Museum paintings retrieval and people detection

Marco Cagrandi
University of Modena and Reggio Emilia
Department of Engineering Enzo Ferrari
203232@studenti.unimore.it

Alessio Ruggi
University of Modena and Reggio Emilia
Department of Engineering Enzo Ferrari
203689@studenti.unimore.it

Daniele Lombardo
University of Modena and Reggio Emilia
Department of Engineering Enzo Ferrari
202214@studenti.unimore.it

Abstract

Image processing, retrieval and people detection are important computer vision applications. Here we present our work on the "Galleria Estense" dataset which contains videos and images from the "Galleria Estense" of Modena. We propose a method to detect and retrieve the paintings, the statues and the people in the museum based on different approaches: one with pure image processing and one with YOLOv3 network, trained with a custom annotated dataset of paintings and statues images.

1. Introduction

We have tackled the problem of detect paintings, statues and people in a museum (the Galleria Estense). Our objectives were: rectify the paintings correcting the perspective distortions, retrieve the correct paintings in the dataset, provide a precise segmentation of both detected paintings and statues, sum up all the information we had and link the paintings and the people to a precise room in the museum. In order to achive our results, as further discussed in section 3, many techniques are involved: image processing topics as edge detection, connected components analysis, relevant ROI detection, object segmentation; image retrieval; deep learning and neural networks.

1.1. The dataset

The dataset contains videos and images from the "Galleria Estense" of Modena.

We have 208 short videos taken using different cameras, aspect ratios and resolutions. Some video was taken with a GoPro camera which introduced some distortion and some video has frames particularly blurred due to the motion applied to the camera.

We have a database of 95 images that should represents all paintings of the "Galleria Estense", but during the development of the project we realized that many paintings were missing and so we had to expand the paintings db adding some paintings taken from the "Galleria Estense" website to improve the retrival and rectification tasks. We added 23 paintings that are named with a fixed suffix "A" followed by a zero-based sequential identifier, *e.g.* "A000.jpeg".

Finally the dataset contains also a CSV file with informations for every painting present in the db, including the position of the painting in the museum as a room number, and also an image representing the plant of the museum to do the people detection task. When we expanded the painting db we expanded the CSV file accordingly to maintain consistency.

2. Related works

To detect the paintings and statues in the museum we trained a YOLOv3 network with our custom annotated dataset. YOLO (acronym for "You Only Look Once") is an object detection network able to detect objects in images parsing the image only once, saving computation time w.r.t. other detection networks, however maintaining a good degree of precision. We choosed YOLOv3 because it can achieve good performaces both in terms of detection and speed as described in YOLOv3 paper *Joseph Redmon and Ali Farhadi* [3]. After the detection with YOLOv3, we used GrabCut algorithm for segmentation. GrabCut is an algorithm used for foreground extraction with minimal user interaction. User should inputs the rectangle which border the object of interest: everything outside this rectangle is considered sure background, while everything inside the rectangle may be both foreground or background. This minimal user interaction required by GrabCut is reduced to zero in our case, because, in the case of statues, YOLOv3 network inputs the rectangle, while in the case of painting the input ROIs are the one detected in the Painting Detection.

3. Approach

The proposed approach involve several elements: Painting detection, Painting retrieval, Painting rectification, People detection and Statue detection.

3.1. Painting Detection

The first thing to do in order to detect relevant objects is to perform edge detection, but right before we have precessed the image with a gaussian filter in order to remove the gaussian noise and a bilateral filter. A good method to detect relevant edges is the Canny algorithm [1]. It has been proposed with the following empirically determined thresholds

Thresholds	
Low	High
400	400

Table 1: Canny thresholds

with a Sobel operator [2] with aperture of 5. With low threshold and high threshold set to an equal value, we don't accept weak edges, even if connected with strong edges.

The next step is to perform a Dilation 3x3 followed by an Erosion (Closing), in order to connect the edges of the same relevant object, and to detect the connected components in the output image thus obtaining a list of ROIs. In order to discard unrelevant ROIs, we have set some rules that have to be satisfied in order to be considered a relevant ROI:

$$ratio_{ROI} = \frac{max(width_{ROI}, height_{ROI})}{min(width_{ROI}, height_{ROI})} < 3 \quad (1)$$

