**Université de la Manouba**

**Ecole Nationale des Sciences de l'Informatique**



Mini-rapport du Stage d'été II1

-Stage de Programmation-

**Sujet**

|  |
| --- |
| **Construction d’une base de contour à partir d’une base d’image de mosaïque** |

**Réalisé par**

HAMMEMI Mohamed Soulaimen

BEN KRAIEM Mohamed Amin

JEMAI Siwar

Année Universitaire 2020/2021

1. **Présentation générale du sujet**

Il s’agit d’une application développée en Matlab qui détecte le contour d’une image mosaïque.

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse.

Cette technique réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image.

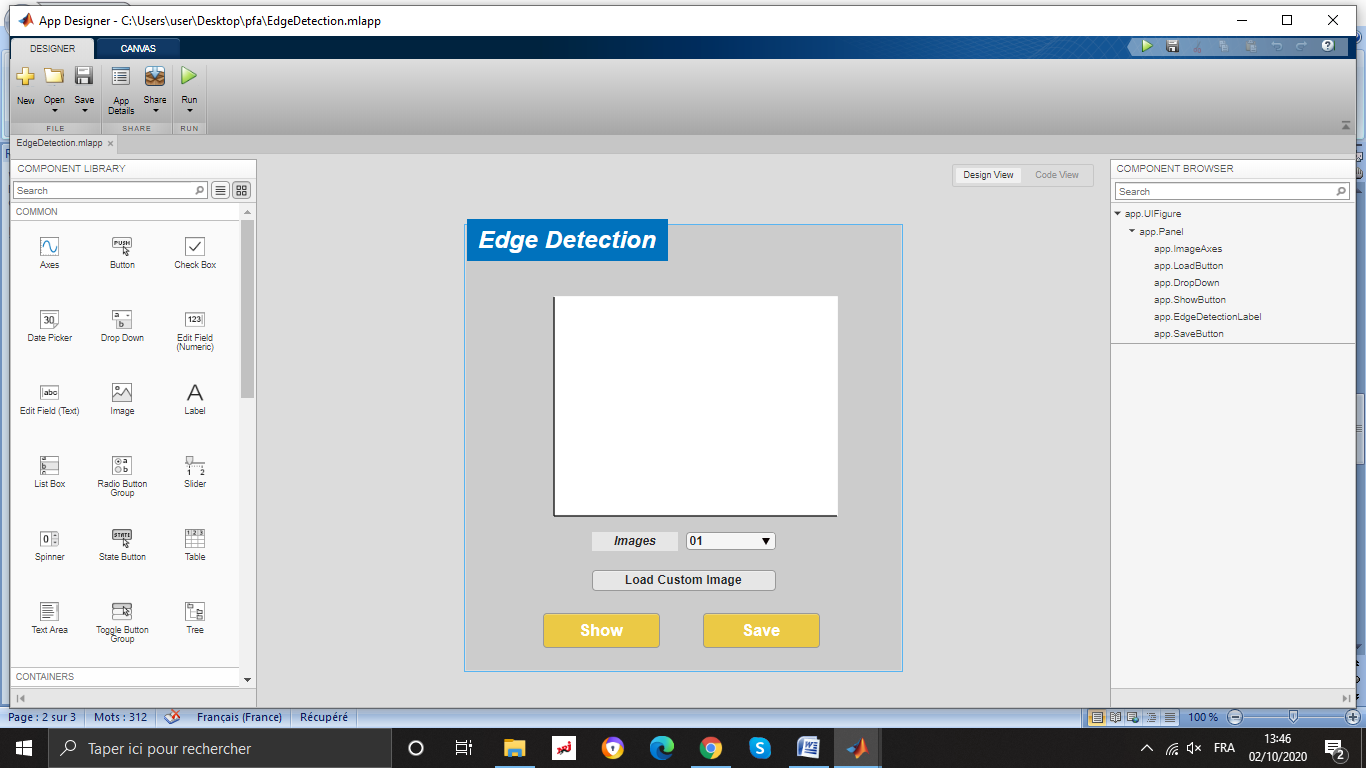
1. **Présentation des fonctionnalités demandées de l'application**

Afin de détecter le contour il faut passer par ces étapes :

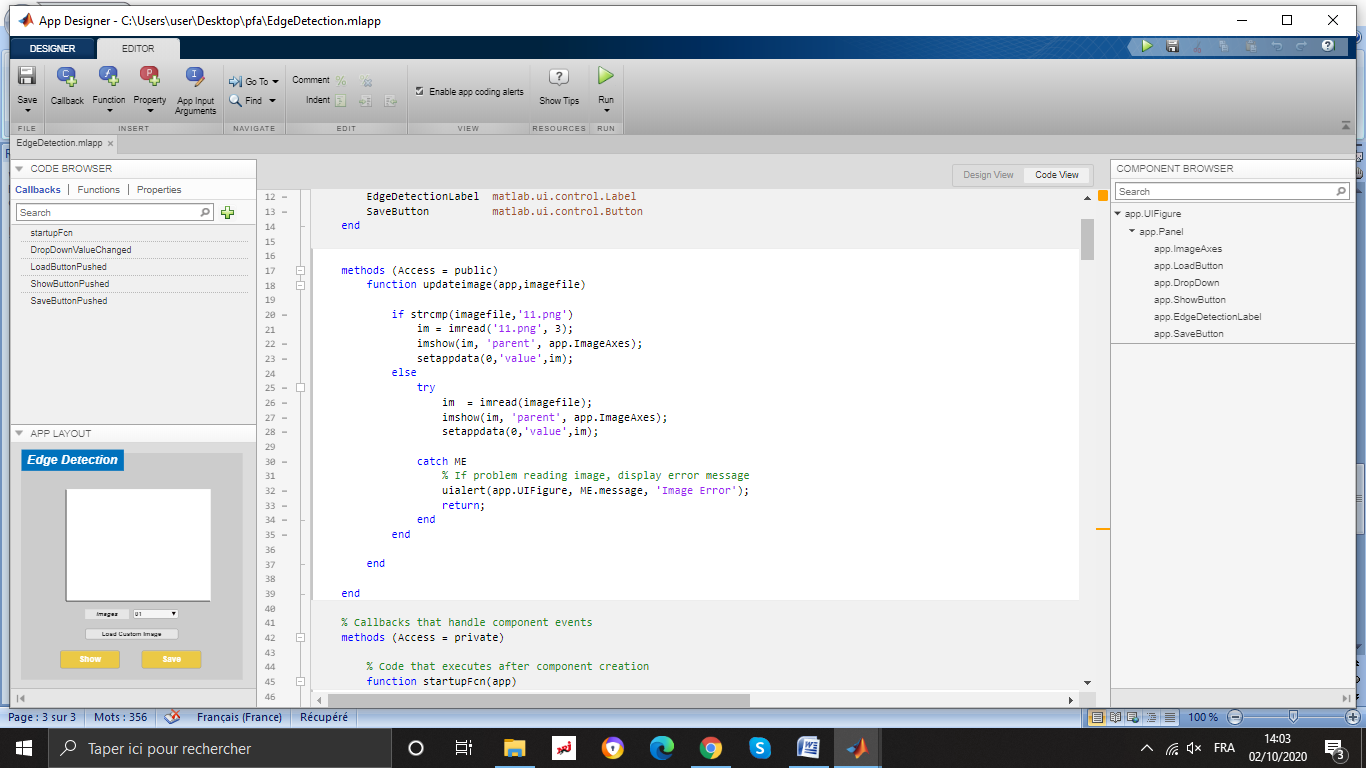
1. Le filtrage de l’image
2. Le calcul de gradients et normes
3. La suppression des non-maximums
4. Le seuillage par hystérésis
5. Trouver le seuil automatiquement
6. **Présentation de la conception de la solution proposée** 
   1. **Architecture du programme :**

