Summary of the Technical Aspects of the Object Detection Implementation

1. Design Rationale

The system focuses on preprocessing, bounding box detection and evaluation. The preprocessing step groups a set of predefined image transformations that isolate the areas of interest in the image, bounding box detection identifies the objects in the processed image, and the evaluation step measures performance compared to the ground truth. The design emphasizes modularity, with independent functions for each processing step.

2. Algorithm Explanation

Preprocessing Pipeline

- **Input:** Raw images and a reference background.
- **Steps:**
    1. **Block Region:** Black out irrelevant areas using predefined polygon coordinates.
    2. **CLAHE:** Enhance contrast for better feature visibility.
    3. **Gaussian Blur:** Reduce noise.
    4. **Background Subtraction:** Remove static background features.
    5. **Sharpening & Median Filtering:** Improve edge details and remove noise.
    6. **Thresholding & Binarization:** Segment image into binary for edge detection.
    7. **Canny Edge Detection:** Highlight object boundaries.
    8. **Morphological Closing:** Fill small gaps in detected edges.
- Output: Preprocessed binary image.

Bounding Box Detection

- **Sliding Window Approach:** Scans the preprocessed image using a sliding window.
- **Contour Filtering:** Identifies contours and filters them based on:
    - Aspect ratio.
    - Minimum area.
- **Non-Maximum Suppression (NMS):** Eliminates redundant overlapping bounding boxes based on Intersection over Union (IoU) threshold.

Evaluation Metrics

- **IoU-Based Matching:** Matches predicted boxes with ground truth boxes using an IoU threshold.
- **Object-Level Metrics:**
    - **Precision, Recall, F1-score, and Accuracy.**
    - **Visualization:** Real and predicted bounding boxes are overlaid on images.
- **Image-Level Metric:** Mean squared error between the count of predicted and actual bounding boxes.
    - **Visualization:** Plotted real count vs detected count

3. Pseudocode

1. Load images and annotations
2. Prepare background image
3. For every image:
    a. Preprocess it:
        i. Convert to grayscale and block irrelevant regions.
        ii. Apply CLAHE, Gaussian blur, and background subtraction.
        iii. Perform sharpening, median filtering, thresholding, and edge detection.
        iv. Apply morphological operations for refinement.
    b. Detect Bounding Boxes:
        i. Use a sliding window to identify contours.
        ii. Filter contours based on area and aspect ratio.
        iii. Apply Non-Maximum Suppression (NMS) to reduce redundancy.
    c. Plot image with real and detected bounding boxes:
    d. Match predicted bounding boxes with ground truth using IoU.
    e. Calculate precision, recall, F1-score, and accuracy.
4. Visualize results and calculate average metrics.

4. Quantitative Results

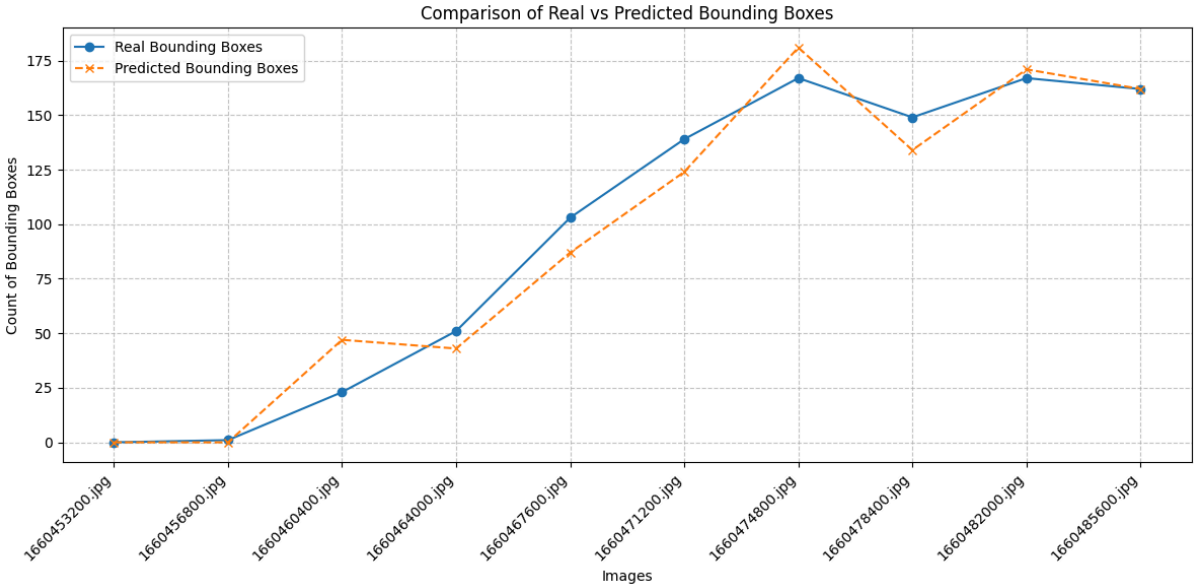
- **Performance Metrics (Average):**
    - Precision: 38.5%.
    - Recall: 41.2%.
    - F1-score: 39.2%.
    - Accuracy: 40.4%.
- **MSE of Bounding Box Counts:** 155.9
- **Comparison Graph:** Illustrates discrepancies between predicted and actual bounding box counts across all images.