$$area_{ROI} > 0.015 \cdot area_{frame} \quad (2)$$

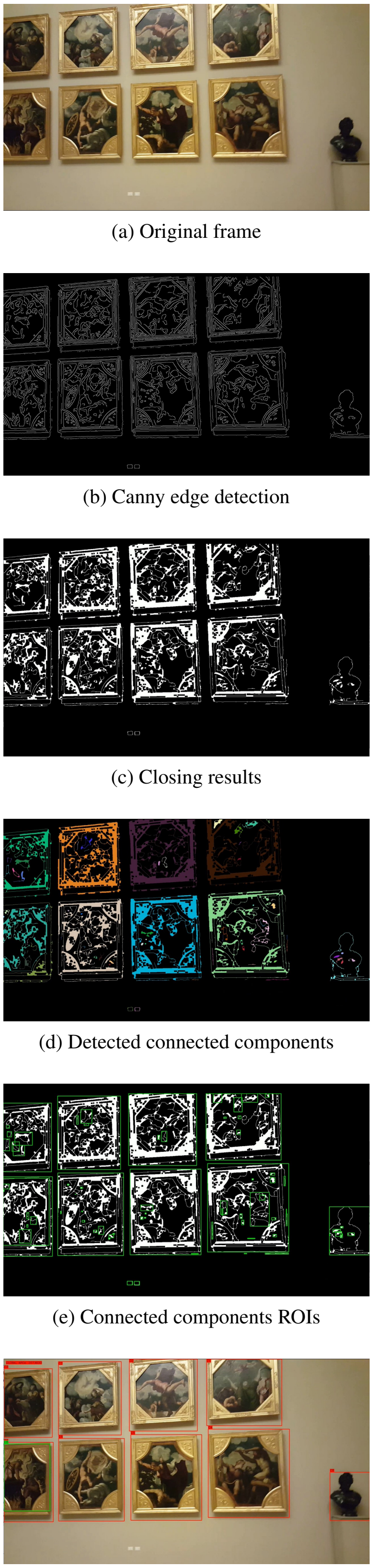
$$max\_overlap = 0.8 \quad (3)$$

If the overlap is greater than the threshold, the bigger ROI survives.

We then propose an optional optimization, named **otsu_optimization** and activated with the `otsu_opt_enabled` flag set to True,

$$otsu\_th(ROI) > 1.3 \cdot otsu\_th(frame) \quad (4)$$

where $otsu\_th(x)$ is the function that computes the Otsu threshold. This optimization helps when it comes to discard the ROIs containing *e.g.* wall sections, but sometimes affects the paintings that are overexposed to the light.



(a) Original frame

(b) Canny edge detection

(c) Closing results

(d) Detected connected components

(e) Connected components ROIs

(f) Relevant ROIs; in green the discarded ROIs due to the overlap

Figure 1: Relevant ROIs results

3.2. Painting Retrival

Painting retrival consists in the identification of ROIs passed from the Painting Detection. A ROI is extrapolated from the frame, and it is compared with all the painting in the database: - if the ROI is recognized as one of the painting in the database (with a certain level of security), it's labeled and information to decribe it are retrived from the database - if the ROI does not match any painting, they are classified as "unidentified object". The comparison is done with ORB, which is a keypoint detector, just like SIFT and SURF, but not patented. For every detected ROI, we use ORB to find the keypoint, and we compare them with every keypoints of evry paint in the database. The comparison of the keypoint returns an avarage distance: this distance represent how different are the keypoints found in the considered paint from the key of the ROI; the smaller the distance, the higher the probability that the considered paint from the database is the one that was filmed in the frame.

3.3. Painting Rectification

The paint with the smaller average distance from the database, that we will call the "Best Candidate", may not coincide with the object in the ROI; this may happend for many reasons: - the object is not a paint - the object is a paint, but the quality of that particular frame is very bad (the camera moving, the picture is overexposed to light) - the object is a paint, but is not in the databse Painting Rectification can be very helpful: with painting retrival, we select the paint that has the smallest avarage distace from the object in the roi, then, we try to rectificate the object in the frame based on the keypoints of the Best Candidate, and we obtain a rectified image, If the Best Candidate was actually the paint in the ROI, the rectification should be good, not distored: otherwise, the rectification will bring to a "Black Hole": a Black Hole happends when the wrong keypoints are used for the rectification. it can therefore be inferred that a Black Hole happend when our Best Candidate is wrong, it does not concide with the object in the paint. We can discard the first Best Candidate, and get the paint from the database which has the second smallest average distance. We repeat the process with the first 5 paint with smallest average distance, ot untill we obtain a good rectification.

