

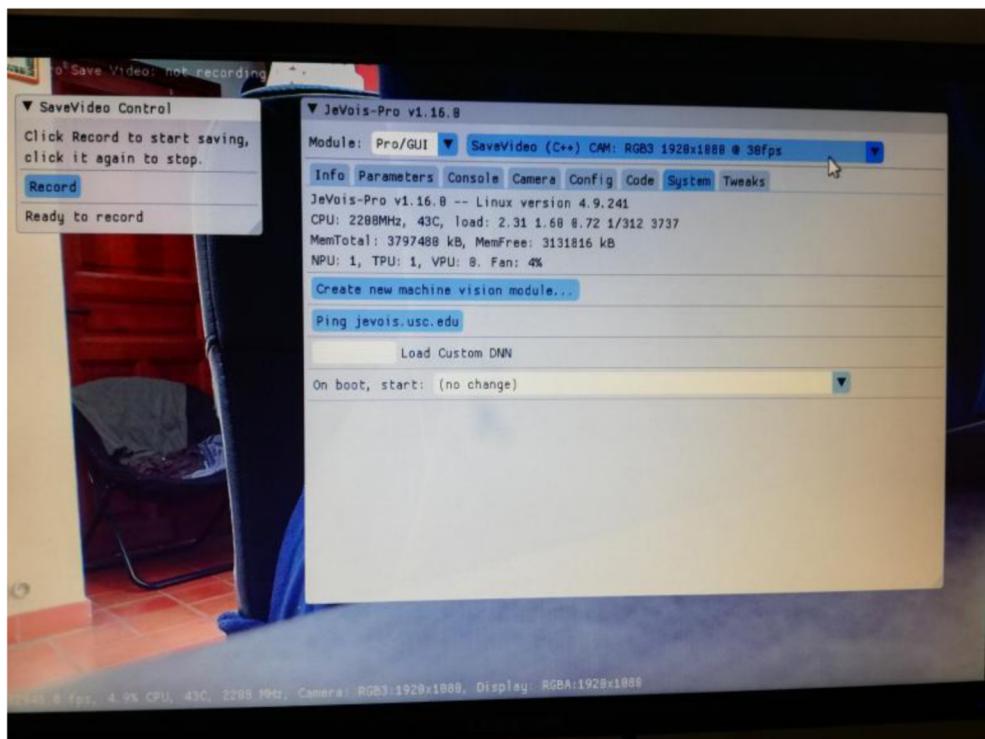
# Annexe : guide d'utilisation des caméras JeVois-Pro

Ce guide permet de modifier les paramètres du logiciel des caméras JeVois-Pro, pour pouvoir coder dessus, créer de nouveau modules, et Automatiser leurs démarrage avec les paramètres nécessaires.

- I. Changer d'interface**
- II. Changer les paramètres de démarrage**
- III. Créer un nouveau module sur le JeVois**
- IV. Ajouter un nouveau modèle YOLO sur la JeVois**
- V. Notions et commandes linux de base**

