Master in Computer Vision *Barcelona*

Module: M1

Project: Content Based Image Retrieval

Coordinator: Ramon Morros

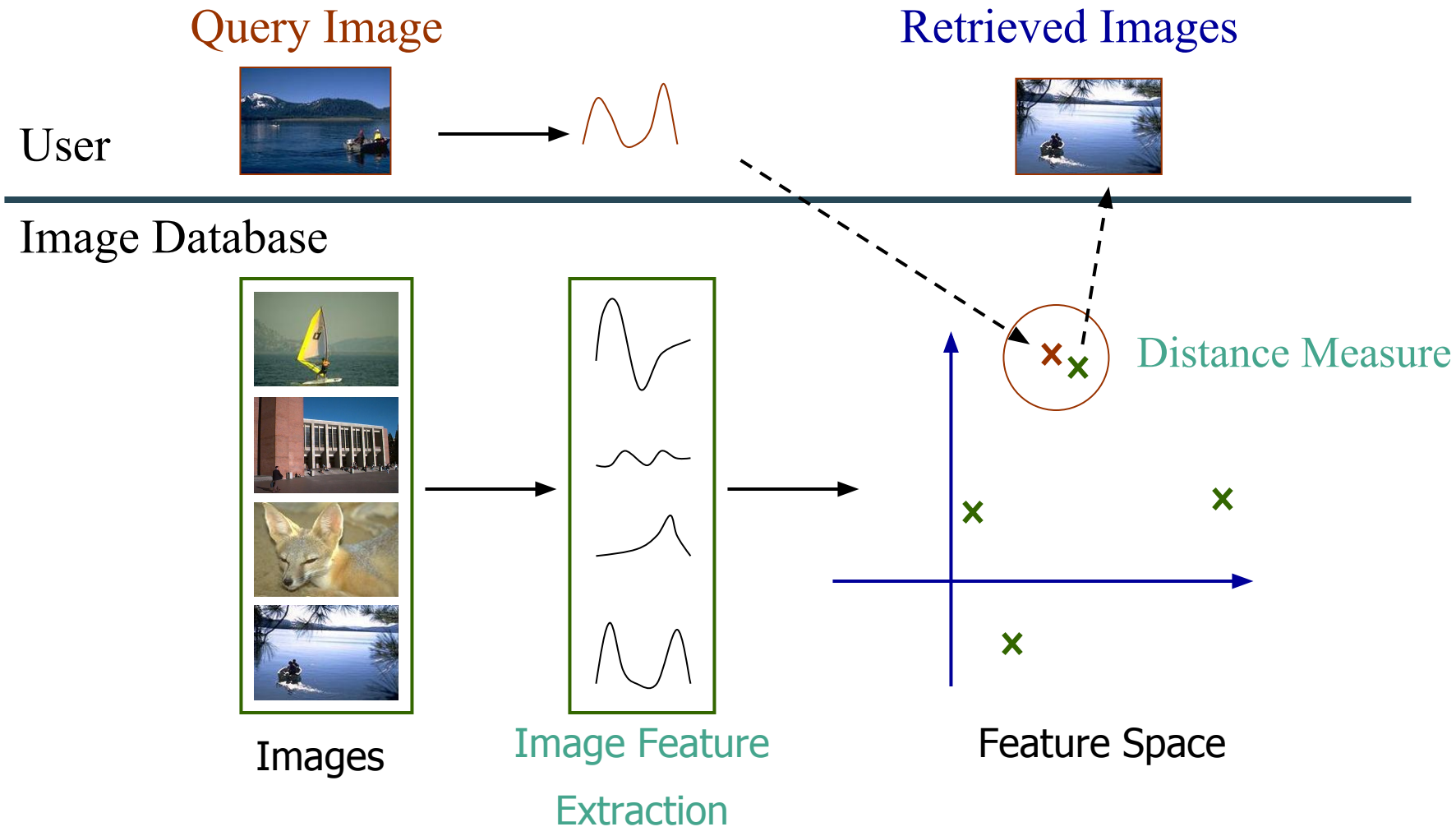
M1 – Content Based Image Retrieval

- **Main goal**
  - To learn the basic concepts and techniques to build a simple query by example retrieval system for finding paintings in a museum image collection.

