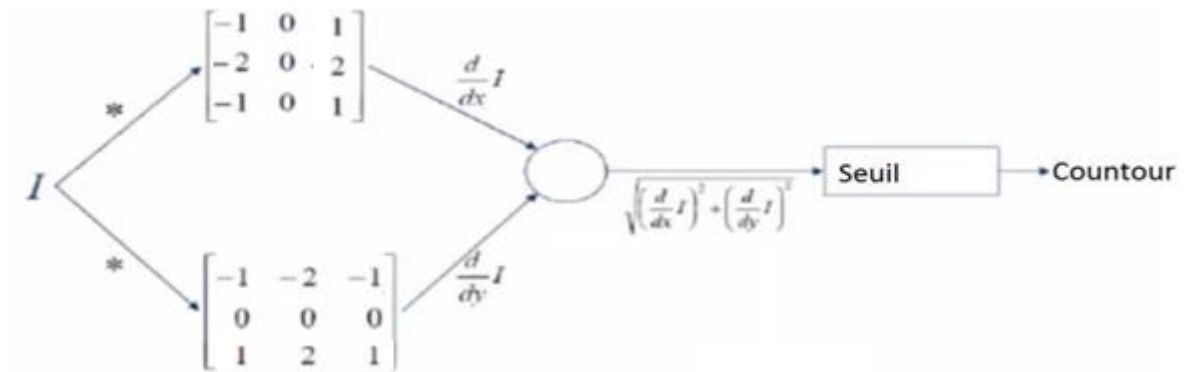


1- Exercice 1

La figure suivante présente le schéma de la méthode utilisant le masque de (Sobel)



En suivant les étapes de ce schéma définir le contour dans l'image suivante

$I = \begin{bmatrix} 10, & 9, & 9, & 4, & 0 \\ 0, & 255, & 255, & 255, & 2 \\ 5, & 255, & 255, & 255, & 3 \\ 7, & 255, & 255, & 255, & 3 \\ 8, & 10, & 8, & 5, & 0 \end{bmatrix}$

Pour cela veuillez déterminer tout d'abord les fonctions suivantes :

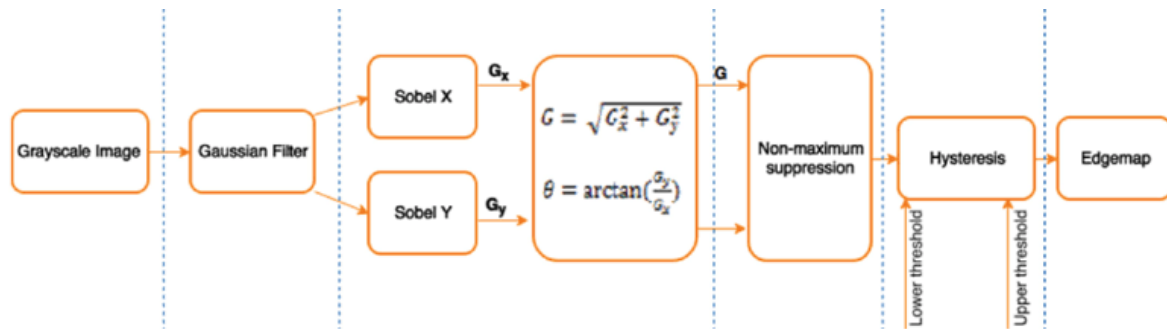
1-1 appliquer_masque(I, M) \rightarrow applique un masque M sur une image I

1-2 gradient_sobel(I) \rightarrow Calcule gradient_sobel Gx, Gy, G et le seuil

1-3 Affiche les contours et exécute votre code sur une image real

Exercice 2

La figure suivante présente le schéma de l'algorithme (Canny)



En suivant les étapes de ce schéma définir le contour dans l'image suivante

```

I = [
[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
[255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
[255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
[255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
[255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
[255, 255, 110, 110, 110, 110, 110, 110, 110, 110, 110, 110, 255],
[255, 255, 255, 170, 170, 170, 170, 170, 170, 170, 170, 255, 255],
[255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
[255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
[255, 255, 255, 255, 170, 10, 170, 170, 10, 170, 255, 255, 255],
[255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
[255, 255, 255, 255, 255, 170, 80, 80, 170, 255, 255, 255, 255],
[255, 255, 255, 255, 255, 170, 170, 170, 170, 255, 255, 255, 255],
[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]
]
  
```

Pour cela veuillez déterminer tout d'abord les fonctions suivantes :

- 1- `gaussian_blur(I, kernel_size=3)` → Applique le filtre gaussien
- 2- `gradient_direction` → calcule θ pour chaque pixel
- 3- `non_maximum_suppression` → affine les contours
- 4- `hysteresis_thresholding` → applique le double seuil et connecte les bords faibles aux forts
- 5- `canny(I)` → fonction complète prête à l'emploi

Affiche les contours pour cette matrice et exécute votre code sur une image real