VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum-590018.

DEEP LEARNING MINI PROJECT REPORT

ON

Automated OMR Scoring and Evaluation

Submitted in Partial fulfilment for the V Semester, BE, Artificial Intelligence & Data Science

Submitted by:

Elvis Bibu	**1BM23AD021**
Kirthan A	**1BM23AD030**
Sai Prashanth	**1BM23AD051**

Under the Guidance of:

Mrs. Sangeetha S
Assistant Professor
Dept. of AI & DS

2025-2026

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

BMS College of Engineering

Bull Temple Road, Basavanagudi, Bengaluru-560019

(Autonomous Institute under Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi, NAAC Accredited with 'A++' Grade)

BMS College of Engineering

Bull Temple Road, Basavanagudi, Bengaluru-560019

(Autonomous Institute under Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi,

NAAC Accredited with 'A++' Grade)

Department of Artificial Intelligence & Data Science

CERTIFICATE

Certified that the mini project work entitled "**Automated OMR Scoring and Evaluation**" carried out by **Elvis Bibu, Kirthan A and Sai Prashanth**, Bonafide students of BMS College of Engineering, in partial fulfillment for the award of Bachelor of Engineering in Artificial Intelligence & Data Science of the Visvesvaraya Technological University, Belgaum during the year 2025-26. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library.

Signature of the Guide
Sangeetha S
Asst. Prof. Dept. of AI & DS
BMSCE, Bengaluru.

Signature of the HOD
Dr. Indiramma M
Prof. & Head, Dept. of AI& DS
BMSCE, Bengaluru.

DECLARATION

We the undersigned students of 5th semester, Department of Artificial Intelligence & Data Science, BMS College of Engineering, declare that the mini project entitled "**Automated OMR Scoring and Evaluation"**, is a Bonafide work of us and our project is neither a copy nor by any means a modification of any other engineering project.

We also declare that this mini project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Name	USN	Signature
Elvis Bibu	**1BM23AD021**	**……**
Kirthan A	**1BM23AD030**	**……**
Sai Prashanth	**1BM23AD051**	**……**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We are grateful to our institution, **BMS College of Engineering**, with its ideals and inspirations for having provided us with the facilities, which has made this project a success.

We earnestly thank **Dr. Bheemsha, Principal, BMS College of Engineering,** for facilitating academic excellence in the college and providing us with the congenial environment to work in, that helped us in completing this project.

We wish to extend our profound thanks to **Dr. Indiramma M, Prof. and Head, Department of Artificial Intelligence & Data Science, BMSCE,** for giving us the consent to carry out this project.

We would like to express our sincere thanks to our guide **Prof. Sangeetha S, Assistant Professor, Department of Artificial Intelligence & Data Science, BMSCE,** for her able guidance and valuable advice at every stage, which helped us in the successful completion of the project.

We would like to thank all the teaching and non-teaching staff for their valuable advice and support.

We would like to express our sincere thanks to our parents and friends for their support.

Elvis Bibu

Kirthan A

Sai Prashanth

ABSTRACT

As the use of multiple-choice examinations increases in educational institutions, the manual evaluation of Optical Mark Recognition (OMR) sheets becomes time-consuming, error-prone, and inefficient. OMR evaluation is the process of detecting marked responses on a printed answer sheet and computing the final score based on an answer key. Traditional OMR systems rely heavily on fixed templates and rule-based processing, making them highly sensitive to improper shading, misalignment, and noise in scanned images, leading to inaccurate results.

The project proposes an automated OMR evaluation and scoring system using deep learning techniques to overcome the limitations of conventional methods. The system is capable of detecting answer bubbles, identifying filled responses using a Convolutional Neural Network (CNN), and comparing the detected answers with a provided answer key for automatic scoring. The proposed system also performs image preprocessing, bubble detection, classification, and result visualization.

The project further introduces a user-friendly web-based interface where users can upload the OMR answer sheet image and the corresponding answer key in CSV format. The system calculates individual question responses, determines correctness, and generates the final score instantly. This project offers a reliable, fast, and scalable solution for automated examination evaluation, reducing human effort and improving accuracy.

