

Manuel d'application d'un prototype de robot de production de béton précontraint

L'application permet d'utiliser le bras Niryo One pour pouvoir sélectionner des objets à placer sur des intersections de droites détectées par caméra. Cette application possède une interface graphique permettant à l'utilisateur de réaliser les choix d'objets à déplacer ainsi que les intersections sur lesquels les déplacer.

Ce document présente comment mettre en œuvre et utiliser l'application pour réaliser ces actions.

I. Utilisation de l'application

1) Environnement de travail :

L'application a été développée pour être exécutée sous Windows. Elle nécessite Python3 (Python 3.7.9), opencv, Pyqt et tensorflow. Pour pouvoir installer les versions compatibles nécessaires à l'application, les fichiers requirements.txt et setup.bat (dossier Install du GitHub) ont été créés afin de rendre plus facile l'installation de toutes les librairies nécessaires avec les bonnes versions. requirements.txt contient les librairies nécessaires et le fichier setup.bat permet de créer un environnement virtuel "niryo_one_env" sur votre ordinateur qui contiendra toutes les versions correctes des librairies pour l'exécution du programme.

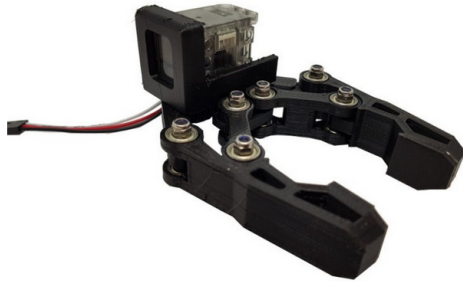
- Récupérer le code source depuis le dépôt Git
https://github.com/errikosmes/Bras_KP1
- Se placer dans le répertoire cloné
- Exécuter le fichier setup.bat dans le dossier où vous souhaitez créer votre environnement.
- Activer l'environnement. dans votre invite de commande à l'emplacement où vous avez créé l'environnement, taper la commande :
"niryo_one_env\Scripts\activate.bat"

```
C:\Users\ameli\Desktop\testbat>niryo_one_env\Scripts\activate.bat
(niryo_one_env) C:\Users\ameli\Desktop\testbat>
```

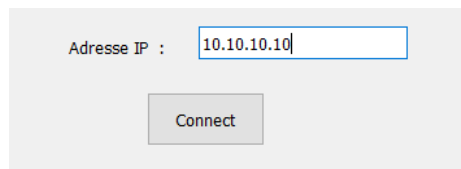
Puis vous pouvez vous déplacer là où se trouve votre fichier à exécuter.

2) Exécution

- Mettre en place la pince “large gripper” (pince numéro 2)

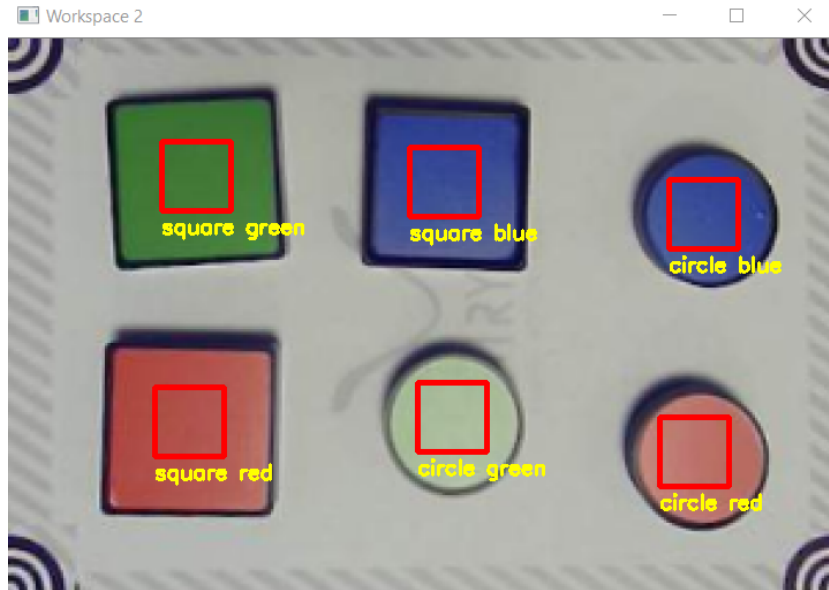


- Allumer le robot et attendre la mise en place de la led bleu.
- Se connecter au robot : Soit par cable RJ45 , soit par wifi (mdp wifi : niryooone)
- Lancer l'application via invite de commande. Activer votre environnement virtuel puis python ./main.py (en se plaçant dans le fichier src du projet). Le lancement du programme peut mettre quelques dizaines de secondes, car le logiciel charge le module TensorFlow nécessaire à la détection des différents objets.
- Une fenêtre s'ouvre permettant de renseigner l'adresse IP du robot. Si vous êtes connecté en wifi, renseignez l'IP “10.10.10.10”, si vous êtes connecté par câble, l'adresse IP est : “169.254.200.200”. Cliquez ensuite sur “connect”.

