

RAPPORT PROJET IOT & ML

# EXAM MONITORING SYSTEM

## I. INTRODUCTION

Dans le cadre de notre seconde année Master EID2, il nous est proposé un projet nous permettant de mettre en pratique nos connaissances et nos compétences professionnelles au travers d'un cahier des charges ayant pour finalité la conception et le développement d'un système de surveillance des examens en ligne.

Ayant une passion commune pour le *IOT et ML* , notre groupe composé de *Hajar Fahsi* et *Mouaad Kriouile* , on a saisi l'opportunité d'exploiter cet intérêt commun pour soumettre l'ébauche d'un projet personnel innovant au responsable de la matière Monsieur Aomar Osmani.

## II. BESOINS ET OBJECTIFS DU PROJET

Le projet consiste principalement à concevoir un système de surveillance des examens en ligne.

La surveillance à distance peut fonctionner de différentes manières selon le type de solution et les exigences de l'examen en ligne.

Notre système est basé sur une surveillance en ligne en direct, le surveillant dédié surveille l'étudiant et son environnement à distance via un flux en direct. À sa discrétion, il peut suspendre ou mettre fin au test s'il soupçonne un comportement suspect.

Les outils requis afin de mettre en place notre système sont un **ordinateur portable** de l'étudiant avec une **caméra et un micro** qui fonctionnent correctement ainsi qu'un **téléphone portable** avec une caméra qui fonctionne. Ce dernier nous servira comme détecteur des objets qui se trouvent dans l'environnement de l'étudiant.

## II. ASPECTS TECHNIQUES

Tout comme la surveillance traditionnelle dans un centre d'examen, le logiciel de surveillance des examens garantit que les tests à distance sont effectués de manière fiable, équitable et standardisée.

Une fois que le candidat se connecte à l'examen, notre système utilise une authentification biométrique sous forme d'une reconnaissance faciale du candidat via la webcam de son ordinateur pour vérifier son identité à partir de la base de données pré-enregistrée. Une fois approuvé, le candidat peut commencer le test.

Les données sont récupérées et vérifiées tout au long du test, à partir de la webcam et la caméra du téléphone du candidat, ainsi que son micro. Nous allons détailler dans la partie suivante les fonctionnalités qui ont été mises en place afin de garantir que le test à distance soit effectué de manière fiable, équitable et standardisée, tout comme la surveillance traditionnelle dans un centre d'examen.

Afin d'implémenter nos fonctionnalités, on a utilisé le langage **python** pour gérer tout ce qui concerne la webcam et le micro de l'ordinateur et le langage **Swift** pour gérer les données qui proviennent de la caméra du téléphone.

La communication entre le serveur qui est implémenté en python et le client (Swift) se fait à travers **les sockets**. La communication débutera dès que le candidat commence l'examen, le client envoie en permanence des messages au serveur pour l'informer que la caméra du téléphone est toujours allumée et reçoit des informations, si ce n'est plus le cas, le serveur considère cela comme cas de triche et arrête l'examen, la même chose pour le client.

On a utilisé la bibliothèque **OpenCV** afin de récupérer les données de la webcam et les bibliothèques **face\_recognition** et **speech\_recognition** de python afin d'utiliser la reconnaissance faciale et vocale.

En ce qui concerne Swift, il n'existe pas vraiment une bibliothèque qui sert à détecter les objets (personne, téléphone, feuille ..), on a dû donc trouver un moyen de convertir un modèle implémenté en python en Swift qui s'appelle **YOLOv3Tiny**. Afin de récupérer les données de la caméra du téléphone on a implémenté les fonctions de la classe **AVCaptureVideoDataOutputSampleBufferDelegate**.

Swift a également été utilisé afin de développer les interfaces de l'application du téléphone du candidat.

### III. FONCTIONNALITÉS

Notre système dispose de nombreuses fonctionnalités qu'on a mises en place afin d'assurer la fiabilité de l'examen, on s'est mis d'accord sur les conditions du déroulement de l'examen, une fois une des règles est enfreinte l'examen est immédiatement interrompu par le serveur.

on vous présentera dans la suite les cas de fraudes reconnues par notre système.

