٢٠٢٥

Questionnaire Analysis

QUESTIONNAIRE ANALYSIS REPORT

HAMZA MOHAMMED

Name: Hamza Obaid
ID: 120211251
Subject: Digital Image processing Project

Contains

Introduction

This project focuses on the automated evaluation of multiple-choice answer sheets using image processing techniques. The system processes scanned images of answer sheets to extract and evaluate the marked answers.

Methodology

Image Preprocessing

Conversion to Grayscale:

The system converts the input image into grayscale to simplify the processing steps.

Binary Thresholding:

A binary threshold is applied to the grayscale image to isolate the answer bubbles. This step enhances the contrast between the marked and unmarked areas.

Contour Detection and Perspective Correction

Largest Contour Detection:

The algorithm identifies the largest contour in the thresholded image, assuming it represents the boundary of the answer sheet.

Warping the Image:

To correct any perspective distortions, the identified contour is used to warp the image into a top-down view, ensuring that the grid layout is accurately aligned.

Answer Extraction

Splitting the Image into Boxes:

Once the image is correctly oriented, it is divided into a grid of boxes. Each box corresponds to an individual answer bubble.

Evaluation of Marked Responses:

The system analyzes each box by counting the number of marked pixels. A threshold is applied to determine whether a particular bubble has been marked, and the selected answers are then recorded.

Implementation

Programming Language and Librarie

The project is implemented in Python using:

- OpenCV for image processing tasks.
- NumPy for numerical operations.
- Pandas for tabular data management.
- Matplotlib for image visualization.

Modular Design

The code is structured into functions for each major step:

- preprocess_image(): Handles grayscale conversion and thresholding.
- find_largest_contour(): Identifies the answer sheet boundary.
- warp_image(): Corrects the perspective.
- split_into_boxes(): Segments the image into answer areas.

- evaluate_answers(): Determines the chosen answers.
- process_image(): Integrates all the steps to process an input image.

User Interaction

The system accepts user input for the number of questions and answer choices, making it adaptable to various answer sheet formats.

Results and Testing

Test Cases

Multiple test cases were executed using different scanned images. The extracted answers are displayed in a structured table, and the system's performance depends on the quality of the input images and the precision of the contour detection.

Observations

The automated system demonstrates a high level of efficiency in detecting marked answers, although its accuracy is influenced by factors such as lighting conditions, image quality, and variations in answer sheet design.

Limitations

Assumption of Uniformity

The method assumes that the answer sheet has a consistent layout. Variations in the design or misalignment during scanning can affect the accuracy of contour detection and subsequent processing.

Fixed Thresholding

The threshold values used for detecting marked bubbles may not be optimal for all scenarios. Variations in marking intensity (e.g., lightly filled bubbles) might not be captured accurately.

Limited Adaptability

The current implementation is tailored for specific answer sheet formats and may require modifications to handle diverse layouts or different marking styles

Conclusion and Future Work

Conclusion

The project successfully automates the grading process for multiple-choice answer sheets, reducing the workload associated with manual grading.

Future Improvements

Potential areas for enhancement include:

- Incorporating deep learning models to improve detection accuracy.
- Expanding the system to handle various answer sheet formats automatically.
- Developing a graphical user interface (GUI) to facilitate ease of use for non-technical users.