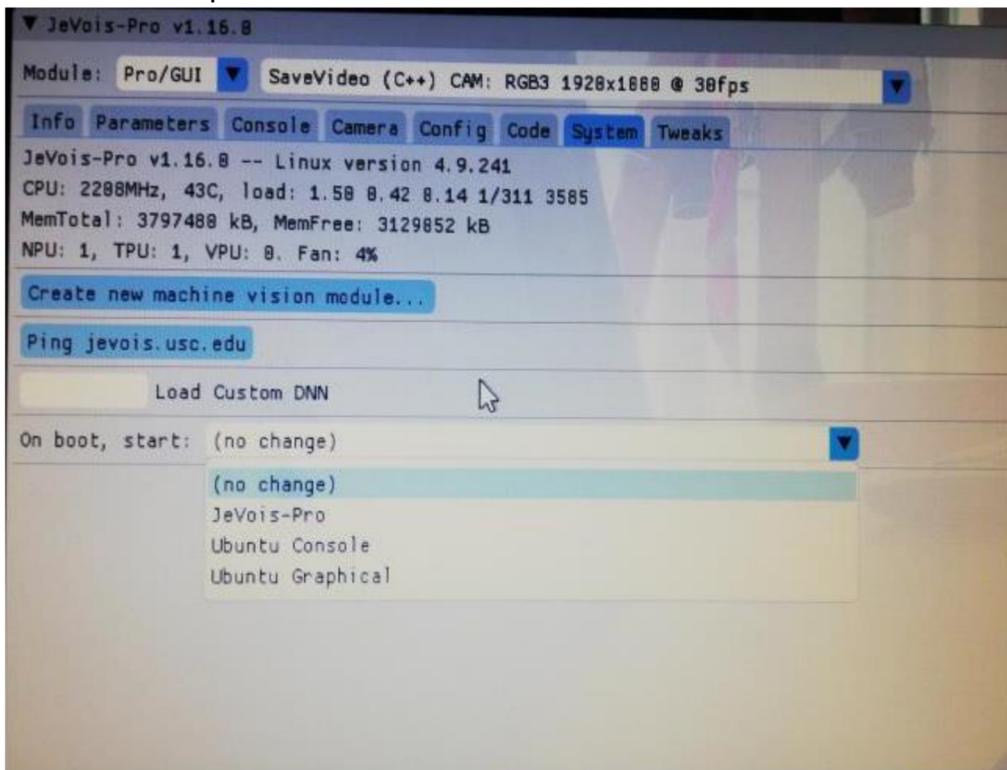
## **I. Changer d'interface**

Lorsque la caméra est branchée pour la première fois, le système boot sur l'interface JeVois-Pro : il consiste en un interface avec une console et des onglets permettant de modifier différents paramètres, et d'une vue sur le flux vidéo, permettant de visualiser les détections réalisés par les différents modules :



Il peut arriver que vous ayez besoin de changer de type d'interface, que ce soit pour coder ou modifier des fichiers de configuration. Pour cela, il faut :

1. Se rendre dans l'onglet **System** de l'interface
2. Cliquer sur le menu déroulant **On boot, start :**
3. Choisir une option :



- **JeVois-Pro** : interface de base sur lequel vous vous trouvez à ce moment
- **Ubuntu Console** : console linux du système, sans interface graphique mais simplement un shell. Peut être utile pour modifier un paramètre dans un fichier de configuration rapidement, mais déconseillé si vous n'êtes pas trop familier avec linux
- **Ubuntu Graphical** : bureau linux classique, pratique pour pouvoir utiliser la caméra comme un ordinateur, pour pouvoir coder dans un vrai IDE ou pour modifier des fichiers de configuration facilement

Une fois le mode sélectionné, vous pouvez éteindre la caméra en vous rendant dans l'onglet **Console**, et en tapant la commande :

```
shell shutdown now
```

Une fois la caméra éteinte, débrancher l'alimentation et la rebrancher. La caméra va booter sur le mode que vous avez sélectionné.

Si vous décidez de booter en mode Console / Graphique, vous aurez besoin de vous authentifier pour continuer. Les identifiants sont :

```
User : jevois  
Password : jevois
```

Si vous bootez en mode console, vous pouvez aussi vous connecter en mode administrateur. Pour cela les identifiants sont :

```
User : root  
Password : jevois
```

Pour changer de mode à nouveau, il a plusieurs façon de procéder :

1. Depuis le bureau linux, accéder à la console en appuyant sur

```
Ctrl + alt + F1
```

Cela affiche la console linux. Pour revenir sur le bureau, appuyer sur :

```
Ctrl + alt + F7
```

Une fois sur la console, vous accéder à un écran de connexion. Connecter vous en tant qu'administrateur (`root:jevois`) et tapez la commande suivante :

```
/usr/bin/jevoispro.sh
```

Cela lance l'interface JeVois-Pro classique, et vous laisse modifier le mode de démarrage.

