

Fantesini Alessandro 1884107

-Box Blur Dopo un primo approccio con la blur gaussiana (xblursigma e yblursigma nei params), ho implementato una box blur a kernel dinamico che viene applicata 3 volte consecutive per simulare con costo computazionale minore una gaussiana classica. Il parametro int boxblur rappresenta il sigma (intensità della sfocatura).

```
(out/piazza_del_popolo_04.jpg)
(out/faro_tramonto_04.jpg)
(out/zarautz_skate_04.jpg)
```

-Sobel Edge Detection L'ispirazione di questo filtro (input per poi il Canny Edge Detector) e per l'Edge Degrees To Color l'ho avuta da questo video su YouTube <https://www.youtube.com/watch?v=uihBwtPIBxM&t=1s>. Applico prima una Box Blur con matrice 3x3, poi con 2 kernel sempre 3x3 mi trovo i valori di Gx e Gy (risultati sul pixel dei kernel) con i quali posso sia trovarmi l'intensità dell'Edge (il valore G), sia il suo angolo tramite l'arcotangente. Il parametro float edgeDetection deve essere diverso da 0 per applicare questo filtro.

```
out/piazza_del_popolo_01.jpg
out/faro_tramonto_01.jpg
out/zarautz_skate_01.jpg
```

-Edge Degrees To Color Sempre ispirato dal video di YouTube precedentemente linkato, questa funzione converte l'angolo dell'edge precedentemente calcolato in un colore tramite la ruota dei colori. Il parametro edgeColor attiva questo filtro, mentre il parametro edgeDetection indica la soglia minima dell'intensità dell'edge affinché sia convertito in colore. L'immagine di test rende meglio l'idea della conversione.

```
out/test_01.jpg

out/piazza_del_popolo_02.jpg
out/faro_tramonto_02.jpg
out/zarautz_skate_02.jpg
```

-Canny Edge Detector Prendendo l'output di Sobol, come prima cosa passo l'immagine per una funzione di soppressione delle linee che non fa altro che tentare di rendere meno spessi gli edge. Infine applico Canny tramite i valori in input CannyMin e CannyMax.

```
out/piazza_del_popolo_03.jpg
out/faro_tramonto_03.jpg
out/zarautz_skate_03.jpg
```