# BLG453E - Computer Vision
# Homework 3

**Student:** Emil Huseynov
**Student ID:** 150210906
**Course:** BLG453E

**Assignment:** Homework 3
**Due Date:** January 5, 2024

# Contents

# 1 Introduction

This report presents the solutions to the tasks in BLG453E - Computer Vision Homework 3. Each question involves implementing computer vision algorithms from scratch, demonstrating the results, and analyzing them.

# 2 Question 1: Template and Feature-Based Matching

## 2.1 Problem Description

In this task, we aim to locate Luigi and Wally in their respective scenes using both template-based techniques (Cross-Correlation, SAD, SSD) and feature-based methods (SIFT). The goal is to evaluate and compare these techniques in terms of accuracy and robustness.

## 2.2 Methodology

### 2.2.1 Template Matching

The following template matching techniques were implemented:

- **Cross-Correlation:** Measures similarity between the template and the region of interest.
- **Sum of Absolute Differences (SAD):** Computes the absolute difference between the template and the region.
- **Sum of Squared Differences (SSD):** Computes the squared difference between the template and the region.

### 2.2.2 Feature-Based Matching

SIFT (Scale-Invariant Feature Transform) was utilized to detect and match key points between the templates and the scenes.

## 2.3 Results and Analysis

Results for each method were visualized in the Jupyter Notebook. Key observations:

- Template-based methods struggled with occlusions and slight rotations.

- Feature-based methods (SIFT) were robust to occlusions and rotations but computationally expensive.

Figures showing annotated scenes and template matches are presented below:

# 3 Question 2: Noise Inclusion and Gradient Magnitudes

## 3.1 Problem Description

The task involves adding Gaussian noise to an image, removing the noise using Gaussian filters, and calculating gradient magnitudes using Sobel filters.

## 3.2 Methodology

- **Noise Inclusion:** Gaussian noise with varying standard deviations ($\sigma$) was added to the images.
- **Noise Removal:** A 5x5 Gaussian filter was implemented to smooth the noisy images.
- **Gradient Magnitudes:** Sobel filters were applied to compute gradients in the x and y directions, followed by calculating the gradient magnitude.

## 3.3 Results and Analysis

Figures illustrating the process are provided below:

# 4 Question 3: Hough Transform for Line and Circle Detection

## 4.1 Problem Description

This task involves detecting straight lines and circles in an image using the Hough Transform. The results of the custom implementation were compared with OpenCV's built-in functions.

## 4.2 Methodology

- **Edge Detection:** The Canny Edge Detection algorithm was used to identify edges in the image.
- **Hough Transform:** The accumulator-based algorithm was implemented for line and circle detection.

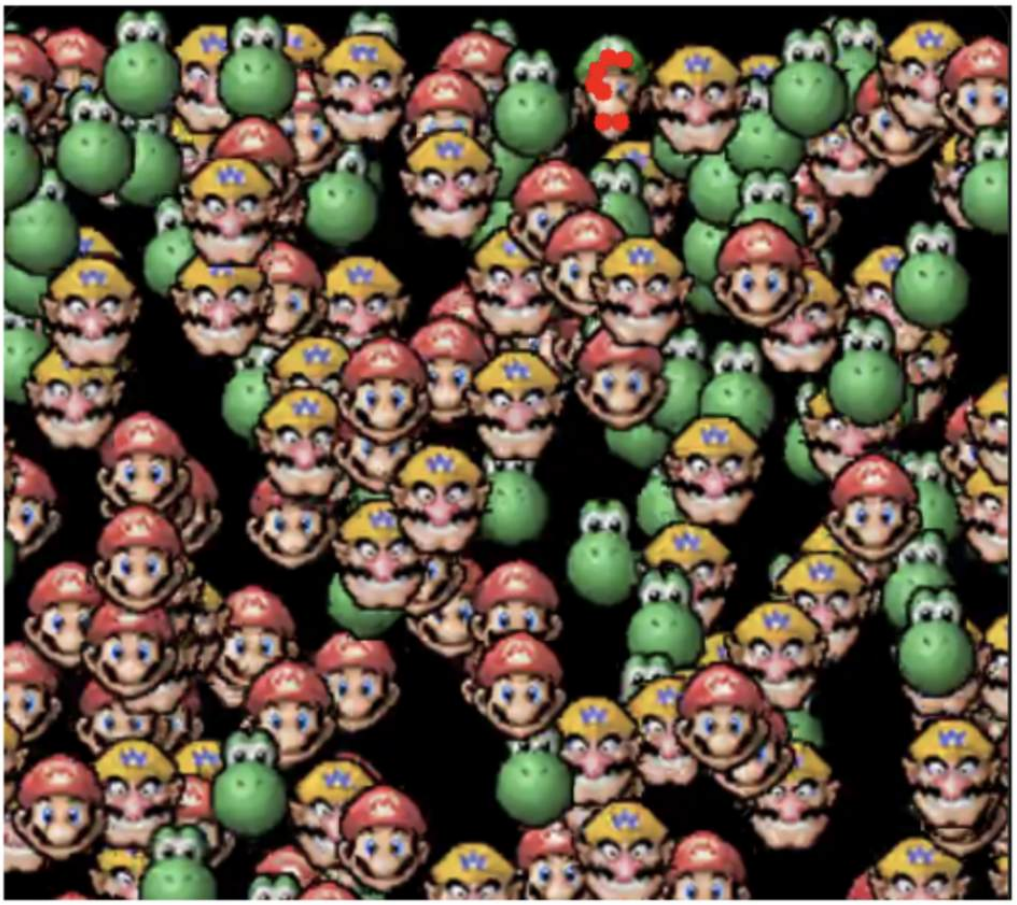
## 4.3 Results and Analysis

# 5 Conclusion

This homework demonstrated the implementation of fundamental computer vision techniques. The results highlight the strengths and limitations of each method, emphasizing the importance of selecting appropriate techniques based on the application.

