Rapport d’avancement :

Description du sujet :

Réaliser un système de transmission intelligent à base de la carte Orange Pi / Raspberry Pi, qui envoie des photos et des vidéos en streaming via les trois générations : 2G ,3G et la 4G.

Effectuer le traitement des données reçues.

Partie théorique :

1. Outil de streaming :

Ffmpeg :

FFmpeg est une collection de logiciels libres destinés au traitement de flux audio ou vidéo (enregistrement, lecture ou conversion d'un format à un autre).

Il peut supporter plusieurs formats conteneurs audio/vidéo comme AVI, MPEG…, ainsi qu’une multitude de formats de compression tels que MPEG-4 AVC (H.264), MPEG-4 HEVC (H.265)… pour la vidéo, et JPEG, PNG… pour l’image.

Ffmpeg pour le streaming :

FFmpeg peut diffuser par l’un des deux moyens suivants : soit il envoie le flux vers un «autre serveur», qui le renvoie à plusieurs clients, soit le transmet via UDP / TCP directement à un destinataire unique.

2. Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning:

On peut considérer le deep learning, le machine learning et l'intelligence artificielle comme un ensemble de poupées russes imbriquées les unes dans les autres. Le deep learning est un sous-ensemble du machine learning, et le machine learning est un sous-ensemble de l’AI, terme générique désignant tout programme informatique intelligent. En d'autres termes, tout machine learning est une intelligence artificielle, mais pas toute intelligence artificielle un machine learning, et ainsi de suite.

Les domaines d’application et usages potentiels d’une Intelligence Artificielle sont de plus en plus divers : compréhension du langage naturel, reconnaissance visuelle, robotique, système autonome, Machine Learning …

Le deep learning :

Le DL (qui est donc un sous-domaine du ML) repose sur ce qu’on appelle des réseaux de neurones artificiels (neural networks), c’est-à-dire un ensemble de neurones (ce sont de petites calculatrices qui effectuent une opération mathématique) qui s’envoient des nombres en fonction de leurs liaisons, jusqu’à des neurones de sortie. Grâce à cette architecture, le DL est capable de reconnaître des visages, de synthétiser des textes ou encore de conduire une voiture autonome !

Un algorithme va adapter les liaisons entre ses neurones (il va les renforcer ou les détruire), pour qu’en sortie on ait une bonne approximation des données d’entrée.

Il existe plusieurs algorithmes de deep learning, tant qu'on s'intéresse précisément au 'traitement d'images', on utilise les réseaux de neurones convolutifs (Convolutional Neural Networks CNN), ils sont spécialisés dans le traitement de l'image, ils appliquent des filtres à des données pour en faire ressortir de nouvelles informations (par exemple, faire ressortir les contours dans une image peut aider à trouver où est le visage).

DL frameworks :

Étant donné que le DL (deep learning) est la clé pour l'exécution de tâches d'un niveau de sophistication supérieur, leur construction et leur déploiement s'avèrent être un défi pour les scientifiques et les ingénieurs de données (data engineers) du monde entier. Nous disposons aujourd'hui de plusieurs frameworks qui nous permettent de développer des outils pouvant offrir un meilleur niveau d'abstraction tout en simplifiant les défis de programmation difficiles. Chaque framework est construit de manière différente à des fins différentes.

Le framework que j'ai choisi utiliser est : Tensorflow

Un des meilleurs frameworks du DL qui a été adopté par plusieurs géants tels que Airbus, Twitter, IBM et d'autres, principalement en raison de la grande flexibilité de son architecture système.

Le cas d'utilisation le plus connu de TensorFlow doit être Google Translate, associé à des fonctionnalités telles que le traitement du langage naturel, la classification / synthèse de texte, la reconnaissance parole / image / écriture…

