OMR BASED SENTIMENT INSIGHTS

Final Year Project Report

AAMIR AZIZ **(BAI-21S-001)**

AMJAD AZIZ **(BAI-21S-002)**

SESSION 2021-2025

Supervisor

Syed Muhammad Hassan Zaidi

Project report in partial fulfillment of the requirements for the award of Bachelors of Science (AIMS) degree.

IN

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

SINDH MADARESSATUL ISLAM UNIVERSITY KARACHI, PAKISTAN

(February 2025)

STUDENT’S DECLARATION

We hereby declare that this project report is based on our original work except for citation and quotation which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at SINDH MADARESSATUL ISLAM UNIVERSITY or other institute.

1) Name: AAMIR AZIZ (BAI-21S-001)

Signatures: - __________________

2) Name: AMJAD AZIZ (BAI-21S-002)

Signatures: - __________________

OMR BASED SENTIMENTS INSIGHTS

APPROVAL FOR SUBMISSION

THE DEPARTMENT OF ARTIFICIAL INTELLIGENCE, SINDH MADRESSATUL ISLAM UNIVERSITY, ACCEPTS THIS THESIS SUBMITTED BY **AAMIR AZIZ** and **AMJAD AZIZ** IN ITS PRESENT FORM AND IT IS SATISFYING THE DISSERTATION REQUIREMENTS FOR THE AWARD OF BACHELOR DEGREE IN ARTIFICIAL INTELLIGENCE.

SUPERVISOR:

Syed Muhammad Hassan Zaidi

Lecturer

DEAN/HOD OF DEPARTMENT:

DR. AFTAB AHMED SHAIKH ________________________

DATED: February 2025

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

SINDH MADRESSATUL ISLAM UNIVERSITY,

KARACHI, PAKISTAN

ABSTRACTION

This study introduces an advanced Optical Mark Recognition (OMR) system integrated with sentiment analysis to enhance data interpretation across various domains. Unlike traditional OMR, which only captures filled responses, the proposed system detects emotional markers such as stress, confidence, and confusion, offering deeper cognitive and psychological insights. By leveraging machine learning and predictive analytic, this framework automates sentiment assessment, improves accuracy, and streamlines large-scale data processing.

The system is particularly beneficial in education, where it helps evaluate student performance and emotional states, and in business sectors, where it enhances survey reliability and market research. The technical feasibility is ensured through high-resolution scanning, AI-driven classification, and statistical validation, demonstrating its efficiency in handling large datasets with high accuracy. Experimental results confirm its effectiveness in extracting sentiment-based insights, making it a versatile tool for public opinion polling, institutional feedback, and policy evaluation.

Future enhancements will focus on real-time emotion tracking and expanding its applicability to diverse assessment environments, further optimizing decision-making and personalized feedback mechanisms.

Keywords:

Optical Mark Recognition (OMR), Sentiment Analysis, Machine Learning, Predictive Analytic, Student Performance, Market Research, Emotion Detection, Data Interpretation, AI in Education, Large-Scale Sentiment Assessment

ACKNOWLEDGEMENT

In the name of Allah, the most Gracious and the Most Merciful.

Peace and blessing of Allah be upon Prophet Muhammad ﷺ

Firstly, we would like to thanks Allah SWT for blessing us with the hope of this new idea and then We would like to express our deepest gratitude and appreciation to Syed Muhammad Hassan Zaidi, our dedicated and supportive course supervisor, for his invaluable guidance and unwavering encouragement
throughout the completion of our Final Year Project (FYP). His expertise, insightful feedback, and commitment to our academic growth have been instrumental in shaping the outcome of this project.

Sir Hassan Zaidi's careful focus on detail and his skill in offering constructive feedback significantly improved our research methods, bolstered our analytical strategies, and elevated the overall standard of this documentation. His patience, approachability, and readiness to dedicate time and energy to resolving our questions and issues contributed greatly to our shared personal and professional growth.

DEDICATION

We are incredibly thankful to our families and friends for their steadfast support, understanding, and encouragement during this challenging journey. Their faith in our capabilities and their continuous motivation have served as a significant source of inspiration. This project could not have been achieved without the combined efforts and assistance of these individuals and many others who have played a role in our academic and personal development. We are genuinely grateful and honored by their involvement in our journey. Finally, we want to convey our sincere gratitude to all the participants who generously took part in the data collection process, allowing us to gather important insights and perform a thorough analysis.

TABLE OF CONTENTS

CHAPTER 02 – LITERATURE REVIEW

CHAPTER 03 – REQUIREMENT SPECIFICATION & DESIGN

CHAPTER 04 – METHODOLOGY

CHAPTER # 01

INTRODUCTION

1.1 OVERVIEW:

Conventional OMR systems mostly record filled-in replies, but they are unable to identify student emotions like reluctance, stress, or confidence [1]. This constraint results in imprecise evaluations since crucial psychological elements impacting performance go unnoticed [2]. The evaluation method is laborious and prone to human error since teachers sometimes rely on subjective interpretations or extra questionnaires to determine students' feelings [3]. Similarly, because OMR is unable to analyze sentiment, institutional input and market research have limited insights [4]. When distinct tools are needed for sentiment analysis, processing huge datasets becomes inefficient and data handling becomes more difficult [5].

If we see the OMR scanning , the lack of emotional detection in tests hinders individualized instruction, which makes it difficult for teachers to offer support that is specific to each student's requirements [6]. In order to get around these issues, sentiment analysis combined with OMR allows for emotion-aware evaluations that provide more in-depth information than just pointing out right and wrong responses [7]. In this integrated approach, automation optimizes large-scale data processing, reduces manual labor, and improves evaluation accuracy [8]. Businesses can improve their market research tactics, and institutions can use sentiment-based data to enhance test conditions [9]. In the end, detecting emotions in tests promotes more equitable assessments, effective data processing, and individualized education [10].

1.2 CONTRIBUTION

By incorporating sentiment analysis, this study improves optical mark recognition (OMR), turning it from a simple grading tool into a sophisticated system that can decipher respon dents' emotional and behavioral tendencies.
This study broadens the reach of traditional OMR systems, which only identify filled bub bles, by analyzing student markings for stress, confidence, and hesitancy. This provides i mportant insights regarding cognitive load and engagement. Instantaneous feedback and flexible decision-making are made possible by real- time sentiment analysis, which improves the fairness and insight of assessments. Its uses in market research, institutional evaluations, and education are expanded by the ca pacity to categorize replies into three attitude categories: positive, negative, and neutral. By ensuring that assessments are accurate and representative of respondents' psychologica l states, this research helps to create a more inclusive, data-driven assessment framework. It lays the groundwork for better exam circumstances, more individualized learning metho ds, and improved decision-making across a range of fields.

1.2.1 MOTIVATION

The desire to improve traditional optical mark recognition (OMR) by including sentiment analysis is what motivates this research, turning it from a straightforward response detection tool into a potent instrument for comprehending human emotions. This combination enhances data interpretation in business, education, and other fields by providing deeper insights regarding a respondent's confidence, stress, or hesitancy beyond grades and surveys. This study intends to close the gap between structured response recognition and sentiment-driven analytics by creating more intelligent, flexible, and economical OMR systems, opening the door to more intelligent and flexible decision- making.

Many students in Pakistan, as well as many other nations, frequently struggle to correctly fill out bubble sheets during exams, particularly those who study at night because of work or lack of resources. Inadequate lighting, stress, and fatigue might affect their accuracy and cause assessments to be misinterpreted. In order to find patterns of tension, confidence, or hesitancy in student markings, this study combines emotion analysis with optical mark recognition (OMR). By improving exam settings, ensuring fair evaluations, and better understanding students' psychological states, this innovation can help educators create a more welcoming and encouraging learning environment. identical but shorter rather than shorter.

1.2.2 PROBLEM DECLARATION

a) **Faced challenges in data collection**

- Processing multiple-choice questions (MCQs) on a single bubble sheet presents difficulties.
- Detecting and analyzing responses separately from a sheet containing 50 MCQs requires advanced processing capabilities.

b) **Survey-Based Approach**

- Conducted a survey using seven variants of bubble sheet formats.
- Designed to assess student confidence, stress levels, and marking behavior without external support.

c) **Lack of Sentiment Analysis in OMR**

- Traditional OMR systems lack frameworks for integrating sentiment analysis.
- Extracting meaningful insights from marked responses remains a challenge.

1.2.3 OBJECTIVES

- **Enhancing Academic Assessments**: Improve traditional examination methods by integrating sentiment analysis with OMR to detect stress, confidence, and confusion in student responses, ensuring fair and accurate evaluations.
- **Supporting Students in Challenging Conditions**: Address difficulties faced by students in Pakistan, especially those studying at night due to work or resource

constraints, by identifying fatigue-related marking errors and improving assessment reliability.

- **Behavioral Insights for Educators**: help teachers and institutions understand students' psychological states, learning patterns, and exam-taking behavior.
- **Applications in Academic Research and Institutional Evaluations**: Extend the use of sentiment-driven OMR for student performance tracking, curriculum effectiveness analysis, and psychological research in education. This study aims to bridge the gap between traditional assessments and behavioral analytics, fostering a more inclusive and data-driven academic environment.

SCOPE

This allowing for a deeper understanding of respondent behavior beyond simple answer detection. Traditional OMR systems are limited to identifying filled bubbles, whereas this study introduces the capability to assess stress, confidence, and confusion in markings, providing valuable insights into cognitive and emotional states.

The study has broad applications in education, where it can help teachers evaluate not just student performance but also their psychological state during exams. Institutions can leverage this data to refine assessment strategies, improve exam environments, and offer targeted support to students dealing with stress or anxiety.

Beyond academia, the integration of sentiment analysis in OMR is valuable for businesses and research organizations conducting surveys and behavioral studies. Understanding sentiment through structured responses enables more adaptive decision-making, ensuring a fairer, more insightful, and data-driven approach to assessment and feedback analysis. The introduction of image-based and camera-based OMR further improves accessibility, reducing costs and making the technology more scalable for diverse applications.

1.2.4 MODULES

Classification & Reporting

a) User Interface

The system provides a user-friendly interface for uploading Optical Mark Recognition (OMR) sheets. Users can easily visualize extracted bubbles, classified responses, and sentiment predictions. The interface is designed to require minimal manual intervention and supports batch processing for efficiency.

- Performance Evaluation Performance metrics include:
- Accuracy: Measures the percentage of correctly classified instances.
- Precision, Recall, and F1-Score: Evaluates the effectiveness of the classification model.
- Confusion Matrix: Identifies misclassifications between sentiment classes.
- Processing Time: Ensures optimized execution for large datasets.

b) Feasibility Study

c) Risks Involved Potential risks

- Technical Risks: Image quality variations affecting circle detection.
- Operational Risks: Incorrect classification due to ambiguous markings.
- Financial Risks: Costs associated with storage, processing, and computational resources.

d) Resource Requirement

- Hardware: High-resolution scanner, GPU-supported machine for processing.
- Software: OpenCV, Python, NumPy, Pandas, and machine learning libraries.
- Human Resources: Data scientists, software developers, and domain experts.

e) Solution Application Areas

- Education: Automated grading of multiple-choice answer sheets.
- Surveys & Polling: Analyzing responses in large-scale surveys.
- Psychological Assessments: Evaluating responses based on sentiment analysis.

f) Tools/Technology Used

- Programming Language: Python
- Libraries: OpenCV, Pandas, NumPy, Matplotlib, Scikit-Learn

- Frameworks: Streamlet for UI integration

g) Expertise of Team Members

- Data Scientists: Responsible for feature engineering and model training.
- Software Developers: Implementing image processing and classification algorithms.
- Project Manager: Ensuring project milestones and feasibility assessment.

h) Domain of the Project

The project focuses on sentiment analysis of OMR sheets. It integrates machine learning for recognizing responses based on bubble intensity and solidity. It is relevant in fields requiring automated assessment, including education and psychological research.

i) What This Application Cannot Do?

It does not support handwritten answer recognition.

It is not designed for subjective responses requiring natural language processing.

It cannot process poor-quality scans with excessive noise.

j) Deliverables

- Functional UI: A web-based interface for uploading and processing OMR sheets.
- Preprocessing Pipeline: Automated circle extraction and classification.
- Machine Learning Model: A trained logistic regression classifier for sentiment prediction.
- Comprehensive Report: Summary of predictions and statistical insights.