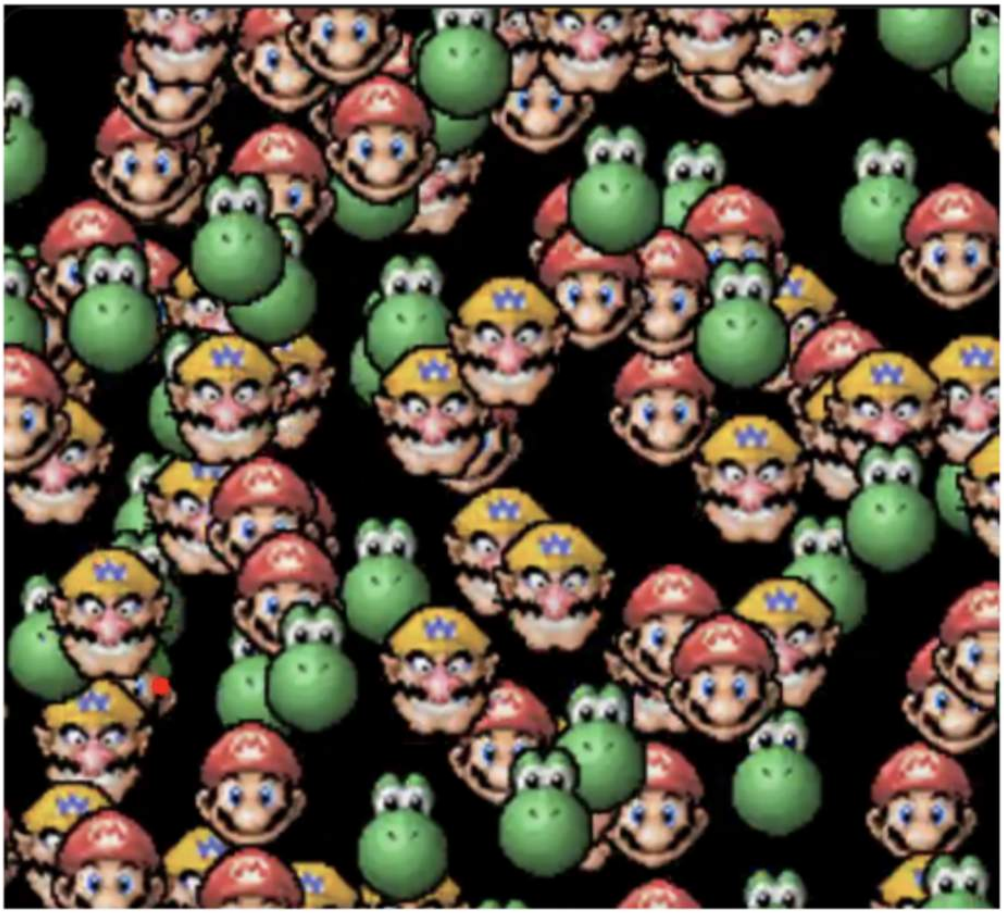
# 6 Results

Matches for Scene1 + Normal Luigi

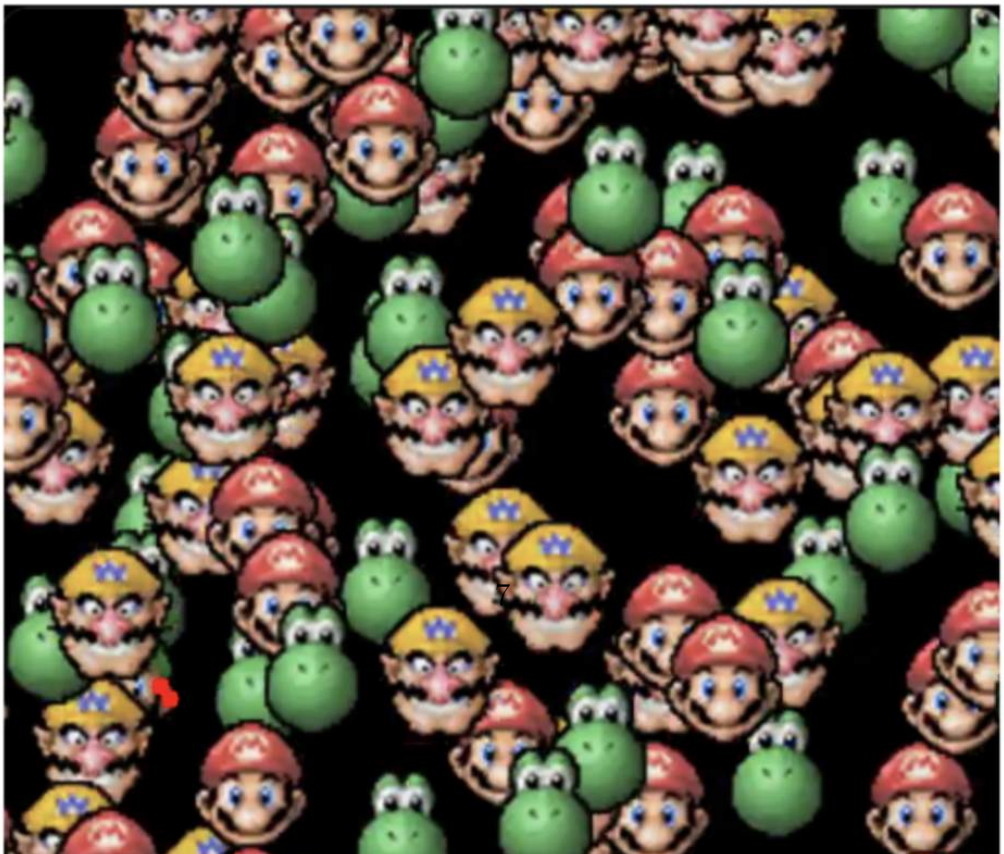


Matches for Scene1 + Rotated Luigi

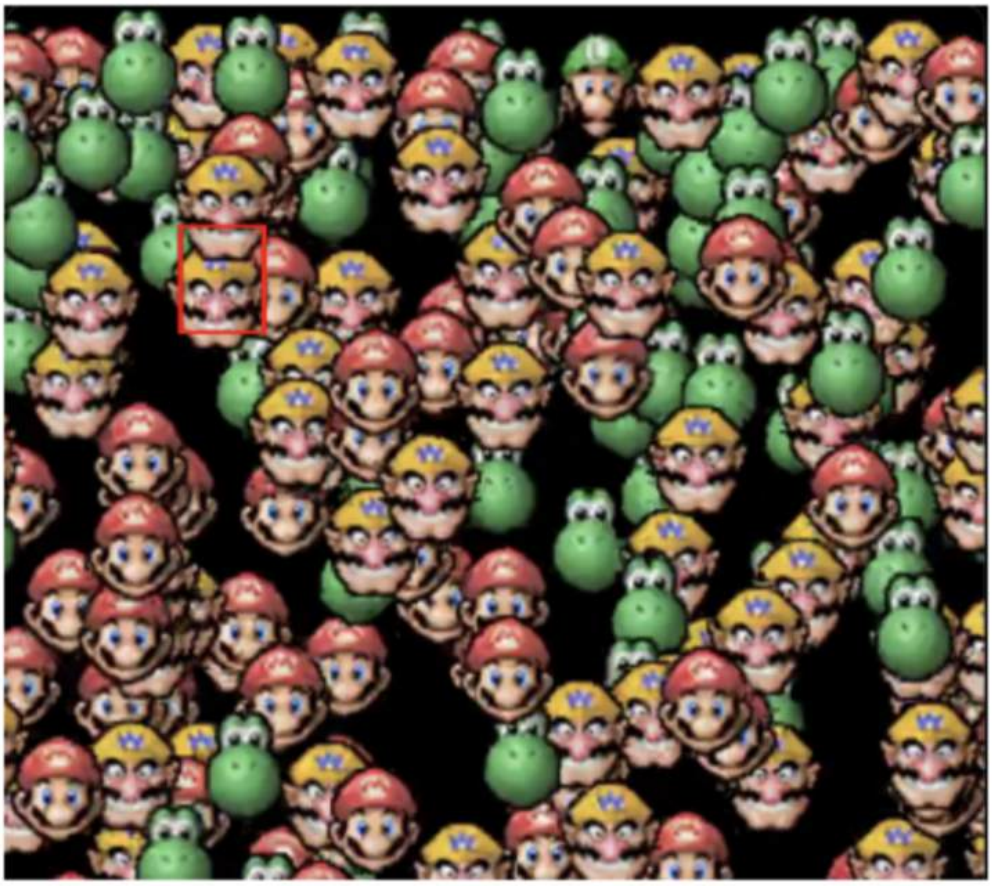
Matches for Scene2 + Normal Luigi



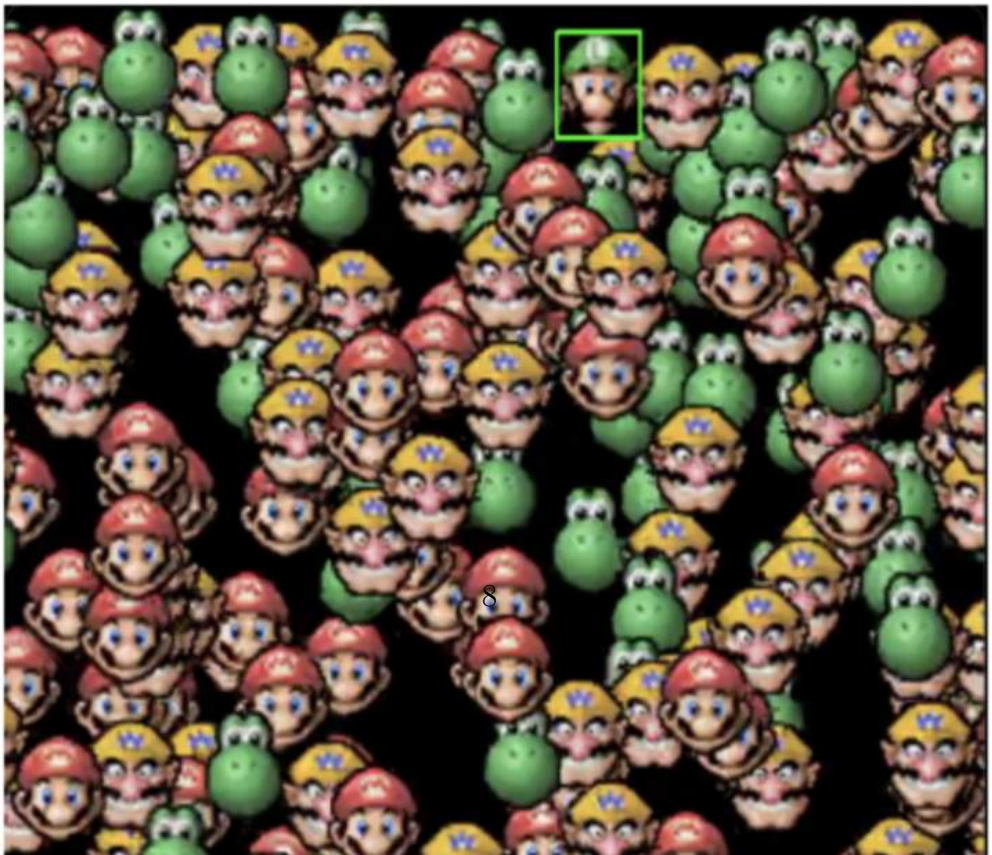
Matches for Scene2 + Rotated Luigi



