



École Polytechnique de l'Université de Tours  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 14  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

**Département Informatique**  
**5<sup>e</sup> année**  
**2020-2021**

**Cahier d'analyse**

# **Canne connectée pour aveugles**

**Encadrants**

Gilles VENTURINI  
[gilles.venturini@etu.univ-tours.fr](mailto:gilles.venturini@etu.univ-tours.fr)

Université François-Rabelais, Tours

**Auteurs**

Djawad M'DALLAH MARI  
[djawad.mdallah-mari@etu.univ-tours.fr](mailto:djawad.mdallah-mari@etu.univ-tours.fr)

DII5 2020-2021

Version du 4 janvier 2021

# Table des matières

---

<b>1</b>	<b>Cahier d'analyse</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Reconnaissance d'objet . . . . .	3
1.2.1	Modèles . . . . .	3
1.2.2	Intégration dans une application Android . . . . .	5
1.3	Informers l'utilisateur . . . . .	5
1.3.1	Synthèse vocale . . . . .	5
1.3.2	Vibration . . . . .	5
1.4	Navigation . . . . .	5
1.4.1	Guide d'utilisation . . . . .	5
1.4.2	Accès aux réglages . . . . .	5

# Cahier d'analyse

---

## 1.1 Introduction

Ce cahier d'analyse s'inscrit dans le cadre du projet Canne connectée pour aveugles. Il vise à présenter les analyses faites pour répondre aux besoins exprimés dans le cahier de spécifications. Une lecture au préalable du cahier de spécifications est donc recommandée afin de comprendre le contexte et les enjeux du projet.

Nous verrons donc dans ce document une analyse sur l'application Android à développer. Nous verrons en particulier quelques méthodes de reconnaissances d'objet pour le mobile, la méthode qui sera mise en place pour informer l'utilisateur et également comment garantir à l'utilisateur une interface adaptée à ses contraintes.

## 1.2 Reconnaissance d'objet

### 1.2.1 Modèles

L'un des besoins primaire pour la réalisation de ce projet est la reconnaissance d'objet. En effet, toute l'application dépend de la capacité du modèle à détecter et identifier un objet. Afin de répondre à ce besoin, il faudrait faire un inventaire des modèles de reconnaissance d'objet disponible puis faire des comparaisons. Pour mesurer les performances de chaque modèle, des critères doivent être établis ainsi que des conditions de fonctionnement bien défini. Cela permettrait d'avoir un environnement d'exécution commun pour chaque modèle, et donc des mesures cohérentes.

#### Modèles disponibles

Avec la librairie TensorFlow, nous disposons d'un grand panel de modèles pré-entraînés (disponibles sur <https://tfhub.dev>). Cependant, les modèles adaptés pour TensorFlow Lite restent peu, notamment ceux liés à la reconnaissance d'image et plus particulièrement la reconnaissance d'objets (**Object Detection**) (disponible sur <https://tfhub.dev/s?deployment-format=lite&module-type=image-object-detection>). En effet, il existe que 3 modèles officiels dans cette catégorie : SSD MobileNet, Mobile Object Localizer et East Text Detector. Parmi ces 3, on peut déjà abandonner le East Text Detector puisqu'il s'agit ici de détecter du texte. En revanche, au niveau des modèles de **Classification**, il existe un peu plus de modèles compatibles TensorFlow Lite (<https://tfhub.dev/s?deployment-format=lite&module-type=image-classification>).

## Object Detection vs Classification

Les modèles de reconnaissances d'objet (Objet Detection) sont capables de localiser et identifier plusieurs objets sur une même image.