CHAPTER # 02

LITERATURE REVIEW

2.1 INTRODUCTION:

Optical Mark Recognition (OMR) is a technology used to read and interpret human- marked data, such as bubble sheets. While OMR primarily focuses on detecting filled bubble. In the realm of data collection and analysis, Optical Mark Recognition (OMR) has emerged as a powerful tool for efficiently processing human-marked responses, particularly in educational assessments and surveys.[1],[2],[3] Traditionally, OMR technology has been employed to automate the grading of multiple-choice questions by detecting filled bubbles on answer sheets. As organizations increasingly seek to understand the sentiments and opinions of their respondents, the integration of sentiment analysis with OMR presents a novel approach to interpreting qualitative data.[4] By mapping the responses captured through bubble sheets to specific sentiment categories, researchers can gain valuable insights into public opinion, customer satisfaction, and overall sentiment trends.[5] This synergy between OMR and sentiment analysis not only streamlines data processing but also enhances the depth of understanding derived from quantitative responses, paving the way for more informed decision-making.[6],[7],[8]

Sentiment analysis, also referred to as opinion mining, mood analysis, or opinion analysis, represents an intricate computational approach that integrates statistical methodologies, advanced natural language processing (NLP) techniques, and sophisticated machine learning (ML) algorithms to systematically examine textual content, enabling the identification, extraction, and interpretative evaluation of subjective opinions, intrinsic emotions, and underlying sentiments within written discourse, with the principal objective of determining the affective polarity of a text by classifying subjective expressions into sentiment clusters—positive, negative, or neutral—thereby facilitating its application across diverse domains requiring automated decision-making and behavioral analysis [9],[10],[11]

A key technological advancement that complements sentiment analysis in automated data processing and large-scale evaluation is Optical Mark Recognition (OMR), a system designed for the efficient detection, interpretation, and computational processing of human-marked responses,[12] typically in the form of check boxes or filled bubbles on standardized assessments, surveys, and institutional forms, thereby enabling rapid and precise data collection, which is integral to optimizing large-scale sentiment assessment and survey-based sentiment analysis models; [13] OMR technology, initially conceived as a hardware-based solution, has progressively evolved into a software-driven framework that employs image processing algorithms to analyze scanned documents, yet its widespread adoption remains hindered by limitations in flexibility, accessibility, and cost- efficiency, as documented extensively in the literature [14].

The synergy between sentiment analysis and Optical Mark Recognition is evident in their shared objective of automating the interpretation of human input, whether through textual sentiment classification or structured response processing, both of which are extensively applied in academic evaluations, market research, social analytics, and institutional feedback mechanisms, where computational efficiency and accuracy are paramount; consequently, OMR has gained prominence across multiple sectors, including automated test grading, large-scale community surveys, empirical evaluations, and administrative documentation, thereby reinforcing the essential role of sentiment-based data analytics in contemporary decision-making frameworks [15].

Image-based Optical Mark Recognition (OMR) obviates the necessity for proprietary hardware apparatus or pre-structured answer sheets, thereby mitigating both capital expenditure and recurrent operational costs. Furthermore, this sophisticated solution empowers users with the flexibility to architect bespoke answer sheets utilizing conventional word processing software.[16] The efficacy of image-based OMR hinges upon the meticulous scanning of documents, subsequently leveraging advanced image processing algorithms and intricate pattern recognition methodologies to precisely localize and categorize the designated response bubbles within the answer sheet.[17]

Camera-based document analysis faces challenges such as illumination issues, zooming, low resolution, and distortion. This paper presents a solution for camera-based OMR that overcomes these problems by utilizing specially designed **bubble sheets** with detection marks. [18]

IMPACT OF OPTICAL MARK RECOGNITION (OMR) SENTIMENT ANALYSIS ON BUBBLE SHEETS

The integration of Optical Mark Recognition (OMR) technology with sentiment analysis represents a paradigm shift in the way data is extracted, processed, and interpreted from bubble sheets.

a. Augmented Data Accuracy and Operational Efficiency

The automation facilitated by OMR technology significantly mitigates errors associated with manual data entry and grading.[19] By precisely capturing responses from bubble sheets, this technology ensures that sentiment analysis is conducted on high-fidelity data, thereby improving the reliability and validity of analytical outcomes. The reduction in human intervention also enhances efficiency, allowing large datasets to be processed with minimal discrepancies.[20]

b. Expedited Data Processing and Interpretation

The integration of OMR with sentiment analysis enables rapid data processing, converting handwritten or marked responses into structured digital formats almost instantaneously.[21] This acceleration is particularly beneficial in time-sensitive environments, such as standardized testing, organizational feedback collection, and real-time market sentiment evaluations, where swift decision-making is crucial.[22]

c. Quantification of Sentiment-Oriented Data

By categorizing responses into sentiment clusters—such as positive, negative, and neutral—OMR sentiment analysis transforms qualitative feedback into quantifiable metrics. [23] This structured classification provides data-driven insights into consumer behavior, student engagement, and employee satisfaction, allowing organizations to discern patterns and trends that inform strategic decision-making.[24]

d. Enhanced Feedback Mechanisms for Institutional and Corporate Use

Incorporating sentiment analysis into OMR-based assessments facilitates a more nuanced understanding of respondent perspectives. In the educational sector, for instance, the ability to gauge students' sentiments towards coursework, faculty performance, or institutional policies can enable educators to implement targeted pedagogical enhancements.[25] Similarly, in corporate settings, organizations can assess employee morale and customer satisfaction, fostering continuous improvement.[26]

e. Integration with Advanced Analytical Frameworks

OMR-generated sentiment data can be further refined by integrating it with machine learning algorithms, natural language processing (NLP), and predictive analytics.[27] These advanced methodologies allow for deeper exploration of sentiment trends, correlation mapping, and forecasting, thereby strengthening evidence-based decision-making processes in diverse industries.[28]

f. Cost-Effectiveness and Resource Optimization

By eliminating the need for labor-intensive manual grading and data entry, OMR sentiment analysis reduces operational costs and optimizes resource allocation. [29] The automation of data collection and interpretation not only minimizes human labor expenses [30], but also enhances scalability, making it a cost-efficient solution for institutions and enterprises with extensive data-processing needs.[31]

g. Real-Time Sentiment Insights for Dynamic Decision-Making

With advancements in AI-driven OMR technologies, organizations can achieve real-time sentiment analysis, enabling instantaneous responses to stakeholder feedback.[32], [33] This capability is particularly advantageous in dynamic environments such as market research, crisis management, and policy-making, where rapid sentiment shifts necessitate swift adaptive measures.[35],[36],[37]

h. Expanding the Scope of OMR Sentiment Analysis

Beyond its conventional application in academic assessments, OMR sentiment analysis has evolved into a versatile tool with applications in public opinion polling, consumer behavior studies, healthcare feedback systems, and governmental policy evaluations.[38] Its adaptability across multiple sectors underscores its potential to revolutionize how institutions collect, interpret, and act upon sentiment-based data.[39],[40]

2.2 Table. # 1 Summarized Version of Literature Review Past Researched Papers

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
1	2016	Sujon, M. A. K.	Optical Mark Reader (OMR) design and development	Developed an OMR system as part of a doctoral dissertation	Limited scope in real-time processing and deep learning integration
2	2024	Enes Alperen Buğaz, Orhan Akbulut, Aysun Taşyapi Çelebi,	Deep learning-based order form recognition	Applied deep learning for structured document recognition	No application in academic assessments or psychological analysis

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
		Uğur Yildiz			
3	2020	Hussain, T.	Multimedia computing	Covered various aspects of multimedia processing	No direct application to OMR technology
4	2021	de Elias, E. M., Tasinaffo, P. M., & Hirata Jr, R.	Advances, difficulties, and limitations in OMR	Explored challenges in OMR accuracy and adaptability	Lacked sentiment analysis integration in OMR
5	2013	AL-Marakeby	Fast camera-based OMR system	Developed a high-speed OMR using camera-based methods	Did not explore emotional state detection in OMR
6	2003	David Doermann, Jian Liang, Huiping Li	Camera-based document analysis	Discussed progress in document image recognition	No sentiment analysis application in OMR
7	2007	Li Zhang, Andy M. Yip, Chew Lim Tan	Removing shading distortions in camera-	Proposed methods for improving scanned document quality	Lacked real-time emotional state detection

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
			based document images		
8	2005	K. Chua, L. Zhang, Y. Zhang, C. Tan	Restoration of warped document images	Provided fast and stable solutions for image restoration	Did not focus on OMR sentiment analysis
9	1998	Bergeron, B.P.	OMR technology and applications	Discussed OMR evolution and applications	Did not address behavioral analysis through OMR
10	2023	Liu, Xing et al.	Multimodal text recognition and sentiment analysis	Used visual text recognition for sentiment analysis in online content	No direct OMR application for student evaluation
11	2018	Tümer, Abdullah & Küçükkara, Zeki	Image processing-oriented OMR evaluation	Developed an OMR system based on image processing	Lacked emotional state recognition in responses
12	1981	Smith, A.	Improving	Focused on making	No exploration of

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
		M.	OMR usability	OMR easier for users	behavioral insights
13	N/A	IBM Archives	IBM 805 Test Scoring Machine	Provided historical context of automated test scoring	No sentiment analysis integration
14	2013	NYTimes Article	Evolution of standardized testing	Examined history and impact of standardized tests	No technological advancements related to sentiment analysis in OMR
15	2015	Veronese, K.	History of Scantrons	Discussed origins of Scantrons	Did not explore improvements or psychological aspects
16	2015	Patel, R., Sanghavi, S., Gupta, D., Raval, M. S.	Mobile OMR system development	Proposed a low-cost mobile-based OMR system	Did not integrate sentiment analysis
17	2019	Xu, Y. et al.	Deep learning for scene text detection	Proposed a deep direction field for irregular text recognition	No direct OMR application
18	2019	Wang, W.	Shape-	Used a progressive	Lacked relevance to

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
		et al.	robust text detection	scale expansion network	OMR-based applications
19	2019	Huang, R. et al.	Image blur classification and removal	Focused on enhancing image clarity	Did not integrate with OMR processes
20	2022	Sumady, O.O. et al.	Optical text recognition from distorted images	Reviewed OCR advancements for distorted text	No dedicated OMR functionality
21	2021	Zhu, Y. et al.	Arbitrary-shaped text detection	Proposed Fourier contour embedding	Lacked application to OMR
22	2015	Ren, S. et al.	Faster R-CNN for object detection	Improved real-time object recognition	Not tailored for OMR applications
23	2016	Redmon, J. et al.	YOLO object detection	Proposed a unified real-time detection system	No OMR-specific developments
24	2020	Cao, L. et al.	OMR text detection	Applied CTPN and enhanced YOLOv3	Focused on handwriting rather

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
				for student exercise recognition	than bubble marks
25	2017	Liao, M. et al.	Text detection using deep neural networks	Introduced a fast text detector	No direct link to OMR development
26	2021	Dai, P. et al.	Scene text detection	Proposed progressive contour regression	Did not address OMR-based improvements
27	2020	Li, Y. et al.	Psoriasis severity evaluation	Used deep learning for medical image recognition	No relevance to OMR analysis
28	2020	Liao, M. et al.	Scene text detection with binarization	Improved real-time text detection	Did not apply to OMR
29	2020	Liu, Y. et al.	Figure and caption extraction	Proposed an instance segmentation model	No direct application to OMR
30	2019	Liu, J. et al.	Pyramid mask text detector	Proposed an end-to-end trainable model	Lacked focus on OMR

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
31	2016	Shi, B. et al.	Sequence recognition for scene text	Applied neural networks for text analysis	No focus on OMR systems
32	2017	Yin, F. et al.	Scene text recognition with convolutional models	Developed sliding convolutional character models	Not specifically for OMR
33	2020	Yousef, T. & Janicke, S.	Text alignment visualization	Surveyed visualization techniques	No relevance to OMR research
34	2016	He, P. et al.	Deep convolutional sequences for text reading	Applied deep learning for scene text analysis	No direct OMR application
35	2019	Luo, C. et al.	MORAN attention network for text recognition	Used multi-object rectified attention for OCR	Lacked OMR-specific enhancements
36	2018	Shi, B. et	ASTER:	Developed an	No integration with

Reference No.	Year	Author Name	Focus Area	Key Contributions	Research Gap
		al.	Attentional scene text recognizer	adaptable text recognition system	OMR
37	2015	Karunanayake, N.	OMR sheet evaluation using webcam	Proposed template matching for answer sheet analysis	Did not use advanced deep learning techniques
38	2019	Rasiq, G. R. I., Al Sefat, A., Hasnain, M. F.	Mobile-based MCQ answer analysis	Developed a mobile system for MCQ evaluation	Lacked AI-driven OMR enhancement
39	2019	Afifi, M. & Hussain, K. F.	Flexible MCQ evaluation using image classification	Improved adaptability in MCQ-based testing	No sentiment analysis component
40	2021	Jingyi, T., Hooi, Y. K., Bin, O. K.	Enhanced OMR answer matching precision	Applied image processing techniques for accuracy	No deep learning integration

2.3 CHALLENGES

a) **Inability to Detect Student Emotions**

Traditional OMR systems only recognize filled responses, failing to capture students' confidence, stress, or confusion during exams. This lack of emotional insight limits educators' ability to offer personalized support.

b) **Inaccurate Assessments**

Without sentiment analysis, assessments rely solely on correct or incorrect answers, overlooking psychological factors that may affect performance. This can lead to unfair evaluations, especially for students under stress.

c) **Manual Effort and Subjectivity**

Teachers and researchers must rely on subjective interpretations or additional surveys to gauge student emotions, making the process time-consuming and prone to human error.

d) **Limited Insights in Surveys and Research**

In market research and institutional feedback, OMR alone cannot interpret respondent sentiments, restricting organizations from fully understanding public opinion, customer satisfaction, or employee engagement.

e) **Inefficiency in Large-Scale Data Processing**

Traditional methods require separate tools for sentiment analysis, increasing processing time and reducing efficiency in handling large datasets from exams, surveys, or research studies.

f) **Missed Opportunities for Personalized Learning**

Without detecting emotional states, educators cannot tailor learning approaches based on student needs, missing chances to improve engagement and performance.