Scene 1 - Normal Luigi - CC results

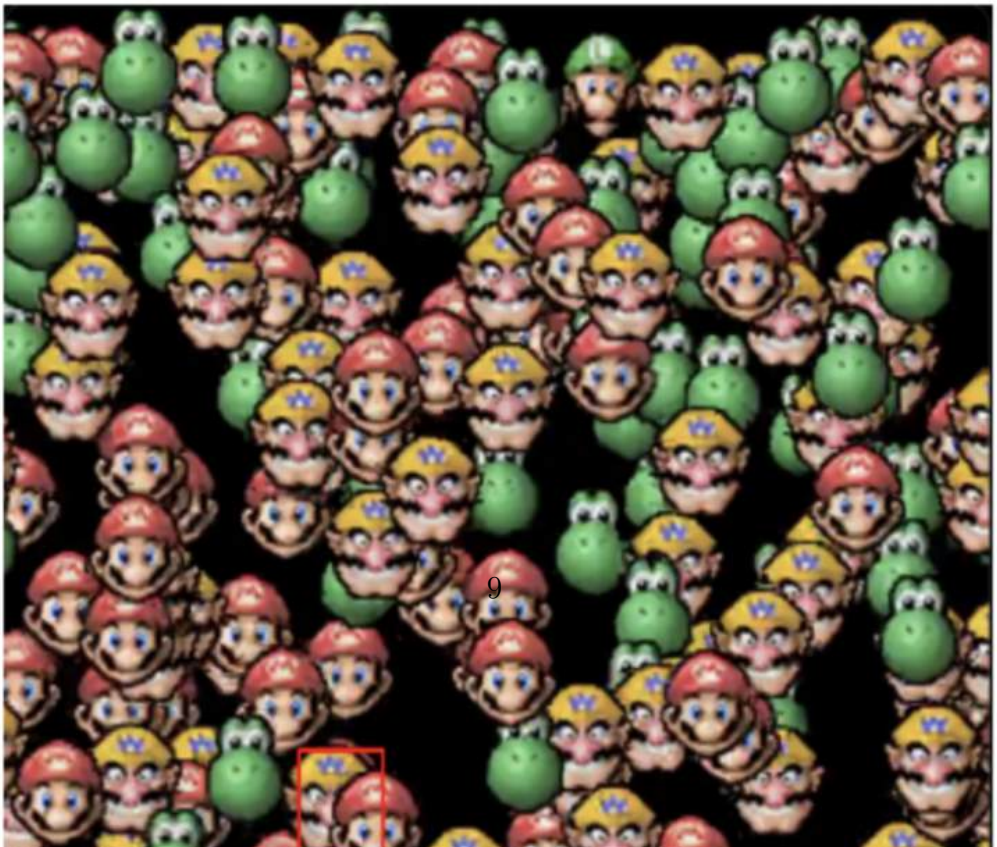


Scene 1 - Normal Luigi - SAD results



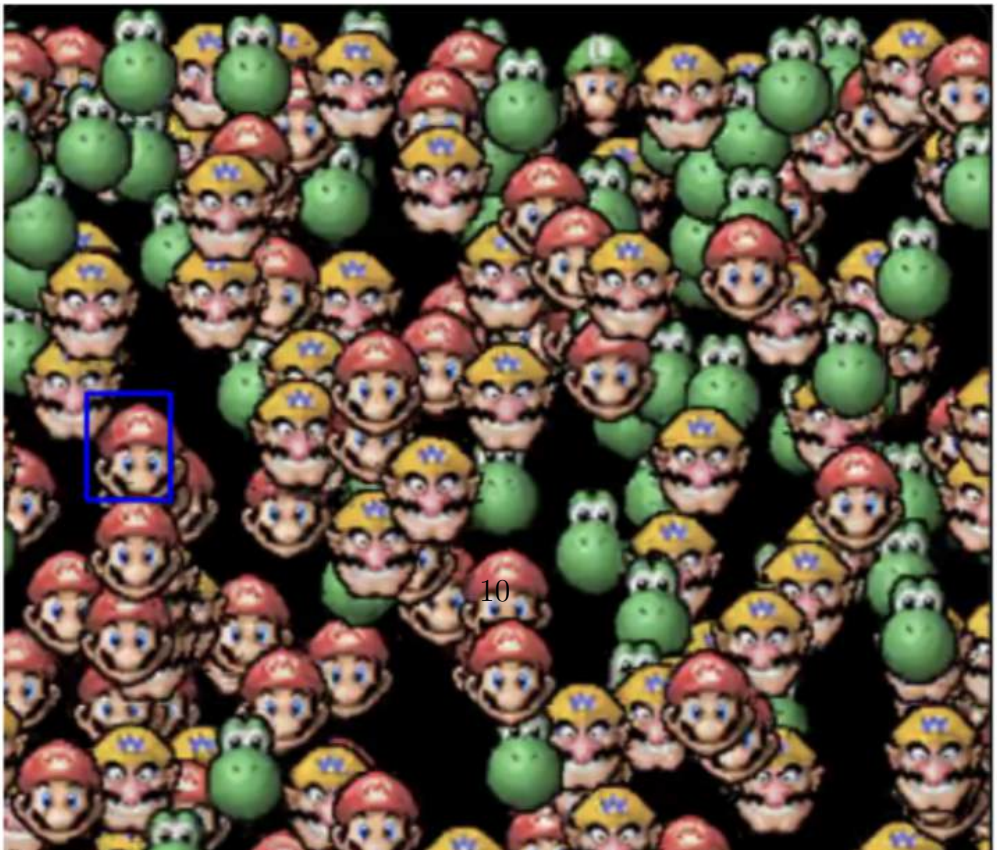
Scene 1 - Normal Luigi - SSD results

Scene 1 - Rotated Luigi - CC results

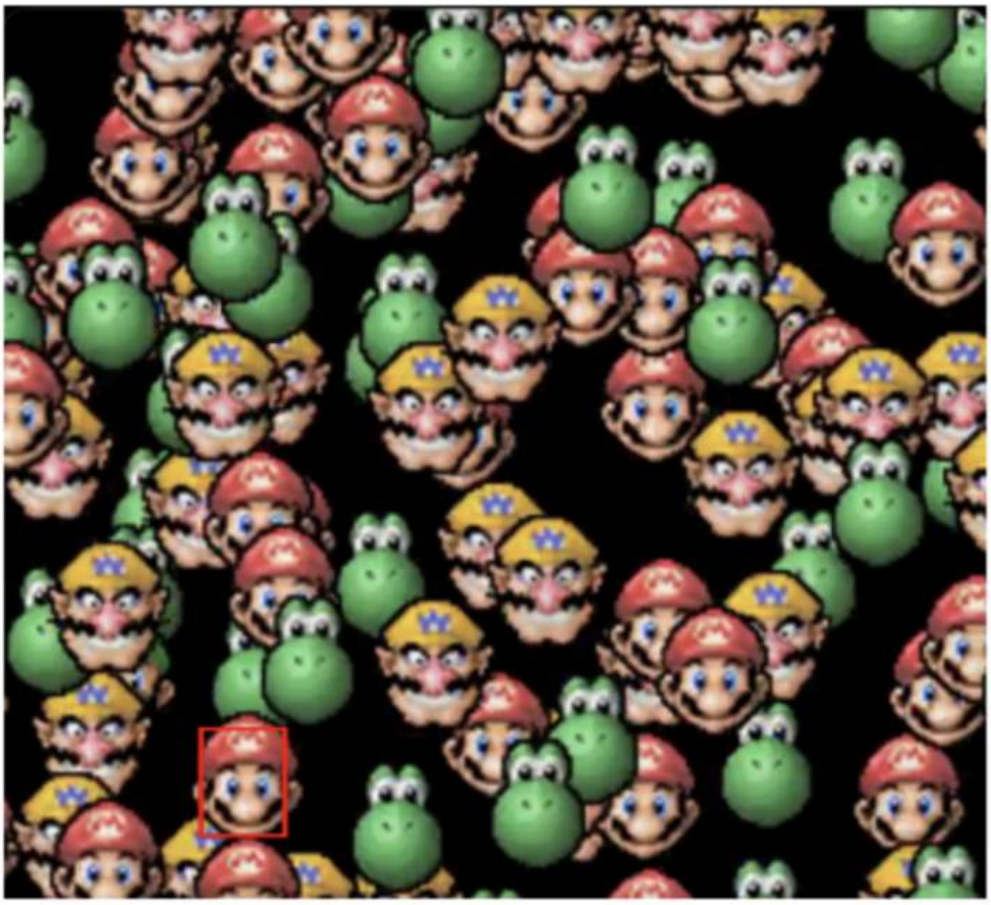


Scene 1 - Rotated Luigi - SAD results