A screenshot of a web interface for connecting to a robot. It features a text input field labeled "Adresse IP :" containing the value "10.10.10.10". Below the input field is a button labeled "Connect". The entire interface is set against a light gray background.

Le Bras va alors réaliser une calibration automatique si aucune calibration a été réalisé auparavant, puis va se placer au dessus du "workshop"

- Une fenêtre avec l'image du workshop s'ouvre et permet de cliquer sur les différents objets détectés par l'intelligence artificielle.
Si vous voulez remettre à jour l'image vous pouvez cliquer sur le bouton capture (un temps de mise à jour de quelques secondes aura lieu).
Vous pouvez alors choisir les différents objets que vous souhaitez déplacer dans l'ordre que vous le désirez grâce à votre souris, une fois sélectionné le carré de l'objet passe de la couleur rouge à vert. Il est possible de sélectionner un objet déjà sélectionné.
- Une fois la sélection réalisée, appuyez sur “Entrée”.



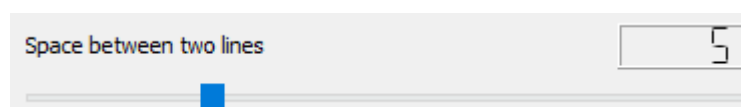
SI CETTE FENÊTRE NE S'OUVRE PAS : Cela signifie que la caméra ne détecte pas le workshop. Vérifiez que les 4 points permettant de cadrer le workshop sont visibles par la caméra.

- Le bras va maintenant se placer sur le "workspace" en face de lui. Une nouvelle fenêtre va s'ouvrir pour montrer ce workspace et détecter les intersections où les objets peuvent être déposés. Pour cela, le bras utilise 3 valeurs : sensibilité, espace entre deux lignes, et espace entre deux points :
 - Sensibilité : La taille des lignes minimale que le programme détecte, si des lignes ne sont pas détectées, il faut augmenter ce seuil. Si trop de lignes sont détectées, il faut alors diminuer ce seuil.



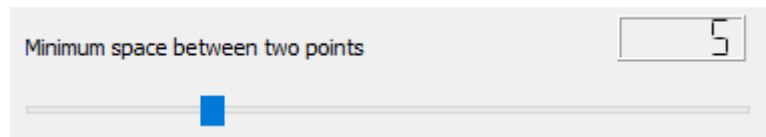
ATTENTION : plus il y a de lignes détectées, plus le programme nécessite de temps de calcul, si le seuil est mis trop haut, le programme va alors mettre beaucoup de temps à afficher la mise à jour.

- Espace entre deux lignes. Le programme va parfois trouver plusieurs lignes très proches pour une même ligne tracée. Le paramètre "espace entre deux lignes" regroupe les lignes proches en une même ligne. Plus cette valeur est grande, plus les lignes éloignées seront aussi supprimées, plus cette valeur est faible, plus des lignes éloignées seront regroupées ensemble



- Espace entre deux points : Si plusieurs lignes sont détectées pour une même ligne, le programme trouvera alors plus d'intersections qu'il n'y en a vraiment.

Cette valeur permet de regrouper les intersections proches. Plus cette valeur est grande, plus les points éloignés seront regroupés ensemble. Plus cette valeur est faible, plus les points éloignés seront conservés.



- Ces trois paramètres sont remis à jour lorsque vous cliquez sur “Capture”. Pour bien régler ces paramètres, nous vous conseillons de procéder dans cet ordre :
 - Régler la valeur de sensibilité permettant de détecter toutes les lignes que vous souhaitez avoir (même si trop de lignes sont détectées).
 - Régler ensuite les paramètres “Espace entre deux lignes” et “espace minimum entre deux intersections” pour avoir le nombre d’intersections que vous désirez détecter.
- Vous pouvez ensuite sélectionner les intersections sur lesquelles vous souhaitez déposer les objets dans l’ordre que vous le désirez avec la souris. Une fois sélectionné, le carré de l’intersection passe de la couleur rouge à vert. Il est possible de sélectionner un objet déjà sélectionné. Lorsque vous avez choisi, appuyer sur entrée.
- Le robot exécute alors les actions, puis s’arrête en se remettant dans sa position de repos. Pour relancer le programme vous devez alors fermer la fenêtre principale, et relancer le programme.

