

Imprimante I3D et la Classification



Cet atelier a pour objectif d'aborder l'apprentissage automatique à partir de la mini-caméra de l'imprimante I3D. Nous verrons

- 1- comment automatiser la collecte des données avec Simulink en effectuant des mesures des dimensions d'objets à partir des images collectées,
- 2- comment utiliser le Classification Learner App de MATLAB pour étudier les techniques de classification,
- 3- comment ensuite embarquer le modèle généré pour superviser le déplacement de la caméra.

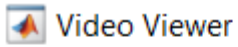
Il faudra bien connecter les cables USB de la mini-webcamera et de la carte smoothie sur votre ordinateur.

Contrôler la position de la caméra avec la carte smoothie

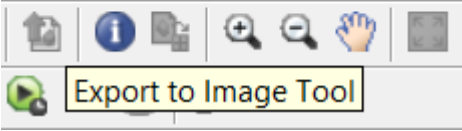
Ouvrir le modèle Connexion_smoothie_captures.slx modifier/vérifier les différents ports de connexion.

Nous utilisons les commandes G-code pour contrôler les moteurs de l'imprimante afin de positionner la caméra à 2 hauteurs différentes 157.32mm et 177.32mm (en Z). Il est conseillé de prendre connaissance des différentes commandes G-code en impression 3D se trouvant dans le document I3D_CommandesGcode.pdf.

Afin de faire une correspondance entre les pixels (480x640) et la distance en mm, vous pouvez utiliser l'outil "Export to Image Tool" du bloc Video Viewer du modèle:



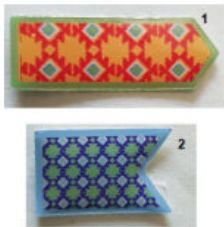
File Tools View Simulation



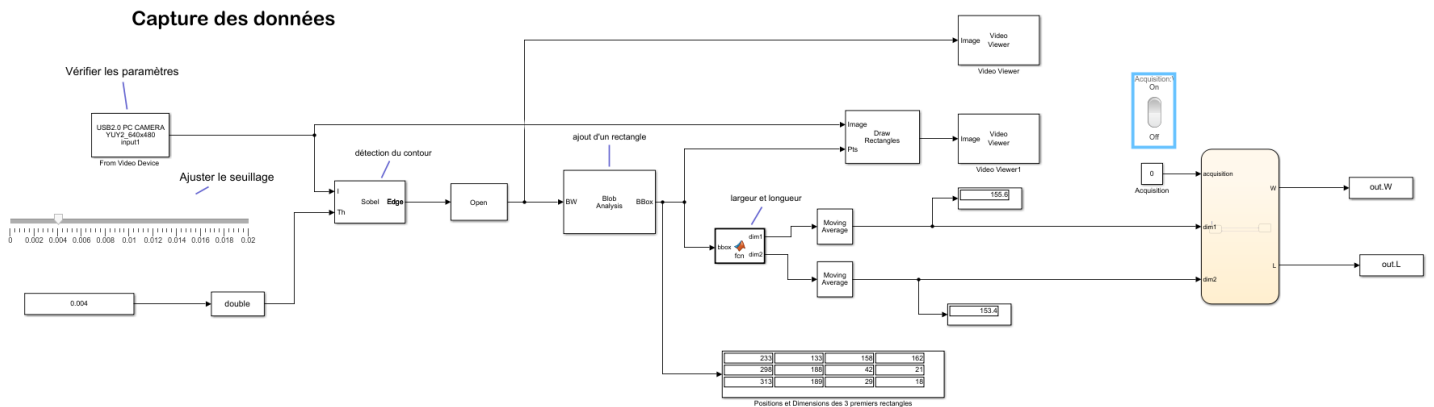
Collecte des données dans un fichier Excel

Vous pourrez tester plusieurs positions de la caméra et déterminer la meilleure hauteur pour la caméra en fonction des objets ou caractéristiques de l'image.

