languages, font types and styles in which text can be written, and the complexity of the rules of languages among others. As a result, techniques from various disciplines of computer science such as image processing, pattern classification and natural language processing among other approaches are used to address these challenges.

The OCR technology is extensively applied in several fields. Cutting-edge OCR systems can manage nearly all image file formats, including the complicated image files like scanned documents or pictures obtained from mobile phones (Bhute and Meshram, 2014). The entire procedure of changing a file from a picture format to editable documents is separated into several steps, however, is completed by the software systems itself. Therefore, the submission of the assignments in image formats will be the first entry point of the developed system. Thereafter, the OCR will be used to extract the text from the submitted assignment and save it into a text file.

Identifying the most significant image elements, like pixel density and inversion, is the most critical part of the process. Several OCR system algorithms designed for such processes require the use of a limited variety of font sizes and fronts calling for rescaling and inversion of the images before undergoing processing step. Despite all of these developments, every step in the OCR process is important to assure the image data from student's assignments are obtained. The whole OCR analysis procedure will be unsuccessful in case one of the listed steps is not accurately managed. Therefore, it is very important that each algorithm should work properly considering the chances of having a variety of image attachments from students, necessitating the use of the best universal OCR system.

4.2.1. General System Design

The first step of the automated assessment system requires the student to submit their programming assignment in an image format. Thereafter, the data is passed through the different phases of the iMarking® system including the OCR for text extraction, web technology for student and teacher interface, and database for storage and management of the analysed assignment image data. The web interface and database components of the are both composed of two sub-sections; one for the submission process and one for the marking process. Figure 4.1 below depicts the overall structure of the proposed system design.

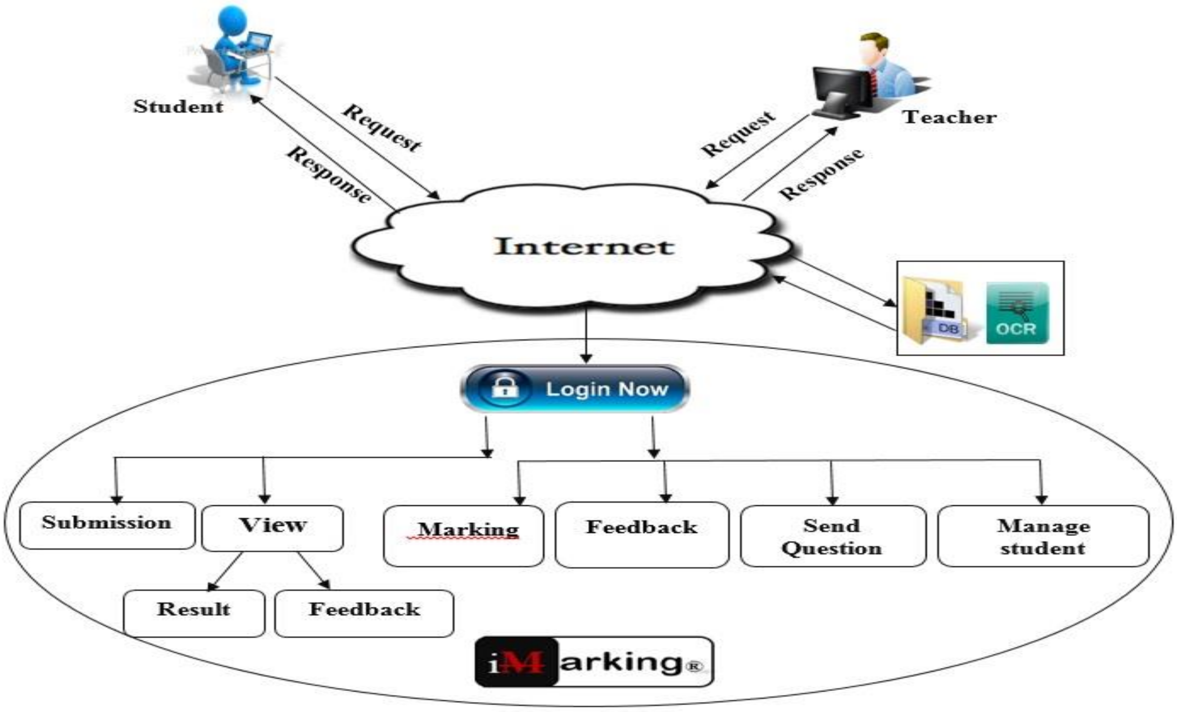


Figure 4-1: System architecture (Masbah and Lu, 2018).

4.2.2. Back-End Design

The iMarking® system is implemented using Java programming language and MySQL as database management system. The adoption of the OCR service is considered as a core part of the designed system. This section describes the adoption of OCR service into the system, with details and components used in the design of the back-end. Java programming language is used to develop the back-end of the system, while Java Server Pages (JSP) is used to develop the front-end of the iMarking® system. MySQL is used as the database management system. It further highlights the hardware and software specifications of the designed system in relation to the Hardware Specifications table 4.1 below.

The MySQL database is designed in an effort to simplify the front-end task of navigating through records of data between different multiple tables (Severdia and Crowder, 2010). The database view also helps in simplifying complex queries in connection to the web interface. These database views are defined by a series of MySQL statements that connect with several underlying tables. This approach enables the database view to hide the complexity of underlying tables from the end-users and external applications. A MySQL statement used for selecting particular data from a view is similar to that used in selecting specific data from a table. The users may have different privileges such as to read and /or write by viewing and updating or simply viewing the data as in the case of students. The developer has the privilege to view and update the data in information tables for the student records but does not have the privilege to read and/or update the tables associated with marking results. These mechanisms help in preventing the designer from any accidental use of UPDATE or DELETE on all records as these would tamper with a database already populated with records. For instance, the delete and update clauses can erase entire database records in case they are missing the WHERE clause before its condition is defined in the statements.

Data loss can also be caused by a missing semicolon before the WHERE condition, leading to the erasure of data. Nonetheless, MySQL database has the provision for using the TRUNCATE TABLE command, which can erase all the data in linked tables, and the designer cannot end the process by a DELETE trigger to save the data from being lost. There is a good measure to escape the use of the 'root' password for common functions, as a genuine keyboarding blunder can lead to calamitous repercussions for all databases under the master login. As such, the system has multiple accounts that can be created in order to present a way of restricting access and limiting each user's (student and teacher) ability to make any changes to the database through the front-end web interface. Database usage can be further restricted to a specific database table or column and a wide range of privileges can be assigned to different users, such as the privilege for the teachers to use UPDATE and DELETE database, while the students have the right to use SELECT and INSERT amongst other commands.

In terms of indexing, MySQL allows the use of a minimum of sixteen indices, depending on the type of the table created. It is important for the designer to avoid the unwarranted use of indices due to space complexity. In a broader view, every index consumes extra memory capacity by taking more storage space, which is subject to expansion as it has to be updated by the database server because of insertion, deletion and modification of the data stored in the table. The views serve as a good way of preventing accidental deletion and restricting any other database users (teachers and students) from deleting databases, table values, and data records. In this way, it is possible to easily avoid accidental erasure or modification of data in database tables, while applying restrictions to users from access to certain modules.

The contemporary database experts need to appreciate and understand the application of database systems to different processes. This is useful especially in the education sector for transforming the quality of service delivery to students through the accurate and timely marking of programming assignments for instance. The database experts can familiarise themselves with the development of database systems for both business intelligence and web-based applications such as the automated assignments marking system.

In keeping with Severdia and Crowder (2010), the authors explore database development and management practices, which outline all the characteristics of designing a database, its access, implementation, application development, and management, along with essential data analysis as used in the context of business intelligence. The design background in this resource presented a good foundation for having a hands-on project guide required skilfully developing and managing databases.

The help from existing literature was immense on the provision of detailed instruction through a number of simple, step-by-step case-centred methods of database design. In light of an introduction to database components, several case studies have been used to demonstrate the modelling process of a MySQL database, where the management of table structures using standard database normalisation rules played a major role. Dunlap (2006) gives more details on the conversion of a data model to a relational database model, defining the exhaustive use of MySQL database management system in managing database objects and in querying databases. The author explains major data-accessing approaches and then indicates how to develop complex applications that depend on programming interfaces.

The Java programming language helps in the implementation of algorithms for OCR, which aided the design of the iMarking® system. Different Java classes have been used to test and improve the algorithm. In another effort of the automated marking system, Kastelan, Kukolj, Pekovic, Marinkovic and Marceta (2012) presented a search system for text extraction based on the open-source OCR algorithm. The system is used for functional verification of computers' different sets. The final goal was to develop a program code that recognised a digit on an image and scanned against the answers given by the teacher. Using this approach, the involvement in a section of work done by most software engineers in a bid to create strong OCR systems. This software design practice was undertaken as a Java programming venture.

Table 4-1: Hardware and Software Specifications.

Hardware Specifications	
CPU	Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz (2 CPUs),
Memory	2GB
Software Specifications	
Operating System	Windows 7 Ultimate 32-bit (6.1, Build 7600).
Compiler	JDK 1.8
Database	MySQL 6.0

IEEE (2003) explains that the rising need for high performance embedded systems is creating new opportunities for the application of speech recognition systems. In a number of ways, the requirements of embedded computing vary with regards to those old-style all-purpose systems. These variations, and the fact that embedded systems like the iMarking® system have more serious limitations on cost and power consumption, cause designers to stall on their progress with implementation of computationally-demanding systems. The system requirements in this research accommodated most current machines used by many students, installed with Windows 7 operating system, which is simple and easy to use. The operating system also supports most of the applications required for the development of the iMarking® system and that adds to the reason for its choice.

The specifications help in assessing the use of contemporary computer architecture characteristics and compiler systems for conducting real-time text character recognition using OCR. In relation to the experimental results presented in this research, applying a strategic set of compiler optimisation, a 3 GHz CPU processor with 2 GB RAM size is an optimal consideration of requirements for cache resources to accomplish the real-time computing needs of the developed automatic assignment marking system and power constraints of an advanced speech recognition application.

Herein, a systematic investigative process has been adopted to increase or revise the current knowledge of automatic marking with respect to the phases of designing, testing and evaluating an advanced automatic image marking process system, which sheds light the functionality of the incorporated modules, including the back-end features. Hall (2006) cites that systematic process analysis, when used in a single case, gives some rationale for drawing causal interpretations. Given that the main models being tested are expressed in accordance to their application in a wide variety of circumstances and the appropriate causal process is defined in sufficient depth, a lot can be obtained from ascertaining the presence of the process in a single case. Since the system is comprehensive and diverse, the extrapolations and interpretations made in the analysis are not essentially less explanatory than correlations calculated between a small number of causal variables used and the results in multiple scenarios. In real-life applications, therefore, it is necessary to use a systematic process analysis after a feasibility study is conducted. In many scenarios, it was provided that the number of tested variables must be small to accommodate the collection of an all-encompassing set of results in every instance.

As discussed previously, the basis for the iMarking® system is to allow for submission of students' assignments in image form, while using OCR to extract the text from the submitted files (Masbah and Lu, 2016). Despite the fact that OCR can be facilitated in such advance manner, there are still many challenges to overcome in the development of iMarking®, in efforts of deploying a more user-oriented system. The reason for that challenge is actually the standard output of images provided by the students continues to consist of low features, such as colour textures and quality. iMarking® is a real-time examination system which is able to provide a platform for compilation, execution and grading of programming student assignments as per the criteria set by the examiner.

4.2.3. Front-End Design

This section explains the parts of the iMarking® system that are directly accessed by the users (students and teacher), allowing them to have access to further processes within their privilege rights. In the system development life cycle, building a prototype is key to the achievement of an acceptable fully functional system for users. In this thesis, the prototype has been designed based on a web interface that enables the student to submit their assignments in the shape of the image and then get marked automatically and also get feedback accordingly. It is expected that by sharing the web-based interface as part of coding data, it becomes possible to examine a range of programming problems. However, there is a significant and practical pedagogical value in the computer-based automatic assessment of such tests. The task of storing the image coding results and then comparing with the pre-existed answers in an accurate level is difficult and relatively unbearable. There is also another challenge related to the quality of images, which were identified to be a bit low. Thus, there was an enormous need to further investigate how to improve the quality and the readability of the images for scoring large-scale assessments.

The web-based application helps achieve a convenient interaction between the automatic marking system and the users. This approach to the user interface is considered as the best application to design for this research project because it uses a website, commonly accessed as a webpage, as the main interface serving users as the system's front-end. It offers benefits to both students and teachers as the system users, as they have the privilege to access the application from any computer that is data connected to the internet via a standard browser. The building web-based applications for users are an involving task for the designers, considering all the system requirements in terms of functionality. The interface contrasts with the outmoded traditional desktop applications that are run from local computer machines after installation. For instance, most programs used in computers like IBM SPSS Statistics, Microsoft Office Suite and SAS among others, are desktop applications. In contrast, programs such as online banking applications, Dropbox, Gmail and Google Drive among other applications, have communication link with a given database. These applications allow users to perform various functions using web browsers such as Google Chrome, Mozilla Firefox, Safari, Opera Explorer and Edge among others, rather than using the software installed on their computers; hence, the reason for using web-based applications in this work. The modern-day internet users are very familiar with Google applications, which are possibly the most eminent among web-based applications. In this research, an online assignment marking tool, known as iMarking®, has been developed and integrated into the web environment. This indicates that nearly any desktop application can be built as a web-based application in a similar fashion to that implemented in the iMarking® system.

4.2.3.1: Benefits of Web Interface in the Design

The use of web-based application design is critical to this research, especially for the student and teacher interfaces. It has several advantages that blend in the education sector and business environments at large as described in this section. Firstly, the web interface is cost effective and universally accessible by system users (Yang and Liu, 2012). The fact that a web browser is required in accessing the iMarking® system allows users to have a common environment for access.

In this research, the web interface has been systematically tested with various commonly used browsers for any incompatibilities and to ensure if it could be accessed from any web browser. Google Chrome proved to be better suited for its accessibility and system resources allocation in the general system requirements on Windows 7 Ultimate operating system edition. As such, it is needless to test it using different web browsers and operating systems. It also made the lifecycle development and debugging much easier, due to its simplicity, saving unnecessary costs.

In relation to the traditional approaches, the web-based application is accessible from anywhere, provided that users are connected to the internet. This advantage gives the teachers and students the latitude to choose when and where to log into the system and execute a particular task. Universal accessibility from any place makes the interface usable globally from any location, helping international students to submit their programming assignments even when on vacation back in their home countries. Therefore, the designed web-based application eradicates the notion of remaining in a specific and fixed location, like in classrooms, school offices and laboratories, in order to submit or check assignments by students and teachers, respectively.

Secondly, the web interface is easy to customise and can be accessed by a wide variety of computer devices (Ali, Alrasheedi, Ouda and Capretz, 2014). In desktop application environments, system designers have to physically get to the computer workstation and manually update the content and apply hotfixes in the back-end. Nevertheless, iMarking® system's web platform makes it easier to make changes to the website remotely and apply necessary hotfixes to the web server database in a custom manner.

Different groups, such as the student interface group and teacher interface group, can have different customised appearances for their application environment. The administrator can remotely define various preferences and looks to the interfaces to give it a set of distinctive characteristics for different user groups. Users with different devices such as tablets, portable digital assistants, smartphones and phablets among others, can conveniently access this platform, provided that they have a browser and internet connection. In addition, the users have the privilege of interacting with the interface in a manner deemed suitable by them.

Thirdly, the web interface is flexible in terms of managing the existing workload (Barrett and Kipper, 2010). In the design, various specifications were improvised for the hosting web server where the capacity of the process and storage space were adjusted accordingly. This implies that whenever there is an increase in the number of students in the programming class, a corresponding increase in the number of users using system resources can be experienced, which can be remedied by upgrading the processor and storage capacities. This can also be boosted by being clustered through load balancing, in order to distribute the tasks within the servers. In the present day, most companies like Google and Microsoft normally use multiple low-priced servers such that a failure in one server does not take the whole system down. Similarly, more servers can be added to boost the capacity of the current services without necessarily affecting the overall system performance of the application.

The web interface has increased security and a variety of flexible core technologies (Formanek and Zaborsky, 2017). Web-based systems are characteristically implemented on dedicated servers, which are constantly checked and maintained by expert server administrators. In comparison to the desktop platforms, no client machine is monitored physically implying a tighter security and avoiding any possible breaches. In the building of a web-based application, a variety of core technologies can be applied. With the help of the Java-based solutions (Java Platform Enterprise Edition), technologies like Java Server Pages and Servlets are enabled. Hence, the web interface is generally secure, easy to build and more convenient for the users.

Lastly, the image processing by OCR runs the major process of comparison and marking of the assignments (in the shape of the image) automatically. As cited in the design phase, a number of OCR algorithms need some predefined range of font sizes and foregrounds to be applied in order for the image to be rescaled and inverted before processing. In spite of the all these developments, every step in the OCR process is important in order to get the image data from student's assignments. The whole OCR process will fail in case only one of its steps cannot handle a given image correctly, therefore, it is very important that every algorithm must work correctly on the highest range of images, that is why the best universal OCR system was used in this research.

4.3. Summary

In this chapter, the design and implementation of the proposed optical character recognition-based approach for the automatic image marking process are explained in detail. The proposed approach is implemented in the form of a web-based application which consists of two main designs: (i) back-end and (ii) front-end. The back-end design discusses the adoption of the OCR service which is considered a core part of the designed system. The image processing in the OCR phase incorporates the submission and marking processes of the uploaded images. This phase is the most critical to the system as it forms the tasks of marking scripts uploaded by students, making comparisons with the teacher's reference files. In addition, the proposed novel evaluation metrics are implemented, and integration between OCR and evaluation metrics is discussed. Meanwhile, the front-end design discusses the implantation of interfaces that enable both teachers and students to interact with the system functions. The student interface provides a GUI for students to submit their image files using a web-based online platform. These files are the programming assignments in an image format, which require marking by their respective teachers who happen to be substituted by the automated marking image processing system. The teacher interface feature enables the teacher to log in to the system with the correct metric number and process the images using the automatic image processing system. Moreover, they are able to add students to their class using the web interface and assign them with respective student IDs and names. Finally, the benefits of having a web interface at the front end is thoroughly explained.

