Università degli Studi di Modena e Reggio Emilia

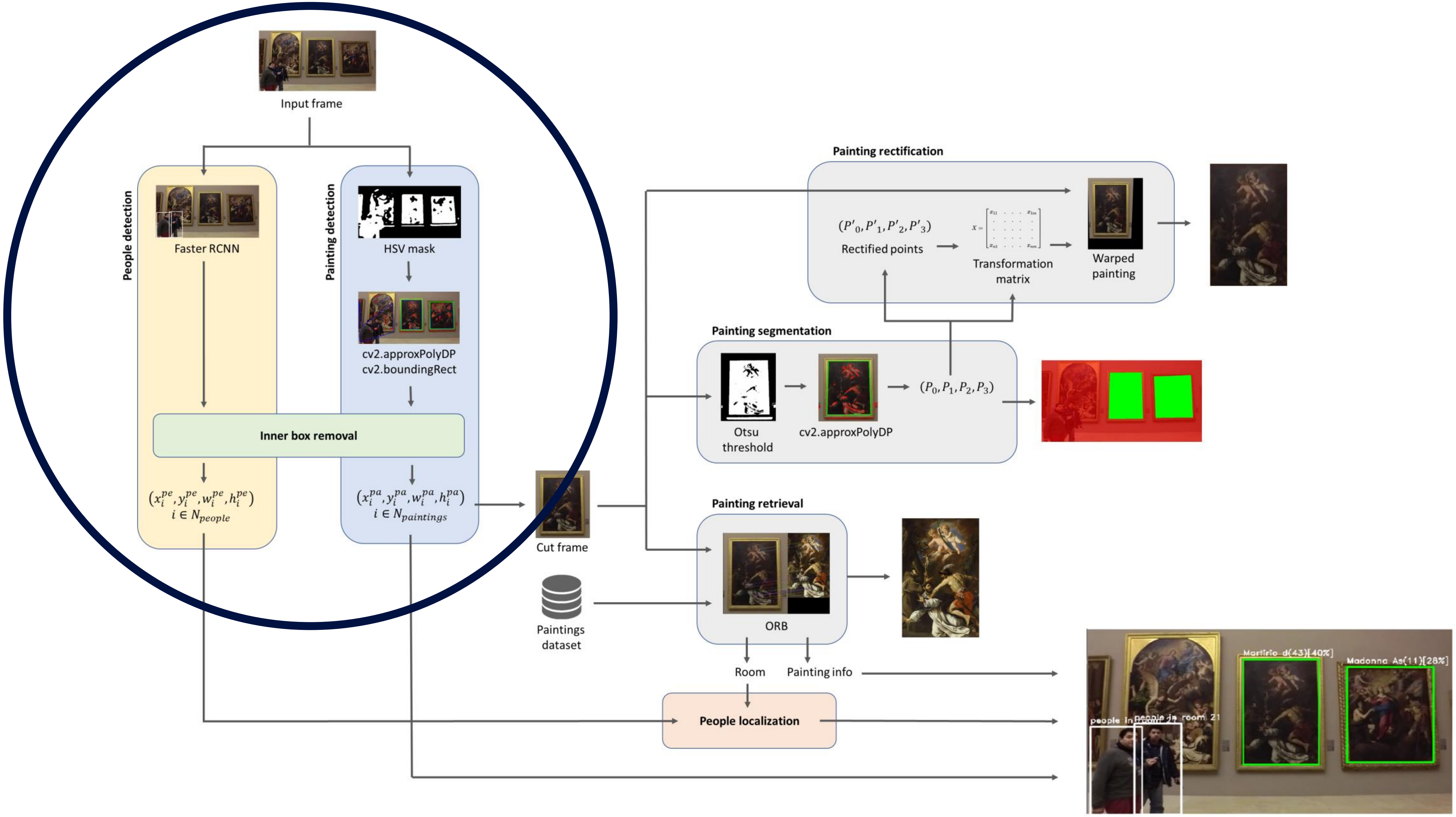
Laurea Magistrale in Ingegneria Informatica
a.a. 2019-2020

Vision and Cognitive Systems

Final project

Group 28 – Bertellini, Caputo Imbriaco, Doganieri

The project pipeline



Input frame

People detection

Faster RCNN

Painting detection

HSV mask

cv2.approxPolyDP
cv2.boundingRect

Inner box removal

$(x_i^{pe}, y_i^{pe}, w_i^{pe}, h_i^{pe})$
$i \in N_{people}$

$(x_i^{pa}, y_i^{pa}, w_i^{pa}, h_i^{pa})$
$i \in N_{paintings}$

Cut frame

Paintings dataset

Painting rectification

(P'_0, P'_1, P'_2, P'_3)
Rectified points

$x = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}$
Transformation matrix

Warped painting

Painting segmentation

Otsu threshold

cv2.approxPolyDP

(P_0, P_1, P_2, P_3)