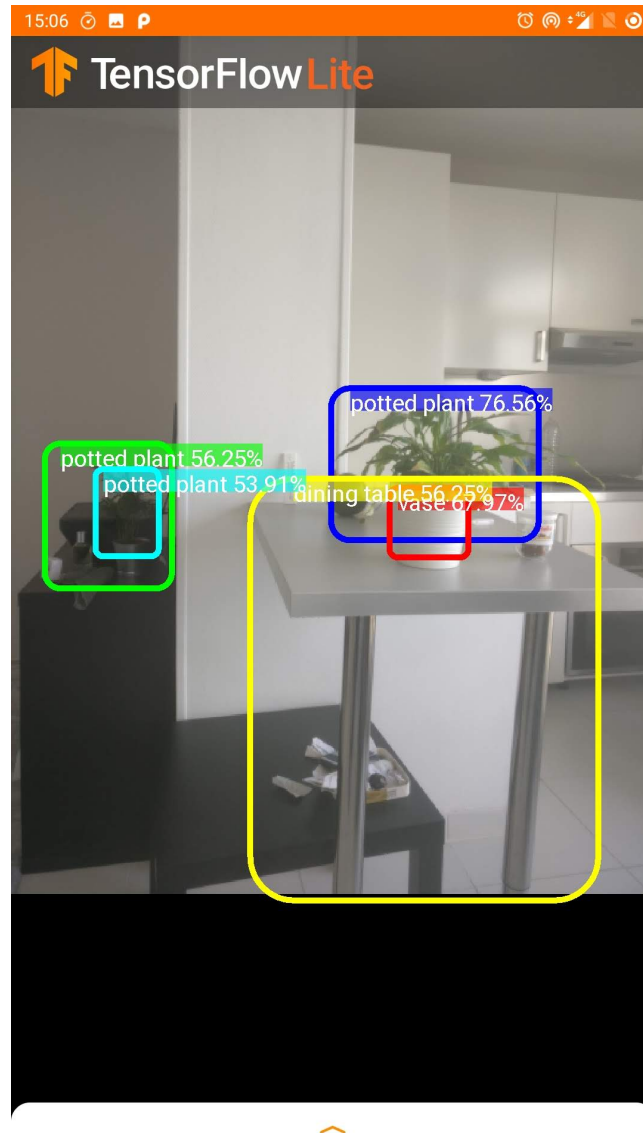


FIGURE 1.1 – Exemple de reconnaissance d'objet

Ce type de modèle est entraîné avec des objets de différentes classes (vêtements, fruits, etc.). Avec TensorFlow lorsqu'on met à l'entrée de ce type de modèle une image, on obtient en sortie une liste d'objets avec chacun sa localisation, sa classe et un degré de confiance (qui correspond à la fiabilité de l'objet identifié).

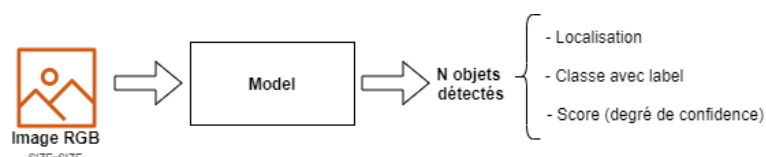


FIGURE 1.2 – Schema principe de fonctionnement d'un modèle

En revanche, les modèles de classification avec TensorFlow ne détecte qu'un seul élément sur une image. Ces modèles sont entraînés sur une seule classe générique (exemple : vêtements, aliments, plantes, etc) qui va ensuite être capable d'identifier l'élément de manière plus précise. Exemple :

- Classe vêtements : T-shirt, jean, ...
- Classe aliments : salade, pâtes, ...
- Classe fruits : pomme, banane, ...
- Classe insectes : sauterelle, abeille, papillon, ...
- ...

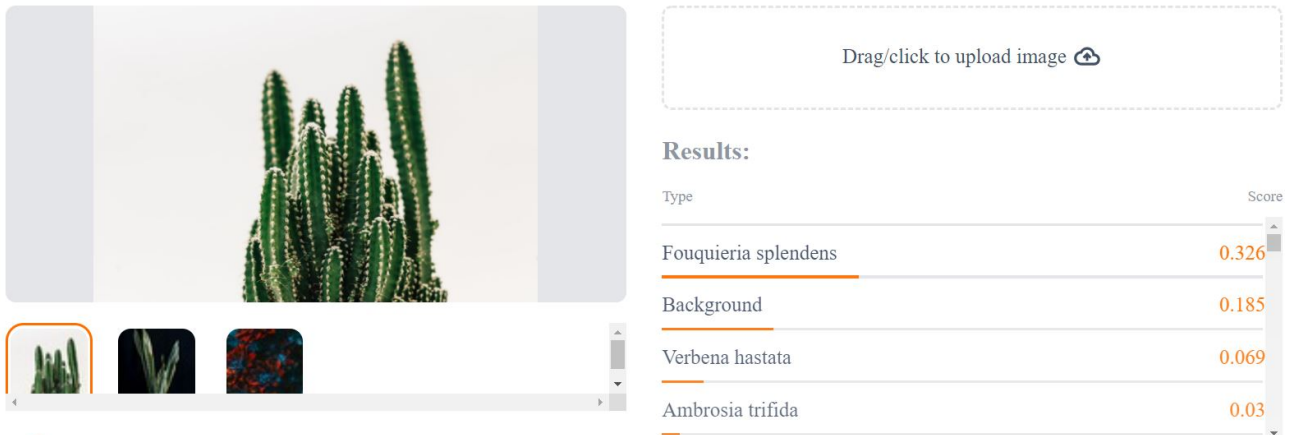


FIGURE 1.3 – Exemple de classification de plantes

## Choix du modèle

benchmark.. id 1 objet à la X(objet id) classification banque d'image

### 1.2.2 Intégration dans une application Android

les méthodes d'intégration fonctionnement entrée sortie

## 1.3 Informer l'utilisateur

### 1.3.1 Synthèse vocale

fonctionnement parametrage rendre le message compréhensible : viseur, vibreur encodage-trame (1mots,phrase,1vib,2vib,vib long..?) diag de classe

### 1.3.2 Vibration

## 1.4 Navigation

### 1.4.1 Guide d'utilisation

### 1.4.2 Accès aux réglages

# Table des figures

---

1.1	Exemple de reconnaissance d'objet . . . . .	4
1.2	Schema principe de fonctionnement d'un modèle . . . . .	4
1.3	Exemple de classification de plantes . . . . .	5

# Liste des tableaux

---

# Canne connectée pour aveugles

---

Département Informatique  
5<sup>e</sup> année  
2020-2021

Cahier d'analyse

**Résumé :** Cahier d'analyse canne connectée pour aveugles

**Mots clefs :**

**Abstract:**

---

**Keywords:** Encadrants  
Gilles VENTURINI  
[gilles.venturini@etu.univ-tours.fr](mailto:gilles.venturini@etu.univ-tours.fr)

Université François-Rabelais, Tours

**Auteurs**  
Djawad M'DALLAH MARI  
[djawad.mdallah-mari@etu.univ-tours.fr](mailto:djawad.mdallah-mari@etu.univ-tours.fr)

DII5 2020-2021