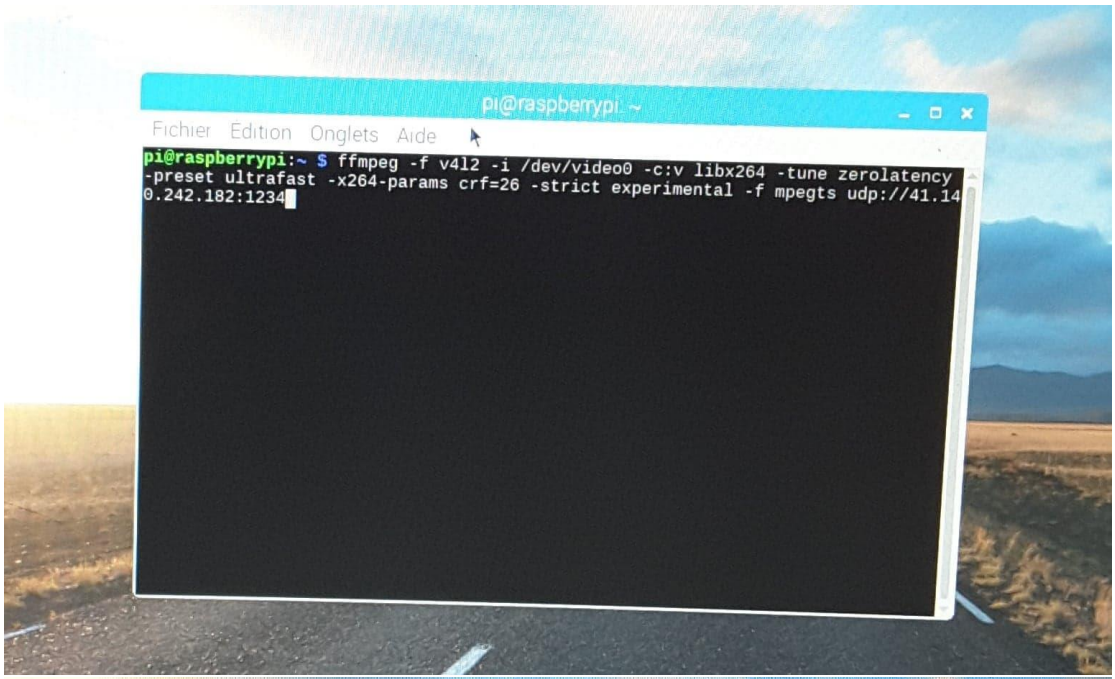
TensorFlow est disponible à la fois sur le bureau et sur le mobile et prend également en charge des langages tels que Python, C ++ et R pour créer des modèles du DL ainsi que des bibliothèques d'encapsulation.

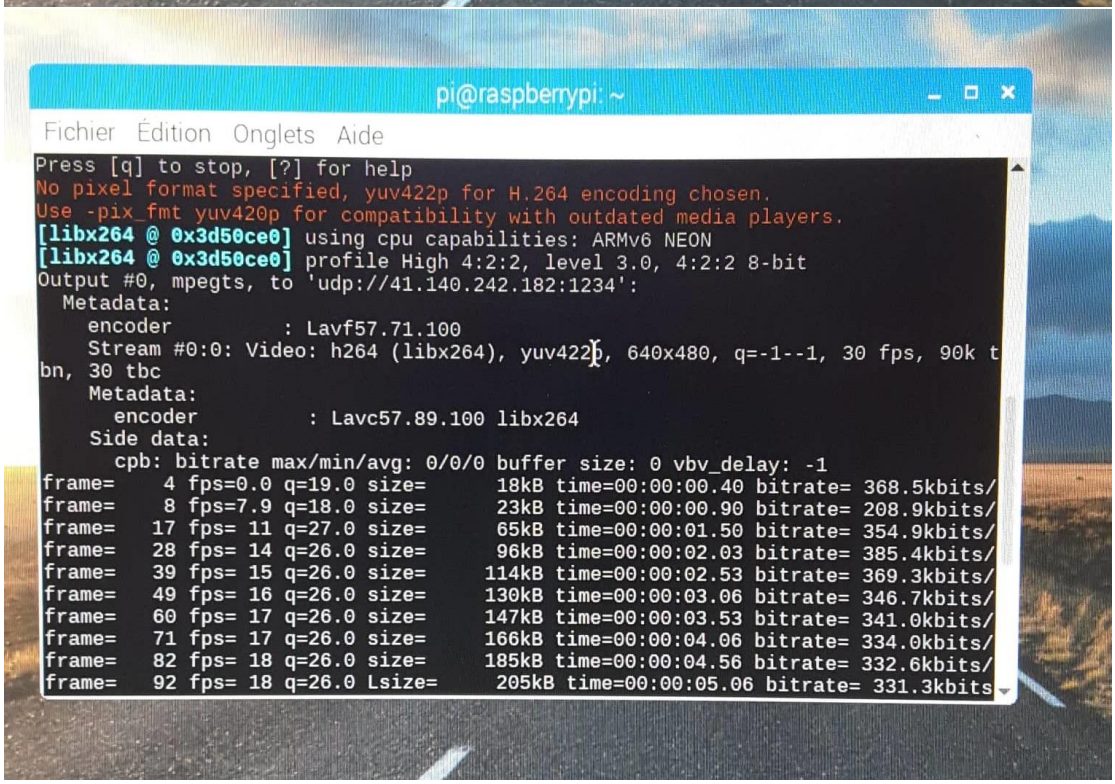
Application :