CONTENTS

LIST OF FIGURES

Figures **Page No.**

LIST OF TABLES

Tables **Page No.**

CHAPTER 1

INTRODUCTION

1.1 General Introduction

In the modern education system, assessments and examinations play a critical role in evaluating the performance and understanding of students. With the increase in the number of students appearing for competitive examinations, entrance tests, school and university-level assessments, the process of evaluating answer sheets has become a major challenge. Traditional evaluation methods are slow, require extensive human involvement, and are prone to errors due to fatigue and subjectivity. This increasing demand for fast, accurate, and reliable evaluation methods has led to the adoption of automated technologies.

Optical Mark Recognition (OMR) is one such technology that is widely used for the evaluation of objective-type examinations. OMR is a process in which data is captured from specially designed forms by detecting the presence or absence of marks in predefined positions. These forms typically contain multiple-choice questions where candidates mark their answers by shading bubbles. OMR technology is widely used in examinations, surveys, voting systems, feedback collection, and psychological tests.

However, conventional OMR systems come with several limitations. They require specialized OMR scanners, strict sheet formats, proper alignment, uniform printing quality, and precise shading of bubbles. Slight mistakes such as light shading, over-marking, stray marks, folded sheets, poor scanning quality, or variations in lighting can lead to incorrect detection of answers. Additionally, traditional OMR scanners are expensive, bulky, and are not easily accessible to smaller institutions or remote users.

With recent advances in Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL), new approaches have emerged to overcome the shortcomings of conventional OMR systems. Deep learning models, especially Convolutional Neural Networks (CNNs), have shown remarkable performance in image classification and object detection tasks. These models can learn complex features directly from raw images, making them ideal for applications such as handwritten digit recognition, face detection, medical imaging, and document analysis. When applied to OMR evaluation, deep learning enables more robust and flexible detection of marked and unmarked bubbles even under real-world distortions.

1.2 Optical Mark Recognition (OMR) Technology

Optical Mark Recognition is a technique used to read human-marked data from document forms. It works by detecting the presence of darkened areas on paper in predefined bubble positions. The most common applications of OMR include:

- Competitive examinations
- School and university tests
- Government recruitment exams
- Survey forms
- Voting systems
- Psychological assessments

In a typical OMR process, the answer sheets are scanned using a high-speed scanner. The scanner converts the physical sheet into a digital image. This image is then processed by OMR software, which detects the positions of predefined bubbles and determines whether they are filled or empty. The detected responses are then compared with a stored answer key to generate the final score.

Traditional OMR systems rely heavily on rule-based thresholding techniques, where pixel intensity values are used to determine whether a bubble is filled. These methods work well only when the sheet quality, scanning conditions, and marking style are consistent. However, in real-world scenarios, students may shade bubbles lightly, use different pens or pencils, erase answers, or make partial marks, which often leads to incorrect recognition.

1.3 Limitations of Conventional OMR Systems

Although OMR technology has been in use for several decades, traditional OMR systems suffer from several drawbacks:

1. **Rigid Template Dependency**
   Traditional OMR software requires strict templates. Even a small misalignment between the scanned image and the template can lead to incorrect detection.
2. **Shading Sensitivity**
   These systems fail when bubbles are lightly filled, overfilled, or partially erased.

3. **High Cost**
   Specialized OMR scanners and licensed evaluation software are expensive and unaffordable for many schools and small institutions.

4. **Poor Performance Under Noise**
   Dust, stains, shadows, folds, and camera distortions affect detection performance badly.

5. **Limited Flexibility**
   Traditional OMR systems do not work well with mobile camera images.

6. **Manual Supervision Required**
   Many systems still require human verification and manual corrections.

These limitations create the need for a more intelligent, adaptive, and cost-effective OMR evaluation solution.

1.4 Role of Deep Learning in Image-Based OMR Evaluation

Deep learning is a subfield of machine learning that uses multi-layered neural networks to automatically learn features from data. Unlike traditional image processing approaches that rely on manually designed features, deep learning models can extract relevant features on their own through training.

In image-based OMR systems, deep learning plays a crucial role in:

- Detecting bubble regions
- Identifying filled and empty bubbles
- Handling image noise and distortions
- Improving classification accuracy
- Eliminating template dependency

Convolutional Neural Networks (CNNs) are particularly well-suited for this task due to their ability to detect spatial patterns and textures. CNNs can distinguish between:

- Empty bubbles

- Lightly shaded bubbles
- Properly filled bubbles
- Overfilled bubbles

This makes deep learning-based OMR evaluation far superior to traditional rule-based systems.