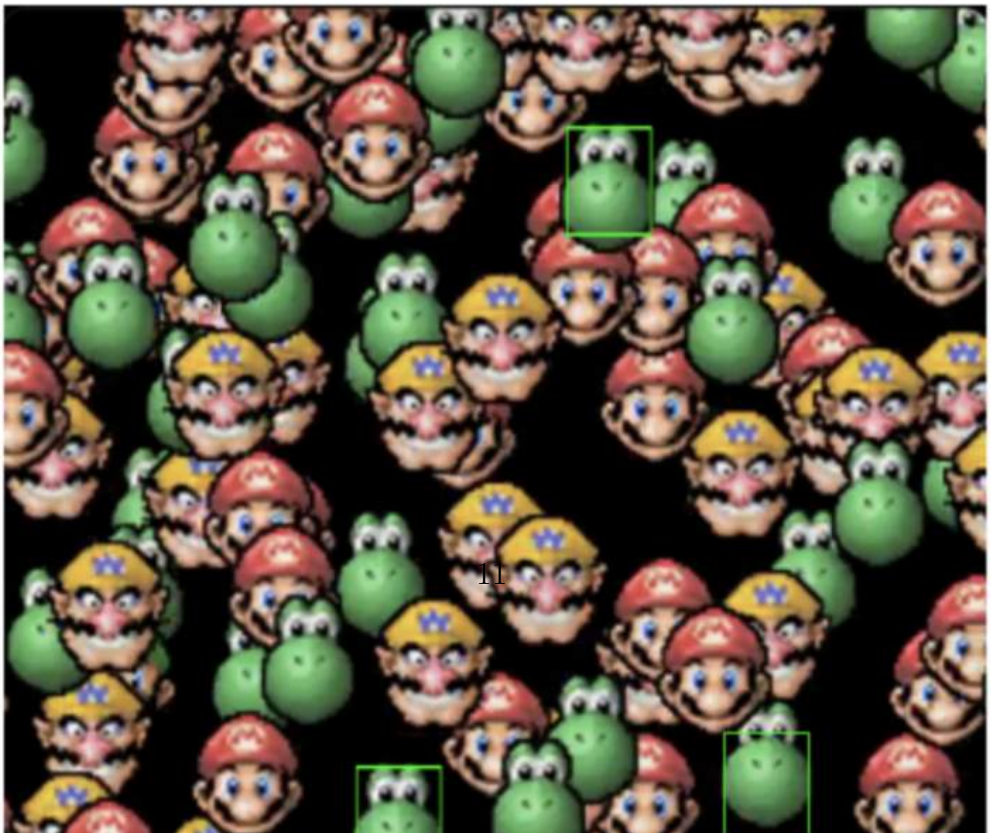
Scene 1 - Rotated Luigi - SSD results



Scene 2 - Normal Luigi - CC results



Scene 2 - Normal Luigi - SAD results



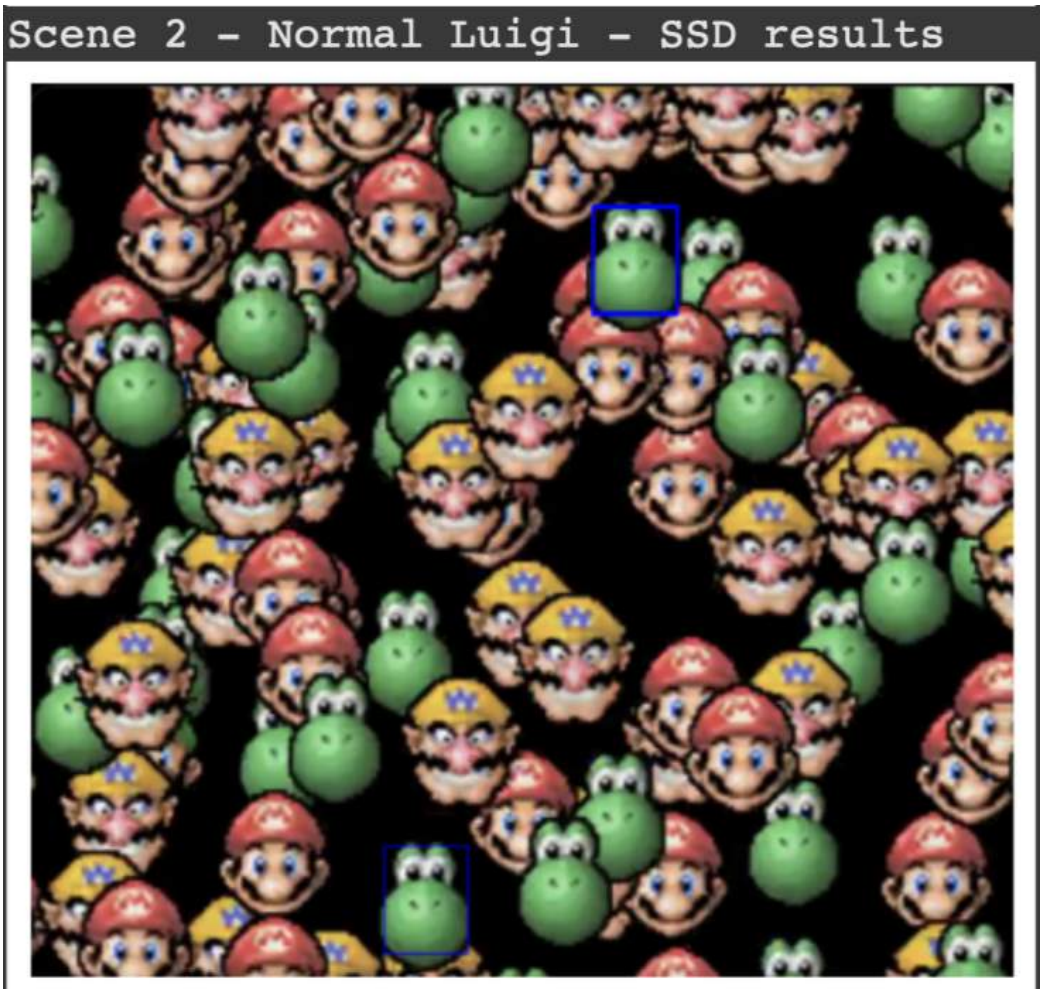



Figure 7: Template Based Matching Results for Luigi

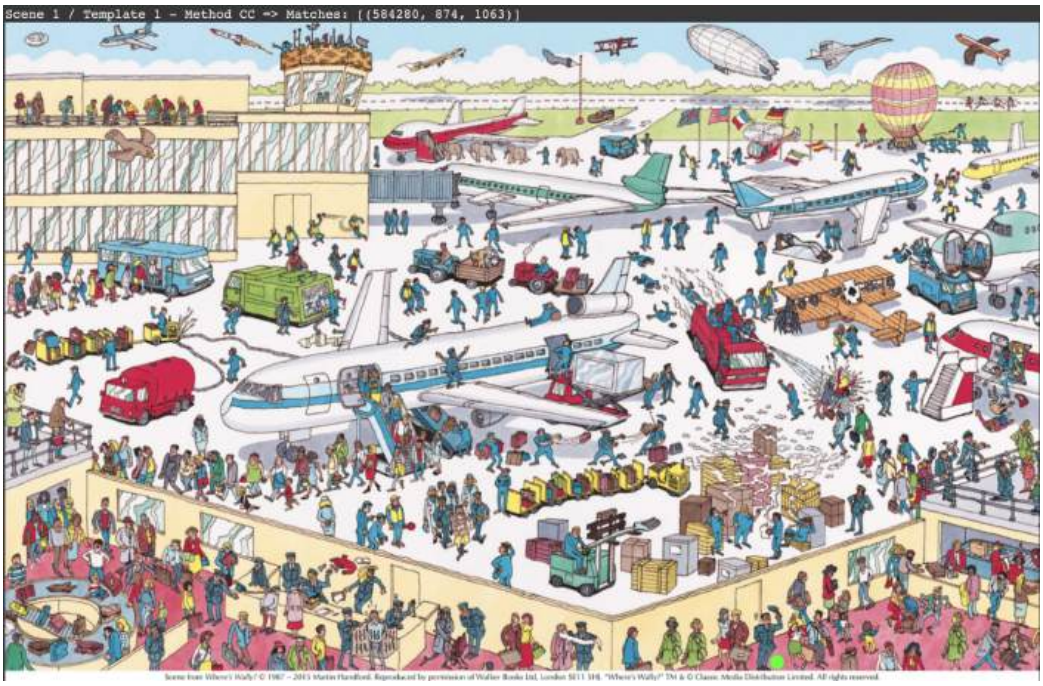



Figure 8: Template Based Matching Results for Wally