1. Traitement au niveau du Raspberry Pi

1.1. Streaming + compression :

Envoyer le stream avec ffmpeg (au niveau du rasp) et le recevoir avec opencv-python (au niveau du pc).

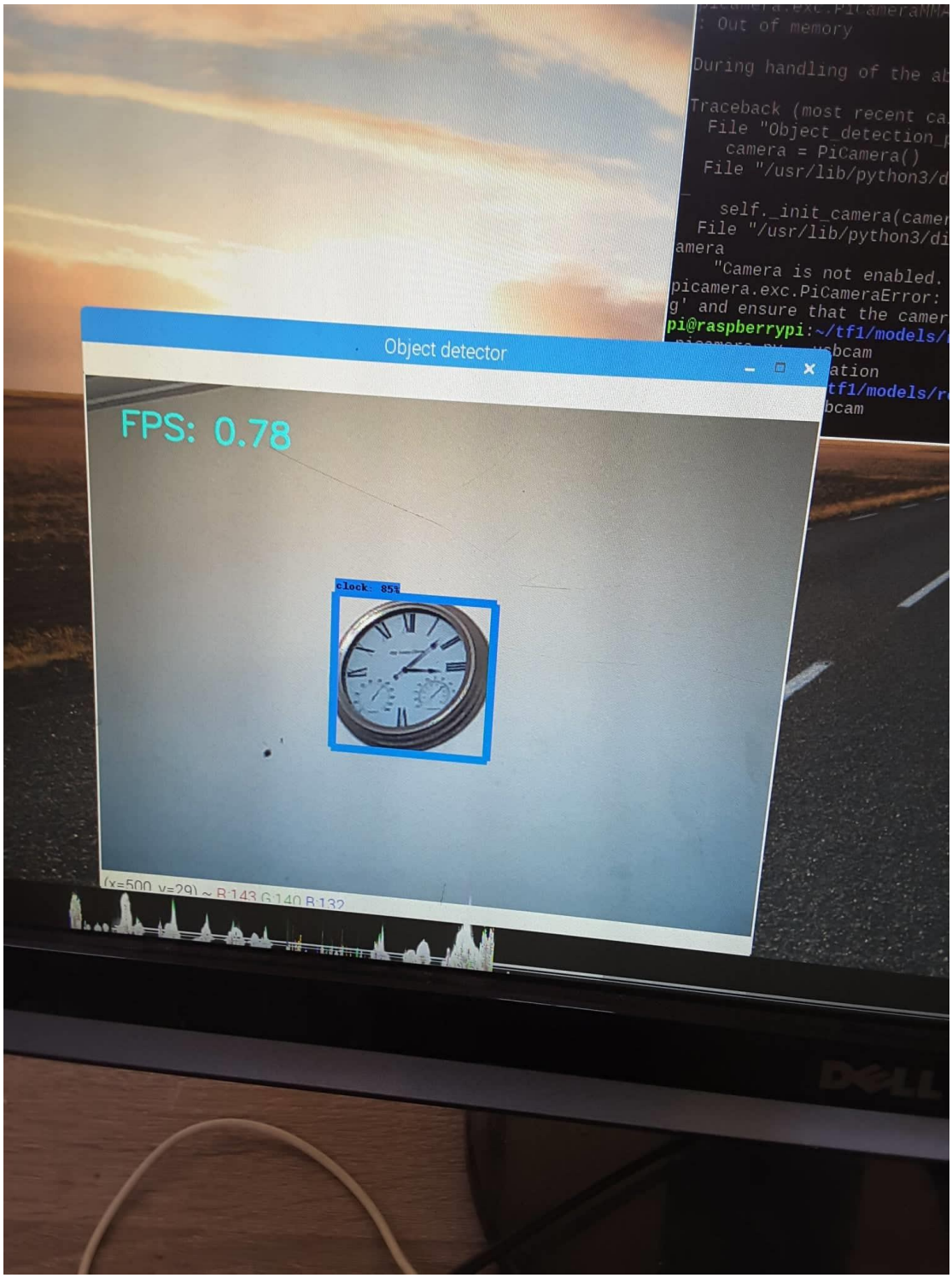




Réception :

