

# Relazione progetto ESI

Fabiola Fabretti VR437439  
Cesare Montresor VR437525

## Obiettivi

L'elaborato si propone di riuscire a rilevare difetti in una tessitura, evidenziandone la posizione e la dimensione direttamente sull'immagine senza bisogno di input da parte dell'utente.

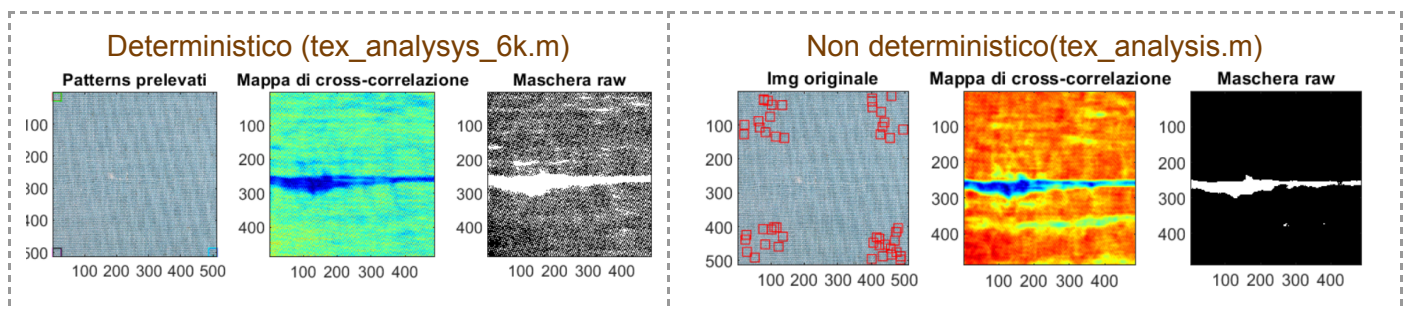
## Dataset

Le immagini utilizzate per costruire lo script sono state prelevate dai seguenti dataset, reperibili online:

- [msminhas93/FabricDefect: A weakly annotated fabric defect dataset.](#)
- [AITEK FABRIC IMAGE DATABASE](#) (convertito in JPEG, scremato e ritagliato)

## Algoritmo

Abbiamo generato due versioni dell'algoritmo: una deterministica, che prende sempre gli stessi kernel, e una che invece prende un numero casuale e molto alto di kernel. Lo scopo è quello di filtrare via rumore o eventuali kernel presi sul difetto, e accentuare la mappa di cross-correlazione facendo la media di tutti i risultati.



L'algoritmo a cui siamo giunti - dopo numerosi tentativi - è il seguente:

1. **Caricamento** del file da analizzare
2. **Individuazione della dimensione ottimale** (probabile) **dei kernel** da usare la cross-correlazione secondo la funzione `find_pattern_size`.
3. **Creazione due mappe di correlazione**, usando dimensione di kernel che massimizza **Correlazione** e un'altra per il **Contrasto**
  - a. **VERSIONE NON DETERMINISTICA**
    - i. Estrazione di **pattern random** vicino agli angoli dell'immagine
    - ii. Salvataggio delle coordinate estratte
    - iii. Calcolo **cross-correlazione** e **media con le precedenti**
  - b. **VERSIONE DETERMINISTICA**
    - i. Calcolo crosscorrelazione
  - c. **Individuazione della soglia ottimale con il metodo di Otsu** (`greythresh`) e creazione della maschera
  - d. Pulizia della maschera grezza usando `imopen`, `imclose` e `bwareaopen`.

#### 4. Scelta della maschera ottimale:

- Di default, prende CORR.
- Se CONT ha un numero di aree contigue minore, o se ha lo stesso numero di aree contigue con percentuale di selezione maggiore, prendo CONT.
- Se sia CONT che CORR non portano risultati soddisfacenti (  $<0.1\%$  dei pixel ), ovvero non selezionano nulla, avviamo lo script “di riserva” **gabor\_emergency**, che **utilizza un filtraggio di Gabor per provare ad approssimare le zone non coerenti**.

