
Laboratorio di Immagini

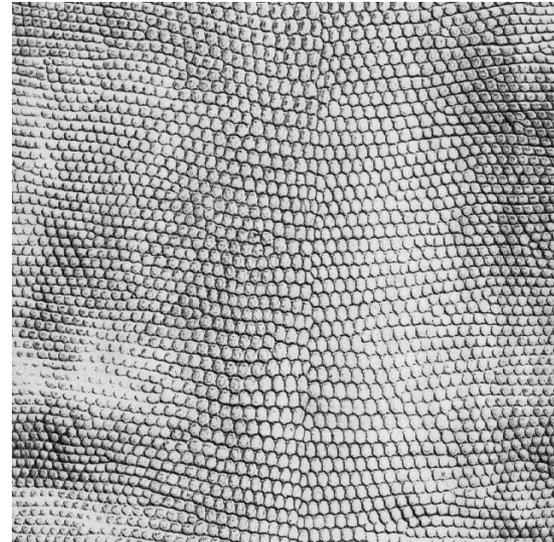
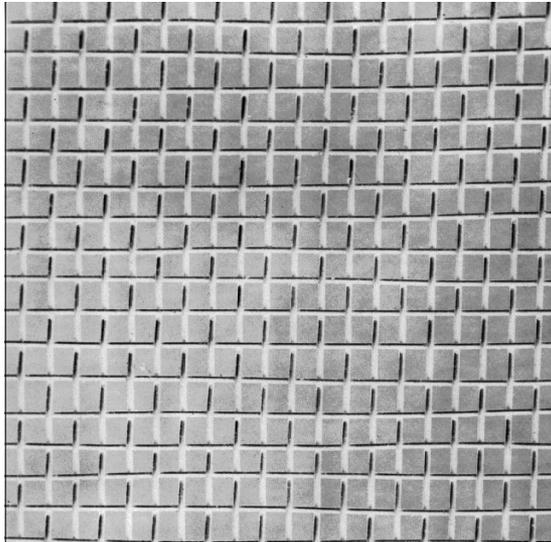
Esercitazione 7:

Texture Classification

Dr. Mauro Zucchelli
05/05/2016

Texture

- Porzione di immagine contenente pattern ripetuti
- In genere non ha valori di grigio uniformi
- Presentano molti edge



