Week 2

Museum Painting Retrieval

Group 8: José Manuel López Camuñas, Marcos V. Conde, Alex Martin Martinez

Index

1. Image retrieval with text
2. Multi-histogram retrieval
3. Method for QSD1:text detection
4. Methods for QSD2:detect paintings
5. Results
6. Discussion and conclusions

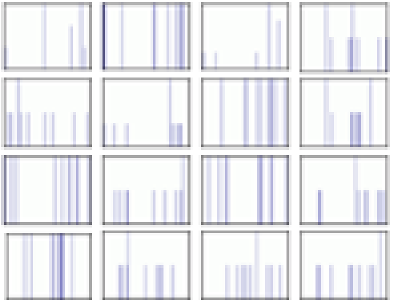
Multi-histogram retrieval

Before starting with the methods used for his weeks dataset another method was implemented to retrieve images with lasts weeks images from QSD2.

Instead of getting the histogram from the whole image, the image was divided in blocks and then the histogram for each part that came for the division was calculated.

Multi-histogram retrieval

The extracted histograms were then stacked and the resulting vector was used to compute the similarities between the DDBB images and the ones from QSD2 week 1.



Compute similarities and retrieve image

Results: Multi-histogram

The histograms computed were 3D histograms with 16 bins per axis.

Therefore for each image we have a descriptor vector:

[number of blocks, 16 , 16 , 16] → [n_blocks, 4096] and flatten

With those histograms the metric to compute the similarity of the images was the Hellinger distance.

With the QSD2-W2, the method could not retrieve any of the images correctly. We believe the problem is the high-dimensional descriptor, therefore, we reduce the number of blocks from 16 to 4, and the number of bins from 16 to 8. This did not solve our problem. Other options that we tested are: flattening the image, use PCA to reduce the dimensionality of the descriptor. However, the results didn’t improve.

Image retrieval with text

The images that had to be retrieved this week had the names of the pictures in black or white boxes over the image of the picture. This text had to be detected and removed before the retrieval was performed to obtain better results.

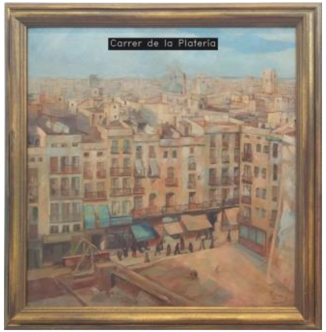


Image with text

Text detection and mask creation processing

Image with text text removed

Method for QSD1:text detection

In order to detect the text and produce a bounding box so as to remove it from the image the process followed was the following.

As the text was balck/white over a white/black box a top-hat and a blackhat filter were used remove other regions that didn’t present this behaviour.

Image after tophat filter

Image after blackhat filter

Method for QSD1:text detection

When using the **tophat filter** the white letters resulted in higher intensity and when using the blackhat the black letters were the ones highlighted. In order to detect the letters in any of the cases both resulting images from tophat and blackhat filters were added and divided by two, and then the resulting image was thresholded to leave only the higher intensities.

Tophat

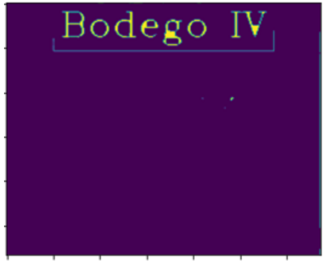
+

Backhat

Applying threshold to generate binary image

Threshold value = 70
(founded by trial and error)

Binarized image



Method for QSD1:text detection

To the resulting binarized image it was applied an opening filter followed by a dilation. Then this image was used to detect contours based on the difference in pixel intensities founded in the image and only the contour with the greatest area was selected as the bounding box of the text and the contour was used to generate the mask that removes the text in the image.

