

10 DÉCEMBRE 2023

**PROJET
TECHNIQUE D'APPRENTISSAGE ARTIFICIEL**

ROBOT DE SAUVETAGE

SAve-IGO	2
Context	2
Mes objectifs	2
Les données d'apprentissage	3
Fabrication de mon modèle	3
La détection d'object	4
Les difficultés	4
Les sources	4

Nawfoel Ardjoune

MASTER 1 INFORMATIQUE

SAVE-IGO

CONTEXT

Je réalise ce projet dans le cadre du cours de Technique d'Apprentissage artificiel. Je projet est séparé en 2 partie, une pour la robotique que ne j'approfondirai pas dans ce rapport et celle qui nous intéresse pour l'apprentissages artificiel. A l'heure du rendu le projet n'ai pas encore complètement terminé car il reste quelque bug et la détection des voix à réaliser. De plus je souhaite codé l'ensemble du projet et donc limiter les dépendances et faire un réseaux de neurones par mes propre moyen.

MES OBJECTIFS

Mon projet consistait à programmer un Robot de détection de personne en détresse autonome, le SavelGO. Dans cet optique mon objectif consistait à faire en sorte que le robot puisse se déplacer, détecter des personnes, voir des membres ou reconnaître des appel à l'aide.

Mes objectifs étaient donc:

- Faire des recherche sur les technologies à utilisé
- récolter des données d'apprentissage et les traiter
- Utiliser les données ainsi traiter pour créer mon propre modèle
- Le modèle devra détecter un certain nombres de choses
- Tester mon modèle
- Récupérer un flux video et/ou audio les analyser pour y appliquer mes modèles
- Faire en sorte que le robot soit utilisable à distance ou se déplace tout seul.

LES DONNÉES D'APPRENTISSAGE

J'ai récolter des données d'apprentissage sur [Kaggle](#), plus de 11 000 images de chaussures et de mains. Cependant les données récolter était des données brut dont il manquais les fichiers CSV. Je les ai séparer et ai supprimer une partie pour ne pas avoir un trop grand déséquilibre pour mon modèle et ainsi éviter de le biaisé.

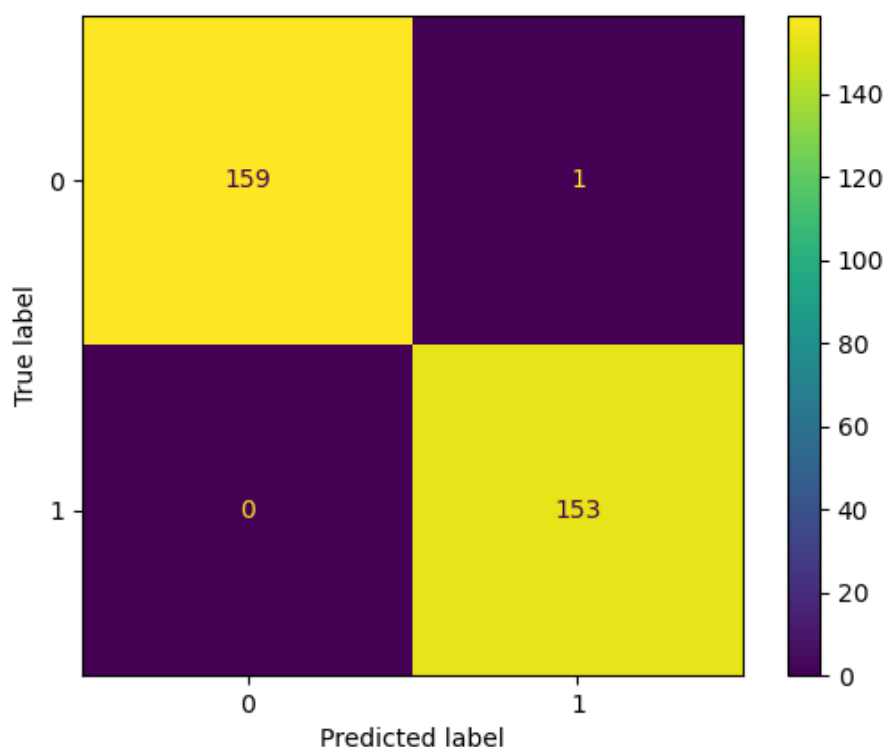
FABRICATION DE MON MODÈLE

J'ai utiliser pour la fabrication de mon modèle des librairie comme Sklearn, numpy ou joblib.

J'ai commencer importer mes images, les traiter pour qu'elles aient le même format et qu'elles soit en nuances de gris avant de les classer. J'ai ensuite assembler mes 2 ensembles d'images (mains et chaussure) afin d'utiliser des la fonction `train_test_split()` de la librairie sklearn j'ai utiliser 20% de la taille totale de mes données pour le test. Grâce à cela je récupère mes jeux de donnés afin d'entraîner et de tester mon modèle. J'ai fais le choix d'utiliser un modèle SVM qui collait parfaitement à mes données qui sont des images. J'ai donc normaliser mes données avant de remplir mon modèle.

Les résultat était très concluant :

```
➤(base) → projet python3 reco.py  
Précision du modèle : 99.92%  
➤(base) → projet █
```



LA DÉTECTION D'OBJECT

Pour réaliser une détection en temps réel j'ai pour les test utiliser la Webcam de mon ordinateur. Ainsi Que les librairie OpenCV et le modèle YOLO en partition avec mon modèle préalablement enregistrer. Je récupère l'image de mon flux video avec openCV et grâce à YOLO je suis maintenant capable de suivre et de classer les objets ou les gens. J'utilise donc les zone d'intérêt détecter par YOLO pour appliquer mon modèle et déduire is une cette zone correspond à une main ou à une chaussure par exemple.

LES DIFFICULTÉS

Il à été très difficiles de trouver des données celles utiliser ne me conviennent pas parfaitement. Avoir des images de pieds est difficile pour des raisons de vie privées pour ne citer que ça.

Il à aussi été difficile de traiter les images pour qu'elles conviennent au bon format de mes fonctions ou encore des modèle notamment avec le flux vidéo.

Le temps de traitement de mes images lors de la fabrication de mon modèle était assez long environ 1heures à chaque fois.

J'ai encore des bug avec l'image qui parfois affiche tout comme des chaussures ou tout comme des mains peu importe ce que je lui montre alors que le modèle à une très bonne précision en dehors du flux vidéo en temps réel

En temps réel le score de confiance des classe de mon modèle est négatif

LES SOURCES

Datasets des mains : <https://www.kaggle.com/datasets/shyambhu/hands-and-palm-images-dataset>

Datasets des chaussures : <https://www.kaggle.com/datasets/die9origephit/nike-adidas-and-converse-imaged>

OpenCV: <https://opencv.org/>

YOLO : <https://github.com/ultralytics/yolov3>