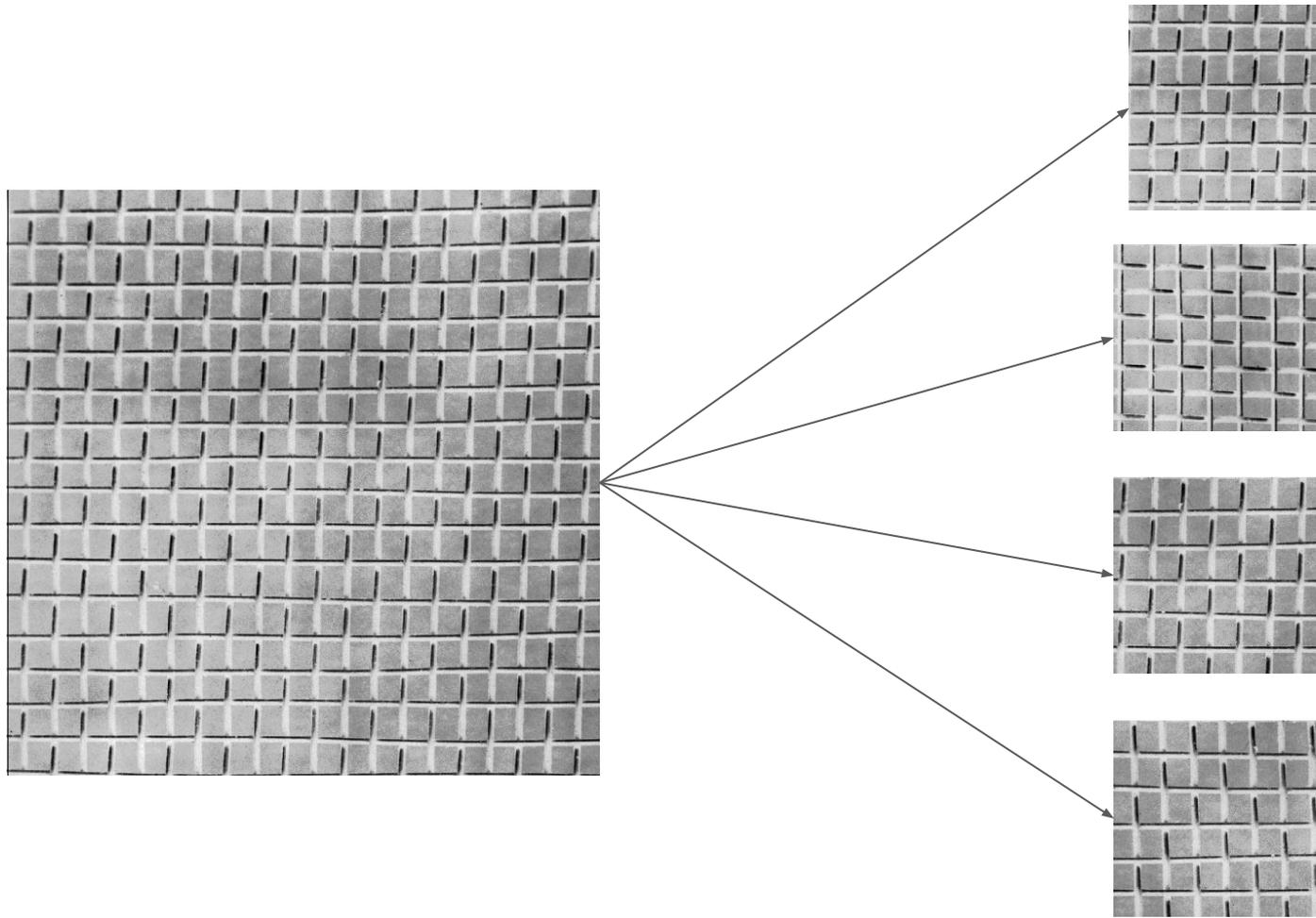
Texture classification

- Riuscire a distinguere le diverse texture
 - Data una texture decidere a che classe appartiene
 - Abbiamo bisogno di un **dataset di texture** da usare per verificare a che classe appartiene
-

Step 1: creo il dataset

- Uno dei dataset di texture più usati è il **Brodatz**
 - Ve ne ho preparato un piccolo sottoinsieme
 - Per creare il dataset posso frammentare l'immagine di ogni texture in diverse sotto-immagini (magari aggiungendo delle rotazioni) per creare il **training set** e il **testing set**
-

Step 1: creo il dataset



640x640

128x128

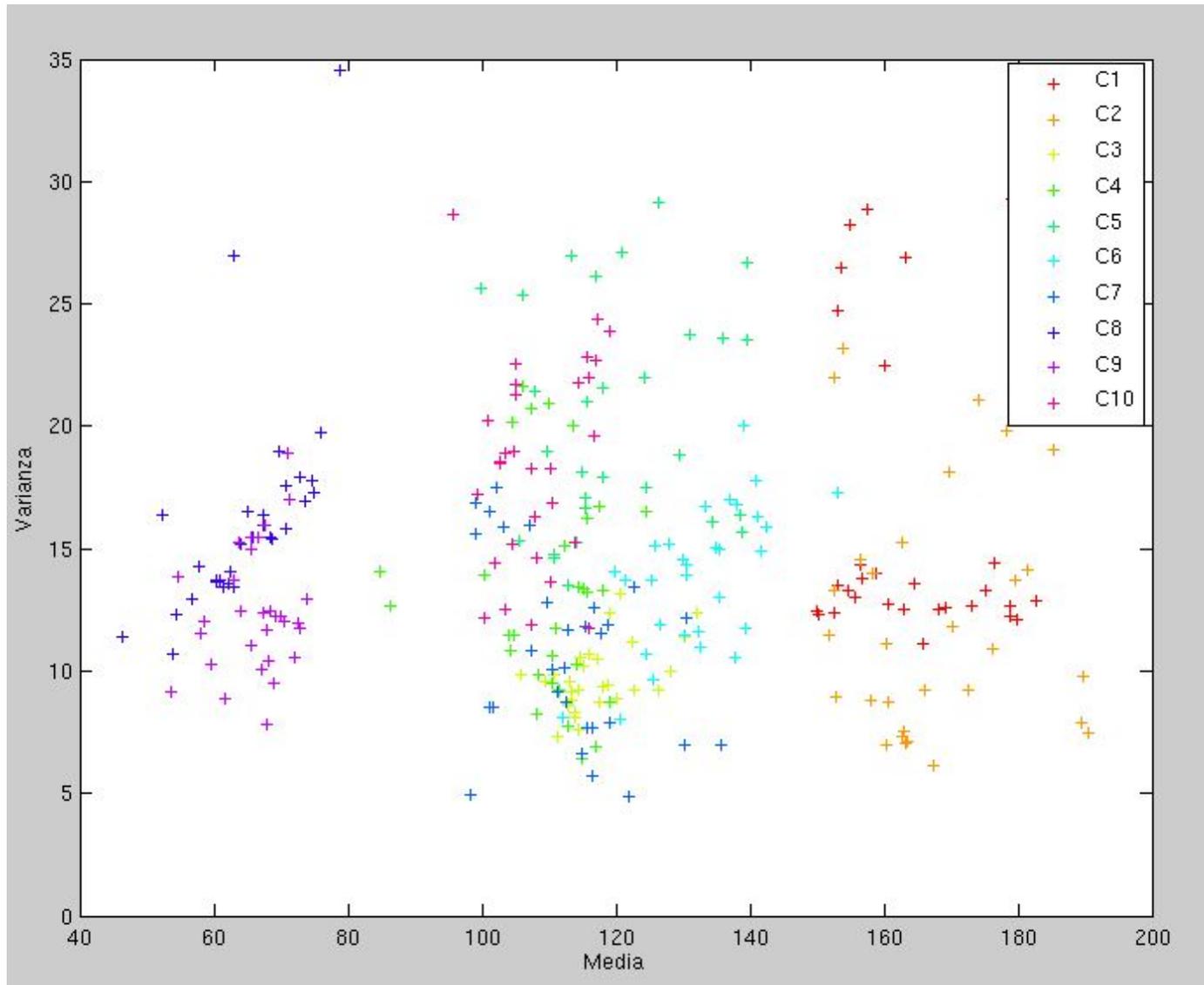
Step 1: creo il dataset

```
name_vec = {'D1.gif','D3.gif','D4.gif','D18.gif','D23.gif','D28.gif','D36.gif','D52.gif','D66.gif','D87.gif'};
n_img = length(name_vec);
img = imread(['Brodatz/',name_vec{1}]);
labels_training = zeros(30*n_img,1);
training = zeros(30*n_img,128,128);
c=1;
for i=1:n_img;
    img = imread(['Brodatz/',name_vec{i}]);
    for j=1:30
        training(c,:,:) = extract_random_subregion(img, 128, 128);
        labels_training(c) = i;
        c=c+1;
    end
end
labels_testing = zeros(10*n_img,1);
testing = zeros(10*n_img,128,128);
c=1;
for i=1:n_img;
    img = imread(['Brodatz/',name_vec{i}]);
    for j=1:10
        testing(c,:,:) = extract_random_subregion(img, 128, 128);
        labels_testing(c) = i;
        c=c+1;
    end
end
```