#### 5. Applicazione della maschera all'immagine

#### 6. Visualizzazione/salvataggio delle immagini richieste dalle impostazioni.

Le varianti precedenti, poi si sono evolute in questa, sono state:

- Filtraggio in **frequenza** con filtro **gaussiano**.
- Usare **N kernel di dimensione variabile**.
- Usare come dimensione di kernel l'**ultimo picco della caratteristica** di Correlazione della matrice di **co-occorrenza**
- Creare anche una **mappa usando il valore di Contrasto** e **scegliere** quale delle due
- Aggiungere la funzione **gabor\_emergency** come ultimo tentativo di soluzione se le due maschere precedenti sono vuote.
- Estrazione **N kernel random per ciascuna dimensione** (anziché i 6 standard agli angoli)
- Sampling casuale di pattern di **dimensioni variabili**, con selezione dei pattern con cross correlazione media **contro se stessi** più alta.

## Struttura dei files

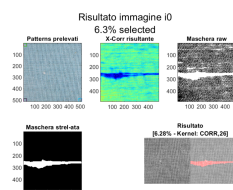
Il progetto è suddiviso nei seguenti file:

**tex\_analysis.m**

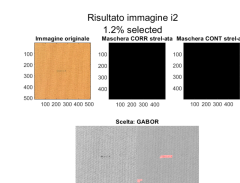
È il main del progetto, e contiene la maggior parte del codice che implementa l'algoritmo sopracitato.

Contiene una serie di impostazioni per personalizzare l'esecuzione:

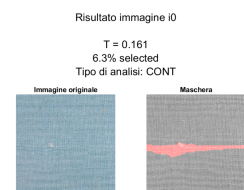
- analyze\_just\_one**: permette di analizzare una sola immagine o tutta la cartella in batch
- rand\_image** e **unrand\_number** permettono di scegliere un'immagine specifica o lasciare la scelta al caso
- show\_resume**, **show\_resume\_choice**, **show\_result** sono tre diverse visualizzazioni del risultato:



**show\_resume**



**show\_resume\_choice**



**show\_result**

**is\_reliable.m**

Prende una maschera e l'immagine a cui è riferita, e ritorna il **numero di componenti connesse** e la **percentuale di pixels selezionati** - ovvero i parametri che usiamo per determinare grossolanamente quale delle due maschere generate è più *reliable*.

*gabor\_emergency.m*

E' l'algoritmo di emergenza, che viene attivato solamente se l'algoritmo "tipico" non produce alcun risultato. La logica dell'algoritmo è la seguente:

- Per ciascuna orientazione (0,90) genera una mappa calcolata come **media di gaborfilter di magnitudo da 2 a 30**.
- Genera la maschera utilizzando il **metodo di Otsu**, con una soglia alzata artificialmente in quanto abbiamo notato che tendeva a essere più bassa del dovuto.
- Guarda ciascuna maschera ottenuta e **sceglie** quella probabilmente corretta con un'euristica identica a quella usata nel file main per la scelta fra CONT e CORR.
- **Ritorna la maschera e la mappa risultante dal filtro.**

*find\_pattern\_size.m*

Presa un'immagine, ritorna due valori ideali di dimensione di kernel - uno che massimizza localmente la proprietà correlation e uno contrast della matrice di co-occurrence.

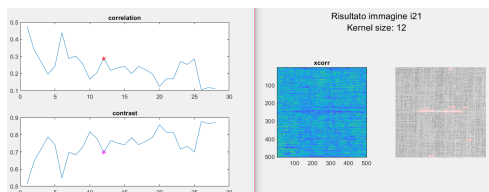
- In un ciclo, **greycomatrix** calcola la matrice di co-occorrenza fra un pixel e il suo *i*-esimo vicino, creando un array di valori per ciascun *i* da 5 a 30.
- **graycoprops** estrae delle caratteristiche (nel nostro caso correlazione e contrasto) per ciascuno di questi *i*.
- Si restituisce l'ultimo picco per la relazione di correlazione (intuitivamente sensata) e di contrasto (che abbiamo notato essere, a volte, più corretta osservando l'andamento con **observe\_kernelsize**).

*fileloader.m*

Semplice script che carica il file desiderato dati i flag relativi ai files.

Abbiamo incluso anche le seguenti funzioni, che non sono utilizzate direttamente all'interno del progetto ma ci sono tornate molto utili per capire come la modifica dei parametri influenzasse il risultato.