Can Framis museum dataset

- **Scope**
  - Image retrieval based on color, texture, textual information
  - Image retrieval based on keypoints and local descriptors
  - Denoising, orientation correction and picture cropping
  - Morphological filters to detect and remove overlying text from images
  - Evaluation of system performance
- **Applicability**
  - Almost any (small) query by example problem

M1 – Content Based Image Retrieval



M1 – Content based image retrieval

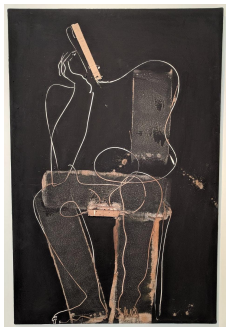
- **Goal:** search images from a large image database (DB) based on visual contents

- **Similarity:** based on visual features: color, shape texture
  - e.g. color:
    - how to characterize ‘color’?
    - how to measure ‘color similarity’?

- **CBIR steps:**
  - Index the DB: generate descriptors (e.g. histograms) for all images; this can be done offline
  - Extract features from the query image (e.g. compute histogram)
  - Compute distances between descriptors from the query image and each DB image descriptor, and order the DB images according these distances

M1 – Dataset

- **Paintings dataset**



- **Query set 1 (QS1)**: cropped images, almost no rotation



- **Query set 2 (QS2)**: uncropped image (background)

M1 – Week 1

- Given the museum and the query dataset.
  For each image in the query dataset, retrieve the K most similar images in the Museum dataset, ordered by score

- Image descriptor: image gray-level / color **histograms** (**1D**)

Task 1: Create Museum and query image descriptors (BBDD & QS1)

Task 2: Implement / compute similarity measures to compare images

Task 3: Implement retrieval system (retrieve top K results)

Task 4: Evaluation using map@k

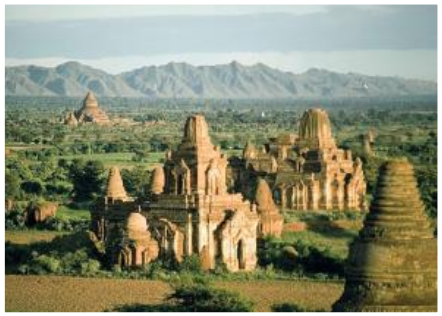
Task 5: Background removal using color (QS2). Compute descriptor on foreground (painting)

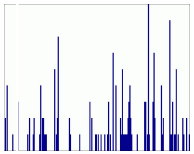
Task 6: Evaluation of picture masks and retrieval system (QS2)

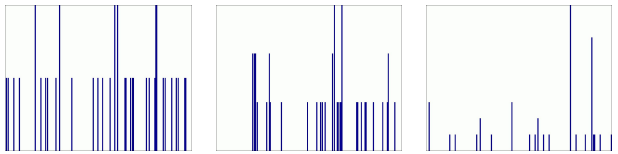
M1 – Week 1

Task1: Compute image descriptors (QS1)

- Color Histogram:
  - gray level / concatenate color component histograms
  - color space RGB, CieLab, YCbCr, HSV
  - Compulsory to use **1D** histograms!







M1 – Week 1

Task 2: similarity measures

- Euclidean distance
- L1 distance
- χ^2 distance
- Histogram intersection (similarity)
- Hellinger kernel (similarity)
- …

$$D(h_1,h_2)=\sqrt{\sum_{i=1}^{N}\left(h_1(i)-h_2(i)\right)^2}$$

$$D(h_1,h_2)=\sum_{i=1}^{N}|\,h_1(i)-h_2(i)\,|$$

$$D(h_1,h_2)=\sum_{i=1}^{N}\frac{\left(h_1(i)-h_2(i)\right)^2}{h_1(i)+h_2(i)}$$

$$I(h_1,h_2)=\sum_{i}^{N}\min(h_1(i),h_2(i))$$

$$K(h_1,h_2)=\sum_{i=1}^{N}\sqrt{h_1(i)\,h_2(i)}$$

M1 – Week 1

Task 3: For each image in QS1, compute similarities to museum images

Task 4: Evaluation

- Return the top k images (highest score, lowest distance)
- Evaluation with MAP@k (Mean Average precision at K)
  - Average precision computed for each query
  - Mean over all queries