Step 2: estraggo le feature

- Dobbiamo riuscire ad estrarre delle feature numeriche da ogni immagine
 - Esempio:
 - Media
 - Varianza
-

Step 2: estraggo le feature



Step 2: estraggo le feature

```
f_training = zeros(30*n_img,2);

for i=1:30*n_img;
    f_training(i,1) = mean(mean(squeeze(training(i,:,:))));
    f_training(i,2) = std(mean(squeeze(training(i,:,:))));
end

f_testing = zeros(10*n_img,2);
for i=1:10*n_img;
    f_testing(i,1) = mean(mean(squeeze(testing(i,:,:))));
    f_testing(i,2) = std(mean(squeeze(testing(i,:,:))));
end

colors= hsv(n_img);
figure;
for i=1:n_img;
    plot(f_training(((i-1)*30+1):((i-1)*30+30),1),f_training(((i-1)*30+1):((i-1)*30+30),2),'+', 'Color',colors(i,:))
    hold on
end
legend('C1','C2','C3','C4','C5','C6','C7','C8','C9','C10');
xlabel('Media')
ylabel('Varianza')
```

Step 3: classifico

- Possiamo usare un classificatore a piacere
 - Il K-NN è sempre un'ottima scelta per iniziare!
 - Calcoliamo l'errore
-

Step 3: classifico

```
labels_knn = my_knn(f_training,labels_training,f_testing,5);
```

```
error = sum(labels_testing~=labels_knn')
```

Step 3: classifico

```
labels_knn = my_knn(f_training,labels_training,f_testing,5);
```

```
error = sum(labels_testing~=labels_knn')
```

```
error =
```

```
36
```

Matrice di confusione

textures	1	2	3	4	5	6	7	8	9	10	% correct	
1	841	0	0	0	0	0	0	0	0	0	100.00%	
2	0	840	1	0	0	0	0	0	0	0	99.88%	
3	2	0	839	0	0	0	0	0	0	0	99.76%	
4	0	0	0	841	0	0	0	0	0	0	100.00%	
5	0	0	88	0	753	0	0	0	0	0	89.54%	
6	0	0	134	0	0	707	0	0	0	0	84.07%	
7	0	66	284	0	0	0	491	0	0	0	58.38%	
8	0	0	58	0	0	0	0	783	0	0	93.10%	
9	0	0	71	0	0	0	0	0	770	0	91.56%	
10	0	4	4	0	0	0	0	0	0	833	99.05%	
				Average recognition rate								91.53%