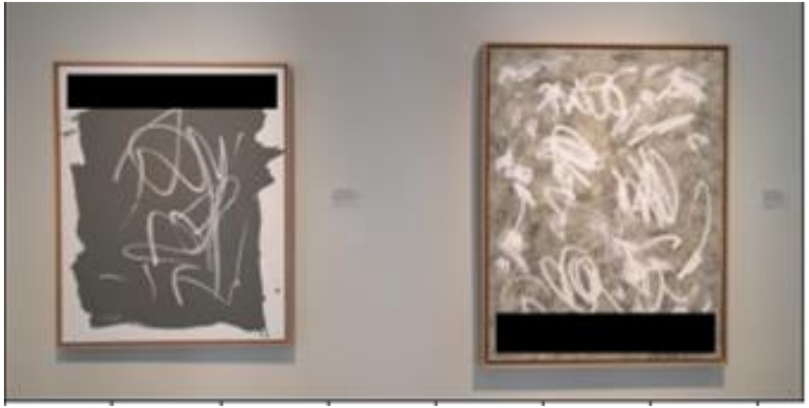
Result after opening and dilation of binary image

contour extraction and mask generation

Resulting image after applying the mask to the original image

Results:Text detection

Examples of good results:



Results:Text detection

The text detection algorithm behaved well on the majority of the images but with paintings that presented a lot of contrasts the filters couldn't remove the painting details resulting in masks that erases almost all the image.

Original image

Bad Result after applying text detection and the generated mask

Results:Text detection

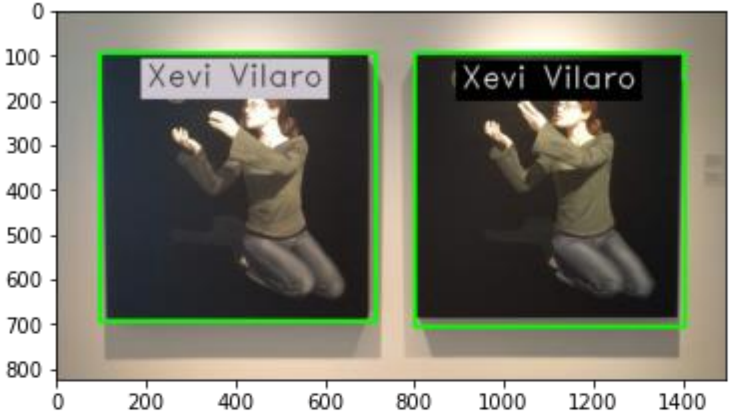
With the ground truth bounding boxes the mean IoU over all the generated masks was calculated. The mean IoU for the QSD1 presented a value of 0.6016

	Iou
QSD1	0.6016
QSD2	0.1957

Table 1: We report: *Intersection over Union (IOU) metric for the text detection generation methods performed to the QSD1 and QSD2.*

Methods for QSD2: Detect paintings

In a more realistic setup, multiple pictures might appear in the same image. We aim to detect the main paintings in a given image using **Morphological Transformations**.



Methods for QSD2: Detect paintings

The step-by-step process is as follows:

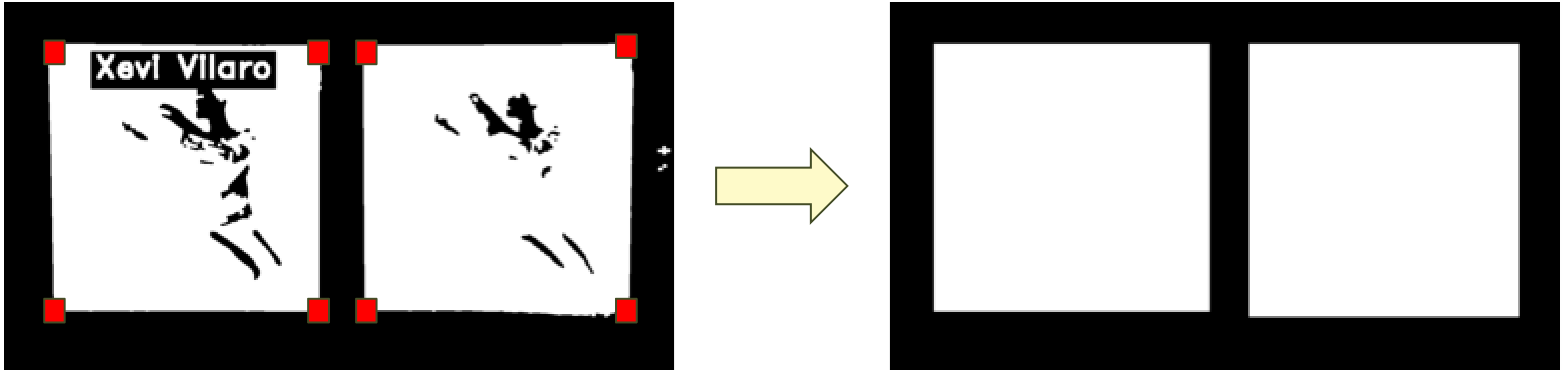
1) Convert the RGB image to Grayscale and apply Otsu's threshold (as we did last week).
2) Apply a **closing** transformation (Dilation followed by Erosion), followed by a dilation operation. We fine-tuned the operations and the kernels such that the IOU was maximized. At the end this combination was the best. We use squared kernels of size 10x10.

