DIGITAL IMAGE PROCESSING COURSE

EXAMINATION REPORT

02.04.2025

Pham Van Truong

22022564

Contents

1. Introduction **1**

2. Preprocessing **2**

3. Methods **3**

3.1. Template Matching ... 3

3.1.1. Operating Principle ... 3

3.1.2. The CCOEFF_NORMED Method ... 3

3.1.3. Detailed Implementation Steps ... 3

3.1.4. Advantages of the CCOEFF_NORMED Method ... 4

3.1.5. Limitations of the CCOEFF_NORMED Method ... 5

3.2. Feature Matching ... 5

4. Experiments and Results **5**

5. Discussion **7**

6. Conclusion **7**

1. Introduction

This report documents the process of detecting specific objects within a main image using digital image processing techniques, with a focus on template matching, as part of the mid-term exam for the Digital Image Processing course. The goal is to evaluate the effectiveness of different parameters and approaches in achieving accurate detection and visualization, leveraging the GitHub repository: Digital-Image-Processing-Course-Mid-Term-Exam. To enhance understanding, this report includes visual aids such as the initial image, preprocessed main image, example template, and final detection result.

Figure 1: The original image before any processing.

2. Preprocessing

Before preprocessing the images, the original image is first split into two sub-images: the main image and the templates image, which contains the objects that need to be detected.

Figure 2: Main image.

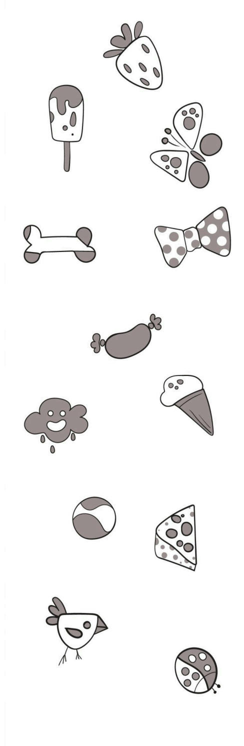

Figure 3: Templates image.

After that, each object in the templates image is isolated (by applying morphological closing with a larger kernel to connect object parts in templates image) and saved individually. Here is an example of one of the twelve objects.

Figure 4: Object 1.

To enhance the accuracy of object detection, the following preprocessing steps were applied to both the main image and the template images:

- Conversion to gray-scale to simplify the image data, reducing color complexity.
- Application of Gaussian blur to reduce noise, smoothing the image for better feature detection.

- Histogram equalization to improve contrast, making object features more distinguishable.

3. Methods

This section details the detection methods, focusing on template matching and briefly exploring feature matching.

3.1. Template Matching

Template matching is an image processing technique used to find areas of a template image within a larger source image.

3.1.1. Operating Principle

Template matching performs comparison between a template and a source image by sliding the template over each position in the source image and calculating the correlation between them. The result is a matrix of correlation values, where the highest value typically indicates the most suitable matching position.

3.1.2. The CCOEFF_NORMED Method

Normalized Cross-Correlation Coefficient (CCOEFF_NORMED) is one of the most effective methods for template matching. This method can be represented by the following formula:

$$R(x, y) = \frac{\sum_{x',y'} [T'(x', y') \cdot I'(x + x', y + y')]}{\sqrt{\sum_{x',y'} T'(x', y')^2 \cdot \sum_{x',y'} I'(x + x', y + y')^2}}$$

Where:
- T' is the template after subtracting the mean value
- I' is the region of the source image (with the same size as the template) after subtracting the mean value
- $R(x, y)$ is the correlation value at position (x, y) in the result image

3.1.3. Detailed Implementation Steps

The template matching process using the CCOEFF_NORMED method includes the following steps:

1. **Prepare input data**:
   - Source image: The larger image in which we want to search for the template
   - Template image: The smaller pattern that we want to find in the source image
2. **Preprocessing**:
   - Convert images to grayscale
   - Apply noise filtering techniques
3. **Calculate the correlation matrix**:

- For each position (x, y) in the source image:
  1. Extract the image region $I_{x,y}$ with the same size as the template
  2. Calculate the mean value of the template $\overline{T}$ and the image region $\overline{I}_{x,y}$
  3. Calculate: $T' = T - \overline{T}$ and $I'_{x,y} = I_{x,y} - \overline{I}_{x,y}$
  4. Calculate the numerator:

$$\sum_{x',y'} [T'(x', y') \cdot I'(x + x', y + y')]$$

  5. Calculate the denominator:

$$\sqrt{\sum_{x',y'} T'(x', y')^2 \cdot \sum_{x',y'} I'(x + x', y + y')^2}$$

  6. Calculate the correlation value $R(x, y)$ by dividing the numerator by the denominator

4. **Normalize the results**:
   - The value of $R(x, y)$ is normalized in the range $[-1, 1]$
   - A value of 1 means a perfect match
   - A value of −1 means an inverse match (the template and image region are negative versions of each other)
   - A value of 0 means no correlation
5. **Determine matching positions**:
   - Find the coordinates $(x_{\max}, y_{\max})$ where $R(x, y)$ reaches its maximum value
   - If the value $R(x_{\max}, y_{\max})$ exceeds a certain threshold, we consider it a matching position
   - In case we want to find multiple matching positions, we can apply non-maximum suppression or find local maxima
6. **Final result**:
   - Mark or extract the matching region from the source image
   - Draw a bounding rectangle around the matching region with dimensions corresponding to the template

