

FEI – Traccia dell'esercitazione di laboratorio 05

[N.B. Si suppone di partire da un progetto con le modifiche e aggiunte richieste nelle esercitazioni precedenti correttamente completate]

- 1) Completare l'implementazione della classe `ConvoluzioneByteInt` che esegue l'operazione di convoluzione utilizzando il filtro "linearizzato" e gli offset precalcolati. L'operazione deve restituire un'immagine di tipo `Image<int>`. Suggerimenti:
 - Fare riferimento alle dispense per la descrizione della convoluzione e dell'utilizzo del filtro "linearizzato" con offset precalcolati;
 - Esaminare i membri della classe `ConvolutionFilter<T>`;
 - Per semplicità, ignorare la proprietà `AdditionalBorder`.Per verificare la correttezza del codice implementato, confrontare i risultati ottenuti con la corrispondente classe della libreria (`ByteToIntConvolution`).
- 2) Creare una nuova classe (derivata da `ImageOperation`) che permetta di effettuare lo smoothing (vedi dispense) di un'immagine grayscale. La classe dovrà avere un parametro (proprietà con apposito attributo) di tipo intero (la dimensione del filtro di smoothing).
- 3) Creare una nuova classe (derivata da `ImageOperation`) che permetta di effettuare lo sharpening (vedi dispense) di un'immagine grayscale. La classe dovrà avere un parametro (proprietà con apposito attributo) di tipo `double` compreso fra 0.0 e 1.0 (il peso della risposta al filtro di sharpening).
- 4) *[Facoltativo]* Esaminare come è stata realizzata l'interfaccia utente personalizzata per modificare i parametri della classe `ConvoluzioneByteInt`, in particolare l'uso dell'attributo `CustomAlgorithmPreviewOutput` e lo `UserControl SimpleConvolutionParameterControl`. In modo analogo, personalizzare la visualizzazione e modifica dei parametri della classe implementata al punto precedente. A tal fine creare un nuovo `UserControl` che implementi l'interfaccia `IAlgorithmPreviewParameters` e contenga appositi controlli (ad esempio una `TrackBar` o un `NumericUpDown`).