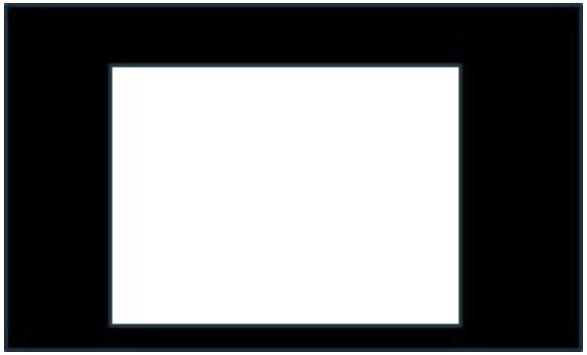
Metric available at: https://github.com/benhamner/Metrics

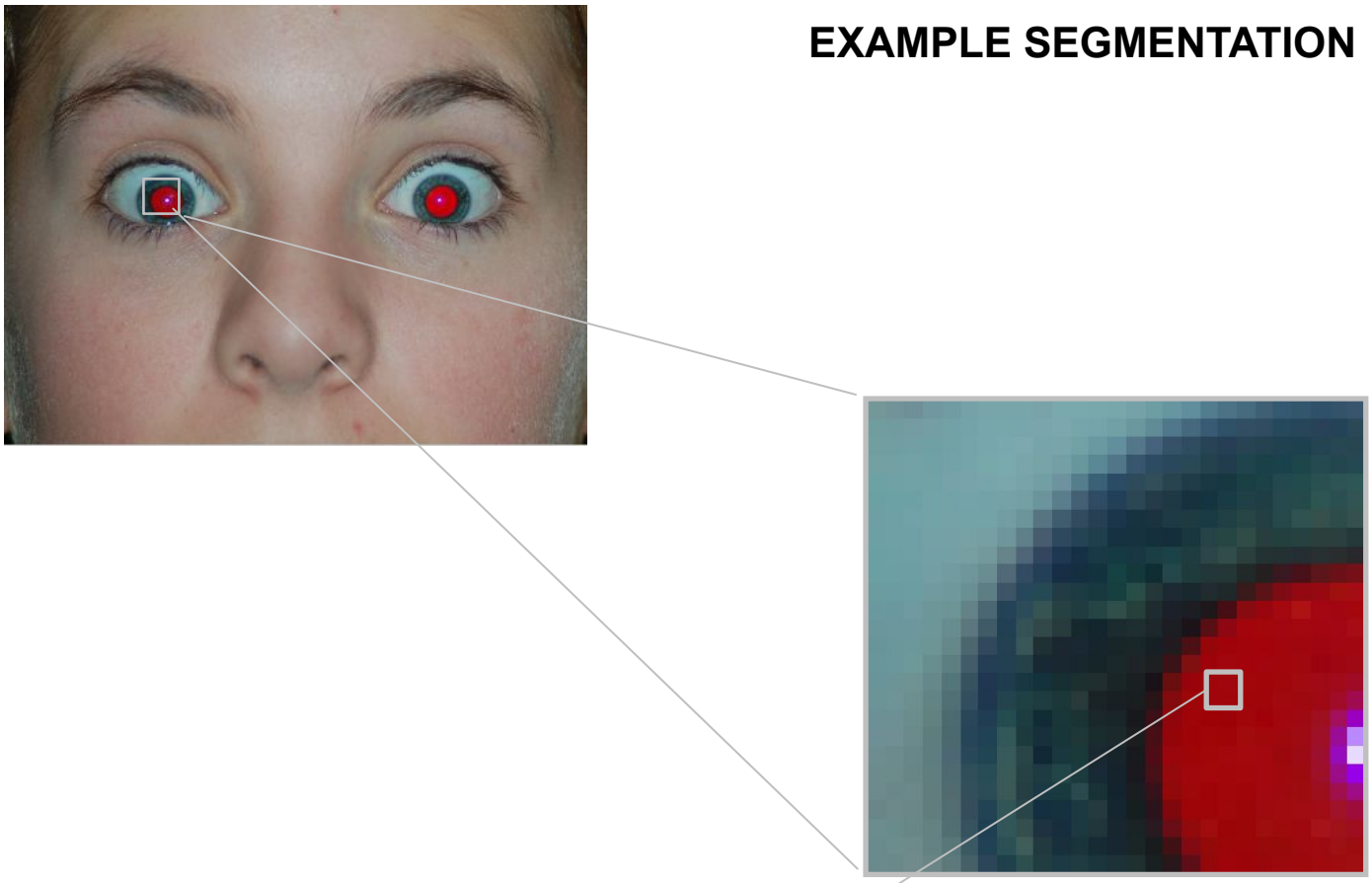
$$AP@K = \frac{\sum_{i=1}^{K} P@i}{K}$$

M1 – Week 1

Task 5: For each image in the QS2, remove background **using the background color** (e.g. model background distribution, color thresholds)

- Create a binary mask to evaluate the method
- Compute the descriptors on the foreground pixels.
- Do not use contour detectors, object detectors, etc. Just color!