- ⚠ **Attention à bien lancer la commande en mode Console pour éviter des bugs graphiques importants**
- 2. Si vous avez booté sur le mode Console, la marche à suivre est la même que celle-ci-dessus. Se connecter en mode administrateur, et lancer l'interface avec la commande.
- 3. Vous pouvez également changer le mode de boot depuis la Console linux ou un terminal si vous êtes sur le bureau Linux. Pour cela, il vous suffit de taper la commande suivante :

```
sudo systemctl set-default jevoispro.target #JeVois
sudo systemctl set-default multi-user.target #Console
sudo systemctl set-default graphical.target #Bureau
```

Si vous souhaitez quitter le mode JeVois pour revenir sur la console, 2 méthodes existent :

1. Dans la console, tapez la commande

```
restart
```

Cela va arrêter l'interface JeVois et vous renvoyez sur la console. Vous pouvez ensuite retourner sur le bureau si vous le souhaitez.

2. Allez dans l'onglet **System** et sur la page, cliquer sur le bouton **Create new machine vision module** puis sur **Cancel** sur le menu qui s'affiche. Cela a pour effet de faire crash l'interface JeVois, et vous renvoie sur la console. Bien que non prévue pour cet effet, cette méthode est bien rapide que la première méthode.

## II. Changer les paramètres de démarrage

Pour pouvoir utiliser la caméra dans un robot, il est important d'automatiser le démarrage de la caméra. Pour cela, il existe 2 fichiers de configuration important. Ces fichiers sont situés dans le chemin suivant :

```
/jevoispro/config/
```

Le premier fichier est le fichier **params.cfg** : il permet de régler les paramètres du port série et des trames sortantes. Seulement 3 lignes nous intéressent dans ce fichier :

```
serialdev=/dev/ttys4
serial:baudrate=460800
serial:linestyle=LF
```

La première ligne est l'adresse physique du port série.

La deuxième ligne permet d'indiquer la baudrate. Dans notre cas le baudrate était toujours mis à 460800.

La troisième ligne permet d'indiquer le caractère de fin de chaque trame :

- **LF** pour [\n]
- **CR** pour [\r]
- **CRLF** pour [\r\n]
- **Zero** pour [\0]

Plus d'information à l'adresse <http://jevois.org/doc/UserSerial.html>

**⚠ Le port utilisé sur les robots d'Eurobot et de la RoboCup est situé à l'adresse /dev/ttyS4 et non pas /dev/ttyS0 comme indiqué par le documentation JeVois**

Le deuxième fichier est le fichier `initscript.cfg` : il permet de gérer les paramètres de démarrage de la caméra et le module à utiliser. Les seules lignes à utiliser sont les suivantes (les autres doivent rester commentées) :

```
setmapping 0
setpar serout Hard
setpar serstyle Normal
```

La première ligne indique le module à utiliser. Pour connaître le numéro d'un module, il faut :