Methods for QSD2: Detect paintings

We then apply a contour research algorithm using “*cv2.findContours*”.

The boxes/masks are squared using the vertex (red colour), which are the max/min values in the x and y axis for each contour.

If many boxes are found, we only use the 2 biggest boxes.

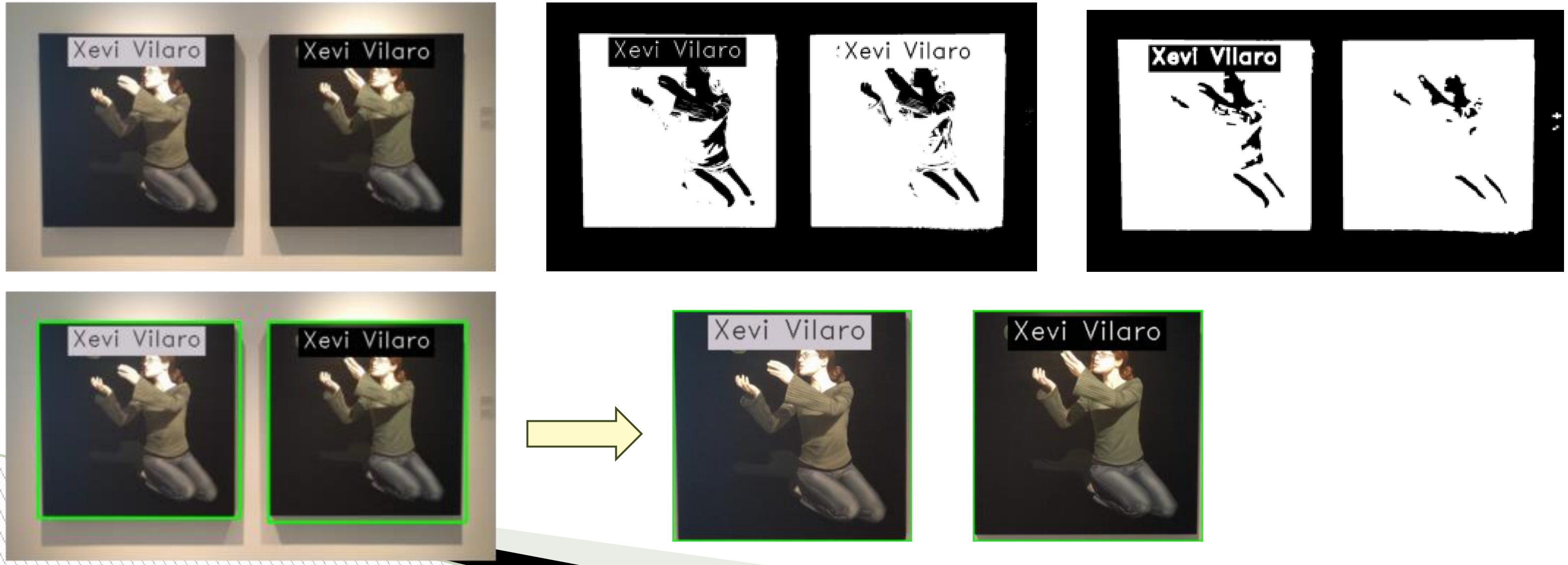


Produce mask. IOU 0.945 vs GT.

Results: detect paintings

The following is a step-by-step result sample.

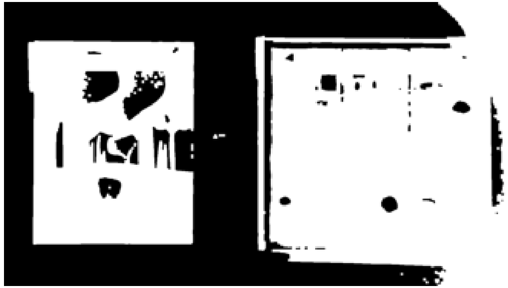
The mean IOU score using QSD2 is: 0.784.