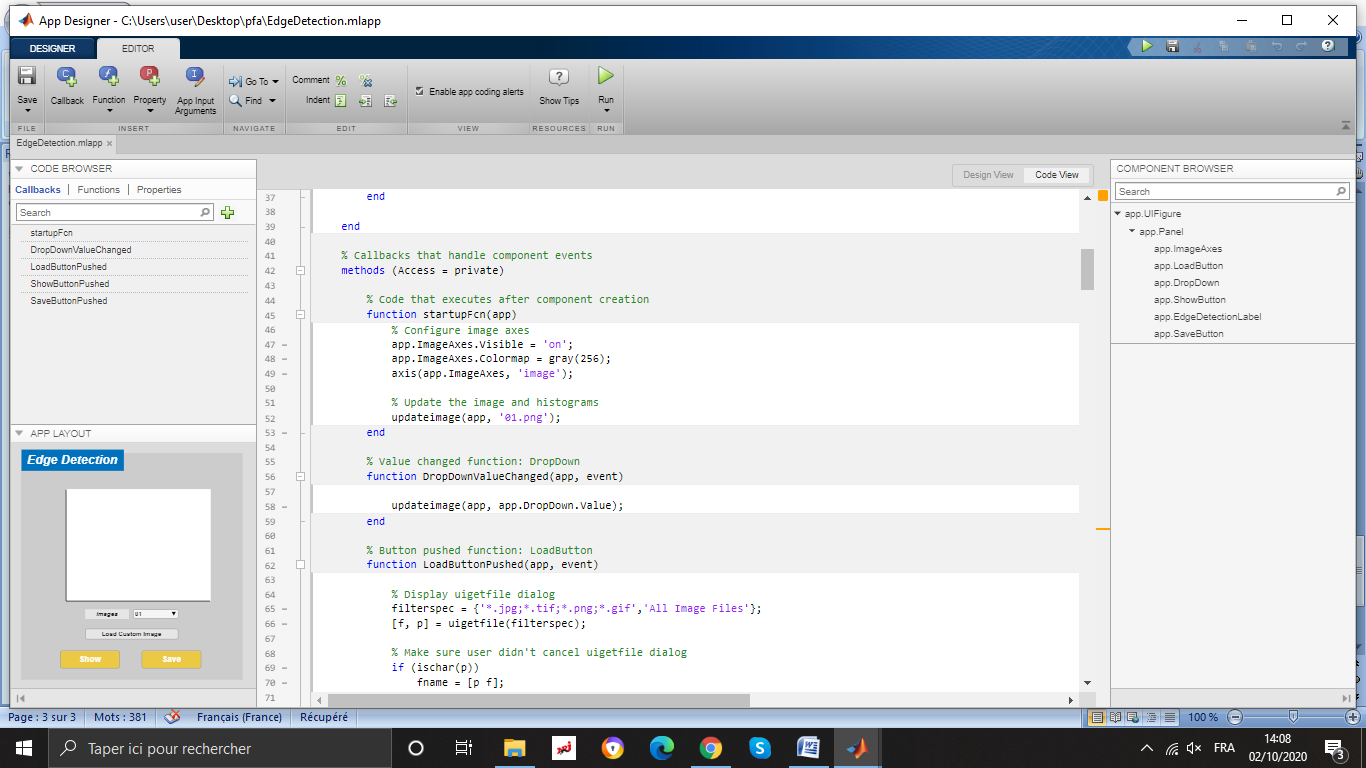
Le projet contient principalement :

* Un fichier matlab EdgeDetection.mlapp contenant le code source.
* Un fichier texte edm.txt contient la matrice finale.
* Les images mosaïques.
* Une base de donnée contenant les images après leurs détection du contour.
  1. **Détails de la solution**

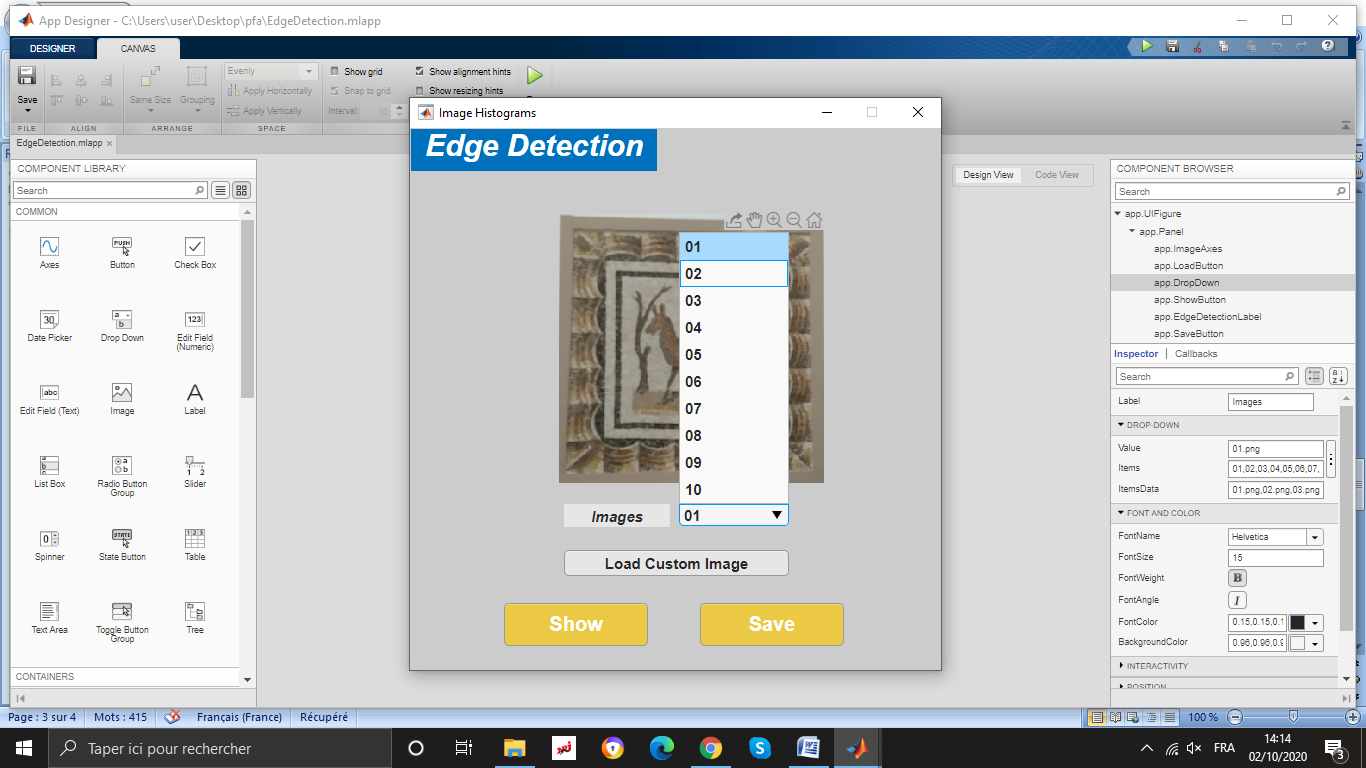
Lors du développement de l’application on a choisi app designer pour réaliser une interface graphique interactive comme l’indique cette image :