1. Accéder à l'onglet **Console** de l'interface JeVois-Pro
2. Taper la commande suivante :

```
listmappings
```

3. La liste de tous les modules s'affiche, avec tous à gauche le numéro correspondants :

```

AVAILABLE VIDEO MAPPINGS:

 8 - OUT: NONE 8x8 @ 8fps CAM: YUV 648x488 @ 30fps MOD: JeVois:DemoARtoolkit C++
 1 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 60fps MOD: JeVois:FirstVision C++
 2 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 60fps MOD: JeVois:FirstPython Python
 3 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 60fps MOD: JeVois:ObjectTracker C++
 4 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 60fps MOD: JeVois:SaveVideo C++
 5 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 30fps MOD: JeVois:RoadNavigation C++
 6 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 30fps MOD: JeVois:DemoArUco C++
 7 - OUT: NONE 8x8 @ 8fps CAM: YUV 328x248 @ 15fps MOD: JeVois:PythonObject6D Python
 8 - OUT: NONE 8x8 @ 8fps CAM: YUV 176x144 @ 120fps MOD: JeVois:FirstVision C++
 9 - OUT: NONE 8x8 @ 8fps CAM: YUV 176x144 @ 120fps MOD: JeVois:FirstPython Python
10 - OUT: RGB3 1024x512 @ 30fps CAM: RGB3 512x512 @ 30fps MOD: JeVois:PyCoralSegment Python
11 - OUT: RGB3 512x288 @ 30fps CAM: RGB3 512x288 @ 30fps MOD: JeVois:PyCoralDetect Python
12 - OUT: RGB3 320x264 @ 30fps CAM: YUV 320x248 @ 30fps MOD: JeVois:PyCoralClassify Python
13 - OUT: JVUI 8x8 @ 30fps CAM: RGB3 1920x1088 @ 30fps MOD: JeVois:SaveVideo C++
14 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1088 @ 30fps CAM2: GREY 1280x720 @ 30fps MOD: JeVois:DemoArUco C++
15 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1088 @ 30fps CAM2: GREY 1920x1080 @ 30fps MOD: JeVois:EdgeDetection C++
16 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1088 @ 30fps CAM2: GREY 1024x576 @ 30fps MOD: JeVois:MarkersCombo C++
17 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1088 @ 30fps CAM2: GREY 1280x720 @ 30fps MOD: JeVois:DemoARtoolkit C++
18 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1080 @ 30fps CAM2: RGB3 1024x576 @ 30fps MOD: JeVois:DNN C++
19 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1080 @ 30fps MOD: JeVois:DemoIMU C++
20 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1080 @ 30fps CAM2: GREY 1024x576 @ 30fps MOD: JeVois:DemoQRcode C++
21 - OUT: JVUI 8x8 @ 30fps CAM: YUV 1920x1080 @ 30fps CAM2: RGB3 1024x576 @ 30fps MOD: JeVois:DemoQRcode C++
```

Type JeVois commands here:  Input

Il suffit alors de noter le numéro de votre module et le renseigner dans le fichier `initscript.cfg` pour que la caméra démarre automatiquement sur ce module.

⚠ à chaque fois que vous ajouter / supprimer un module sur la caméra, l'ordre de mapping est modifié et les numéros ne sont plus les mêmes.

La deuxième ligne du fichier permet d'initialiser le port série. Ainsi toutes les informations du module (coordonnées, nombre d'objets détectés...) seront envoyées sur le port série.

La troisième ligne indique le format des trames à utiliser :

- Normal : N2 ID Xcenter Ycenter Width Height
- Detail : D2 ID x1 y1 x2 y2 x3 y3 x4 y4

Plus d'infos à l'adresse <http://jevois.org/doc/UserSerialStyle.html>

- Il y a certaines particularités à noter :
  - le mode Detail du module ArUco en C++ est constitué de la manière suivante : D2 ID 4 x1 y1 x2 y2 x3 y3 x4 y4 (4 étant le nombre de points)
  - le mode Detail du module DNN renvoie 2 fois les coordonnées du 2<sup>ème</sup> point soit D2 ID x1 y1 x2 y2 x3 y3 x2 y2
  - le mode Normal du module DNN ne renvoie pas les coordonnées du centre de l'objet, mais les coordonnées du point en haut à gauche de la boîte englobant l'objet
  - pour le module DNN, ID corresponds à « classe:%confiance »

Il est également possible de gérer des paramètres propre à chaque module. Par exemple, pour le module ArUco en C++, l'option « Dopose » permettait de calculer l'inclinaison du tag. Pour initier ce paramètre, il suffit d'ajouter cette ligne dans le fichier `initscript.cfg` :

```
setpar dopose true
```

Il en va de même pour tous les autres paramètres.

