Crowd counting using OpenCV and Python

Martí Gelabert Gómez

University of the Balearic Islands

November 29, 2022

Objective

We need to generate and algorithm capable of **crowd counting**...

Objective

We need to generate and algorithm capable of **crowd counting**...

So we need to :

- Have some kind of ground truth

Objective

We need to generate and algorithm capable of **crowd counting**...

So we need to :

- Have some kind of ground truth
- An algorithm to Detect persons

Objective

We need to generate and algorithm capable of **crowd counting**...

So we need to :

- Have some kind of ground truth
- An algorithm to Detect persons
- And be able to quantify our performance

Labeling

First, we need to **label** our images and establish some kind of **criteria** to decide which persons to label.

Occlusions

Compact information

Masked areas

Person detection

Now, we need a way for **detecting people**:

Person detection

Now, we need a way for **detecting people**:

- Gabor filtering ✗

Person detection

Now, we need a way for **detecting people**:

- Gabor filtering ✗
- Edge detector ✗

Person detection

Now, we need a way for **detecting people**:

- Gabor filtering ✗
- Edge detector ✗
- Applying derivatives ✗

Person detection

Now, we need a way for **detecting people**:

- Gabor filtering ✗
- Edge detector ✗
- Applying derivatives ✗
- Background removal ✓

Background Removal

Background Removal

Remark

The operations will be done using **grey scale** images.

We need to do something with the **illumination** of the images and improve the **contrast**.

CLAHE

Background Image

Average image ✗

Gaussian Blur

Substraction

Binarization

Binarization

cv2.threshold(substracted, 100, 255, cv2.THRESH_BINARY)

OTSU ✗

Masking

Dilation

Dilation

Find contours

Output

Results

Evaluation

- Only detections containing labels will be considered **TP**.
- If detection contains more than one person it will count as only one detection.
- Massive or tiny regions will be discarded.
- Some labels of the ground truth could be double checked.