Chapter 5 Results and Discussion

5.1. Overview

This chapter comprises two main sections: the presentation and description of the results, and the discussion. The results of the different parameters from the respondents are presented, namely, age, gender, satisfaction level and the consistency of the image using an automatic image marking system to provide accurate feedback. The results were generated from a questionnaire distributed to fifteen respondents.

Additionally, in the discussion section, the results of the system, in terms of its effectiveness in resolving the issues mentioned in the problem statements, are evaluated. Manual assessment marking has had a negative impact on students' performance, which informs the rationale of this study in the problem statement. It is recognised that the development of an automated program marking tool to automate the marking process would provide feedback to orientate the students' learning and allow teachers to be more creative instead of manually marking assignments.

The system results obtained, while tested using dummy variables provided by some students as discussed below, help in the evaluation of the stability, reliability and accuracy of the iMarking® system. The results are further compared with the outcomes of other existing benchmarks within the industry. The benchmarking process allows for comparison of the designed system in relation to design processes, the general architecture and overall performance metrics used by the frontiers in the computer industry. At the same time, the benchmarking process can enhance the selection, planning and delivery of projects in the context of research management.

5.2. Results representation

As regards the analysis of image OCR, the study began by analysing the demographic distribution of the respondents. The importance of explaining the values of age brackets and demographics in user behaviour for any computer-based system has increased in recent years. The technological development and fear of change have made it necessary to perform age comparison of user behaviour concerns for the platforms set for use. These include standards and their variation across the age groups and general demographics.

Several authors investigate the value differences across age and gender groups, but mostly within single culture analyses (Shoham, Florenthal, Rose and Kropp, 1998). The examination of value variations between gender parities of males and females explains the value differences across age, and the immediate effect of gender and age on value importance. As such, these authors give a primary cross-nation overview of the effect of age and gender on value preferences.

In keeping with the data sampled from corresponding couples, many findings have emerged. In the first place, a number of values are preferred more by females than males. Furthermore, value preferences vary in terms of age across the gender groups. Lastly, by making a comparison between the findings and those reported in previous studies, different preferences relating to value across various cultures are noted. The outcomes of the current study are important for global institutions of higher learning, which require segmentation analysis of performance indices in terms of gender representation in different cultural environments.

The first item to be investigated in the analysis is the age distribution of the respondents. The findings of the questionnaire, which was distributed to fifteen respondents, are presented in the bar graph in Figure 5.1. The study also considers the gender of respondents; the results are presented in Figure 5.2.

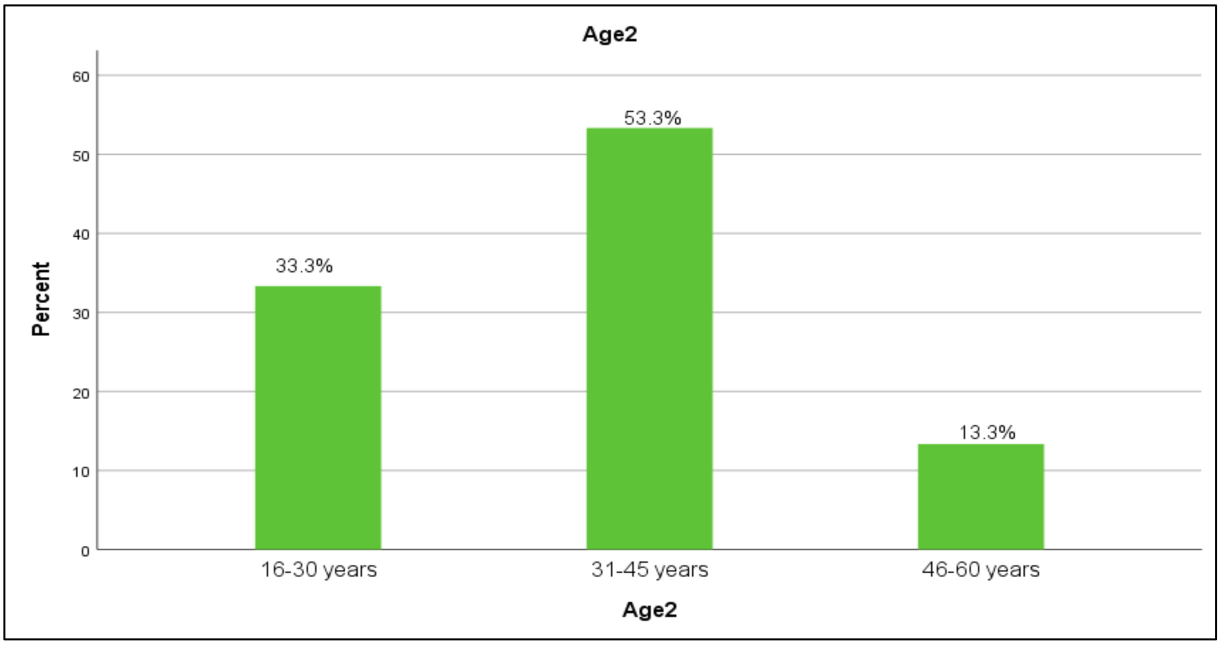


Figure 5-1: Age distribution of the respondents

The above figure illustrates the differences between the age groups of respondents. The 16–30 years group and 46–60 years group represent 33.3% and 13.3%, respectively, while the 31–45 years group has the highest distribution among all the age groups investigated.

Table 5-1: Descriptive Statistics.

	N	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
What is your age?	15	21	49	34.53	10.035	.074	.580	-1.453	1.121
Valid N (listwise)	15								

The youngest respondent in this study was 21 years old while the oldest respondent was 49 years with a mean age of 34.53 and a standard deviation of 8.689.

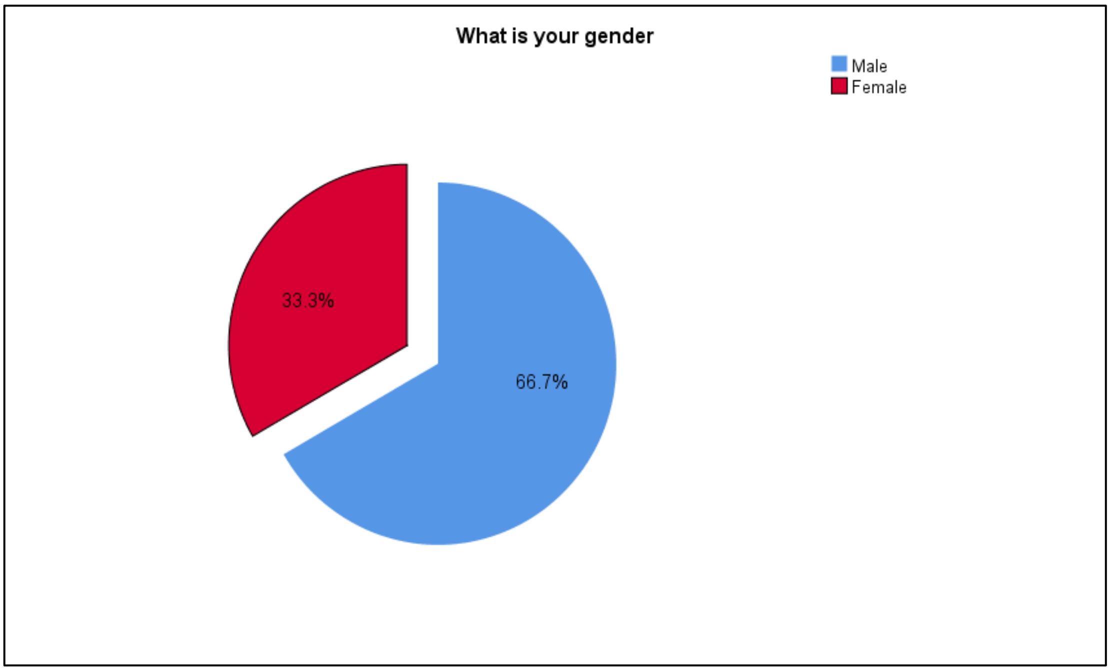


Figure 5-2: Gender distribution of the respondents.

The investigation of the respondents' distribution between genders is also an important parameter in the present study. As shown in Figure 5.2 the distribution is higher for males than females, 66.7 % and 33.3 %, respectively, among the respondents.

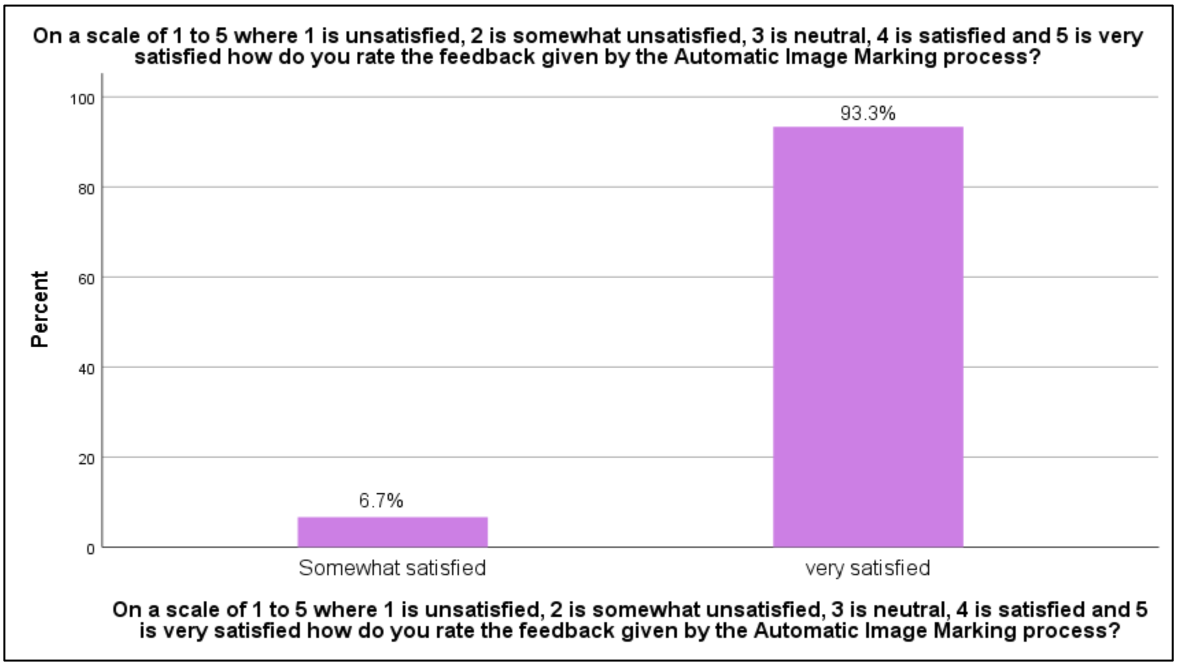


Figure 5-3: Satisfaction level with the Automatic Image Marking system.

Another element taken into consideration is the respondents' satisfaction percentage. The above figure shows that among the respondents, 93.3% are very satisfied with the OCR system, compared with the marginally low 6.7%, who are not very satisfied with the system.

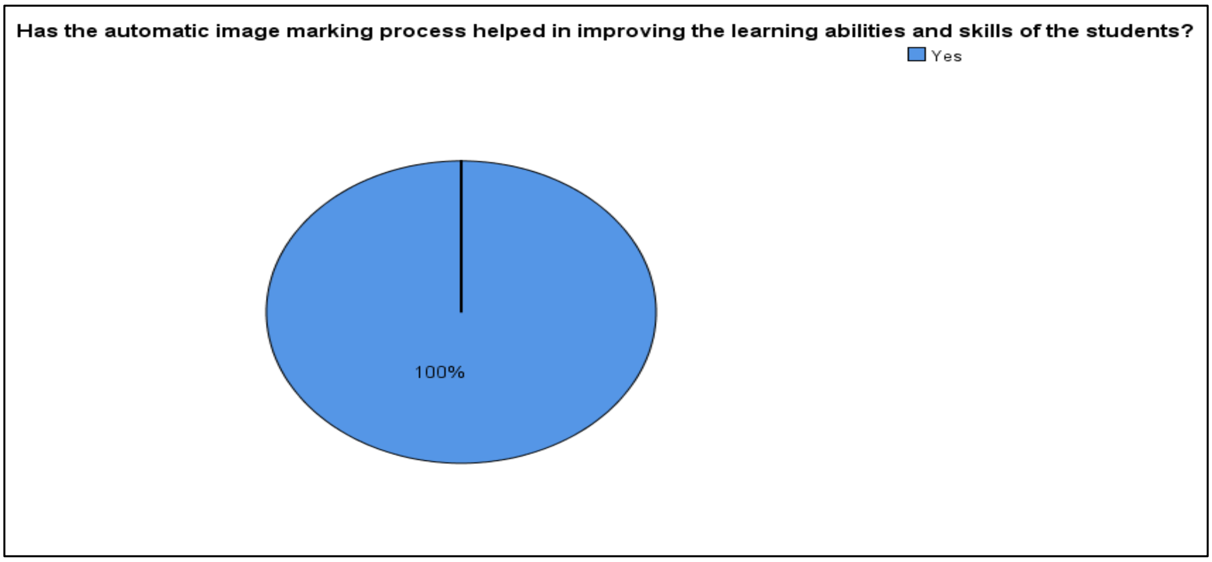


Figure 5-4: Impact of automatic image marking system (iMarking®) on improving learners' abilities and skills.

The above Pie Chart demonstrates the impact of the iMarking® on improvement of learners' abilities and skills. Significantly, the results show 100% improvement in the learners' abilities and skills with the iMarking® system.

Table 5-2: Gender Distribution.

	Has the automatic image marking process helped in improving the learning abilities and skills of the students?		
	Yes	**No**	**Total**
Male	100%	0%	100.0%
Female	100%	0%	100.0%
Total	100%	0%	100.0%

Both genders shared the same sentiments that automatic image marking systems would help improve the learner's ability and skills. The study then sought to establish the difference in opinion by both the genders. The researcher conducted a chi-square test and presented the results in table below.

Table 5-3: Pearson Chi-Square.

What is your gender? * Has the automatic image marking process helped in improving the learning abilities and skills of the students? Cross tabulation.				
			Has the automatic image marking process helped in improving the learning abilities and skills of the students?	
			Yes	Total
What is your gender?	Male	Count	10	10
		Expected Count	10.0	10.0
	Female	Count	5	5
		Expected Count	5.0	5.0
Total		Count	15	15
		Expected Count	15.0	15.0

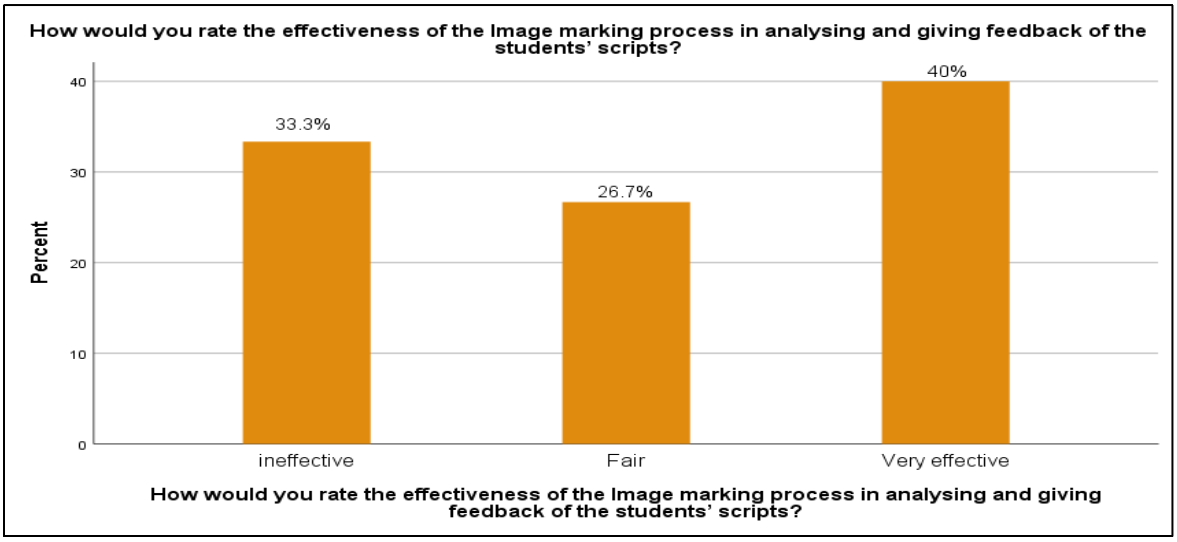


Figure 5-5: Effectiveness of the automatic image marking system.

The results of the effectiveness of the iMarking® system in analysing and providing feedback on the students' scripts are shown in Figure 5.5. The system is very effective with marginal significance of 40%, and rated 33.3% and 26.7 % as ineffective or fair, respectively.

The high percentage rate of ineffectiveness is due to several reasons; for instance, some of the respondents are not confident in the system effectiveness of the image marking process. Moreover, some do not understand the objective of the study and have no clear idea about the OCR system. Another possible reason is that respondents are anxious that an unclear image could be sent despite the originality of the answer.