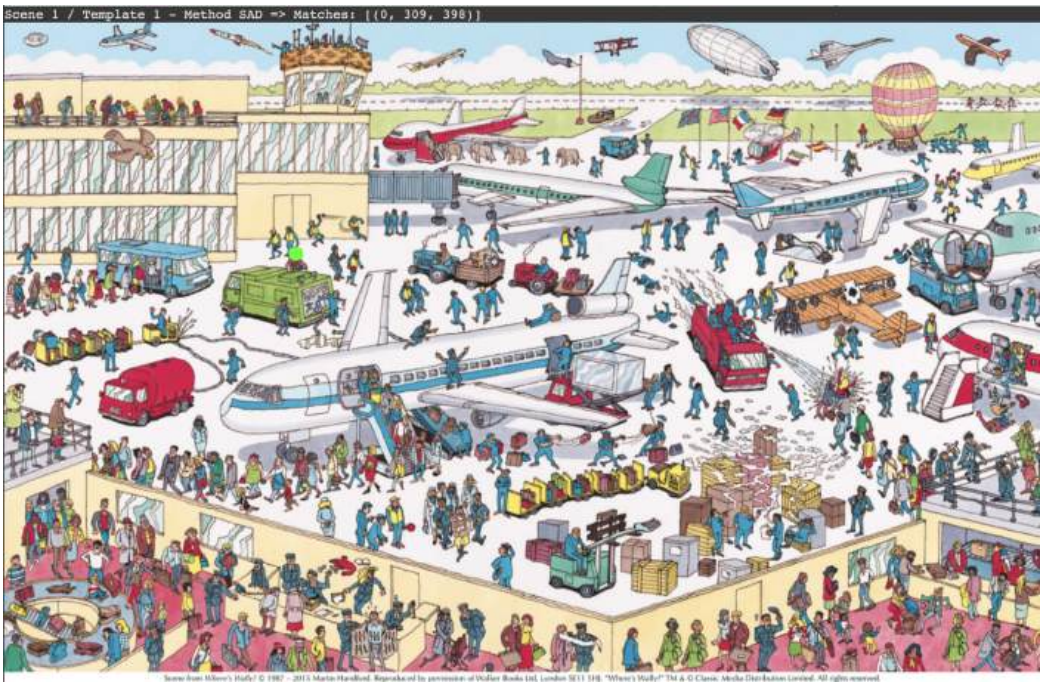



Figure 9: Template Based Matching Results for Wally

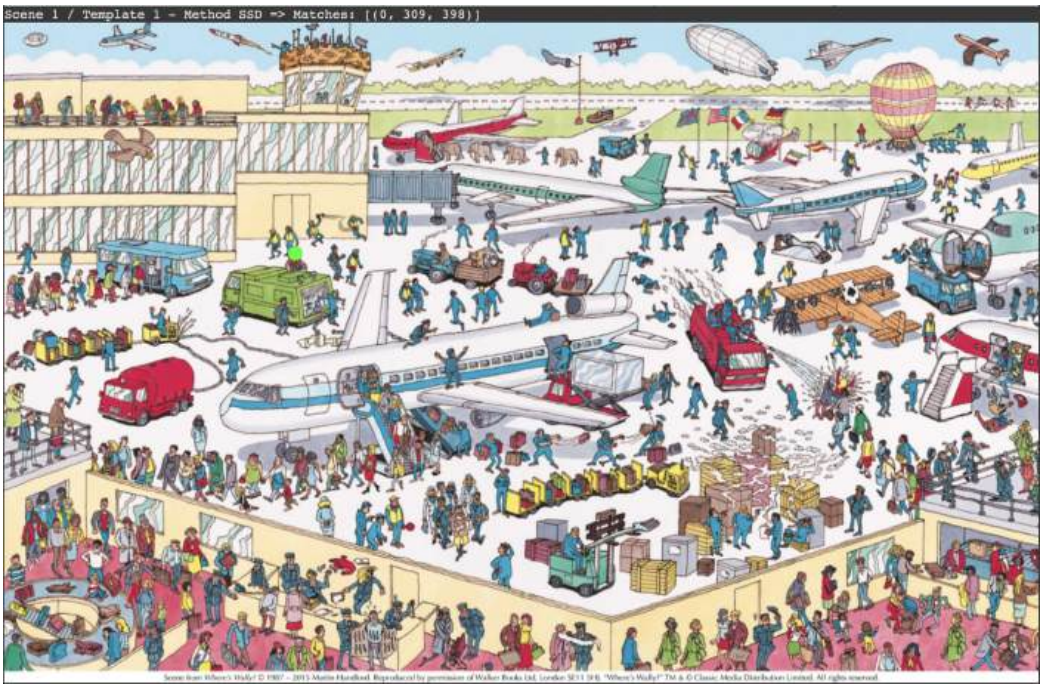



Figure 10: Template Based Matching Results for Wally

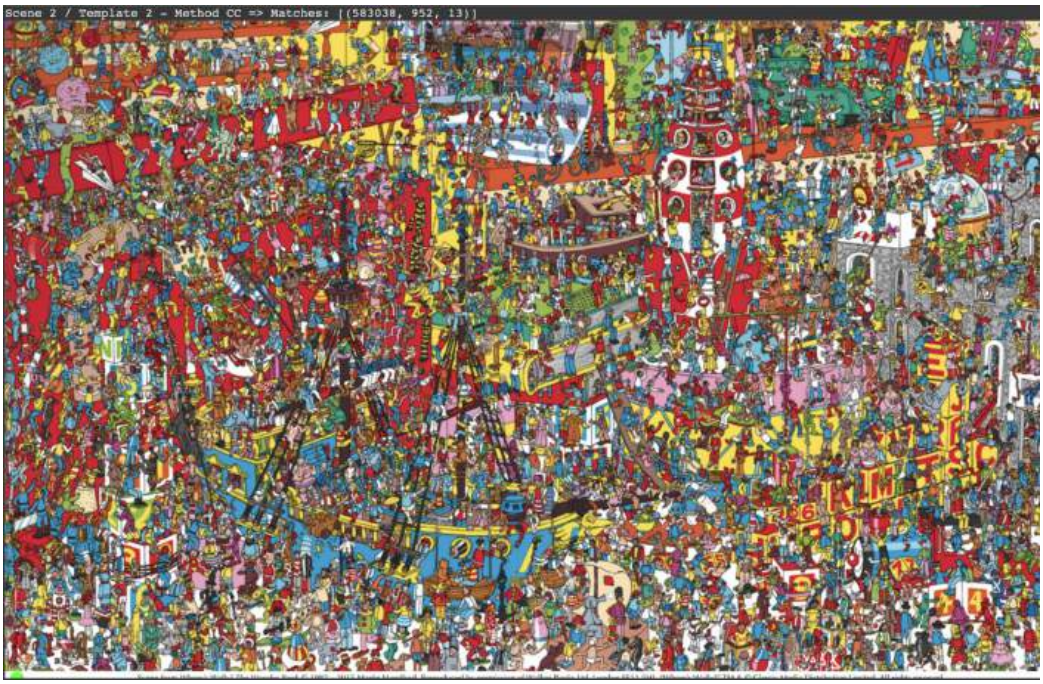



Figure 11: Template Based Matching Results for Wally


