1) Descrivere il funzionamento degli operatori di Sobel ed elencarne alcune possibili applicazioni.

# Fondamenti di Elaborazione di Immagini Matricola:

**Prova del 27-Gen-2012** (90 minuti) **Cognome e Nome:**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

2) Che cos’è la “Hit-or-Miss transform”? A cosa può servire?

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

3) Descrivere il funzionamento dell’algoritmo di etichettatura delle componenti connesse.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

4) Basandosi sulla libreria di classi utilizzata nel corso, implementare in C# una classe “Binarizza”, derivata da ImageOperation<Image<byte>, Image<byte>>, che binarizzi un’immagine grayscale utilizzando come soglia locale (ossia specifica per ciascun pixel) il valore corrispondente al minimo fra: 1) la media dei livelli di grigio in un intorno di 5x5 pixel e 2) la media dei livelli di grigio dell’intera immagine. In prossimità dei bordi, dove non tutto l’intorno è disponibile, considerare il livello di grigio dei pixel mancanti pari a zero. Le classi della libreria riportate nel diagramma seguente possono essere utilizzate (senza doverle re-implementare) per semplificare lo svolgimento dell’esercizio.



5) Basandosi sulla libreria di classi utilizzata durante il corso, implementare in C# un metodo (non è necessario implementare la classe che lo contiene) che riceva in input un’immagine grayscale e che: i) la binarizzi utilizzando la classe implementata nell’esercizio precedente; ii) conti il numero di componenti connesse (considerando il foreground pari a 255) con perimetro inferiore a metà della propria area.

Le classi della libreria riportate nel diagramma seguente possono essere utilizzate (senza doverle re-implementare) per semplificare lo svolgimento dell’esercizio.