1.5 Need for the Proposed System

In today's rapidly evolving educational and evaluation environment, there is a strong demand for faster, more accurate, and cost-effective assessment systems. Traditional manual and scanner-based OMR evaluation methods are no longer sufficient to handle large volumes of answer sheets efficiently, especially under real-world constraints such as image noise, improper shading, and mobile-captured distortions. Hence, there is a critical need for an intelligent, flexible, and automated OMR evaluation system that can operate without expensive hardware, adapt to varying image conditions, and deliver reliable results in real time. The proposed deep learning–based OMR evaluation system addresses these challenges by combining image processing and CNN-based classification to provide a highly accurate, scalable, and user-friendly solution suitable for modern educational institutions.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 Introduction

Requirement Analysis is one of the most important stages in the development of any software system. It involves studying the problem in detail, understanding user expectations, identifying system limitations, and determining what the proposed system should achieve. A clear and accurate requirement analysis helps avoid errors during later stages of development and ensures that the final system meets real-world needs.

For the Automated OMR Evaluation and Scoring System, requirement analysis focuses on replacing manual and scanner-based evaluation methods with an intelligent, automated, and deep learning–based solution. This phase studies the drawbacks of existing OMR systems and defines the functional, non-functional, and system-level requirements needed to develop a reliable and efficient evaluation platform.

2.2 Overall Description of the System

The Automated OMR Evaluation and Scoring System is designed to automatically evaluate multiple-choice answer sheets using image processing and deep learning techniques. The system accepts scanned or camera-captured images of OMR sheets as input along with a CSV-based answer key. It performs image preprocessing, detects the answer bubbles, classifies them as filled or unfilled using a trained Convolutional Neural Network (CNN), and compares the detected answers with the correct answers from the answer key to generate the final score.

The system does not require specialized OMR scanning hardware and can operate using standard images captured from mobile phones or scanners. The entire process is carried out through a web-based interface, making the system accessible from any location. This makes the solution flexible, portable, and suitable for modern educational and assessment environments.

2.3 Operating Environment

The operating environment defines the platform on which the system is expected to function smoothly. The proposed system is designed to run on commonly used operating systems such

as Windows and Linux. Python is used as the primary programming language because of its simplicity, rich library support, and strong community backing in the fields of artificial intelligence and image processing.

The system uses Streamlit for creating the web interface, OpenCV for image preprocessing and bubble detection, and TensorFlow along with Keras for building and running the deep learning model. Additional libraries such as NumPy and Pandas are used for numerical computation and handling answer key data. The system is expected to run efficiently on systems with moderate hardware configurations.

2.4 Functional Requirements

Functional requirements define the specific operations that the system must perform to fulfil its purpose. The Automated OMR Evaluation and Scoring System must be capable of accepting OMR sheet images from users and also allow the upload of answer keys in CSV format. The system must preprocess the uploaded image to remove noise and enhance clarity. It must then detect all the answer bubbles present on the sheet and classify each bubble as either filled or empty using a deep learning model.

After detecting and classifying the responses, the system must compare the detected answers with the correct answers provided in the answer key. Based on this comparison, it must calculate the final score and display individual question-wise results. The system must also visually highlight the correct answers on the OMR sheet so that users can easily verify the evaluation. These functional requirements ensure that the system completely automates the evaluation process from input to final result generation.

2.5 Non-Functional Requirements

Non-functional requirements describe the quality and performance characteristics of the system rather than specific operations. Accuracy is the most critical requirement for the proposed system, as even a small error in bubble detection or classification can lead to incorrect evaluation. The system must be fast enough to process OMR sheets within a few seconds so that it can be used for large-scale examinations.

Reliability is another important factor, as the system must operate consistently under different image conditions such as variations in lighting, shadows, and scanning quality. The system

must also be user-friendly so that teachers and administrators with minimal technical knowledge can operate it easily. In addition, the system must be scalable to support a large number of users and secure enough to ensure safe handling of uploaded images and result data.