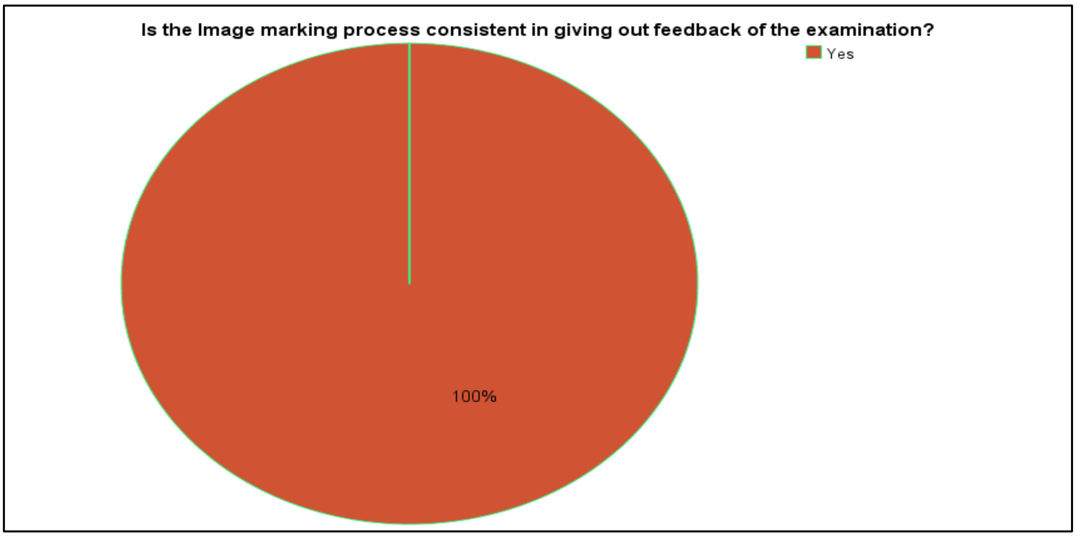


Figure 5-6: Consistency of automatic image marking system in giving accurate feedback.

The respondents' opinions were then sought on the accuracy of the OCR system in marking the examinations. The results shown in figure 5.6 illustrate the consistency of up to 100% accuracy in terms of consistent marking and giving examination feedback.

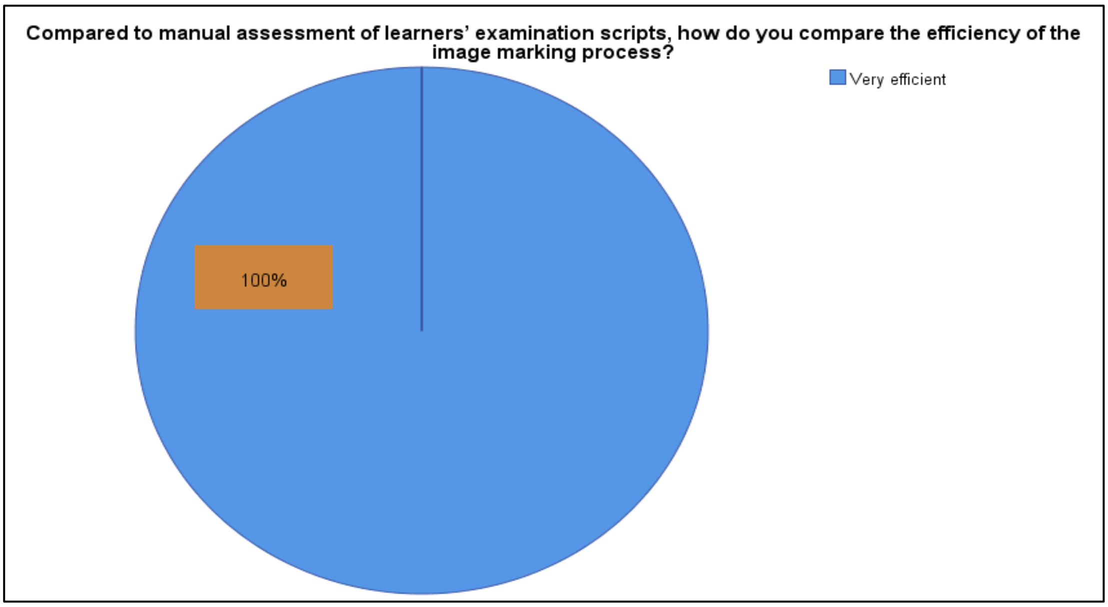


Figure 5-7: Efficiency of automatic image marking system.

The research also considers the respondents' opinions on the efficiency of the image marking system in comparison with the manual assessment of learners' examination scripts. The results show that the image marking system is much more efficient (100%) when compared with the manual assessment.

The summary of the statistical results is shown in table 5.4.

Table 5-4: Approval rating of the image marking system features.

Statistics						
		Recognition of the script speed	The complexity of the system implementation	Pattern recognition of the script	Feature detection by the system	Biasness of the OCR system
N	Valid	15	15	15	15	15
	Missing	0	0	0	0	0
Mean		2.67	3.73	3.33	2.80	3.47
Median		2.00	4.00	4.00	3.00	4.00
Mode		4	4	4	2	5
Std. Deviation		1.397	1.280	1.113	1.373	1.457
Range		4	4	4	4	4

The approval ratings of features related to the proposed image marking system are presented for the data analysis output files of every individual item. The respondents were also asked about their views on the similarity of the input in the systems and the results of marking. The response is presented in Figure 5.8 below.

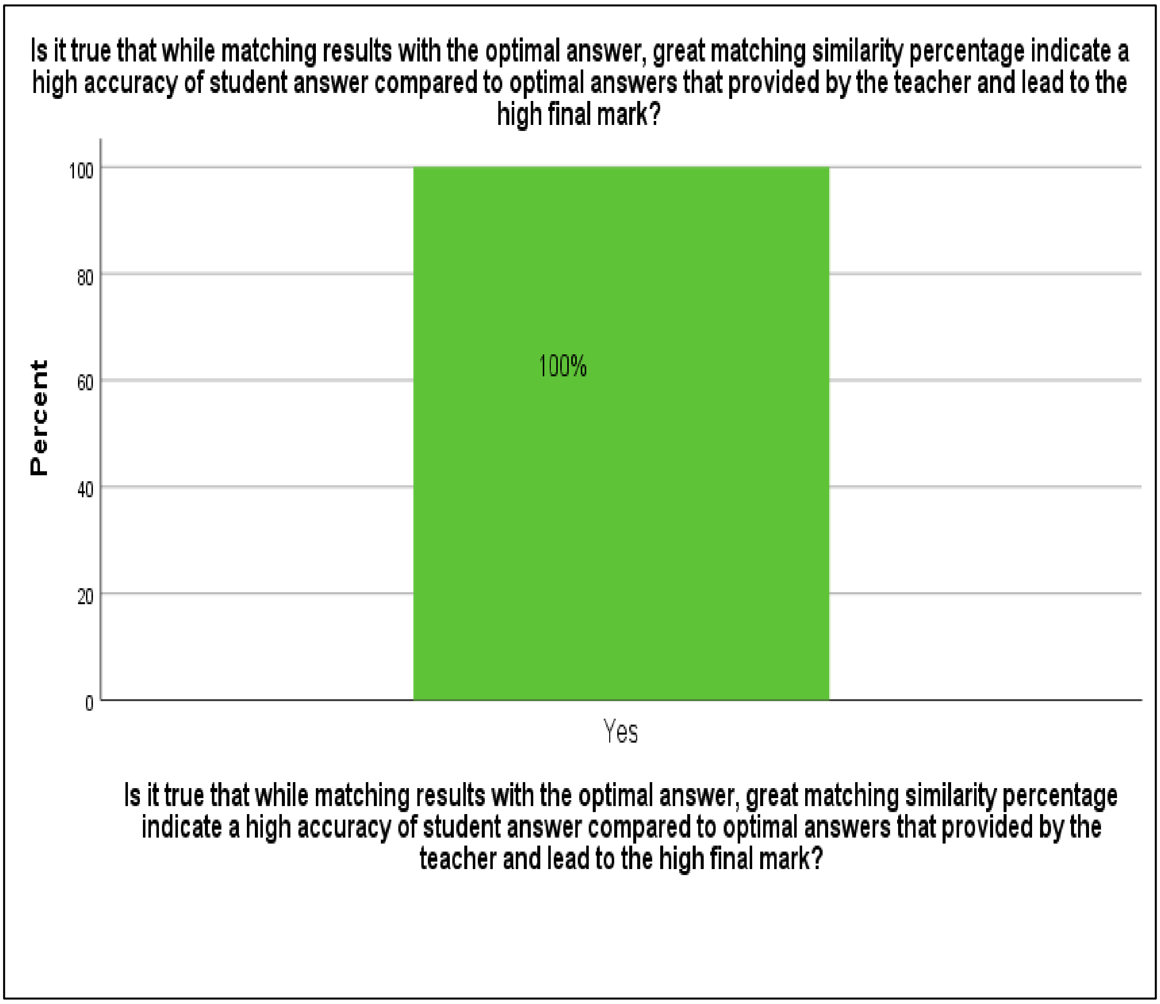


Figure 5-8: Significant matching similarity produces high marks.

The investigation of responses as regards the similarity of the input in the system and the marking results shows that 100% agreed on the significant matching similarity of the student's answer and the optimal answer provided by the teacher, leading to a high final mark.

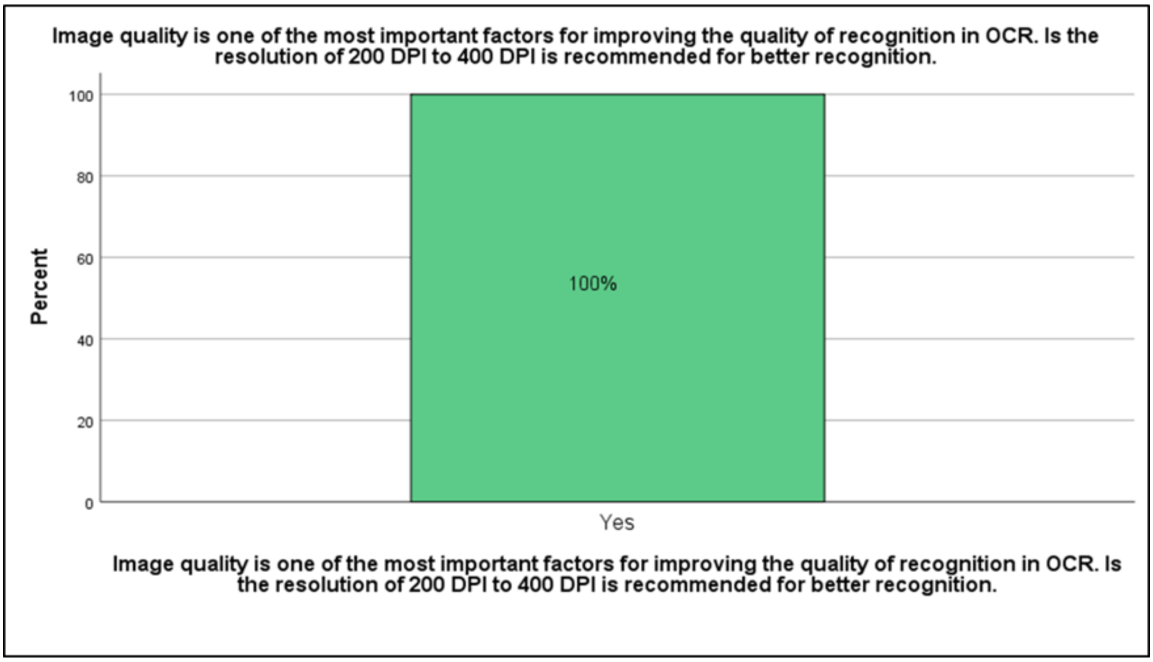


Figure 5-9: Image quality determines the quality of recognition.

Image quality is another determining factor for improving the quality of the recognition in the OCR system. The respondents make judgements on the image quality and recognition performance. Figure 5.9 shows that 100% agree on the effect of the image quality on the recognition of the OCR system, and that a 200 to 400 DPI image should be recommended for better recognition.

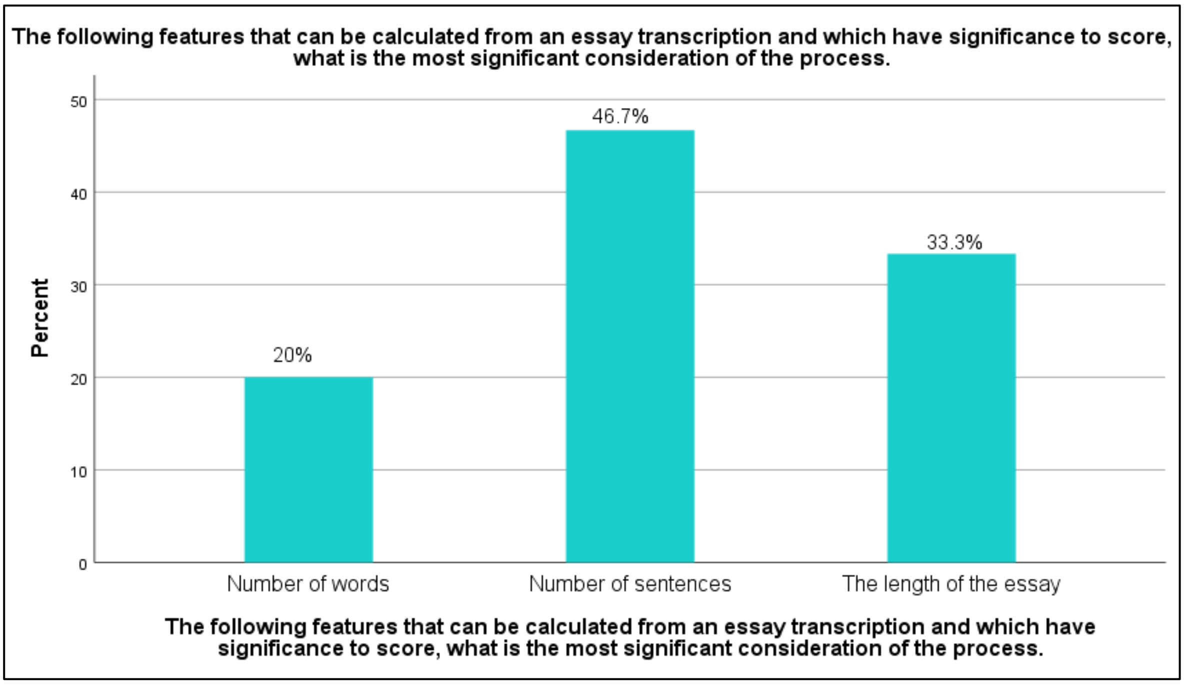


Figure 5-10: Significant features in essay transcriptions considered during image marking.

The above figure illustrates the features of the essay that can be calculated. The results show that number of sentences is most significant, with 46.75%, while the length of the essay and number of words are 33.3% and 20% respectively.

The contents of the submitted image are extracted using OCR, while comparing every extracted bit of information with a prescribed answer model to determine whether there is a match. Figure 5.11 provides screenshots of both the students' submitted answers and the text extracted by the OCR web-service.

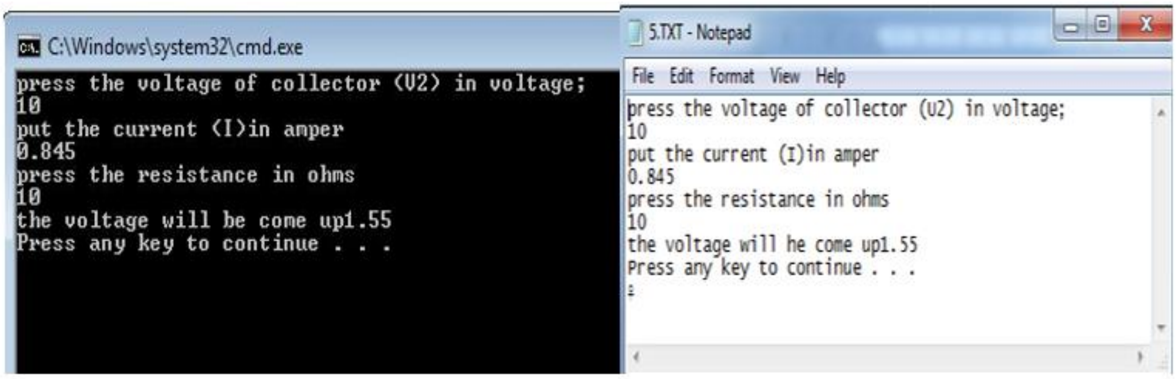


Figure 5-11: Images from students' submitted answers and the text extracted

The online OCR system is generated using an optical scanner and the resulting image creates the graphics file by detecting the individual coding patterns, as well as performing feature detection, also known as feature extraction. Hossain, Amin and Yan (2012) describe a rapid feature extraction method for OCR, which computes the projection of each part formed, by means of partitioning an image. The extraction process explains that it can be classified as a statistical approach of a JPEG image represented using a set of n features, which can be considered as a point of n dimensions in the featuring space.

The main objective of featuring a selection of OCR web technology is to construct linear decision boundaries in the feature space, which correctly separates the different characters of an image in an appropriate way. Normally, this approach is used to reduce the dimension of the feature set, mainly for easy extraction and fast computation, where reconstruction of the exact image is only required in an editable format.

The experimental results further highlight the testing of the iMarking® system on a sample of 100 student answers and the comparison of different images. Appendix A contains the results with a comparison between different images, showing the percentage of match and mismatch in the 'Matching' and 'Mismatching' table columns. The quality of image recognition is largely dependent on the image quality itself. For image recognition purposes, a resolution of 200-400 DPI is ideally desired. The researcher built the prototype and all screenshots of step-by-step access are provided as an explanation of how the prototype works. On the main page, there are three main options, where the user can select various buttons with respect to the tasks to be executed. Both the teacher and the student can use these buttons. When a student is using the system, after clicking the submission button or the image of the system, a different page will pop up. On this page, the user enters his or her student or teacher credentials to log in; that is, the ID and password issued during the registration procedure.

On the upload page, the student uploads the image and sends it to the teacher's page. Once it is confirmed that the image has been uploaded, the teacher can log in using an ID and password to do some checks prior to processing for assessment and feedback purposes. The relevant teacher feedback can be given to the student, with some control to influence the provided answers. Machera (2017) states that feedback from students is used to evaluate their performance in a classroom setting. The system also judges the quality of the image and provides feedback to improve any future retrieval processes. For feedback on provided answers, the iMarking® mechanism closely depends on the online interface, either to allow a direct input or to analyse the quality of the image. Once the student's images and teacher's questions are both uploaded, iMarking® verifies the submitted answers in comparison with the teacher's answers and starts processing to match the results.