Metrics Table:

	Precision	Recall	F1	Accuracy	MSE
1660453200.jpg:	0,00%	0,00%	0,00%	0,00%	0
1660456800.jpg:	0,00%	0,00%	0,00%	0,00%	1
1660460400.jpg:	36,17%	70,83%	47,89%	73,91%	576
1660464000.jpg:	34,88%	28,85%	31,58%	29,41%	64
1660467600.jpg:	56,32%	48,51%	52,13%	47,57%	256
1660471200.jpg:	51,61%	48,48%	50,00%	46,04%	225
1660474800.jpg:	57,46%	59,77%	58,59%	62,28%	196
1660478400.jpg:	46,27%	44,29%	45,26%	41,61%	225
1660482000.jpg:	52,05%	54,27%	53,13%	53,29%	16
1660485600.jpg:	50,62%	57,34%	53,77%	50,62%	0

As we can see we have decent metrics given the complexity of the problem but there is room for improvement.

Comparison graph:



As we can see there is a strong correlation between the count of real bounding boxes and predicted ones. Which shows that our method can estimate the number of people in the beach satisfactorily.

5. Sample Outputs

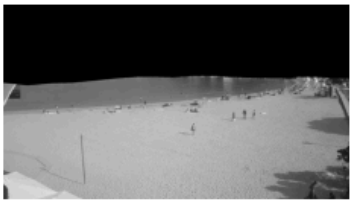
The following images will be shown for image **1660464000.jpg** and **1660474800.jpg**

- **Intermediate Steps:** Images processed through each preprocessing stage are displayed to visualize transformations (e.g., original, blocked-out, CLAHE-applied, etc.).
- **Bounding Box Visualizations:** Images with overlaid ground truth (green) and predicted boxes (red).

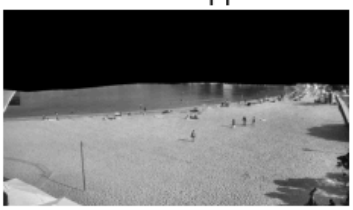
1. 1660464000

2. Grayscale

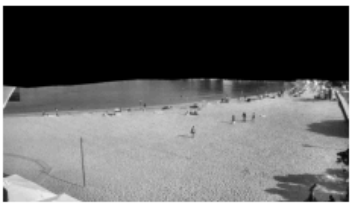
3. Blocked Out



4. CLAHE Applied



5. Gaussian Blurred



6. Background Subtracted

7. Sharpened

8. Median Filtered

9. Thresholded (70,255)

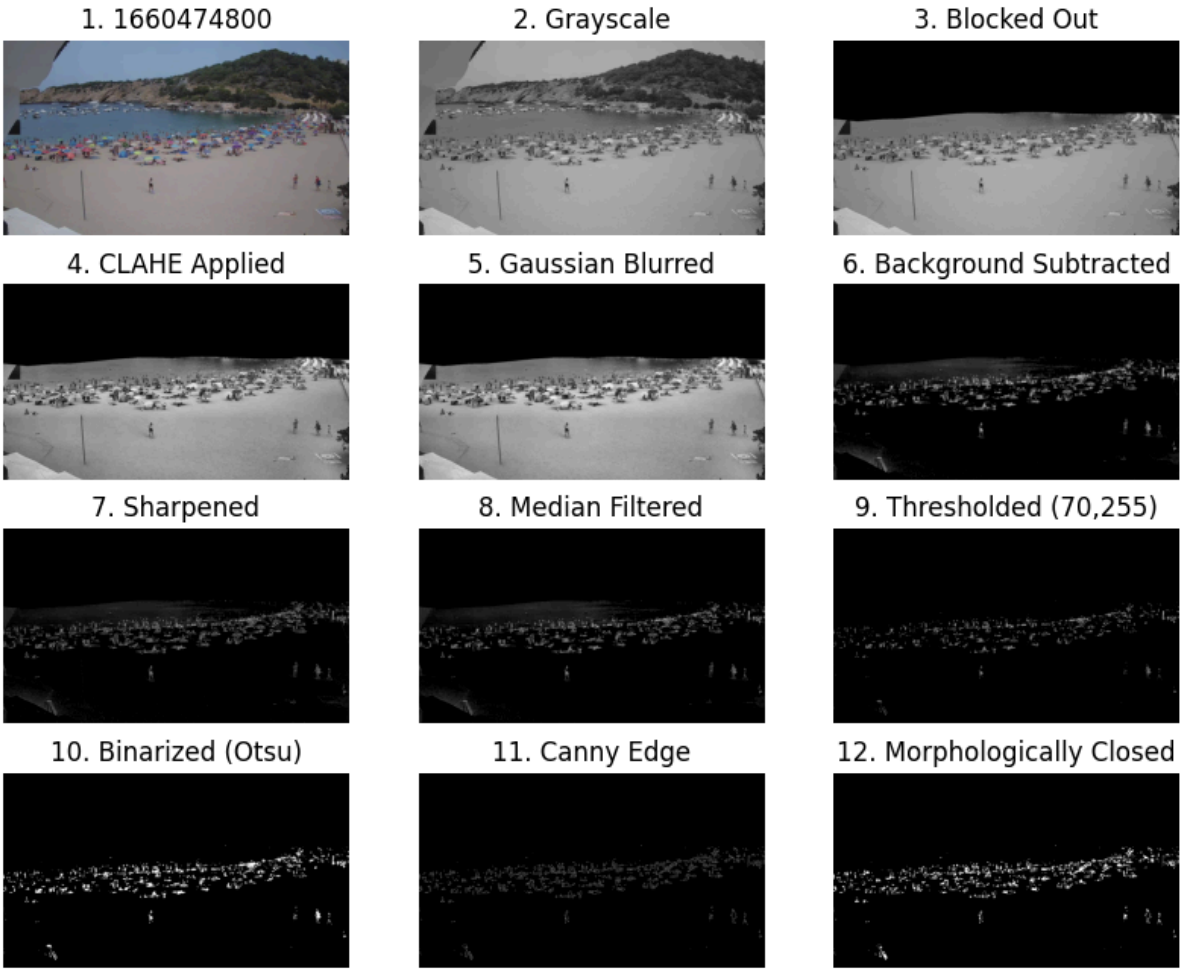
10. Binarized (Otsu)