Painting retrieval

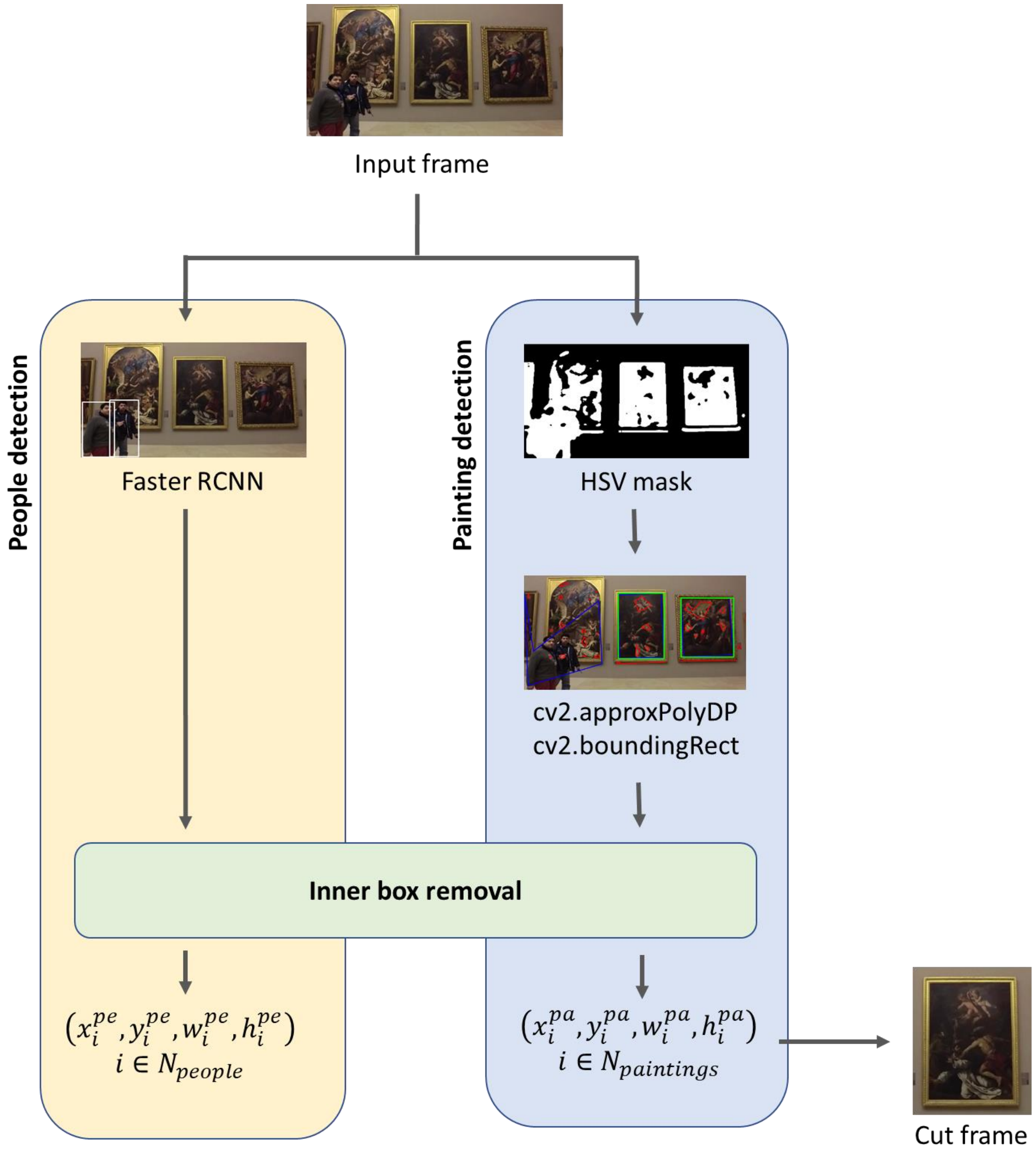
ORB

Room

Painting info

People localization

The detection pipeline



Input frame

People detection

Faster RCNN

Painting detection

HSV mask

cv2.approxPolyDP
cv2.boundingRect

Inner box removal

$(x_i^{pe}, y_i^{pe}, w_i^{pe}, h_i^{pe})$
$i \in N_{people}$

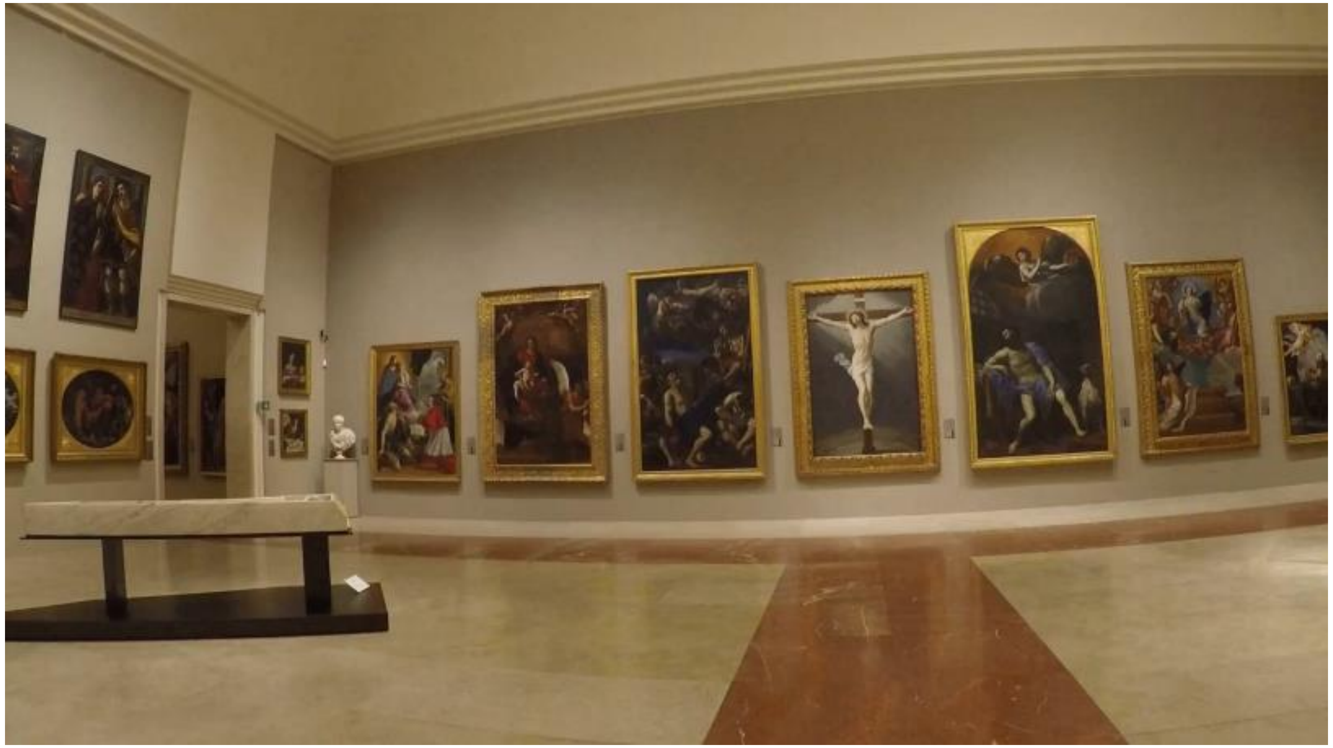
$(x_i^{pa}, y_i^{pa}, w_i^{pa}, h_i^{pa})$
$i \in N_{paintings}$

Cut frame

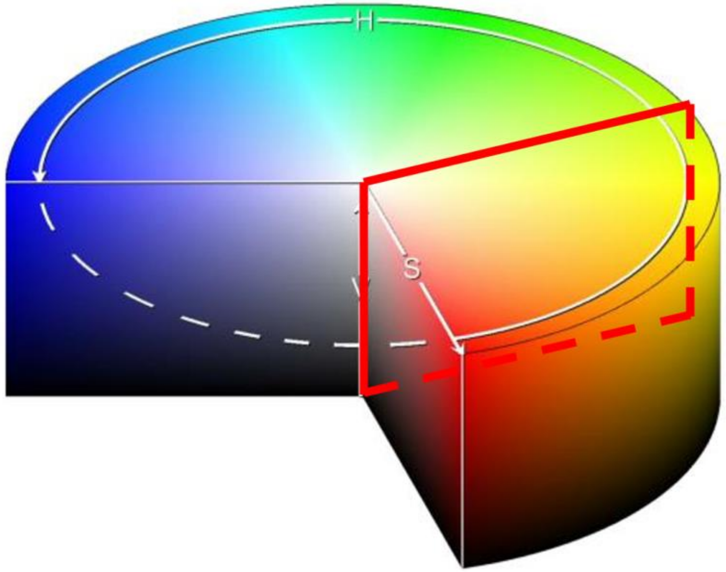
Painting detection

The painting detection is based on three main assumptions:

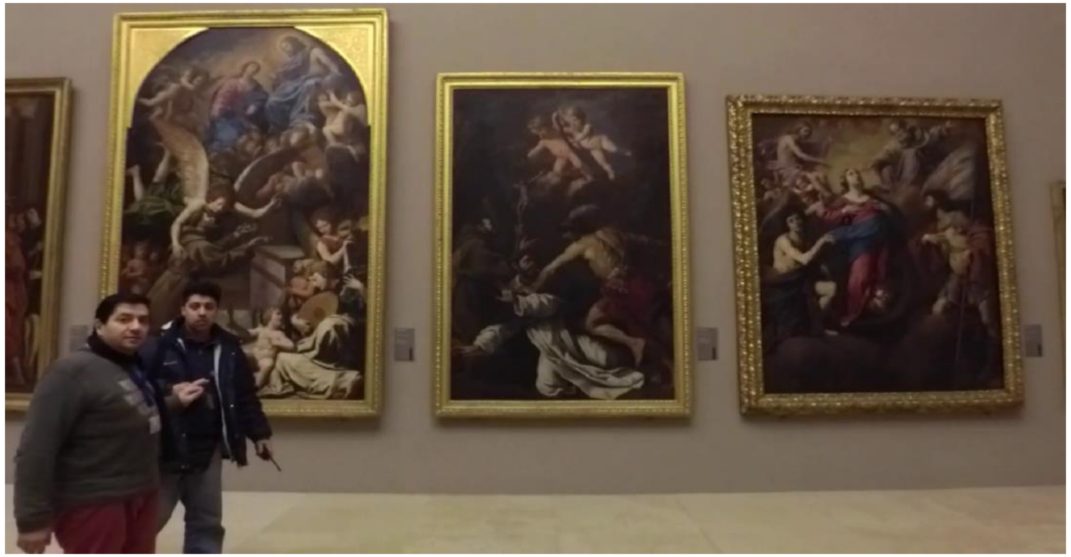
- **The wall has a lighter color** with respect to the paintings
- The wall color on average has a **hue between 0 and 80**
- Almost all the **paintings are rectangular** and most of the circular ones have a rectangular frame

