



---

IUT SORBONNE NORD

COMPTE RENDU – Détection d’Objets en Temps Réel

---

VAN NGOC Sean

Réalisation : Mars-Mai 2025

## Table des matières

Introduction.....	3
Objectif du Projet : .....	3
Fonctionnement général : .....	4
Technologies utilisées : .....	4
Expérience utilisateur : .....	4
Avantages : .....	4
Coût de conception : .....	5
Code source (extrait) : .....	5
Conclusion : .....	5
Annex : .....	7

## Introduction

Dans le cadre de mes activités au sein de l'association ESP LABORATORY, située au 99 Avenue Jean-Baptiste Clément à Villetaneuse (93430), j'ai mené un projet autour de la détection d'objets en temps réel par intelligence artificielle, reposant sur l'utilisation d'une webcam et d'un navigateur web, sans recours à des bibliothèques externes ni à une infrastructure lourde.

Cette initiative a été développée au sein du Fablab de l'IUT de Villetaneuse, également connu sous le nom d'IUTVLAB, un espace dédié à l'innovation technologique et aux projets d'expérimentation. Ouvert à l'ensemble de la communauté universitaire de l'USPN, ce laboratoire met à disposition des équipements numériques et des outils collaboratifs, contribuant au développement de projets dans un cadre semi-professionnel.

L'objectif du projet était de concevoir un outil capable de reconnaître automatiquement des objets (comme un téléphone, une bouteille ou une personne) à l'aide du modèle pré-entraîné COCO-SSD, exécuté directement dans le navigateur via TensorFlow.js. Le serveur local, développé en Python pur, permet à l'utilisateur d'accéder à l'interface de détection sans installation logicielle, garantissant une solution légère, rapide et fonctionnelle.

Ce travail m'a offert l'opportunité de mobiliser mes compétences en développement Python, en intégration de modèles de vision artificielle et en réalisation de systèmes interactifs, tout en expérimentant une solution IA accessible, économique et facilement transposable dans un contexte pédagogique ou associatif.

## Objectif du Projet :

Le but de ce projet est de mettre en place une application simple, locale et rapide d'exécution permettant de détecter des objets à l'aide d'une webcam.

Aucune installation de bibliothèque externe n'est nécessaire, ce qui le rend parfait pour des contextes associatifs ou pédagogiques.

## Fonctionnement général :

L'application utilise un fichier Python uniquement !

Ce script lance un petit serveur web en local, qui est accessible sur <http://localhost:8000>, qui affiche une page HTML intégrée. Une fois que la page est lancée, l'utilisateur doit donner l'accès à la caméra. Le flux vidéo est analysé en directe avec le modèle COCO-SSD, fourni par TensorFlow.js, qui détecte automatiquement les objets comme les bouteilles, téléphones, personnes, ordinateurs, etc.

## Technologies utilisées :

On utilise principalement les technologies suivantes :

- Python : uniquement les bibliothèques standard (aucune installation requise)
- HTML + Javascript : Principalement pour l'interface utilisateur dans le navigateur
- TensorFlow.js : Moteur IA embarqué dans le navigateur
- COCO-SSD : Modèle pré-entraîné de détection d'objets courants (90 objets)

**VOIR ANNEX**

## Expérience utilisateur :

L'utilisateur lance le fichier Python dans un terminal (Python, Visual Studio Code, Spyder etc.). Il accède ensuite à la page via le navigateur. L'image de la webcam sera affichée avec des rectangles rouges dessinés autour des objets détectés. Le nom de chaque objet est affiché avec un score de confiance.

## Avantages :

Ce projet présente plusieurs avantages :

- Il fonctionne sans aucune installation compliquée.
- Il utilise des technologies modernes accessibles depuis n'importe quel navigateur.
- Il est totalement autonome et fonctionne en mode local, sans connexion internet.
- Il est facile à modifier ou à enrichir avec d'autres fonctionnalités.

## Coût de conception :

Le projet a été réalisé avec des outils 100% gratuits et open source :

Ressource	Coût
Python (langage et serveur HTTP)	0€
Navigateur	0€
TensorFlow.js + COCO-SSD	0€
Matériel → PC + Webcam	Déjà disponible dans l'association
<b>COÛT TOTAL DU PROJET</b>	<b>0€</b>

## Code source (extrait) :

Voici un aperçu du code utilisé :

Nous avons caché le code HTML car il prenait toute la place.

```
1  import http.server
2  import socketserver
3
4  PORT = 8000
5
6  > html_content = """ ...
68
69  class MyHandler(http.server.SimpleHTTPRequestHandler):
70      def do_GET(self):
71          if self.path in ['/', '/index.html']:
72              self.send_response(200)
73              self.send_header("Content-type", "text/html; charset=utf-8")
74              self.end_headers()
75              self.wfile.write(html_content.encode("utf-8"))
76          else:
77              super().do_GET()
78
79  with socketserver.TCPServer(("", PORT), MyHandler) as httpd:
80      print(f"🚀 Serveur actif sur http://localhost:{PORT}")
81      httpd.serve_forever()
```

## Conclusion :

Ce projet démontre la faisabilité d'un système de vision par ordinateur léger, totalement embarqué, en utilisant uniquement :

- Python standard pour le serveur http local
- JavaScript pour l'interface et la logique côté client

- TensorFlow.js pour l'inférence de modèles d'intelligence artificielle directement dans le navigateur sans dépendances ni installation lourde.
- 

L'intégration du modèle pré-entraîné COCO-SSD permet d'exploiter un réseau de neurones convolutifs moderne (SSD – Single Shot Multibox Detector), capable de détecter et classifier jusqu'à 90 objets différents en temps réel, avec une latence faible et une précision acceptable (>50 % de confiance).