*observer\_kernelsize.m*

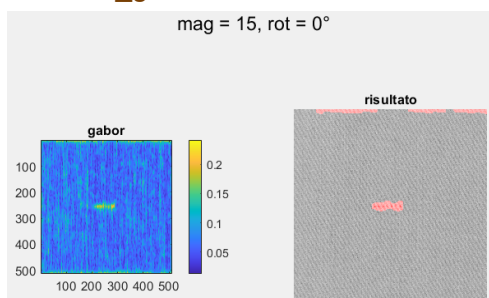


Apri una schermata con l'andamento dei parametri di **greycomatrix** al variare dell'offset.

In un'altra finestra, mostra (animazione) come cambia la maschera al variare della dimensione del kernel.

E' risultata utile in quanto ha permesso di intuire, empiricamente, che sembra esserci una correlazione fra le maschere accettabili e i massimi di questi due valori.

*observer\_gaborfilter.m*



Lavora analogamente a **gabor\_emergency** per generare una maschera, ma mostra la generazione iterativa delle due mappe.

Alla fine fa vedere quale delle due verrebbe selezionata usando i criteri di euristica.

## Testing dell'efficacia

Per cercare di capire quale soluzione potesse essere la migliore, abbiamo scritto sviluppato diverse soluzioni e le abbiamo confrontate scelto che sembrava adattarsi meglio ad un maggior numero di varianti.

Abbiamo riportato i risultati delle interazioni più significative su un foglio Excel.

Per esempio, la prima colonna rappresenta una delle prime versioni del codice originale, che già include una prima forma di thresholding automatico sulla mappa di correlazione.

Le colonne successive sono varie tecniche con le quali abbiamo cercato di migliorare.

IMG	Solo corr	Tanti sample random	Autochoose cont/corr	Autochoose new parameters	Gabor filter	Autochoose + sample random
i0	1	1	1	1		1
i1	0	.5	.5	1		1
i2	0	0	0	0	→ 1	1
i3	0	0	0	0		0
i4	1	1	1	1		1
i5	1	.5	1	1		1
i6	1	1	1	1		1
i7	1	1	1	1		1
i8	.5	1	.5	1		1
i9	0	1	1	1		1
i10	0	1	1	1		1
i11	1	1	1	1		1
i12	1	1	1	1		1
i13	1	.5	1	1		.5
i14	.5	1	.5	1		1
i16	0	0	0	0	→ 0	0
i17	1	1	1	1		1
i18	1	1	1	1		1
i19	1	1	1	1		1
i20	1	1	1	1		1
i21	0	.5	0	0		.5
i22	1	0	1	1		1
i23	1	.5	1	1		1
i24	1	1	1	1		1
i25	.5	.5	0	0		1
i26	.5	.5	1	1		1
i28	.5	1	1	1		1
i29	1	1	.5	1		1
i30	1	1	1	1		1
i31	0	0	0	0	→ 1	1
i32	1	1	1	1		1
i33	.5	0	.5	1		.5
i34	0	1	0	0		0
i35	0	1	0	0		1
i36	0	0	0	0		0
i38	.5	.5	0	0		1
i80	0	0	0	0	→ 1	.5
i81	0	0	1	1		0
i83	0	0	0	0		0
i84	0	0	0	0		0
i85	.5	0	0	0		0
i86	0	.5	0	0		.5
i87	0	1	0	0		.5
i88	1	1	1	1		1
i89	1	1	1	1		1
i90	1	1	1	1		0
i91	0	0	0	0	→ 0	1

1 = Accettabile o buono  
.5 = Quasi accettabile  
0 = Non accettabile

<b>Solo correlazione</b>	<b>Tanti sample random</b>
corretti: 21	corretti: 25
fixable: 8	fixable: 9
<b>Autochoose corr/cont</b>	<b>Bonus: Gabor</b>
corretti: 25	+ 3
fixable: 5	
<b>Versione finale D</b>	<b>Versione finale ND</b>
Autochoose, gabor, parametri migliorati	Versione finale D + random sampling (seed 1)
corretti: 33	corretti: 32
fixable: 0	fixable: 6