Concernant le module ArUco, nous avions eu besoin de spécifier le dictionnaire utilisé pour pouvoir détecter le numéro sur les tags ArUco de la coupe de France. Pour cela, il faut créer le fichier `params.cfg` dans le dossier suivant :

```
/jevoispro/modules/JeVois/DemoArUco/
```

Dans ce fichier, il faut simplement écrire la ligne suivante :

```
dictionary = D4X4_100
```

Ainsi le module de détection ArUco pourra détecter des tags avec des valeurs plus grande.

Pour tous ces fichiers de configuration, les modifications peuvent être faites en branchant la carte SD de la JeVois dans un ordinateur Windows. La carte SD est divisé en 3 partitions :

- La partition **boot** qui contient les fichiers de démarrage du système. Il ne faut pas y toucher sinon la caméra ne démarrera plus
  - La partition **jevoispro** qui contient les modules de la caméra et les fichiers de configuration
  - La partition **rootfs** qui contient tous les fichiers du système d'exploitation.
- 
- La partition **rootfs** partition est partitionnée avec le système de fichier ext4 (le système de fichier de linux) qui ne peut pas être lu par Windows. Quand la carte SD est branché dans un ordinateur Windows, celui-ci va vous proposer de formater la partition puisqu'il n'arrive pas à la lire. Il ne faut surtout pas accepter et simplement cliquez sur « refusez ». Il n'y a pas besoin d'accéder à cette partition pour modifier les fichiers de configuration.

Il peut cependant y avoir besoin d'accéder aux fichiers de la partition **rootfs**. Le seul moyen est d'utiliser un ordinateur linux ou de modifier ces fichiers directement depuis la caméra, que ce soit en mode console ou en mode graphique.

Par exemple, pour passer le clavier en azerty (clavier par défaut en qwerty), il faut :

- Aller dans le dossier `/etc/default/`
- Ouvrir le fichier `keyboard` et modifier la ligne suivante :

XKBLAYOUT=«us»

Par la ligne

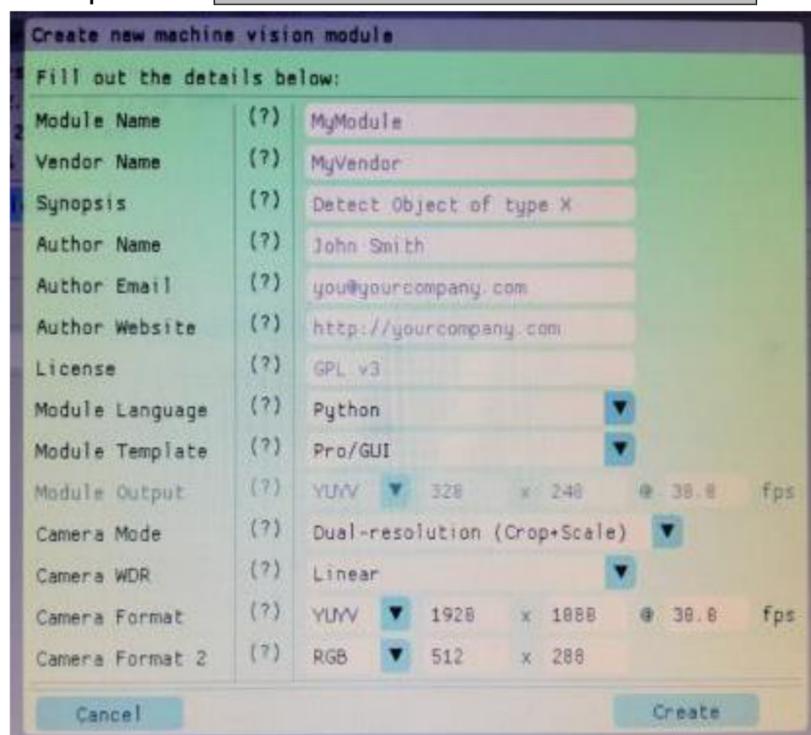
XKBLAYOUT=«fr»

Après avoir sauvegarder le fichier et redémarré la caméra, la disposition de clavier par défaut sera en azerty.