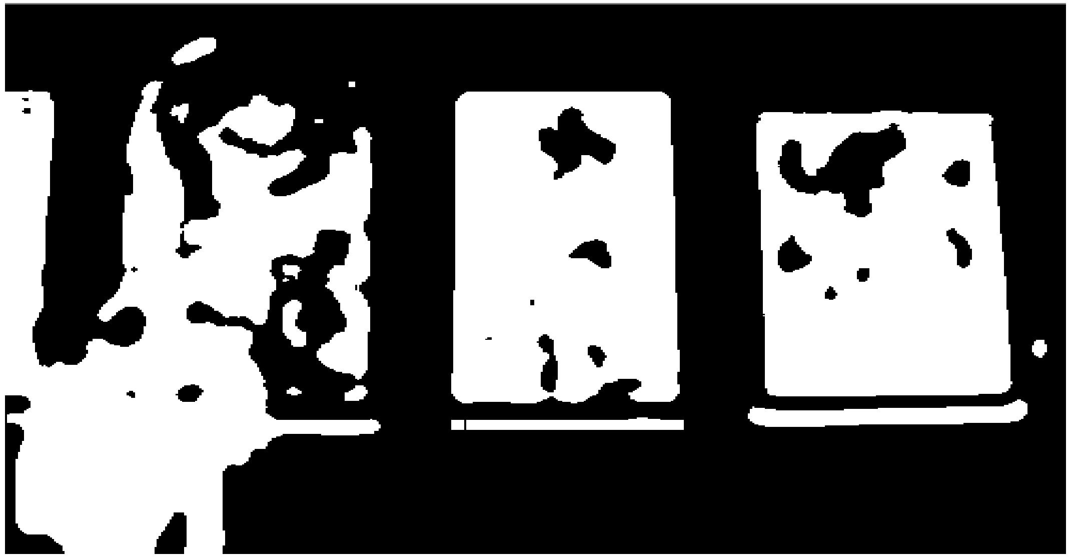


HSV Color Space

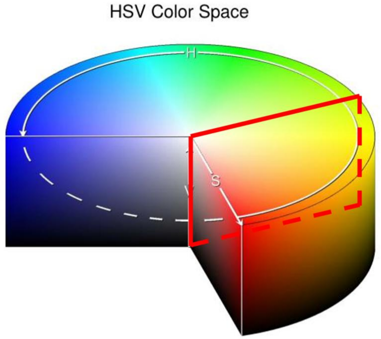


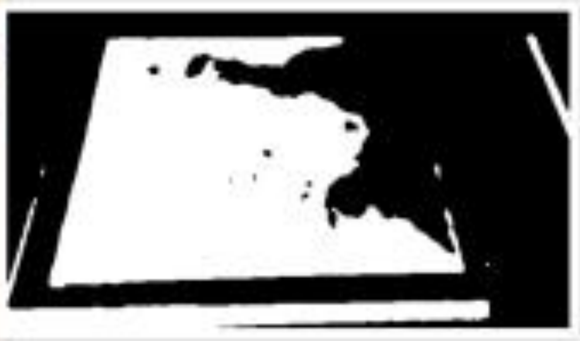
HSV Mask



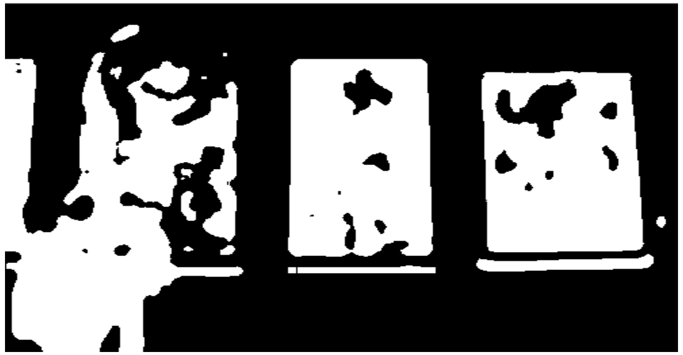


Using cv2.inRange



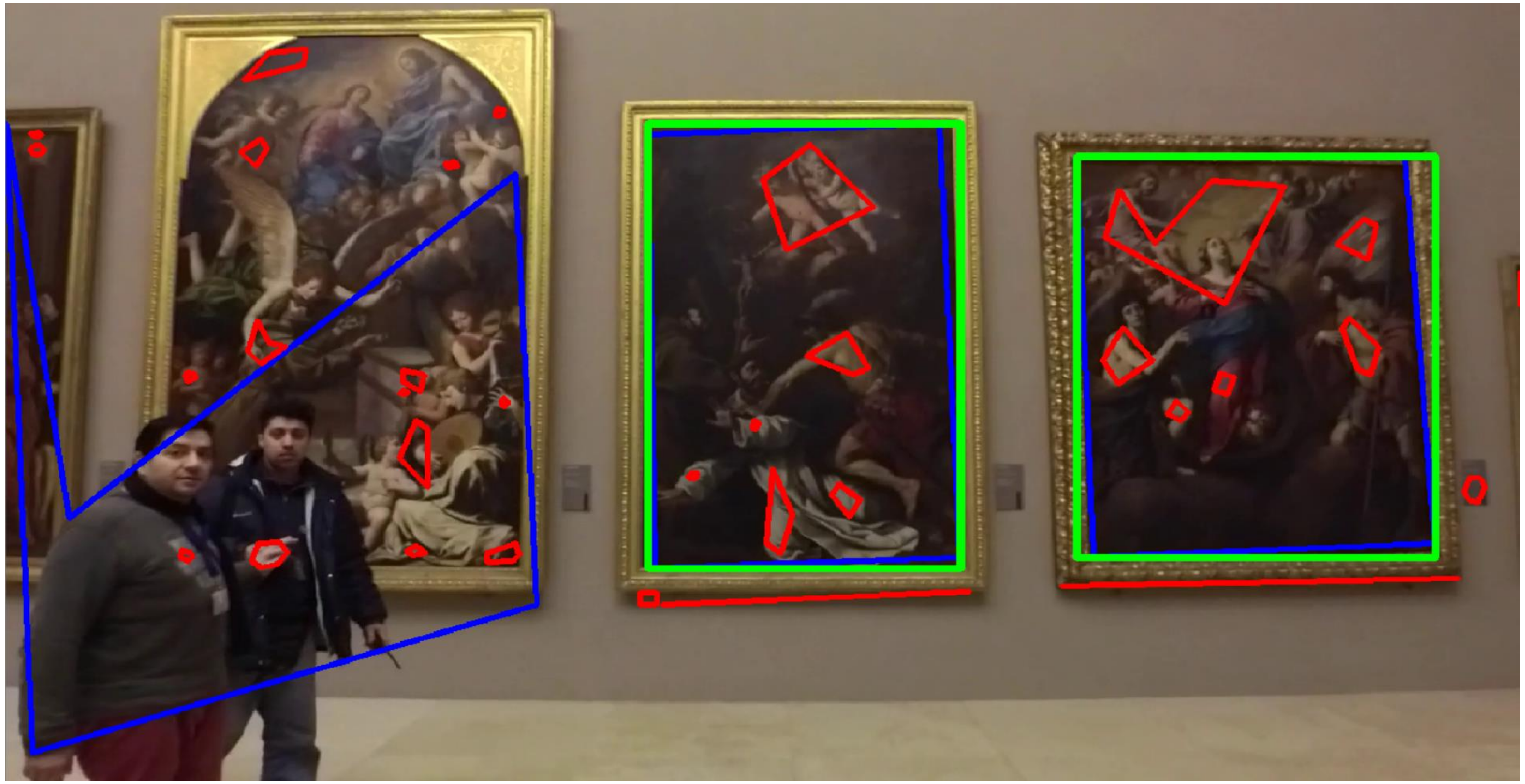


Find contours



Using cv2.findContours

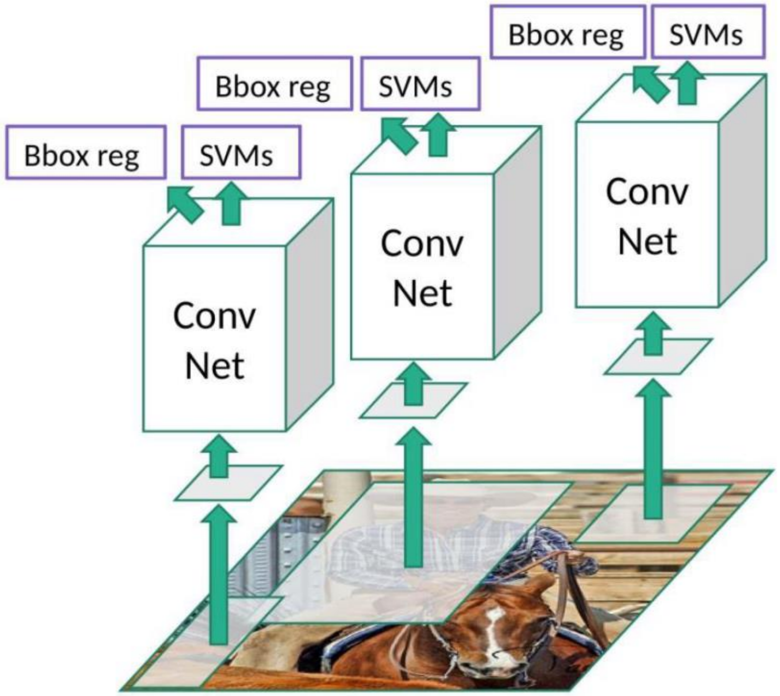
+ cv2.approxPolyDP



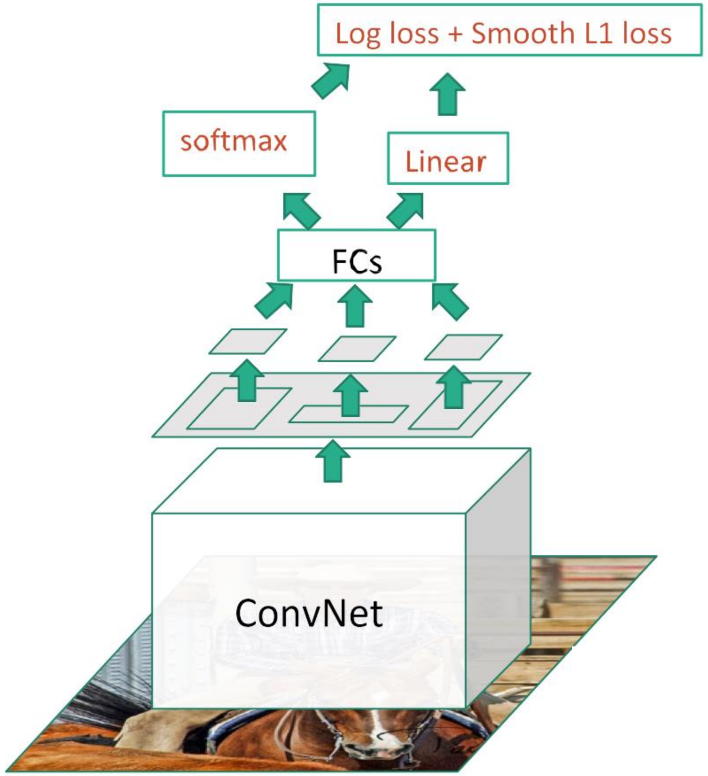
Filtering:

- Areas greater than 15000 pixels
- N. sides equal to 4

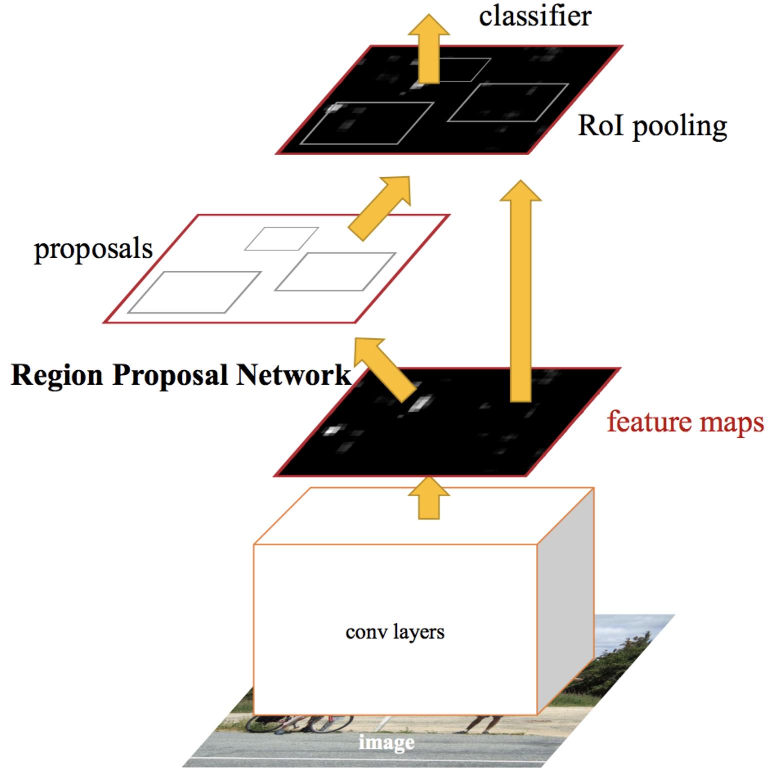
Object detection architectures



R CNN

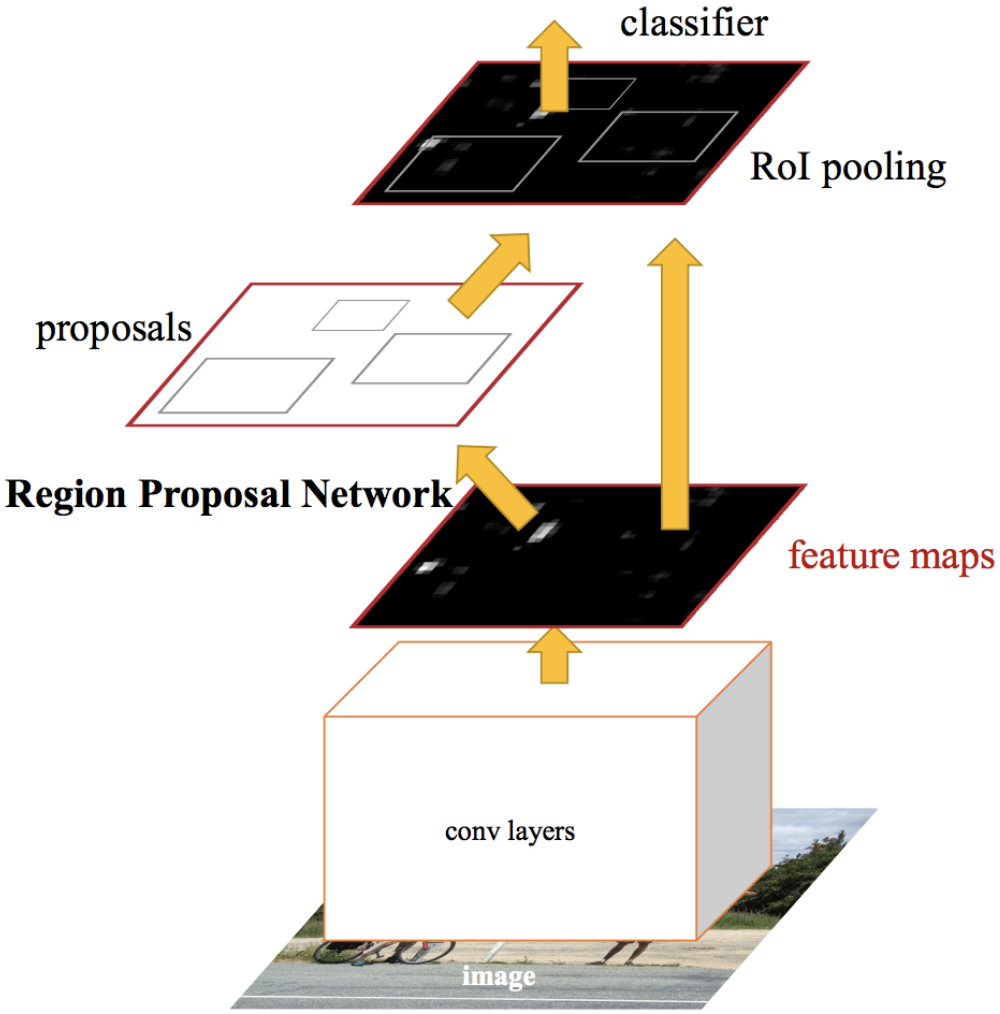


Fast R CNN

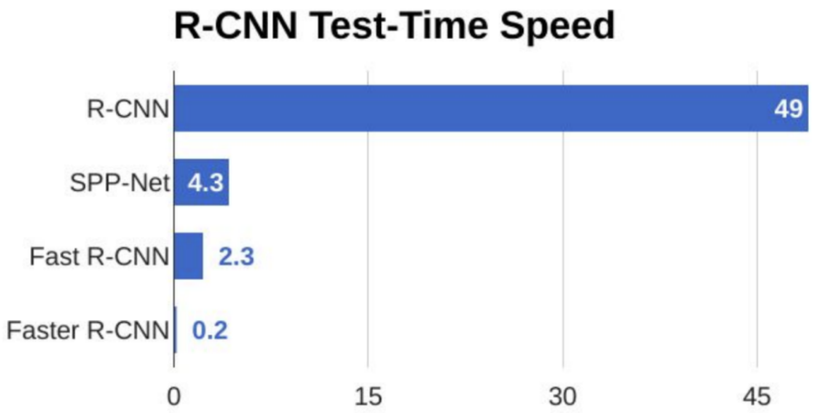


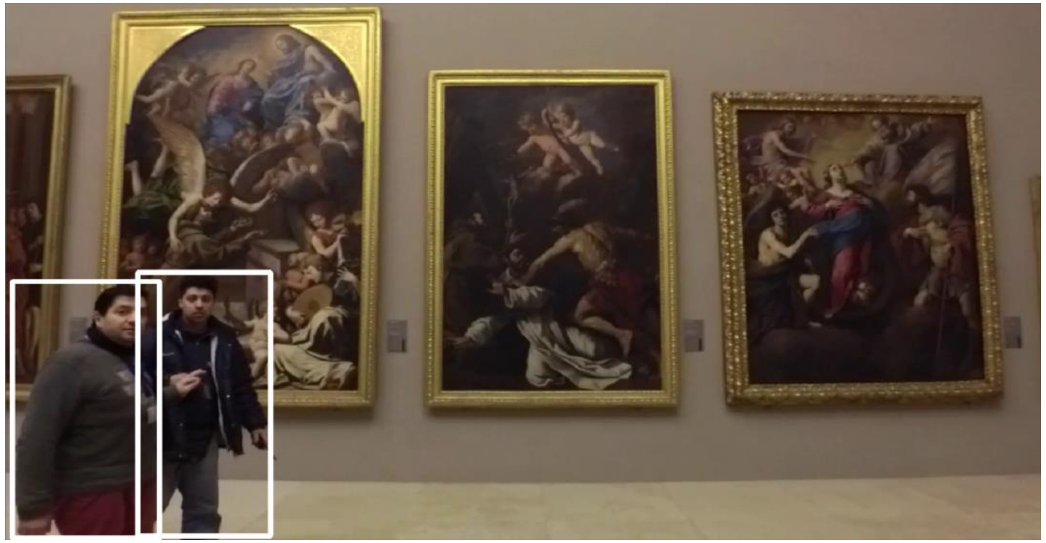
Faster R CNN

People detection



Faster RCNN Pretrained on COCO DATASET

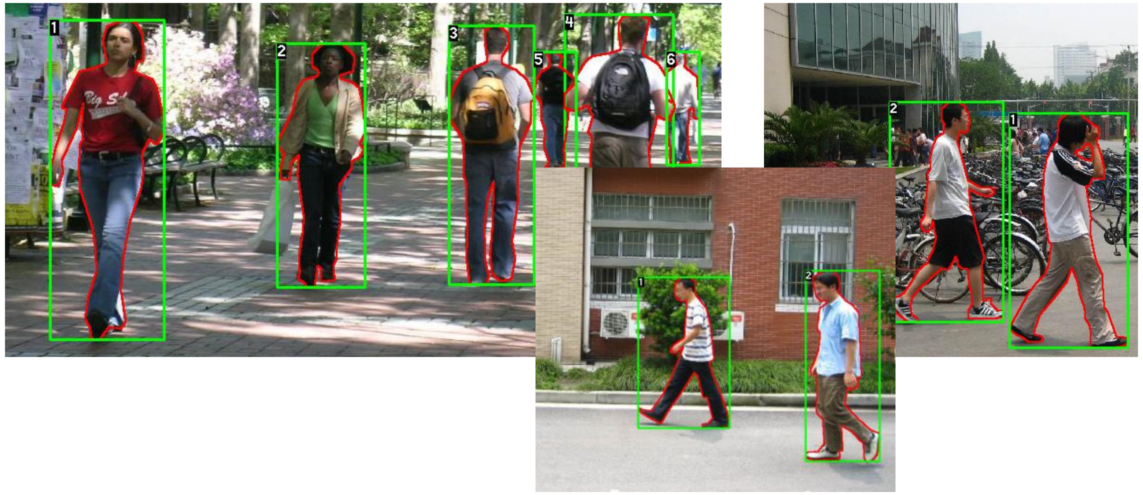




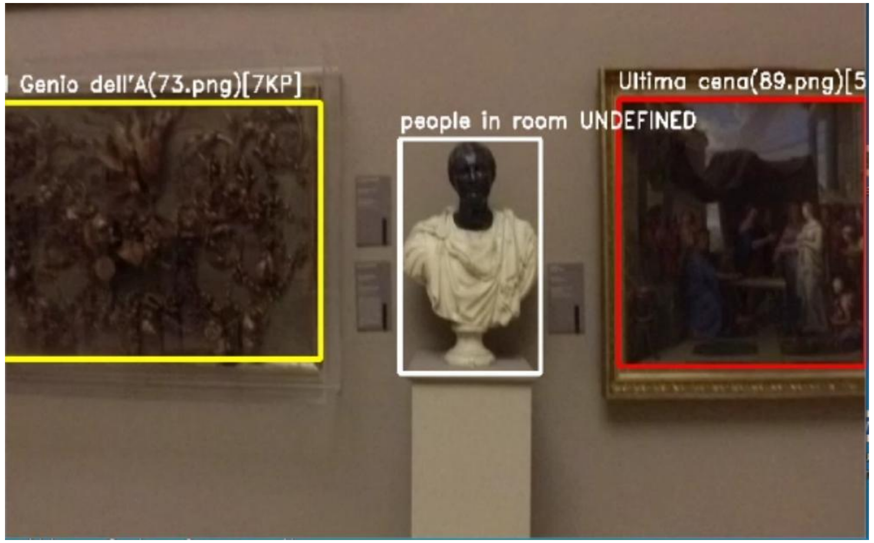
Pedestrian dataset finetune attempt

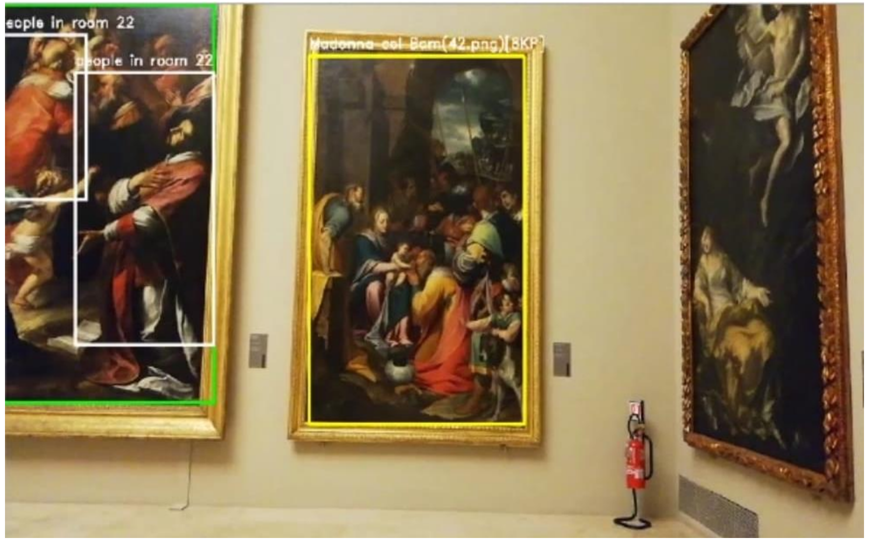
In order to avoid FALSE POSITIVE detections:

- Statues
- Potraited person

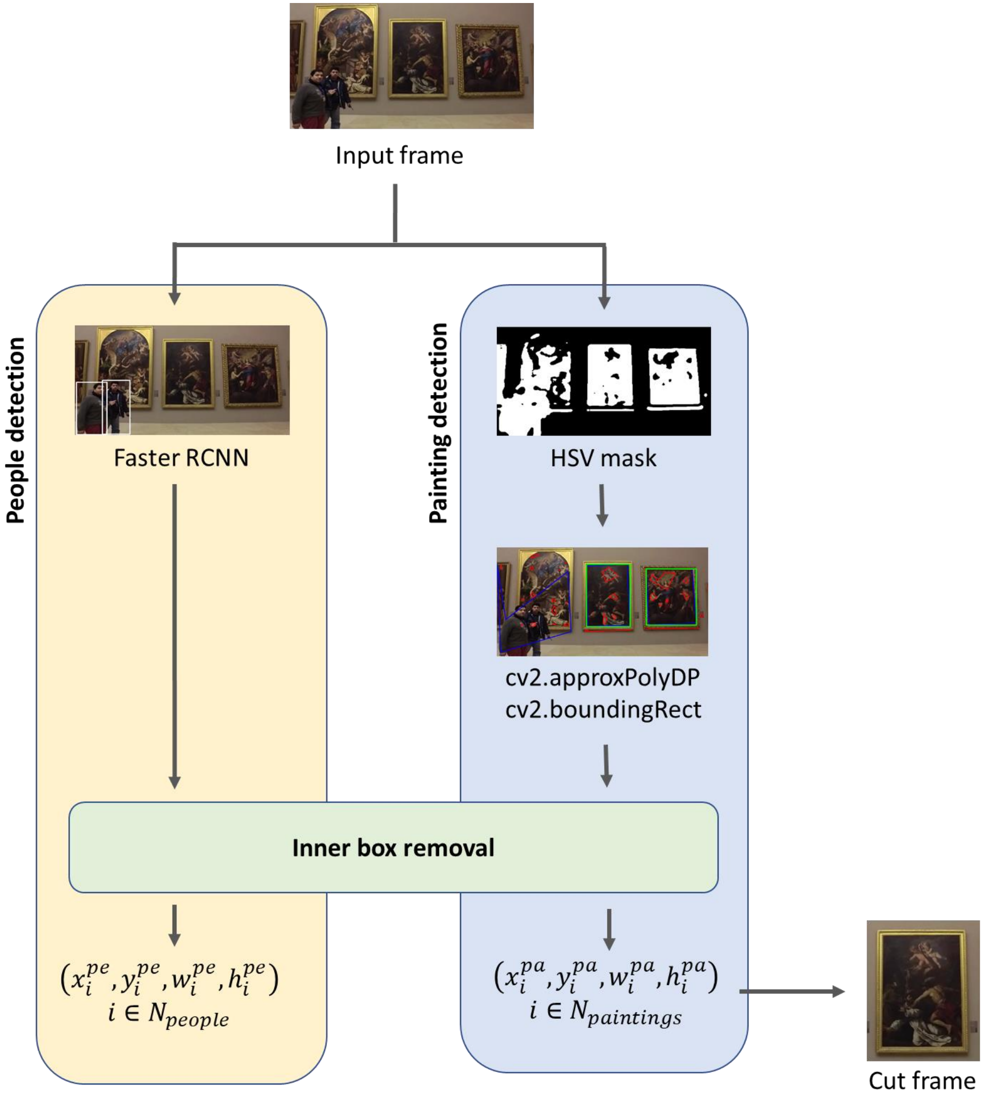


Penn-Fudan Database for Pedestrian Detection and Segmentation





Inner box removal



Input frame

People detection

Faster RCNN

Painting detection

HSV mask

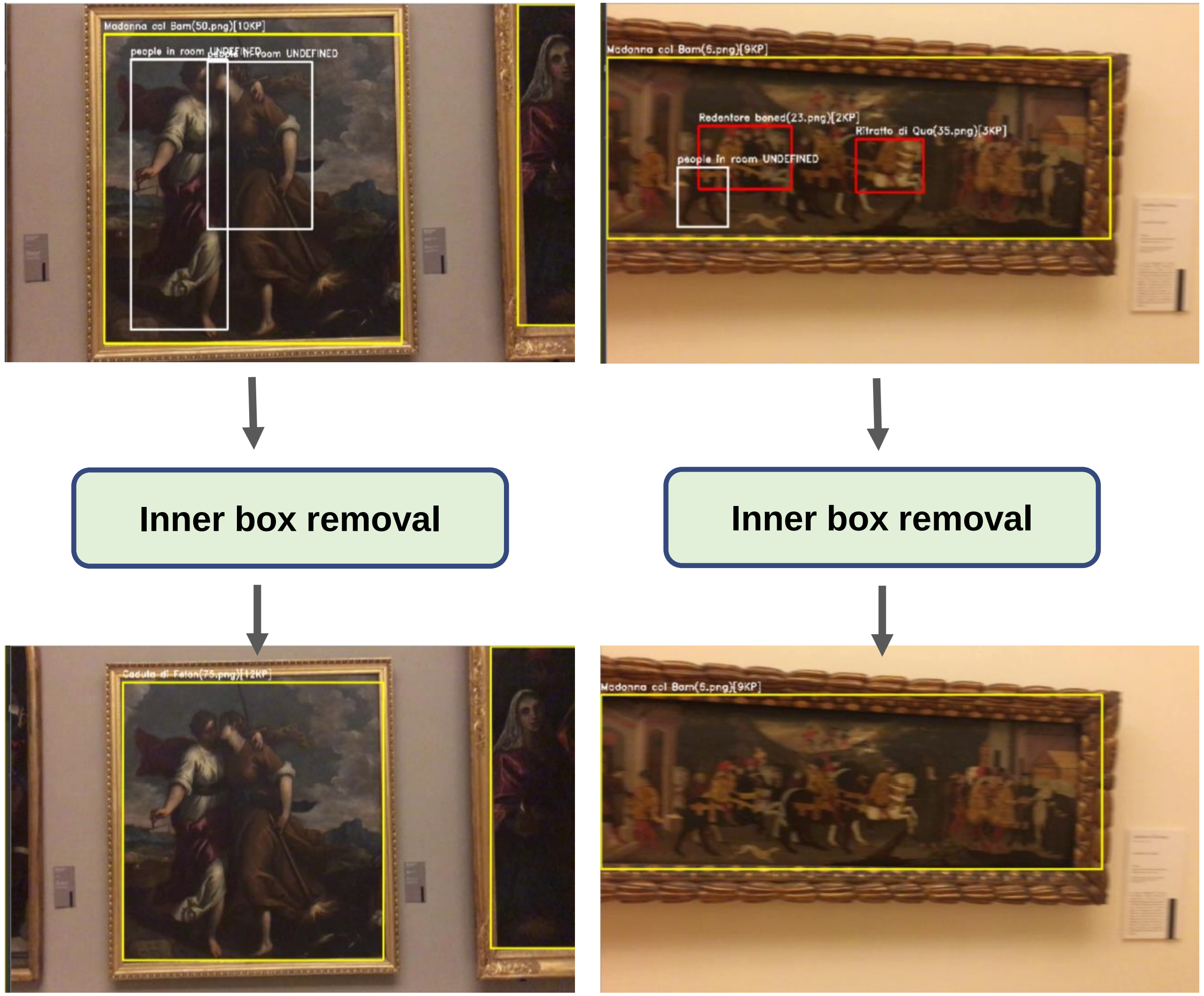
cv2.approxPolyDP
cv2.boundingRect

Inner box removal

$(x_i^{pe}, y_i^{pe}, w_i^{pe}, h_i^{pe})$
$i \in N_{people}$

$(x_i^{pa}, y_i^{pa}, w_i^{pa}, h_i^{pa})$
$i \in N_{paintings}$

Cut frame

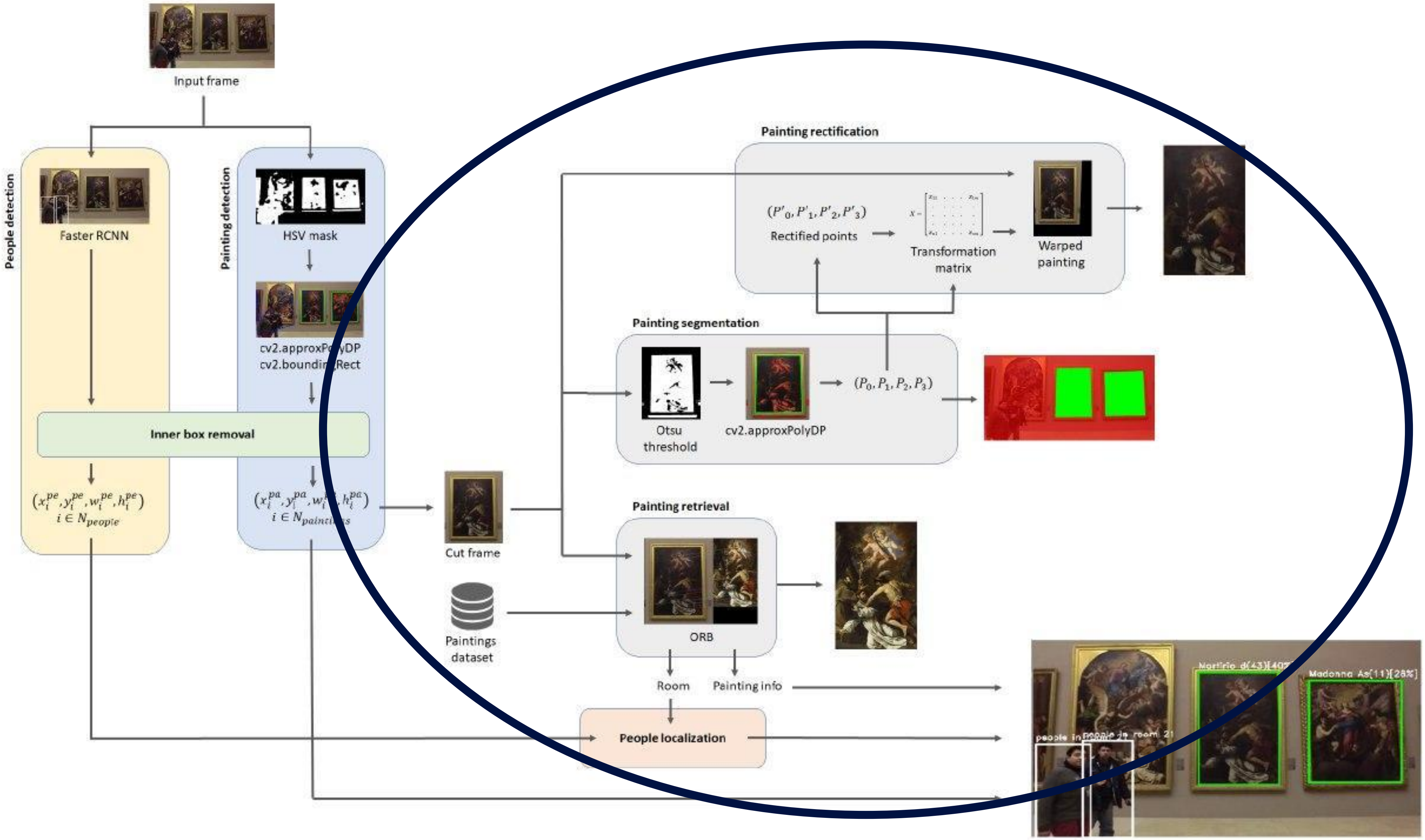


Inner box removal

Inner box removal

Madonna col Bam(50.png)[10KP]
people in room UNDEFINED
people in room UNDEFINED
Madonna col Bam(5.png)[9KP]
Redentore bened(23.png)[2KP]
Ritratto di Qua(35.png)[3KP]
people in room UNDEFINED
Caduta di Feton(75.png)[12KP]
Madonna col Bam(5.png)[9KP]

Pipeline

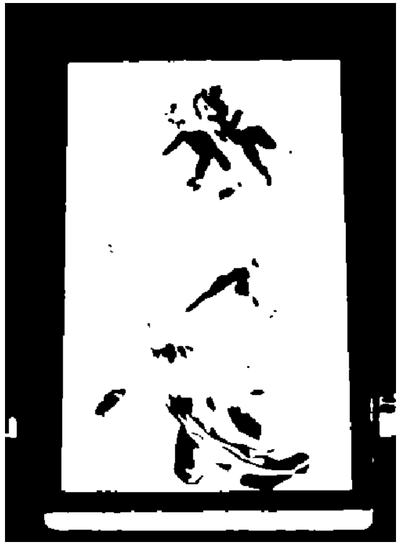

Input frame
People detection
Faster RCNN
Painting detection
HSV mask
cv2.approxPolyDP
