11761 - Image Analysis and Video Processing
Intelligent Systems
Universitat de les Illes Balears

PROJECT 1

Antonio Nadal
Phornphawit Manasut

Lecturer:
Ayed, Ines

1st Semester

December 12, 2023

Contents

1 Introduction

The allure of the beach, with its vast expanse of sand and sea, presents a captivating yet challenging environment for computational image analysis. Beach scenes are dynamic, characterized by varying lighting conditions, diverse crowd densities, and ever-changing backgrounds that include moving water and shifting sands. These factors introduce significant complexities in accurately detecting and counting people, a task that is not only crucial for safety and management purposes but also for understanding human behavior in recreational spaces. This is especially true for hot-tourist destinations like Palma where the crowding of a beach can represent a safety issue for tourits.

This project delves into image analysis with focus on human detection and people counting in beach environments. We use 10 different images captured under varied lighting conditions and crowd scenarios, the objective is to explore and assess multiple methods of image preprocessing and human detection without relying on neural network approaches. Traditional image processing techniques, despite being overshadowed by the recent surge in neural network applications, still hold significant value due to their lower computational requirements and interpretability. By employing a range of non-neural network methods, this project aims to report the most effective combinations of traditional preprocessing and detection methods for human detections with relatively static background.

2 Code Pipeline

The steps to perform humans detection can be splitted into 3 big task: Preprocessing, Detection, and Evaluation. Hence, the code is structured as such and Object-Oriented Programming is employed for cleanliness of the code and quick combinations of different preprocessing or detection methods.

2.1 Preprocessing Classes

The preprocessing stage is crucial in preparing beach images for subsequent human detection. This stage employs a series of image preprocessing classes, each tailored to address specific challenges presented by beach environments:

1. BackgroundPreprocessor: This class is designed to mask out non-relevant areas of the image, focusing on regions of interest. It creates a mask based on predefined background points and applies it to the grayscale image to filter out unnecessary background details.

2. ImageAveragingPreprocessor: Utilizing a set of background images, this preprocessor reduces background noise. It achieves this by blurring the background images, computing their average, and subtracting this average image from the input image to enhance foreground features.

3. ThresholdingPreprocessor: Applies a binary threshold to the image. It creates an in-range mask based on the provided upper and lower threshold values and further refines the image using Otsu's thresholding method. This step is pivotal for distinguishing objects from the background.

4. WaterThresholdingPreprocessor: Specifically targets the water regions in beach images. It converts color images to HSV, applies a threshold to isolate the water region, and removes these regions from the grayscale image, effectively reducing the influence of water in the detection process.

5. SandThresholdingPreprocessor: Similar to water thresholding but focused on sand areas. It applies a threshold to the grayscale image to identify sand regions and removes them, thereby reducing the interference of sandy areas.

6. SobelPreprocessor: Employs the Sobel operator to enhance edge features in the image. It computes the gradient magnitude and applies a threshold to isolate prominent edges, crucial for detecting human figures against complex backgrounds.

7. CLAHEPreprocessor: Enhances the contrast of the image using Contrast Limited Adaptive Histogram Equalization (CLAHE), making it easier to distinguish human figures from the background.

8. OtsuPreprocessor: Another method of thresholding, Otsu's method automatically determines the threshold value for binary segmentation, helping to distinguish foreground objects from the background effectively.

9. PreprocessingPipeline: This class allows for the sequential application of multiple preprocessing steps. It iteratively applies each preprocessing method to the image, progressively refining it for better detection results.

10. MultipleANDBitProcessingPipeline: Combines the results of multiple preprocessing pipelines using a bitwise AND operation. This approach synergizes the strengths of individual pipelines, resulting in a more refined and focused image for detection tasks.

Each preprocessing class plays a specific role in enhancing image quality, reducing noise, and highlighting features pertinent to human detection in beach scenarios. The usage of pipeline allows us to easily experiment with different combination of preprocessing steps.

2.2 Detector Classes

The project employs a variety of detection methods, each utilizing different techniques to identify human figures in beach images. These methods are encapsulated in several classes, inheriting from a generic detector class:

1. SimpleBlobDetector: This detector uses the Simple Blob Detection algorithm provided by OpenCV. It is configured to identify blob-like structures in the image, which can correspond to human figures in certain scenarios. The detector identifies keypoints in the image and returns their coordinates as potential human locations.