Hattie and Timperley (2007) acknowledge that feedback is one of the most powerful influences on learning and achievement. In the system prototype testing and analysis, a sample of 100 students' answers was used to test the automatic image marking process system with teacher feedback. All the students submitted their assignments in JPEG (image files) image format, which was later converted to a readable form, i.e., TXT (text files), using the online OCR converter. An examination of the results obtained from the OCR converter shows 100% matching level in most cases of matching comparisons, revealing a high matching similarity percentage. This implies a high accuracy of student answers when compared with optimal answers provided by the teacher, leading to higher final marks for the students. The mismatching percentage can also drive from result matching similarity. Tables 5.5 and 5.6 explain the calculation of the matching similarity for each category. However, the verification techniques for automatic assessments of programming assignments have been successfully executed with input and output feedbacks in order to improve the overall results.

In the present study the results are matched with optimal answers. This presents a novel evaluation metric to calculate the matching similarity and mismatching percentage of the submitted student answers, when compared with the optimal answers. The researcher arrived at the novel evaluation metric empirically, by performing a deep investigation and advanced analysis of the sample images to ensure robustness of the designed evaluation metric in calculating matching similarity and mismatching percentage.

The matching similarity and mismatching percentage calculation is based on the content type of the submitted programming answer. The content of the answers is divided into two categories: 1) pure numbers or text, and 2) mix of digits and text. The high matching similarity percentage indicates a high accuracy rate for the student answers compared with optimal answers provided by teachers, which in turn suggests a high final mark for the students. The mismatching percentage can also drive from result matching similarity. In the following subsections, it is explained the calculation of matching similarity for each category.

First, the computation of matching similarity for category 1 (pure digits or text) provides a detailed explanation of the novel measurements for computing the matching similarity of the assignment answers that contain only digits or text, as shown in Table 5.5. The computation process has seven steps, which are all described in Table 5.5.

Table 5-5: Calculation of matching similarity for category 1.

Method 1		
Index	Variables	Description
1	X,FM,TLO	The system uses variable X to *calculate the value of X, which is a constant value where* $X = \frac{FM}{TLO}$ ..... Equation 1 Where FM means the final mark for the question, whereas TLO means the total number of lines in the original file.
2	G, NWO, NWT	The system uses variable G to calculate the grade for each line (text or numbers) where $G = \frac{X}{NWO} \times NWT$ ..... Equation 2 NWO means number of words in the original file, and NWT means number of words in the target file.
3	Final Mark	To calculate the summation of final mark for target file, the system uses variable FM where $FM = \sum_{k=1}^{n} G_K$ ..... Equation 3

4	P	To calculate the percentage of matching for each line, the system uses variable P, which is a constant value where P $=\frac{1}{TLO} \times 100$ …… Equation 4
5	MPL ,NILT,NILO	To calculate the percentage of matching for each line, the system uses variable MPL where MPL $=\frac{\text{NILT}}{\text{NILO}} \times P$ …… Equation 5 NILT means number of identical words in the line in target file, and NILO means number of identical words in the line in the original file.
6	TPM	To calculate the total of percentage matching, the system uses variable TPM where TPM $=\sum_{k=1}^{n} \text{MPL } k$ ..... Equation 6
7	TPMIS	To calculate the total percentage of mismatching, the system uses variable TPMIS where TPMIS $= 100 - \text{TPM}$..... Equation 7

Secondly, the computation of matching similarity for category two (mix digits and text) section provides a detailed explanation of the novel measurements used for computation of the matching similarity metric for those answers that contain a mix of digits and text. The computation process has twelve steps, as shown in Table 5.6.

Table 5-6: Calculation of matching similarity for category two.

Method 2		
Index	Variables	Description
1	X	The system calculates the value of X, which is a constant value using Equation 1 in table 1.
2	G1	To calculate the grade for string or numbers, the system uses variable G1, whereas *G1*= $\frac{\text{X}}{2}$ …… Equation 9
3	SP, NISO,NIST	To calculate the grade for line in the string part, the system uses variable SP, whereas SP $=\frac{\text{G1}}{NISO} \times NIST$ …… Equation 10 Where NISO= number of identical words in the string part (in original file) and NIST = number of identical words in the string part (in original file).
4	NP, NINO ,NINT	To calculate the grade for line in the string part, the system uses variable NP, whereas NP $=\frac{\text{G1}}{\text{NINO}} \times NINT$ …… Equation 11 *Where* $NINO$ = number of identical words in the number part (in original file) and $NINT$ = number of identical words in the number part (in original file).

5	FML	To calculate the summation of final mark for target file for each line, the system uses FML variable, whereas $FML = SP + \mathrm{NP}$….. Equation 12
6	FM	To calculate the summation of final mark for target file, the system uses FM variables, whereas $FM = \sum_{k=1}^{n} \mathrm{FML\ k}$ …… Equation 13
7	P	To calculate the percentage of matching, the system uses P variable, which is constant value using Equation 4 in table 1.
8	PL	System calculates percentage for the line (PL), which contains (text or numbers), whereas PL $=\frac{P}{2}$ … Equation 15
9	PS	To calculate the percentage matching for the line, which contains strings (PS), whereas PS$=\frac{NIST}{NISO} \times \mathrm{PL}$ … Equation16
10	PN	To calculate the percentage matching for the line, which contains numbers (PN), whereas PN$=\frac{NINT}{NINO} \times \mathrm{PL}$ … Equation17
11	MPL	To calculate the summation of percentage matching for each line (MPL) in target file, whereas $\mathrm{MPL} = \mathrm{PS} + \mathrm{PN}$ … Equation18
12	TPM, TPMIS	The variables mentioned in table 1 and using equations 6,7

In the majority of OCR-based approaches, the marking process is done manually; the grade is handwritten on the answer paper then OCR is utilized to extract the given grade and save it in the database. These approaches suffer from the following core drawbacks: (i) this process is time- and effort-consuming; (ii) mistakes might be made in totalling up the marks; and (iii) unclear handwriting and background colour of the exam paper might negatively affect the accuracy of OCR.

Depending solely on OCR for grading and marking is not accurate as there are many factors that might affect its accuracy. Therefore, in this research, novel evaluation metrics are proposed for marking and grading of the submitted student answers to achieve the second objective of this research. These novel evaluation metrics do not depend on OCR for marking and grading, rather they depend on an automated procedure and novel metrics to mark and grade the student’s answers. The student’s answers are categorized based on the answer content into two main categories. Category 1 means the content contains pure numbers while Category 2 means the content contains a mix of digits and text. Categorization of the content and the metrics are formulated based on the observation and experiments of different question samples. The aim of these evaluation metrics is to calculate the similarity match based on seven novel equations for Category 1, and 12 novel equations for Category 2.

As shown in Tables 5.5 and 5.6, TPM and TPMIS are calculated for each category (i.e. pure numbers or text and a mix of digits and text) based on novel equations that vary based on the category.

Each equation consists of different factors for Category 1 and Category 2 which are listed in Tables 5.5 and 5.6, respectively. The proposed approach will first utilize OCR to extract the text from the input images that represent the student answer and the original image which represents the optimal solution, then the factors that pertain to the original image (such as TLO and NWT) will be calculated. The text extracted from the input images and original images consists of at least one line which will be compared against each other; if the first line consists of pure text or numbers then the equations of Table 5.5 will be calculated otherwise the equations of Table 5.6 will be calculated. The proposed approach will mark each line individually based on different factors such as MPL, NILT and NILO for the first category or PS and PN for the second category, and the marking process will be stopped when the last line of the input text extracted from input images is reached. The final grade will be the summation of the marks for each line of input text. Table 5.7 shows an illustrative example of the evaluation metrics.

Table 5-7: An illustrative example on the evaluation metrics.

	Input Text	**Optimal answer text**		
Rectangle shape area formula Please enter L : 10 Please enter W: 10 Result: 90		Rectangle shape area formula Please enter L : 10 Please enter W: 10 Result: 100		
Lines of input text		**Lines of optimal answer text**	**Category**	**Mark**
1	Rectangle shape area formula	Rectangle shape area formula	1 (pure text)	10
2	Please enter L : 10	Please enter L: 10	2 (Mixed)	10
3	Please enter W: 10	Please enter W: 10	2 (Mixed)	10
4	Result: 90	Result: 100	2 (Mixed)	5
	Similarity Match			95%

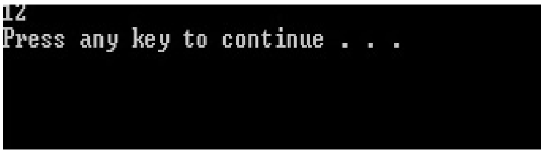
As shown in Table 5.7, the text that corresponds to the input and original images is extracted, then the evaluation metrics will be applied to each line of the input text that represents the student's answer. The similarity match will be calculated based on the equations listed in Tables 5.5 and 5.6. For example, in line number 1, the equations in Table 5.5 are used as this line contains pure text, while for line number 2, the equations in Table 5.6 are used as this line contains mixed digits and text.

Nirmala (2010) argues that we normally interact with documents with text printed on complex colour backgrounds and requiring the heavy use of OCR to extract the content. The readability of textual contents in such documents is lowly rated, which is down to the complexity of the background in the colouring of the foreground text data alongside the colours of the background. Segmenting the foreground text within such document image files is very much needed for an easier reading of the file contents using human means or by a machine. The author designed an innovative approach for the extraction of text from coloured document images with complex backgrounds that informs this study's objectives. In addition, the arithmetic side of the algorithm process can be rearranged by changing the inner loop, the size and the image parameter. During the process, the allocated function reserves a vector in the memory with the dimension provided as a parameter.

A sample of 100 questions was collected and made available. Each question is associated with five different student answers, which were used for the evaluation process, as shown in Table 5.8. The sample in this research is divided into two categories. The first category represents those student answers that consist of pure digits or text, as shown in Figure 5.12, while the second category contains those answers that are in the form of both numbers and text, as shown in Figure 5.13. In table 5.7, T1-Q1 and T1-Q2 have the exact similarity with Q1 and Q2, respectively, which means that the student who submitted these answers should receive the full mark.

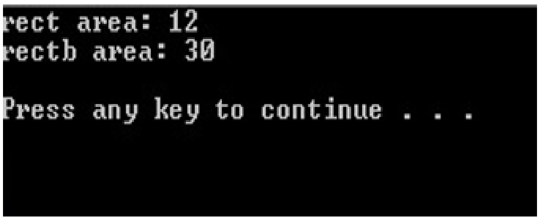
Table 5-8: Sample of the question and its associated student.

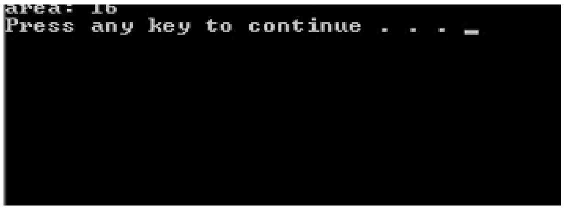
No.	Comparison between Original Image No	With Target Image No.
1	O1	T1-Q1
		T2-Q1
		T3-Q1
		T4-Q1
		T5-Q1
2	O2	T1-Q2
		T2-Q2
		T3-Q2
		T4-Q2
		T5-Q2



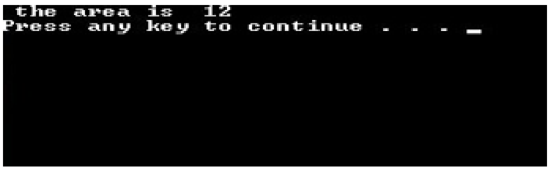
(A) (B)

Figure 5-12: Sample of student answers that consist of text and numbers.





(A) (B)



(C) (D)

Figure 5-13: Sample of student answers that consist of text and numbers.

In the experimental results presented, the system calculates the matching percentage, mismatching percentage and the final mark for category 1 using the equations given in Table 5.5. The system then calculates the matching percentage and mismatching percentage for category two using the equations given in Table 5.6. Figure 5.14 shows the average matching and mismatching percentage for each category of 100 samples. In this experiment, the teacher enters the mark for questions, which is 30, and uploads the optimal answer. Figure 5.14 shows the calculated final mark for student answers based on the sample given in Table 5.8.

Figure 5.14 shows that the proposed system can accurately calculate the student's final mark, which is attributed to the novel evaluation metric proposed here. This has an advanced analysis of extracted text, in addition to the adaption of the OCR web service. Moreover, the quality of the submitted image negatively impacts the accuracy of the representation of the student's answers. One of the most essential aspects crucial to the enhancement of the recognition quality is the image quality. It is recommended to have a resolution of between 200 and 400 DPI to achieve a high recognition rate. The table below reveals, in terms of time, the similarities between the proposed system and the manual technique.

Table 5-9: The comparison between the proposed system and the manual method in terms of the processing times.

Image No.	Processing Times in Seconds (Proposed System)	Processing Times in Seconds (Manual Way)	Saved Time
T1-Q1	0.00000154	5	4.99999846
T2-Q1	0.000001155	6	5.999998845
T3-Q1	0.0000016	8	7.9999984
T4-Q1	0.000001001	9	8.999998999
T5-Q1	0.000001222	10	9.999998778
T1-Q2	0.0000018	30	29.9999982
T2-Q2	0.00000173	25	24.99999827
T3-Q2	0.0000019	25	24.9999981
T4-Q2	0.00000197	37	36.99999803
T5-Q2	0.00000173	42	41.99999827
Total Saved Time			196.9999844

Table 5.9 shows that the proposed system outperforms the manual method of marking the students in terms of processing time. The proposed system saves up to 197 seconds for marking a total of ten questions and this aligns with the proposed system's objectives, which is to save the teacher's time. The processing time can be further decreased by increasing the specification of the proposed system hardware.

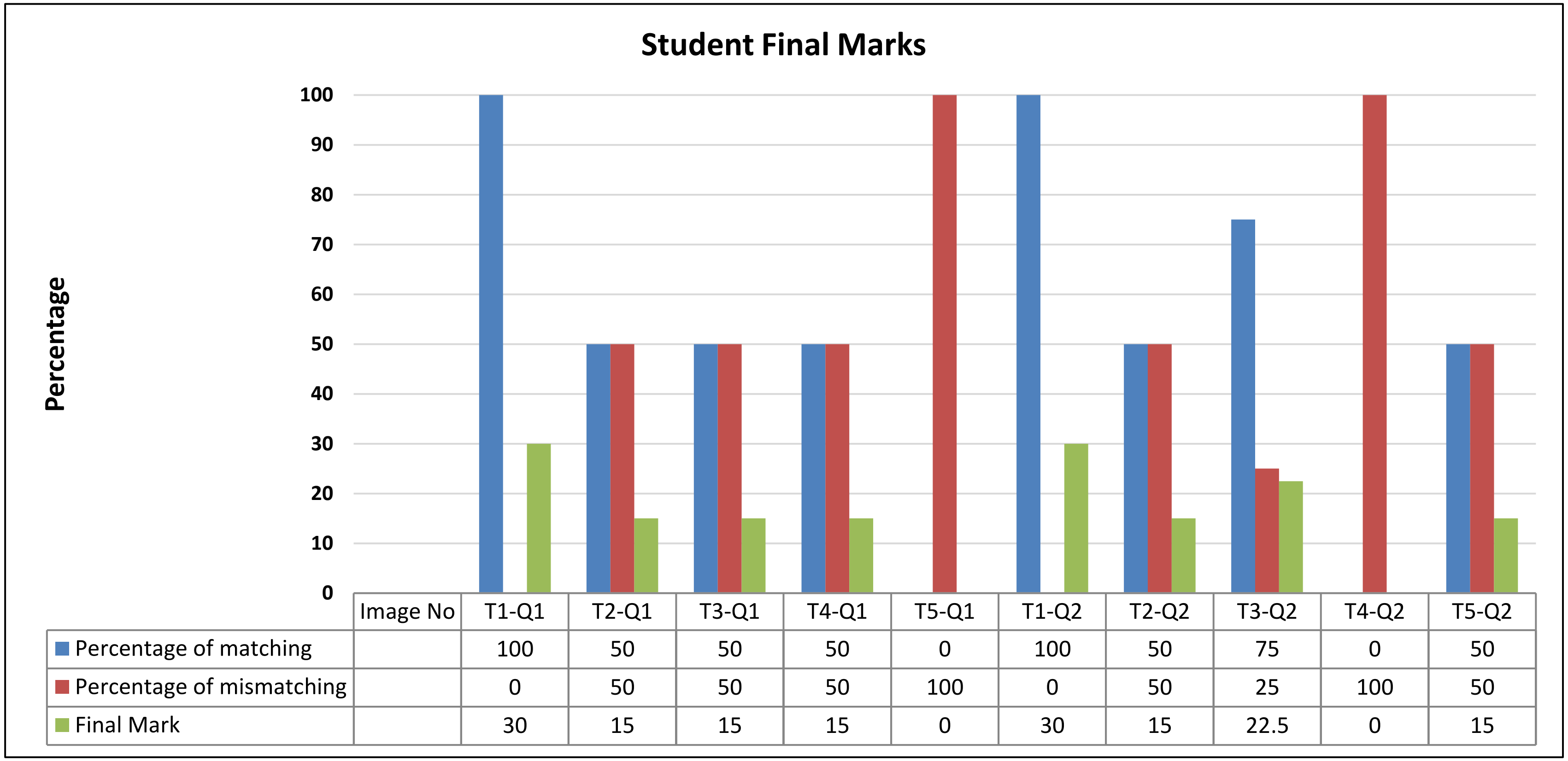


	Image No	T1-Q1	T2-Q1	T3-Q1	T4-Q1	T5-Q1	T1-Q2	T2-Q2	T3-Q2	T4-Q2	T5-Q2
Percentage of matching		100	50	50	50	0	100	50	75	0	50
Percentage of mismatching		0	50	50	50	100	0	50	25	100	50
Final Mark		30	15	15	15	0	30	15	22.5	0	15

Figure 5-14: Student final marks.