2.6 User Requirement Analysis

The primary users of the Automated OMR Evaluation and Scoring System include teachers, examination administrators, schools, colleges, and online assessment platforms. These users expect a solution that significantly reduces manual workload and evaluation time. They require a system that allows easy uploading of OMR sheet images and answer key files without the need for technical expertise.

Users expect quick and accurate result generation with clear display of selected answers, correct answers, confidence levels, and final scores. The system should also provide visual verification of results to build trust and transparency in automated evaluation. Overall, the system must be intuitive, efficient, and reliable to ensure widespread adoption.

2.7 System Constraints

The performance of the system depends on several constraints identified during requirement analysis. The quality of the uploaded OMR sheet image plays a major role in determining detection accuracy. Images that are blurred, heavily shadowed, or extremely distorted may reduce the effectiveness of the detection process.

The system also assumes that the bubbles on the OMR sheet are properly aligned and arranged in a structured manner. Extreme variations in sheet design may require additional customization. The system requires a basic level of computational resources to execute deep learning inference efficiently. Internet connectivity is required if the system is deployed as a web application.

2.8 Feasibility Study

The proposed system is technically feasible because all required development tools, libraries, and frameworks are freely available and widely used in industry and academia. Deep learning

frameworks such as TensorFlow and OpenCV are mature and reliable for real-world image processing applications.

From an economic perspective, the system is highly feasible as it eliminates the need for expensive OMR scanners and licensed evaluation software. This makes it affordable even for small schools and institutions. Operationally, the system is easy to use and does not require extensive training, making it suitable for real-world deployment without major changes to existing workflows.

2.9 Data Requirement Analysis

The Automated OMR Evaluation and Scoring System requires different types of data for proper functioning. The primary input data consists of OMR sheet images captured through scanners or mobile cameras. The system also requires a structured answer key provided in CSV format for comparison and scoring.

In addition to input data, the system requires training data in the form of synthetic and real bubble images to train the CNN model. All collected data must be stored and processed securely to ensure data integrity and result accuracy. Proper formatting and validation of the input data are necessary to avoid processing errors during evaluation.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction to Software Requirement and Analysis

Software Requirement and Analysis is one of the most critical phases of any system development life cycle. It involves understanding the problem deeply, identifying what the system should do, and defining both the functional and non-functional requirements. A well-defined requirement analysis ensures that the final system meets user expectations and performs efficiently in real-world conditions.

For the Automated OMR Evaluation and Scoring System using Deep Learning, the requirement analysis focuses on building a system that is:

- Accurate in detecting filled answers
- Fast in processing large datasets
- User-friendly for non-technical users
- Flexible enough to work with different image qualities

3.2 Software Environment

The proposed system is developed using modern open-source tools and frameworks to ensure reliability, scalability, and ease of deployment.

3.2.1 Operating System

- Windows 10 / 11
- Linux (Ubuntu or equivalent)

3.2.2 Programming Language

- **Python** is used due to its:
    - Simplicity
    - Large number of AI and image processing libraries

- Strong community support
- Easy integration with web frameworks

3.2.3 Frameworks and Libraries Used

- **Streamlit** – Web application interface
- **OpenCV** – Image preprocessing and bubble detection
- **TensorFlow & Keras** – Deep learning model development
- **NumPy** – Numerical operations
- **Pandas** – Handling answer key and result data

Software Components	Purpose
Python	Core Programming
OpenCV	Image Processing
TensorFlow	CNN Model
Pandas	Dataset Handling
Streamlit	Web Interface

Table 3.2 Software applications

3.3 Data Requirements

The system requires various forms of data for operational and training purposes. The primary input data consists of OMR answer sheet images captured using scanners or mobile cameras. The second major input is the answer key, which is provided in a structured CSV file format.

The system also requires training data in the form of synthetic and real bubble images for training the Convolutional Neural Network. The training data must include both filled and unfilled bubble samples under different lighting and noise conditions. All data must be validated before processing to ensure reliable output.

3.4 Interface Requirements

The user interface of the system is developed using Streamlit and is designed to be simple, interactive, and user-friendly. The interface provides options to upload OMR sheet images and answer key CSV files. It also displays the pre-processed grayscale image, bubble detection visualization, and evaluation results.

The interface presents the results in a tabular format showing question number, selected answer, correct answer, confidence level, and result status. It also displays the final score prominently. Visual feedback is provided by marking the correct answers directly on the OMR sheet image, improving trust and verification.