Figure 12: Template Based Matching Results for Wally

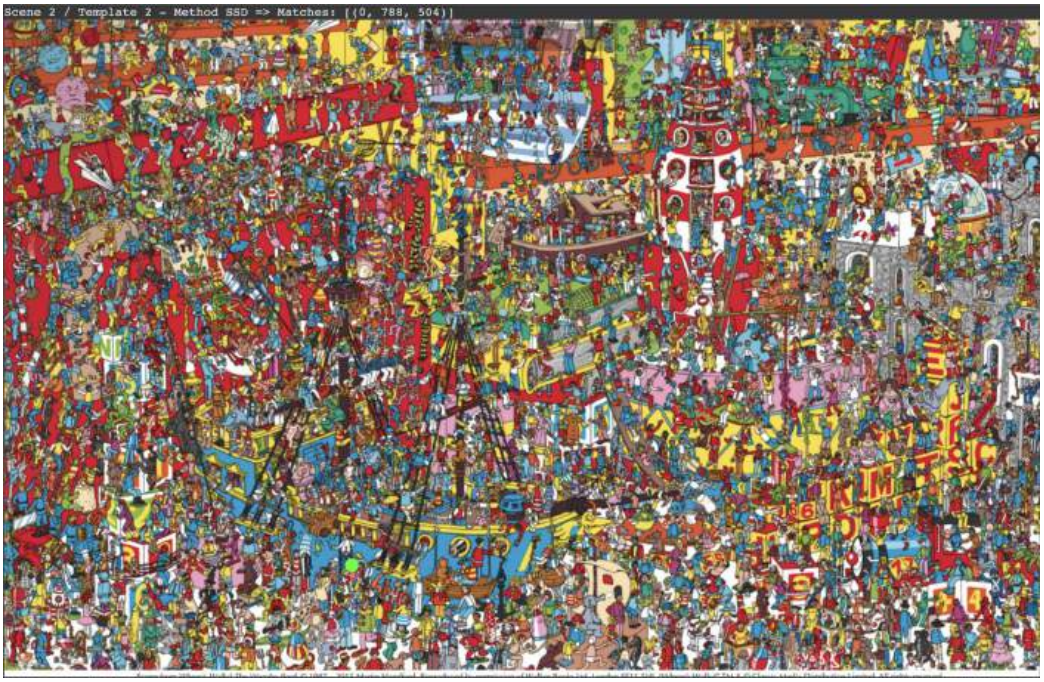



Figure 13: Template Based Matching Results for Wally

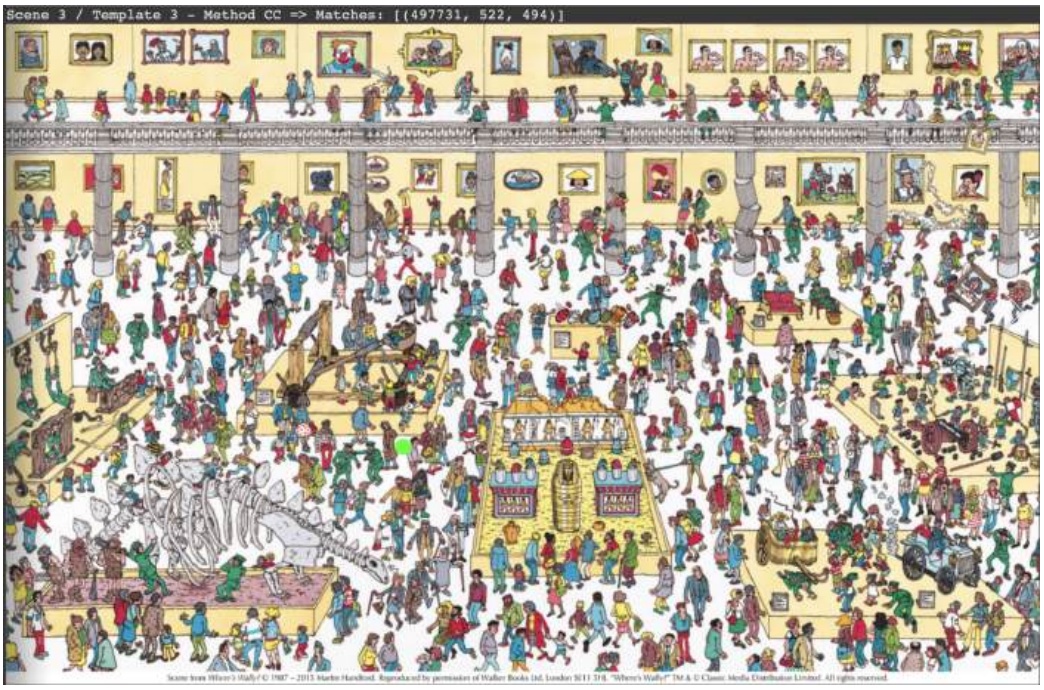



Figure 14: Template Based Matching Results for Wally

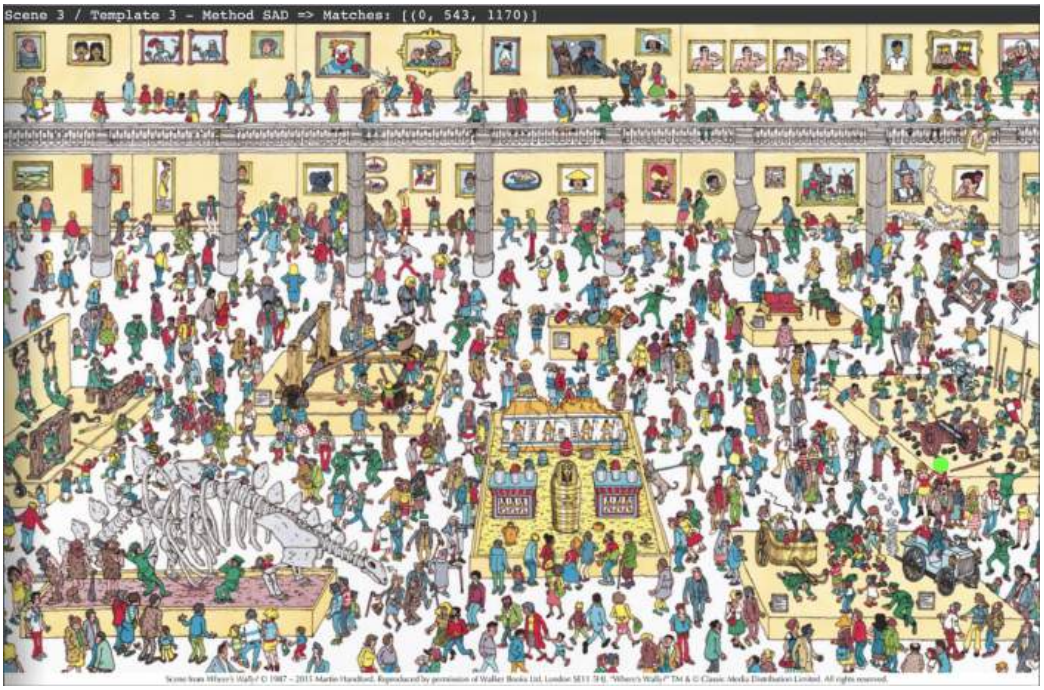



Figure 15: Template Based Matching Results for Wally