immagini di black hole etc etc

Technically, we still use keypoint to detect if the rectified image is a Black Hole. After the rectification, we compute the keypoints on the rectified image: after that, we compute the avarage distance of the keypoints just calculated from the keypoints of the Best Candidate, that we already have. If the rectification was successful, the avarage distance shoul be low, otherwise it will be high, the rectification was a Black Hole and we have to try with another image. We set three threshold: - if the average distance is less than 35, it means that we are very confident that we have found the right paint in the database; for this reason, we identify the paint with the information retrieved from the databese and we set in qhich room we are in. - if the average distance is greater than 35, but less than 60, we are pretty sure that we get the right paint, but we are not sure enough to set the room - if the average distance is greater tha 60, we are in the situation of a Black Hole, the keypoints are too different, so we have to try with another paint

3.4. People Detection

We used YOLOv3 network pre-trained on COCO dataset to detect people on videos, getting the returned ROIs to draw them on output video.

3.5. Statue Detection

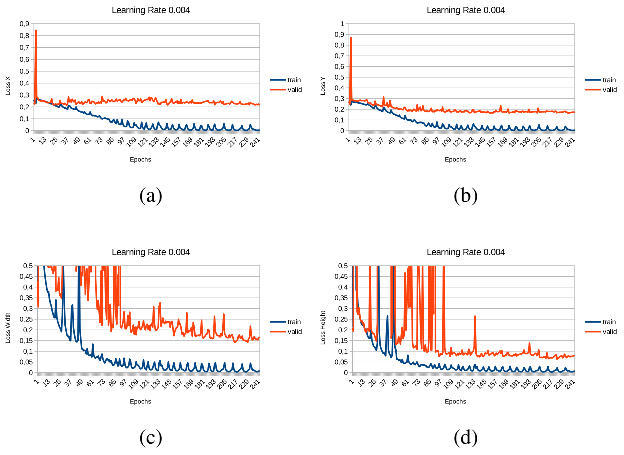
To detect the statues in the museum we fine-tuned the YOLOv3 network starting from weights trained on COCO dataset. At the beginning we thought to detect statues and paintings to improve the detection made with the Image Processing approach, so we trained the network on 2 classes: Paintings and Statues.

We made our custom annotated dataset ripping frames from all videos of the museum, storing them with a step of 250 frames for each video, obtaining 605 images mainly composed by paintings with only a 10% of images containing statues. We tried to train the network with this small and unbalanced dataset using different `Learning Rate` values but the result was unsufficient.

After those tries we thought to use the network only to detect the statues, because the Image Processing method was still better to detect paintings but often it wasn't able to detect statues. So we learned the lesson and we tried to balance the dataset selecting manually frames with statues from videos obtaining 282 images, each containg at least a statue with different point of views. We annotated those images with more focus on statues than paintings, then we made a simple data augmentation script to flip all statues frames, obtaining 564 images with statues. Finally our dataset contains 1169 images, slightly balanced in number of paintings and statues.

Using Adam optimizer with `Non Maxima Suppression` value of 0.4, `Confidence Threshold` of 0.8 and a `Learning Rate` of 0.004 we achieved an AP of **6.39%**, considering also the painting class that we discarded in detection phase.

To train the network we used an Nvidia GTX 1050 with 4Gb and it took us almost 4 days to reach the 242th epoch with a batch size of 2, then we just stopped the training due to overfitting on `confidence loss` [2]. We achieved the best results of AP with the 177th epoch and we used those weights to detect statues on videos trying different combinations of `Non Maxima Suppression` and `Confidence Threshold` values, defining them to



(a) (b) (c) (d)

Figure 2: (a) Loss X, (b) Loss Y, (c) Loss Width and (d) Loss Height

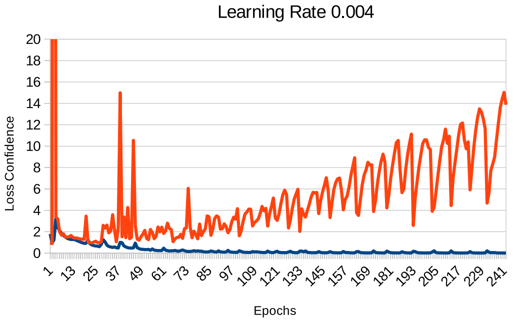


Figure 3: The Loss Confidence begins to overfit after few epochs, probably due to bad labeling of paintings when we expanded the dataset.