2.4 SOLUTIONS

a) **Emotion-Aware Assessments**

By detecting stress, confidence, or confusion in student responses, educators can gain deeper insights into student performance beyond correct and incorrect answers, leading to more supportive and fair assessments.

b) **Automated Sentiment Analysis in Exams**

The integration of sentiment analysis with OMR automates the detection of emotional states, eliminating the need for manual interpretation and making assessments more efficient and data-driven.

c) **Improved Accuracy in Evaluations**

Identifying patterns in marking behavior helps minimize misinterpretations due to stress or fatigue, ensuring a more accurate representation of student knowledge and capabilities.

d) **Enhanced Large-Scale Data Processing**

The combined system streamlines the processing of thousands of responses, reducing the time and resources needed for grading, surveys, and research analysis.

e) **Better Exam Conditions and Student Support**

Institutions can use sentiment-based insights to improve exam environments by addressing factors like lighting, fatigue, and stress-inducing conditions, leading to better student performance.

f) **Advanced Insights for Market Research and Surveys**

Businesses and research organizations can analyze customer or employee sentiments from survey responses, leading to more informed decision-making based on emotional trends.

g) **Personalized Learning and Feedback**

Educators can tailor teaching strategies by identifying students who require additional support, enabling adaptive learning techniques that enhance engagement and performance.

2.5 THEORY

This Research is Based on the Following Assumptions:

a) **Accuracy of Sentiment Detection**

   It is assumed that the proposed OMR system can accurately detect stress, confidence, or confusion based on variations in bubble markings.

b) **Consistency in Marking Patterns**

   Students exhibit consistent marking behaviors under different emotional states, allowing sentiment analysis to identify patterns effectively.

c) **Impact of Stress on Marking Accuracy**

   It is assumed that stress, fatigue, or external factors influence the way students fill out bubble sheets, affecting response accuracy.

d) **Survey Responses Reflect Real-World Behavior**

   The data collected from the seven variants of bubble sheet formats accurately represent students' marking tendencies in real exam conditions.

e) **OMR Sentiment Analysis Enhances Assessment Fairness**

   The integration of sentiment analysis with OMR improves evaluation accuracy and fairness by accounting for psychological factors that may impact student performance.

f) **Generalizability of Findings**

   The insights gained from this research can be applied across various educational institutions and assessment frameworks, ensuring broader applicability.

CHAPTER # 03

REQUIREMENT SPECIFICATION & DESIGN

3.1 PROPOSED SYSTEM AND REQUIREMENT SPECIFICATION

The proposed system performs sentiment analysis on Optical Mark Recognition (OMR) sheets. The main goal is to extract bubble responses from scanned OMR sheets, classify them into meaningful categories, and then predict the respondent's sentiment using engineered features such as area and solidity. The sentiment labels used are "Confidence," "Confusion," and "Stress."

3.2 FUNCTIONAL REQUIREMENTS

a) Data Acquisition – Process scanned OMR sheets, extract bubbles, and classify them as filled, empty, or containing letters.
b) Circle Extraction – Use OpenCV and Hough Circle Transform to detect and extract circular regions.
c) Classification – Differentiate bubbles based on pixel intensity using threshold-based classification.
d) Feature Extraction – Compute area and solidity, assigning sentiment labels based on predefined thresholds.
e) Exploratory Data Analysis (EDA) – Structure datasets, visualize distributions, and analyze feature correlations.
f) Model Training & Evaluation – Train a Logistic Regression model and assess performance using accuracy, classification reports, and confusion matrices.
g) Prediction & Integration – Implement single-image prediction and develop an automated pipeline for sentiment analysis.

3.3 NON-FUNCTIONAL REQUIREMENTS

a) Performance – Optimize image processing and feature extraction while enhancing model performance through adaptive parameter tuning.
b) Scalability – Enable batch processing and seamless integration with additional sentiment labels or classifiers.
c) Robustness – Ensure accuracy across varying image qualities and lighting conditions while implementing error handling for incomplete inputs.

3.4 USABILITY REQUIREMENTS

a) The system should have a user-friendly interface for uploading OMR sheets.
b) Clear visualization of extracted data and prediction results.
c) Minimal manual intervention required for processing.

3.5 EFFICIENCY REQUIREMENTS

a) Optimized circle detection and feature extraction using OpenCV.

b) Efficient classification using Logistic Regression.

c) Fast processing to handle real-time analysis if needed.

3.6 CONSISTENCY

a) Consistent handling of different OMR sheet formats.

b) Uniform classification rules based on predefined thresholds.

3.7 COMMON ENTITY RELATIONSHIP DIAGRAM SYMBOLS

Table #2: Common entity relationship diagram symbols for sentiment analysis according to OMR

ENTITY SYMBOL	NAME	DESCRIPTION
●	Strong Filling	Fully filled bubble, indicating high confidence.
◑	Moderate Filling	Partially filled bubble, indicating confusion.
○	No Fill	Empty bubble, indicating stress or no response.

3.8 EXTERNAL ENTITY

- **Users:** Upload scanned OMR sheets for sentiment analysis.
- **System Components:** Image processing module, feature extraction module, classification model.

a) PROCESS

- **Image Preprocessing:**
    - Convert scanned OMR sheets to grayscale.
    - Apply Gaussian blur and thresholding techniques.
- **Circle Detection and Extraction:**
    - Use Hough Circle Transform to detect bubbles.
    - Extract and save individual bubbles.
- **Classification and Feature Extraction:**
    - Categorize bubbles based on intensity.
    - Compute area and solidity for filled bubbles.
- **Model Training & Prediction:**
    - Train a logistic regression model on extracted features.
    - Predict sentiment labels for new OMR sheets.

b) DATA STORE

- The extracted features (area, solidity, intensity) are stored in a structured dataset.
- Labeled data is saved for model training and evaluation.
- Results are formatted into a report for users.

c) DATA FLOW

- **Input:** Scanned OMR sheets.
- **Processing:** Image preprocessing → Circle detection → Feature extraction → Sentiment prediction.
- **Output:** Classified sentiment labels displayed in a report.

CHAPTER # 04

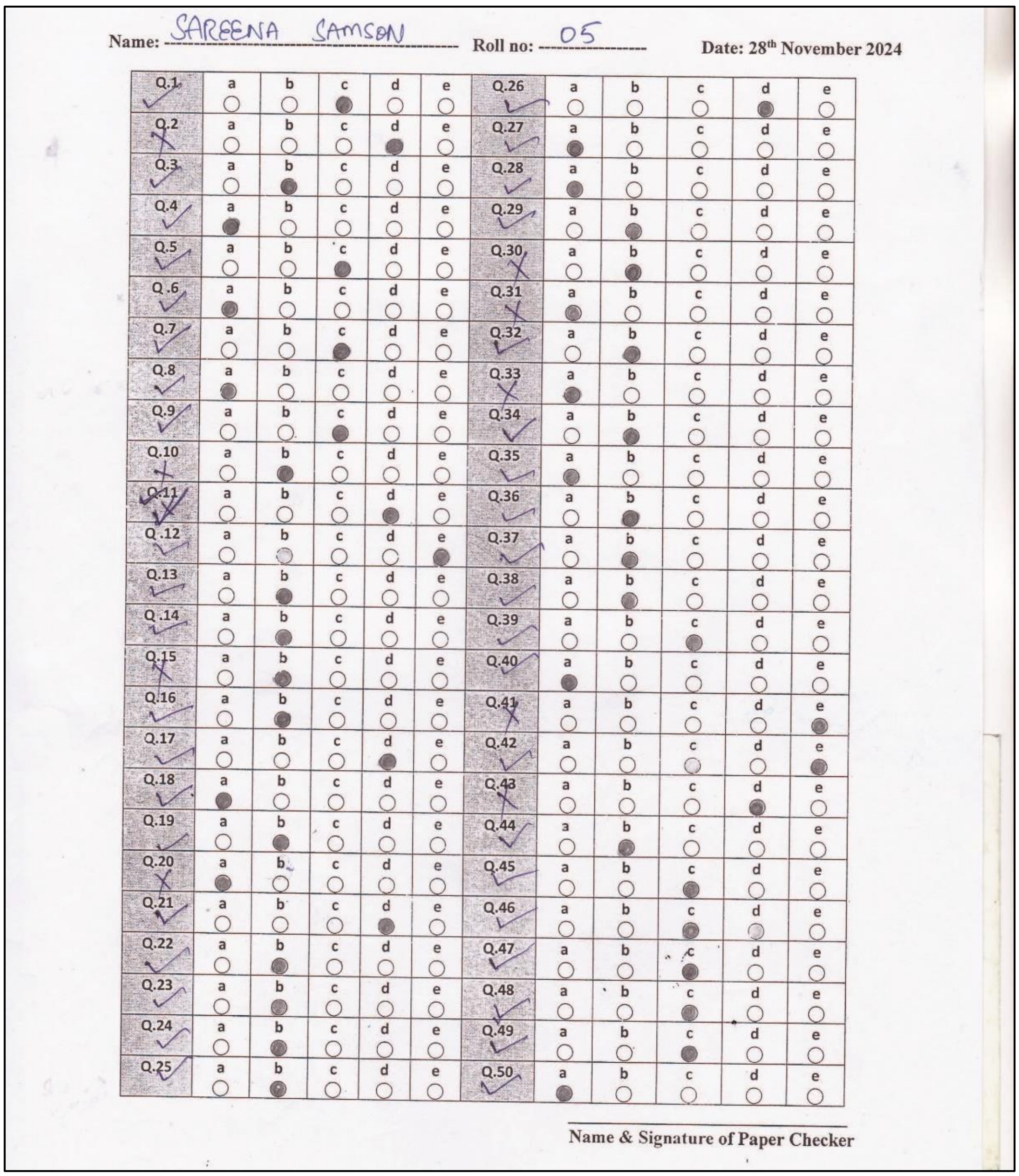
METHODOLOGY

4.1 DATA ACQUISITION AND PREPARATION

4.1.1. DATASET DESCRIPTION

The study utilizes a dataset comprising approximately 150 pages of Optical Mark Recognition (OMR) sheets. Each page is a scanned document containing multiple circular bubbles, which serve as the primary regions of interest for sentiment analysis. These sheets, originally in image format (JPG/PNG), are acquired from real-world examination or survey settings.

One of the pictures in the dataset is shown in Figure 1.1.



4.1.2 PRE-PROCESSING PIPELINE

The overall processing pipeline consists of the following stages:

1. Circle Extraction: Detect and extract individual circular regions (bubbles) from the OMR sheets
2. Circle Classification: Differentiate among filled bubbles, empty bubbles, and bubbles containing letters.
3. Feature Extraction and Labeling: Compute discriminative features (e.g., area and solidity) and assign sentiment labels based on these features.
4. Exploratory Data Analysis (EDA) and Statistical Testing: Assess the quality of the extracted features and their distributions.
5. Model Training and Evaluation: Train a logistic regression classifier using the engineered features and evaluate its performance.
6. Pipeline Integration: Seamlessly integrate all stages into a unified processing framework.

4.2. CIRCLE EXTRACTION

4.2. 1. EXTRACTION USING HOUGH CIRCLE TRANSFORM

The extraction phase is implemented in the script Bubbles.py. The primary steps are as follows:

Image Preprocessing:

Each OMR sheet is loaded and converted to grayscale. Gaussian blurring (with a 5×5 kernel) is applied to reduce noise and smooth the image, which is essential for robust circle detection.

a) Circle Detection:

The Hough Circle Transform (via cv2.HoughCircles) is employed to detect circular shapes. The transform parameters are carefully selected: dp (Inverse Resolution Ratio): Set to 1.2 to balance accuracy and computational load.

b) **Min-Distance:**

A minimum distance of 100 pixels is enforced between detected circle centers to avoid overlapping detections.

c) Edge Thresholds:

The parameters param1 (set to 500) and param2 (set to 28) control the sensitivity of the underlying Canny edge detector and the accumulator threshold, respectively.