#### RECONNAISSANCE DES OBJETS

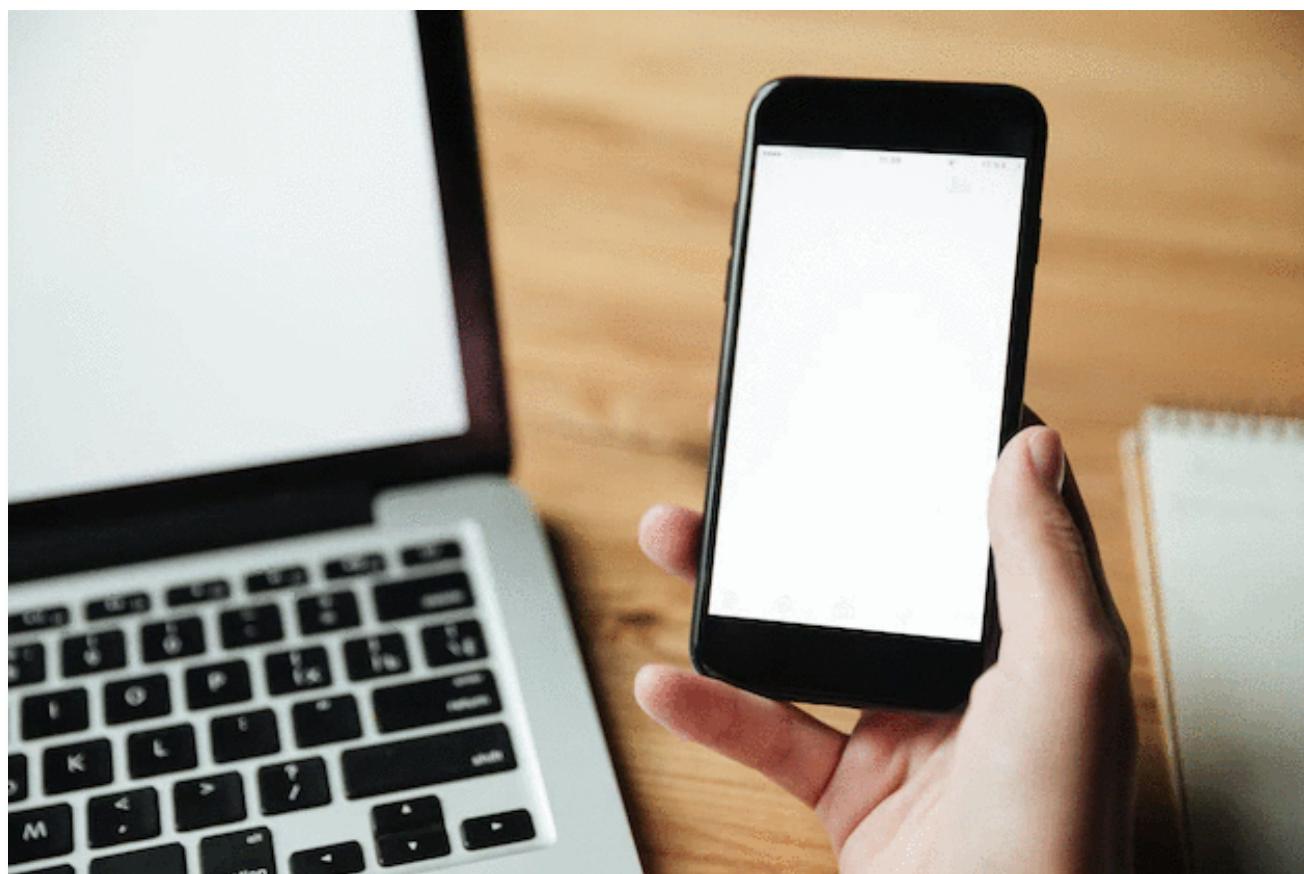
Le candidat a le choix d'utiliser le téléphone autant qu'une caméra ou de l'utiliser autant qu'un détecteur de mouvement, le système démarre une fois qu'une des deux options est choisie.

The image shows a smartphone screen displaying the 'ExamMonitoringSystem' application. The top status bar shows the time as 02:35, signal strength, and battery level. The main interface has a light gray header with the text 'ExamMonitoringSystem'. Below the header, there is a section titled 'CHOOSE PHONE ROLE' and another titled 'BY FOLLOWING THE INSTRUCTIONS BELOW'. At the bottom of the screen, there are two blue rounded rectangular buttons labeled '3 : HEAD' and '2 : HAND' respectively. In the center of the screen, there is an illustration of two people sitting at desks, each using a laptop. Red numbers 1, 2, and 3 are placed above their heads and hands to indicate the monitoring zones for head and hand movement detection.