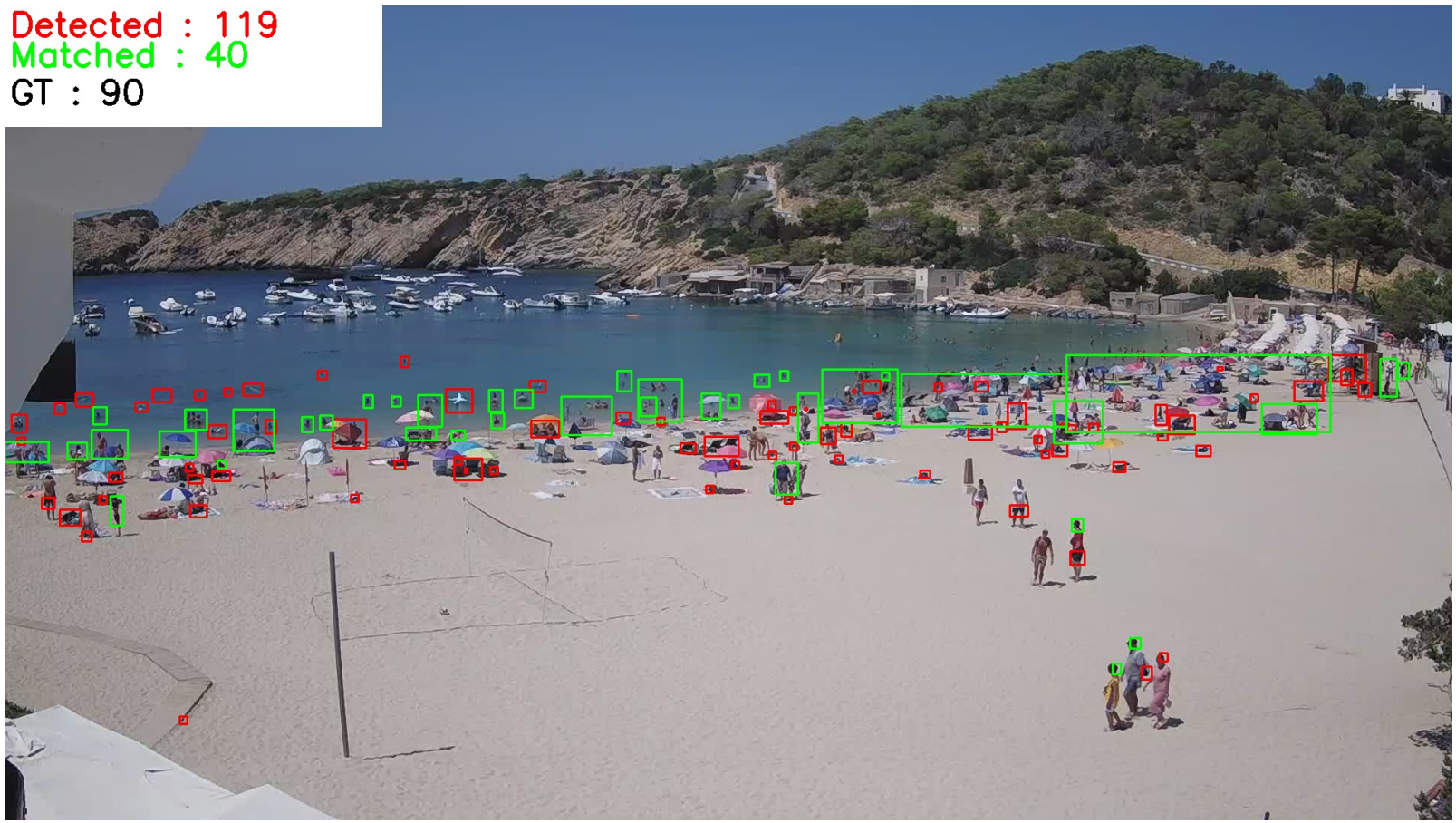
Results

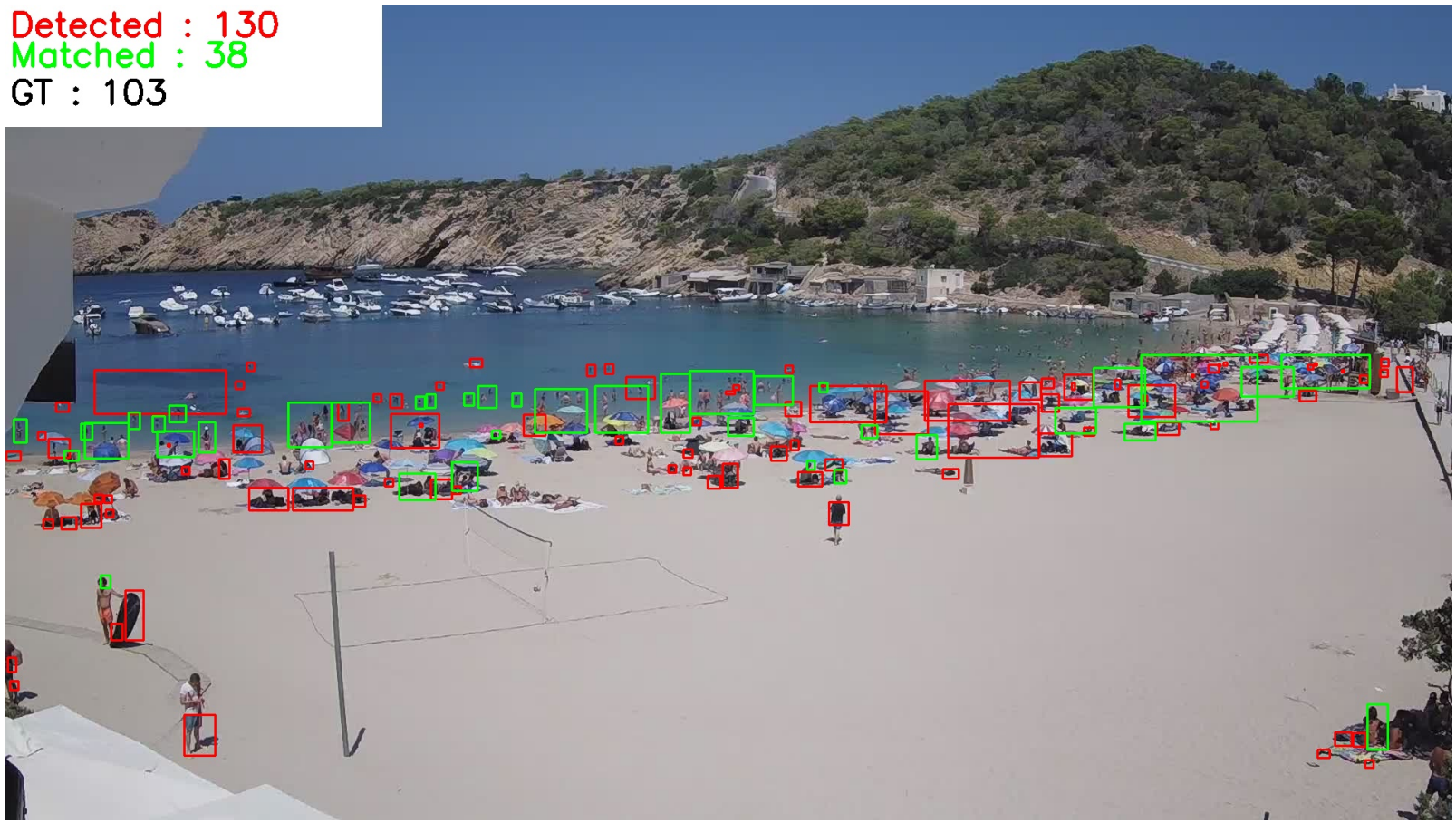
files	precission	recall	f1 score	gt	detected	matched
1660309200.jpg	0.336	0.444	0.383	90	119	40
1660302000.jpg	0.292	0.369	0.326	103	130	38
1660294800.jpg	0.333	0.458	0.386	72	99	33
1660320000.jpg	0.363	0.363	0.363	135	135	49
1660287600.jpg	0.200	0.471	0.281	17	40	8
1660298400.jpg	0.347	0.311	0.328	106	95	33
1660305600.jpg	0.360	0.396	0.377	101	111	40
1660316400.jpg	0.358	0.345	0.352	139	134	48
1660291200.jpg	0.353	0.346	0.350	52	51	18