Radius Constraints: Only circles with radii between 18 and 23 pixels are considered, based on prior empirical observations.

d) Circle Extraction and Saving:

For each detected circle, the bounding box is computed and the corresponding region of interest (ROI) is cropped. These cropped regions are then saved as individual image files for further analysis.

4.3. CIRCLE CLASSIFICATION

4.3.1. CLASSIFICATION FRAMEWORK

The extracted circle images are processed by anotherBubbles.py to classify them into three distinct categories:
Filled Circles: Bubbles that are sufficiently marked.
Empty Circles: Unmarked bubbles with high average intensity.
Letter Circles: Regions containing printed or handwritten characters.

4.3.2. INTENSITY-BASED CLASSIFICATION

Each circle image is analyzed by computing its average pixel intensity. The following heuristic thresholds are applied:
Empty Circles: Images with an average intensity greater than 200.
Letter Circles: Images with an average intensity between a dynamically adjusted threshold (e.g., around 160) and 200.
Filled Circles: Images with an average intensity below the dynamic threshold.
The classification results in the segregation of images into separate folders for filled, empty, and letter circles, thereby ensuring that only filled bubbles are forwarded for sentiment analysis.

4.4 FEATURE EXTRACTION AND LABELING

4.4.1. FEATURE ENGINEERING

For each filled bubble, features are extracted using the following procedure (implemented in FYP.ipynb):

a) Preprocessing:

Each image is re-read in grayscale and thresholded using an inverse binary threshold. This step accentuates the filled regions.

b) Contour Detection:

Contours are identified using cv2.findContours. For each detected contour, the following features are computed:

c) **Area (A):** The number of pixels enclosed by the contour.

d) Bounding Box and Solidity (S):

The bounding box is determined, and the solidity is computed as:
S = \frac{A}{w \times h}

4.5. LABEL ASSIGNMENT

We have extracted three parameters from the survey we have conducted.
"**https://forms.office.com/r/XrMfKpCzH9**"
1ST Page

- Your Name (Optional)

- Profession/Occupation (Optional)

- Age Group
    - Under 18
    - 18-24
    - 25-34
    - 35-44
    - 45-54
    - 55-64
    - 65 and above

2ND Page


Select the option that best reflects your emotion based on your feelings *

- Confidence
- Confusion
- Stress

Fig # 1.2

fig
#

1.3

fig # 1.4

Fig # 1.5

fig # 1.6

After getting more than 50 responses, we analyzed the data using statistical methods and found that majority of the people who asses the labels were somehow connected to the solidity and area and correlation of these two around 0.72, hence we have assigned these parameters to our dataset.

Based on the computed solidity, each filled bubble is assigned one of three sentiment labels with following parameters:

a) Confidence > 0.400
b) Confusion > 0.178
c) Stress > 0.100

Bubbles not meeting these criteria are discarded, ensuring a clean and relevant dataset for training.

4.6. EXPLORATORY DATA ANALYSIS AND STATISTICAL VALIDATION

4.6.1 DESCRIPTIVE ANALYSIS

The extracted features (Area and Solidity) along with their corresponding labels are consolidated into a Pandas Data Frame. Summary statistics (mean, median, standard deviation, etc.) are computed to understand the central tendencies and variability of the data.

4.6.2 DATA VISUALISATION

a) Histograms and KDE Plots:

Histograms with kernel density estimation overlays are used to visualize the distribution of Area and Solidity.

b) Scatter Plots:

A scatter plot of Area versus Solidity, with points color-coded by label, provides a visual assessment of class separability.

c) Correlation Heatmaps:

A correlation matrix is computed and visualized to examine the relationship between the features.

4.6.3 STATISTICAL TESTING

a) Normality Tests:

The Shapiro-Wilk test is applied to both Area and Solidity to evaluate whether they follow a normal distribution.

b) **ANOVA:**

One-way ANOVA tests are conducted to determine whether the differences in Area and Solidity across the three sentiment classes are statistically significant. This helps validate the discriminative power of the selected features.

4.7 FEATURE SCALING AND HANDLING CLASS IMBALANCE

4.7.1 SCALING

The features are standardized using the Standard Scaler to ensure they contribute equally to the model training process:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

4.7.2 CLASS IMBALANCE ANALYSIS

The distribution of the three sentiment classes is visualized using count plots. Any imbalance in the dataset is noted, as it could bias the learning process. Future iterations could involve applying oversampling or class weighting if severe imbalance is observed.

4.8 . MODEL TRAINING AND EVALUATION

4.8.1 DATA SPLITTING

The dataset is partitioned into training and testing sets using an 80:20 split to enable robust evaluation of the model's generalization capability.

4.8.2 CLASSIFIER SELECTION

A Logistic Regression model is chosen for its simplicity and interpretability. The model is configured with a one-vs-rest strategy to handle the multi-class problem and is set to run with a maximum of 1000 iterations to ensure convergence.

4.8.3 TRAINING AND EVALUATION METRICS

a) Training:

The model is trained on the scaled features with the encoded sentiment labels.

b) Performance Evaluation:

The trained model is evaluated using: Accuracy score instance of Scikit Learn

c) **Accuracy: 93%**

Overall percentage of correctly classified instances.93

Classification Report: Detailed metrics including precision, recall, and F1-score for each sentiment class.

d) **Confusion Matrix:**

A heatmap visualizes misclassifications between classes.

Precision-Recall Curves: These curves are plotted for each class to assess the trade-off between precision and recall across different thresholds.

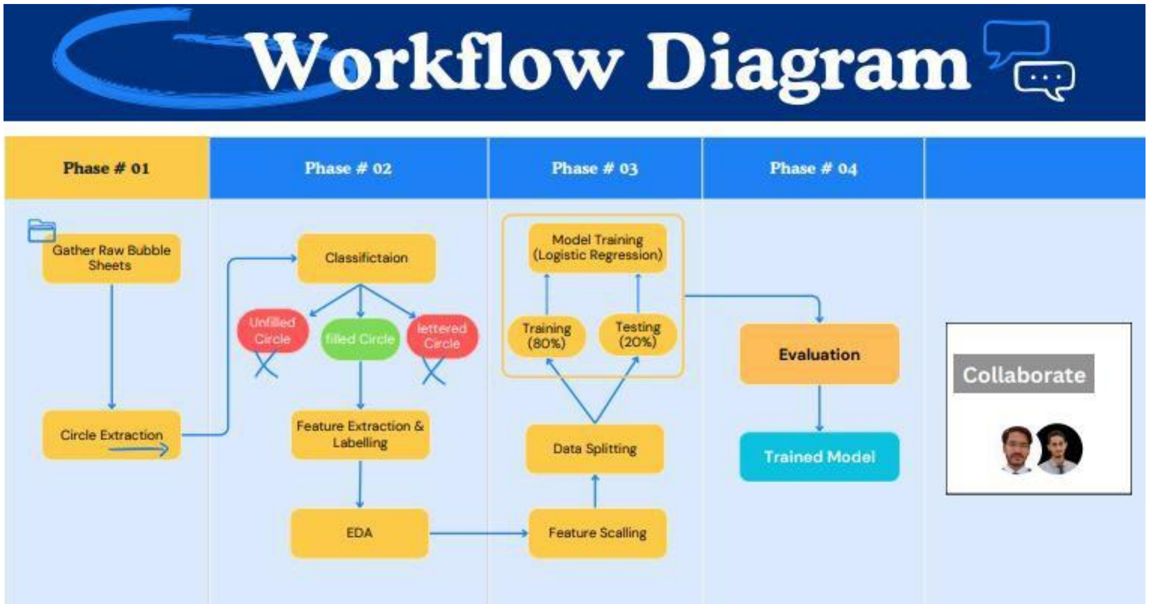


Fig # 1.7

4.9 PREDICTION AND PIPELINE INTEGRATION

4.9.1 SINGLE IMAGE PREDICTION

A dedicated function (e.g., result) demonstrates the end-to-end prediction on a single image. The process involves:

1. Preprocessing the test image.
2. Extracting the feature (Area and Solidity) using contour analysis.
3. Scaling the feature using the previously fitted Standard Scaler.
4. Generating a prediction with the trained Logistic Regression model.
5. Visualizing the test image along with the predicted sentiment label.

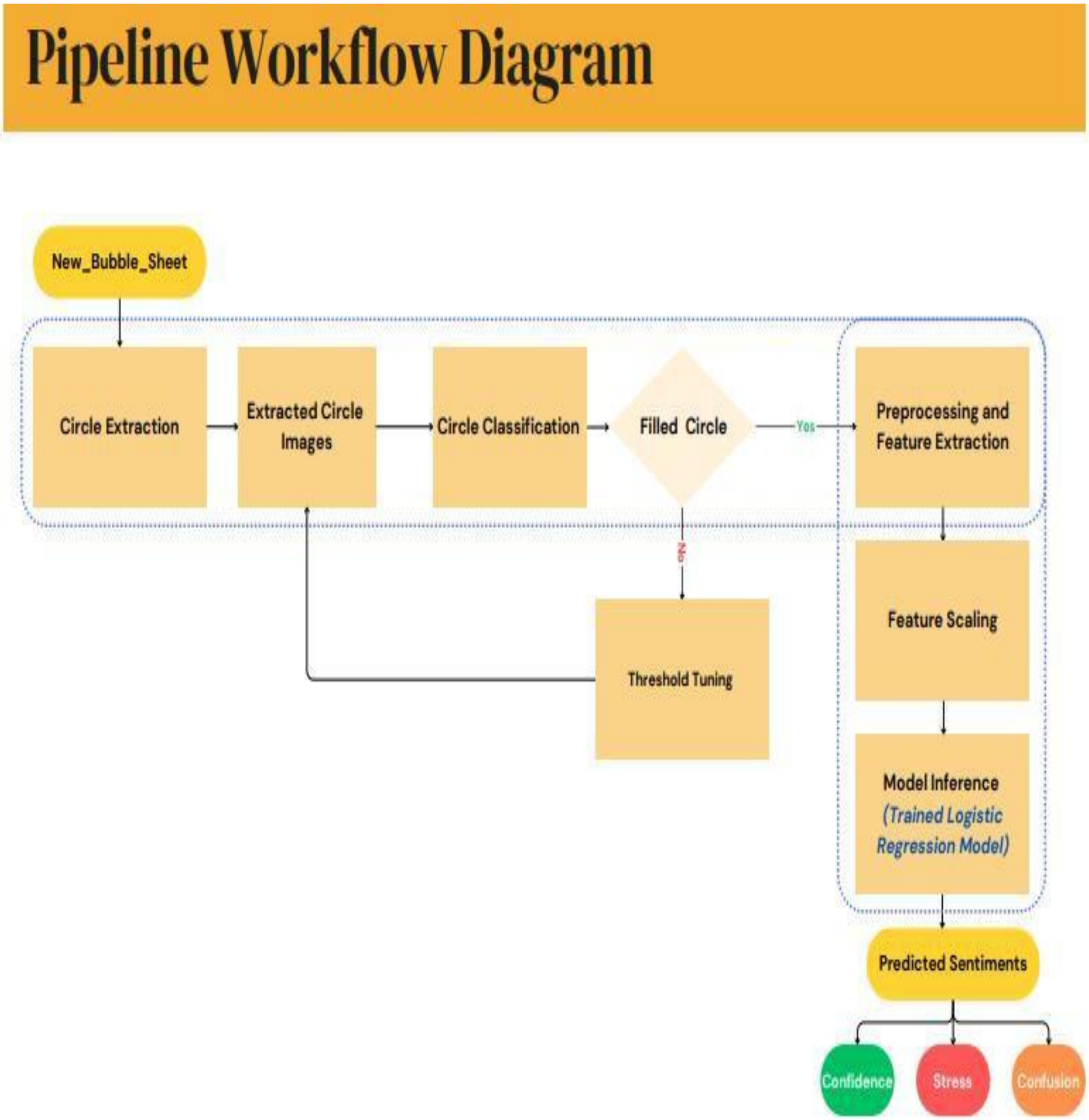
4.9.2 END-TO-END PIPELINE

The entire workflow is integrated into a pipeline that automates:

1. Circle Extraction: Using Bubbles.py to extract all bubbles from a new OMR sheet.
2. Circle Classification: Using anotherBubbles.py to separate filled bubbles from others.
3. Feature Extraction and Data Preparation: Processing filled bubbles to compute features and assign labels.
4. Model Inference: Running the trained model to predict sentiment labels for the extracted and processed bubbles.

