

Exercice 0 - Mise en place

Créez un nouveau projet Qt en incluant openCV :

Algorithm 1: .pro file

```
LIBS += -L /opt/local/lib -lopencv_core -lopencv_highgui -lopencv_imgproc  
INCLUDEPATH += -I /opt/local/stow/opencv-3.4.3
```

Vérifiez que tout fonctionne en lançant le code suivant :

Algorithm 2: Affichage de fenetre

```
#include "opencv2/opencv.hpp"  
#include "opencv2/highgui/highgui.hpp"  
  
int main(int argc, char** argv) {  
  
    cv::namedWindow("Output", 1);  
    cv::Mat output = cv::Mat::zeros( 120, 350, CV_8UC3 );  
  
    cv::imshow("Output", output);  
    cv::waitKey(0);  
  
    return 0;  
}
```

Exercice 1 - Hello World!

Écrire "Hello World :)" en vert au milieu de la fenêtre.



Figure 1: Résultat final attendu.

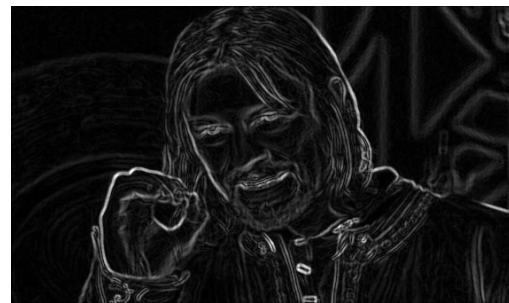
Exercice 2 - Premier filtre

Votre programme devra :

- Lire une image depuis l'ordinateur
- Lui appliquer le filtre de Sobel
- Enregistrer l'image sur l'ordinateur



(a) Input image



(b) Output image

Figure 2: Résultat final attendu.

Exercice 3 - Manipulation d'images

Votre programme devra :

- Lire une image stéréo depuis l'ordinateur
- La découper en deux images séparées
- Enregistrer les deux images générées sur l'ordinateur



Figure 3: Exemple d'image stéréo à découper.

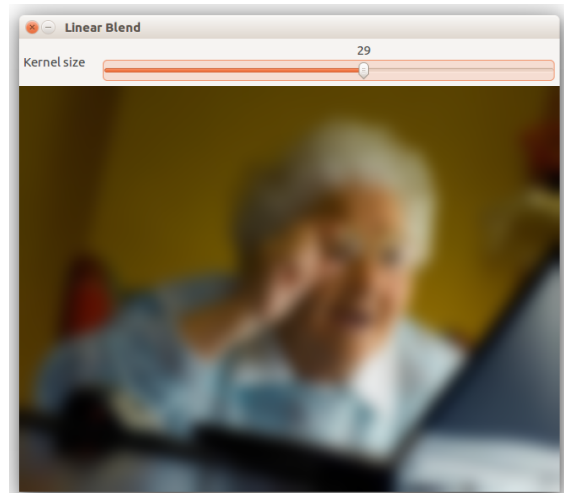
Exercice 4 - Première GUI

Votre programme devra :

- Lire une image depuis l'ordinateur
- Lui appliquer un flou Gaussien
- Afficher un slider permettant à l'utilisateur de modifier la taille du noyau
- Mettre à jour immédiatement l'affichage avec le nouveau paramètre



(a) Kernel_size = 0



(b) Kernel_size = 29

Figure 4: Résultat final attendu.

Exercice 5 - Premier programme complet

Votre programme devra :

- Lire une image depuis l'ordinateur
- Lui appliquer un filtre au choix parmi au moins trois (choix avec slider)
- Modifier le ou les paramètres du filtre (choix avec slider)
- Enregistrer l'image une fois le traitement terminé (enregistrer avec slider)