

Content-Based Image Retrieval

A series of horizontal lines in blue and white, located at the bottom right of the slide, extending from the left edge of the white area.

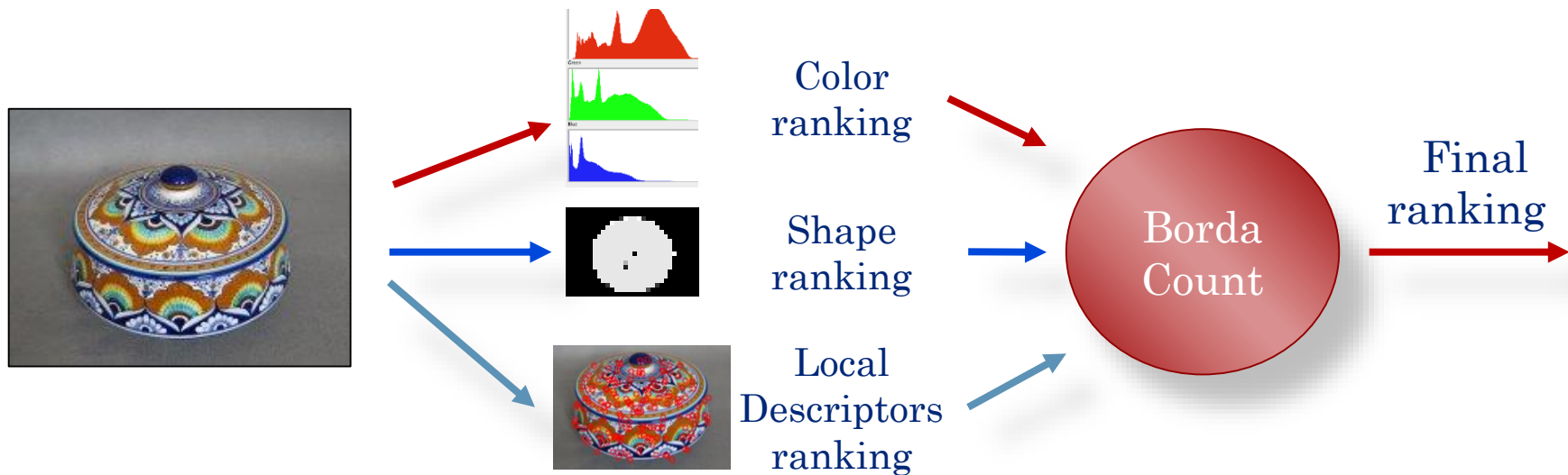
Annalisa Franco
annalisa.franco@unibo.it

Content-Based Image Retrieval

- Obiettivo dell'esercitazione è la realizzazione di un sistema per la ricerca di immagini secondo il paradigma «*query by example*».
- Il sistema opera all'interno di un database di immagini rappresentanti oggetti in ceramica dipinti con disegni tipici della tradizione.
- L'utente propone una query selezionando l'immagine di esempio desiderata e il sistema deve **visualizzare le immagini del database più simile all'esempio dell'utente**.
- **Prima di poter eseguire delle ricerche è necessario codificare le immagini del database attraverso opportune feature numeriche** che ne descrivano le caratteristiche salienti (colore, forma, texture). Anche la query dell'utente dovrà essere codificata allo stesso modo.