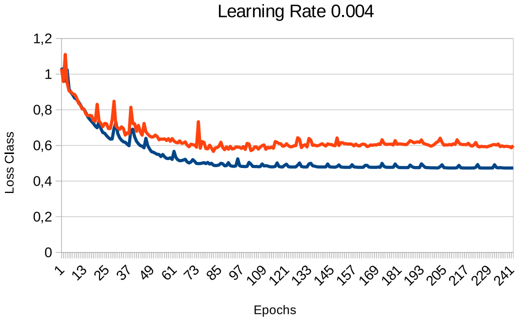


Figure 4: The Loss Class has a good trend.

0.1 and 0.98 respectively.

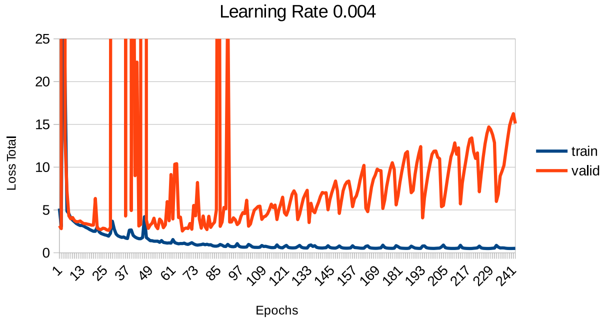
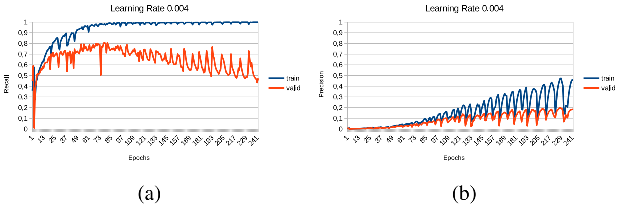


Figure 5: The Total Loss is affected mainly by the loss confidence curve [2], while the loss class [3] and localization losses [1] are good enough to improve the precision [5].



(a) (b)

Figure 6: (a) Recall and (b) Precision curves

3.6. Segmentation

The last section of the project deals with segmentation, both of paintings and Stautes. We used GrabcCut algorithm as defined in *Carsten Rother, Vladimir Kolmogorov, Andrew Blake* [4] : exploiting the ROIs received both from Painting Detection and YOLOv3, we were able to use GrabCut algorithm without user interactions. In particular, we took evry ROI extrapolated in each frame, and enlarged it: in this manner, we had a section of the frame, in which we could draw the rectangle to define what was definitely background. At this point, we used GrabCut to extrapolate what in the ROI was our object of interest, was that a paint or a statue. We highlighted the objects of interest in green

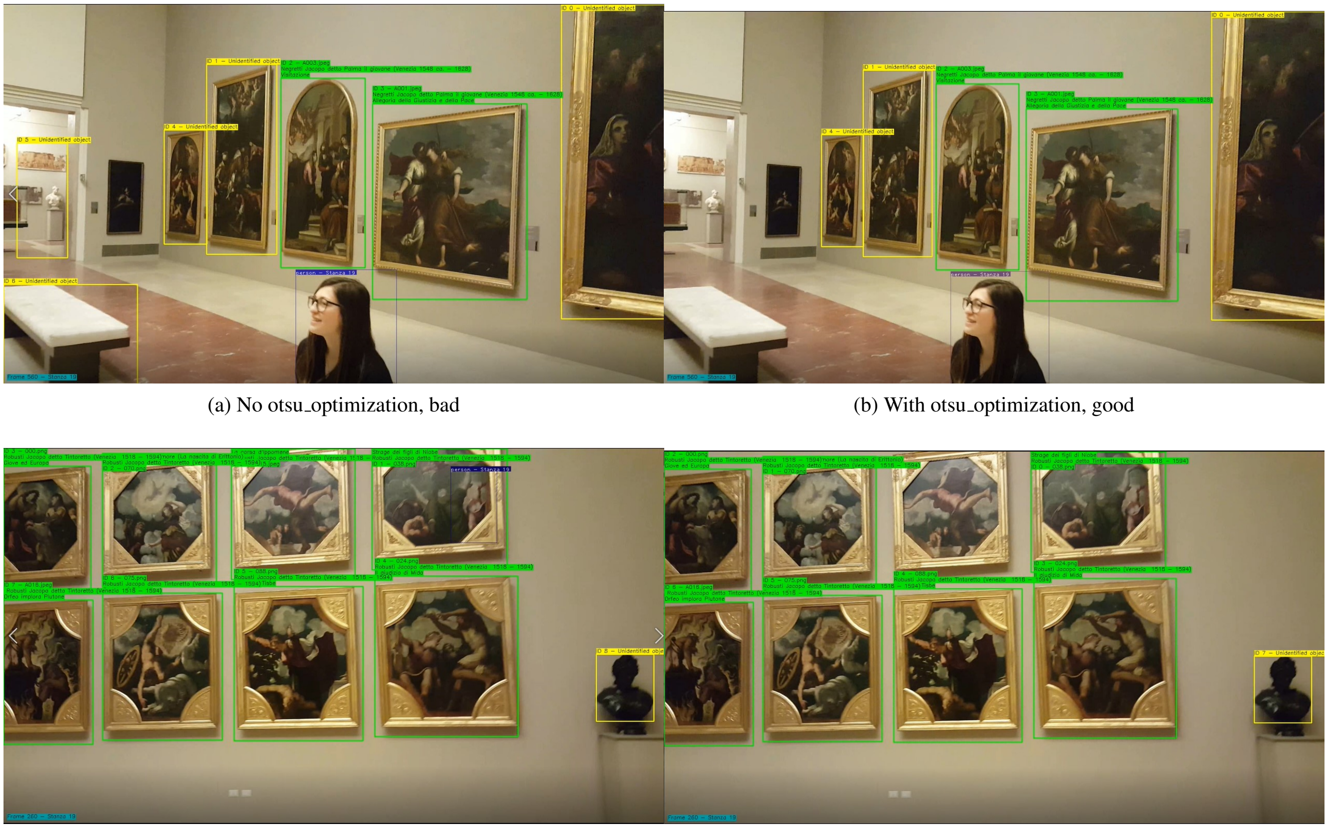
***Foto con frame e rettangolo che identifa sfondo/unkown,Foto con statue e quadri in verde**