Ce projet met donc en œuvre :

- Une chaîne complète client-serveur locale (Python + HTML/JavaScript)
- Inférence embarquée côté navigateur, compatible avec les contraintes d'environnement d'apprentissage (pas d'installation, pas de GPU requis)
- 

Il constitue une base pédagogique idéale pour illustrer :

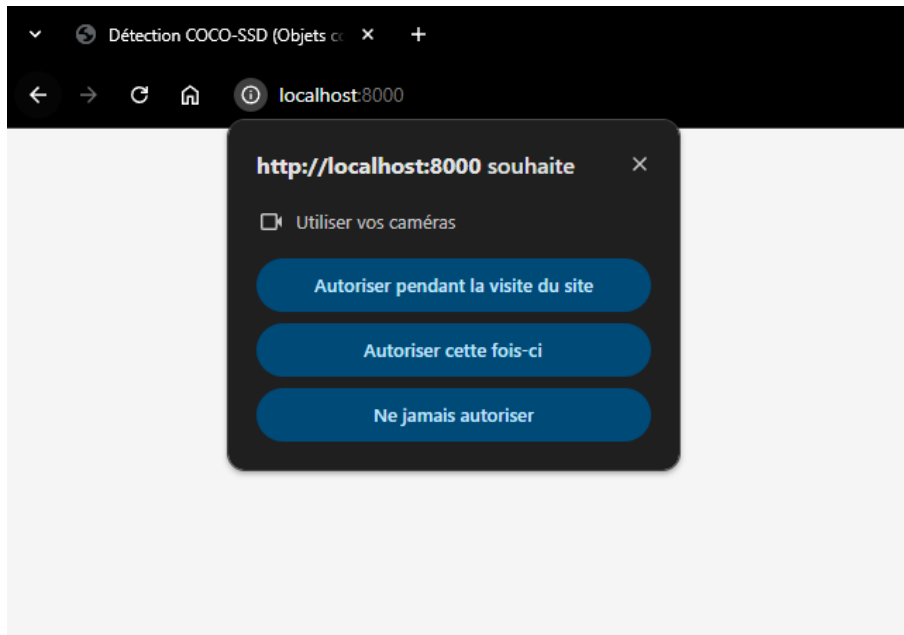
- La structure d'un pipeline IA léger
- Fonctionnement d'un modèle de détection d'objets
- Interaction entre serveur local Python et navigateur
- 

Ce projet est donc un excellent point de départ pour initier les membres de l'association aux notions de :

- Traitement d'image en temps réel
- Réseaux de neurones conventionnels\*
- Inférence IA dans le navigateur
- Développement de prototypes IA accessibles

Réseaux de neurones conventionnels : est un modèle d'intelligence artificielle conçu pour analyser des images. Il fonctionne en balayant l'image par petites zones pour détecter des formes, motifs et objets, ce qui le rend très efficace pour la reconnaissance visuelle (ex : visages, téléphones, bouteilles, etc.)

## Annex :



Objets disponibles sur le modèle COCO-SSD : <https://github.com/amikelive/coco-labels/blob/master/coco-labels-paper.txt>

...	1	person	47	cup
	2	bicycle	48	fork
	3	car	49	knife
	4	motorcycle	50	spoon
	5	airplane	51	bowl
	6	bus	52	banana
	7	train	53	apple
	8	truck	54	sandwich
	9	boat	55	orange
	10	traffic light	56	broccoli
	11	fire hydrant	57	carrot
	12	street sign	58	hot dog
	13	stop sign	59	pizza
	14	parking meter	60	donut
	15	bench	61	cake
	16	bird	62	chair
	17	cat	63	couch
	18	dog	64	potted plant
	19	horse	65	bed
	20	sheep	66	mirror
	21	cow	67	dining table
	22	elephant	68	window
	23	bear	69	desk
	24	zebra	70	toilet
	25	giraffe	71	door
	26	hat	72	tv
	27	backpack	73	laptop
	28	umbrella	74	mouse
	29	shoe	75	remote
	30	eye glasses	76	keyboard
	31	handbag	77	cell phone
	32	tie	78	microwave
	33	suitcase	79	oven
	34	frisbee	80	toaster
	35	skis	81	sink
	36	snowboard	82	refrigerator
	37	sports ball	83	blender
	38	kite	84	book
	39	baseball bat	85	clock
	40	baseball glove	86	vase
	41	skateboard	87	scissors
	42	surfboard	88	teddy bear
	43	tennis racket	89	hair drier
	44	bottle	90	toothbrush
	45	plate		
	46	wine glass		