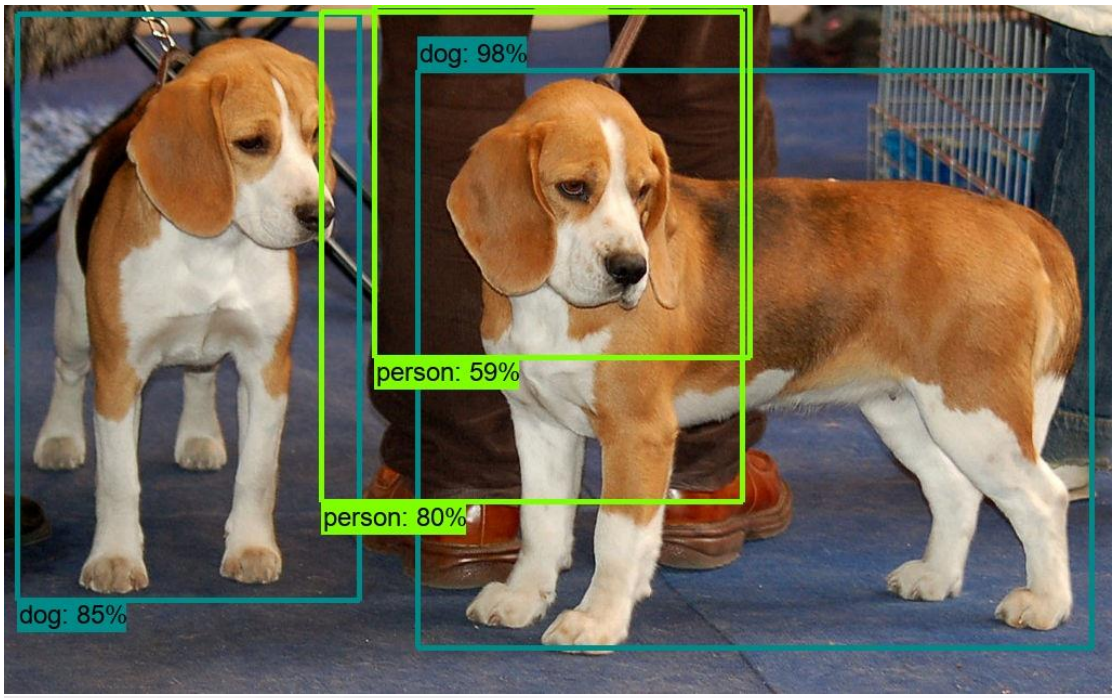
```
C:\Users\HP>ffplay udp://192.168.1.200:1234
ffplay version 3.3.4 Copyright (c) 2003-2017 the FFmpeg developers
  built with gcc 7.2.0 (GCC)
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-bzlib --e
nable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libblur
ay --enable-libfreetype --enable-libmp3lame --enable-libopenjpeg --enable-libopu
s --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libtheora --ena
ble-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --enable-lib
x264 --enable-libx265 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp
--enable-libvidstab --enable-cuda --enable-cuvid --enable-d3d11va --enable-nvenc
 --enable-dxva2 --enable-avisynth --enable-libmfx
  libavutil      55. 58.100 / 55. 58.100
  libavcodec     57. 89.100 / 57. 89.100
  libavformat    57. 71.100 / 57. 71.100
  libavdevice    57.  6.100 / 57.  6.100
  libavfilter     6. 82.100 /  6. 82.100
  libswscale      4.  6.100 /  4.  6.100
  libswresample   2.  7.100 /  2.  7.100
  libpostproc    54.  5.100 / 54.  5.100
[h264 @ 0000000000521ca0] non-existing PPS 0 referenced  0B f=0/0
    Last message repeated 1 times
[h264 @ 0000000000521ca0] decode_slice_header error
[h264 @ 0000000000521ca0] non-existing PPS 0 referenced
[h264 @ 0000000000521ca0]     nan    :  0.000 fd=   0 aq=    0KB vq=    0KB sq=
decode_slice_header error
```