La reconnaissance des objets a été implémenté sur Swift, elle est capable de détecter plus que 1000 objets. On a mis en place des conditions afin d'interdire que certains objets se trouvent autour du candidat lors de l'examen ( téléphone portable, feuille avec écriture manuscrite, books et notamment une autre personne ...

Voici un exemple d'une mise en situation où la caméra de téléphone a pu détecter un téléphone portable qui se trouve dans son ongle de vue, puis envoie un message au serveur, ce dernier a arrêté l'examen.

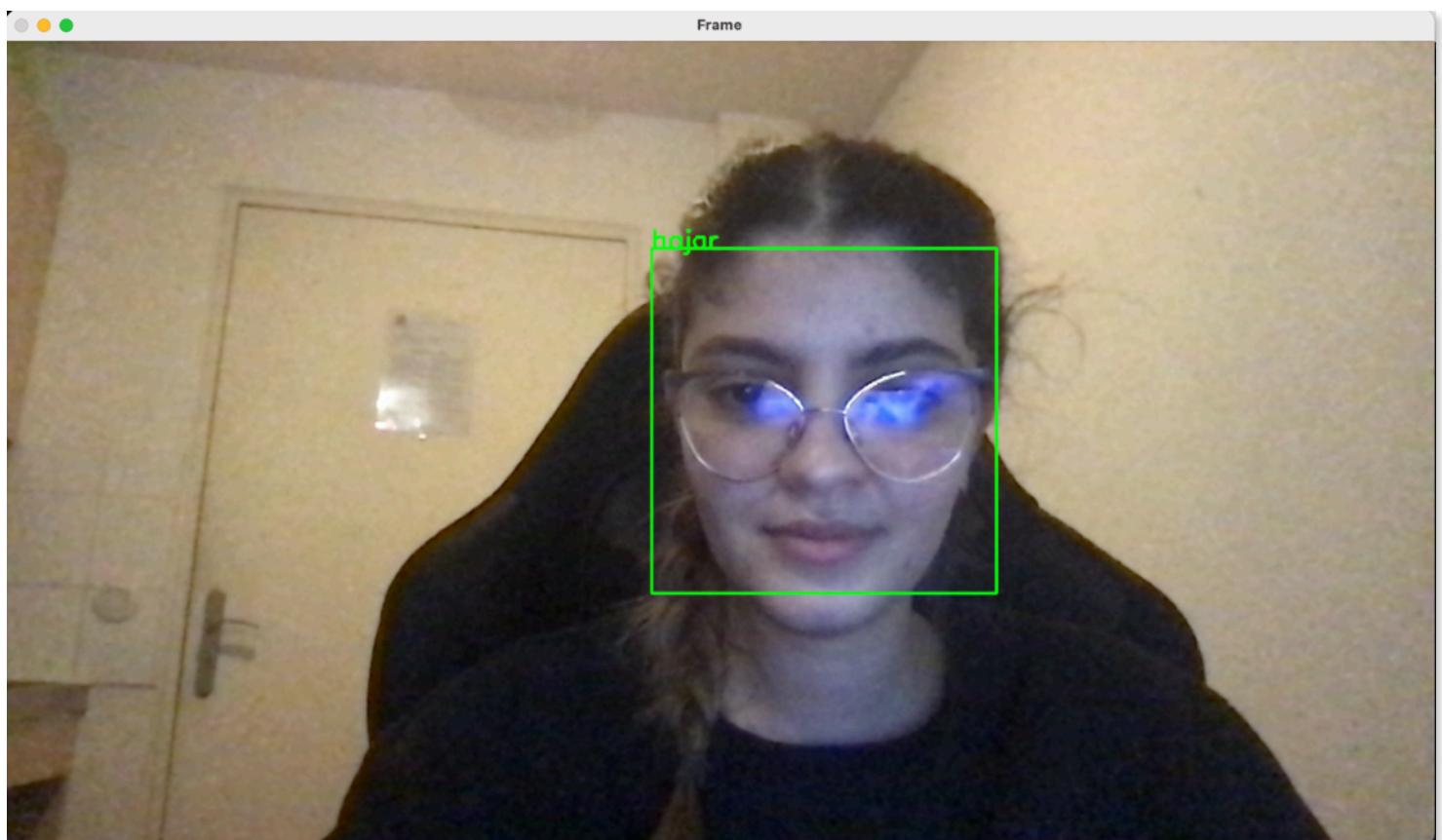
```
detected laptop confidence 0.9740106
detected laptop confidence 0.97960615
detected laptop confidence 0.97610795
detected laptop confidence 0.9735477
detected laptop confidence 0.9705416
detected laptop confidence 0.9641013
detected laptop confidence 0.97309536
detected laptop confidence 0.9742409
detected tvmonitor confidence 0.6052256
detected laptop confidence 0.97121435
detected tvmonitor confidence 0.6501806
detected tvmonitor confidence 0.66133076
detected tvmonitor confidence 0.6989486
detected tvmonitor confidence 0.63125885
detected laptop confidence 0.9616442
detected laptop confidence 0.972171
The phone is shaking
detected person confidence 0.968211
Cas de fraude détecté
```