Results

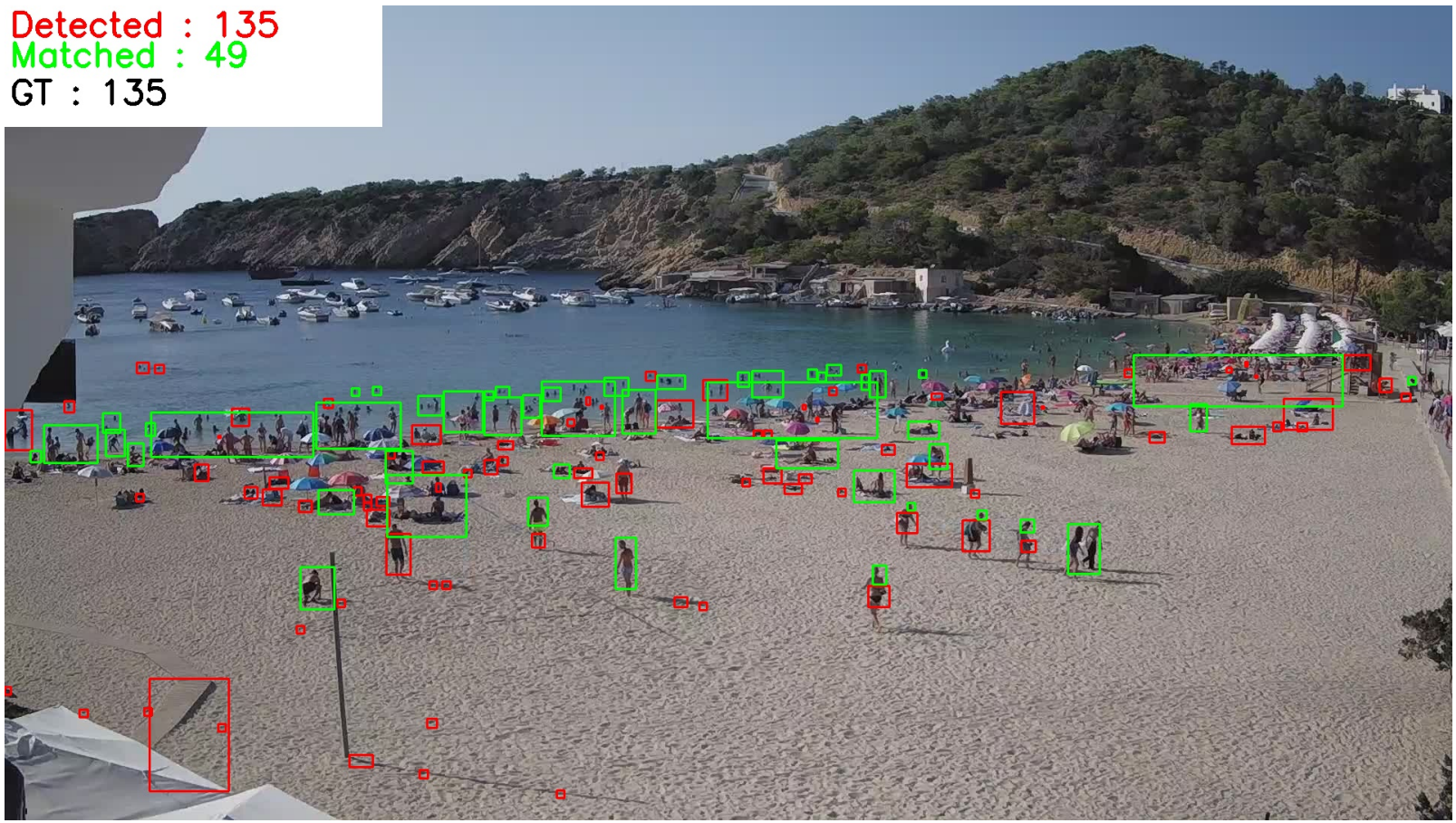
MSE	341.667
Macro-average precision	0.327
Macro-average recall	0.389
Macro-average F1	0.349