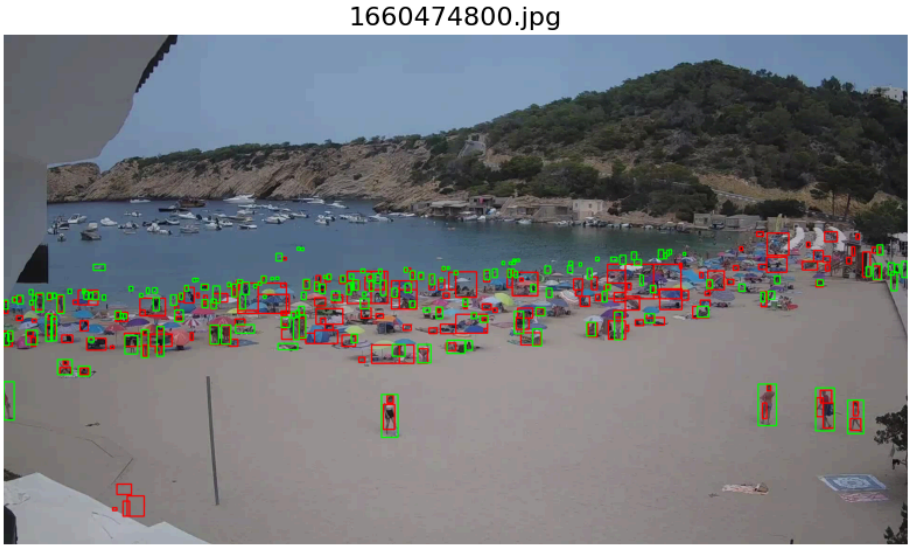
11. Canny Edge

12. Morphologically Closed

1660464000.jpg

true_positives: 15
false_positives: 28
false_negatives: 37
accuracy: 0.29
recall: 0.29
precision: 0.35
f1: 0.32
real_bboxes: 51
predicted_bboxes: 43
mse: 64





true_positives: 104
false_positives: 77
false_negatives: 70
accuracy: 0.62
recall: 0.60
precision: 0.57
f1: 0.59
real_bboxes: 167
predicted_bboxes: 181
mse: 196

As we can see in the results the processing steps get rid of the background until only the interesting objects are left using different techniques:

- Deleting non interesting part of the image (sky and mountain)
- Deleting background (Image subtraction with empty beach)
- Making edges shapes (Sharpening kernel)
- Deleting noise (Median filter)

- Deleting water reflection (Thresholding)

6. Strengths and Weaknesses

Strengths:

- **Modular Design:** Easy to modify and extend each stage of processing.
- **Comprehensive Metrics:** Evaluates detection comprehensively using object and image-level criteria.
- **Visualization:** Provides clear insights into performance with bounding box overlays and comparison graphs.

Weaknesses:

- **Hardcoded values:** Performance relies on hardcoded parameters found through trial and error
- **IoU Threshold Sensitivity:** Performance highly dependent on IoU threshold.
- **False Positives in Complex Backgrounds:** Subtle edges can lead to false bounding boxes.