3.1.4. Advantages of the CCOEFF_NORMED Method

- **Brightness invariance**: This method is not affected by linear changes in image brightness due to normalization
- **Normalized values**: Results are in the range $[-1, 1]$, making it easy to set thresholds
- **Efficiency**: This method typically provides better results compared to simpler methods like SAD (Sum of Absolute Differences) or SSD (Sum of Squared Differences)
- **Accurate detection**: Provides high accuracy when the template and image region have similar structures

3.1.5. Limitations of the CCOEFF_NORMED Method

- **Not rotation invariant**: This method is not effective when the template is rotated relative to the source image
- **Not scale invariant**: Not effective when the template and object in the source image have different sizes
- **Computationally expensive**: Requires many calculations, especially with large images
- **Sensitive to occlusion**: Performance decreases when the object is partially occluded

3.2. Feature Matching

An alternative approach using feature detection and matching was explored but not pursued due to suboptimal results in bounding box accuracy, as it failed to enclose objects clearly.

4. Experiments and Results

Several experiments were conducted to optimize the detection process:

1. **Initial Attempt:**
    - Confidence threshold: 0.7
    - Detected: 2 out of 12 objects
    - Issue: High threshold led to missed detections, indicating it was too restrictive.
2. **Second Attempt:**
    - Confidence threshold: 0.4
    - Detected: 12 out of 12 objects
    - Issue: Incorrect bounding boxes for objects 4, 5, 6, and 8, detected at a scale of 0.5 instead of 0.75, suggesting scale was a critical factor.
3. **Adjusted Scale Range:**
    - Set minimum scale to 0.75 to align with the observed effective scale for most objects.
    - Result: Accurate detection of all 12 objects with correct bounding boxes, resolving previous issues.
4. **Feature Matching Attempt**

- Detected all 12 objects but with poorly drawn bounding boxes, leading to the decision to abandon this approach in favor of template matching.

To summarize the experiments, the following table outlines the key parameters and results:

Experiment	Confidence Threshold	Minimum Scale	Objects Detected	Bounding Box Accuracy
First Attempt	0.7	0.5	2/12	N/A
Second Attempt	0.4	0.5	12/12	Poor (objects 4,5,6,8)
Final Adjustment	0.4	0.75	12/12	Accurate
Feature Matching	N/A	N/A	12/12	Poor

Here is the final detection result:

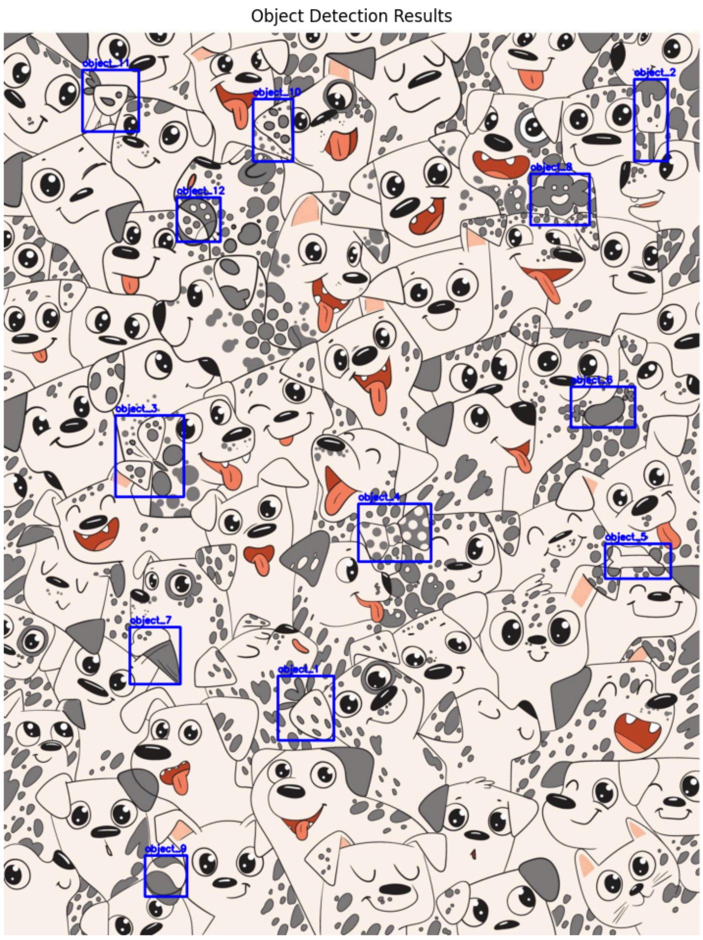

Figure 5: The final detection result with accurate bounding boxes.

5. Discussion

The initial high confidence threshold of 0.7 was too restrictive, leading to many missed detections. Lowering it to 0.4 allowed for the detection of all objects but introduced inaccuracies in bounding boxes, particularly for objects detected at a scale of 0.5. This highlighted the importance of scale in template matching, as objects 4, 5, 6, and 8 required a scale closer to 0.75 for accurate representation. Adjusting the minimum scale to 0.75 resolved these issues, demonstrating that scale alignment is critical for bounding box accuracy.

The feature matching approach, while capable of detecting all objects, failed to provide precise bounding boxes, likely due to its reliance on feature points rather than global image similarity, making it less suitable for this application.

6. Conclusion

Through iterative adjustments to the template matching parameters, particularly the scale range, the project successfully achieved accurate detection and localization of all target objects within the main image. This demonstrates the effectiveness of template matching when properly tuned for the specific characteristics of the images and objects involved. Future work could explore automated methods for parameter tuning or the integration of machine learning techniques to further enhance detection capabilities, potentially addressing cases where manual adjustments are less effective.