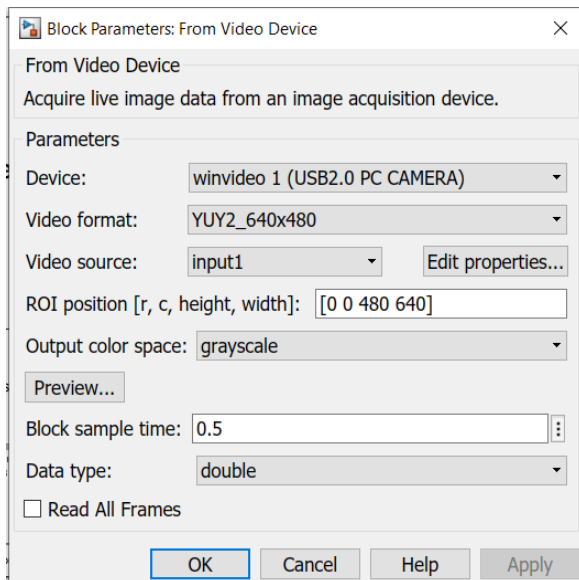
Choisissez 2 objets distincts



et ouvrir le modèle Sobel1.slx



Lors de l'acquisition, bien vérifier le contenu du bloc "From Video Device". Attention à la dimension des matrices 480x640:



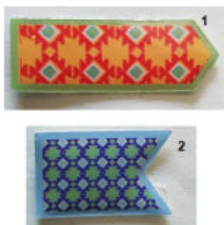
Le traitement consiste à effectuer à une détection de contour avec la méthode de Sobel. En fonction de la luminosité de la salle, ajuster le seuillage et la luminosité de la caméra à l'aide la molette. Nous utilisons le bloc "Blob Analysis" pour positionner un rectangle délimitant l'objet. Nous collectons alors 2 dimensions:

- largeur avec la variable W
- longueur avec la variable L

Le diagramme Stateflow nous permet de suspendre les acquisitions pour déplacer l'objet sur le plan.

Pour éliminer les redondances dans les mesures, exécuter le script `clean_data.m` et construire un fichier Excel comprenant 3 colonnes: W, L et Forme.

Concernant la Forme de l'objet, utilisez une étiquette numérique (1 ou 2):



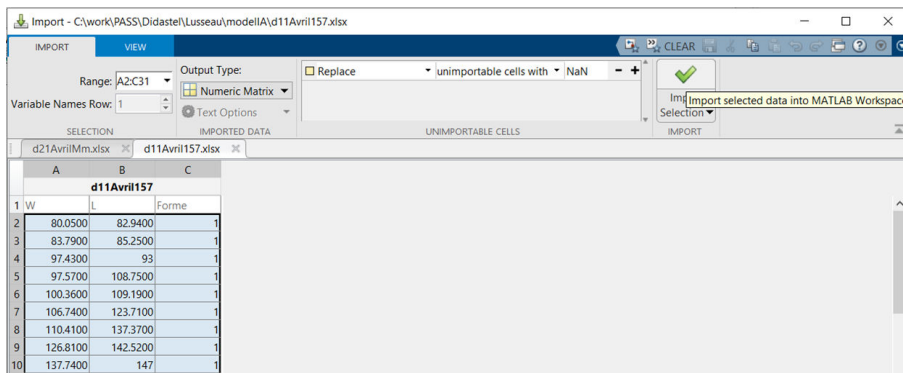
Pour la suite, vous pouvez utiliser les fichiers `d11Avril157.slx` et `d11Avril177.slx` que nous avons créés.

Apprentissage Automatique: Classification

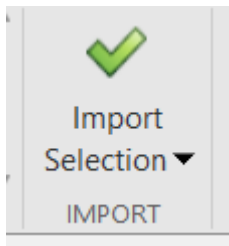
Charger des données existantes dans une matrice

Cliquer sur votre fichier excel créé. Par exemple, `d11Avril157.xlsx` par exemple, en utilisant l'outil d'import.

Sélectionner le format Numeric Matrix:



et cliquer sur



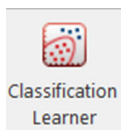
"Import Data". Une matrice à 3 colonnes sera créée dans le workspace de MATLAB.