2. HOGSvmDetector: The Histogram of Oriented Gradients (HOG) combined with a Support Vector Machine (SVM) is used for human detection. This method is particularly effective in identifying human shapes due to its ability to capture edge and gradient structure characteristic of human forms. The detector returns the center points of the detected bounding boxes, which are assumed to be the locations of humans. This class result is omitted from our final result because the pretrain model result is extremely bad.

3. HaarCascadeDetector: Utilizes Haar Cascade classifiers, which are effective for detecting human faces. This method is based on the rapid extraction of features from the image and the use of trained cascade function classifiers. The detector provides the top-left corner coordinates of each detected human figure. This class is not utilized in our experiments.

4. ContourDetector: This detector applies contour detection to identify human figures. By analyzing the contours in the image, the detector focuses on shapes that resemble human figures. The method calculates the center of mass (centroid) of each contour, using these points as potential human locations.

5. DBSCANHumanDetector: Implements the DBSCAN clustering algorithm to detect clusters of points that could represent humans. This method is particularly useful in crowded beach scenes where individual detection might be challenging. The detector first extracts features (points of interest) from the image and then applies DBSCAN to these features. The centroids of the resulting clusters are assumed to be the locations of humans or groups of humans.

Each of these detectors has unique strengths and is suitable for different scenarios encountered in beach environments. For instance, the SimpleBlobDetector under the right parameters have high recall but low precision while the DBSCANHumanDetector has high precision but relatively low recall.

2.3 Evaluation Class

The evaluation of human detection methods in this project is performed using the NearestPointEuclideanEvaluation class, a specific implementation of an abstract Evaluation framework. Here's how the evaluation process works:

1. Distance-Based Matching: For each detected point (a potential human location identified by the detection methods), the evaluation algorithm finds the nearest ground truth point, which represents the actual human location in the image.

2. Threshold Comparison: It then compares the Euclidean distance between each detected point and its nearest ground truth point to a predefined threshold. This threshold determines how close a detection must be to a ground truth point to be considered correct.

3. Classification of Detections: Based on this comparison, detections are classified as:
   (a) True Positives (TP): Detected points that are within the threshold distance of a ground truth point.
   (b) False Positives (FP): Detected points that are further than the threshold distance from any ground truth point.
   (c) False Negatives (FN): Ground truth points that do not have any detected point within the threshold distance.
   (d) True Negatives (TN): This is 0 for object detection task.

4. Quantitative Metrics: The evaluation then calculates standard performance metrics:

(a) Precision: The ratio of true positives to the total number of detected points ($\frac{TP}{TP+FP}$).

(b) Recall: The ratio of true positives to the total number of ground truth points ($\frac{TP}{TP+FN}$).

(c) F1 Score: The harmonic mean of precision and recall, providing a balance between them. ($2 \times \frac{Precision \times Recall}{Precision \times Recall}$)

(d) Mean-Squared Error: On the task of counting number of people on the beach. Squared Error is calculated for differences of the count per image and their average is given afterwards for the total.

5. Visualization: Additionally, the evaluation process includes a function to visualize the results as a confusion matrix for each image and cumulatively for all images. This matrix visually represents the distribution of true positives, false positives, true negatives, and false negatives.

This evaluation method provides a comprehensive assessment of the detection accuracy, balancing the need to correctly identify human figures while minimizing false detections and missed detections.

3 Methodology

3.1 Preprocessing Pipelines

Each of these pipelines represents a strategic approach to preprocessing beach images, tailored to overcome specific challenges such as varying backgrounds, water and sand interference, and changing lighting conditions. By experimenting with these diverse pipelines, the project aims to identify the most effective preprocessing strategies for enhancing human detection in beach environments.

3.1.1 Basic Background Subtraction

This pipeline employs a basic background subtraction method. It starts with a custom background preprocessing to mask out certain areas of the image, followed by image averaging and subtraction using specific background images (for example: '10.png' and '9.png'). The pipeline concludes with a thresholding process to further enhance object detection. This pipeline is particularly effective in scenarios where the background remains relatively static.

In our experiments, we notice that image averaging subtraction is not very useful with 2 background images, we therefore create 2 separate pipeline: one that subtract using ('10.png') and another that subtract using ('9.png'). ANDBit operation is performed on the 2 resulting images and allow us to much better handle different lighting conditions.