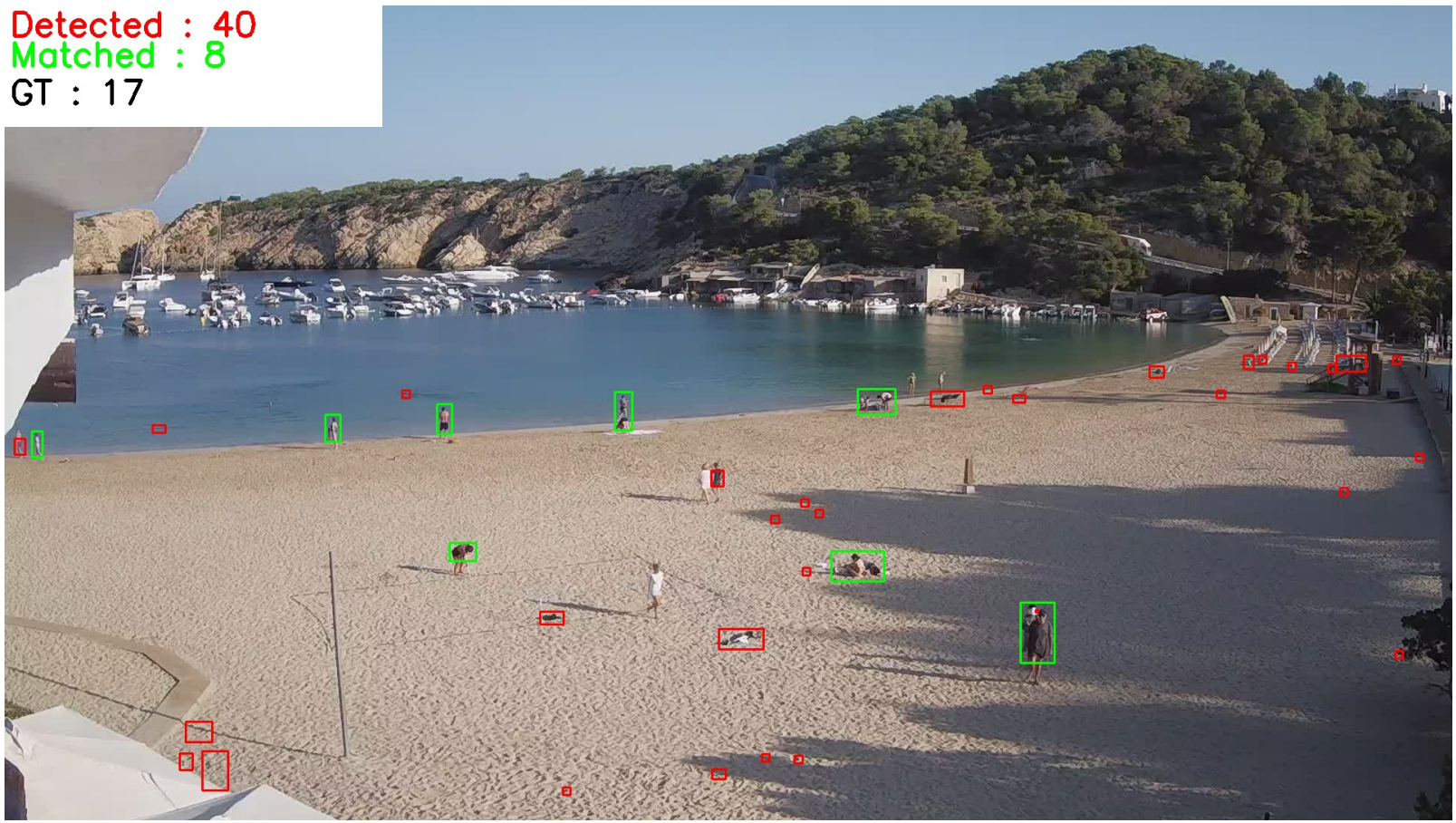
Outputs


Detected : 119
Matched : 40
GT : 90


Detected : 130
Matched : 38
GT : 103

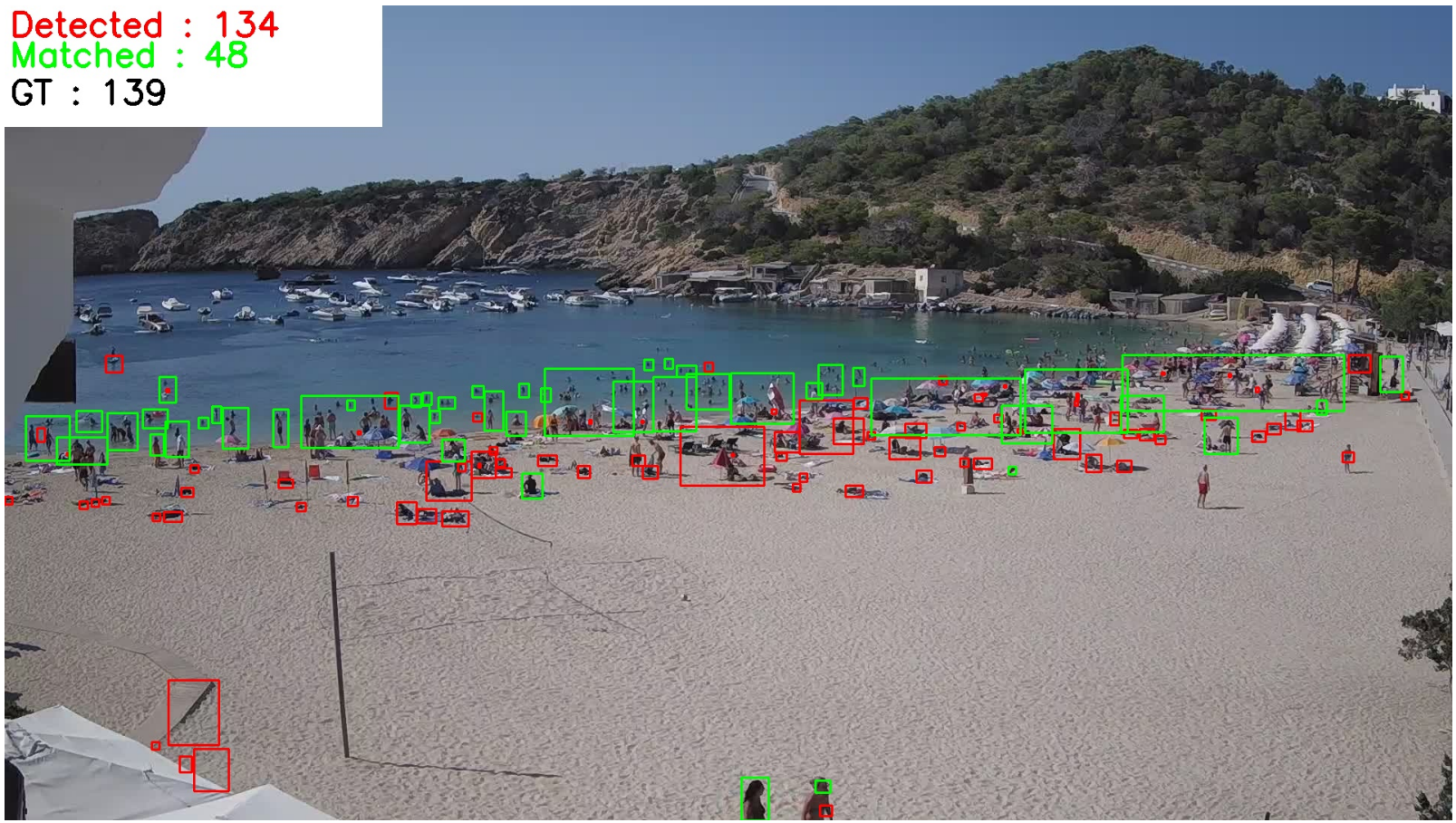

Detected : 99
Matched : 33
GT : 72

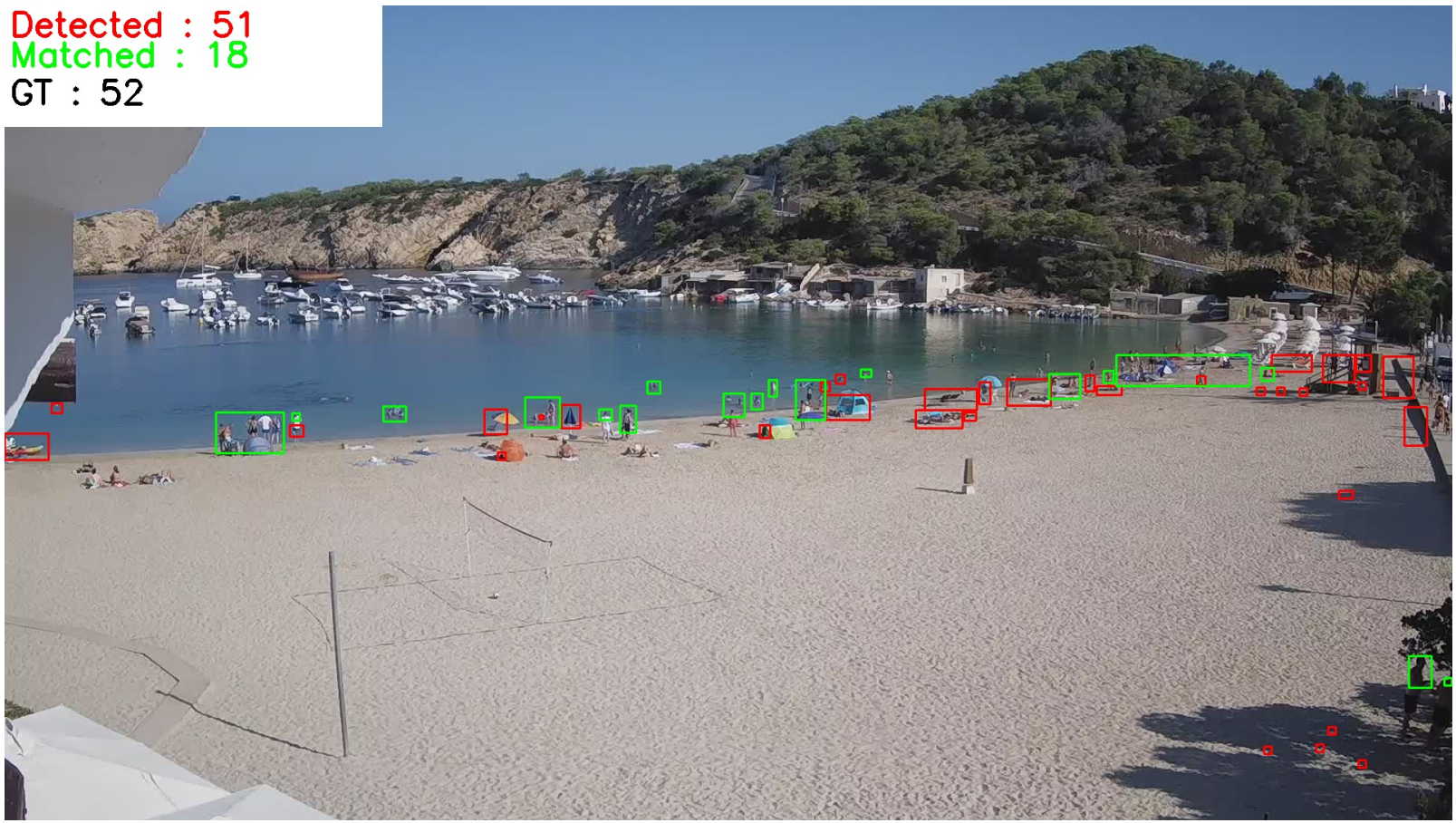

Detected : 135
Matched : 49
GT : 135


Detected : 40
Matched : 8
GT : 17


Detected : 95
Matched : 33
GT : 106


Detected : 111
Matched : 40
GT : 101


Detected : 134
Matched : 48
GT : 139


Detected : 51
Matched : 18
GT : 52

Conclusions

Conclusions

- Not seeking perfect detection.
- Brusque changes confuse our algorithm (i.e. cast shadows).
- Working in color could be not ideal.
- The results are really fragile.

Thank you!

Any Questions?