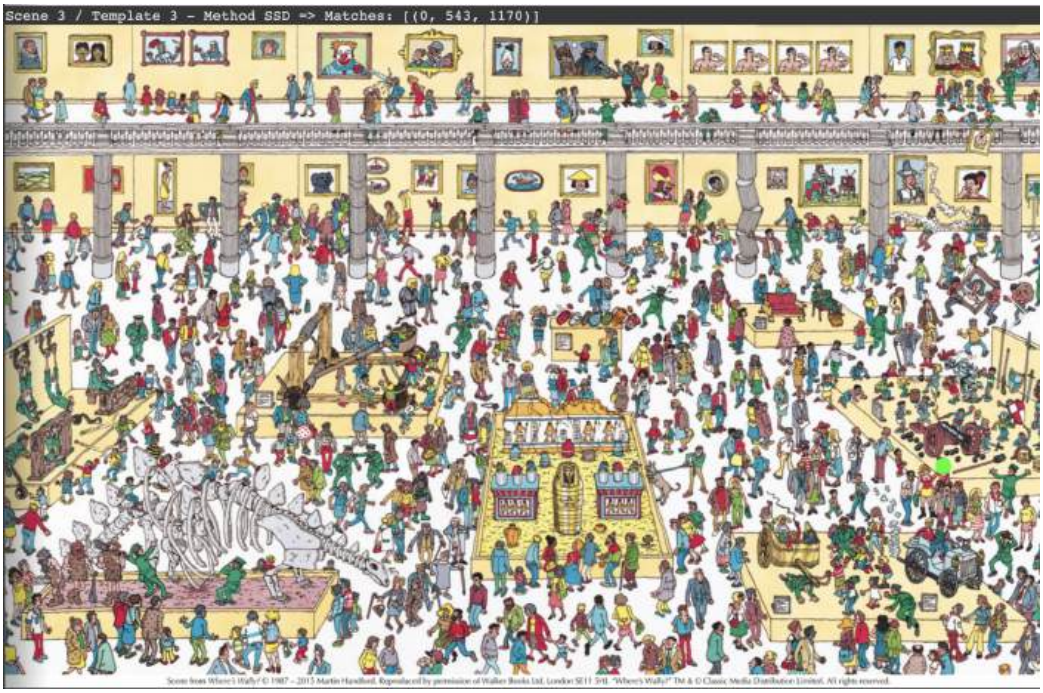



Figure 16: Template Based Matching Results for Wally

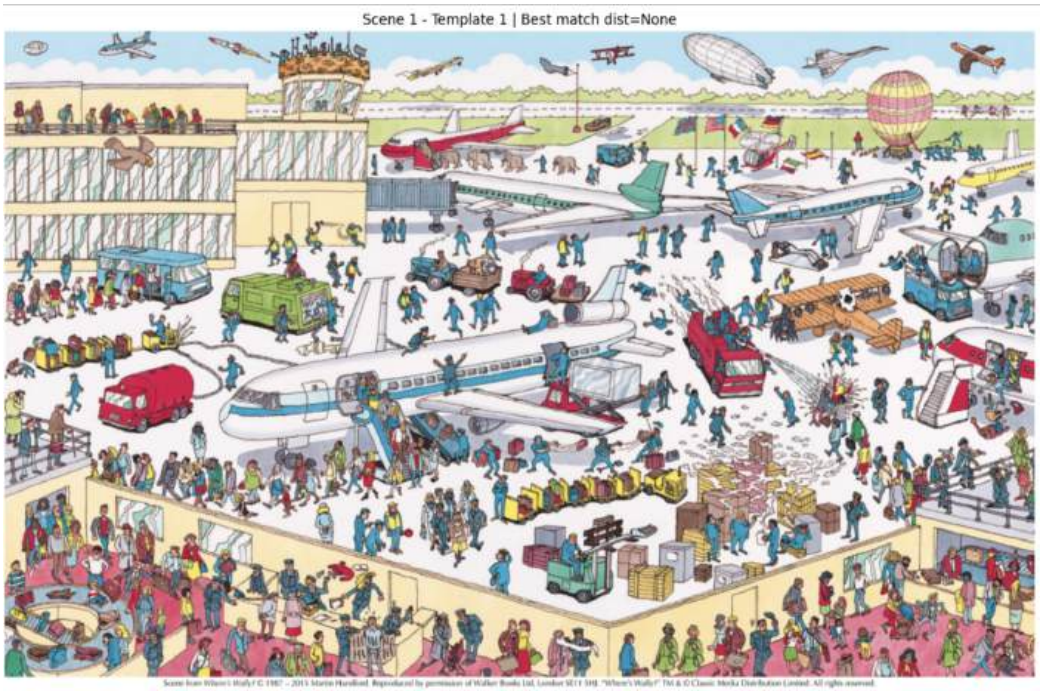



Figure 17: Feature-Based Based Matching Results for Wally

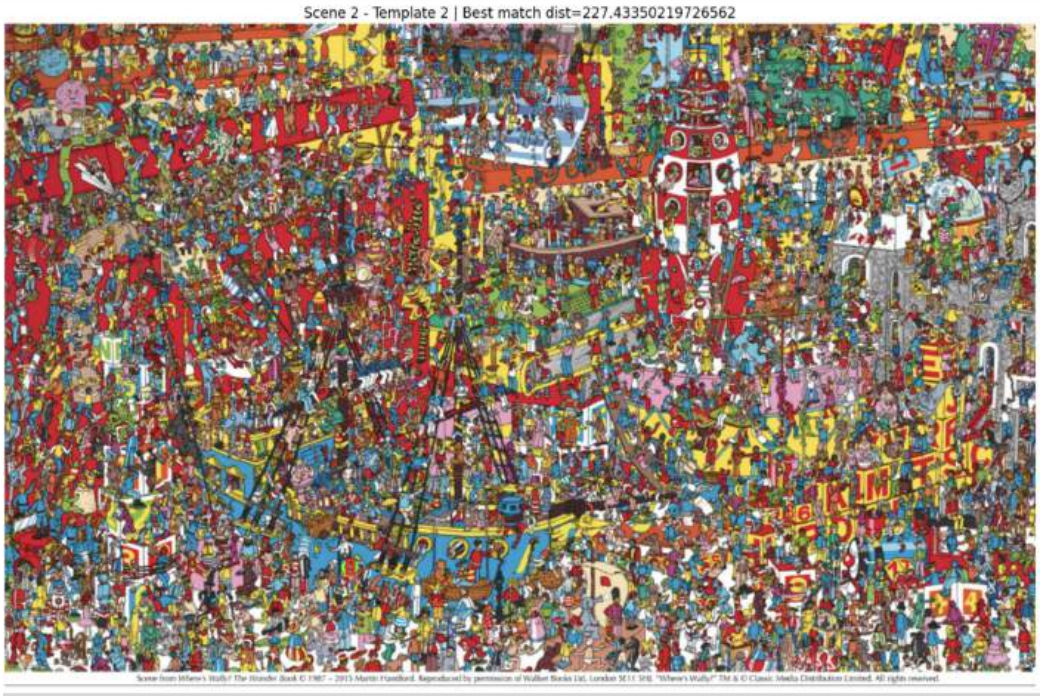



Figure 18: Feature-Based Based Matching Results for Wally

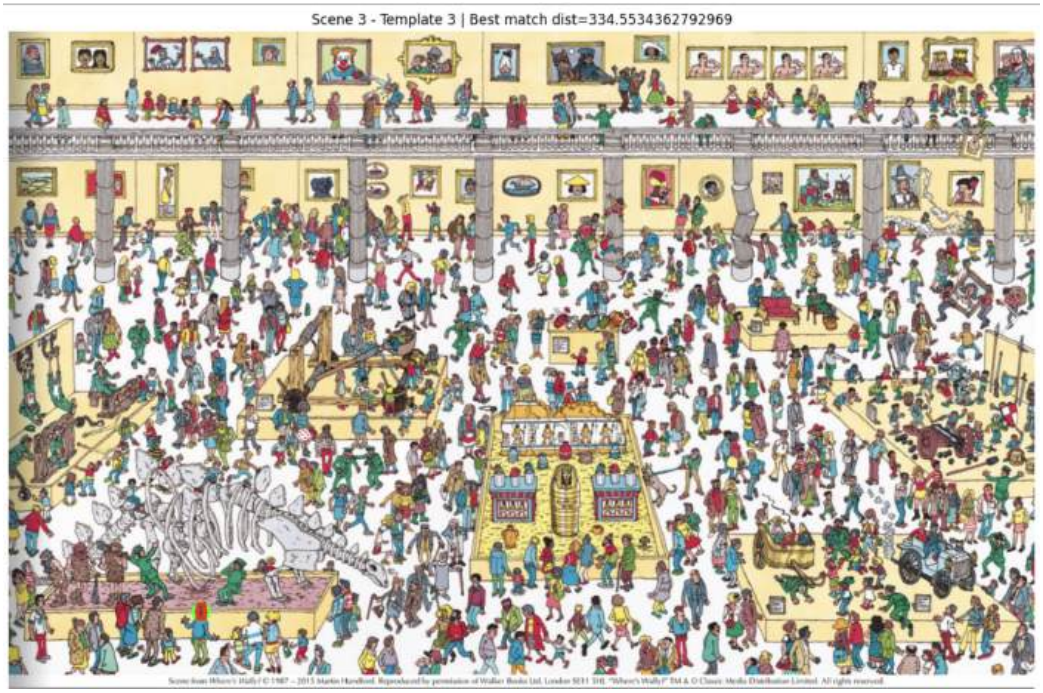