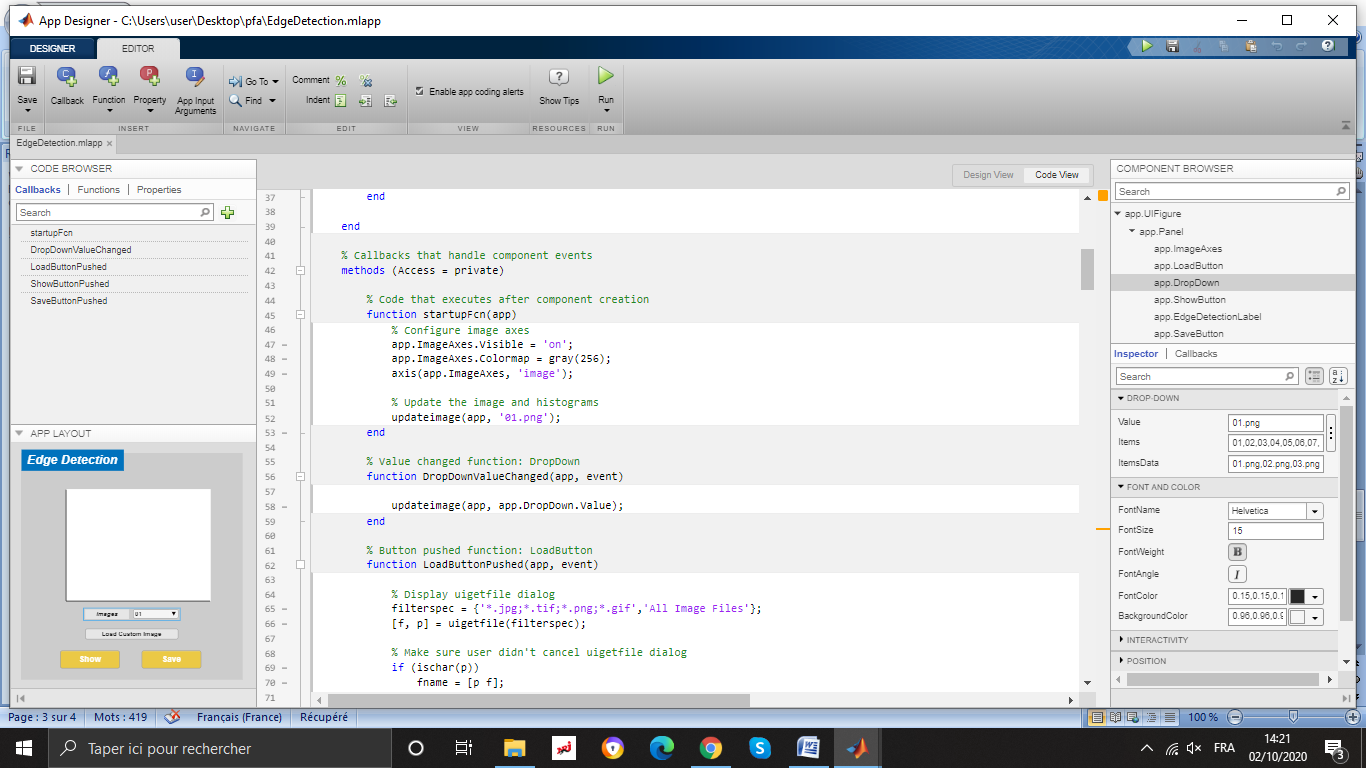
Le code est divisé en une fonction updateimage et un callback qui gère les événements de composants à travers ses fonctions.

1. La fonction **updateimage** a pour but de mettre à jour l’image choisie par l’utilisateur au niveau de l’app.imageAxes
2. Le Callback est divisé en 5 fonctions :
3. Fonction **startupFcn**:son rôle est de configurer les axes d’image , l’image par défaut et le colormap



1. Fonction **DropDownValueCanged :** Au niveau de l’interface on trouve un menu déroulant qui nous permettre de choisir une image de la liste indiquée pour l’afficher au niveau de app.ImageAxes(en faisant appel à ).

**Interface graphique Code source**

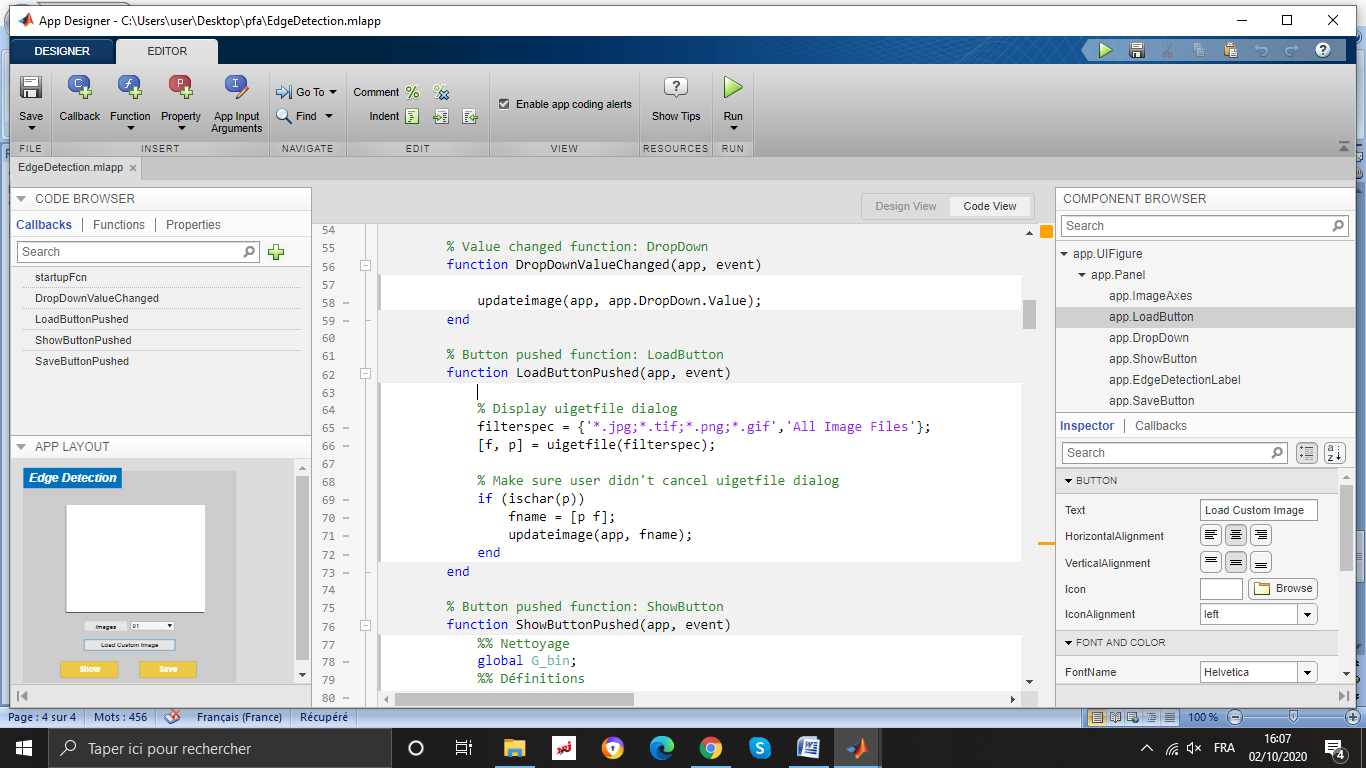
****

1. Fonction **LoadButtonPushed** : Cette fonction permet de charger une autre image selon le choix de l’utilisateur .

En cliquant sur le bouton ‘Load Custom Image’ une fenêtre s’ouvre comme l’indique l’image suivante .