### III. Créer un nouveau module sur le JeVois

L'interface JeVois-Pro offre la possibilité d'ajouter ses propres modules de traitement d'image en python, avec des nombreuses bibliothèques préinstallées comme OpenCV, numpy,...

Pour créer un nouveau module, il faut se rendre dans l'onglet `System` de l'interface puis cliquer sur `Create new machine vision module` :



Pour créer un module, il suffit simplement de lui donner un nom (Module Name) et un auteur (Vendor Name). Les autres informations sont facultatives et seulement à titre informatif. Une fois le module créé, le fichier de code en python est généré dans le dossier suivant :

`/jevoispro/modules/VendorName/ModuleName/ModuleName.py`

Le fichier généré est un exemple d'utilisation de la caméra en python, à vous de le modifier à votre guise. Par exemple, mon module de tracking de la balle avec OpenCV ou mon module de collecte d'images pour un dataset en format JeVois peuvent être trouvés à l'adresse suivante :

[https://github.com/Theophile-Wemaere/Stage-GEII/tree/main/JeVois\\_modules](https://github.com/Theophile-Wemaere/Stage-GEII/tree/main/JeVois_modules)

Plus d'information à l'adresse <http://jevois.org/doc/ModulePythonTutorial.html>

#### **IV. Ajouter un nouveau modèle YOLO sur la JeVois**

La caméra JeVois permet également de faire tourner des DNN (pour Deep Neural Network). Pour cela, 2 modules DNN existent, un avec une résolution de 1920x1080 et un avec une résolution de 640x480. La caméra est fournie avec des exemples de DNN détectant un certains nombre de classes, mais il est bien évidemment possible d'utiliser ses propres modèles en fonction de ce que l'on souhaite détecter (des robots par exemple).

Pour cela, une fois l'entraînement de votre réseau terminé, rendez-vous sur le site <https://jevois.usc.edu/mc/> qui est le convertisseur officiel de JeVois pour les DNN. Sur la page, les informations à rentrer sont :

- Votre email (pour recevoir le modèle converti)
- Le nom du modèle (ex : yolov3-tiny-ball)
- Le processeur utilisé. Pour YOLO, on choisit le NPU.
- La source d'entraînement. Pour YOLO, il s'agit de Darknet
- Le type du modèle, dans notre cas « Object detection »
- Le type de détection. Pour Yolo, choisir la version utilisée (ex : yolov3-tiny)
- La taille d'entraînement (input size) spécifié dans votre fichier .cfg
- Le fichier .cfg utilisé pour entraîner votre modèle
- Le fichier .weights de votre modèle
- Le fichier .name (avec le nom de chaque classe) de votre modèle
- Une image sur laquelle le modèle s'est entraîné. Si possible, choisir une image ressemblant à celle que le modèle verra sur la caméra (donc une photo prise de la caméra par exemple). Ne pas oublier d'indiquer le nombre de classe présente sur l'image.

Tous les autres paramètres peuvent être laissés tels quels.  
Une fois la conversion terminée, vous recevrez par mail votre modèle dans un fichier .zip (cela ne prends pas plus de 10minutes).

Pour utiliser votre modèle dans la JeVois, il suffit d'extraire tous les fichiers du document zip dans le dossier suivant :

/jevoispro/share/dnn/custom/

Une fois votre dossier extrait, vous devriez normalement avoir 5 fichiers dans le dossier custom :

- Un fichier .json
- Un fichier .nb
- Un fichier .yml
- Deux fichier .txt

Le fichier model\_name-classes.txt contient le nom de chaque classe, et peut être modifié si vous souhaitez modifier le nom d'une classe.