3.5 Performance Requirements

The system must process each OMR sheet within a short time interval to ensure suitability for large-scale evaluations. The bubble detection accuracy must be greater than 95 percent under standard image conditions. The result generation must be consistent and error-free.

The system must be capable of handling multiple evaluations without performance degradation. The computational efficiency of the deep learning model must be optimized to ensure real-time processing.

3.6 Security Requirements

The system must ensure secure handling of uploaded images and answer key files. Unauthorized users must not be allowed to access stored results or uploaded data. Temporary files generated during processing must be safely removed after evaluation.

The system must ensure that data is not modified during transmission and that results are displayed accurately without tampering.

3.7 Operating Environment

The system requires the following operating environment for smooth functioning:

- **Operating System**: Windows 10/11, Linux

- **Programming Language**: Python
- **Development Tools**: VS Code, Jupyter Notebook
- **Web Framework**: Streamlit
- **Libraries Used**:
    - OpenCV – Image processing
    - TensorFlow & Keras – Deep learning model
    - NumPy – Mathematical operations
    - Pandas – Data handling

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Introduction

This chapter explains the overall analysis and design of the Automated OMR Evaluation and Scoring System. The analysis phase focuses on understanding how the system behaves, how data flows between different components, and how user interactions are handled. The design phase defines the internal structure of the system, including modules, system architecture, data flow, and interface layout.

The proposed system is designed with a modular architecture so that each component such as image preprocessing, bubble detection, deep learning classification, and result visualization works independently yet in coordination with other modules. This approach improves system reliability, scalability, and maintainability.

4.2 System Architecture Overview

The system architecture defines the high-level structure of the Automated OMR Evaluation and Scoring System. It illustrates how input data flows through different processing stages and how the final output is generated. The system follows a layered architecture consisting of the user interface layer, processing layer, deep learning layer, and result visualization layer.

In the first layer, the user interacts with the web-based dashboard and uploads the OMR sheet image and answer key. In the second layer, image preprocessing and bubble detection are performed using OpenCV. In the third layer, the Convolutional Neural Network classifies the detected bubbles as filled or unfilled. In the final layer, the detected responses are compared with the answer key and the final score is generated and displayed.

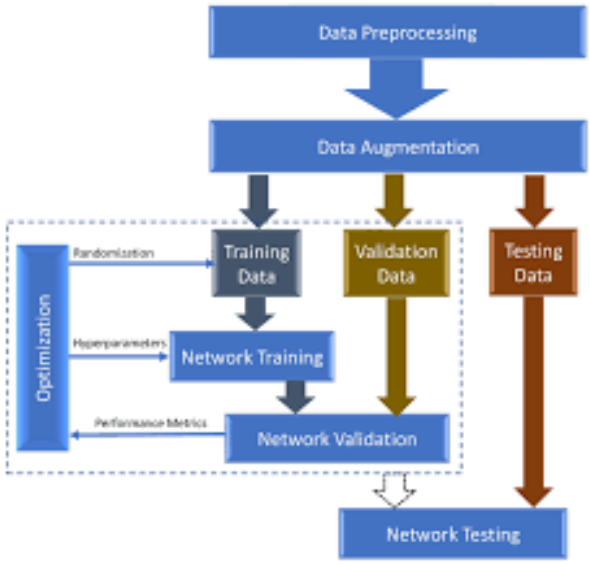

Fig 4.2 – Basic Deep Learning Architecture Design

4.3 Module Description

The system is divided into the following major modules:

1. User Interface Module
2. Image Preprocessing Module
3. Bubble Detection Module
4. Bubble Classification Module
5. Answer Matching and Scoring Module
6. Result Visualization Module

Each module is designed to perform a specific task and communicate with other modules through well-defined data flow. This modular design improves accuracy, maintainability, and makes future enhancements easier to implement.

4.4 User Interface Module

The user interface module is responsible for all interactions between the user and the system. It is developed using Streamlit and provides a simple web-based dashboard. Through this interface, the user can upload the OMR answer sheet image and the answer key CSV file.

The dashboard also displays intermediate outputs such as the grayscale image, binary image, detected bubbles, and final evaluation results. The interface ensures that users with minimal technical knowledge can operate the system efficiently.