Figure 5.14 shows the final marks of the students using the OCR system. The marking system was set up as original image (O) against the Target image number. T1 Q1 and T1 Q2 match 100% the original image and those students achieve full marks in comparison with the other target images.

5.3. Results discussion

During the experimentation and testing phase, the iMarking® system was tested on a sample of 100 respondents, with regard to their images answers. All images were tested systematically as they were uploaded into the system. The experimental results indicate that the designed system was successfully tested on a sample of original teacher answers, which were compared with an original file containing student answers. The results are illustrated in appendix A, showing the comparison between different images and the percentage of matching and mismatching, with a final mark of 30.

The purpose of OCR is to provide reliability, effectiveness, efficiency and a flawless approach, which offers rapid automatic recognition feedback on big volumes of student assignments. This study is aimed at analysing the performance of the image OCR. The analysis started by examining the demographic distribution of the respondents; the age distribution was the first item to be investigated.

From the questionnaire data analysis results, table 5.1 reveals that the youngest respondent in this study is 21 years old, while the oldest respondent is 49 years with a mean age of 34.53 and a standard deviation of 10.035. Details on age ranges are given in Figure 5.1, presenting the percentages of age distribution. Figure 5.2 shows the gender distribution of the respondents: ten males and five females participated in this study. Bonasio (2017) notes common agreement that nurturing diversity is fundamental to innovation in research, and gender equality is important in accomplishing this objective.

Serious concerns associated with gender difference and bias must be considered in studies, in order to support a data-informed methodology for adopting interventions and policies regarding gender inequality. The author adds that many debates surrounding gender disparity are motivated by experience and speculation (Bonasio, 2017). Though this is a good point to begin the argument, there is a gap in the literature that makes it challenging to move to operational approaches and policy. The current study fills this gap by focusing on gender-based data, leading to an empirical understanding that is an essential part of informed decision-making. Development is ongoing, although irregular, which means women are not motivated to study science and technology, especially computer sciences and programming, as they are beginning to pick up the pace of progress. The provided data can be utilised as a foundation for research pioneers, stakeholders in government and organisations involved in financing and legislation, more extensively.

Figure 5.3 shows that 6.7% of the respondents are somewhat satisfied with the automatic image marking system, while 93.3% are very satisfied. This confirms the importance and major contribution of the proposed method in terms of convenience, efficiency and innovation. The researcher then sought to determine whether the image marking system had any impact on the students who adopted it. Figure 5.4 shows that all the respondents agree that the adoption of automatic image marking would improve abilities and skills. The study then compared the results among the gender groups. As can be seen in table 5.2, 100% of both females and males share the same view, that automatic image marking systems would help improve the learner's abilities and skills. The study then sought to establish the difference in opinion by both genders.

The researcher conducted a chi-square test (see Table 5.3 for the results). Figure 5.5 shows that 26.7% of the respondents report that the automatic image system is fair and reliable, 33.3% state that the system is ineffective, while for the remaining 40%, the system is very effective. Figures 5.6 and 5.7 show that 100% of the respondents report that the automatic image marking system is consistent and reliable.

The respondents were then asked to rate the functionality features of the OCR. A summary of the frequencies is given in Table 5.4. Mean scores show that the respondents rate the recognition speed of the system as very poor (2.67), poor (3.73), acceptable (3.33), good (2.80) and, finally, excellent (3.47). Regarding the complexity of the implementation, the descriptive statistics show a mean of 3.73. The respondents also rate the system's speed of pattern recognition with a mean of 2.67. Consequently, the biasness of the OCR system is rated at a mean of 3.47. Figure 25 shows that 100% of the respondents indicate a strong positive correlation between the similarity match and the marks produced by the system.

From figure 5.9, it can be seen that 100% of the respondents believe that image quality would greatly impact the quality of recognition. The study further sought to establish those features in the essay transcriptions that are significant for image marking. Twenty percent of the respondents report that the number of words is most significant, for 46.7%, it is the number of sentences, and, finally, 33.3% cite the length of the essay as the most significant in the process of marking and awarding marks.

Figure 5.10 presents the significant features in essay transcriptions considered during image marking. Upon submission of the image, the text content of the image is extracted line-by-line and compared with the teacher's optimal answers. Moreover, Figure 5.3 proves that 6.7% of the respondents are somewhat satisfied with the automatic image marking system, while 93.3% are very satisfied. The majority of the respondents agree that the adoption of automatic image marking would improve the abilities and skills of the students compared with 17% who felt otherwise.

The significant features considered in essay transcriptions during image marking, as shown in Figure 5.10, indicate that the number of sentences is the principal determinant with 46.7%. One of the methods used in this study is the calculation of the final mark and percentage of matching for each line, which is a necessity as the original image contains strings only. The second method calculates the final mark and percentage of matching for each line. Combining these two methods of calculation enhances the recognition capability of the OCR system, thereby leading to better outcomes than those reported in previous studies.

According to Choudhary, Sharma and Kumar (2015), in their study of the OCR technique, specifically, the character deformation and touching problems that normally arise when performing high-speed printing processes in the machine vision industry, it is challenging for OCR systems to perform segmentation and recognition of characters properly. The solution to this problem comes from the graph theory, which is a good technique for resilience to deformation and touching effects. This theory applies the techniques of graph theory and Dijkstra's shortest path algorithm, with the aim of reducing training time. Additionally, the OCR method explored in this study takes a shorter time compared with previous studies. It segments characters easily and exceptionally, recognising all segments and merging regions, while obtaining the required optimal segments using the graph theory.

This work has made a notable contribution to refining the performance of OCR technique using the method applied in the present research, which has managed to identify the special contents from the image files, while recognising both numbers and text content. The results, showing percentage matching scenarios, can be seen in appendix A. These are, arguably, excellent results in comparison with previous studies, and contribute greatly to the field of computer science.

In relation to Table 5.4, the approval ratings were measured with respect to the effectiveness of the image marking process in analysing student scripts and providing feedback. The results show that for 33.3% of the respondents, the method is ineffective, 26.7% rate it as fair, while 40% confirm that the method is very effective.

Simple random sampling refers to an approach applied in scrapping a smaller sample size from a bigger population to use for research purposes. This helps in making generalities concerning the larger groups, allowing the smaller group of students extracted from a larger group of programming students to be characterised deductively. The method is one of the many methods applied by most statisticians and scientific researchers in the extraction of a smaller sample size from a larger general target population. In comparison with other sampling methods such as the stratified random sampling and probability sampling, the simple random sample has several advantages that outweigh possible limitations in the current research. MacNulty, Tallian, Stahler and Smith (2014) emphasise that using a smaller sample size of a group in random simple sampling increases the success rate of a study.

The benefits of a simple random sample are simplicity, ease of use and high accuracy rates in terms of representing a larger population. According to Elsayir (2014), simple random sampling has high precision and accuracy, enabling researchers to generate a simple random sample through extraction from the comprehensive list of a larger population. After the extraction, the sample is then randomly selected, and a particular number of people are included in the sample. Using a simple random sample allows all participants in the larger population to have equal chance of being selected by the method. The benefits of simplicity in usage and accuracy of representation means that there is no necessity to split the sample group into sub-groups or take measures besides the extraction of the number of research subjects required at random from the larger population. In addition, the only necessities are that the selection exercise is random and every member of the bigger group has an equal probability of selection.

Selecting students randomly from a larger population of programming students also leads to a sample size that is characteristic of the group being investigated in the current study. A sample size of 100 is small and can show low sampling error when simple random sampling is applied appropriately. In any type of research conducted on a particular target population, applying a representative sample to make inferences, interpretations and generalisations concerning the larger group is imperative. It should be noted that a biased sample can result in drawing incorrect conclusions about the larger population. This method is a milestone in the field of computing, adding value to the existing literature and body of knowledge for OCR systems in computer programming. The matching similarity and mismatching percentage calculation in this research are based on the content type of the submitted programming assignments. The contents of the answers are further divided into two categories: Table 5.5 presents the calculation of pure numbers or text, and Table 5.6 illustrates the calculation of a mixture of both digits and text. The high matching similarity percentage indicates the high accuracy of student answers compared with optimal answers provided by the teacher. The mismatching percentage can also drive from result matching similarity. The following subsections explain the calculation of matching similarity for each category.

The achievement of 100% matching in this research can be supported by the theory of performance scalability in a previous study, in which the matching helped in handling the performance scalability of such high-accuracy systems as OCR (Pillai, 2012). In this article, the author investigates possible methods for improving performance for large databases by bringing together two major concepts: counting the shape perspectives and pyramid matching kernels. This results in the fast retrieval of objects being compared, improvement of their recognition and general classification. The method used in this study increases the efficiency and effectiveness of assignments marking for programming teachers.

The capacity to launch instantly a search criterion for every image uploaded by the students and to cross-reference with the teacher's answers is immensely useful. In particular, the approach helps in a learning environment where a large volume of scanned files needs to be handled, and there is a high document inflow from all the students of the Department of Computer Science and programming classes. Students can also now use the instant feedback feature for the documents retrieved from the online marking interface, rather than waiting for the comments from the lecturer/teacher, which would take much longer than the system. This OCR method is quick and accurate, making sure that the uploaded document's content remains intact while saving time too. When integrated with other technologies like scanning and file compression techniques, the benefits of OCR are further demonstrated. The rate of workflow is also increased since teachers no longer have to waste time on manually marking the programming assignments for large numbers of students; thus, they can work faster and more effectively.

5.4. Evaluation

This research is very critical to the current implementation of automated assessment of programming assignments and any future developments. One contribution is to provide a novel evaluation metric to calculate the matching similarity and mismatching percentage of the submitted student answers when compared with the optimal answers. The concepts in this project are also quite different to studies published previously dealing with a similar or related subject, e.g. the use of optical handwriting recognition (OHR). In addition to OHR, the study examines two other concepts dealing with reading comprehension studies, as well as automatic essay scoring. It also integrates three theories of handwriting recognition, automatic essay scoring (abbreviated as A.E.S) and reading comprehension. The concepts of OCR are outlined, and a system for automatic image marking proposed using the novel evaluation metric, which demonstrates characteristics of OHR and OCR.

Table 5-10: Comparison of three main systems.

Article	System Used	Contribution
Ahmadzadeh, Namvar and Soltani (2011)	JavaMarker	This system checks the style of the code used within the assignment and produces messages when a better style is expected. In other instances, penalty marks are considered for inappropriate code style. For this system to make an educational contribution, the designer allows students to submit assignments more than once.
Guskey (2010)	Two-part Marking System	This system assists parents in understanding the reporting from every marking period. It also increases their concerns regarding what seems like low grades and allows them to know whether their children are academically developing at a suitable rate.
Symeonidis (2006)	Computer-Based Assessment (CBA) System	This system helps in the examination of the viability of automatically assessing Java-based programming coursework. The CBA system has the advantage of giving students practical benefits.

The novel evaluation metric of the system proposed in this work is based on an experimental approach, where a deep investigation and advanced analysis of the sample questions are performed to ensure the robustness of the designed evaluation metric in calculating matching similarity and mismatching percentage is achieved. The experimental approach adopted is fundamental to the designed novel evaluation metric. The experimental approach achieves this through the performance of a deep investigation, as well as an advanced analysis of the sample questions. This facilitates the robustness of the designed novel evaluation metric.

However, by comparing the methods utilised in this study with those proposed by Cheang, Kurnia, Lim and Oon (2003), major similarities can be observed. For instance, in terms of similarity, the authors adopt an experimental form of approach that is not optimal for use. In the experiment undertaken, eleven programming-related assignments are handed to the students. The authors only use eleven image files, but the current research uses 100 images, highlighting that the results reported here are stronger than those in any previous study. This is indicative of the scholarly credibility of present study.

Primarily, Cheang, Kurnia, Lim and Oon (2003) use the experimental approach so that "when students submit their program, the system reveals to the students the results on each test case". Their study proposes and, subsequently, implements a model that best acts as an automated generic grader (Cheang, Kurnia, Lim, and Oon, 2003). As such, it successfully creates an automated generic grader model, whereas this study proposes a system for automatic image marking based on OCR.

The 2008 study conducted by Suleman utilises an approach that is quite distinctive from the method utilised in this study. Suleman's fundamental intention is to provide an evaluation for an automated marking system, by utilising the method of comparative analysis. The "output from programs was compared to predefined output to determine marks for submissions" (Suleman, 2008). The method adopted by Suleman aids in the provision of a report regarding the design of the automated marking system, which is in contrast to what is achieved by the proposed system design in the present work.

Conversely, the method adopted by Naudé, Jean and Dieter (2010) in their study resembles that of Suleman's, but not that proposed in the present study. This points to the fact that they also seek to present their automatic marking system. The researchers posit that "our approach is different in that we seek to assess program source code directly, by comparing the structure of submissions" (Naudé, Jean and Dieter, 2010). Thus, the approach that they adopt enables them to make implicit recognition with regard to the similarities between distinct figures present in the submitted assignments and those in previous works (Naudé, Jean and Dieter, 2010). Their study assesses program source codes directly by comparing the submissions' structure.

Similarly, the theories applied in this study may vary when compared with other studies dealing with the same topic or those that are closely linked to the subject matter. For instance, the Halstead theory is used in Al-Ja'afer and Sabri's 2005 study. The Halstead theory is known widely due to its exemplary benefit when utilised in the measurement of the complexities of different software (Al-Jáafer and Sabri, 2005). In the Halstead theory, "software science defines additional metrics such as program vocabulary, program length, program level, program size, and program and data difficulty" (Al-Jáafer and Sabri, 2005). Furthermore, "as these factors increase, the complexity of the program increases and the mark decreases" (Al-Jáafer and Sabri, 2005). The proposed system in this study mainly utilises a mathematical model for calculating matching and mismatch percentages for OCR, as explained below.

Testing of New Models

Figure 5.15 presents the original file (optimal answer), whereas Figures 5.16, 5.17 and 5.18 display the target files (submitted answers). It can be seen that there is a need for marking the target files and clarification on how to calculate the final mark mathematically, as well as the percentage of matching in target files by using category 1.

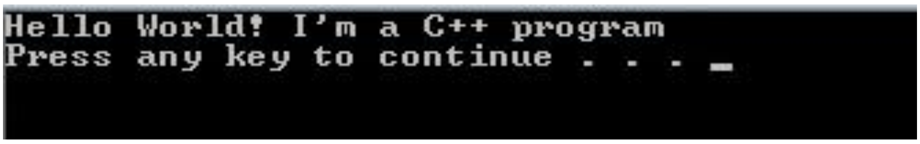


Figure 5-15: The original file.

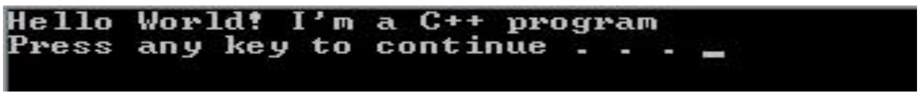


Figure 5-16: The target file (submitted image T1).

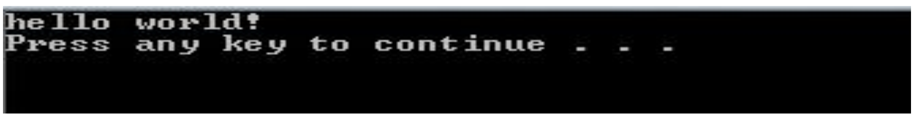


Figure 5-17: Target file (submitted image T2).

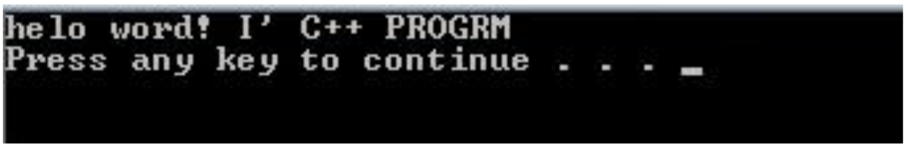


Figure 5-18: The target file (submitted image T3).

Applying the Two Categories:

I. Method 1

In order to calculate the final mark and percentage of matching for T1, method 1 should be applied because the original image consists of only strings:

Numbers of lines in an original file (TLO)=1

Grade for each line $X = \frac{FM}{TLO} = \frac{30}{1}$ =30 $\therefore themarkforeachlineis 30$

Grade for the line which contains (Strings)(G) =

$$\frac{X}{NWO} \times NWT = \frac{30}{6} \times 6 = 30$$

To calculate the summation of final mark for target file (T1):

$FM = \sum_{k=1}^{n} G_K$ ∴ FM for T1=30

To calculate the percentage of matching for each line in (T1):

Percentage of matching for each line P $= \frac{1}{TLO} \times 100$

P$= \frac{1}{1} \times 100$ =100

To calculate the percentage of matching for each line:

MPL $= \frac{NILT}{NILO} \times P$

MPL $= \frac{6}{6} \times 100 = 100\%$

$TPM = \sum_{k=1}^{n} MPL\, k$ where TPM=100%

$TPMIS = 100 - TPM$ where TPM=100-0=100%

Now, we need to calculate the mark for T2 by using category 1:

TLO=1

x $= \frac{30}{1}$ =30 $\therefore themarkforeachlineis 30$

G $= \frac{X}{NWO} \times NWT = \frac{30}{6} \times 2 = 10$

$FM = \sum_{k=1}^{n} G_K$ ∴ FM for T2=10

P $= \frac{1}{TLO} \times 100$

P$= \frac{1}{1} \times 100$ =100%

MPL $= \frac{NILT}{NILO} \times P$

MPL $= \frac{2}{6} \times 100 = 33.33\%$

$TPM = \sum_{k=1}^{n} MPL\, k$ where TPM=33.33%

$TPMIS = 100 - TPM$ where TPM=100-33.33=66.67%

Then, we compute the mark for T3 by using category 1:

TLO=1

x $= \frac{30}{1}$ =30 $\therefore themarkforeachlineis 30$

G $= \frac{X}{NWO} \times NWT = \frac{30}{6} \times 0 = 0$

$FM = \sum_{k=1}^{n} G_K$ ∴ FM for T3=0

P $= \frac{1}{TLO} \times 100$

P$= \frac{1}{1} \times 100$ =100%

$\text{MPL} = \frac{NILT}{NILO} \times P$

$\text{MPL} = \frac{0}{6} \times 100 = 0\%$

$TPM = \sum_{k=1}^{n} MPL\,k$ where TPM=0%

$TPMIS = 100 - TPM$ where TPM=100-0=0%

II. Method 2

To calculate the final mark and percentage of matching for T1, method 2 should be applied as the original image contains numbers and strings:

Figure 5-19: The original file (optimal answer).