4. Results

In this section, some results frame and training graphs will be shown.

5. Discussion

- migliorare la rete con dataset più ampio e bilanciato
- potremmo eliminare alcune ROI inutili (e.g. le etichette) utilizzando alcune tecniche di Image Processing


ID 0 – Unidentified object
Negretti Jacopo detto Palma il giovane (Venezia 1548 ca. – 1628)
Frame 560 – Stanza 19
Frame 260 – Stanza 19

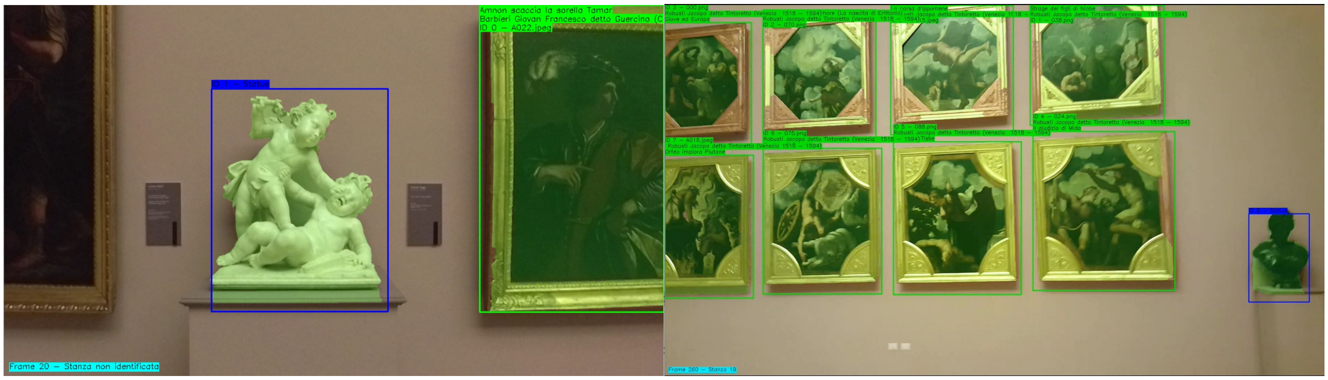
(a) No otsu_optimization, bad

(b) With otsu_optimization, good

(c) No otsu_optimization, good

(d) With otsu_optimization, bad

Figure 7: Example of otsu_optimization (no statue detection nor segmentation)


Amnon scaccia la sorella Tamar
Barbieri Giovan Francesco detto Guercino (C
ID 0 – A022.jpeg
Frame 20 – Stanza non identificata
Frame 260 – Stanza 19

(a)

(b)

Figure 8: Example of segmentation and statue detection

- Retrival, Rectification e Detection funzionerebbero meglio se avessimo un database di immagini completo anche con statue

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[2] N. Otsu. A threshold selection method from gray-level his-

tograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[3] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “grabcut”: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04, page 309–314, New York, NY, USA, 2004. Association for Computing Machinery.