**Le code source :**

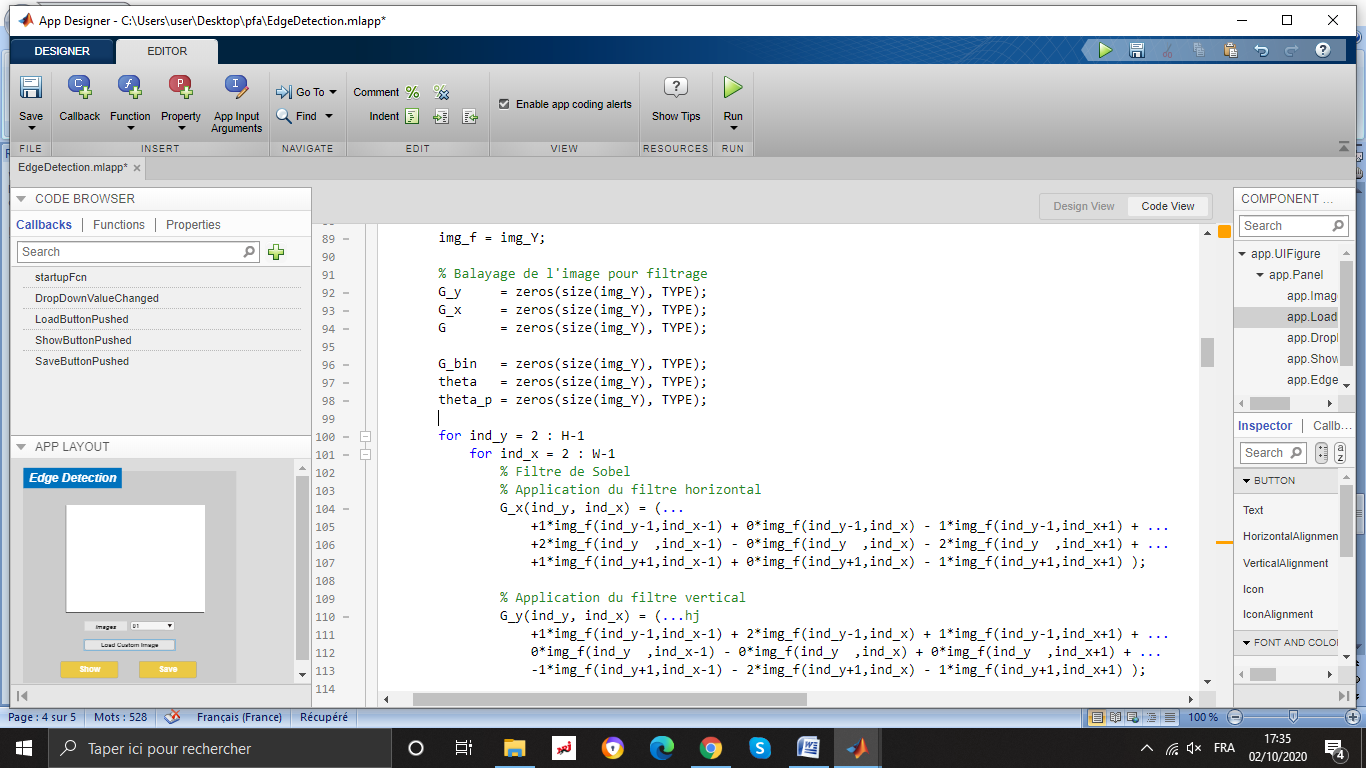
****

1. Fonction **ShowButtonPushed :** En cliquant sur le bouton ‘show’ , la fonction se déclenche. Elle commence par remplir les matrices par des zéros pour faire le filtrage .

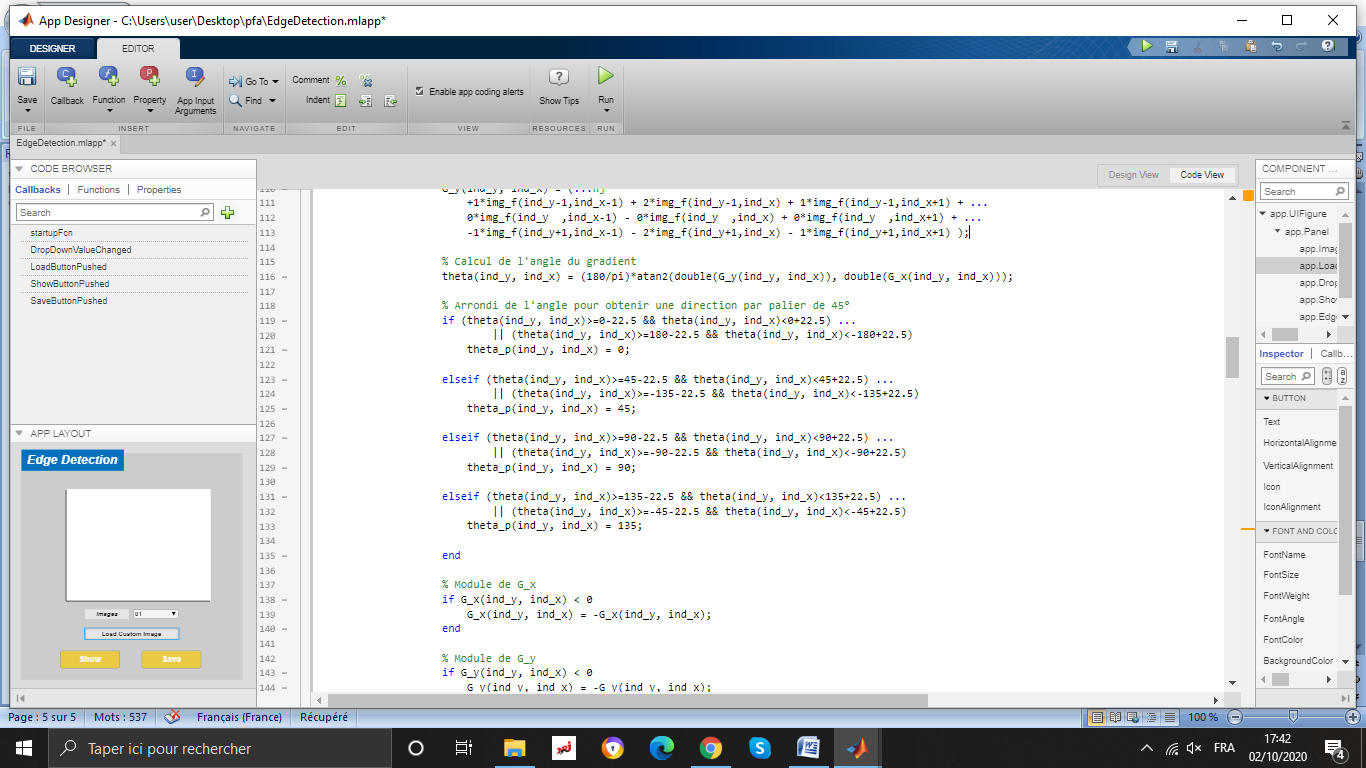
Après il y a le filtrage horizontal et vertical en faisant un produit de convolution avec les matrices [-1 0 1] et [-1 -2 -1] pour obtenir G\_x et G\_y.