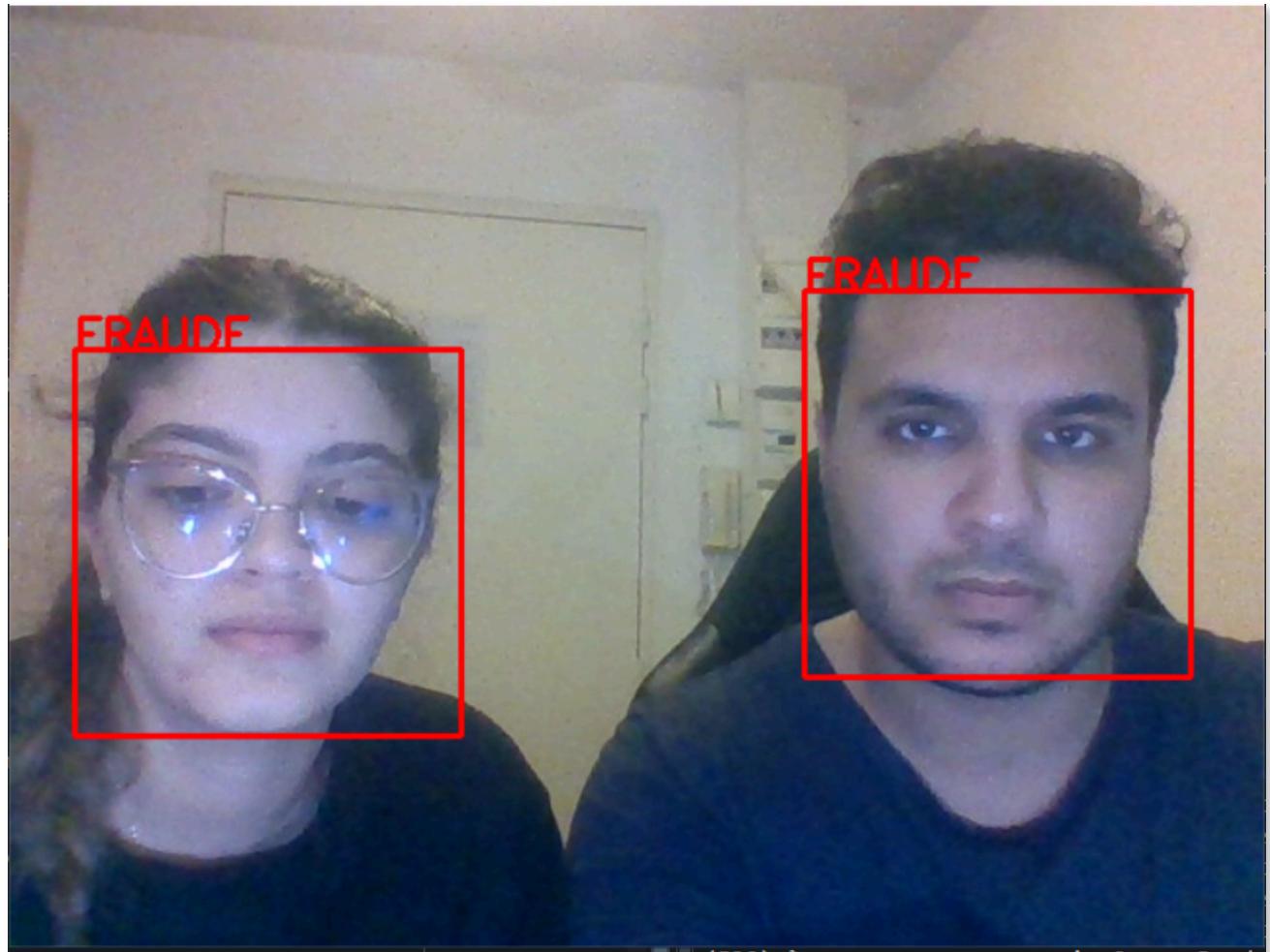
## RECONNAISSANCE FACIALE

La reconnaissance faciale est implémenté sur Python, le modèle doit être entraîné à l'avance sur des photos de candidat qui seront stockées dans une base de données. Pour cela chaque candidat doit préalablement déposer 3 photos de son visage.