4.5 Image Preprocessing Module

The image preprocessing module prepares the input OMR image for accurate bubble detection. The uploaded image is first resized for uniformity. It is then converted into grayscale to reduce computational complexity. Noise is removed using bilateral filtering, which smoothens the image while preserving edges.

Further, operations such as thresholding and edge detection are applied to enhance bubble boundaries. These preprocessing steps are crucial because the quality of bubble detection depends heavily on the clarity of the input image.

4.6 Bubble Detection Module

The bubble detection module identifies circular regions that correspond to answer bubbles on the OMR sheet. The system uses the Hough Circle Transform technique to detect circles of varying sizes. Multiple parameter combinations are used to ensure that bubbles of different radii are accurately detected.

Duplicate detections are removed by checking the distance between detected circles. The detected bubbles are then sorted based on their spatial positions and grouped row-wise to represent different questions.

CHAPTER 5

IMPLEMENTATION

5.1 Software and Environment Setup

The implementation of the proposed system is carried out using Python and several open-source libraries. The following packages are required to run the system:

pip install streamlit opencv-python numpy pandas tensorflow

Python acts as the core programming language, OpenCV is used for image processing, TensorFlow and Keras are used for deep learning, NumPy supports numerical operations, Pandas handles CSV files, and Streamlit is used to deploy the web-based interface.

5.2 Deep Learning Model Architecture Implementation

A Convolutional Neural Network (CNN) is implemented to classify bubbles as filled or unfilled. The model consists of convolution layers, pooling layers, and fully connected layers.

```
def build_bubble_classifier():
    model = models.Sequential([
        layers.Input(shape=(40, 40, 1)),
        layers.Conv2D(32, 3, activation="relu", padding="same"),
        layers.MaxPooling2D(2),
        layers.Conv2D(64, 3, activation="relu", padding="same"),
        layers.MaxPooling2D(2),
        layers.Flatten(),
        layers.Dropout(0.3),
        layers.Dense(128, activation="relu"),
        layers.Dense(2, activation="softmax")
    ])
    model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])
```

Fig 5.2 – Deep learning model

5.3 Synthetic Data Generation for Training

To train the CNN model without requiring a large manually labeled dataset, synthetic bubble images are generated.

```
def generate_synthetic_data(n_samples=2000, size=40):
    """Generate realistic OMR bubble training data"""
    X, y = [], []

    for _ in range(n_samples // 2):
        # Empty bubble
        empty = np.ones((size, size), dtype=np.uint8) * 250
        center = size // 2
        radius = size // 3
        cv2.circle(empty, (center, center), radius, 50, 2)
        noise = np.random.normal(0, 8, empty.shape).astype(np.int16)
        empty = np.clip(empty.astype(np.int16) + noise, 0, 255).astype(np.uint8)
        X.append(empty)
        y.append(0)
```

Fig 5.3- Synthetic Data using open CV

5.4 Bubble Detection Implementation

Hough Circle Transform is used to detect circular bubbles on the OMR sheet.

```
    circles1 = cv2.HoughCircles(
        blurred,
        cv2.HOUGH_GRADIENT,
        dp=1,
        minDist=20,
        param1=50,
        param2=20,
        minRadius=10,
        maxRadius=25
    )
    if circles1 is not None:
        all_circles.extend(circles1[0])
```

Fig 5.4 – Small Bubble Implementation

5.5 Bubble Classification Implementation

Each detected bubble is resized and passed through the CNN model for classification.

pred = model.predict(crop_input, verbose=0)[0]

cnn_score = pred[1]

A combined confidence score is calculated using CNN output and pixel darkness values.

CHAPTER 6

TESTING AND RESULTS

6.1 Bubble Detection Results

The bubble detection module was tested on multiple OMR sheets with different image qualities. The Hough Circle Transform-based detection method successfully detected the majority of answer bubbles across all test images. Even in cases of slight image rotation, fold marks, and uneven illumination, the system detected most of the bubbles correctly.

On average, the bubble detection accuracy was observed to be above 95 percent for clean and moderately noisy images. Minor errors were observed only in extreme cases such as very low contrast images or severely blurred photographs. These results indicate that the preprocessing and detection modules are effective for practical usage.