Extinction du robot :

ATTENTION : Pour éteindre le robot de manière correcte, il faut appuyer sur le bouton utilisateur du robot durant 3 secondes (jusqu’à ce que la led à l’arrière du robot passe au rose). Après cela, attendre que la led devient rouge, puis basculer l’interrupteur sur la position d’arrêt (O)

II. Ajout de détection d'objet par IA

Pour implémenter la détection d’objets avec l’IA, il faut d’abord commencer par créer une base d’apprentissage. Vous devez utiliser le programme python “labelling” fourni dans le dossier **Visual_Learning_by_Artificial_Intelligence** du GitHub. Ce programme vous permettra de créer un dossier pour chaque nouvel objet et de prendre des images de cet objet dans le workspace que vous choisirez. Pour choisir le workspace il suffit de modifier la position d’observation “observation_pose” du robot. Pour avoir une base d’objet complète, nous vous conseillons de prendre 20 à 30 photos de chaque objet dans des conditions différentes (position, luminosité ...).

Une fois tous vos objets labellisés, vous pouvez entraîner votre modèle. Pour ce faire, il suffit d'exécuter le programme "training.py" qui va utiliser un modèle de deep learning pour faire de la reconnaissance d'image. Vous pouvez modifier votre modèle selon les caractéristiques que vous souhaitez lui donner.

Une fois l'entraînement terminé, un dossier nommé **model** va être créé contenant votre modèle et ses caractéristiques.

Vous pouvez alors prendre ce dossier et le remplacer dans la partie **src/model IA/model**, pour tester votre modèle dans le programme principal.

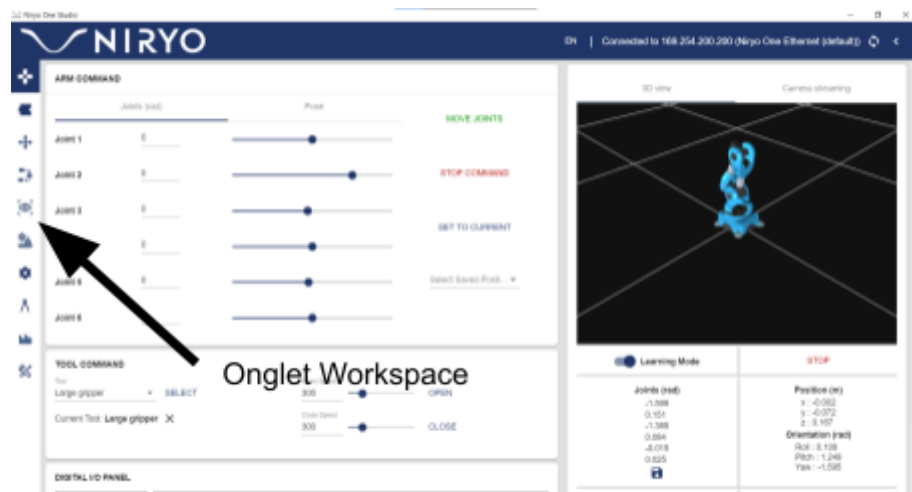
ATTENTION : Si les objets ajoutés ne sont pas des cercles / carré de couleurs, il est aussi nécessaire de changer la fonction find_objects_workshop_ML pour que le programme puisse détecter ces objets.

III. Débuggage

- Problème de positionnement de workspace :
Si le robot n'est pas précis dans la prise ou la pose des objets, une redéfinition des workspace peut être nécessaire. Cela doit aussi être réalisé lorsque vous déplacez le workspace.
Pour cela lancer l'application NiryoOne studio (disponible sur le site de niryo one) ,




puis se rendre dans l'onglet workspace :



Les deux workspace utilisés dans l'application sont : "workspace_default" et



"Workshop_v2". Il faut alors les supprimer et les redéfinir avec le bouton  , puis suivre les instructions proposées par l'application. ATTENTION : nommer les workspace avec les bon nom exact : "workspace_default" pour le workspace face au robot, et "Workshop_v2" pour le magasin.

Si le problème persiste après avoir redéfini le workspace, et que l'articulation 4 semble décalée, il peut être nécessaire de démonter et remonter un des moteurs du bras du robot. La vidéo "how-to-change-motor-4.mp4" montre comment réaliser cela.

- Affichage dans la console : "[ERROR] ATTENTION z is too small!" et arrêt du robot. Cette erreur signifie que le robot a calculé un mouvement en dessous de la surface du workshop. Il n'effectue alors pas l'action pour ne pas s'endommager. Cela est souvent dû à la mauvaise pince qui est fixée sur le bras. Celle-ci doit être la pince numéro 2 : large Gripper.
- Le robot ne se remet pas en position de repos après l'exécution du programme et la levée d'une erreur.
Un programme zero.py est présent dans le dossier src du projet. Ce programme permet de se connecter à l'adresse IP indiquée dans le fichier, et donner l'ordre de se mettre en position de repos ("Learning mode"). Pour l'exécuter, il suffit d'utiliser la commande python, zero.py, en ayant changé l'adresse IP adéquate dans le fichier. Ce programme est très utile lorsque l'on teste de nouvelles fonctionnalités avec le robot, puisque si une erreur apparaît et qu'aucun ordre de remise en position de repos n'a pas été envoyé, le bras va rester dans sa position actuelle, et les moteurs vont continuer à tourner pour cela.