Content-Based Image Retrieval

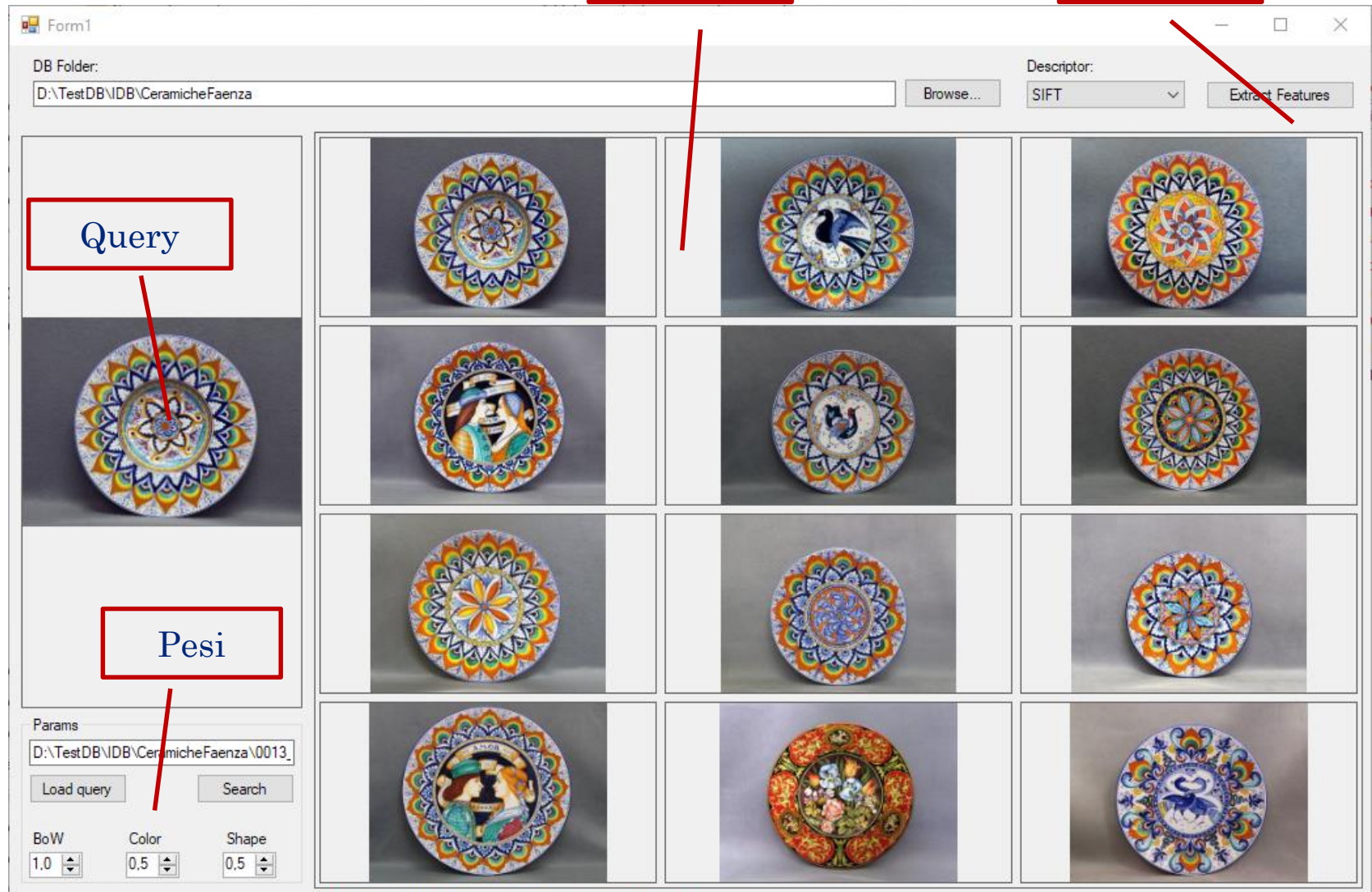


- Il metodo Borda Count si basa su un **meccanismo di voto**.
- **Ciascun tipo di feature** produce un proprio **ranking** delle immagini del database, restituendo come risultato le **K immagini più simili** alla query.
- Per il ranking finale, BC attribuisce all'immagine in posizione i nella classifica ($i \in [0, K - 1]$) un **punteggio** $s_i = K - i$.
- In questa applicazione, viene attribuito un **peso** (nel range $[0,1]$ a ciascun tipo di feature, che si usa come fattore moltiplicativo per il punteggio s_i .

L'applicazione

Risultato
(ranking)

Training



ImageRetrievalSystem

- Questa classe implementa il meccanismo di ricerca delle immagini per similarità.
- Utilizza feature extractor di diverse tipologie (`ColorFeatureExtractor`, `ShapeFeatureExtractor`, `BoWFeatureExtractor`) e ne combina i risultati utilizzando il metodo Borda Count.
- Le principali proprietà sono:
 - `WeightBoW`, `WeightColor`, `WeightShape`: pesi dei diversi tipi di feature.
- I principali metodi pubblici sono:
 - `Train(string[] dbImageFilePaths)`: metodo per l'estrazione delle feature dalle immagini del database. Produce `dbColorDescriptors`, `dbShapeDescriptors` e `dbBoWDescriptors` che vengono poi usati per le ricerche.
 - `Search(Image<Bgr, byte> query, int imagesToRetrieve)`: metodo per l'esecuzione di query di similarità.

To do: color (1)

- Implementare l'estrazione dei descrittori di **colore** nella classe `ColorFeatureExtractor`. In particolare dev'essere implementato il metodo

```
public float[] ComputeDescriptor(Image<Bgr, byte> img)
```

che prende in input un'immagine a colore e deve restituire il relativo descrittore come array di valori float.