- Si vous souhaitez automatiser le lancement du module DNN sur la JeVois, la méthode vue précédemment ne fonctionne pas entièrement pareil. Les paramètres du modules ne seront pas réglés dans le dossier /jevoispro/config/initscript.cfg mais directement dans le fichier model\_name.yml dans le dossier custom/

Les paramètres de démarrage comme le numéro du module, le format de la trame seront toujours dans le dossier initscript.cfg, mais les paramètres du modules sont à spécifiés dans le fichier model\_name.yml. Pour le module DNN, il est important de spécifier un threshold minimum (confiance minimum de détection) et un nms (Non Maximal Suppression) minimum. Bien qu'il s'agisse des paramètres du modules, ils ne peuvent pas être configurer avec l'option « setpar » dans le fichier initscript.cfg , mais à la place doivent être spécifié dans le fichier model\_name.yml. Par exemple, si l'on veut un threshold minimum de 25% et un nms maximum de 10% il faut ajouter les lignes suivantes le fichier .yml :

```
nms: "10"  
thresh: "25"
```

Le fichier ressemblera donc à la capture suivante :

```

1 %YAML 1.0
2 ---
3
4 yolov3-tiny-ball-robot:
5   preproc: Blob
6   nettype: NPU
7   postproc: Detect
8   detecttype: RAWYOL0v3tiny
9   model: "dnn/custom/yolov3-tiny-ball-robot.nb"
10  intensors: "NCHW:8S:1x3x540x960:DFP:7"
11  outtensors: "NCHW:8S:1x21x17x30:DFP:2,NCHW:8S:1x21x34x60:DFP:1"
12  rgb: true
13  nms : "10"
14  thresh: "25"
15  mean: "127.5 127.5 127.5"
16  scale: ".0078431372"
17  classes: "dnn/custom/yolov3-tiny-ball-robot-classes.txt"
18  processing: Async
19

```

Ainsi, lorsque le module démarrera, les paramètres thresh et nms seront initialisés à 25% et 10% chacun.

## V. Notions et commandes linux de base

Ci-dessous une liste de commande linux pouvant servir sur la caméra. Ces commandes peuvent être lancées depuis la console linux ou depuis un terminal en mode graphique. Elles peuvent être utilisé depuis la console JeVois avec la syntaxe suivante :

shell votre commande

Eteindre la caméra	shutdown now
Redémarrer la caméra	reboot
Supprimer un fichier ou un dossier	rm -rf votre_dossier_ou_fichier
Copier un fichier	cp /chemin/fichier /chemin/nouveau/fichier
Copier un dossier	cp -r /chemin/dossier /nouveau/dossier
Déplacer un fichier/dossier	mv /chemin/dossier /nouveau/chemin
Renommer un fichier/dossier	mv nom_dossier nouveau_nom
Changer de dossier	cd /chemin/du/dossier
Remonter d'un dossier	cd ..
Afficher le dossier où l'on se trouve	pwd
Lister les éléments dans un dossier	ls (dossier actuel) ou ls /chemin/dossier
Lister les éléments cachés	ls -a

Changer clavier vers azerty (1)	loadkeys fr
Changer clavier vers azerty (2)	setxkbmap fr
Lancer une commande en admin	sudo votre_commande
Afficher le nom d'utilisateur	whoami
Passer en mode admin	sudo su (exit pour quitter)
Passer en mode user (depuis admin)	su jevois (exit pour quitter)
Quitter la console/terminal	exit

Les 2 méthodes pour changer la disposition du clavier vu ci-dessus ne sont pas permanente. La première change la disposition pour la console linux, et la 2<sup>ème</sup> pour tous le système, mais ne peut être lancé que depuis un terminal de l'interface graphique. Contrairement à la méthode vue plus tôt dans ce guide, la disposition prend effet immédiatement mais est oubliée lorsque la caméra redémarre.

#### Lancer l'interface graphique depuis la console :

Si vous avez bootez sur la console mais que vous voulez accéder au bureau linux sans repassez redémarrer la JeVois, il suffit de taper la commande suivante :

```
startx
```

Cela a pour effet de démarrer l'environnement graphique Linux.