This pipeline is modular and can be deployed as a standalone application or integrated into a web interface (e.g., using Streamlit) for real-time processing and visualization.

Fig # 1.8


Pipeline Workflow Diagram
New_Bubble_Sheet
Circle Extraction
Extracted Circle Images
Circle Classification
Filled Circle
Yes
No
Preprocessing and Feature Extraction
Feature Scaling
Threshold Tuning
Model Inference
(Trained Logistic Regression Model)
Predicted Sentiments
Confidence
Stress
Confusion

CHAPTER # 05

REQUIREMENT SPECIFICATION & DESIGN

5.1 PROPOSED SYSTEM AND REQUIREMENT SPECIFICATION

The proposed system performs sentiment analysis on Optical Mark Recognition (OMR) sheets. The main goal is to extract bubble responses from scanned OMR sheets, classify them into meaningful categories, and then predict the respondent's sentiment using engineered features such as area and solidity. The sentiment labels used are "Confidence," "Confusion," and "Stress."

5.2 PROJECT ARCHITECTURE

The system follows a modular architecture comprising image preprocessing, feature extraction, classification, and sentiment prediction modules. Each module is designed to efficiently process OMR sheet data and produce reliable sentiment predictions.

The methodology involves multiple stages:

a) **Data Acquisition and Preparation**: Collecting and pre-processing scanned OMR sheets.

b) **Circle Extraction**: Identifying and extracting individual bubbles using the Hough Circle Transform.

c) **Circle Classification**: Categorizing bubbles into filled, empty, or letter-containing.

d) **Feature Extraction and Labeling**: Computing key features such as area and solidity for classification.

e) **Exploratory Data Analysis and Statistical Testing**: Analyzing feature distributions and validating classification criteria.

f) **Model Training and Evaluation**: Using a Logistic Regression classifier for sentiment analysis.

g) **Pipeline Integration**: Creating an end-to-end automated system for real-time OMR sheet analysis.

5.3 DESIGN CUSTOMIZATION

a) The system is adaptable to various OMR sheet formats.

b) Threshold values for classification can be fine-tuned for different datasets.

c) The feature extraction process allows customization to include additional features if needed.

d) Image pre-processing steps such as noise reduction and edge detection can be modified for improved accuracy.

e) The classification model can be swapped for a more advanced machine learning technique if required.

5.4 BACKEND

5.4.1 BACKEND FRAMEWORK AND LANGUAGE

- Implemented in Python using OpenCV for image processing and scikit-learn for machine learning.
- Uses Pandas and NumPy for data handling and feature computation.
- Utilizes Standard Scaler for feature normalization and Logistic Regression for sentiment classification.

5.4.2 SURVEY BASED RESEARCH

- In our project, we utilize Google Forms to collect survey responses, which are then processed for sentiment analysis. Statistical analysis helps identify patterns and trends in the sentiment data, which can be used for decision-making.

5.4.3 COMMAND LINE INTERFACE (CLI)

After collecting the data via Google Forms, statistical analysis is performed using a CLI interface. Users can input commands and view the output directly on the terminal, displaying relevant statistical metrics such as sentiment distributions, frequency counts, and sentiment trends.

5.4.4 CODING SECTION:

a) **IMPORTS AND LIBRARIES**

```
import os import cv2 import numpy as np import pandas
as pd import seaborn as sns import matplotlib.pyplot as
plt
from sklearn.preprocessing import LabelEncoder, StandardScaler from sklearn.model_selection import
train_test_split from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, precision_recall_curve from scipy
import stats
```

b) **IMAGE PREPROCESSING AND FEATURE EXTRACTION**

```
def preprocess_image(image_path):     img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)     if img is None:
raise FileNotFoundError(f"Image not found or cannot be loaded:
{image_path}")
    _, binary = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY_INV)     return img, binary

def extract_features(binary_image):     contours, _ = cv2.findContours(binary_image, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)     features = []     for contour in contours:
        area = cv2.contourArea(contour)     x, y, w, h = cv2.boundingRect(contour)
solidity = area / (w * h)     features.append((area, solidity))     return features

def assign_label(solidity):     if solidity > 0.40:
        return "Confidence"     elif solidity > 0.178:
return "Confusion"     elif solidity > 0.01:
return "Stress"     else:     return None
```

c) LOAD AND PREPROCESS IMAGES

```
def load_and_preprocess_images(image_folder):    images = []    labels = []    filenames = []    for root, _, files in os.walk(image_folder):        for file in sorted(files):            if file.lower().endswith(('.jpg', '.png')):                image_path = os.path.join(root, file)                img, binary = preprocess_image(image_path)            features = extract_features(binary)
                if not features:            print(f"Warning: No features found in {file}")
                for area, solidity in features:                label = assign_label(solidity)
if label is None:
                    continue
                images.append((area, solidity))
                labels.append(label)                filenames.append(file)

    return np.array(images), np.array(labels), filenames
```

d) LABEL SUMMARY

```
def label_summary(labels, filenames, label_encoder):    original_labels = label_encoder.inverse_transform(labels)

    label_counts = {        "Confidence": [],
        "Confusion": [],
        "Stress": []
    }

    for label, filename in zip(original_labels, filenames):

        label_counts[label].append(filename)

    print("Label Summary:")    for label, files in label_counts.items():        print(f"{label}: {len(files)} image(s)")        if len(files) > 0:            print(f"  {', '.join(files[:5])}...")
```

e) EXPLORATORY DATA ANALYSIS (EDA)

```
dataset_path = r'C:\Users\Laptop\extracted_bubbles\filled_circles' images, labels, filenames =
load_and_preprocess_images(dataset_path)

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(labels)

areas = [feature[0] for feature in images] solidities = [feature[1] for feature in images]

df = pd.DataFrame({ 'Filename': filenames,
    'Area': areas,
    'Solidity': solidities,
    'Label': labels
})

print(df.describe()) print(df.head()) print("\n")
label_summary(y_encoded, filenames, label_encoder)

plt.figure(figsize=(8, 6))
sns.histplot(df['Area'], kde=True, color='blue') plt.title('Distribution of Area')
plt.xlabel('Area') plt.ylabel('Frequency') plt.show()

plt.figure(figsize=(8, 6))
sns.histplot(df['Solidity'], kde=True, color='green') plt.title('Distribution of Solidity')
plt.xlabel('Solidity') plt.ylabel('Frequency') plt.show()
```

```
            Area     Solidity count  11083.000000  11083.000000 mean     397.075702     0.353870 std      535.943015
0.257893 min     0.500000    0.010417 25%     3.000000    0.125000
50%    23.500000    0.250000 75%    1082.500000    0.665654 max    1622.500000    0.865313

          Filename   Area  Solidity      Label
0    circle100_104.png  1251.5  0.782188  Confidence
1    circle100_106.png  1261.0  0.768902  Confidence
2    circle100_109.png  1250.0  0.743605  Confidence
3 circle100_118.png 1264.0 0.734030 Confidence
4 circle100_125.png 1292.5 0.750581 Confidence
Label Summary:
Confidence: 3667 image(s)
circle100_104.png, circle100_106.png, circle100_109.png,
circle100_118.png, circle100_125.png...
Confusion: 3559 image(s)
circle101_223.png, circle102_119.png, circle102_119.png,
circle102_119.png, circle102_127.png...
Stress: 3857 image(s)
circle102_119.png, circle102_119.png, circle102_127.png,
3    circle102_170.png, circle102_170.png...
```

CHAPTER # 06

RESULT

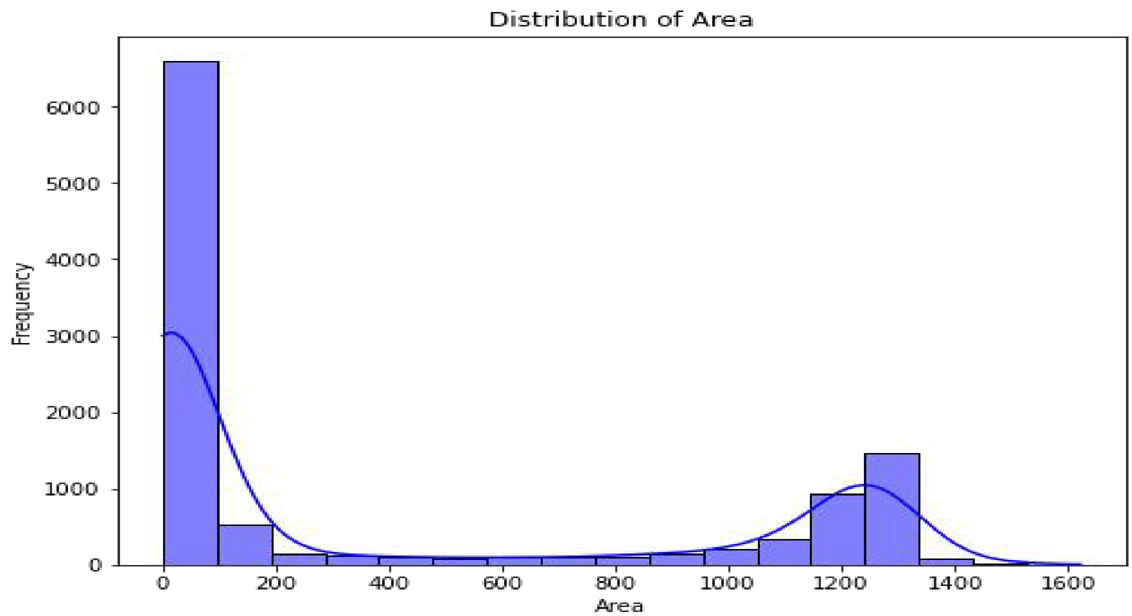


Fig # 01: Represents the distribution of data

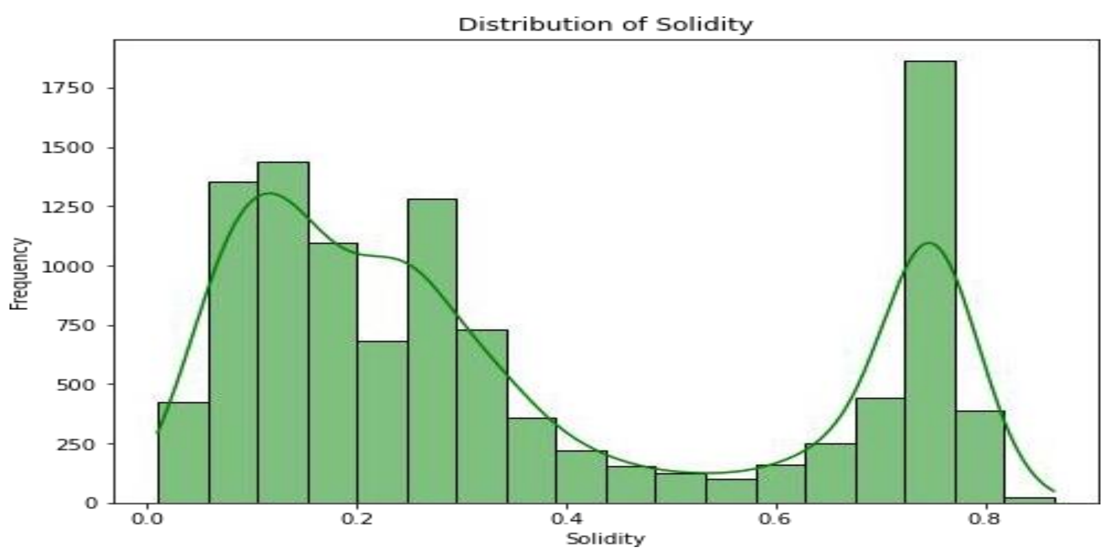


Fig # 02: Represents the distribution of Solidity

6.1 BIVARIATE ANALYSIS

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Area', y='Solidity', hue='Label', data=df,
palette='Set1')
plt.title('Scatter Plot of Area vs Solidity by Label') plt.xlabel('Area') plt.ylabel('Solidity')
plt.legend() plt.show()