3.1.2 Water and Sand Thresholding and Background Subtraction

This pipeline is designed to specifically handle the challenges posed by water and sand in beach scenes. It combines water and sand thresholding preprocessors with background subtraction. By isolating and removing water and sand regions, this pipeline enhances the visibility of human figures against the complex textures of the beach.

3.1.3 CLAHE and Background Subtraction

Incorporating Contrast Limited Adaptive Histogram Equalization (CLAHE), this pipeline aims to improve the contrast of the beach images. CLAHE is combined with background subtraction to produce images with enhanced details, which is crucial for accurate human detection in varying lighting conditions.

3.1.4 Dual Background Subtraction Pipelines

Two separate pipelines were created using different background images (10.jpg and 9.jpg) for the background subtraction process. Both pipelines include background subtraction and thresholding steps. These dual pipelines allow for a comparison of detection effectiveness based on different reference background images, addressing the issue of changing beach scenes across different times or conditions.

3.1.5 Multiple ANDBit Processing Pipelines

Here, we can combine all pipelines into a single pipeline. Resulting masks from each image is combined with ANDBit operation.

3.2 Detectors

3.2.1 Blob Detector

The Blob Detector operates on the principle of identifying bright on dark or dark on bright regions in an image that differ in properties like brightness or color compared to the surrounding. It uses the Simple Blob Detection algorithm provided by OpenCV, which fundamentally works by applying a series of thresholds and finding center points of 'blobs' in the binary images. These blobs are assumed to be potential human figures based on their size, shape, and other definable characteristics such as area, circularity, and convexity. The detector is particularly effective in scenarios where humans appear as distinct 'blob'-like structures against the background.
The parameters used for our experiment are minimum area of 2 and maximum area of 100. Filter by circularity of 0.75. Filter by convexity of 0.75. Filter by convexity of 0.15. Please refer to 'https://learnopencv.com/blob-detection-using-opencv-python-c/' for explanation of these parameters.

3.2.2 Contour Momentum Detector

The Contour Momentum Detector leverages the concept of contour analysis and image moments. It begins by applying edge detection to identify the outlines or contours of objects within an image. Contours are continuous lines or curves that bound or cover the full boundary of objects in the image. After detecting these contours, the detector calculates their centroids (center of mass) using image moments. Image moments are weighted averages

of pixel intensities or a function of such moments, which provide a measure of the intensity distribution of an object. In this context, the centroids of the contours are considered as potential locations of human figures, especially when the contour shapes approximate the human form.
The threshold value used in our experiment is 50.

3.2.3 DBScan Detector

The DBScan (Density-Based Spatial Clustering) Detector is a clustering-based approach. It differs from the previous two detectors by focusing on the spatial distribution of points or features in the image rather than their individual characteristics. This method involves extracting features from the image, such as points of interest, and then using the DBScan algorithm to cluster these points based on their spatial density. DBScan works by identifying high-density areas and expanding clusters from these points. In the context of human detection, each cluster's centroid is calculated and considered as a potential location of a human or a group, making it particularly useful in crowded scenes where individual detection might be challenging.
The minimum number of samples used in this experiment is 40 and the distance of 2 points should be within 4.

3.3 Evaluation Method

This has been explained in previous section and will not be explained again. However, it should be noted that our experiment threshold for euclidean distance is 100.

3.4 Experiments Process

Initially, we use Blob detector as the baseline to test out best pipelines to use. Afterwards we found out that the multiple bit all pipelines combination is the best and as such move on to testing various detectors. The result is detailed in the next section.

4 Results

In quantitative result subsection, a table reporting the results of each pipeline and detection is given while in qualitative result subsection. We will go through different bad and good results of each pipeline to provide ideas on what went wrong and to support our findings in quantitative subsection.

4.1 Quantitative Results

The results from the various preprocessing pipelines, as summarized in the Table 1, present a view of the performance of different human detection methods in beach scenes.

In the case of the Blob Detector, its application across different pipelines shows varied results. The basic pipelines, particularly Basic Pipeline 1, achieved a high recall (0.896 for Basic Pipeline 1 and 0.895 for Basic Pipeline 2), indicating a strong ability to detect most human figures in the images. However, the precision was significantly lower (0.431 and 0.208, respectively), suggesting a higher rate of false positives. The Mean Squared Error (MSE) was notably high for Basic Pipeline 2 (445366.7) due to too many false positives.