Pixel value can be between 0 and 255 (in a typical 8 bits/channel image)
This pixel is (R,G,B)=(169,5,9)

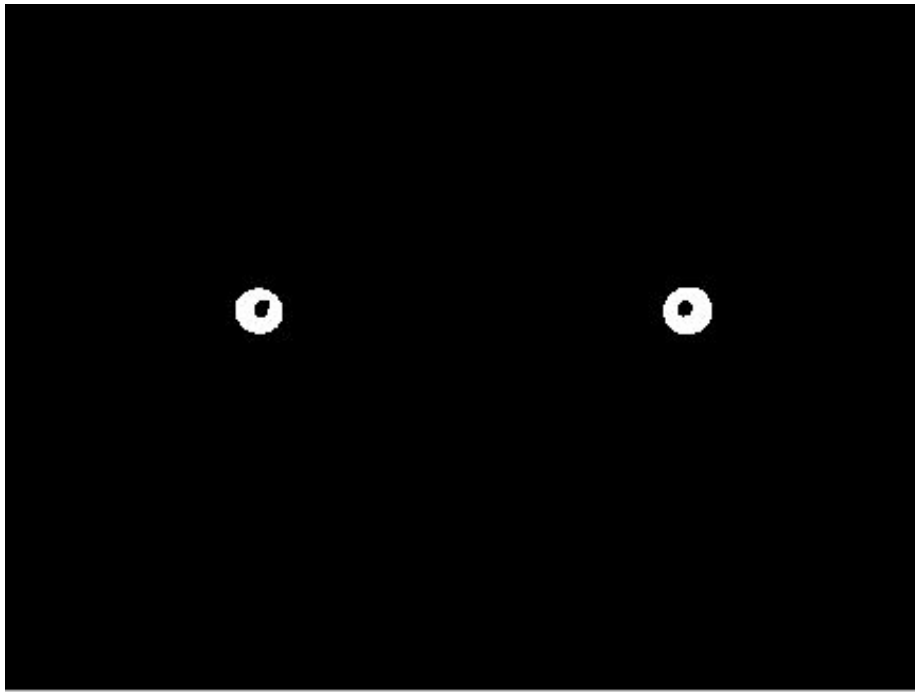
EXAMPLE SEGMENTATION

```
im = imageio.imread('BoldRedEye.jpg');

mask = im[:,:,0]>100 & im[:,:,1]<50 & im[:,:,2]<50;
```

Red channel | Green channel | Blue channel

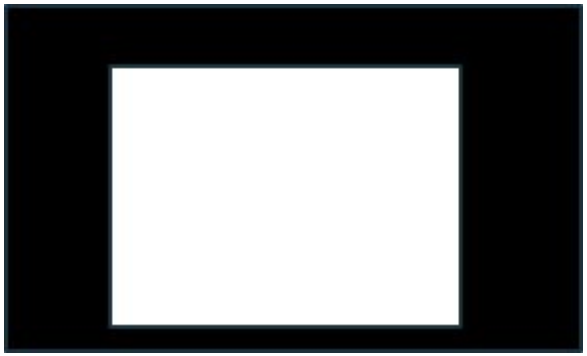
EXAMPLE SEGMENTATION



M1 – Week 1

Task 6: Evaluation

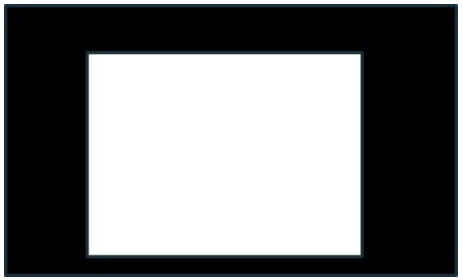
- For background removal: precision, recall, F1



