Painting Detection and Rectification with People Detection

Aniello Panariello - 140195, Fabrizio Sorgente - 133855, and Emanuele Fenocchi - 148869

University of Modena and Reggio Emilia

June 3, 2020

1 Introduction

The aim of this work is to detect paintings and people inside a museum environment and then perform retrieval and rectification of the detected painting from a database of high quality images, we also detect statues. The detection of the three objects is performed with a custom trained YOLO network, while the retrieval is done by ORB keypoints. For the rectification we exploit the keypoints obtained by the ORB to find the homography matrix. Once we have found the paintings and the people we can localize the latter by getting the localization of the painting. The direction in which the person is facing is computed by a face detection and assuming that if the person is not looking at the camera then he is facing a painting.

2 Detection

The detection of paintings and statues is done through a custom YOLOv3 neural network.[**yolov3**] Yolo is an architecture that provides a new approach to object detection, a single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. In order to use this neural network we have labeled 1343 images, 805 of them have been used to train the network, 269 for the validation set and 269 for the test set, providing a total number of 3733 labels. The images of paintings and statues were taken by the frame extracted by some videos recorded in the Galleria Estense, for the person class was instead used a mix of labeled images taken from the previous videos and a some images of people in another museum.

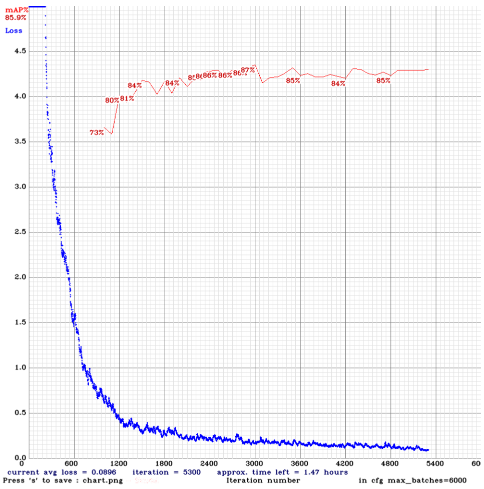
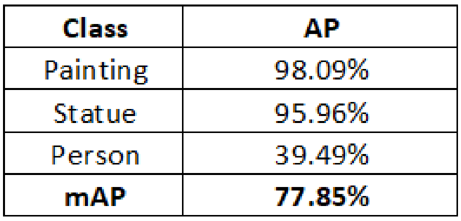


Figure 1: YOLOv3 trained on our custom dataset

The training has been performed with darknet [**darknet**] taking the mAP every 1000 iterations 1, this allowed us to take the best weights that weren't affected by overfitting. In the end we got a powerful neural network with high performance and capable of a good generalization.

Detection Performance				
	Painting	**Statue**	**Person**	**Overall**
TP	520	222	70	812
FP	133	18	28	179
Precision	79.63%	92.50%	71.42%	82%
Recall	-	-	-	85%
IoU	-	-	-	70%

Table 1: Detection performance



Class	**AP**
Painting	98.09%
Statue	95.96%
Person	39.49%
mAP	**77.85%**

Figure 2: Average Precision and Mean Average Precision

2.1 Comparison with previous technique

In a previous pipeline the detection was made without neural networks: the image was preprocessed and then we computed the edges with the Canny edge detector [**canny**]. From the edges we took the signficant borders that identified the paintings and finally draw the region of interest around the borders. Despite this method worked well in the main scenery, it wasn't able to adapt to strong luminance variation and to manage the presence of shadows. The accuracy that was produced didn't satisfy the standard of the project, for this reason we choose to implement a neural network.

3 Painting Retrieval

Painting retrieval uses ORB [**orb**] keypoints detector and descriptor to find matches between two images. ORB, is at two orders of magnitude faster than the old used SIFT [**sift**] and this is the main reason why we have chosen to use this method, in order to achieve the painting retrieval with a good performance/results ratio. An example of retrieval is shown in fig. 3.

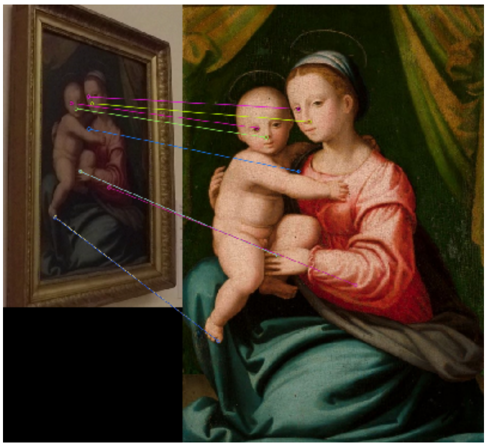


Figure 3: example of painting retrieval

3.1 Improvements

To improve the retrieval and to reduce false matches, we used the ratio test described in Lowe's paper [**sift**], in order to get the best matches. Given the two best matches (the best match and the second best match) for a keypoint, we define a *threshold* and if the ratio between the distance of the two best matches, respectively d_1 and d_2, is above that threshold, $\frac{d_1}{d_2} > threshold$, we reject that keypoint, considering it equivalent to noise and because the best match is not so different from the second one.