correlation_matrix = df[['Area', 'Solidity']].corr() plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap of Area and Solidity') plt.show()
```

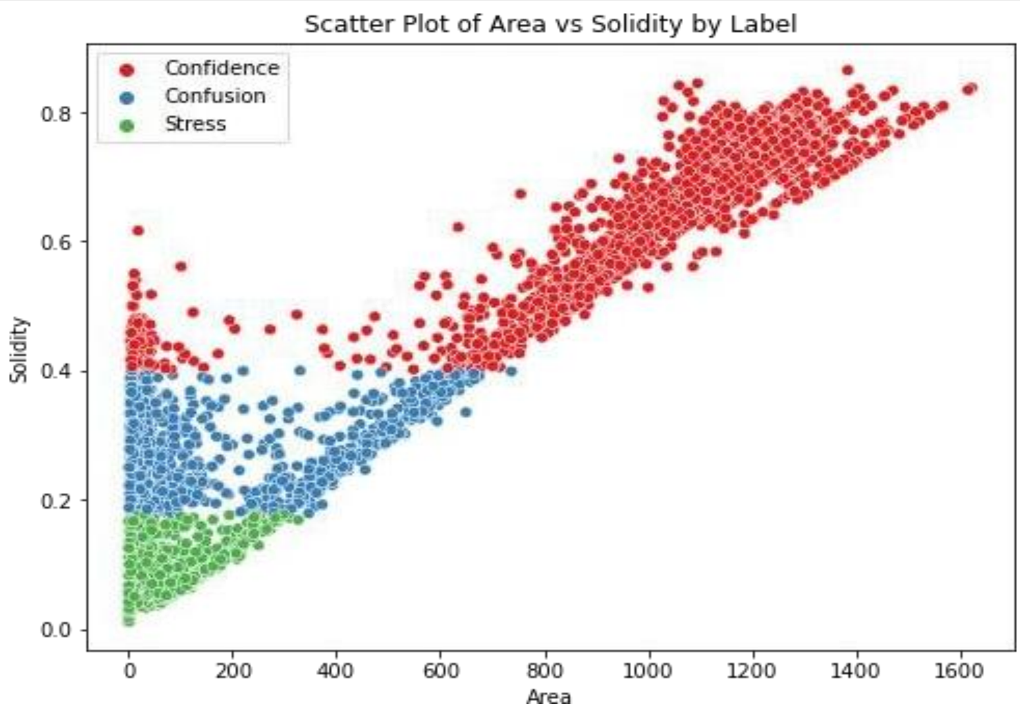
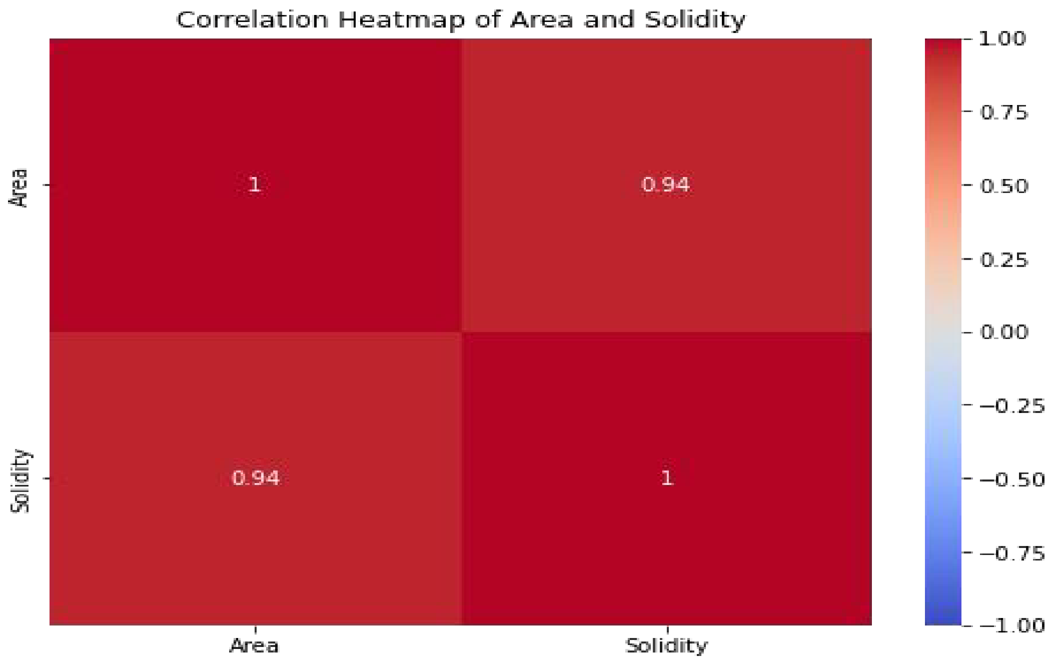


Fig # 03: Scatter plot of area vs solidity



6.2 STATISTICAL TESTS

```
stat_area, p_area = stats.shapiro(df['Area']) stat_solidity, p_solidity = stats.shapiro(df['Solidity'])

print(f"Shapiro-Wilk test for 'Area': Statistic = {stat_area}, p-value
= {p_area}")
print(f"Shapiro-Wilk test for 'Solidity': Statistic = {stat_solidity}, p-value = {p_solidity}")

anova_area = stats.f_oneway(df[df['Label'] == 'Confidence']['Area'],                    df[df['Label'] == 'Confusion']['Area'],
df[df['Label'] == 'Stress']['Area'],                    df[df['Label'] == 'Not Relevant']['Area']) print(f"ANOVA test for 'Area' by
Label: {anova_area}")

anova_solidity = stats.f_oneway(df[df['Label'] == 'Confidence'] ['Solidity'],


                    df[df['Label'] == 'Confusion']
['Solidity'],
                    df[df['Label'] == 'Confusion']
['Solidity'],
```

```
                    df[df['Label'] == 'Stress']
['Solidity'],
                    df[df['Label'] == 'Not Relevant']
['Solidity'])
print(f"ANOVA test for 'Solidity' by Label: {anova_solidity}")
```

```
Shapiro-Wilk test for 'Area': Statistic = 0.6846106052398682, p-value
= 0.0
Shapiro-Wilk test for 'Solidity': Statistic = 0.8456391096115112, p-
value = 0.0
 ANOVA test for 'Area' by Label: F_onewayResult(statistic=nan,
pvalue=nan)
ANOVA test for 'Solidity' by Label: F_onewayResult(statistic=nan, pvalue=nan)

c:\Users\Laptop\anaconda3\lib\site-packages\scipy\stats\ morestats.py:1760: UserWarning: p-value may not be accurate for N >
5000.    warnings.warn("p-value may not be accurate for N > 5000.") c:\Users\Laptop\anaconda3\lib\site-
packages\scipy\stats\stats.py:3621: F_onewayBadInputSizesWarning: at least one input has length 0
warnings.warn(F_onewayBadInputSizesWarning('at least one input '
```

6.3 FEATURE SCALING

```
scaler = StandardScaler()
df[['Area', 'Solidity']] = scaler.fit_transform(df[['Area',
'Solidity']]) print(df.head())
          Filename      Area  Solidity       Label
0  circle100_104.png  1.594317  1.660911  Confidence
1  circle100_106.png  1.612043  1.609394  Confidence
2  circle100_109.png  1.591518  1.511297  Confidence
3  circle100_118.png  1.617641  1.474168  Confidence
4  circle100_125.png  1.670821  1.538347  Confidence
```

6.4 CLASS IMBALANCE

```
plt.figure(figsize=(8, 6))

sns.countplot(x='Label', data=df, palette='Set2') plt.title('Class Distribution') plt.show()
```

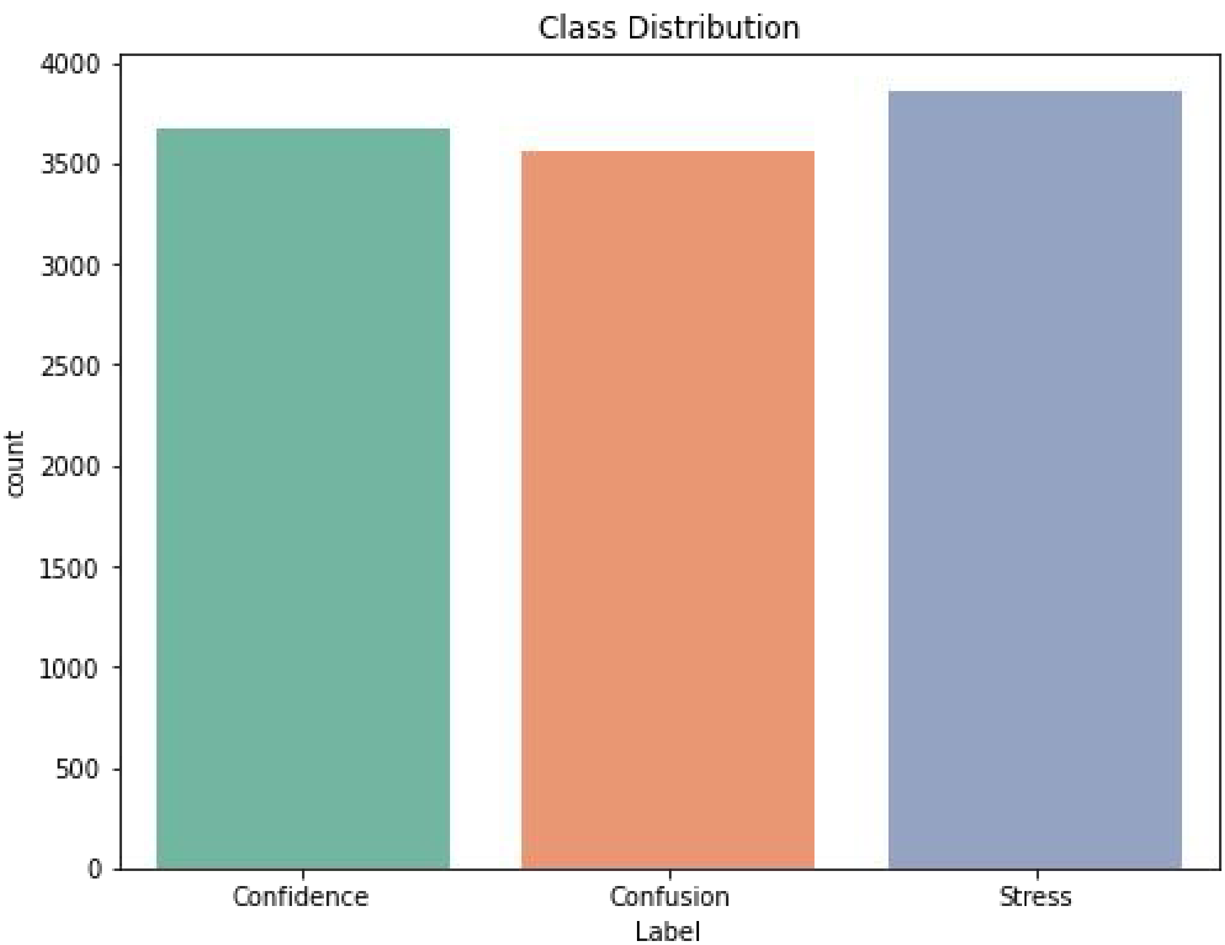
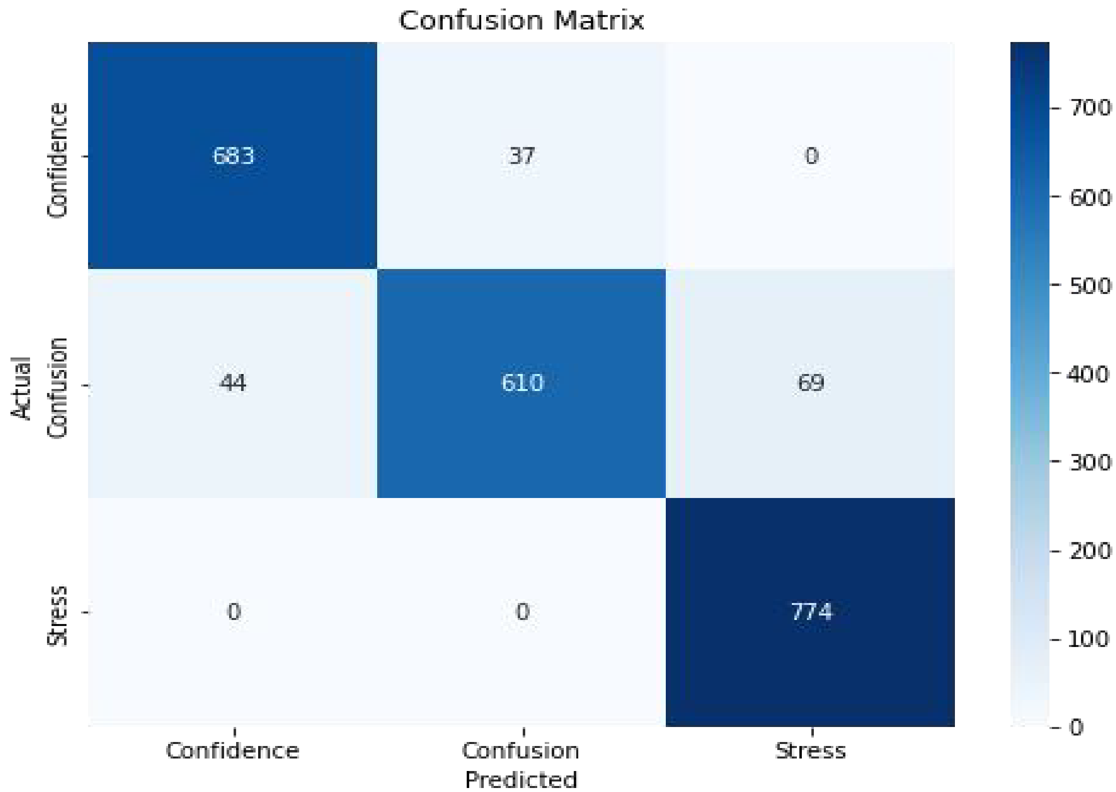


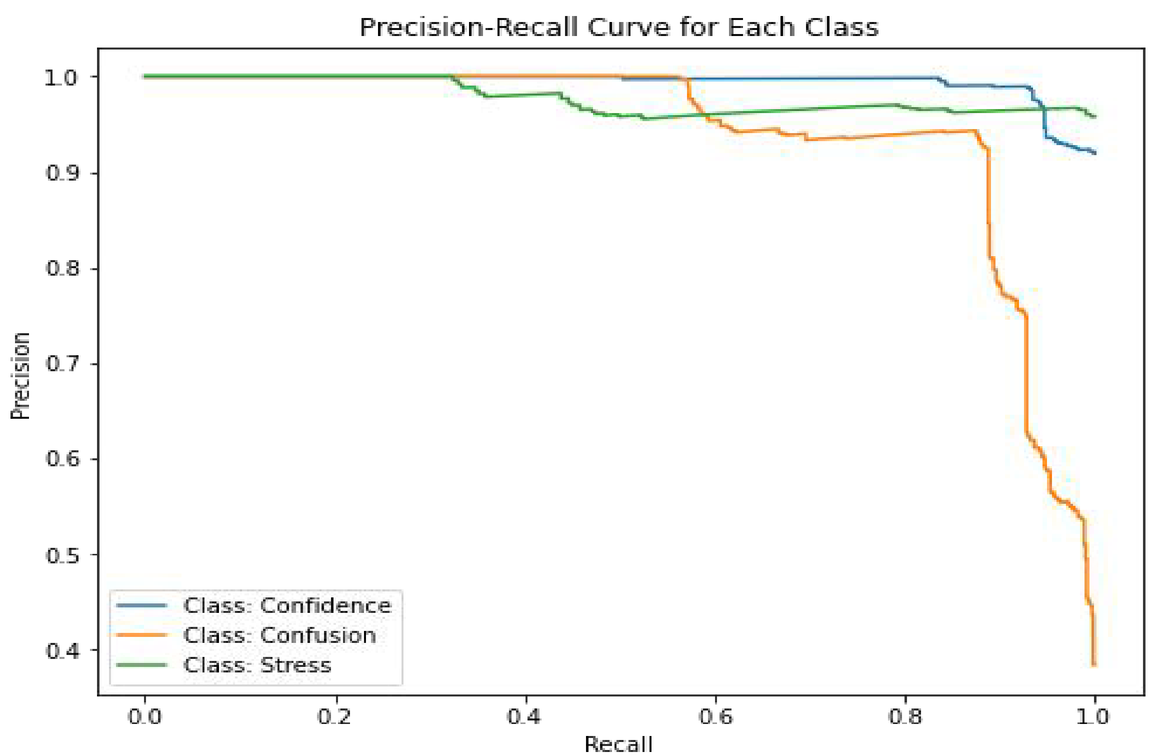
Fig # 04: Class Distribution

6.5 TRAIN MODEL AND EVALUATION