Figure 5-20: Target file (submitted answer).

Numbers of lines in an original file (TLO)=2

$\text{X} = \frac{FM}{TLO} \frac{30}{2} = 15$

$\text{G1} = \frac{X}{2} \frac{15}{2} = 7.5$

Numbers and strings=7.5 Mark for each

Calculates the grade for line(1)

$SP = \frac{G1}{NISO} \times NIST = \frac{7.5}{2} \times 2 = 7.5$

$NP = \frac{G1}{NINO} \times NINT = \frac{7.5}{1} \times 1 = 7.5$

$FML_1 = SP + NP = 7.5 + 7.5 = 15$

Calculates the grade for line(2)

$SP = \frac{G1}{NISO} \times NIST = \frac{7.5}{2} \times 2 = 7.5$

$NP = \frac{G1}{NINO} \times NINT = \frac{7.5}{1} \times 0 = 0$

$FML_2 = SP + NP = 7.5 + 0 = 7.5$

$FM = FML_1 + FML_2 = 15 + 7.5 = 22.5$

To calculate the percentage of matching for line (1):

P $= \frac{1}{TLO} \times 100 = \frac{1}{2} \times 100 =$ 50%

PL $= \frac{P}{2} = \frac{50}{2} = 25\%$

PS$= \frac{NIST}{NISO} \times PL = \frac{2}{2} \times 25$=25%

PN$= \frac{NINT}{NINO} \times PL = \frac{1}{1} \times 25 = 25\%$

$MPL_1 = PS + PN$=25+25=50%

To calculate the percentage of matching for line (2):

P $= \frac{1}{TLO} \times 100 = \frac{1}{2} \times 100 =$ 50%

PL $= \frac{P}{2} = \frac{50}{2} = 25\%$

PS$= \frac{NIST}{NISO} \times PL = \frac{2}{2} \times 25$=25%

PN$= \frac{NINT}{NINO} \times PL = \frac{0}{1} \times 25 = 0\%$

$MPL_2 = PS + PN$=25+0=25%

$TPM = \sum_{k=1}^{n} MPL\,k$=50+25=75%

TPMIS=100-TPM=100-75=25%

Note that image quality plays an imperative role in the improvement of the recognition quality. Therefore, an image resolution range of 200-400 DPI is recommended for a better recognition.

The concepts of the present research are also quite different to those in studies previously published dealing with a similar or related subject. For instance, Srihari, Collins, Srihari, Srinivasan, Shetty and Brutt-Griffler (2008) consider the concepts of OHR. In addition, the study examines two other concepts dealing with reading comprehension studies, and automatic essay scoring (Srihari, Collins, Srihari, Srinivasan, Shetty and Brutt-Griffler, 2008). The work presented by these authors integrates three theories, namely, the handwriting recognition, the automatic essay scoring (abbreviated as AES) and the reading comprehension (Srihari, Collins, Srihari, Srinivasan, Shetty and Brutt-Griffler, 2008). In comparison, the study presented here adopts the concepts of OCR, while proposing a system for automatic image marking.

Klobucar, Elliot, Deess, Rudniy and Joshi (2013) also investigate "the use of AES to identify at-risk students enrolled in a first-year university writing course". Their study comprehensively evaluates the application of AES through the adoption of distinctive methodologies. Some of the most prominent methods which they utilise in their study for the purpose of facilitating this evaluation include construct modelling, response processes, disaggregation, extrapolation, generalisation and consequence (Klobucar, Elliot, Deess, Rudniy and Joshi, 2013). The present study, on the other hand, utilises descriptive statistics to analyse and evaluate the OCR data.

Furthermore, the concepts illustrated in Jones' study (2001) are instrumentally based on how testing may be utilised "to specify programming assignments that are amenable to automated grading". The study utilises the SPRAE framework for guidance with regard to its crafting "of a testable specification and the derivation of program checking scripts" (Jones, 2001). A program fundamental for script checking is developed. Alternatively, Higgins, Symeondis and Tsintsifas largely base their 2002 study on exploratory methods, whereas the present study is mainly based on descriptive statistics. It examines the CourseMaster as an assessment system, which is computer-based and explores the coherency, feedback-richness, customisability, reliability, security and extensibility (Higgins, Symeonidis and Tsintsifas, 2002).

Also, this study used a graphical user interface (GUI), which students use to submit their files using the web-based online platforms of their choice. The choice of GUI was based on two key issues: friendliness and responsiveness. On the other hand, Bautista & Comendador (2016) used Tesseract, stating this to be one of the most popular open sources of OCR. In their study, the researchers applied this method based on the fact that it eliminates the difficult and tedious tasks of manually encoding grades. However, it is more focused on shortening the period taken to provide results than on quality. GUI interfaces are also quite efficient and eliminate the long processes associated with the alignment of data to programming options. While Tesseract encodes more than 100 student grades at a time, hence exposing the data to numerous errors, GUI is not text-based, and therefore encounters fewer errors. As a result, data within a GUI are usually quality since a GUI lessens the cognitive loads for all types of users, trained or untrained.

The study by Bautista & Comendador (2016) indicated that the results of data analyses usually depend on the originality of the source documents. Other factors, such as Dots per Inch in scanned images, are generally significant. In a GUI, however, the distinct types of assessment criteria come with the acknowledgment of valuations of all outputs displayed in the image forms in order for the OCR systems to start analysing them. As a result, using interfaces such as Tesseract means that users will be inclined to follow specific approaches, so that results may be verified and confirmed as being credible. However, for this research, those options of OCR which allow for adjustments of automatic feedback have been used. These adjustments demonstrate the flexibility that comes with this choice, hence changing students' behaviours.

Moving on, this research uses a marking process which offers detailed explanations of novel measurements for the computation of matches between student answers, with a mixture of digits and text. The invention by Uthman, Porter, & Ledgerwood (2012) offers a similar OCR platform, but reaches results differently. In the invention, Uthman, Porter, & Ledgerwood (2012) included the generation of expected data and each answer region was located in the images provided. In essence, the invention evaluates responses by comparing them with data of expected answers, rather than generating corresponding evaluation data. In this study, however, a more careful marking process has been simulated. All computations go through twelve steps, where accuracy from the attachments provided are influenced only by the quality of the images. Evidence for this assertion may be found in Tables 5.5 and 5.6, wherein the automated twelve steps provide feedback, which informs the students about what is expected from them.

On face value, there are remarkable similarities between the Adaptive Thresholding and Card Mask from the research by Bhaskar, Lavassar, & Green (2010), and the current study's Automatic Image Marking Process System. Both systems are crucial parts of OCR, and each of their phases demands the application of systemic methodologies and approaches in the retrieval of images. However, one of the reasons why the Automatic Image Marking Process system was recommended for this study is the fact that the results facilitate extractions with 100% similarity. Bhaskar, Lavassar, & Green (2010) reported difficulties when using MATLAB commands to meet up-sampling threshold arrays and match the sizes of images. This study proves, however, that mismatch for this recommendation is almost impossible. As such, there is more simplicity, interactivity, and friendliness in the interface than many other options available from previous research.

5.5. Summary

The research was designed to investigate the response from 15 respondents using the questionnaire. The analysis of the OCR was conducted taking into consideration different elements among the group of respondents. First, the analysis of age distribution and the gender of the respondents showed that 15–45 years was the major age group (53.3%), while the 16–30 and 46–60 years groups were 33.3% and 13.3%, respectively (Figure 5.1). In contrast, the gender distribution results showed that 66.7% of the population sample of the respondents was male (Figure 5.2). The study also investigated the satisfaction percentage of the respondents; 93.3% were very satisfied with the OCR system (Figure 5.3). The system had a 100% improvement impact on the abilities and skills of learners. Furthermore, according to the results, 40% rated the system as very effective, while for 33.3%, it was ineffective. The high percentage of ineffectiveness was due to several reasons; for instance, some of the respondents were not confident of system effectiveness in the image marking process, while others did not understand the objective of the study and had no clear idea about the OCR system. Another possible reason was that respondents were anxious that students may send unclear images despite the originality of the answers. The results showed that the system was efficient; a significant similarity match produced high marks and image quality determined the quality of the recognition for 100% of the respondents (Figures 5.7, 5.8 and 5.9). The system results obtained, while tested using dummy variables provided by some students, helped in the evaluation of the stability, reliability and accuracy of the iMarking® system. The results were compared with the outcomes of other existing benchmarks within the industry.

Chapter 6 Conclusion and Future Works

Because of the rapidly increasing number of students and educational institutions in recent years, as well as advancements in eLearning systems that are used to provide ubiquitous education processes, computer-based assessment approaches have been exponentially used. Students' assignment marking can be performed manually or automatically. Unfortunately, teachers are overburdened with programming courses, because they do not have enough time to create more assessment activities especially for programming assignments. Therefore, an automated marking tool for marking student assignments is urgently required.

There are many automated systems used to assess and to mark students' programming assignments; such marking systems pay more attention to the compilation and run of the student solution (i.e., programming code). Hence, these systems do not consider the code testing attempts by the students, and assessment feedback is not provided to improve Test-Driven Development. Assessing the programming assignments is not a simple task for the teachers; thus, the automated marking systems make it easier. Such development in marking students' programming assignments automatically is a vital for programming classes. These marking systems are more accurate than the traditional marking approaches in terms of error detection. In addition, the automated marking system provides assessment feedback for both student and instructor.

In this research, a proposed OCR based approach for automatic image marking process is presented. The proposed approach has three objectives:

❖ ***To adopt OCR to extract the text from submitted-images student answers***

The first objective is satisfied by adopting OCR to extract the text from the uploaded image (i.e., uploaded assignment) and save the extracted text in a text file. The extracted text will be use as input for the next stage.

❖ ***To propose a novel evaluation metric to provide effective marking and grading for submitted student answers.***

The second objective is satisfied by proposing novel evaluation metric. This evaluation metric is the core contribution and it is used to calculate the matching similarity and mismatching percentage of the submitted student answers when compared with the optimal answers. Further, the study integrates three theories of handwriting recognition, automatic essay scoring and reading comprehension. The novel evaluation metric of the proposed tool is based on an experimental approach, where a deep investigation and advanced analysis on the sample questions are performed to ensure the robustness of the designed evaluation metric in calculating matching similarity and mismatching percentage is achieved. The experimental approach adopted is fundamental since it aids this study in successfully coming up with the designed novel evaluation metric. The experimental approach achieves this through the performance of a deep investigation, as well as advanced analysis on the sample questions, which facilitates the robustness of the designed novel evaluation metric.

- ***To evaluate the proposed approach in terms of the accuracy of marking and grading for submitted student answers.***

The third objective is satisfied. The proposed tool was evaluated using a sample of 100 questions, with the results showing that the system is efficient in automatically marking student answers and can achieve 100% accuracy, which is a major achievement. During the experimentation and testing phase, the iMarking® system was tested on a sample of 100 respondents, with regard to their image answers. All images were tested systematically as they were uploaded into the system. The experimental results indicate that the designed system was successfully tested on a sample of original teacher answers, which were compared with an original file containing student answers. Thus, the proposed system outperformed the manual way of marking students' assignments in terms of processing time. It saved up to 197 seconds for marking a total of ten questions, and this aligns with one of the proposed system's objectives, which is to save teacher's time. The processing time can be further decreased by increasing the specification of the proposed system hardware.

Subsequently, the proposed system can offer a more efficient and accurate way of marking programming assignments, while reducing the time burden on the teachers and lecturers. The efficiency of the proposed system has been experimentally validated. The experimental results showed a matching percentage of 100% between the student answers and those prescribed by the teachers.

It can be concluded that the adoption of technology (such as OCR) to support the education process is significantly valuable, as regards both students and teachers. To this end, the proposed OCR based approach to automate the marking process for better assessment monitoring for computer science programming students frees the teaching monitor to be more creative. In addition, the presentation of the iMarking® system gives alternative support to securing the test environment. However, the possibility of an outcome has been built in and a further edition could be an option.

Future research should be to explore the feasibility of extending the proposed approach to marking and grading other types of assignment instead of focusing only on programming codes assignments. The proposed approach has also opened interesting avenues for future research in malware detection. There are several research directions:

1. filling the small gap faced by the designed system in syntax similarity, and the proposed approach should pay more attention to the problem of word semantic similarity.
2. Enhancing the OCR algorithm/mechanism accuracy so it can extract the text from low resolution images.
3. Improving the proposed novel evaluation metrics to provide comprehensive details about the assessment process.
4. Proposing generic evaluation metrics that can mark/grade assignments other than those for programming codes.

Reference

Ahmadzadeh, M., Namvar, S., & Soltani, M. (2011). JavaMarker: A marking system for Java programs. *International Journal of Computer Applications (0975 – 8887), 20*(2), 15-20.

Aiken, A. (2005). Moss: A system for detecting software plagiarism. University of California–Berkeley. See www. cs. berkeley. edu/aiken/moss. html, 9.

Ala-Mutka, K., & Jarvinen, H. (2004). Assessment Process for Programming Assignments, *In Proceedings of the IEEE international Conference on Advanced Learning Technologies, IEEE Computer Society*, 181-185. IEEE Xplore

Ali, A., Alrasheedi, M., Ouda, A., & Capretz (2014). A study of the interface usability issues of mobile learning applications for smart phones from the user's perspective. *International Journal on Integrating Technology in Education (IJITE), 3*(4), 1-16. DOI :10.5121/ijite.2014.3401

Al-Jáafer, J., & Sabri, K. (2005). Automark++: a case tool to automatically mark student Java programs. *International Arab Journal of Information Technology* 2(1), 87-96.

Al-Oteawi, S. M. (2002). *The perceptions of administrators and teachers in utilizing information technology in instruction, administrative work, technology planning and staff development in Saudi Arabia* (pp. 1-373). Ohio University.

Arica, N., & Yarman-Vural, F. T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems,* Man, and Cybernetics, Part C (Applications and Reviews), 31(2), 216-233.

Arica, N., & Yarman-Vural, F. T. (2002). Optical character recognition for cursive handwriting. *IEEE transactions on pattern analysis and machine intelligence*, 24(6), 801-813.

Arnow, D., & Barshay, O. (1999). On-line programming examinations using Web to teach. *ACM SIGCSE Bulletin*, 31(3), 21-24.

Bach, J. (1999). Test Automation Snake Oil. *In Proceedings of the 14th International Conference and Exposition on Testing Computer Software.* Windows Tech Journal.

Balci, O. (1994, December). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *In Simulation Conference Proceedings*, 1994. Winter (pp. 215-220). IEEE.

Barrett, D., & Kipper, G. (2010). *Virtualization and forensics: A digital forensic investigator's guide to virtual environments*. Amsterdam: Syngress/Elsevier.

Barriocanal, E., Urbán, M., Cuevas, I., & Pérez, P. (2002). An experience in integrating automated unit testing practices in an introductory programming course, *SIGCSE Bull, 34, ACM Press*, 125-128. ACM Beck, K. (2001). Aim, Fire, *IEEE Software, vol. 18, no. 5*, pp.87-89. IEEE Xplore

Bautista, M. M., & Comendador, B. E. V. (2016). Adoption of an Open Source Optical Character Recognition (OCR) for Database Buildup of the Students' Scholastic Records. *International Journal of Information and Electronics Engineering*, 6(3), 206.

Benford, S. D., Burke, E. K., & Foxley, E. (1993). Experiences with the Ceilidh system.

Bennamoun, M., & Mamic, G. J. (2002). Optical Character Recognition. *In Object Recognition* (pp. 199-220). Springer London.

Berner, S., Weber, R., & Keller, R. (2005). Observations and lessons learned from automated testing. *In Proceedings of the 27th international Conference on Software Engineering*, pp. 571-579. ACM

Bhaskar, S., Lavassar, N., & Green, S. (2010). Implementing optical Character recognition on the Android operating System for Business Cards. *EE 368 Digital Image Processing*.

Bhute, A. N., & Meshram, B. B. (2014). Text based approach for indexing and retrieval of image and video: a review. *Advances in Vision Computing: An International Journal*, *1*(1), 27-38.

Bird, D. K. (2009). The use of questionnaires for acquiring information on public perception of natural hazards and risk mitigation – a review of current knowledge. *Natural Hazards and Earth System Sciences*, *9*(1), 1307-1325.

Bloom, B. S., & Committee of College and University Examiners. (1964). Taxonomy of educational objectives (Vol. 2). New York: Longmans, Green.

Borovikov, E. (2004). *A survey of modern optical character recognition techniques.* Retrieved from https://pdfs.semanticscholar.org/79c9/cc90b8c2e2c9c54c3862935ea00df7dd56ed.pdf.

Bourque, P., Dupuis, R., Abran, A., Moore, J. W., & Tripp, L. (1999). The guide to the software engineering body of knowledge. *IEEE software*, 16(6), 35-44.

Breuel, T. M., Ul-Hasan, A., Al-Azawi, M. A., & Shafait, F. (2013, August). High-performance OCR for printed English and Fraktur using LSTM networks. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*(pp. 683-687). IEEE.

Brown, S., Race, P., & Smith, B. (2004). *500 tips on assessment*. Routledge.

Bull, J. (1993). Using Technology to Assess Student Learning.

Bunke, H., & Wang, P. S. (Eds.). (1997). Handbook of character recognition and document image analysis. World scientific.

Casey, R. G., & Lecolinet, E. (1996). A survey of methods and strategies in character segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 18(7), 690-706.

Chandarana, J., & Kapadia, M. (2014). Optical character recognition. *International Journal of Emerging Technology and Advanced Engineering*, 4(5), 219-223.