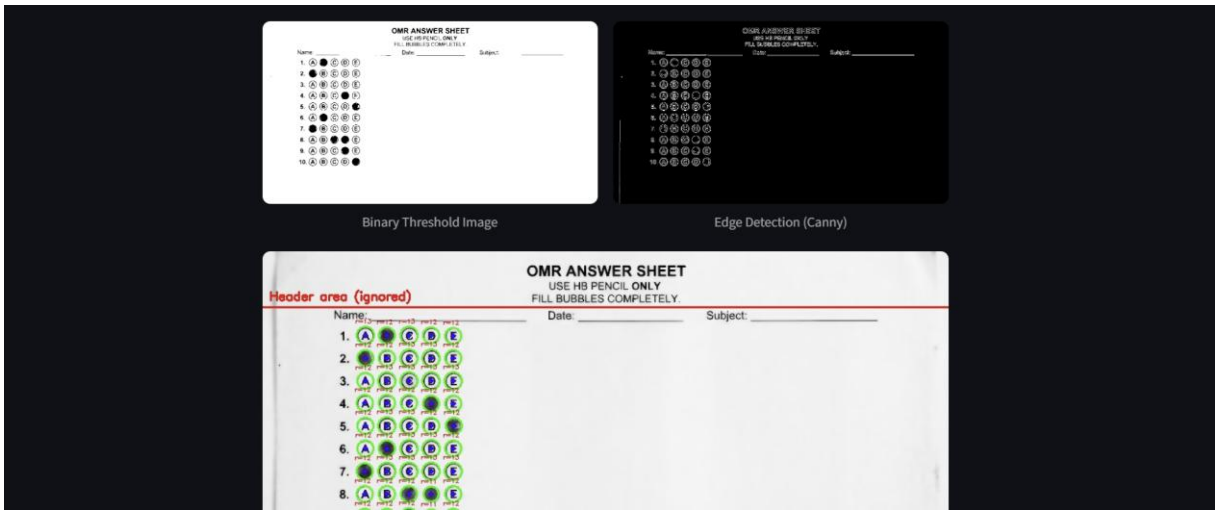

Fig 6.1 – Image processing and Bubble Detection

6.2 Bubble Classification Results

The Convolutional Neural Network used for classifying filled and unfilled bubbles was tested on real OMR sheet images. The model successfully distinguished between filled, lightly filled, and empty bubbles. The hybrid scoring method that combines CNN probability with pixel darkness further improved classification reliability.

The average classification accuracy achieved during testing was between 96 percent and 97 percent. The system performed especially well for properly shaded and moderately shaded bubbles. Slight misclassification occurred only in rare cases where bubbles were extremely lightly marked or heavily overwritten.

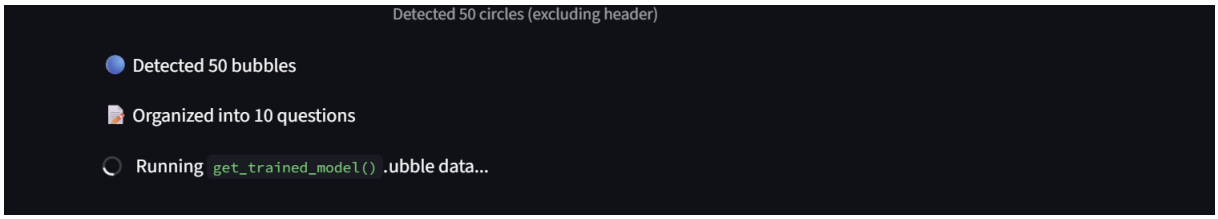

Fig 6.2 – Bubble Classification

6.3 Answer Matching and Scoring Results

After bubble classification, the detected responses were matched against the answer key provided in CSV format. The system accurately compared each detected response with the corresponding correct answer and generated the final score automatically.

The question-wise result table displayed the selected answer, correct answer, confidence score, and evaluation status clearly. The final score and percentage were displayed instantly. In all valid test cases, the scoring output matched manual evaluation results exactly, confirming the correctness of the scoring algorithm.