Istogrammi come feature

- Media e varianza non danno buoni risultati
 - Due feature non sembrano portare a dei buoni risultati
 - Proviamo ad usare più features
 - Istogrammi a N bin
 - Gli istogrammi di matlab di default non vanno bene
-

my_histogram.m

```
function [b,r] = my_histogram(img,n)
```

```
X = double(img(:));
```

```
max(X)
```

```
min(X)
```

```
l = linspace(0,255,n+1);
```

```
b = round((l(1:n)+l(2:n+1))/2);
```

```
X = ceil(X/255.0*(n-1));
```

```
max(X)
```

```
min(X)
```

```
r = zeros(n,1);
```

```
for i=1:n
```

```
    r(i) = sum(X==(i-1));
```

```
end
```

Istogrammi come feature

- Provate a ripetere la classificazione usando gli istogrammi come features!
- Provate a cambiare il numero di bin
 - 5, 10, 20,

Istogrammi come feature

- Il risultato migliora notevolmente!
- Riprovate la matrice di confusione

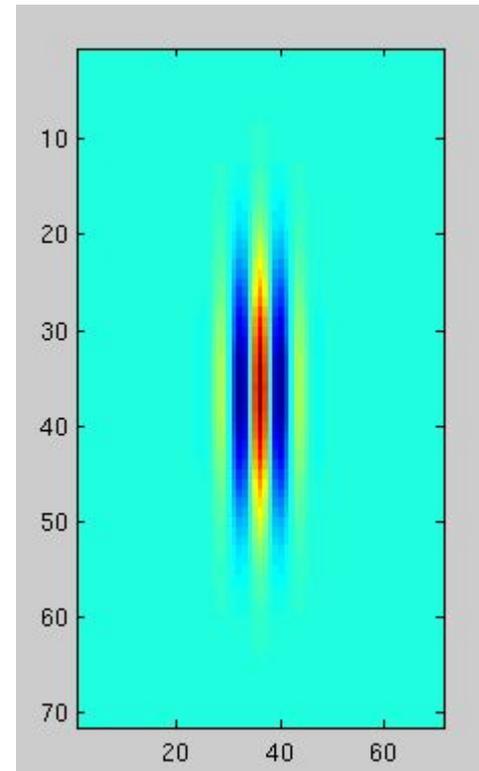
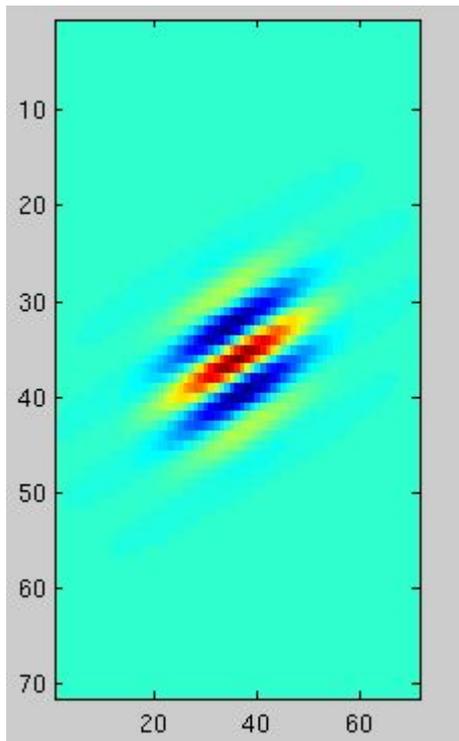
```
error =
```

```
6
```

Funzioni di Gabor

- I filtri più usati per l'analisi di texture
 - Gaussiana moltiplicata per un seno (o un coseno)
 - Simili ai filtri di sobel ma con più gradi di libertà
 - Scala
 - angolo di rotazione
-

Funzioni di Gabor



Funzioni di Gabor

```
lambda = 8;
theta = 0;
psi = [0 pi/2];
gamma = 0.5;
bw = 1;
N = 5;
img1 = squeeze(training(1,:,:));

figure
subplot(1,1,1)
imshow(img1,[])

for n=1:N
    gb = gabor_fn(bw,gamma,psi(1),lambda,theta) + 1i * gabor_fn(bw,gamma,psi(2),lambda,theta);

    f = imfilter(img1, gb, 'symmetric');

    theta = theta + 2*pi/N;

    f2 = abs(f);
    colormap jet;
    figure
    subplot(1,3,1)
    imagesc(real(gb))
    subplot(1,3,2)
    imagesc(imag(gb))
    subplot(1,3,3)
    imagesc(f2)
end
```

Filtri di Gabor come features

- Dai risultati dei filtraggi possiamo estrarre delle feature
 - Provate ad utilizzare la **media** o la **varianza** di ogni filtro come feature per la classificazione!
 - Quale funziona meglio?
 - Ed usandole entrambe?
-

FINE!

The End
