

# 2024 l'Odysée de l'IA

premiers essais avec TensorFlow

Brian Tristam Williams (Elektor)

En explorant le potentiel de TensorFlow Lite sur le Raspberry Pi, cet épisode se penche sur les aspects pratiques de l'IA dans un environnement compact, en testant les limites de ce qui est possible sur des appareils de faible puissance.

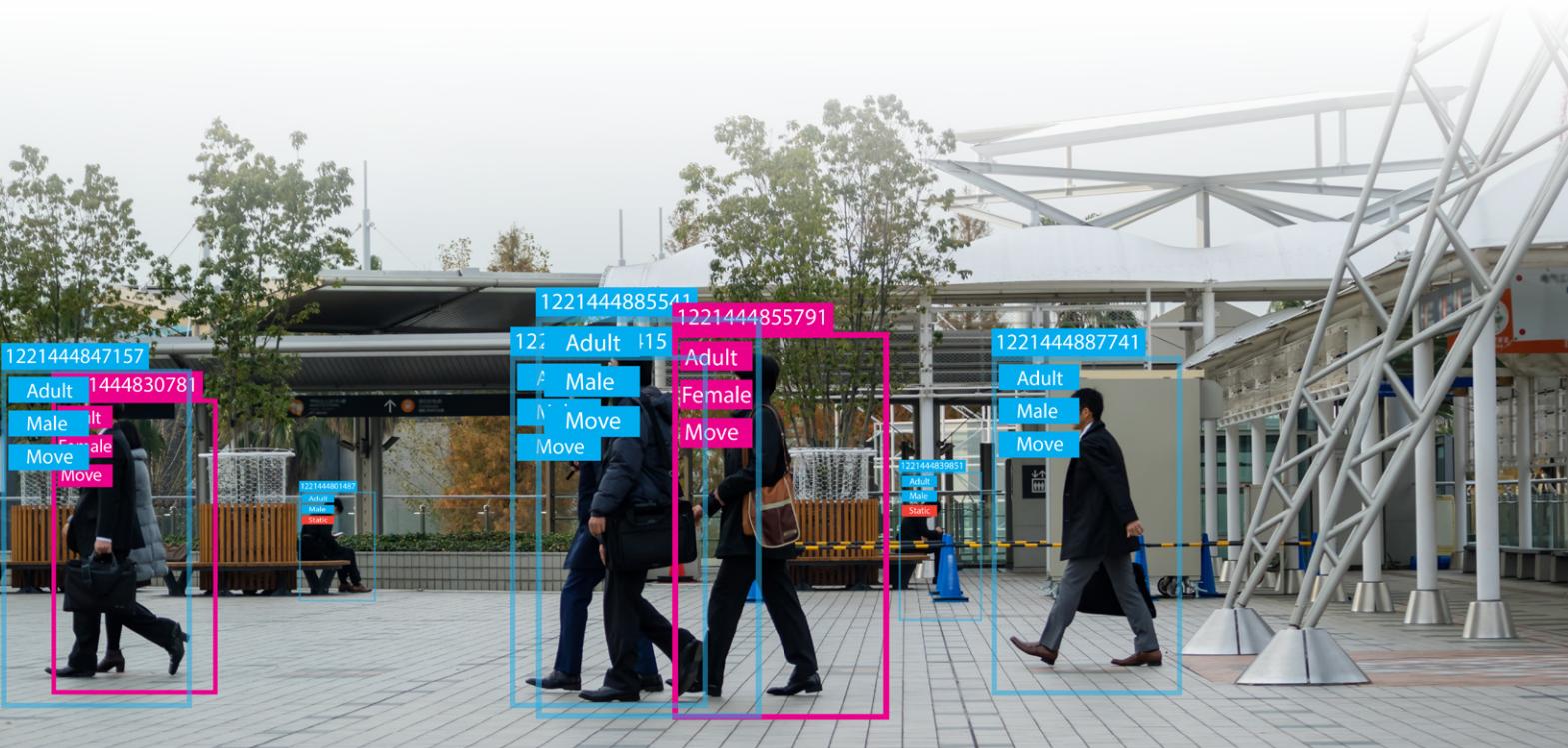


**TensorFlow Lite**

## TensorFlow : quèsaco ?

Développé par l'équipe Google Brain, TensorFlow est une bibliothèque open-source pour le calcul numérique et l'apprentissage automatique (en anglais : machine learning), jouant un rôle essentiel dans l'écosystème plus large de l'IA, qui inclut des techniques d'IA générative telles que ChatGPT. Il s'agit d'une pierre angulaire de l'apprentissage profond, qui permet de créer des modèles capables de traiter et d'apprendre des « données massives » (big data).