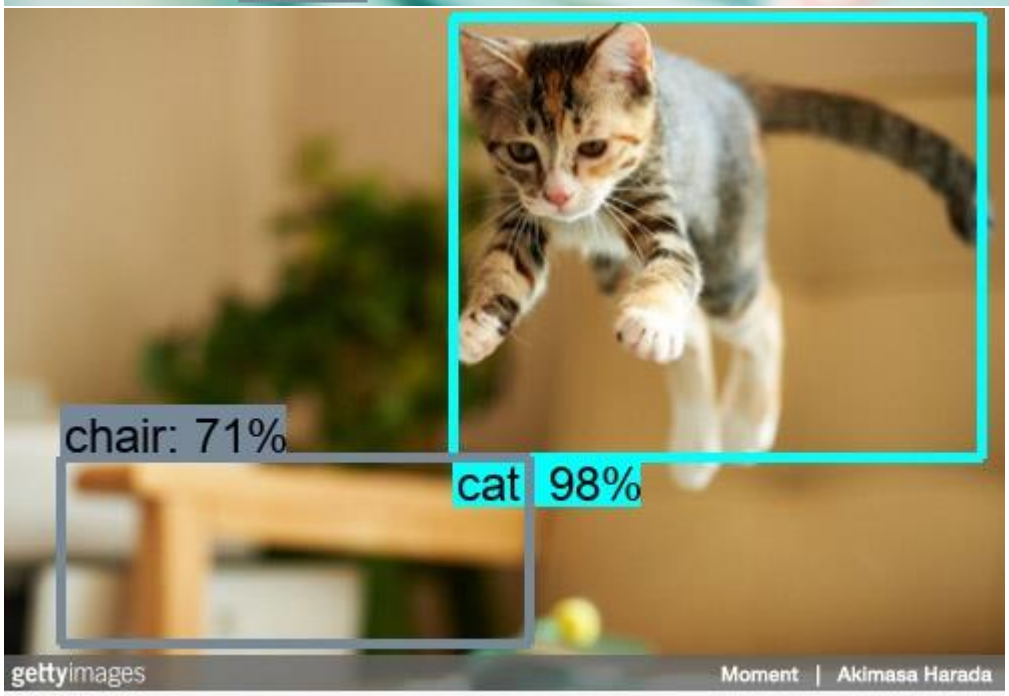
1.2. Traitement de la vidéo :