cv2.boundingRect
Inner box removal
$(x_i^{pe}, y_i^{pe}, w_i^{pe}, h_i^{pe})$
$i \in N_{people}$
$(x_i^{pa}, y_i^{pa}, w_i^{pa}, h_i^{pa})$
$i \in N_{paintings}$
Cut frame
Paintings dataset
Painting rectification
(P'_0, P'_1, P'_2, P'_3)
Rectified points
Transformation matrix
Warped painting
Painting segmentation
Otsu threshold
cv2.approxPolyDP
(P_0, P_1, P_2, P_3)
Painting retrieval
ORB
Room
Painting info
People localization
Martirio d(43)[40%]
Madonna As(11)[26%]
people in room 21
people in room 21

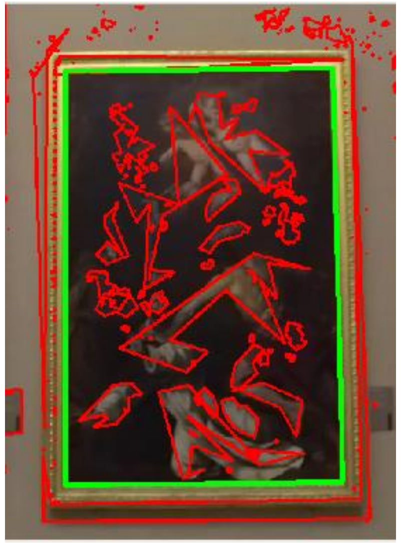
Painting Segmentation

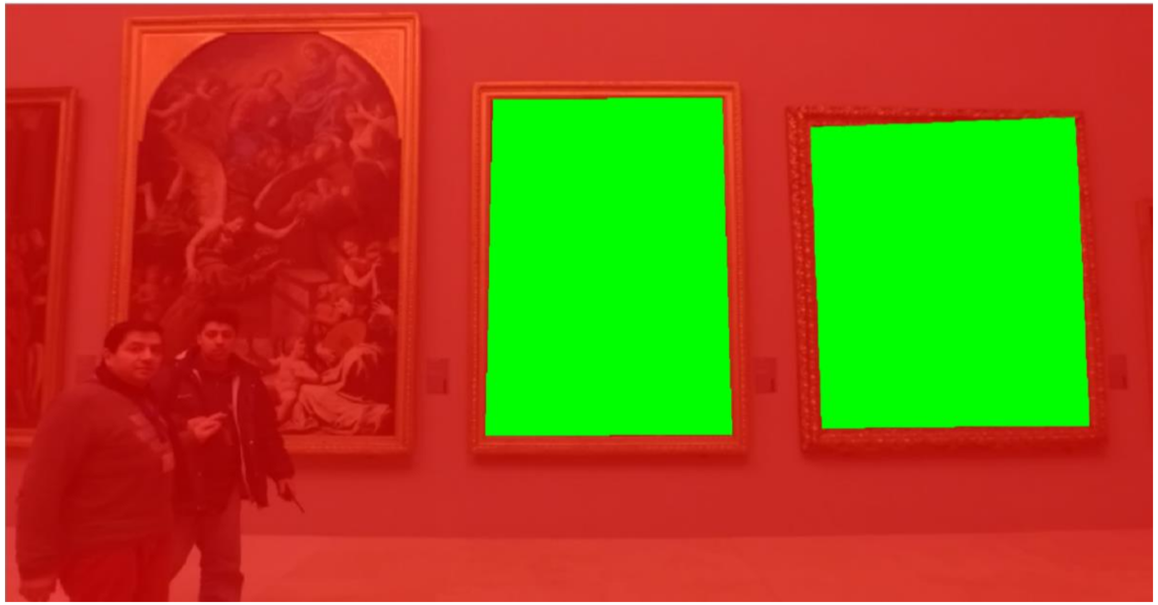
The painting segmentation is computed in order to have the exact area the painting starting from the detection:

- Add a padding;
- Deblurring *cv2.medianBlur;*
- *Otzu thresholding;*
- *FindContours + approxPolyDP*
- Filtering:
  - \#sides equal to 4
  - area at least 1/3 of the total area
- Ordering the corners