The Water and Sand Pipeline and the CLAHE Pipeline, when used with the Blob Detector, showed improvements in recall (0.972 and 0.958, respectively) but still suffered from low precision (0.205 and 0.260, respectively). This pattern suggests that while these pipelines are effective in detecting human figures, they also tend to produce a considerable number of false positives. The F1-Scores for these pipelines were consequently lower (0.339 and 0.410, respectively), and the MSE was relatively high, especially for the Water and Sand Pipeline (113420.6).

When utilizing the Dual Basic Pipeline and Multiple And Bit Pipelines with the Blob Detector, there was a notable improvement in precision (0.671 and 0.696, respectively), while maintaining a good recall (0.843 for both). These pipelines achieved the highest F1-Scores among the Blob Detector setups (0.747 and 0.762, respectively), indicating a more balanced trade-off between precision and recall. The MSE values were significantly lower for these pipelines (992.7 and 719.6, respectively), suggesting a closer alignment with the ground truth.

The Multiple And Bit Pipelines, when paired with the Contour and DBScan Detectors, showed a further enhancement in performance. The Contour Detector exhibited the highest precision (0.82774) but a lower recall (0.67537), leading to a solid F1-Score of 0.743. The DBScan Detector achieved a balanced performance with a precision of 0.734, a recall of 0.672, and an F1-Score of 0.702. Both these detectors had lower MSE values (427.8 for Contour and 279.9 for DBScan), indicating a more accurate count of human figures compared to the Blob Detector.

Overall, the results demonstrate that the choice of preprocessing pipeline and detector significantly impacts the detection accuracy. While Blob Detectors are effective in identifying most human figures (high recall), they tend to generate many false positives (low precision). The Multiple And Bit Pipelines, particularly when combined with Contour and DBScan Detectors, provided a more balanced performance, achieving higher precision without substantially compromising recall.

Preprocessing Pipeline	**Detector**	**Precision**	**Recall**	**F1-Score**	**MSE**
Basic Pipeline 1	Blob	0.431	0.896	0.582	16179.3
Basic Pipeline 2	Blob	0.208	0.895	0.337	445366.7
Water and Sand Pipeline 1	Blob	0.205	0.972	0.339	113420.6
CLAHE Pipeline 1	Blob	0.260	0.958	0.410	88323.5
Dual Basic Pipeline 1	Blob	0.671	0.843	0.747	992.7
Multiple And Bit Pipelines	Blob	0.696	0.843	0.762	719.6
Multiple And Bit Pipelines	Contour	0.82774	0.67537	0.743	427.8
Multiple And Bit Pipelines	DBScan	0.734	0.672	0.702	279.9

Table 1: Performance of different detection method when paired with different preprocessing methods.

4.2 Qualitative Results

Here, we shall look at a few detection to justify the quantitative analysis.

Figure 1 and Figure 2 demonstrate the reasoning for why Dual Basic Pipeline is needed. The background 1 is not very good at certain shadow removal and background 2 is not very good at some areas. Hence, they are combined and result in Figure 3.

The application of the CLAHE Pipeline introduced enhanced contrast and clarity to the masks, particularly beneficial under varying lighting conditions. This pipeline improved the visibility of human figures against the beach background, as depicted in Figure 4. While the precision was modest, the improved contrast led to more distinct and clear detections.

The comparison of detections across different detectors, as shown in Figures 5, 6, and 7, reveals the distinct characteristics of each detection method. The Blob Detector, while having a high recall, showed a tendency to include more false positives. The Contour and DBScan Detectors demonstrated a more balanced approach with improved precision, as evidenced by the comparative visualizations. The mask generated by the combined pipeline can also be seen in these Figures.