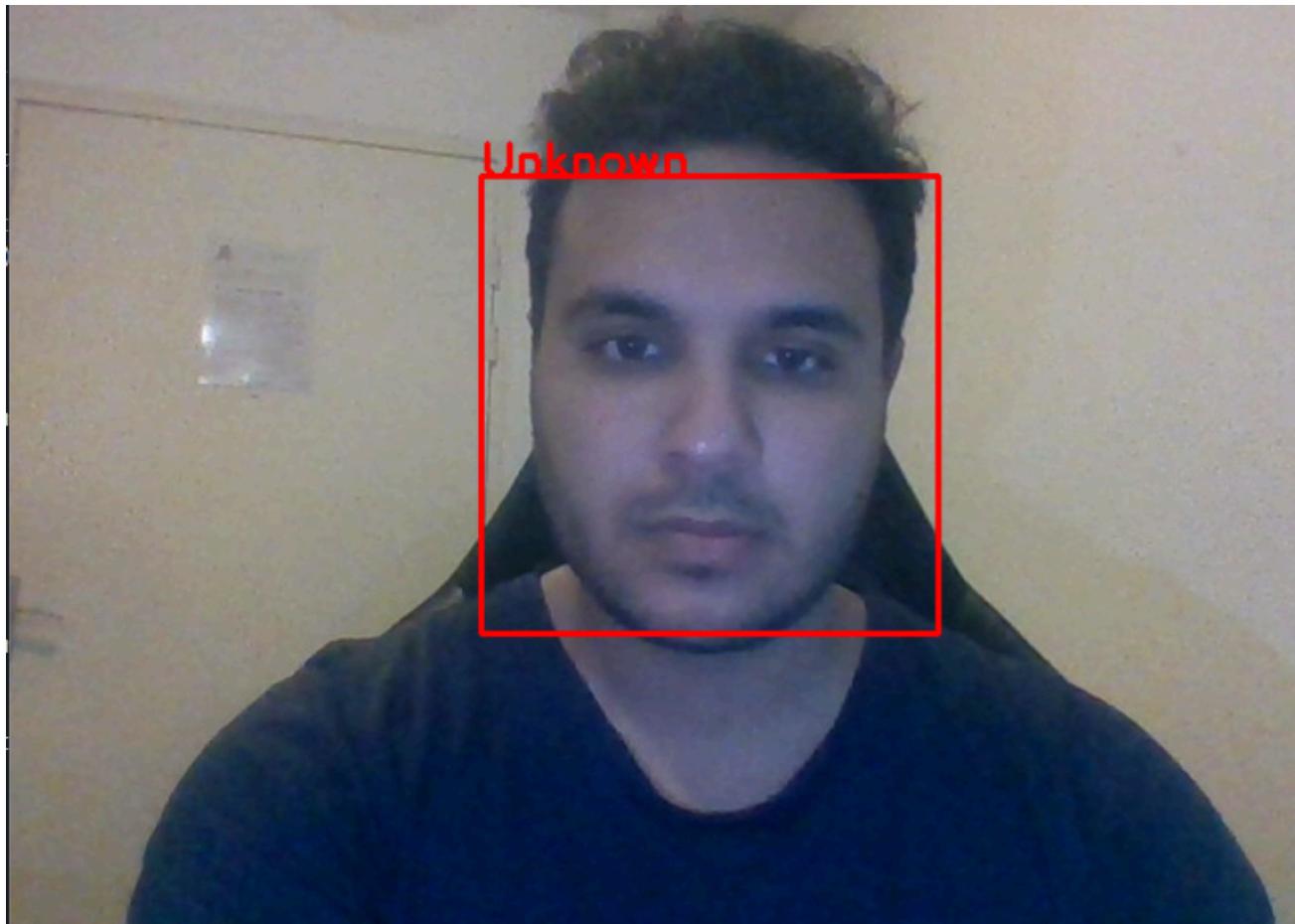
Reconnaissance du candidat au début de l'examen :