$$B = [P_0; P_1;\ P_2;\ P_3]$$







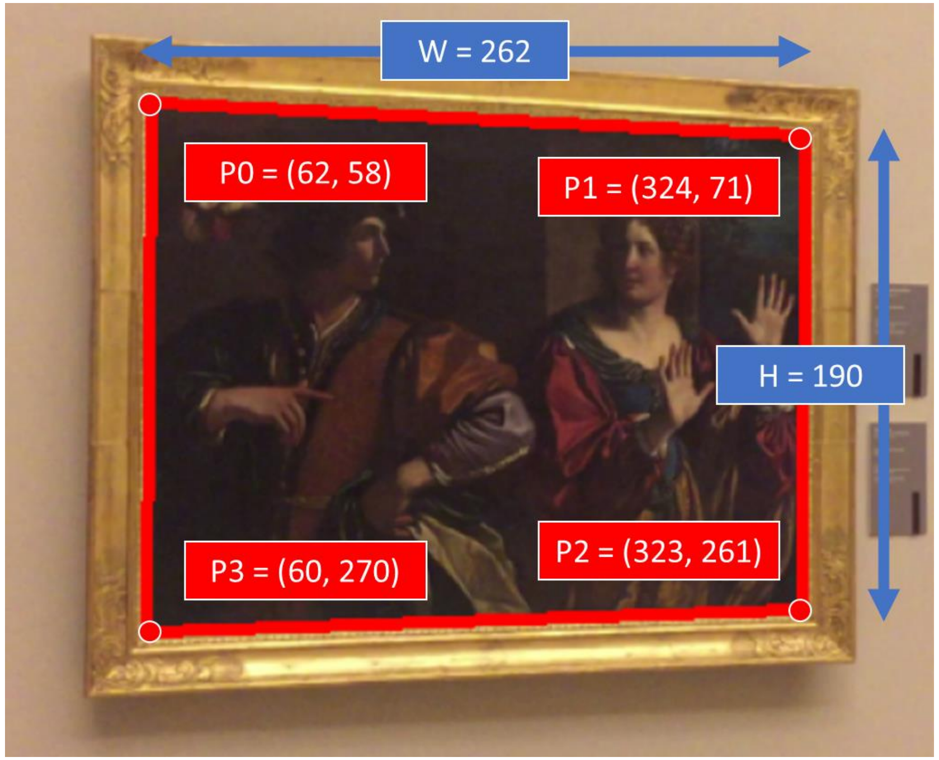
Painting Rectification

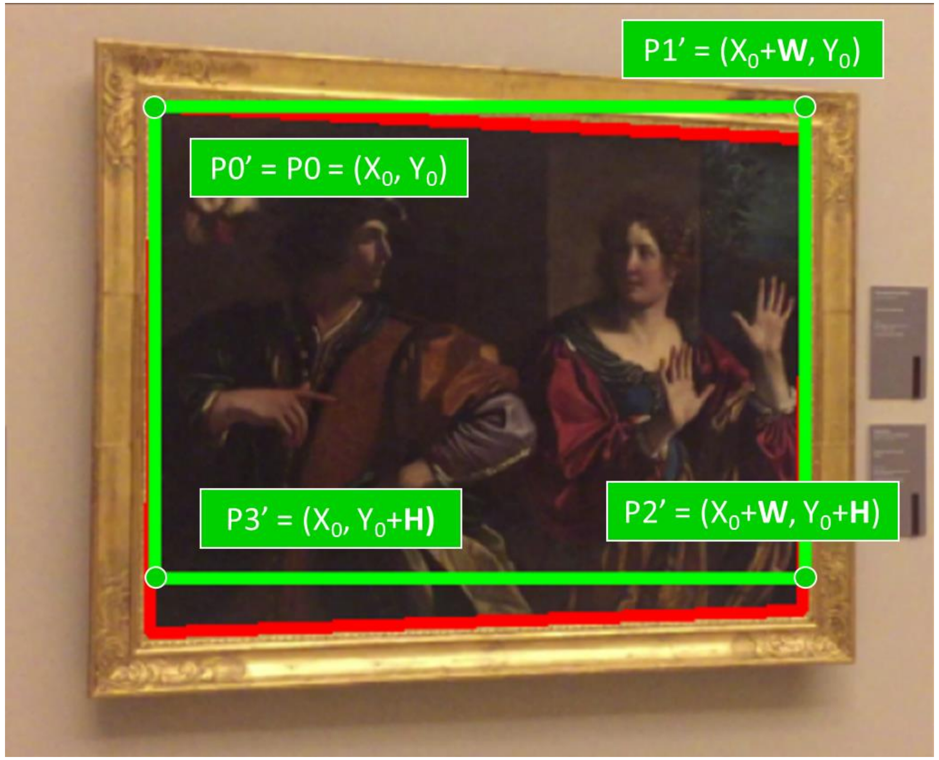
The rectification of the painting is computed uses the cropped frame with the detected painting and the four segmented edge points of the painting $B\ =\ [P_0; P_1;\ P_2;\ P_3]$

Using them and their Euclidean distances

$$H = \sqrt{(x_2\ -\ x_1)^2 + (y_2\ -\ y_1)^2} \qquad W = \sqrt{(x_0\ -\ x_1)^2 + (y_0\ -\ y_1)^2}$$

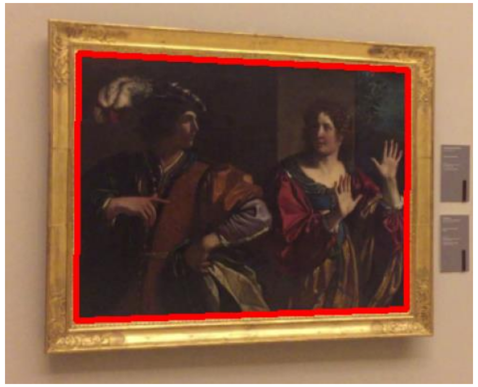
It is possible to create the new rectified box that will contain the rectified painting $B'\ =\ [P'_0; P'_1;\ P'_2;\ P'_3]$

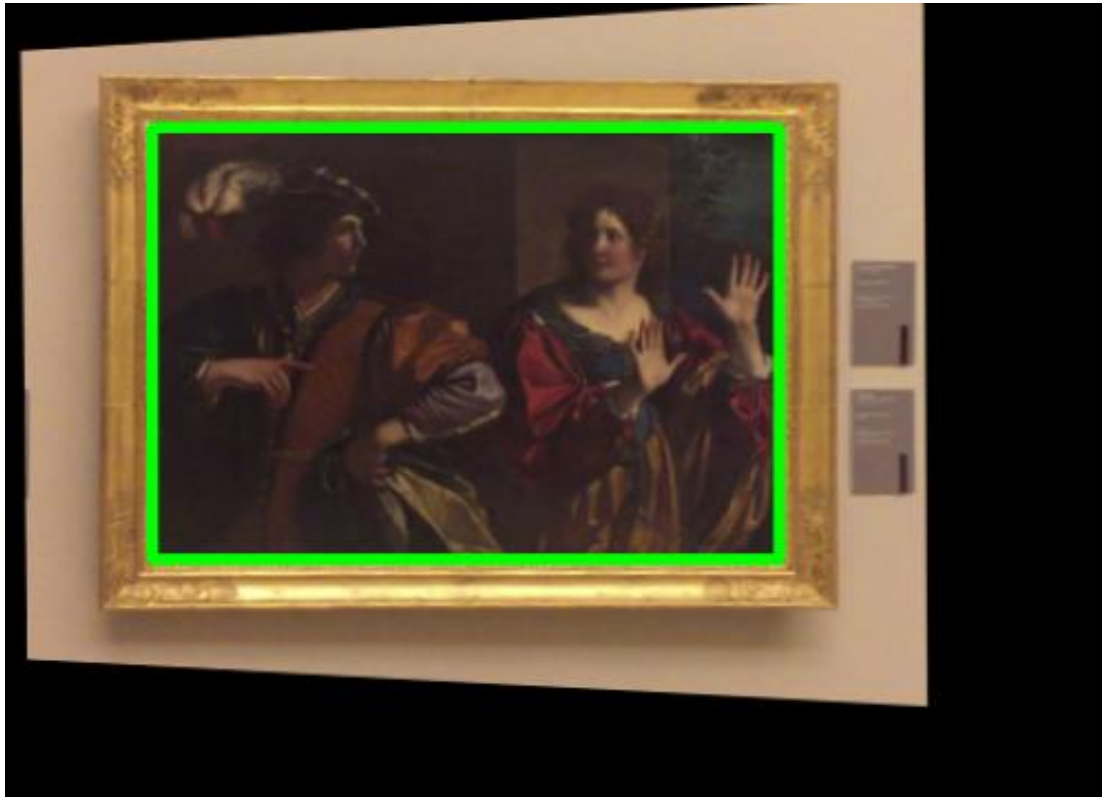


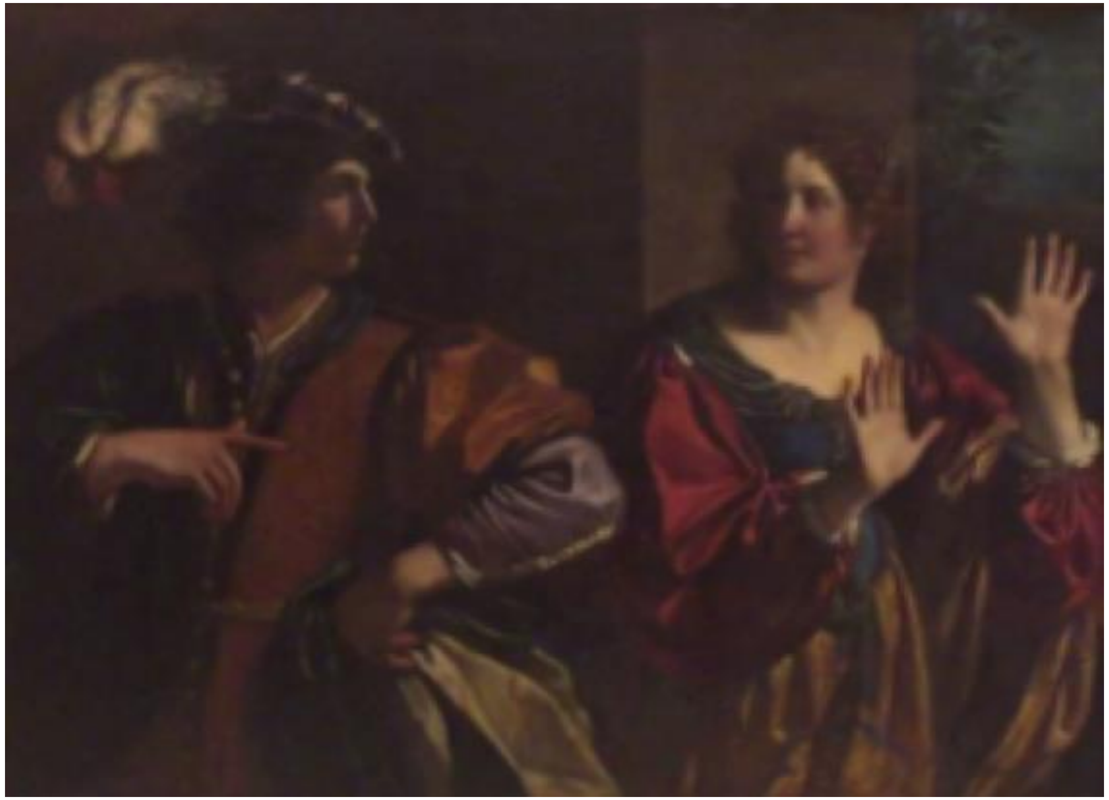

P1' = (X$_0$+**W**, Y$_0$)
P0' = P0 = (X$_0$, Y$_0$)
P3' = (X$_0$, Y$_0$+**H**)
P2' = (X$_0$+**W**, Y$_0$+**H**)