Results: detect paintings

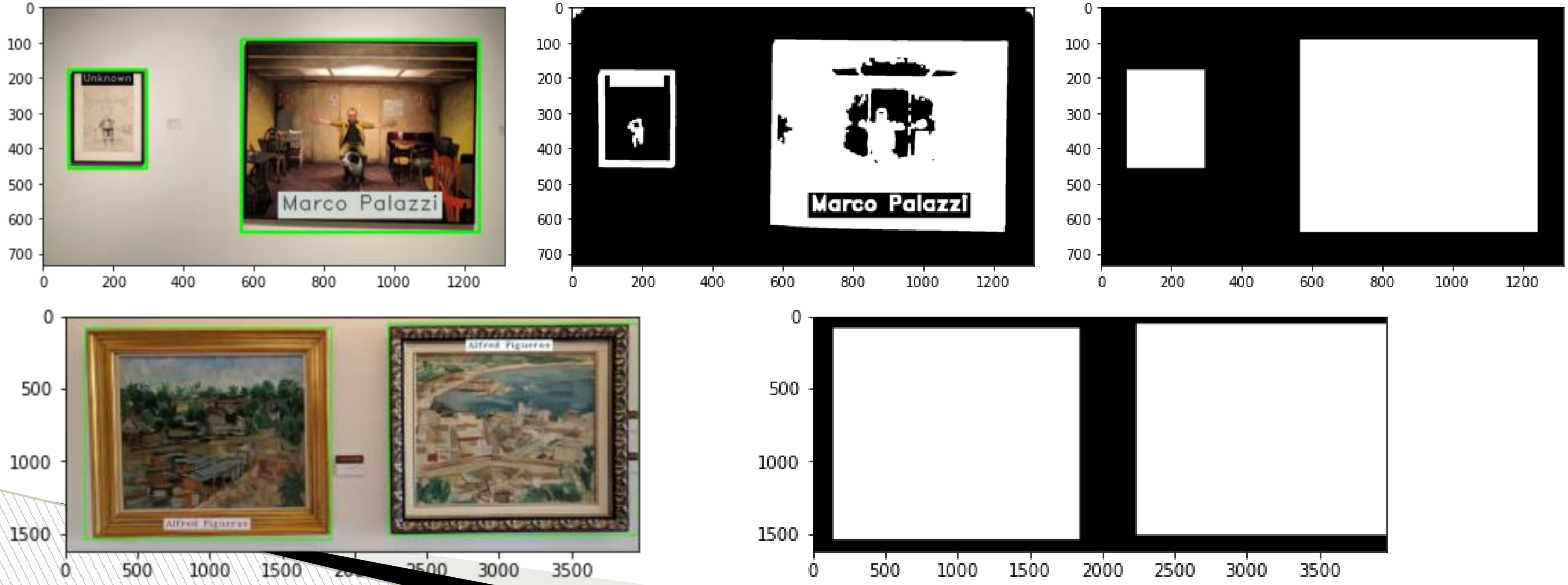
Results: detect paintings

When the morphological transformation does not separate properly the elements, the method fails. The following sample has an IOU 0.73, the illumination and shadows around the right picture make difficult to distinguish the shape/contour of the picture, and thus, the box is not so accurate.



Results: detect paintings

Qualitative Results on the QST2-W2 Dataset



Results: W2 Retrieval Pipeline

Query Image

Detect Paintings

N Crops + Text removal

?

Museum dataset

Feature Extraction + Hellinger distance calculation vs DB

Top-K Results (for each Painting)

Discussion and conclusions

- The text removal required hand-crafted thresholds, we think is tricky to find the proper combination of morphological operations and thresholding, thus the method will not generalize well.

- The single picture background removal and multi-picture background removal work properly on the dataset and does not require extensive fine-tuning. The overall quantitative results (in terms of IoU) and qualitative results are good. However, the method is weak against shadows and illumination.

- We could not test properly our retrieval pipeline due to some bugs, we assume that our methods for image preprocessing, cropping and text removal are good enough to achieve an accurate retrieval system.