[-2 0 2] [0 0 0]

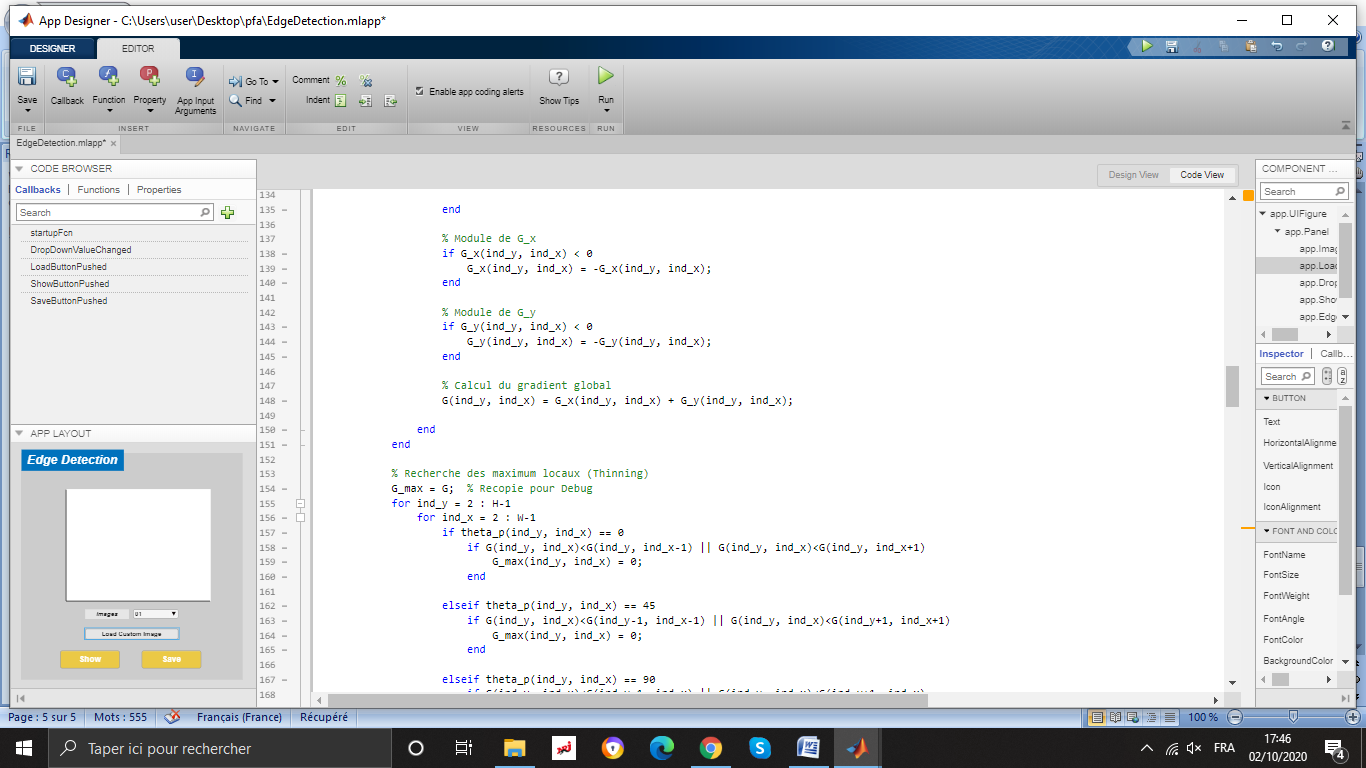
[-1 0 1] [1 2 1]