Painting Rectification

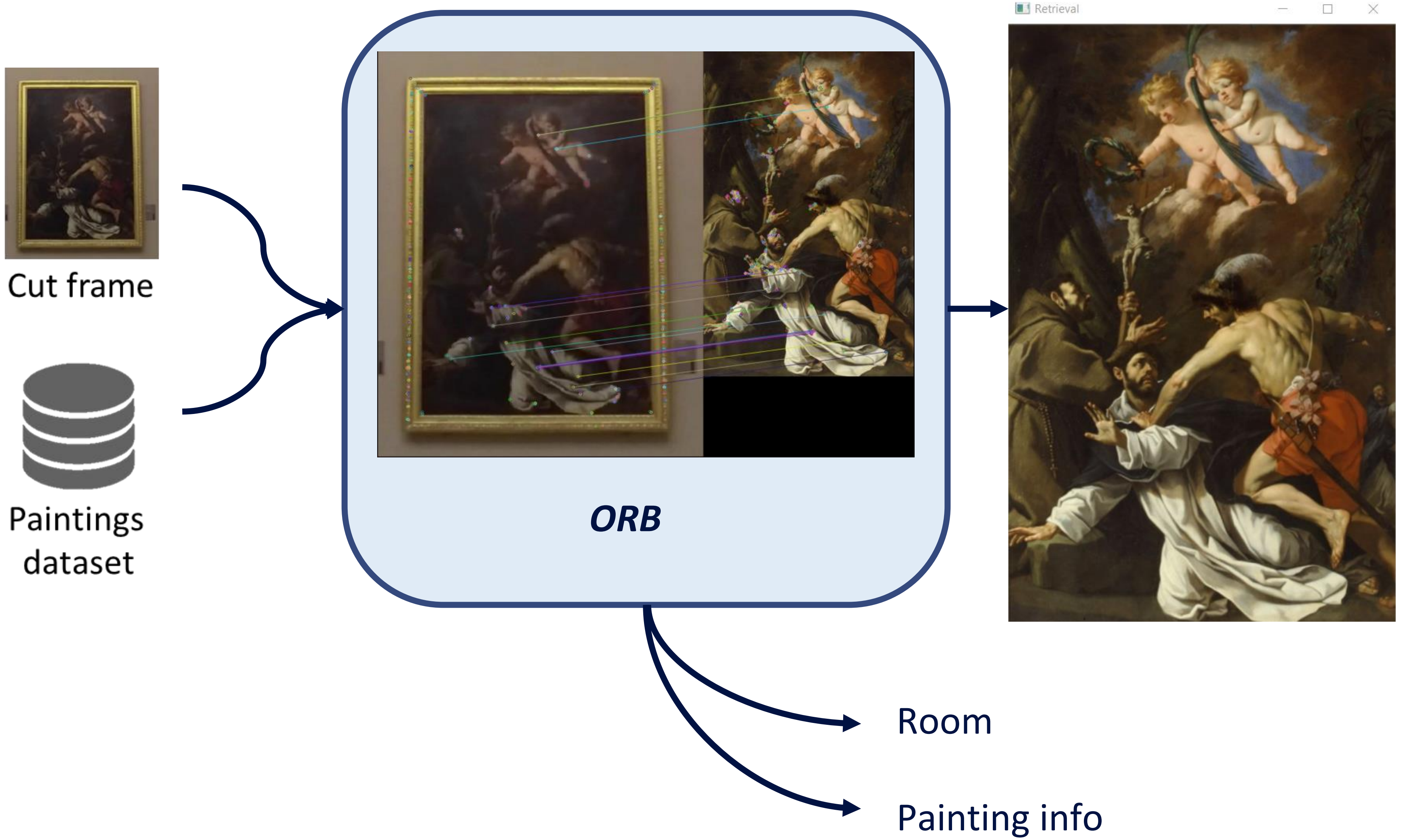
- Transformation matrix : *cv2.getPerspectiveTransform*
- Warped cropped frame : *cv2.warpPerspective*







Painting Retrieval



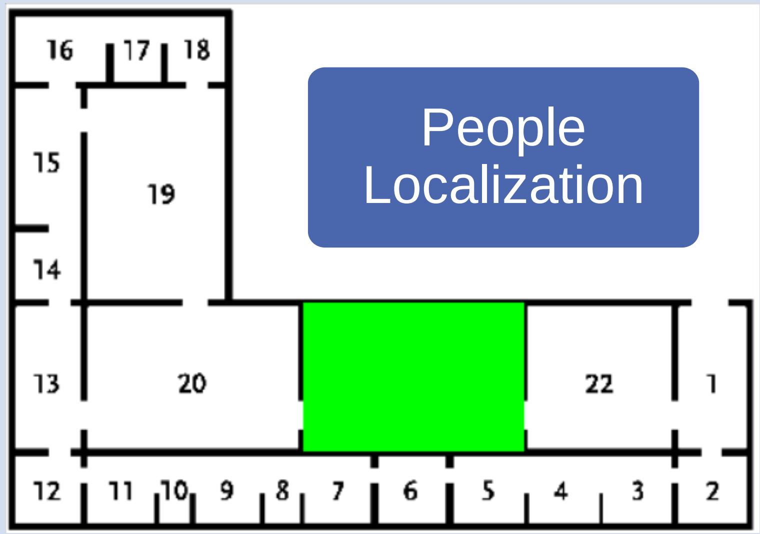
- *BFMatcher.knnMatch()*
- Distance ratio control
- Sorted list
- Bounding box color

People Localization

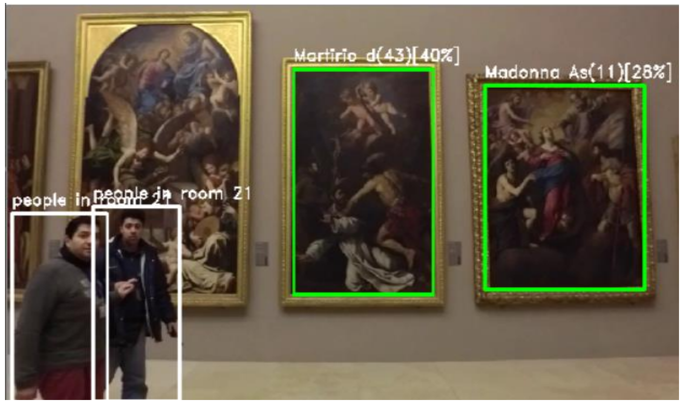
People Detection

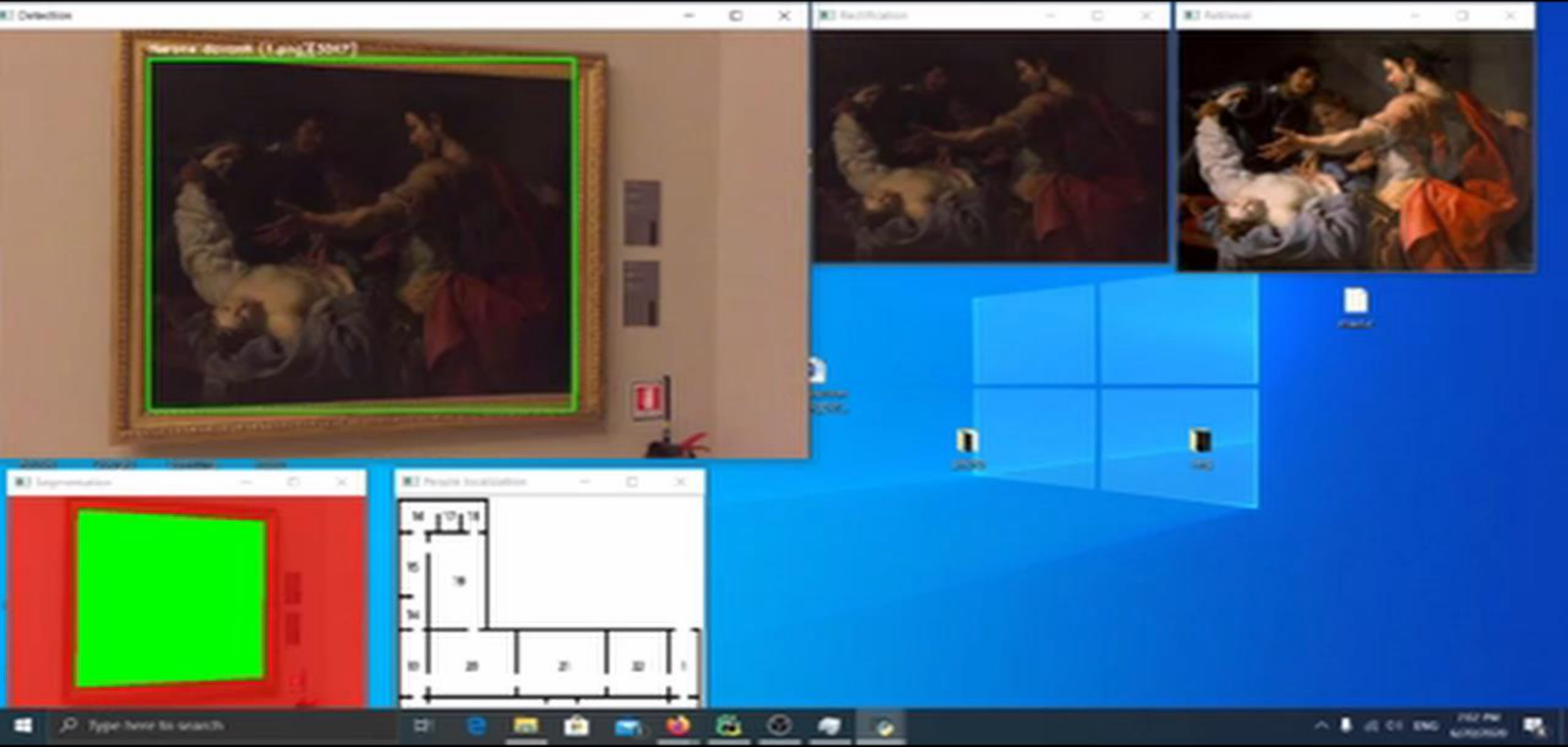
Painting Retrieval

Whenever a person is detected the model can localize her only if in the same frame there is also a painting detected.



People Localization




Detection
Rectification
Retrieval
Segmentation
People localization
16
17
18
15
19
14
10
20
21
22
1
Type here to search
ENG