Charman, D., & Elmes, A. (1998). Computer Based Assessment (Volume 1): A guide to good practice. *SEED (Science Education, Enhancement and Development), University of Plymouth*.

Chaudhuri, A. (2010). Some Experiments on Optical Character Recognition Systems for different Languages using Soft Computing Techniques. Technical Report, Birla Institute of Technology Mesra, Patna Campus, India, 2010. Google Scholar.

Chaudhuri, A., Mandaviya, K., Badelia, P., & Ghosh, S. K. (2017). Optical character recognition systems for different languages with soft computing. *Springer*.

Cheang, B., Kurnia, A., Lim, A., & Oon, W. (2003). On automated grading of programming assignments in an academic institution. *Computers & Education 41*(2), 121-131.

Chen, J., & Zhang, J. (2007). Comparing text-based and graphical user interfaces for novice and expert users. *AMIA Annual Symposium Proceedings Archive, 2007*(2007), 125-129. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655855/.

Cheng, Y., Qu, Y., Shi, H., & Xie, Y. (2010). ID numbers recognition by local similarity voting. *In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference* on (pp. 3881-3888). IEEE.

Choudhary, S., Sharma, S., Kumar, B. (2015). Recognition of printed Oriya script using gradient based features. *India Conference (INDICON) 2015 Annual IEEE,* 1-5. ISSN 2325-9418. DOI: 10.1109/ACPR.2013.193

Choy, M., Lam, S., Poon, C. K., Wang, F. L., Yu, Y. T., & Yuen, L. (2007). *Design and implementation of an automated system for assessment of computer programming assignments.* Paper presented at the International Conference on Web-Based Learning.

Colaco, J., & Borkar, S. (2016). Design and Implementation of Konkani Text to Speech Generation System using OCR Technique. *Imperial Journal of Interdisciplinary Research*, *2*(6).

Conole, G., & Warburton, B. (2005). A review of computer-assisted assessment. ALT-J, 13(1), 17-31.

Dasarathy, B., Sullivan, K., Schmidt, D. C., Fisher, D. H., & Porter, A. (2014, May). The past, present, and future of MOOCs and their relevance to software engineering. *In Proceedings of the on Future of Software Engineering* (pp. 212-224). ACM.

David, J. (2005). Web-based Grading: Further Experiences and Student Attitudes, *In Proceedings of ASEE/IEEE Frontiers in Education Conference, IEEE*, p 18-23. IEEE Xplore.

Dörner, C. (2009). *Tailoring software infrastructures: Integration of end-user development and service-oriented architectures*. Lohmar: Eul.

Doube, W. (2000). Distance teaching workloads. ACM SIGCSE Bulletin, 32(1), 347-351.

Douce, C., Livingstone, D., & Orwell, J. (2005). *Automatic test-based assessment of programming: A review, Journal on Educational Resources in Computing*. ACM

Dunlap, I. H. (2006). *Open source database driven Web development: A guide for information professionals*. Oxford: Chandos.

Dustin, E., Rashka, J., & Paul, J. (1999). *Automated software testing: introduction, management, and performance*. Addison-Wesley Professional.

Edwards, S. (2003). Improving student performance by evaluating how well students test their own programs *J. Educ. Resour. Comput*. 3. ACM

Edwards, S. (2004). Using software testing to move students from trial-and-error to reflection-in-action, *In Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, ACM Press*, pp.26-30, ACM

Edwards, S. H., & Perez-Quinones, M. A. (2008, June). Web-CAT: automatically grading programming assignments. *In ACM SIGCSE Bulletin* (Vol. 40, No. 3, pp. 328-328). ACM.

Elsayir, H. A. (2014). Comparison of precision of systematic sampling with some other probability samplings. *American Journal of Theoretical and Applied Statistics, 3*(4), 111-116. doi: 10.11648/j.ajtas.20140304.16

Everett, G. D., & McLeod Jr, R. (2007). Software testing: testing across the entire software development life cycle.

Falkner, N., Vivian, R., Piper, D., & Falkner, K. (2014). Increasing the effectiveness of automated assessment by increasing marking granularity and feedback units [Abstract]. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education, 1*(14), 9-14. doi>10.1145/2538862.2538896

Fangohr, H., O'Brien, N., Prabhakar, A., & Kashyap, A. (2015). *Teaching Python programming with automatic assessment and feedback provision.* Retrieved from https://arxiv.org/pdf/1509.03556.pdf

Formanek, M., & Zaborsky, M. (2017). Web interface security vulnerabilities of selected European open-access academic repositories. *Liber Quarterly, 27*(1), 45-57.

Gibson, D., & Baek, Y. (2009). *Digital simulations for improving education: Learning through artificial teaching environments*. Hershey, Pa: IGI Global.

Guskey, T. R. (2010). *Grading and reporting student learning.* Retrieved from https://www.ceesa.org/phocadownload/handouts2011/guskeykeynotegradingandreporting.pdf

Hall, P. A. (2006). Systematic process analysis: when and how to use it. *European Management Review, 3*(1), 24-31. doi: 10.1057/palgrave.emr.1500050

Harris, J., Adams, E., & Harris, N. (2004). Making program grading easier: but not totally automatic J. *Comput. Small Coll.* 20, 248-261. ACM 41

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research, 77*(1) 81–112. DOI: 10.3102/003465430298487

Hepplestone, S., & Mather, R. (2007). Meeting rising student expectations of online assignment submission and online feedback. In *Proceedings of the 11th International Computer-Assisted Assessment Conference 2007*, 10-11 July. Learning and Teaching Development, Loughborough University, Loughborough, UK.

Hetzel, W. C., & Hetzel, B. (1991). *The complete guide to software testing*: John Wiley & Sons, Inc.

Higgins, C., Symeonidis, P., & Tsintsifas, A. (2002). The marking system for CourseMaster. *ACM SIGCSE Bulletin 34*(3), 46-50.

Hossain, M. Z., Amin, M. A., & Yan, H. (2012). *Rapid feature extraction for optical character recognition*. Retrieved from https://arxiv.org/pdf/1206.0238.pdf

IEEE (2003). The analysis and design of architecture systems for speech recognition on modern handheld-computing devices. *Conference Proceedings.* DOI: 10.1109/CODESS.2003.1275257

Isaacson, P., & Scott, T. (1989). Automating the execution of student programs *SIGCSE Bull. 21*, 2, 15-22. ACM

Islam, N., Islam, Z., & Noor, N. (2017). A survey on optical character recognition system. *Journal of Information & Communication Technology _ JICT, 10*(2), 1-4.

Isong, J. (2001). *Developing an automated program checkers.* Paper presented at the Journal of Computing Sciences in Colleges.

Jackson, D., & Usher, M. (1997, March). Grading student programs using ASSYST. *In ACM SIGCSE Bulletin* (Vol. 29, No. 1, pp. 335-339). ACM.

Janzen, D., & Saiedian, H. (2006). Test-driven learning: intrinsic integration of testing into the CS/SE curriculum, *In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, ACM Press*, pp.254-258. ACM

John Wiley & Sons.Francis, T. S., & Jutamulia, S. (Eds.). (1998). Optical pattern recognition. Cambridge University Press.

Jones, E. L. (2001). Grading student programs-a software testing approach. *Journal of Computing Sciences in Colleges 16*(2), 85-192.

Jones, E. L. (2001a). *An experiential approach to incorporating software testing into the computer science curriculum.* Paper presented at the Frontiers in Education Conference, 2001. 31st Annual.

Jones, E. L. (2001b). Grading student programs-a software testing approach. *Journal of Computing Sciences in Colleges, 16*(2), 185-192.

Kastelan, I., Kukolj, S., Pekovic, V., Marinkovic, V. & Marceta, Z. (2012). Extraction of text on TV screen using optical character recognition. *Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium*, 153-156.

Klobucar, A., Elliot, N., Deess, P., Rudniy, O., & Joshi, K. (2013). Automated scoring in context: rapid assessment for placed students. *Assessing Writing 18*(1), 62-84.

Korel, B., & Al-Yami, A. M. (1998). Automated regression test generation. *ACM SIGSOFT Software Engineering Notes*, *23*(2), 143-152.

Kovacic, Z., & Green, J. S. (2010). Automated assessment of advanced office skills.

Krutz, R. L., & Vines, R. D. (2010). Cloud security: A comprehensive guide to secure cloud computing. Wiley Publishing.

Kurnia, A., Lim, A., & Cheang, B. (2001). Online judge. Computers & Education, 36(4), 299-315.

Laurillard, D. (2013). Teaching as a design science: Building pedagogical pattern for learning and technology: Routledge.

Li, K., & Wu, M. (2004). *Effective GUI Testing Automation: Developing an Automated GUI Testing Tool*. John Wiley & Sons.

Link, J. (2003). *Unit Testing in Java: How Tests Drive the Code*, Morgan Kaufmann.

Long, H. (2014). An empirical review of research methodologies and methods in creativity studies (2003-2012). *Creativity Research Journal, 26*(4), 427-438. https://doi.org/10.1080/10400419.2014.961781

Ludwig, R., Mauser, W., Niemeyer, S., Colgan, A., Stolz, R., Escher-Vetter, H., & Ludwig (2003). Web-based modelling of energy, water and matter fluxes to support decision making in mesoscale catchments—the integrative perspective of GLOWA-Danube. *Physics and chemistry of the Earth*, *Parts A/B/C*, *28*(14), 621-634

Machera, R. P. (2017). The relevance of classroom audits and student feedback on teacher effectiveness. *Universal Journal of Educational Research*, *5*(1), 54-60. DOI: 10.13189/ujer.2017.050107

MacNulty, D. R., Tallian, A., Stahler, D. R., & Smith, D. W. (2014). Influence of group size on the success of the wolves hunting Bison. *Public Library of Science One.* Retrieved from https://doi.org/10.1371/journal.pone.0112884

Makhoul, J. I., & Schwartz, R. M. (1999). U.S. Patent No. 5,933,525. Washington, DC: U.S. Patent and Trademark Office.

Mangano, A. (2006). Automatic paper grading and student progress tracking system. U.S. Patent Application No. 11/129,973.

Masbah, A. and Lu, J. (2016). Automatic Image Marking Process. INFOCOMP 2016: *The Sixth International Conference on Advanced Communications and Computation*.

Masbah, A. and Lu, J. (2018). The Development of Automatic Image Marking System towards Better Assessment Monitoring for Computer Science Programming Students: *Egyptian Computer Science Journal 42, 2, p. 44-54 11 p., 4.*

Mason, D. V., & Woit, D. M. (1999, March). Providing mark-up and feedback to students with online marking. *In ACM SIGCSE Bulletin* (Vol. 31, No. 1, pp. 3-6). ACM.

Massol, V., & Husted, T. (2003). *J Unit in Action*. Manning Publications.

Matthews, K., Janicki, T., He, L., & Patterson, L. (2012). Implementation of an automated grading system with an adaptive learning component to affect student feedback and response time. *Journal of Information Systems Education*, 23(1), 71.

McGee, P., & Kaner, C. (2004). Experiments with high volume test automation, *SIGSOFT Soft. Eng. Notes* 29, 5, pp.1-3. ACM

Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International Journal of Recent Technology and Engineering*, 2(1), 72-75.

Myers, G. J. (1976). *Software Reliability*. John Wiley & Sons, Inc..Myers, G. (1976). *The Art of Software Testing*, John Wiley & Sons.

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*: John Wiley & Sons.

Naudé, K. A., Jean H. G., & Dieter, V. (2010). Marking student programs using graph similarity. *Computers & Education 54*(2), 545-561.

Nguyen, A., Piech, C., Huang, J., & Guibas, L. (2014, April). Codewebs: scalable homework search for massive open online programming courses. *In Proceedings of the 23rd international conference on World wide web* (pp. 491-502). ACM.

Niblack, C. W., Barber, R., Equitz, W., Flickner, M. D., Glasman, E. H., Petkovic, D., Taubin, G. (1993). *QBIC project: querying images by content, using color, texture, and shape.* Paper presented at the IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology.

Niemeyer, G., & Poteet, J. (2003). Extreme Programming with Ant: Building and Deploying Java Applications with JSP, EJB, XSLT, XDoclet, and J Unit. *Sams*. HTML.

Nirmala, P. N. (2010). Text extraction in complex colour document images for enhanced readability. *Intelligent Information Management, 2,* 120-133. doi:10.4236/iim.2010.22015

Och, F. J., Popat, A. C., Genzel, D., & Jahr, M. E. (2015). U.S. Patent No. 8,953,885. Washington, DC: U.S. Patent and Trademark Office.

Oulasvirta, A., & Abowd, G. D. (2016). User interface design in the 21st century. *Computer, 49*(7), 11-13. doi: 10.1109/MC.2016.201

Ozden, M. Y. (2005). Students' perceptions of online assessment: A case study. *International Journal of E-Learning & Distance Education*, *19*(2), 77-92.

Parkin, H., & Thorpe, L. (2009). Exploring student experiences of e-learning: A phenomenographic approach. *In Bera Annual Conference (September 2-5, 2009). Manchester, UK: University of Manchester*.

Parsons, David. (2012). *Foundational Java: Key Elements and Practical Programming*. Springer-Verlag New York Inc.

Pentland, A. P., Picard, R. W., & Scarloff, S. (1994). *Photobook: Tools for content-based manipulation of image databases.* Paper presented at the IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology.

Pillai, S. (2012). *Shape context matching for efficient OCR.* Retrieved from https://people.csail.mit.edu/spillai/data/papers/scocr-project-paper.pdf

Pressman, R. S. (2000). Software Engineering A Practitioner's Approach. McGraw Hill Higher Education.

Preston, J. A., & Shackelford, R. (1998). A system for improving distance and large-scale classes. ACM SIGCSE Bulletin, 30(3), 193-198.

Qadri, M. T., & Asif, M. (2009, April). Automatic number plate recognition system for vehicle identification using optical character recognition. In Education Technology and Computer, 2009. ICETC'09. International Conference on (pp. 335-338). IEEE.

Rangarajan, K., Swaminathan, N., Hegde, V., & Jacob, J. (2001). Product quality framework: a vehicle for focusing on product quality goals. ACM SIGSOFT Software Engineering Notes, 26(4), 77-82.

Reek, K. (1989). The TRY system or how to avoid testing student programs, *In Proceedings of the Twentieth SIGCSE Technical Symposium on Computer Science Education, ACM Press*, 112-116. ACM.

Reshma, A. J., James, J. J., Kavya, M., & Saravanan, M. (2016). An overview of character recognition focused on off-line handwriting. *ARPN Journal of Engineering and Applied Sciences*, 11(15), 9372-9378.

Ross, S. M., Marrison, G. R., & Lowther, D. L. (2005). Using experimentation methods in higher education research. *Journal of Computing in Higher Education, 16*(2), 39-64.

Rui, Y., Huang, T. S., & Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, *10*(1), 39-62.

Severdia, R., & Crowder, K. (2010). *Using Joomla*. Sebastopol, Calif: O'Reilly Media.

Sheard, J., Dick, M., Markham, S., Macdonald, I., & Walsh, M. (2002). Cheating and plagiarism: perceptions and practices of first year IT students, SIGCSE Bull, 183-187. ACM.

Shepard, T., Lamb, M., & Kelly, D. (2001). More testing should be taught, *Commun. ACM 44*, pp.103-108. ACM.

Shoham, Florenthal, Rose., & Kropp (1998). Differences in value importance: the impact of age and gender in the Israeli population. In *NA - Advances in Consumer Research Volume 25*, 468-474, eds. Joseph W. Alba & J. Wesley Hutchinson, Provo, UT: Association for Consumer Research.

Snyder, R. (2004). Teacher specification and student implementation of a unit testing methodology in an introductory programming course, *J. Comput. Small Coll*. pp.22-32. ACM.

Sommerville, I. (2007). Software Engineering (8th ed.). Harlow, Essex: Addison Wesley. United States of America.

Springmann, U., Najock, D., Morgenroth, H., Schmid, H., Gotscharek, A., & Fink, F. (2014, May). OCR of historical printings of Latin texts: problems, prospects, progress. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage* (pp. 71-75). ACM.

Srihari, S., Collins, J., Srihari, R., Srinivasan, H., Shetty, S., & Brutt-Griffler, J. (2008). Automatic scoring of short handwritten essays in reading comprehension tests. *Artificial Intelligence 172*, 300-324.

Srivastava, A. (2002). Test Automation and Software Development. *Technology Review, 7*, 2002.

Stephen, H. (2003). Teaching software testing: automatic grading meets test-first coding, *In Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM Press*, pp.318-319. ACM.

Suleman, H. (2008). Automatic marking with Sakai." In Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology, 229-236. ACM.

Summons, P., Coldwell, J., Bruff, C., & Henskens, F. (1997, July). Automated assessment and marking of spreadsheet concepts. *In Proceedings of the 2nd Australasian conference on Computer science education* (pp. 178-184). ACM.

Symeonidis, P. (2006). *Automatic assessment of Java programming coursework for Computer Science education.* Retrieved from http://www.cs.nott.ac.uk/~pszcah/pdf/thesis-PavlosSymeonidis.pdf

Thummalapenta, S., Devaki, P., Sinha, S., Chandra, S., Gnanasundaram, S., Nagaraj, D. D., & Sathishkumar, S. (2013, May). Efficient and change-resilient test automation: An industrial case study. In *Proceedings of the 2013 International Conference on Software Engineering* (pp. 1002-1011). IEEE Press.

Tuller, C., & Oblinger, D. (1998). Information technology as a transformation agent. *CAUSE EFFECT*, *20*, 33-45.

Uthman, T. M., Porter, F., & Ledgerwood, A. D. (2012). System and method for using optical character recognition to evaluate student worksheets. U.S. Patent Application No. 13/012,359.

Valenti, S., Neri, F., & Cucchiarelli, A. (2003). An overview of current research on automated essay grading. *Journal of Information Technology Education*, *2*, 319-330.

Wang, T., Su, X., Wang, Y., & Ma, P. (2007). Semantic similarity-based grading of student programs. *Information and Software Technology*, *49*(2), 99-107.