Utilisation du Classification Learner

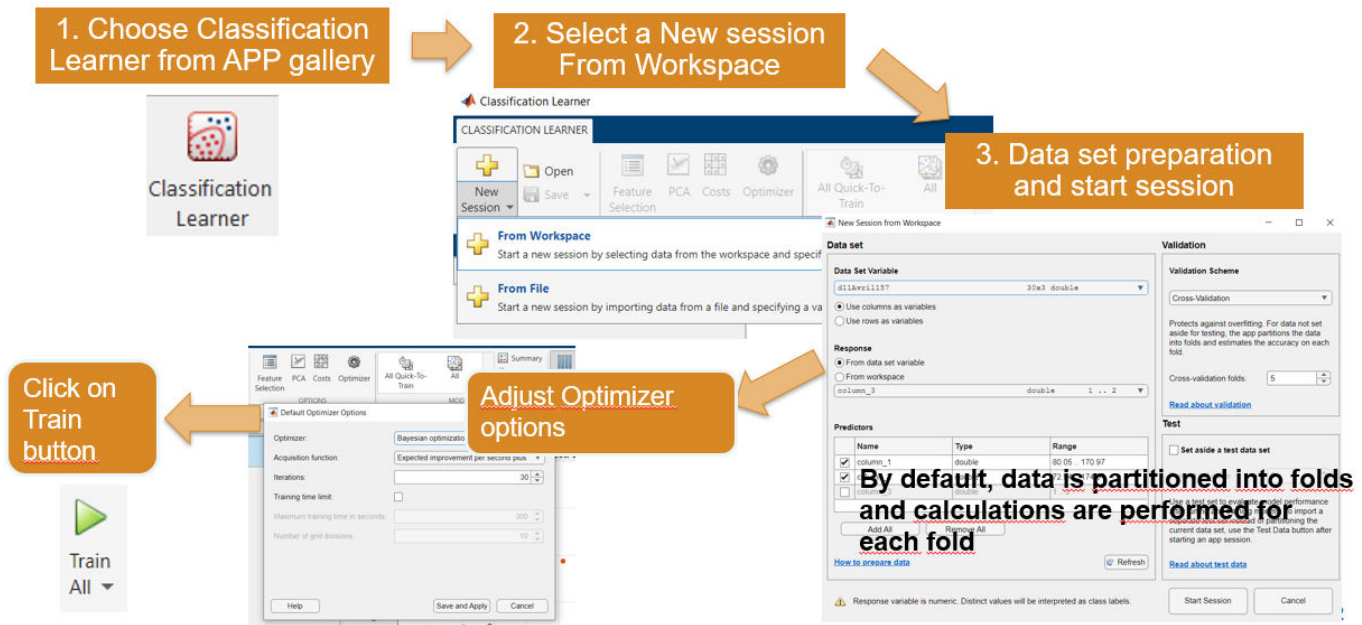
Pour éviter les étapes précédentes, charger les données

```
load('d11Avril157.mat');
%load('d21AvrilMm.mat');
```

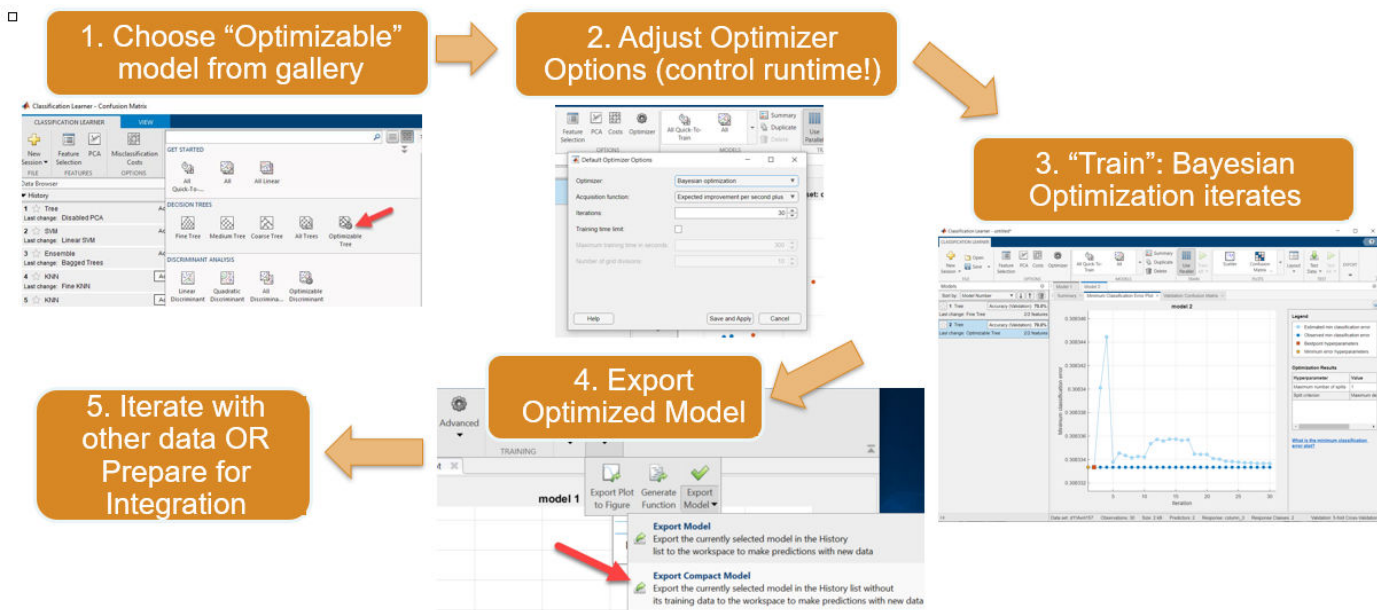
Dans l'onglet APPS de MATLAB, cliquer sur Classification Learner:



Créer une nouvelle session à partir de données du Workspace en sélectionnant la variable d11avril157 du workspace et cliquer sur Start Session:



Suivre le workflow ci-dessous pour chaque méthode: Tree, SVM et KNN:



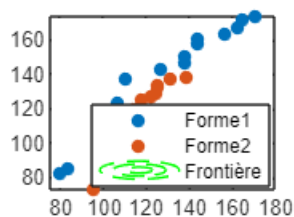
Lors de l'export des modèles Compacts (pour l'intégration dans Simulink), nommez les classifieurs par: trainedModelTree, trainedModelSVM et trainedModelKNN.

Afficher les frontières des modèles prédictifs ou classifieurs

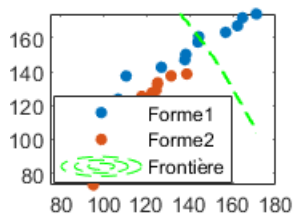
Utiliser la fonction myborder avec 2 entrées: la matrice des données et le classifieur

```
myborder(d11Avril157,trainedModelTree)
```

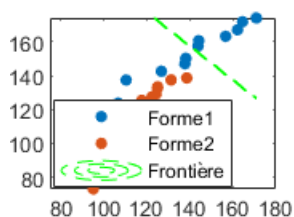
Warning: Contour not rendered for constant ZData




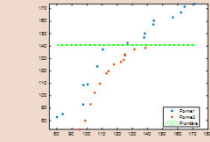
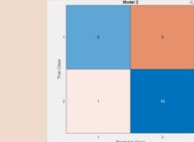

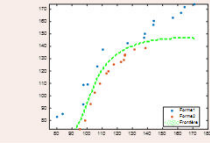
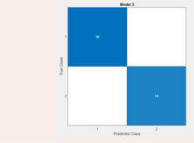
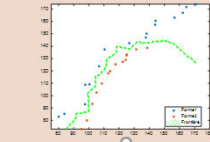
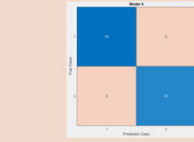
```
myborder(d11Avril157,trainedModelSVM)
```