4.3 Figures

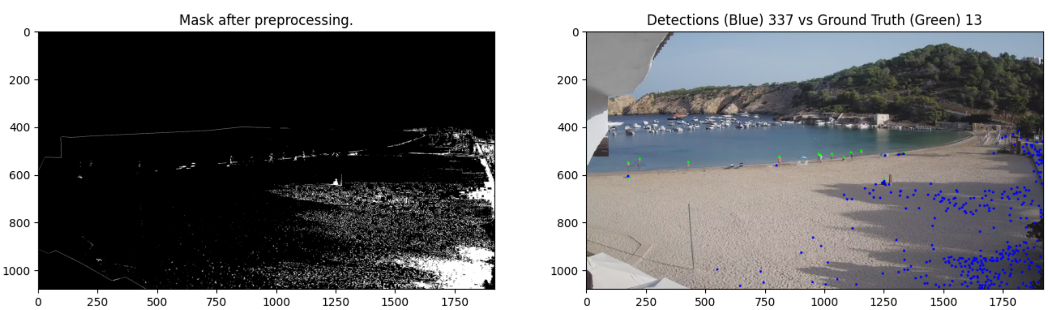

Figure 1: Mask generated by Basic Pipeline 1 and Detections with Blob Detector

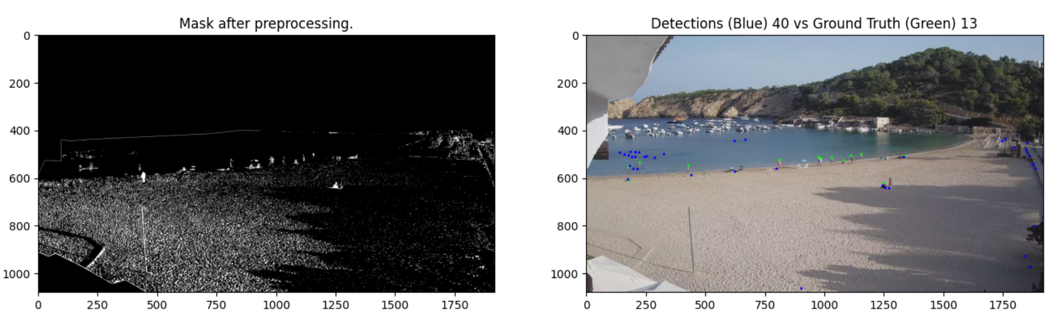

Figure 2: Mask generated by Basic Pipeline 2 and Detections with Blob Detector

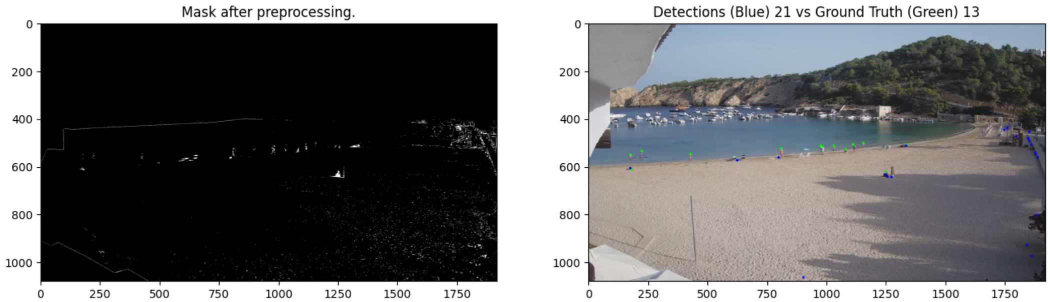

Figure 3: Mask generated by Dual Basic Pipeline 1 and Detections with Blob Detector