Si vous étiez connecté en tant qu'admin, ce serait la session admin. Sinon ce sera la session de l'utilisateur jevois.

Comme vous avez lancé le mode graphique depuis la console, l'environnement graphique linux tourne sur le tty n° 1. Donc si vous souhaitez réaccéder à la console, le raccourci clavier n'est plus

```
Ctrl + alt + F1
```

mais

```
Ctrl + alt + F3
```

Par défaut, la console Linux tourne sur le tty n°1 et l'environnement graphique sur le tty n° 7 (d'où F1 et F7), mais si l'on force le démarrage du bureau depuis la console (donc tty n°1), le bureau tourne sur le tty N°1 et vous pouvez accéder à la console sur le tty n° 3. Pour faire simple, un TTY est une session sur linux, où l'on peut faire tourner des process, des sessions

graphiques ou en ligne de commande. Il est possible de voir tous les utilisateurs connecté sur une session tty avec la commande who

### Utiliser un éditeur de texte :

Par défaut, il y a 3 éditeur installé sur les caméras JeVois : vim, nano et gedit. vim et nano sont des éditeurs en ligne de commandes, cela signifie qu'il n'y a pas besoin d'interface graphique pour les utiliser, et ils permettent de modifier des fichiers depuis une console/terminale. Gedit est un simple éditeur de texte graphique, comme l'application Bloc-Note sur Windows10.

#### **Vim :**

*Vim peut être compliqué à utiliser pour un débutant. Pour que ce tutoriel soit clair, les touches représentées entre [ ] comme la touche [i] sont des touches à presser pour utiliser les différents modes de vim.*

Pour utiliser vim dans un terminale/console, taper la commande suivante :

```
vim fichier_a_modifier
```

Vim s'ouvre avec le contenu du fichier. Vous êtes en mode normal.

- Pour passer en mode édition, appuyer sur la touche [i], vous passer en mode insertion (le mode est affiché en bas). Vous pouvez alors modifier le texte, et vous déplacer avec les flèches directionnelles.
- Pour quitter le mode insertion, appuyer sur la touche [Echap] pour retourner en mode normal
- Pour sauvegarder les modifications, passer en mode normal, puis appuyer sur la touche [:], cela a pour effet d'ouvrir la console vim (en bas du terminale/console) puis taper [w] ([w] pour write = sauvegarder) et [Enter] pour valider.
- Pour quitter, appuyer sur [q] ou ([q][!] si vous souhaiter quitter sans sauvegarder) puis [Enter] pour valider.
- Il est possible de sauvegarder et quitter depuis le mode normal avec l'enchainement de touche suivante : [:][w][q][Enter]

Si un fichier est protégé en écriture, l'ouvrir avec la commande suivante :

```
sudo vim nom_fichier
```

## **Nano :**

Nano est un autre éditeur de texte en ligne de commande, qui peut être plus simple à utiliser pour certains.

Pour ouvrir un fichier avec nano, utiliser la commande suivante :

```
nano nom_fichier  
ou  
sudo nano nom_fichier
```

Vous pouvez ensuite directement modifier le fichier, et déplacer le curseur avec les flèches directionnelles. Pour quitter nano, appuyer sur :

Ctrl + X

On vous demandera alors si vous souhaitez sauvegarder le document. Taper sur la touche [Y] sur oui, et sur la touche [N] si non. Si oui, on vous demandera si vous souhaitez sauvegarder dans le fichier actuelle ou en créer une copie. Appuyer sur [Enter] pour valider.

## **Gedit :**

Gedit est un éditeur de texte graphique très simple à utiliser. Depuis une session graphique, vous pouvez utiliser un explorateur de documents et cliquer sur « open with » puis choisir « text editor ». Depuis un terminal, vous pouvez taper :

```
gedit /chemin/fichier  
ou  
sudo gedit /chemin/fichier
```