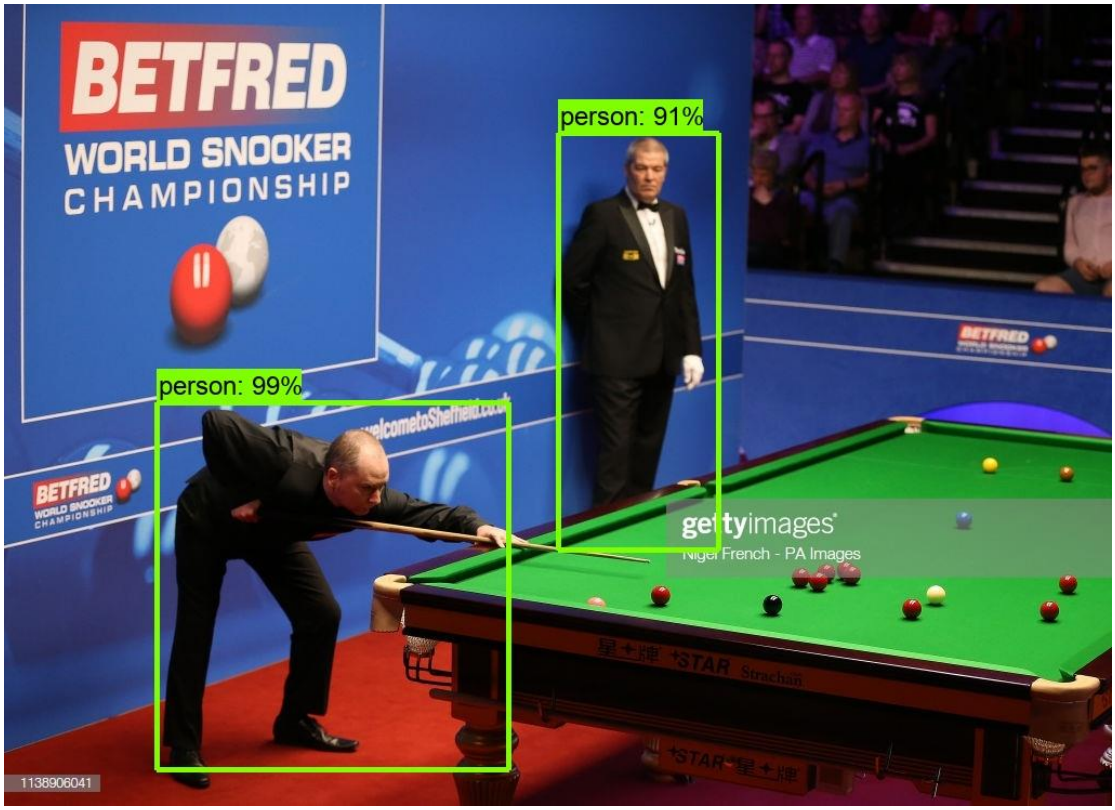
2. Traitement au niveau du pc

Images :

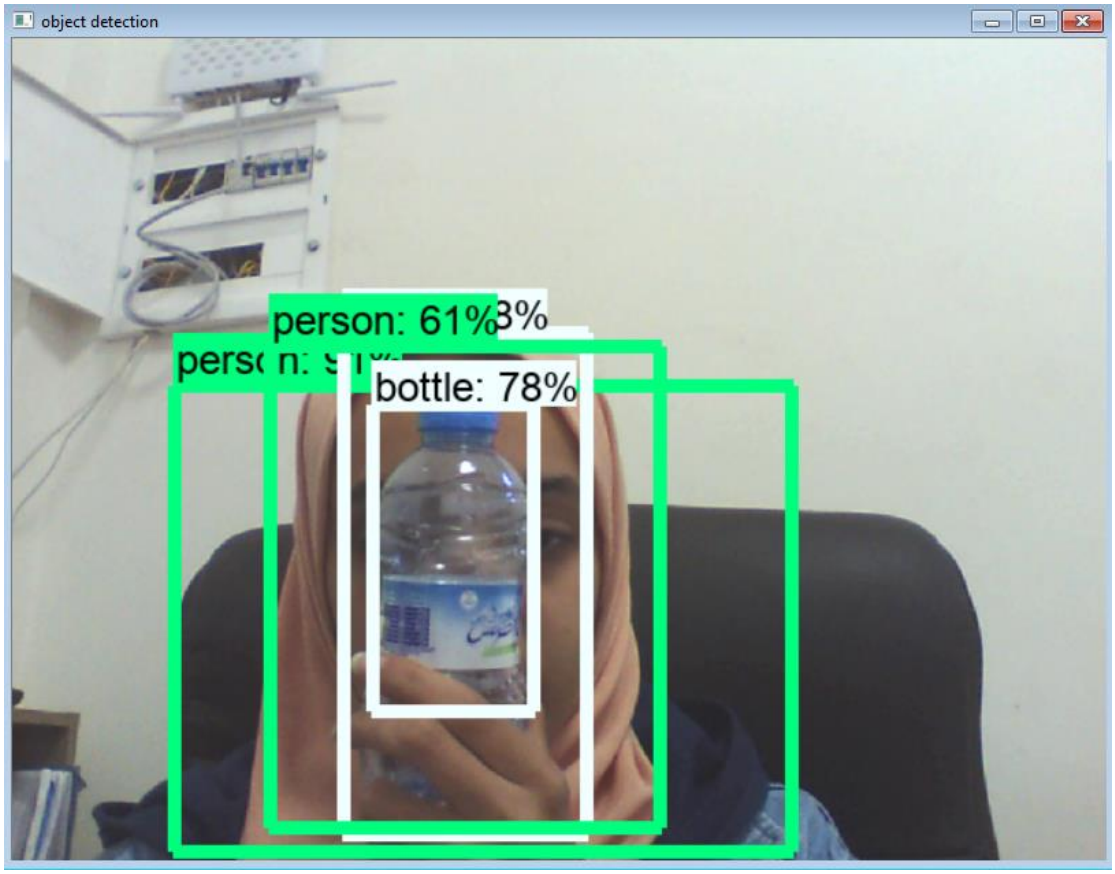




Sample of an embedded image


BETFRED
WORLD SNOOKER
CHAMPIONSHIP
person: 91%
person: 99%
gettyimages
Nigel French - PA Images
1138906041

Vidéo :


object detection
person: 61%
bottle: 78%