```
dataset_path = r'C:\Users\Laptop\extracted_bubbles\filled_circles' X, y, filenames =
load_and_preprocess_images(dataset_path)

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(images, y_encoded, test_size=0.2, random_state=42)

model = LogisticRegression(max_iter=1000, multi_class='ovr', solver='lbfgs')
model.fit(X_train, y_train)

predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions) report = classification_report(y_test,
predictions) print(f"Accuracy: {accuracy:.2f}")
```

```
print("Classification Report:\n", report)

cm = confusion_matrix(y_test, predictions) plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=label_encoder.classes_,
yticklabels=label_encoder.classes_) plt.title('Confusion Matrix') plt.xlabel('Predicted')
plt.ylabel('Actual') plt.show()

y_proba = model.predict_proba(X_test)

plt.figure(figsize=(8, 6)) for i, label in enumerate(label_encoder.classes_): precision, recall, _ =
precision_recall_curve(y_test == i, y_proba[:, i])
    plt.plot(recall, precision, label=f'Class: {label}')

plt.title('Precision-Recall Curve for Each Class') plt.xlabel('Recall') plt.ylabel('Precision')
plt.legend() plt.show()
Accuracy: 0.93

Classification Report:

              precision    recall  f1-score   support
0              0.94       0.95      0.94       720
1              0.94       0.84      0.89       723
2              0.92       1.00      0.96       774
    accuracy                        0.93      2217   macro avg      0.93      0.93      0.93      2217
weighted avg      0.93      0.93      0.93      2217
```


Confusion Matrix
Confidence
Confusion
Stress
Actual
683
37
0
44
610
69
0
0
774
Confidence
Confusion
Stress
Predicted
700
600
500
400
300
200
100
0


Precision-Recall Curve for Each Class
Precision
Recall
Class: Confidence
Class: Confusion
Class: Stress
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.0
0.2
0.4
0.6
0.8
1.0

6.6 PREDICTING LABEL FOR IMAGES

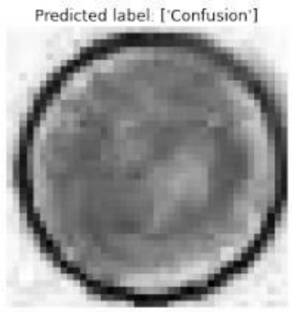
```
def result(test_path):

    # Preprocess the single image
    img, binary = preprocess_image(test_path)       features = extract_features(binary)

    if not features:          raise ValueError("No features found in the image.")

    # Assuming the first feature set is representative       area, solidity = features[0]

    # Scale the features
    scaled_features = scaler.transform([[area, solidity]])

    # Predict the label
    predicted_label_encoded = model.predict(scaled_features)
```

```
  predicted_label =

label_encoder.inverse_transform(predicted_label_encoded)


    print(f"Predicted label for the image '{test_path}':
{predicted_label[0]}")


    plt.imshow(img, cmap='gray')  # Show the image     plt.title(f"Predicted label: {predicted_label}")

    plt.axis('off')      plt.show()

from     Bubbles     import     extract_circles     from     anotherBubbles     import

classify_circles


test_folder = r'C:\Users\Laptop\test'

output_circles_folder = rf'{test_folder}\Extracted_Circles_from_OMR'


print("Starting circle extraction...") extract_circles(test_folder, output_circles_folder)


print("Starting circle classification...") classify_circles(output_circles_folder)
```

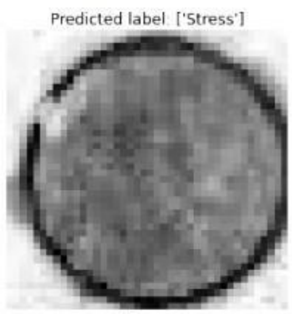
6.7 STARTING CIRCLE EXTRACTION

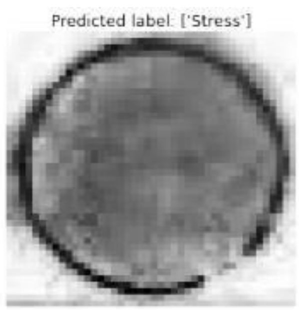
```
1 C:\Users\Laptop\test\1.jpg
Saved circle 1 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_1.png
Saved circle 2 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_2.png
Saved circle 3 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_3.png
Saved circle 4 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_4.png
Saved circle 5 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_5.png
Saved circle 6 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_6.png
Saved circle 7 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_7.png
Saved circle 8 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_8.png
Saved circle 9 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_9.png
Saved circle 10 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_10.png
Saved circle 11 to C:\Users\Laptop\test\Extracted_Circles_from_OMR\
circle2_11.png Extracted_Circles_from_OMR\filled_circles\circle4_92.png circle4_93.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_93.png circle4_94.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_94.png circle4_95.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_95.png circle4_96.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_96.png circle4_97.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_97.png circle4_98.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_98.png circle4_99.png classified as C:\Users\Laptop\test\
Extracted_Circles_from_OMR\empty_circles\circle4_99.png Circles classified and saved int o separate folders!
Empty circles: 552
Filled circles: 132, at C:\Users\Laptop\test\
Extracted_Circles_from_OMR\filled_circles
Letter circles: 15

for each_circle in os.listdir(filled):
    result(os.path.join(filled, each_circle))      # print(each_circle)

Predicted label for the image 'C:\Users\Laptop\test\
Extracted_Circles_from_OMR\filled_circles\circle2_111.png': Confusion

c:\Users\Laptop\anaconda3\lib\site-packages\sklearn\base.py:450:
UserWarning: X does not have valid feature names, but StandardScaler
was fitted with feature names
  warnings.warn(
```

6.8 PREDICTING THE LABELS FOR THE CIRCLES

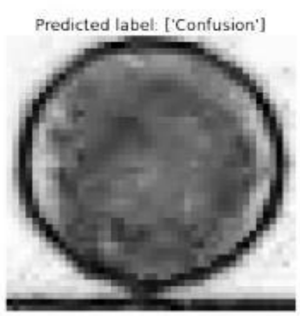



```
Predicted label for the image 'C:\Users\Laptop\test\
Extracted_Circles_from_OMR\filled_circles\circle2_149.png': Stress

c:\Users\Laptop\anaconda3\lib\site-packages\sklearn\base.py:450:
UserWarning: X does not have valid feature names, but StandardScaler
was fitted with feature names
  warnings.warn(
```






```
:\Users\Laptop\anaconda3\lib\site-packages\sklearn\base.py:450:
serWarning: X does not have valid feature names, but StandardScaler
as fitted with feature names
 warnings.warn(