Figure 19: Feature-Based Based Matching Results for Wally

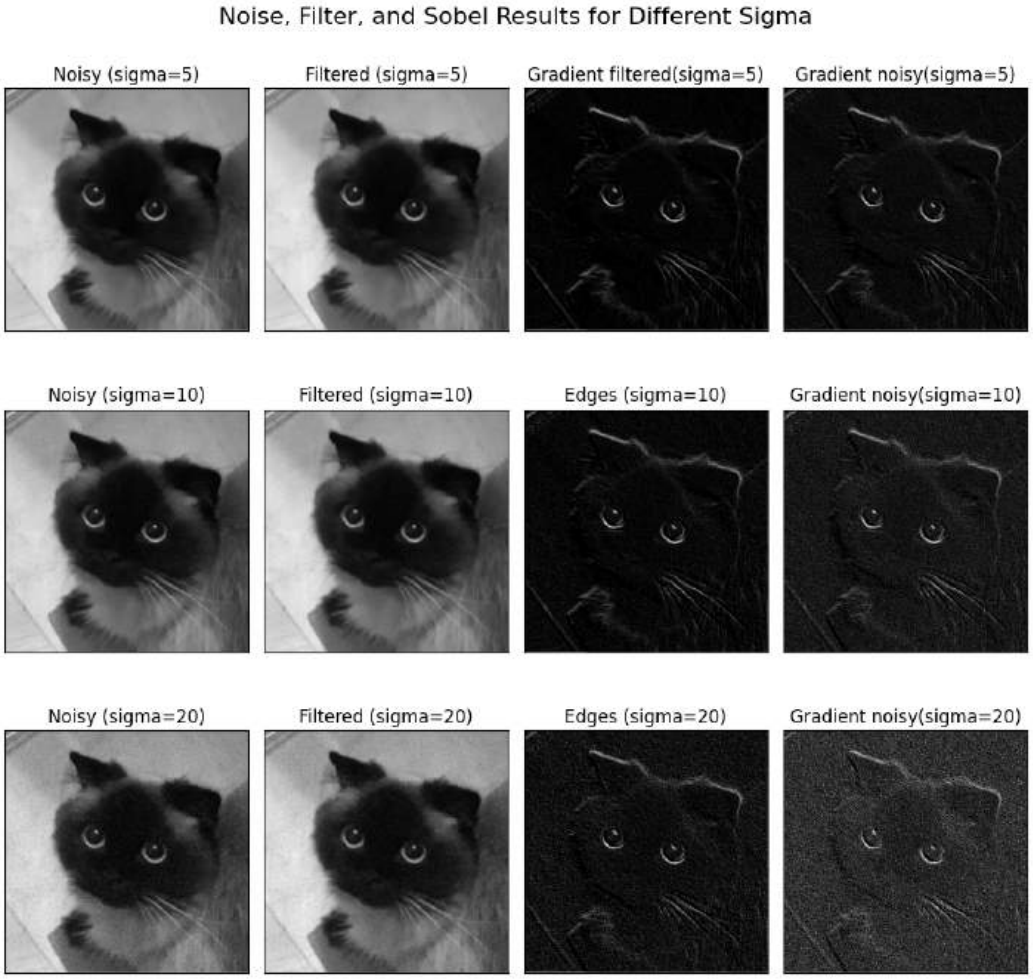



Figure 20: Output

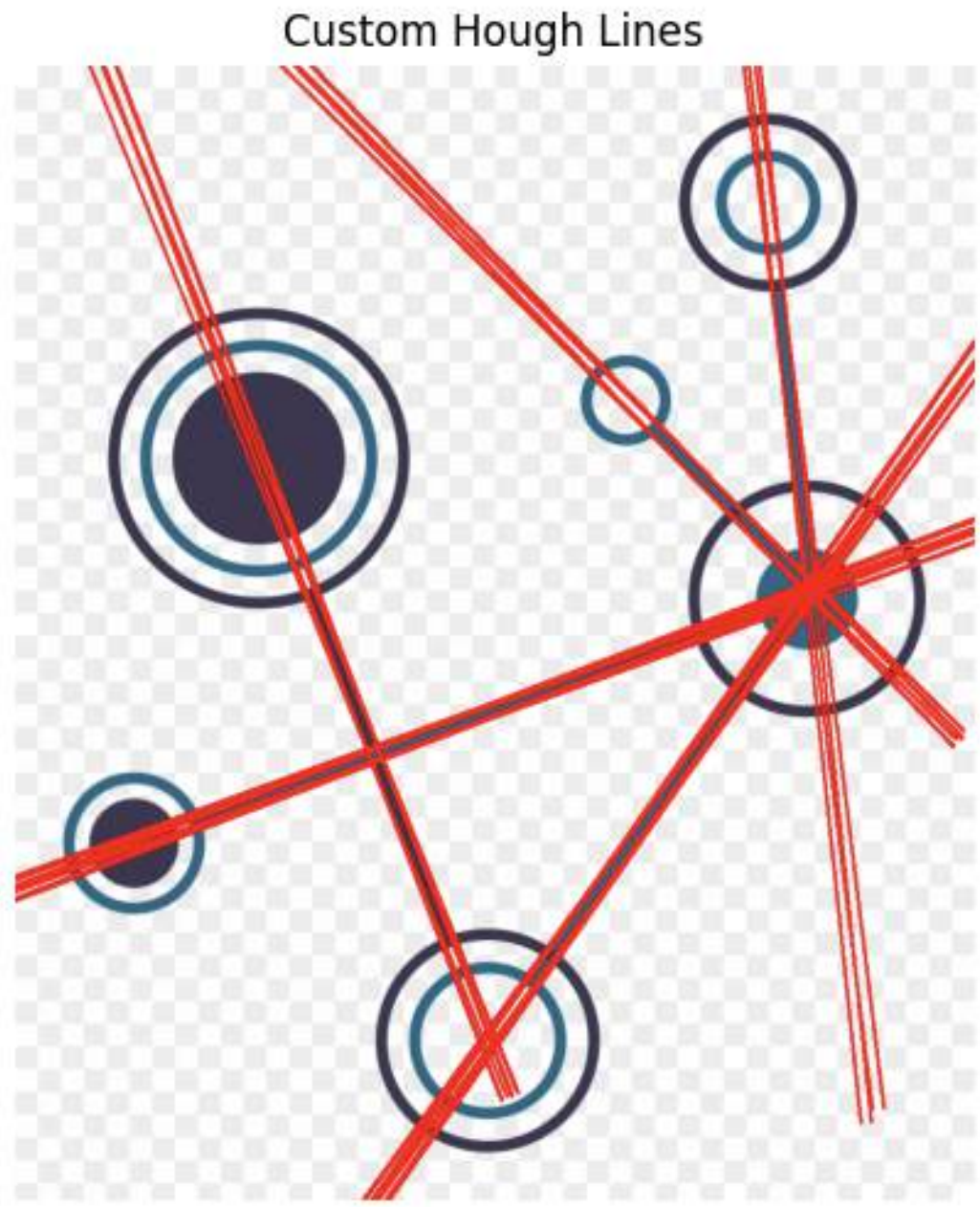



Figure 21: Hough Transform Results for Line Detection

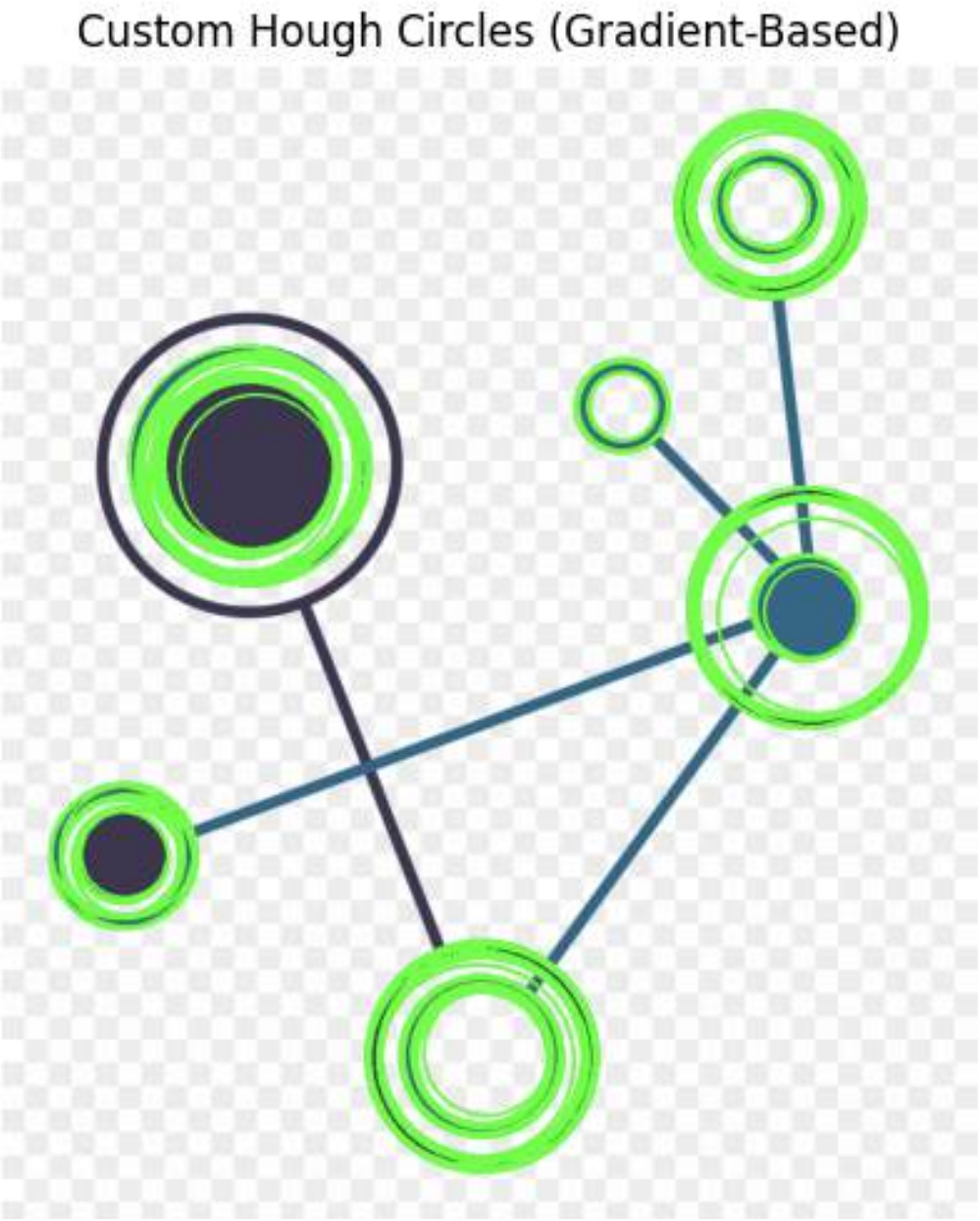



Figure 22: Hough Transform Results for Circle Detection