Warne, K. (2014). Experiences in implementing automated evaluation and grading in a MOOC using a behaviour-driven testing framework (Doctoral dissertation).

Weinberger, A., Dreher, H., Al-Smadi, M., & Guetl, C. (2011). *Analytical assessment rubrics to facilitate semi-automated Essay grading and feedback provision.* Paper presented at the Australian Technology Network Assessment Conference 2011: Meeting the Challenges, Perth, Western Australia.

Williamson, D. M., Bennett, R. E., Lazer, S., Bernstein, J., Foltz, P. W., Landauer, T. K., & Sweeney, K. (2010). Automated scoring for the a

Wilson, R. C. (1995). *UNIX test tools and benchmarks: methods and tools to design, develop, and execute functional, structural, reliability, and regression tests*: Prentice-Hall, Inc

Xie, T. (2005). *Improving Effectiveness of Automated Software Testing in the Absence of Specifications*, Doctoral Thesis. University of Washington. IEEE Xplore

Yang, H., & Liu, X. (2012). *Software reuse in the emerging cloud computing era*. Hershey, PA: Information Science Reference.

Yang, Q., Li, J., & Weiss, D. (2006). A survey of coverage based testing tools, *In Proceedings of the 2006 international Workshop on Automation of Software Test, ACM Press*, pp.99-103

Zeigler, B. P., & Mittal, S. (2005, October). Enhancing DoDAF with a DEVS-based system lifecycle development process. In Systems, Man and Cybernetics, 2005 IEEE International Conference on (Vol. 4, pp. 3244-3251). IEEE.

Appendix

Appendix A: Results of the Percentage of Match and Mismatch

No	Compare between Original Image No	With Target Image No	Percentage of matching	Percentage of mismatching	Final Mark
1	O1	T1	100%	0%	30
		T2	50%	50%	15
		T3	50%	50%	15
		T4	50%	50%	15
		T5	0%	100%	0
2	O2	T1	100%	0%	30
		T2	50%	50%	15
		T3	75%	25%	22.5
		T4	0%	100%	0
		T5	50%	50%	15
3	O3	T1	100%	0%	30
		T2	80%	20%	24
		T3	60%	40%	18
		T4	40%	60%	12
		T5	20%	80%	6
		T6	80%	20%	24
		T7	0%	100%	0
4	O4	T1	0%	100%	0
		T2			26.25
		T3	0%	100%	0
		T4	100%	0%	30
5	O5	T1	87%	13%	26.25
		T3	0%	100%	0
		T4	100%	0%	30

6	O6	T4	100%	0%	30
7	O7	T1	100%	0%	30
8	O8	T1	100%	0%	30
		T2	100%	0%	30
		T3	100%	0%	0
9	O9	T1	100%	0%	30
		T2	96%	4%	27.5
		T3	87%	13%	26.25
		T4	96%	4%	27.5
		T5	96%	4%	27.5
		T6	8%	12%	2.5
10	O10	T1	100%	0%	30
		T2	100%	0%	30
		T3	50%	50%	15
		T4	0%	100%	0
11	O11	T2	100%	0%	30
		T3	83%	17%	25
		T4	33	67%	10
		T5	0%	100%	0
12	O12	T1	100%	0%	30
		T2	50%	50%	15
		T3	100%	0%	0
		T4	50%	50%	15
13	O13	T1	0%	100%	0
		T2	100%	0%	30
14	O14	T1	100%	0%	30
		T2	100%	0%	0
15	O15	T1	100%	0%	30
		T2	100%	0%	30

		T3	83%	17%	25
		T4	42%	58%	12.5
16	O16	T1	100%	0%	30
		T2	0%	100%	0
17	O17	T2	100%	0%	30
		T3	0%	100%	0
		T4	0%	100%	0
18	O18	T2	100%	0%	30
		T3	50%	50%	15
		T4	0%	100%	0
19	O19	T1	100%	0%	30
		T2	0%	100%	0
20	O20	T1	100%	0%	30
		T2	50%	50%	15
		T3	97%	3%	29
		T4	33	67%	10
21	O21	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
22	O22	T1	100%	0%	30
		T2	83%	17%	25
		T3	67%	23%	20
		T4	83%	17%	25
		T5	0%	100%	0
		T6	50%	50%	15
23	O23	T1	100%	0%	30
		T2	0%	100%	0
24	O24	T1	100%	0%	30
		T3	50%	50%	15
		T4	50%	50%	15

		T5	0%	100%	0
25	O25	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
26	O26	T1	100%	0%	30
		T2	80%	20%	24
		T4	91%	9%	27
		T5	0%	100%	0
27	O27	T1	100%	0%	30
		T2	91%	9%	27
		T3	50%	50%	15
28	O28	T1	100%	0%	30
		T3	0%	100%	0
		T4	9%	91%	2.7
		T5	0%	100%	0
		T6	0%	100%	0
		T7	9%	91%	2.7
		T9	9%	91%	2.7
		T10	0%	100%	0
29	O29	T1	100%	0%	30
		T2	50%	50%	15
		T3	50%	50%	15
		T4	0%	100%	0
30	O30	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	87%	13%	26.25
		T5	25%	75%	7.5
31	O31	T1	100%	0%	30
		T2	0%	100%	0
		T3	0%	100%	0

		T4	67%	23%	20
		T5	0%	100%	0
32	O32	T1	100%	0%	30
		T3	100%	0%	30
		T4	0%	100%	0
		T5	0%	100%	0
		T7	67%	23%	20
		T8	0%	100%	0
33	O33	T1	100%	0%	30
		T2	100%	0%	30
		T3	0%	100%	0
		T4	0%	100%	0
		T5	50%	50%	15
		T6	75%	25%	22.5
		T7	25%	75%	7.5
34	O34	T1	100%	0%	30
		T3	100%	0%	0
		T4	50%	50%	15
35	O35	T1	100%	0%	30
		T3	50%	50%	15
		T4	50%	50%	15
36	O36	T1	100%	0%	30
		T2	94%	6%	28.12
		T3	25%	75%	7.5
		T4	0%	100%	0
37	O37	T1	100%	0%	30
		T2	100%	0%	30
		T4	0%	100%	0
		T6	15%	85%	4.28
38	O38	T1	100%	0%	30

		T2	0%	100%	0
		T3	0%	100%	0
39	O39	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	50%	50%	15
		T4	25%	75%	7.5
		T5	67%	23%	20
		T6	0%	100%	0
40	O40	T1	33	67%	10
		T2	80%	20%	24
		T3	90%	10%	27
		T4	50%	50%	15
		T5	83%	17%	25
		T6	0%	100%	0
41	O41	T1	100%	0%	30
		T2	25%	75%	7.5
		T3	0%	100%	0
		T4	0%	100%	0
42	O42	T1	100%	0%	30
		T2	50%	50%	15
		T3	50%	50%	15
		T4	75%	25%	22.5
		T5	0%	100%	0
		T6	87%	13%	26.25
43	O43	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
44	O44	T1	100%	0%	30
		T2	0%	100%	0
		T3	80%	20%	24

		T4	0%	100%	0
45	O45	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
		T4	50%	50%	15
46	O46	T1	100%	0%	30
		T2	67%	23%	20
		T3	33	67%	10
		T4	0%	100%	0
		T5	100%	0%	30
47	O47	T1	100%	0%	30
		T2	50%	50%	15
		T3	100%	0%	0
48	O48	T1	100%	0%	30
		T2	0%	100%	0
49	O49	T1	100%	0%	30
		T2	50%	50%	15
		T4	0%	100%	0
50	O50	T1	100%	0%	30
		T2	94%	6%	28.12
		T3	17%	83%	8.12
		T4	0%	100%	0
51	O51	T1	100%	0%	30
		T2	0%	100%	0
		T3	50%	50%	15
52	O52	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	50%	50%	15
		T4	25%	75%	7.5
		T5	0%	100%	0
		T6	100%	0%	30

53	O53	T1	100%	0%	30
		T2	88%	12%	28.84
		T3	42%	58%	12.69
		T5	0%	100%	0
54	O54	T1	100%	0%	30
		T2	67%	23%	20
		T3	83%	17%	25
		T4	50%	50%	15
		T5	0%	100%	0
55	O55	T1	100%	0%	30
		T2	86%	14%	25.71
		T3	57%	43%	17.14
		T4	43%	57%	12.85
		T5	14%	86%	4.28
		T6	0%	100%	0
56	O56	T1	100%	0%	30
		T3	67%	23%	20
		T4	33	67%	10
		T5	0%	100%	0
57	O57	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	0%	100%	0
		T4	50%	50%	15
		T5	50%	50%	15
58	O58	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
		T4	67%	23%	20
		T5	50%	50%	15
59	O59	T1	100%	0%	30
		T2	86%	14%	25.71

		T3	71%	29%	21.42
		T4	57%	43%	17.14
		T5	43%	57%	12.85
		T6	29%	71%	8.57
		T7	15%	85%	4.28
		T8	0%	100%	0
		T9	50%	50%	15
60	O60	T1	100%	0%	30
		T2	83%	17%	25
		T3	50%	50%	15
		T4	17%	83%	5
		T5	83%	17%	26.64
61	O61	T1	100%	0%	30
		T2	94%	6%	28.12
		T3	0%	100%	0
		T4	50%	50%	15
		T5	96%	4%	27.5
62	O62	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
		T4	87%	13%	26.25
63	O63	T1	100%	0%	30
		T2	67%	23%	20
		T3	33	67%	10
		T4	36%	74%	12.14
		T5	0%	100%	0
64	O64	T1	100%	0%	30
		T2	79%	21%	23.76
		T3	59%	41%	17.63
		T4	38%	62%	11.5
		T5	28%	72%	8.43

		T6	0%	100%	0
65	O65	T1	100%	0%	30
		T2	14%	86%	4.08
		T3	91%	9%	27.2
		T4	0%	100%	0
		T5	41%	59%	13.6
66	O66	T1	100%	0%	30
		T2	46%	54%	13.75
		T3	83%	17%	27.08
		T4	92%	8%	28.88
		T5	33	67%	10
		T6	100%	0%	0
67	O67	T1	100%	0%	30
		T2	67%	23%	20
		T3	33	67%	10
		T4	17%	83%	5
		T5	0%	100%	0
68	O68	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
		T4	33	67%	10
69	O69	T1	100%	0%	30
		T2	44%	56%	13.75
		T3	93%	17%	27.81
		T4	0%	100%	0
70	O70	T1	100%	0%	30
		T2	50%	50%	15
		T3	82%	18%	22
		T4	0%	100%	0
71	O71	T1	100%	0%	30
		T2	83%	17%	25

		T3	33	67%	10
		T4	0%	100%	0
72	O72	T1	100%	0%	30
		T2	50%	50%	15
		T3	0%	100%	0
		T4	67%	23%	20
		T5	50%	50%	15
		T6	50%	50%	15
		T7	33	67%	10
		T8	17%	83%	5
73	O73	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	50%	50%	15
		T4	50%	50%	15
		T5	0%	100%	0
		T6	91%	9%	27.25
		T7	95%	5%	28.5
74	O74	T1	100%	0%	30
		T2	89%	11%	26.64
		T3	78%	22%	23.31
		T4	67%	33%	19.98
		T5	55%	45%	16.65
		T6	44%	56%	13.32
		T7	33%	67%	9.99
		T8	22%	78%	6.66
		T9	11%	89%	3.33
		T10	97%	3%	29.04
75	O75	T1	100%	0%	30
		T2	67%	23%	20
		T3	33	67%	10
		T4	50%	50%	15

		T5	0%	100%	0
76	O76	T1	100%	0%	30
		T2	50%	50%	15
		T4	0%	100%	0
77	O77	T1	100%	0%	30
		T2	33	67%	10
		T3	93%	7%	27
		T4	67%	23%	20
		T5	33	67%	10
78	O78	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	50%	50%	15
		T4	12%	88%	3.75
79	O79	T1	100%	0%	30
		T2	57%	43%	17.5
		T3	67%	23%	20
		T4	91%	9%	26.66
		T5	0%	100%	0
80	O80	T1	100%	0%	30
		T2	86%	14%	25.71
		T4	88	12%	24.99
		T5	29%	71%	8.57
		T6	0%	100%	0
81	O81	T1	100%	0%	30
		T2	65%	35%	19.5
		T3	50%	50%	15
		T4	50%	50%	15
		T5	75%	25%	22.5
		T7	0%	100%	0
82	O82	T1	100%	0%	30
		T2	67%	23%	20

		T3	75%	25%	22.5
		T4	0%	100%	0
		T5	50%	50%	15
83	O83	T1	100%	0%	30
		T2	91%	9%	27.2
		T3	82%	18%	24.48
		T4	73%	27&	21.76
		T5	64%	36%	19.04
		T6	54%	46%	16.32
		T7	45%	55%	13.6
		T8	36%	64%	10.88
		T9	27%	73%	8.16
		T10	18%	82%	5.44
		T11	9%	91%	2.72
		T13	100%	0%	30
		T14	0%	100%	0
84	O84	T1	100%	0%	30
		T2	0%	100%	0
85	O85	T1	100%	0%	30
		T2	38%	62%	10.7
		T3	88%	12%	27.09
		T4	93%	7%	28.52
		T5	96%	4%	29.24
		T6	0%	100%	0
86	O86	T1	100%	0%	30
		T2	12%	88%	3.75
		T3	12%	88%	3.75
		T4	12%	88%	3.75
		T5	87%	13%	26.25
		T6	75%	25%	22.5
		T7	50%	50%	15

		T8	25%	75%	7.5
		T9	0%	100%	0
87	O87	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	15	85%	3
		T4	40%	60%	12
		T5	75%	25%	22.5
		T6	20%	80%	6
		T7	0%	100%	0
88	O88	T1	100%	0%	30
		T2	80%	20%	24
		T3	50%	50%	15
		T4	0%	100%	0
89	O89	T1	100%	0%	30
		T2	85%	15%	25.5
		T3	87%	13%	26
		T4	96%	4%	27.5
		T5	67%	23%	20
		T6	33	67%	10
		T7	7%	93%	2
90	O90	T1	100%	0%	30
		T2	50%	50%	15
		T3	50%	50%	15
		T4	67%	23%	20
		T5	0%	100%	0
91	O91	T1	100%	0%	30
		T2	0%	100%	0

		T3	80%	20%	24
92	O92	T1	100%	0%	30
		T2	75%	25%	22.5
		T3	50%	50%	15
		T4	25%	75%	7.5
		T5	95%	5%	29.06
		T6	50%	50%	15
		T7	0%	100%	0
93	O93	T1	100%	0%	30
		T2	89%	11%	26.66
		T3	83%	17%	24.99
		T4	0%	100%	0
		T5	55%	45%	16.66
94	O94	T1	100%	0%	30
		T2	66%	34%	19.77
		T3	95%	5%	27.13
		T4	0%	100%	0
95	O95	T1	100%	0%	30
		T5	0%	100%	0
		T6	0%	100%	0
		T7	0%	100%	0
		T8	0%	100%	0
96	O96	T1	100%	0%	30
		T2	80%	20%	24

		T3	60%	40%	18
		T4	40%	60%	12
		T5	20%	80%	6
		T6	76%	24%	23
		T7	72%	28%	20.5
		T8	10%	90%	3
97	O97	T1	100%	0%	30
		T2	50%	50%	15
		T3	83%	17%	25
		T4	67%	23%	20
		T5	33	67%	10
		T6	83%	17%	25
		T7	0%	100%	0
98	O98	T1	100%	0%	30
		T2	80%	20%	24
		T3	50%	50%	15
		T4	90%	10%	27
		T5	0%	100%	0
		T6	60%	40%	21
		T7	50%	50%	15
		T8	40%	60%	12
99	O99	T1	100%	0%	30
		T2	90%	10%	27
		T3	80%	20%	24

		T4	80%	20%	24
		T5	0%	100%	0
100	O100	T1	100%	0%	30
		T4	12%	88%	8.75
		T5	4%	96%	1.25
		T6	0%	100%	0

Appendix B: Questionnaire

This questionnaire contains 9 questions, please answer appropriately.

Thank you.

Part 1: Personal Information

1. What is your name?

   ………………………………….

2. What is your age?

   A) 1-15

   B) 15-30

   C) 30-45

   D) 45-60

   E) Over 60

3. What is your gender?

   A) Male

   B) Female

Part 2: Technical Questions

4. On a scale of 1 to 5 where 1 is unsatisfied, 2 is somewhat unsatisfied, 3 is neutral, 4 is satisfied and 5 is very satisfied how do you rate the feedback given by the Automatic Image Marking process?

   1) Unsatisfied
   2) Somewhat unsatisfied
   3) Neutral
   4) Somewhat Satisfied
   5) Very satisfied

5. Has the automatic image marking process helped in improving the learning abilities and skills of the students?

1) Yes
2) No

6. How would you rate the effectiveness of the Image marking process in analysing and giving feedback of the students' scripts?

   1) Not effective at all
   2) Fair
   3) Very effective

7. Is the Image marking process consistent in giving out feedback of the examination?

   1) Yes
   2) No

8. Compared to manual assessment of learners' examination scripts, how do you compare the efficiency of the image marking process?

   1) Very efficient
   2) Not efficient

9. Read the following statements and indicate your approval rating of the system

Statements	**1**	**2**	**3**	**4**	**5**
Speed of recognition of the script					
Complexity of the implementation of the system					
Pattern recognition of the script					
Feature detection by the system					
Biasness of the OCR system					

Is it true that while matching results with the optimal answer, great matching similarity percentage indicate a high accuracy of student answer compared to optimal answers that provided by the teacher and lead to the high final mark?

A) Yes

B) No

10. Image quality is one of the most important factors for improving the quality of recognition in OCR. A resolution of 200 DPI to 400 DPI is recommended for better recognition.

A) True

B) False

11. Features that can be calculated from an essay transcription and which have significance to score include the following except

A) Number of words

B) Number of sentences

C) Average length of a sentence

D) The length of the essay

E) Number of verbs

F) Size and color of the letters