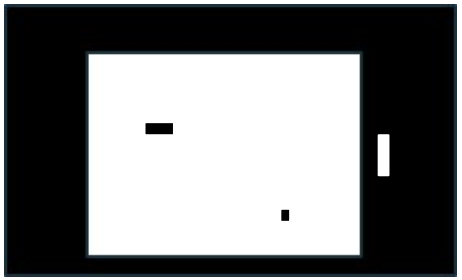
- For retrieval system: MAP@k (Mean Average precision at K)

M1 – Week 1

Mask evaluation using precision, recall and F1-measure



annotation (A)

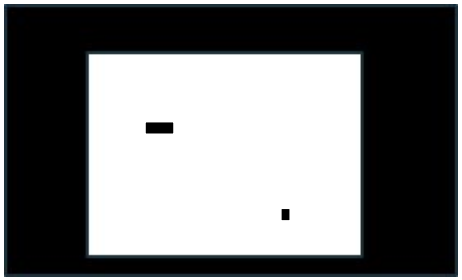


result1 (O)

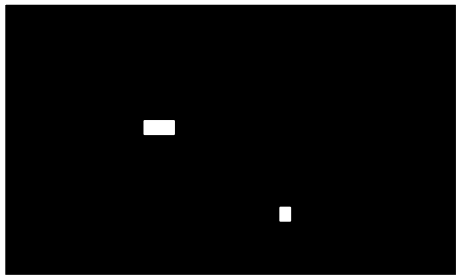
Precision $P = \frac{TP}{TP + FP}$

Recall $R = \frac{TP}{TP + FN}$

F1-measure $F_1 = 2\frac{PR}{P + R}$

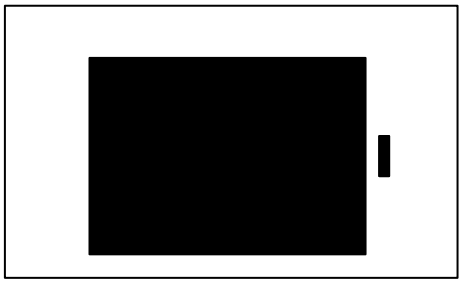


True Positive (TP) = A & O



False Negative (FN) = A & ¬O

False Positive (FP) = ¬A & O



True Negative (FP) = ¬A & ¬O

M1 – Week 1

	map@1	map@5
Method 1		
Method 2		

Provide map@k (k=1, k=5) results for the validation query sets 1&2 (QSD1 & QSD2)

	Precision	Recall	F1-mesure
Method 1			
Method 2			

Provide results for the validation query set 2 (QSD2)

M1 – Week 1

Query sets:

There will be four query sets:

- Two for development (QSD1, QSD2), already available
  - Ground truth provided: correspondences & masks
  - Results for these query sets must appear on the slides
- Two for test (QST1, QST2), released on Sunday (October 9) at 14h
  - Ground truth not provided
  - Teams submit 'blind' results for these query sets, and we score them automatically

M1 – Week 1

Submissions:

- For each test query in QST{1,2}, a list of the K=10 best results:
  - Create a python **list of lists**, with the image ids (**integer numbers**)
    - Example with 3 queries and k=2

    Query: [[q1], [q2], [q3]]

    Result: [[7,2], [76, 4], [43, 12]]

    Where 7 corresponds to image 00007.jpg, etc.

  Note: Deliver pkl files to GDrive folder:
  TeamX/week1/QST1/**method1**/result.pkl
  TeamX/week1/QST1/**method2**/result.pkl
- For QST2, the binary masks must also be submitted
  TeamX/week1/QST2/**method1**/{result.pkl, *.png}
  TeamX/week1/QST2/**method2**/{result.pkl, *.png}

- Submit progress slides
- Deadline results and slides: Sunday Oct 09 at 19:00
- Deadline questions to teams: Monday 10 Oct 2022 at 12:00