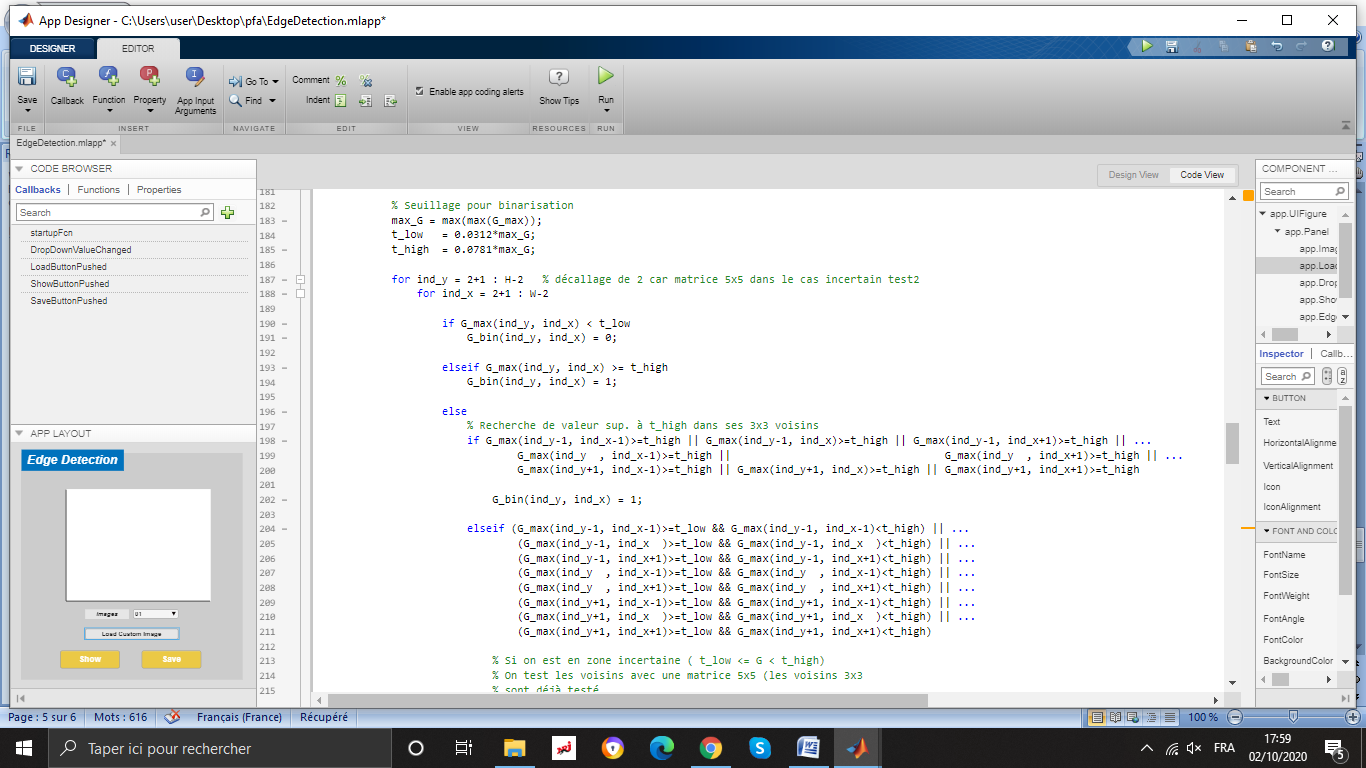


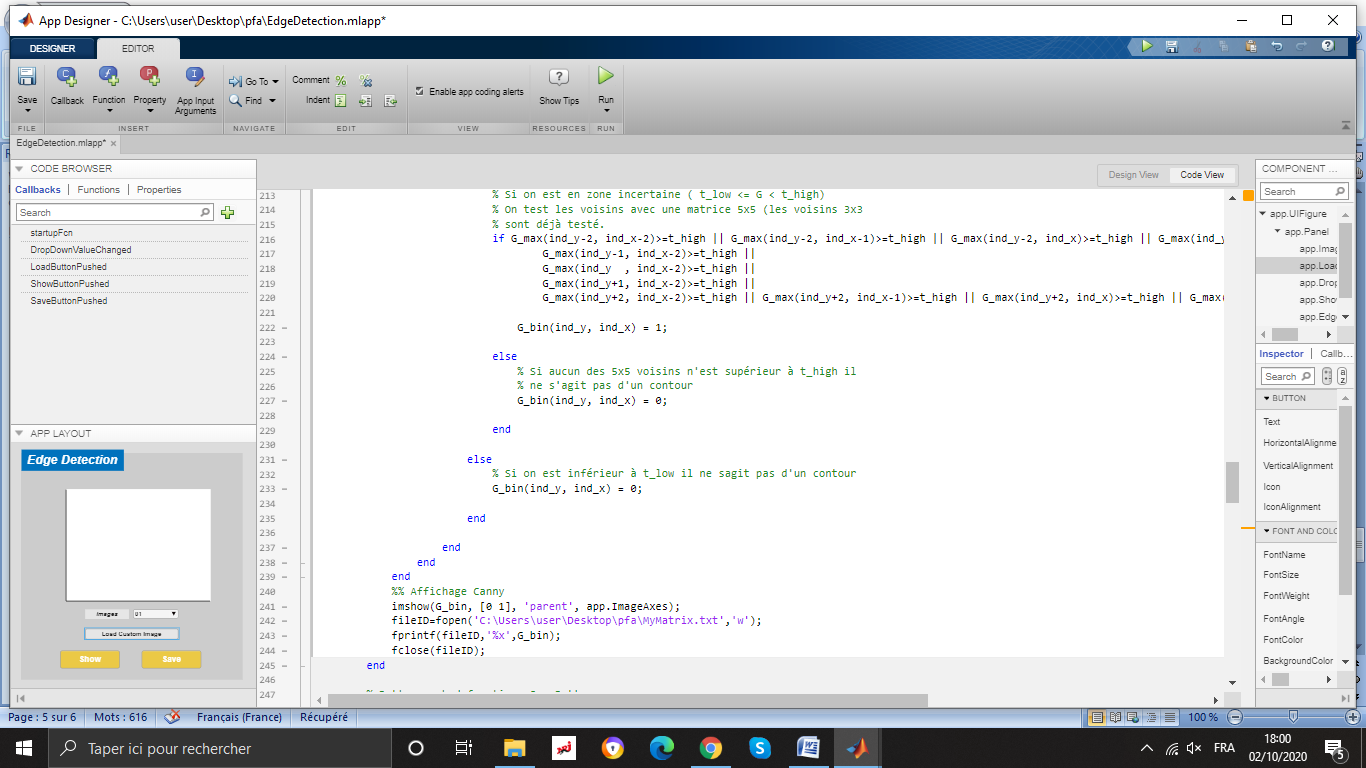
Ensuit il ya le calcul du gradient et l’arrondi selon l’image suivante



Après on cherche le module de G\_x et G\_y et on calcule le gradient global



Pour finalement rechercher les maximums locaux et binariser par seuillage(qui est une technique qui sert à changer une image en niveau de gris en une image dont les valeurs de pixels ne peuvent avoir que des 0 et 1).



La derniére étape de cette fonction est d’afficher le contour dans app.ImageAxes et écrire la matrice résultante dans un fichier texte.

1. Fonction **SaveButtonPushed** : Cette dernière fonction consiste à enregistrer le contour obtenu.
2. **Manuel d’utilisation** 
   1. **Environnement logiciel**

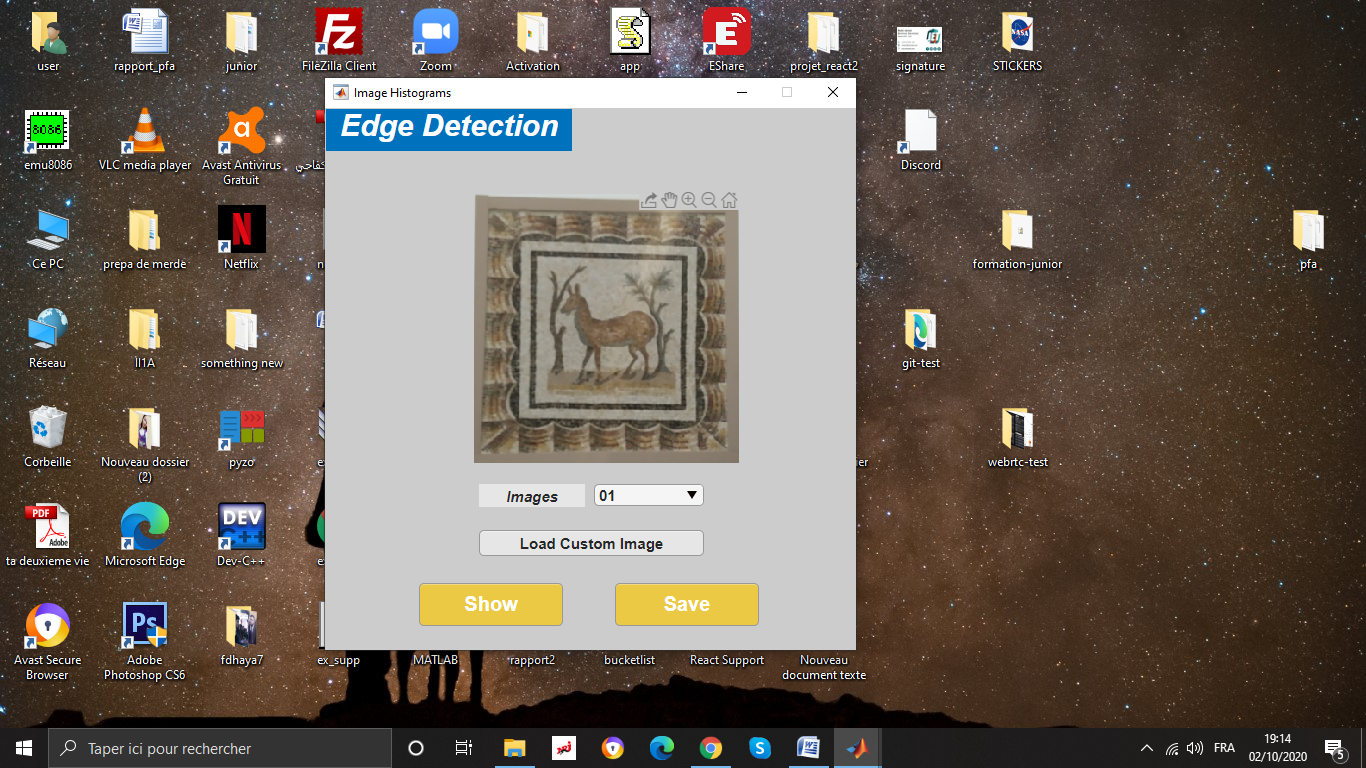
L’application est développée en Matlab qui est un langage de script qui permet de manipuler des matrices et d’afficher des courbes et des données.