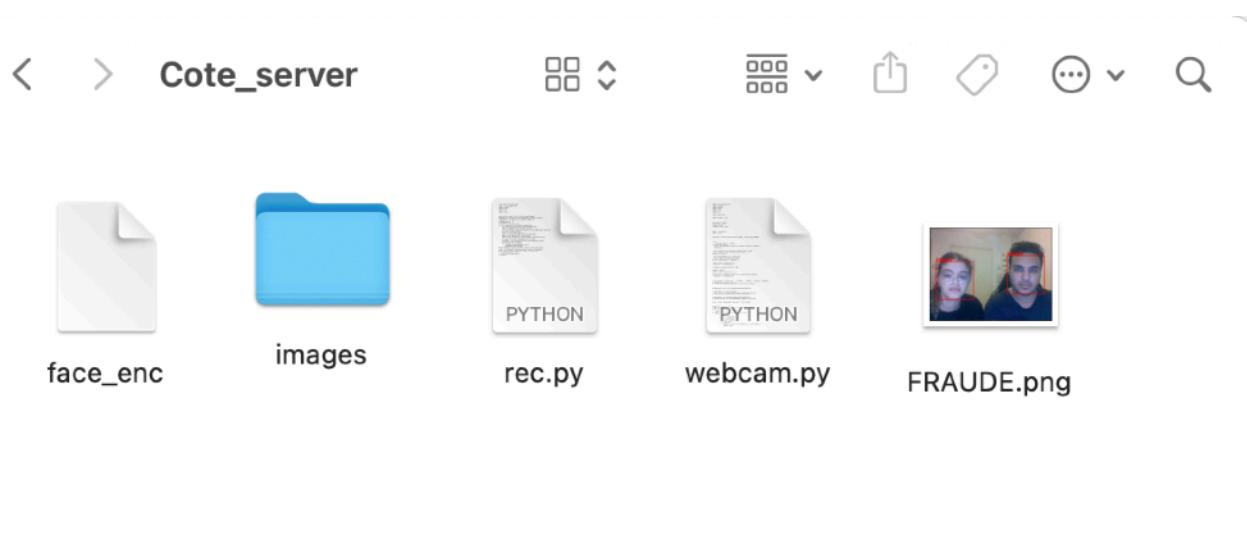
Voici un exemple d'une mise en situation où le système détecte plus qu'un personne :



Le système détecte une personne non reconnue.



En cas de fraude le système prends une capture d'écran avant d'arrêter l'examen.



## RECONNAISSANCE VOCALE

le système détecte un mot non autorisé pendant l'examen ( les mots non autorisés sont les mots qui sont prélevé des questions de l'examen).

**Text: question**  
**'question' is a word not allowed to say during the exam, CHEATING DETECTED**

### RESSOURCES :

<https://anuragajwani.medium.com/how-to-process-images-real-time-from-the-ios-camera-9c416c531749>

<https://www.youtube.com/watch?v=p6GA8ODInX0&t=214s>

<https://stackoverflow.com/questions/41292565/how-to-detect-if-iphone-is-in-motion>

<https://stackoverflow.com/questions/19131957/ios-motion-detection-motion-detection-sensitivity-levels>

<https://www.neuralception.com/object-detection-app/>

<https://www.youtube.com/watch?v=SkgHz8nw5V8>

<https://webpick.info/la-reconnaissance-faciale-avec-python/>

<https://www.titanwolf.org/Network/q/b9df5ce0-122b-4321-93c0-e80865c3ef8b/y>

<https://stackoverflow.com/questions/46011832/how-to-send-an-imageuiimage-from-a-swift-ios-client-program-to-a-java-server-p>

<https://www.youtube.com/watch?v=SkgHz8nw5V8>

[https://github.com/soumilshah1995/Transfer\\_Image\\_Python\\_SOCKET\\_UserInterface/blob/master/my\\_client.py](https://github.com/soumilshah1995/Transfer_Image_Python_SOCKET_UserInterface/blob/master/my_client.py)

<https://developer.apple.com/machine-learning/models/#mouvment>

<https://github.com/Ma-Dan/YOLOv3-CoreML>

<https://www.gfrigerio.com/object-recognition-with-vision-and-coreml/>

<https://www.analyticsvidhya.com/blog/2021/12/guide-for-speech-recognition/>