

# Guida all'utilizzo del codice

Di seguito è descritto come andrebbe eseguito il codice. È opportuno seguire l'ordine, in quanto tra le varie fasi ci sono delle dipendenze, in termini di immagini/file che vengono creati in una fase per poi poter essere utilizzati nelle successive.

Tutti gli script sono comunque suddivisi in paragrafi e commentati per una maggiore leggibilità del codice, e gli stessi nomi delle variabili sono stati scelti in maniera tale da essere il più intuitivi possibile riguardo alla funzione svolta.

\* Nei vari file vanno modificati i path di accesso alle immagini, essendo assoluti, che comunque compaiono solo all'inizio degli script.

## 1. Data Augmentation (cartella: Pupil\_Resizing)

- Il file *resizePupil.m* è il primo da eseguire
- Prende in input il nome di un file (di cui viene passato il path assoluto) e inizialmente consente la creazione delle maschere di verità dell'immagine originale (se non ancora generate) e successivamente esegue automaticamente le operazioni di data augmentation
- Ciascuna immagine generata viene salvata nella stessa cartella dell'immagine originale. Il nome assegnato a ciascuna di esse sarà rappresentativo delle caratteristiche dell'immagine generata (es. : *imageName-1.3-down1.5* indica che l'immagine in questione rappresenta una modifica dell'immagine originale denominata *imageName*, con scale overall pari a 1.3 e scaling verso il basso di 1.5)
- Per ciascuna immagine generata viene allo stesso tempo salvato, in una sottocartella, un file di dati contenente i parametri di scaling per l'immagine originale, utilizzati per calcolare la maschera attesa, per i contorni di iride e pupilla, utilizzata successivamente nel file *errorDetection.m* per analizzare la bontà dell'algoritmo di segmentazione

- Tutti gli altri file presenti nella cartella sono funzioni di supporto utilizzate in *resizePupil.m*

## 2. Error Detection

- *errorDetection.m* prende in input un'intera cartella di immagini ottenute attraverso l'operazione precedente, e valuta l'algoritmo di segmentazione di iride e pupilla, confrontando la sua rilevazione con quella attesa (ovvero quella ottenuta a partire dagli *scalingParams* salvati precedentemente)
- Tutte le rilevazioni vengono automaticamente salvate in un file excel
- Le distanze tra le rilevazioni vengono calcolate attraverso la funzione *masksDistance.m*
- (La prima esecuzione richiederà più tempo di esecuzione per la creazione dei parametri di Hough dell'algoritmo di segmentazione, che verranno poi salvati in una sotto-cartella per un caricamento più veloce nelle successive esecuzioni)

### 3. Template Coding

- Il file *templateCoding.m* prende in input una cartella di immagini ottenute attraverso l'operazione di Data Augmentation, ed effettua un processo iterativo di matching.
- Per ciascuna immagine viene generato un template (salvato in una sotto-cartella) confrontato con il template appartenente all'ultima variazione di scaling overall.
- Tutti i dati ottenuti vengono salvati automaticamente in un file excel, per poter effettuare operazioni di visualizzazione
- Il file *templateCodingBoundaries.m* serve ad analizzare l'efficienza del matching al crescere del parametro di scaling della pupilla