Il permet le travail interactive soit en mode commande, soit en mode programmation tout en ayant toujours la possibilité de faire des visualisations graphiques.

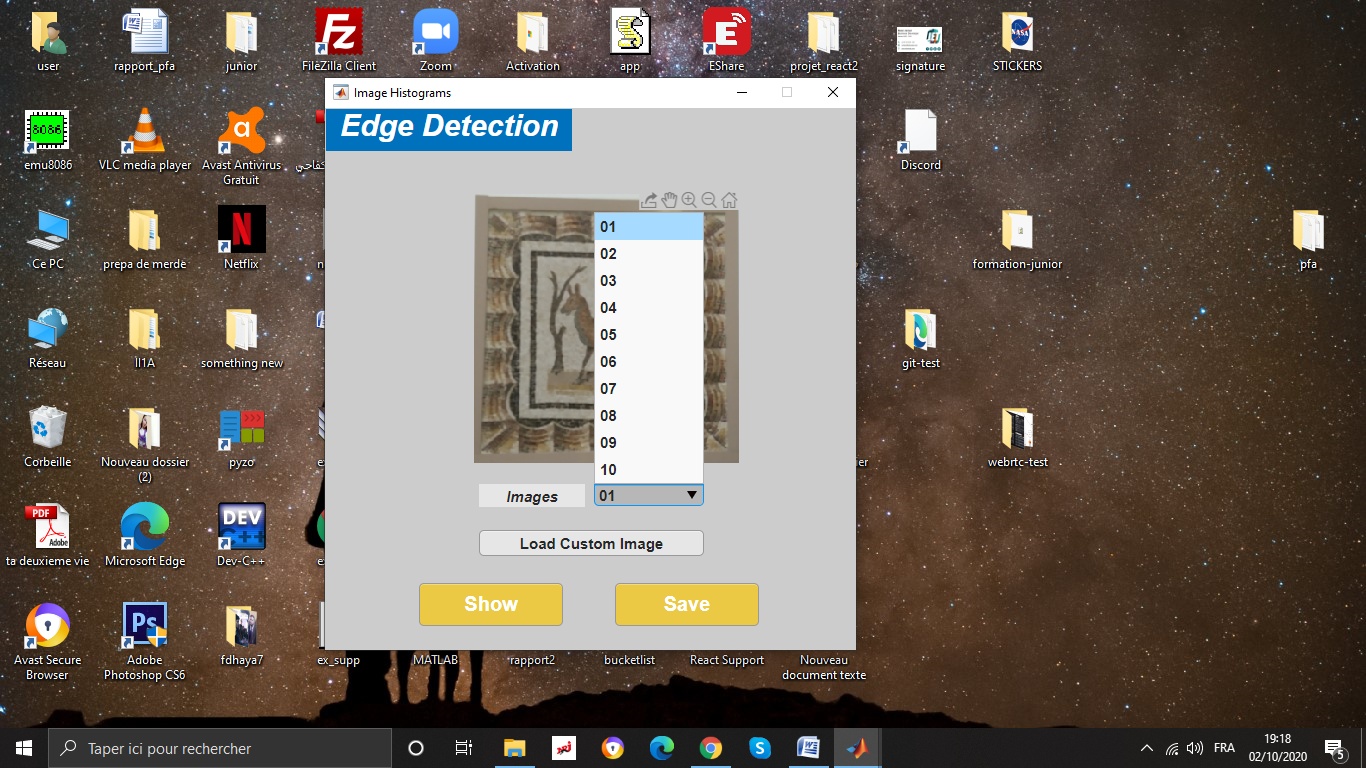
Concernant l’interface graphique utilisateur est fait par app designer qui est un outil de conception pour les applications Matlab. Il nous permet de créer des applications de qualité professionnelle en glissant les composants visuels dans le canevas pour concevoir le design de votre interface graphique utilisateur (GUI) et utilisez l'éditeur intégré pour programmer rapidement son comportement.

* 1. **Présentation du travail réalisé**

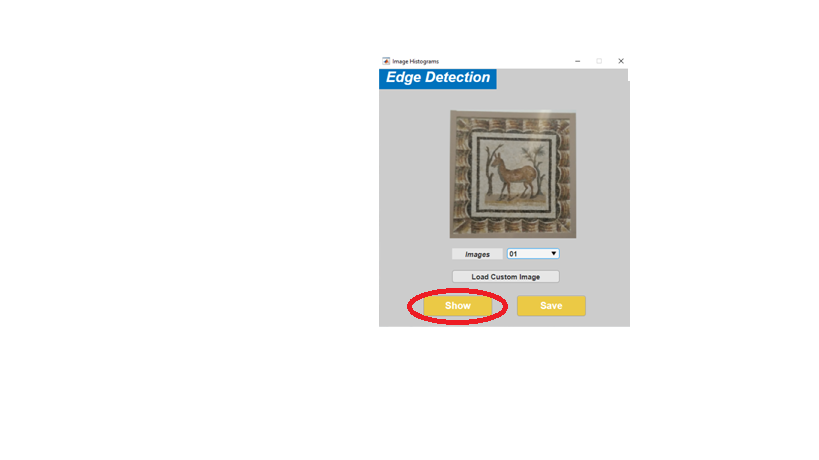
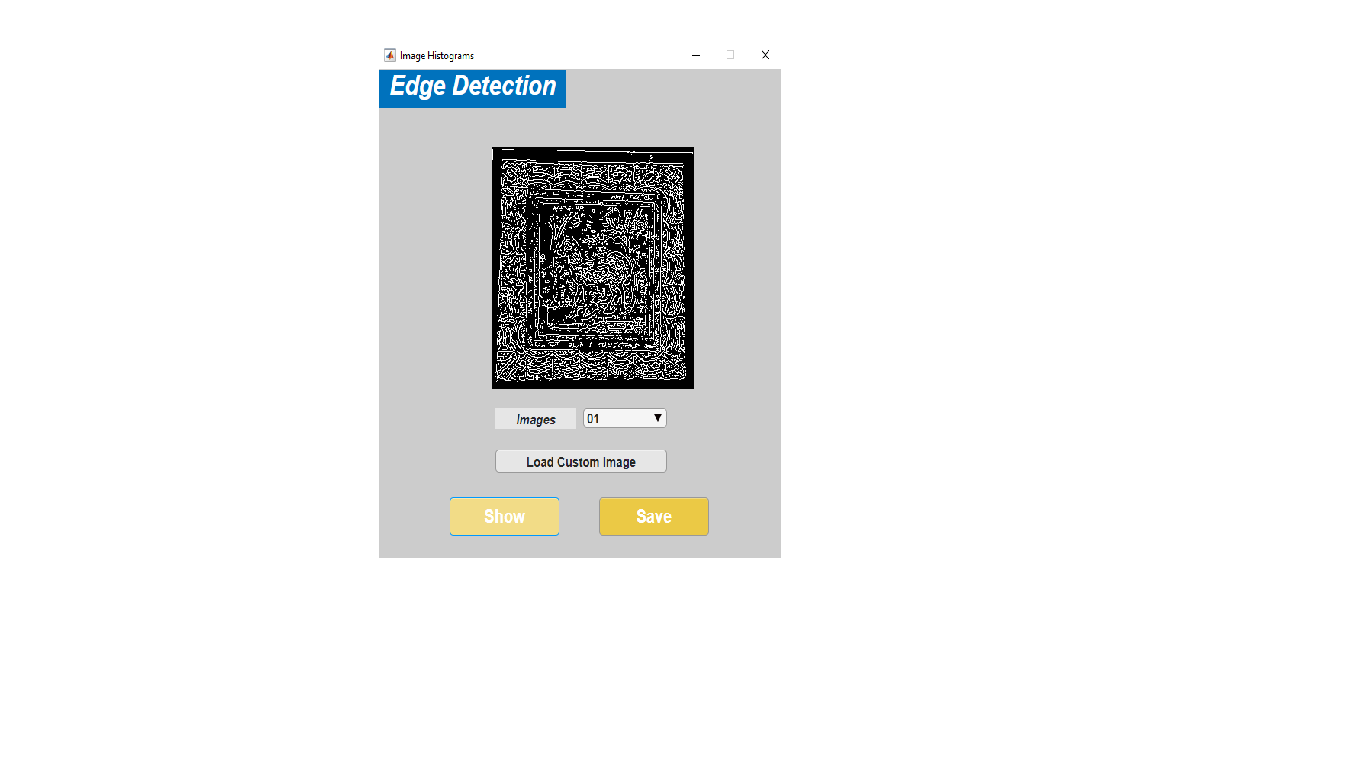
Dés l’ouverture de l’application cet interface s’affichera :