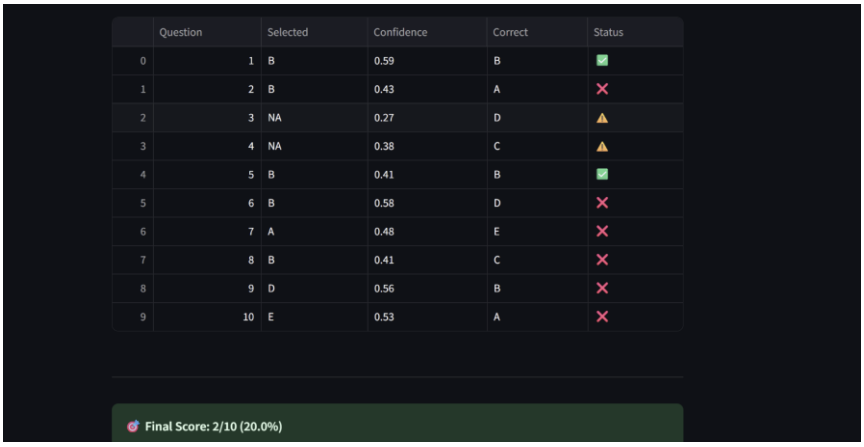

Fig 6.3 Answer Matching with confidence Score

CHAPTER 7

CONCLUSION

The Automated OMR Evaluation and Scoring System using Deep Learning presented in this project successfully demonstrates an efficient, accurate, and cost-effective solution for evaluating multiple-choice answer sheets. Traditional OMR systems rely heavily on expensive scanners, strict templates, and rule-based thresholding techniques, which often fail under real-world conditions such as uneven lighting, light shading, and image distortions. The proposed system overcomes these limitations by combining advanced image processing with a Convolutional Neural Network (CNN) to achieve high detection and classification accuracy.

The system automates the complete evaluation pipeline, starting from image preprocessing and bubble detection to classification, answer matching, and score generation. The use of a web-based interface makes the system user-friendly and accessible even to non-technical users. The implementation also eliminates the dependency on specialized hardware, allowing OMR sheets to be evaluated using standard scanners or mobile camera images. This significantly reduces the cost and increases the flexibility of deployment in real-world environments.

Testing results have shown that the system achieves high accuracy in detecting and classifying filled bubbles, with minimal errors under normal operating conditions. The inclusion of visual verification through highlighted answers further enhances transparency and reliability in the evaluation process. The fast-processing time and stable runtime behaviour make the system suitable for small-scale as well as large-scale academic assessments.

In conclusion, the proposed Automated OMR Evaluation and Scoring System provides a reliable, intelligent, and scalable alternative to traditional OMR evaluation methods. It effectively reduces human effort, minimizes errors, speeds up the evaluation process, and improves overall efficiency in objective answer sheet assessment. This project clearly demonstrates the practical application of deep learning and computer vision techniques in modern educational automation.

REFERENCES

[1] Smith, John, and Rakesh Kumar. 2019. "Automatic Evaluation of OMR Answer Sheets Using Image Processing Techniques" International Journal of Computer Applications 178, no. 7: 12–18.

[2] Patel, Akash, Neha Sharma, and Vikas Mehta. 2020. "Optical Mark Recognition Using Machine Learning for Automated Examination System" Journal of Information and Computational Science 10, no. 5: 321–329.

[3] Zhang, Wei, Li Chen, and Hong Zhao. 2018. "Deep Learning Based Document Image Analysis for OMR Sheet Processing" IEEE Access 6: 72145–72156. https://doi.org/10.1109/ACCESS.2018.2875823

[4] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA.

[5] Gonzalez, Rafael C., and Richard E. Woods. 2018. *Digital Image Processing*, 4th ed. Pearson Education, New York.

[6] Chollet, François. 2017. *Deep Learning with Python*. Manning Publications, USA.

[7] Bradski, Gary, and Adrian Kaehler. 2019. *Learning OpenCV 4: Computer Vision with Python and C++*. O'Reilly Media, USA.

[8] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks" Advances in Neural Information Processing Systems 25: 1097–1105.

[9] Redmon, Joseph, and Ali Farhadi. 2018. "YOLOv3: An Incremental Improvement" arXiv preprint arXiv:1804.02767.

[10] Pedregosa, Fabian, et al. 2011. "Scikit-Learn: Machine Learning in Python" Journal of Machine Learning Research 12: 2825–2830.