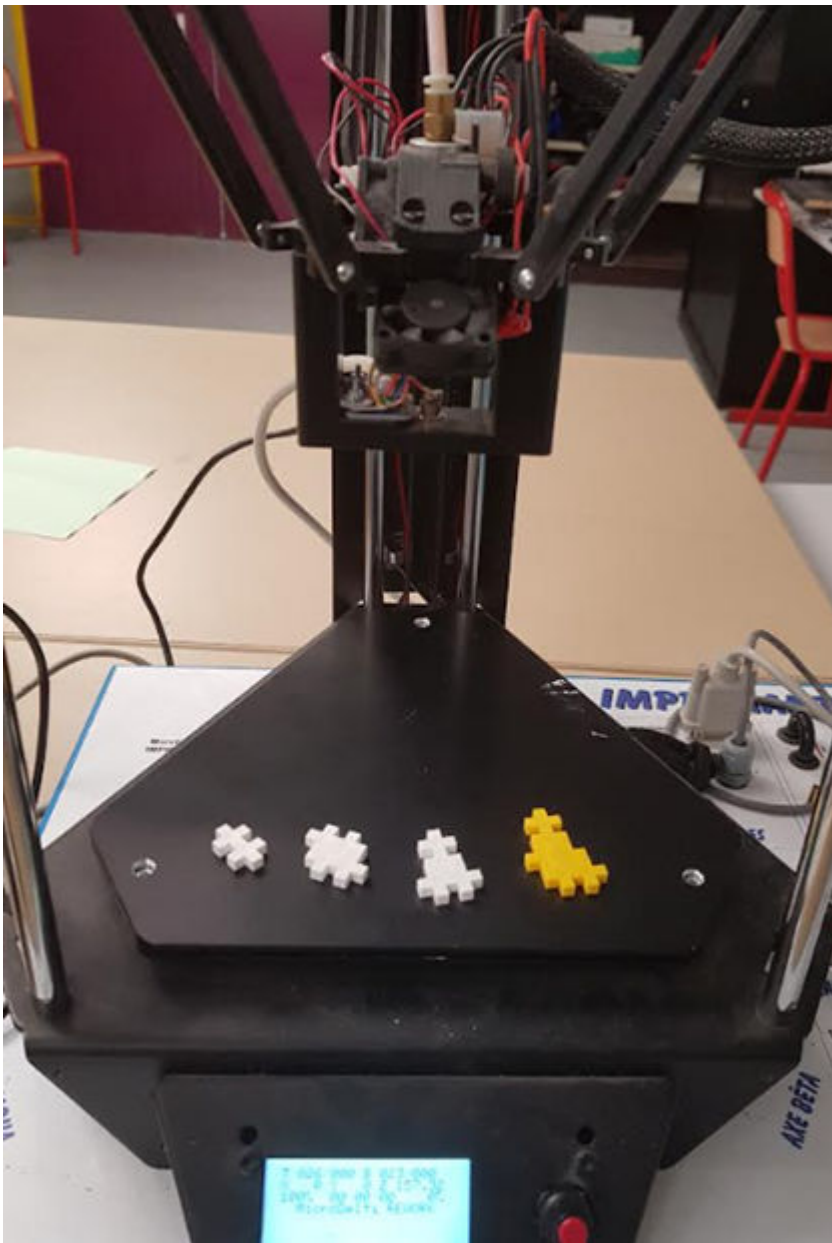
```
myborder(d11Avril157,trainedModelKNN)
```



Voici les les résultats escomptés:

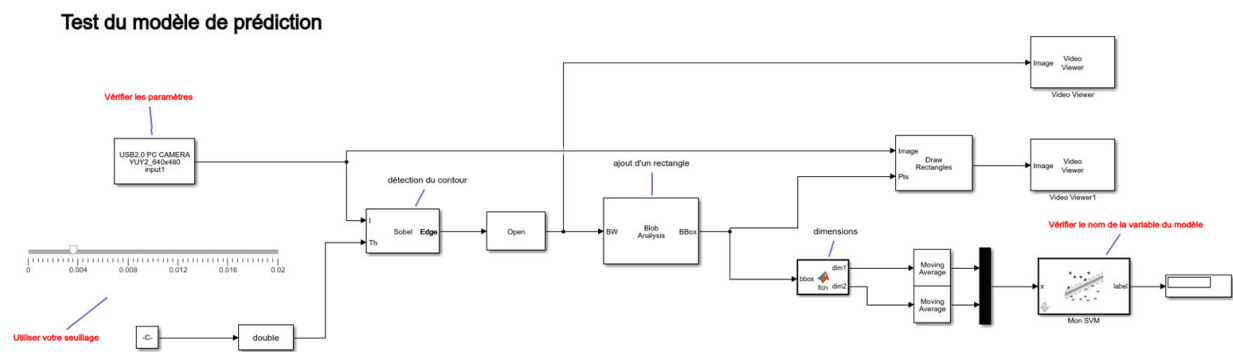
Results	Method	Border	Confusion Matrix
1 	Optimizable Tree Accuracy 70%		
2 	Optimizable SVM Accuracy 100%		
	Optimizable KNN Accuracy 80%		

Tester d'autres objets comme les Basic Mix ++, d'autres techniques d'extraction de caractéristiques ou d'autres hauteurs de la caméra.



Intégration dans Simulink

Ouvrir le modèle Sobel2.slx



Vérifier les différents paramètres sur la connexion avec la caméra, le seuillage et le nom du classifieur.

Le modèle Sobel3.slx intègre un diagramme d'état pour contrôler la position de la caméra en fonction de l'objet.

Les modèles Sobel11.slx, Sobel22.slx et Sobel33.slx comprennent une autre méthode d'extraction de caractéristiques pour de meilleures prédictions avec les objets Basic Mix ++.