Our goal was to keep the number of paintings correctly found in the database high but at the same time, not decreasing the number of correctly not found paintings that are not in the database. We have chosen *threshold* = 0.6 because increasing it even by just 0.1, although it decreased the number of positive answer to paintings not listed in the database, this resulted in an increasing number of wrong matches for the paintings in the database. Decreasing the threshold led to an opposite situation and both of the cases didn't meet our goal.

Computing the same keypoints and descriptors for the same paintings in the database has resulted in a slow start, causing the retrieval to wait a couple of seconds or more. Computing the keypoints and descriptors one time and storing them into a file, reduced the retrieval initialization by more than 40%.

3.2 Evaluation

To evaluate the retrieval, we tested it with a sample of 3 random frames per video, with a total of 100 randomly selected videos out of 208. Some of these frames either did not contain any frames or the region of interest of the paintings exceeded the frame size, therefore 100 frames out of the total of 300 were discarded. 266 are the total paintings detected using our trained model and 45 of these were discarded because they were unrecognizable, due to their small size and/or their brightness. For the remaining 221 paintings, we manually counted every time we saw a wrong or a right answer. More precisely, we checked if the painting was in the database and the retrieval answer, building the matrix in table 2.

		Retrieval answer	
		Found	**Wrong or not found**
Paintings	**In DB**	58	60
	Not in DB	47	56

Table 2: Painting retrieval evaluation results

The result is that 58 out of 221 paintings were in the database and they were correctly retrieved from it, 56 out of 221 were not in the database but the retrieval correctly gave us a no match found. The remaining paintings are the ones that were wrongly retrieved from the database, 60, and the ones that had not an instance in the database but a wrong match was uncorrectly found.

Accuracy is the measurement used to get information on how good our configuration is:

$$\textit{Accuracy} = \frac{58+56}{221} \approx 0.52$$

We think that this result is pretty good, based on the fact that we could increase the real matched paintings, reducing the uncorrectly found paintings that were not actually in the database, adding them manually to it.

4 People Localization

Our approach to achieve this task is based on the quality of the detection and painting retrieval. In fact, if our trained model correctly detect a person, in order to localize that person, we use informations about the paintings detected by the model and retrieved from the database.

4.1 Evaluation

We have access to the paintings_db, with informations about the room where each painting is located, but there are two main problems that worsen our evaluation:

a) The painting retrieval almost performs a random choice when a painting is in the database.

b) We do not take into account the scenario in which the camera and the person are in a room, while a detected painting is in another room, visible through a door.

In our evaluation we discarded all cases that match b) and the results are all depending on the painting retrieval, which are show in table 2.

5 Face detection

We used Haar Cascade classifiers proposed by Paul Viola and Michael Jones in their paper [**haar˙cascade**] in order to achieve this task. First, we use our trained model to detect a person, then the ROI of that person is passed to the face detector. The face detector performs the algorithm to check if a person is facing a pinting, based on these following scenarios:

a) Face found

  (i) Eyes found (at least one)

  (ii) Eyes not found

b) Face not found

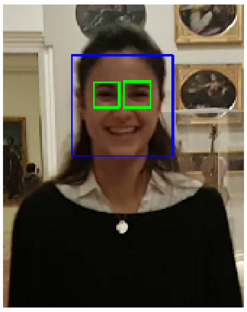


Figure 4: the case in which the face detector correctly found a face from the person ROI

In case (i), if the face is found with its eyes, we assume that the person is facing the camera and there is no painting behind it, like in fig. 4. In case (ii), if the face is found but the eyes are not detected, we assume that the person is possibly facing a painting, this case is a particular scenario where the person could be in profile. The case b) has the same assumptions of the case (ii) because if the face is not detected, this could mean that the person is turning his back to the camera and, therefore, is possibly facing a painting, like in fig 5. In these last scenarios, we take into account the paintings ROI and we check the overlap with the person ROI. If the person ROI overlaps at least one painting, the person is in front of that painting, otherwise the person is not looking at any painting.

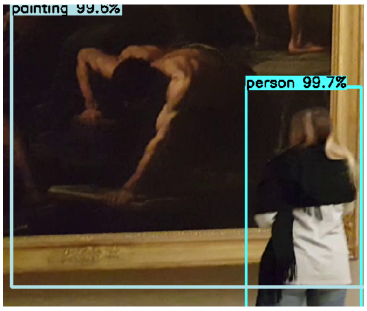


Figure 5: the case where the face detector can't find the face

5.1 Evaluation

The assumptions we've made, have allowed us to model most of the possible cases, but since the videos where there is a clearly visible person are only fews, the test evaluation may not represent the real accuracy of our approach. As a result, the test has been done using 2 frames per second for each videos where there is at least one person for more than 3 seconds, with a total of 8 videos. 466 are the total frames used and only 121 are the optimal candidates for the test, where a person is clearly

		Facing painting	
		True	**False**
Face detector answer	**True**	14	10
	False	13	57

Table 3: Face detection results

visible, and was not too far from the camera. Since the face detector takes in input the person and paitnings ROI, our test is also affected by the quality of our trained model, therefore, not detecting some of the people and paintings, led us to discard another 42 frames.

After all of these reductions, we analyzed a total of 84 frames with 94 people inside it. The evaluation results are shown in table 3 and gave us an accuracy of:

$$Accuracy = \frac{14 + 57}{94} \approx 0.85$$