Pour choisir une image parmi les existantes dans le menu déroulant il suffit de cliquer sur le menu

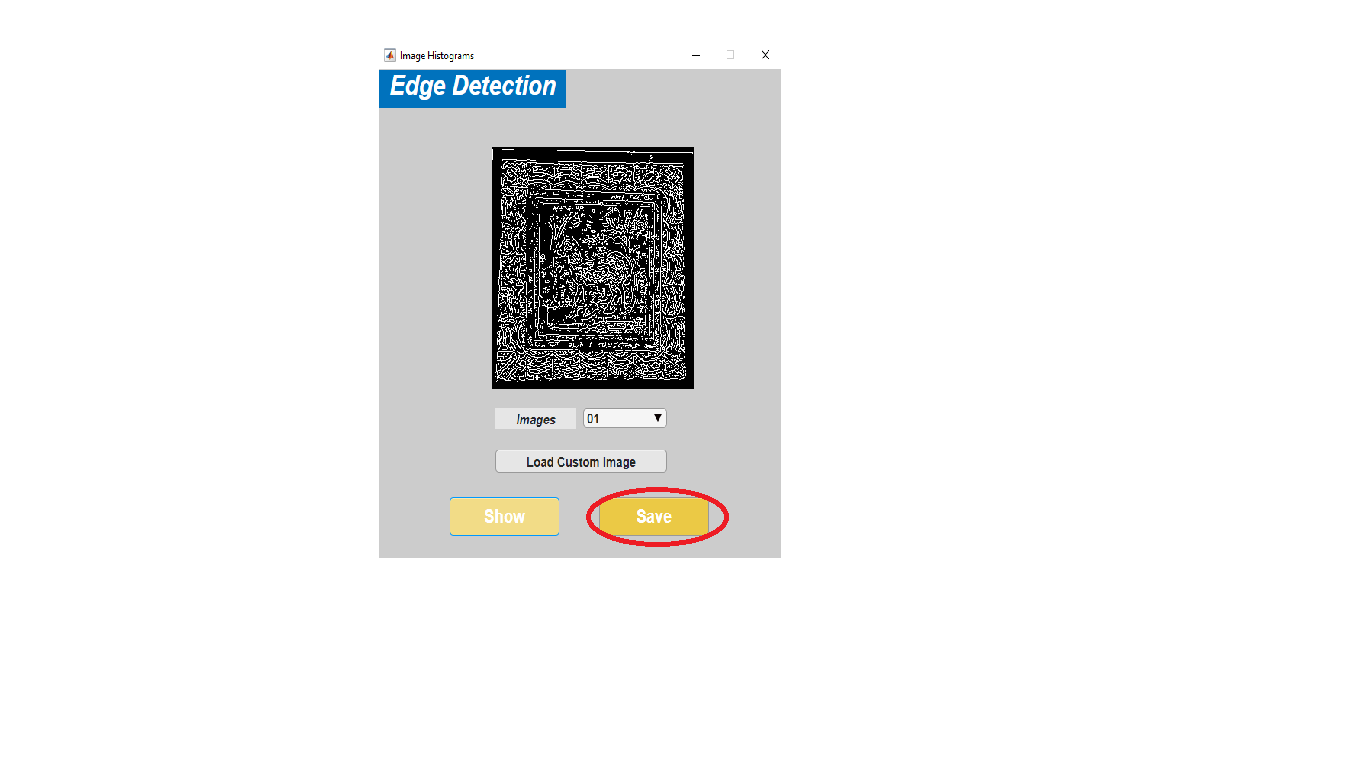


Pour afficher le contour, il suffit d’appuyer sur bouton show et une fenêtre s’affichera comme suit



Résultat

Pour l’enregistrer il suffit de taper sur save comme suit :





Résultat

1. **Les problèmes rencontrés et leurs solutions**

|  |  |
| --- | --- |
| Problèmes rencontrés | Solutions proposées |
| On a rencontré de comprendre le principe de app designer | **On a fait des recherches plus approfondies et on a commencé à travailler étape par étape en parallèle avec des tutos** |
| Au début, on a fait l’interface utilisateur indépendamment de la partie de détection du contour. Le problème est de faire l’implémentation de ces deux parties en un seul code source | **La solution qu’on a choisi est de copier l’erreur et faire une recherche et essayer de le corriger** |

1. **Références**

* <https://colinleverger.fr/assets/projects/CANNY-COLIN-LEVERGER.pdf>
* On a trop utilisé la documentation de Matlab

<https://fr.mathworks.com/help/>

* <https://www.youtube.com/watch?v=Z4msmuJNpI4>
* <https://fr.wikipedia.org/wiki/D%C3%A9tection_de_contours>
* <http://www2.eng.ox.ac.uk/~labejp/Seminar/GUI/Notes.pdf>
* <https://www.youtube.com/watch?v=iga-YS6VbyE&list=PLDzzooPYRJ_XNBUkbCkOM-TBWFeO5Ud4I>
* <https://www.youtube.com/watch?v=5k8RxErq6O8>