- Il descrittore da calcolare è basato sul concetto di **color histogram**
 - Si consiglia di lavorare nello spazio colore **YCbCr**; si può utilizzare a tal fine il metodo `Convert` della classe `Image` messo a disposizione dalla libreria EmguCV. `Ycc` è il tipo corrispondente a YCbCr.
 - Il descrittore da implementare è costituito dalla concatenazione degli istogrammi colore riferiti ai canali **Cb** e **Cr** (ignorare il canale Y).
 - La libreria mette a disposizione la classe `DenseHistogram` per il calcolo dell'istogramma. Un numero di bin pari a 20 può essere sufficiente.
- Per valutare l'efficacia di un tipo di feature è bene attribuire peso 0 alle altre tipologie.

DenseHistogram (1)

▲ Syntax



C# VB C++ F#

```
public class DenseHistogram : Mat
```

Copy

The DenseHistogram type exposes the following members.

▲ Constructors

	Name	Numero di bin	Range [0,255]	Description
	<code>DenseHistogram(Int32, RangeF)</code>	↓	↙	Creates a uniform <u>1-D histogram</u> of the specified size
	<code>DenseHistogram(Int32[], RangeF[])</code>			Creates a uniform <u>multi-dimension histogram</u> of the specified size

Potete provare le due diverse versioni:

- Concatenazione di 2 istogrammi 1D
- Istogramma multidimensionale 2D

DenseHistogram (2)

C# VB C++ F#

Copy

```
public void Calculate<TDepth>(
    Image<Gray, TDepth>[] imgs,
    bool accumulate,
    Image<Gray, byte> mask
)
where TDepth : new()
```

Parameters

imgs

Type: [Emgu.CV.Image<Gray, TDepth>\[\]](#)

images to project

accumulate

Type: [System.Boolean](#)

If it is true, the histogram is not cleared in the beginning. This feature allows user to compute a single histogram from several images, or to update the histogram online.

mask

Type: [Emgu.CV.Image<Gray, Byte>](#)

Can be null if not needed. The operation mask, determines what pixels of the source images are counted

Type Parameters

TDepth

The type of depth of the image

<https://www.emgu.com/wiki/files/4.1.0/document/html/e0764906-a5b4-340e-0ed2-0dd969ff80ca.htm>

DenseHistogram (3)

Gets the bin values.

Namespace: [Emgu.CV](#)

Assembly: Emgu.CV.World (in Emgu.CV.World.dll) Version: 4.1.0.3408 (4.1.0.3408)

▲ Syntax

C#

VB

C++

F#

```
public float[] GetBinValues()
```

Return Value

Type: [Single](#)[]

The bin values

To do: shape (1)

- Implementare l'estrazione dei descrittori di **forma** nella classe `ShapeFeatureExtractor`. In particolare dev'essere implementato il metodo

```
public Image<Gray, byte> ComputeDescriptor(Image<Gray, byte> img)
```

 che prende in input un'immagine grayscale e deve restituire il relativo descrittore come immagine grayscale.
- Il descrittore da calcolare è basato sul concetto di **shape matrix**.



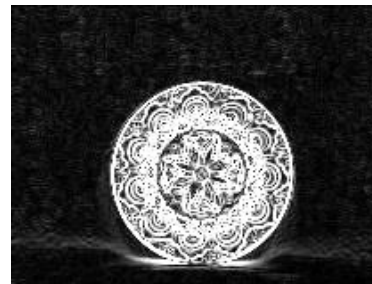
Input image



Gradiente x



Gradiente y



Modulo del
gradiente



Resize
e binarizzazione



Dilation
& Erosion

To do: shape (2)

Suggerimenti

- Per il calcolo del **gradiente** usare il metodo `Sobel` della classe `Image`

```
public Image<TColor, float> Sobel(
    int xorder, ← ordine della derivata in x
    int yorder, ← ordine della derivata in y
    int apertureSize ← dimensione del filtro (usare 3)
)
```

il metodo dovrà essere richiamato due volte per il calcolo rispettivamente del gradiente in x e di quello in y.

- Il **modulo del gradiente** può essere calcolato come segue:

$$g_{mod}(p) = \sqrt{g_x(p)^2 + g_y(p)^2}$$

- Si consiglia di **ridimensionare** l'immagine del modulo del gradiente a dimensione $w=120, h=80$ (o inferiore).
- Per la **binarizzazione** dell'immagine ridimensionata usare il metodo `ThresholdBinary` della classe `Image`.
- Gli **operatori morfologici** per la regolarizzazione finale sono già richiamati nel codice.