redicted label for the image 'C:\Users\Laptop\test\
xtracted_Circles_from_OMR\filled_circles\circle2_215.png': Confusion
```



CHAPTER # 07

7.1 TESTING OF THE PROPOSED FRAMEWORK

7.1.1 DATA ACQUISITION AND PREPARATION TESTING

To verify the integrity of the dataset, multiple test cases were applied:

a) **Format Verification**: Ensured that all OMR sheets were in the required JPG/PNG format.
b) **Data Consistency Check**: Validated that each page contained bubbles and no missing elements.
c) **Image Resolution Assessment**: Assessed the quality of scans to confirm uniformity.

7.1.2. CIRCLE EXTRACTION TESTING

a) **Accuracy of Hough Circle Transform**: The detected circles were manually compared with the original sheets to measure precision.
b) **Parameter Optimization**: Experimented with different edge thresholds and radii constraints to optimize circle detection.
c) **Noise Reduction**: Evaluated the effectiveness of Gaussian blurring in eliminating unnecessary edges.

7.1.3. CIRCLE CLASSIFICATION TESTING

a) **Intensity Threshold Validation**: Checked if the predefined pixel intensity thresholds effectively classified empty, filled, and letter bubbles.
b) **Misclassification Analysis**: Identified and corrected instances where filled bubbles were incorrectly classified as empty or vice versa.
c) **Segregation Efficiency**: Verified that classified bubbles were correctly stored in their respective directories.

7.2 FEATURE EXTRACTION AND LABELING TESTING

a) **Feature Computation Consistency**: Ensured that extracted features (Area, Solidity) were computed accurately across different samples.

b) **Contour Detection Validation**: Cross-checked detected contours to confirm accurate region selection.
c) **Labeling Accuracy**: Random samples were manually labeled and compared with the automated labeling output.

7.3 EXPLORATORY DATA ANALYSIS AND STATISTICAL VALIDATION TESTING

a) **Descriptive Statistics Verification**: Checked the correctness of summary statistics (mean, median, standard deviation).
b) **Visualization Inspection**: Reviewed histograms, scatter plots, and correlation matrices to ensure meaningful insights.
c) **Statistical Test Validation**: Conducted normality tests and ANOVA to confirm feature differentiation among sentiment classes.

7.4. FEATURE SCALING AND CLASS IMBALANCE HANDLING

7.4.1 TESTING

a) **Scaling Accuracy**: Verified that all features were standardized correctly using the StandardScaler.
b) **Class Balance Evaluation**: Generated class distribution plots to assess the presence of imbalance.
c) **Imbalance Mitigation Strategies**: Simulated oversampling and weighting techniques to analyze performance improvements.

7.4.2 MODEL TRAINING AND EVALUATION TESTING

a) **Data Splitting Check**: Confirmed the 80:20 split between training and testing data.
b) **Model Convergence Analysis**: Ensured that Logistic Regression reached convergence within the allowed iterations.
c) **Performance Metrics Evaluation**:

   1. Accuracy and classification reports were reviewed.
   2. Confusion matrices were examined to detect misclassification patterns.
   3. Precision-recall curves were analyzed for threshold optimization.

7.4.3 PREDICTION AND PIPELINE INTEGRATION TESTING

a) **Single Image Prediction Verification**: Tested the system by running sample OMR sheets and verifying sentiment predictions.
b) **End-to-End Pipeline Validation**: Assessed whether each module (extraction, classification, feature engineering, and inference) functioned seamlessly.

Table # 3: Evaluate the testing of project

Testing Category	Test Case	Description	Result
1. Data Acquisition and Preparation Testing	Format Verification	Ensured that all OMR sheets were in the required JPG/PNG format.	☑ Passed
	Data Consistency Check	Validated that each page contained bubbles and no missing elements.	☑ Passed
	Image Resolution Assessment	Assessed the quality of scans to confirm uniformity.	☑ Passed
2. Circle Extraction Testing	Accuracy of Hough Circle Transform	The detected circles were manually compared with the original sheets to measure precision.	☑ Passed
	Parameter Optimization	Experimented with different edge thresholds and radii constraints to optimize circle detection.	☑ Optimized
	Noise Reduction	Evaluated the effectiveness of Gaussian blurring in eliminating unnecessary edges.	☑ Effective
3. Circle Classification Testing	Intensity Threshold Validation	Checked if the predefined pixel intensity thresholds effectively classified empty, filled, and letter bubbles.	☑ Passed
	Misclassification Analysis	Identified and corrected instances where filled bubbles were incorrectly classified.	☑ Adjusted
	Segregation Efficiency	Verified that classified bubbles were correctly stored in their respective directories.	☑ Confirmed
4. Feature Extraction and Labeling Testing	Feature Computation Consistency	Ensured that extracted features (Area, Solidity) were computed accurately across different samples.	☑ Accurate
	Contour Detection	Cross-checked detected contours to	☑ Verified

	Validation	confirm accurate region selection.	
	Labeling Accuracy	Random samples were manually labeled and compared with the automated labeling output.	☑ Matched
5. Exploratory Data Analysis & Statistical Validation	Descriptive Statistics Verification	Checked the correctness of summary statistics (mean, median, standard deviation).	☑ Verified
	Visualization Inspection	Reviewed histograms, scatter plots, and correlation matrices to ensure meaningful insights.	☑ Approved
	Statistical Test Validation	Conducted normality tests and ANOVA to confirm feature differentiation among sentiment classes.	☑ Confirmed
6. Feature Scaling & Class Imbalance Handling	Scaling Accuracy	Verified that all features were standardized correctly using the Standard Scaler.	☑ Standardized
	Class Balance Evaluation	Generated class distribution plots to assess the presence of imbalance.	☑ Evaluated
	Imbalance Mitigation Strategies	Simulated oversampling and weighting techniques to analyze performance improvements.	☑ Considered
7. Model Training & Evaluation Testing	Data Splitting Check	Confirmed the 80:20 split between training and testing data.	☑ Confirmed
	Model Convergence Analysis	Ensured that Logistic Regression reached convergence within the allowed iterations.	☑ Converged
	Performance Metrics Evaluation	Accuracy, classification reports, confusion matrices, and precision-recall curves were analyzed.	☑ Evaluated
8. Prediction & Pipeline Integration Testing	Single Image Prediction Verification	Tested the system by running sample OMR sheets and verifying sentiment predictions.	☑ Verified
	End-to-End Pipeline Validation	Assessed whether each module (extraction, classification, feature engineering, and inference) functioned seamlessly.	☑ Integrated

CHAPTER # 08

8.1 CONCLUSION

In conclusion, integrating sentiment analysis with OMR revolutionizes data interpretation by addressing the limitations of traditional OMR systems. By detecting emotional markers such as stress, confidence, and confusion, this approach enhances the accuracy of assessments in education, business, and research. The automated framework reduces manual effort, improves evaluation efficiency, and supports large-scale sentiment analysis with machine learning and predictive analytics. Despite requiring technical infrastructure and expertise, its benefits in personalized learning, market insights, and decision-making far outweigh the challenges. As a scalable and versatile tool, this system paves the way for more insightful, data-driven evaluations across multiple domains.

8.2 FUTURE TESTING PLAN

To further refine the framework, additional tests will be conducted:

a) **Enhanced Model Evaluation:** Implement deep learning approaches (e.g., CNNs) to compare performance with logistic regression.
b) **Real-World Deployment:** Test the framework on large-scale datasets from diverse sources to ensure generalizability.
c) **Adaptive Feature Selection:** Experiment with advanced feature selection techniques to improve classification accuracy.
d) **Automated Misclassification Correction:** Develop an intelligent feedback mechanism to detect and correct misclassified bubbles.
e) **User Interface Testing:** Integrate and test a web-based interface for real-time processing and visualization of OMR sheets.

8.3 REFERENCE

1. Sujon, M. A. K. (2016). Design & Developed of an Optical Mark Reader (Doctoral dissertation, East West University).
2. Enes Alperen Buğaz, Orhan Akbulut, Aysun Taşyapi Çelebi, Uğur Yildiz, "Deep Learning based Order Form Recognition", 2024 9th International Conference on Computer Science and Engineering (UBMK), pp.1-5, 2024
3. Hussain, T. (2020). MULTIMEDIA COMPUTING. Booksclinic Publishing.
4. de Elias, E. M., Tasinaffo, P. M., & Hirata Jr, R. (2021). Optical mark recognition: Advances, difficulties, and limitations. SN Computer Science, 2(5), 367.
5. AL-Marakeby, Fast Camera based optical mark reader system,Journal of Al Azhar University Engineering Sector (JAUES) 2013 (in press)
6. David Doermann, Jian Liang, and Huiping Li "Progress in Camera-Based Document Image Analysis" International Conference on Document Analysis and Recognition (ICDAR'03)
7. Li Zhang, Andy M. Yip, Chew Lim Tan, "Removing Shading Distortions in Camera-based Document Images Using Inpainting and Surface Fitting With Radial Basis Functions" International Conference on Computer Vision, ICCV 2007.
8. K. Chua, L. Zhang, Y. Zhang, and C. Tan. "A fast and stable approach for restoration of warped document images. " 8th Int'l Conf. on Document Analysis and Recognition, 2005
9. Bergeron, B.P. (1998). Optical mark recognition. Tallying information from filled-in 'bubbles'. Postgraduate medicine, 104 2, 23-5 .
10. Liu, Xing & Wei, Fupeng & Jiang, Wei & Zheng, Qiusheng & Qiao, Yaqiong & Liu, Jizong & Niu, Liyue & Chen, Ziwei & Dong, Hangcheng. (2023). MTR-SAM: Visual Multimodal Text Recognition and Sentiment Analysis in Public Opinion Analysis on the Internet. Applied Sciences. 13. 7307. 10.3390/app13127307.
11. Tümer, Abdullah & Küçükkara, Zeki. (2018). An Image Processing Oriented Optical Mark Recognition and Evaluation System. International Journal of Applied Mathematics, Electronics and Computers. 6. 59-64. 10.18100/ijamec.2018447788.
12. Smith, A. M. (1981). Optical mark reading - making it easy for users. Proceedings of the 9th Annual ACM SIGUCCS Conference on User Services - SIGUCCS '81, 257-263. https://doi.org/10.1145/800079.802600
13. IBM Archives: IBM 805 Test Scoring Machine. https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_9.html. Last accessed January 18, 2023.
14. A Bubble for Your Thoughts? Not for Long. (2013, August 4). Timeline – NYTimes.com. https://archive.nytimes.com/www.nytimes.com/interactive/201 3/08/04/education/edlife/20130804_edlife_testing.html. Last accessed January 18, 2023.
15. Veronese, K. (2015, December 16). The birth of Scantrons, the bane of standardized testing. Gizmodo. https://gizmodo.com/the-birth-of-scantrons-thebane-of-standardized-testin-5908833. Last accessed January 18, 2023.

16. Patel, R., Sanghavi, S., Gupta, D., & Raval, M. S. (2015). CheckIt - a low cost mobile OMR System. TENCON 2015 - 2015 IEEE Region 10 Conference, 1-5. https://doi.org/10.1109/tencon.2015.7372983

17. Xu, Y.; Wang, Y.; Zhou, W.; Wang, Y.; Yang, Z.; Bai, X. Textfield: Learning a deep direction field for irregular scene text detection.

18. IEEE Trans. Image Process. 2019, 28, 5566–5579. [CrossRef]

19. Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; Shao, S. Shape robust text detection with progressive scale expansion network. IN Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019;pp. 9336–9345.

20. Huang, R.; Fan, M.; Xing, Y.; Zou, Y. Image blur classification and unintentional blur removal. IEEE Access 2019, 7, 106327–106335.[CrossRef]

21. Sumady, O.O.; Antoni, B.J.; Nasuta, R.; Irwansyah, E. A Review of Optical Text Recognition from Distorted Scene Image.In Proceedings of the 2022 4th International Conference on Cybernetics and Intelligent System (ICORIS), Prapat, Indonesia, 8–9 October 2022; pp. 1–5.

22. Zhu, Y.; Chen, J.; Liang, L.; Kuang, Z.; Jin, L.; Zhang, W. Fourier contour embedding for arbitrary-shaped text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 3123–3131.

23. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 2015, [CrossRef] [PubMed]

24. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

25. Cao, L.; Li, H.; Xie, R.; Zhu, J. A text detection algorithm for image of student exercises based on CTPN and enhanced YOLOv3. IEEE Access 2020, 8, 176924–176934. [CrossRef]

26. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. Textboxes: A fast text detector with a single deep neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

27. Dai, P.; Zhang, S.; Zhang, H.; Cao, X. Progressive contour regression for arbitrary-shape scene text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7393–7402.

28. Li, Y.; Wu, Z.; Zhao, S.; Wu, X.; Kuang, Y.; Yan, Y.; Ge, S.; Wang, K.; Fan, W.; Chen, X. PSENet: Psoriasis severity evaluation network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 800–807.

29. Liao, M.; Wan, Z.; Yao, C.; Chen, K.; Bai, X. Real-time scene text detection with differentiable binarization. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11474–11481.

30. Liu, Y.; Si, C.; Jin, K.; Shen, T.; Hu, M. FCENet: An instance segmentation model for extracting figures and captions from material documents. IEEE Access 2020, 9, 551–564. [CrossRef]

31. Liu, J.; Liu, X.; Sheng, J.; Liang, D.; Li, X.; Liu, Q. Pyramid mask text detector. arXiv 2019, arXiv:1903.11800.

32. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2016, 39, 2298–2304. [CrossRef]

33. Yin, F.; Wu, Y.-C.; Zhang, X.-Y.; Liu, C.-L. Scene text recognition with sliding convolutional character models. arXiv 2017, arXiv:1709.01727.

34. Yousef, T.; Janicke, S. A survey of text alignment visualization. IEEE Trans. Vis. Comput. Graph. 2020, 27, 1149–1159. [CrossRef] [PubMed]

35. He, P.; Huang, W.; Qiao, Y.; Loy, C.; Tang, X. Reading scene text in deep convolutional sequences. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.

36. Luo, C.; Jin, L.; Sun, Z. Moran: A multi-object rectified attention network for scene text recognition. Pattern Recognit. 2019, 90, 109–118. [CrossRef]

37. Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Aster: An attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 41, 2035–2048. [CrossRef] [PubMed]

38. Karunanayake, N. (2015). OMR Sheet Evaluation by Web Camera Using Template Matching Approach. International Journal for Research in Emerging Science and Technology, 2(8), 40-44.

39. Rasiq, G. R. I., Al Sefat, A., & Hasnain, M. F. (2019). Mobile based MCQ Answer Sheet Analysis and Evaluation Application. 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), 144-147.https://doi.org/10.1109/smart46866.2019.9117468

40. Afifi, M., & Hussain, K. F. (2019). The achievement of higher flexibility in multiple-choice-based tests using image classification techniques. International Journal on Document Analysis and Recognition (IJDAR), 22(2), 127–142. https://doi.org/10.1007/s10032-019-00322-3

41. Jingyi, T., Hooi, Y. K., & Bin, O. K. (2021). Image processing for enhanced OMR answer matching precision. 2021 International Conference on Computer & Information Sciences (ICCOINS), 322-327.https://doi.org/10.1109/iccoins49721.2021.9497172