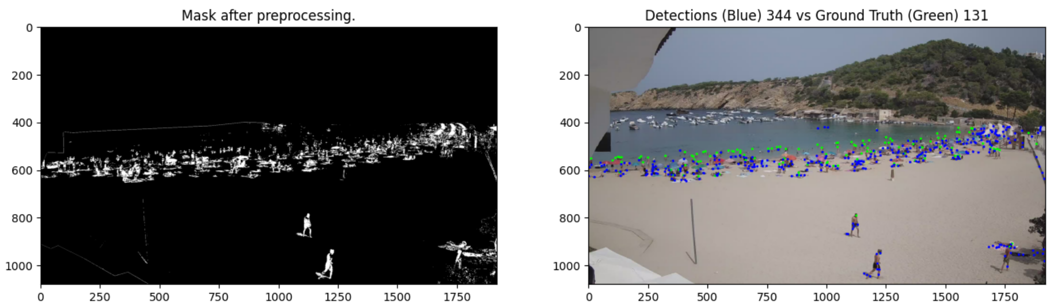

Figure 4: Mask generated by CLAHE Pipeline and Detections with Blob Detector

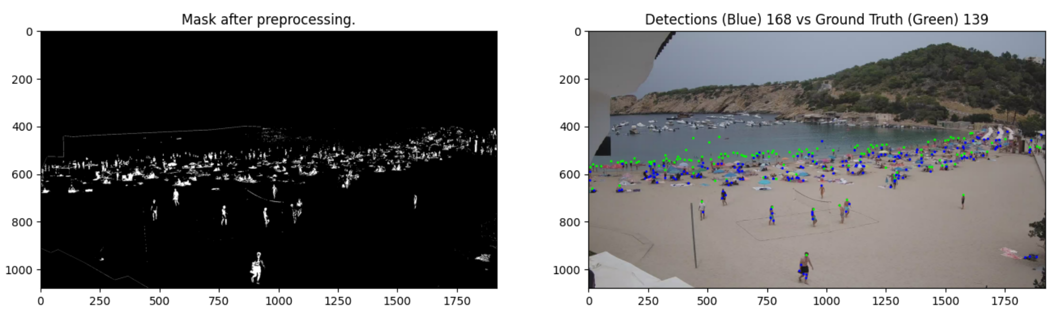

Figure 5: Detection comparison using Blob Detector

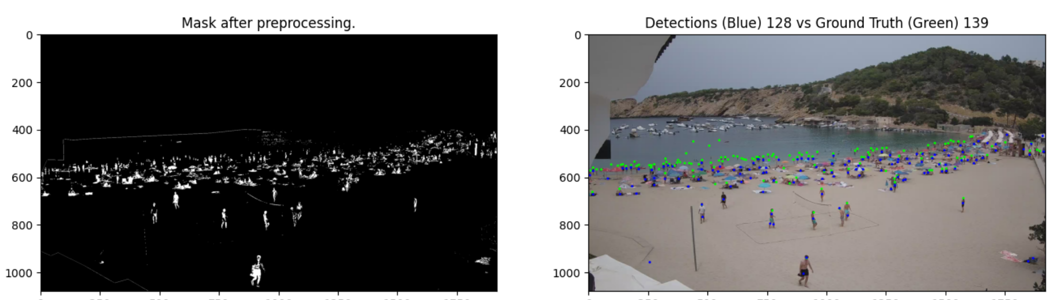

Figure 6: Detection comparison using Contour Detector

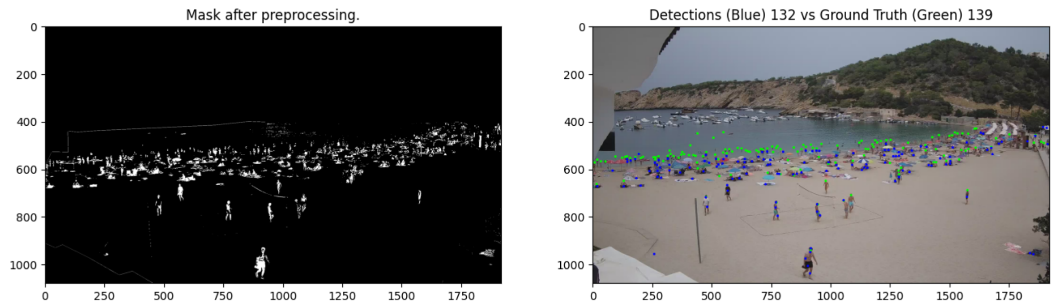

Figure 7: Detection comparison using DBScan Detector

5 Conclusion

In this project, we explored the efficacy of various preprocessing pipelines and detection methods for human detection in beach scenes. The quantitative results demonstrated that different preprocessing strategies, such as basic background subtraction, CLAHE enhancement, and dual pipeline approaches, significantly influence the detection accuracy. Particularly, the Dual Basic Pipeline and Multiple And Bit Pipelines showed notable improvements in precision and F1-Scores when used with Blob, Contour, and DBScan detectors, highlighting the importance of tailored preprocessing in complex environments like beaches. These findings provide valuable insights into optimizing human detection processes without relying on neural network methodologies, underlining the potential of traditional image processing techniques in diverse real-world applications.

Further improvement can be made to the pipelines by finding the correct HSV values for water and sand filtering. In addition, the current combined pipeline background subtraction and AND Bit also subtract same image background and is the cause of no-detection in those images. Hence, it would be wise to create a smart pipeline that detect same image and not use that background. Moreover, multiple detector parameters per image detection could be utilize in order to detect each region separately because the size of humans can changed.

6 References

1. For Blob detection parameters: https://learnopencv.com/blob-detection-using-opencv-python-c/

2. For DBScan: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html