TensorFlow et les modèles d'IA générative sont souvent complémentaires, TensorFlow fournissant la plateforme fondamentale pour créer et entraîner une variété de modèles IA, y compris ceux qui pourraient alimenter les applications génératives. Sa polyvalence lui permet de s'adapter à un large éventail de tâches, allant de simples classifications à des processus décisionnels complexes, ce qui en fait un choix populaire dans les domaines académiques et industriels. Sa capacité à traiter de grands ensembles de données, à reconnaître des modèles et à en apprendre le rend un outil puissant dans le cadre plus général de l'IA générative, qui se concentre sur la création de nouveaux modèles de contenu et de données.



## TensorFlow Lite : moins c'est plus

Dans ce nouvel épisode de notre voyage dans l'IA, j'emprunte un chemin quelque peu inhabituel. Alors que j'ai accès à un PC assez performant équipé d'un GPU RTX 4070 approprié et de suffisamment de RAM, je me trouve attiré par le monde de TensorFlow Lite [1] sur le Raspberry Pi. Il y a un certain charme dans le défi d'optimiser les processus d'IA pour un appareil aussi compact et moins puissant. Mais il ne s'agit pas seulement de faire plus avec moins ; il s'agit de créer des solutions intelligentes, autonomes, portables et efficaces. Parfois, le projet à réaliser ne nécessite pas, ou n'est pas en mesure de supporter, l'encombrement et la puissance d'une tour de bureau. Dans de tels cas, le Raspberry Pi constitue une alternative parfaite. L'exécution de TensorFlow Lite sur cette petite mais performante carte permet de rendre l'IA plus accessible et adaptable à divers environnements, que ce soit à des fins éducatives, pour des projets amateurs ou pour des applications pratiques où l'encombrement et la consommation d'énergie sont primordiaux.

Avec l'essor de l'*edge computing* (informatique en périphérie), le besoin d'outils d'IA à la fois puissants et efficaces s'est accentué. C'est là que TensorFlow Lite, une version plus légère et plus efficace de TensorFlow, entre en jeu, notamment pour des appareils tels que le Raspberry Pi. Le Raspberry Pi étant un nano-ordinateur abordable, mais puissant, fournit la plateforme parfaite pour exécuter TensorFlow Lite, apportant des capacités d'apprentissage automatique au bout des doigts des amateurs, des éducateurs et des professionnels.

L'intégration de TensorFlow Lite dans un petit ordinateur monocarte offre des avantages significatifs pour les applications d'IA sur des systèmes plus grands. TensorFlow Lite est optimisé pour fonctionner sur du matériel léger, ce qui le rend adapté à un large éventail d'applications pratiques autres que la détection d'objets. Il s'agit notamment du traitement de données en temps réel, de la prise de décision locale dans les appareils IoT et de l'exécution de modèles IA pour des tâches telles que la reconnaissance des gestes, les capteurs environnementaux et la surveillance de la santé. Les fabricants de Raspberry Pi peuvent ainsi déployer des modèles d'IA complexes de manière rentable et accessible, ce qui ouvre un monde de possibilités pour l'innovation et l'exploration de l'IA, même dans le cadre de petits projets.

Dans les parties suivantes, nous allons voir les étapes de l'installation de TensorFlow Lite sur un Raspberry Pi et nous préparer aux futures applications innovantes qu'il permet de réaliser.

## Pour commencer

J'ai un tiroir plein de Raspberry Pi, allant du O.G. en passant par le Zero jusqu'au 5, mais je vais utiliser un Raspberry Pi 4 pour ce projet. C'est un bon choix en termes d'accessibilité pour nos lecteurs, parmi les trois, que je vois encore en vente de temps en temps. (D'ailleurs, je n'ai qu'un seul Raspberry Pi 5).

Donc, voici ce que j'ai pour commencer :

- **Raspberry Pi 4 Model B :** Nous allons voir comment il fonctionne avec ses 4 Go de RAM. (J'ai utilisé la commande `free -h` pour me rappeler la quantité de mémoire embarquée).
- **Carte microSD de 32 Go :** Je ne sais pas si j'ai nécessairement besoin d'autant d'espace, mais c'est la première que j'ai trouvée quand j'en ai cherché une.

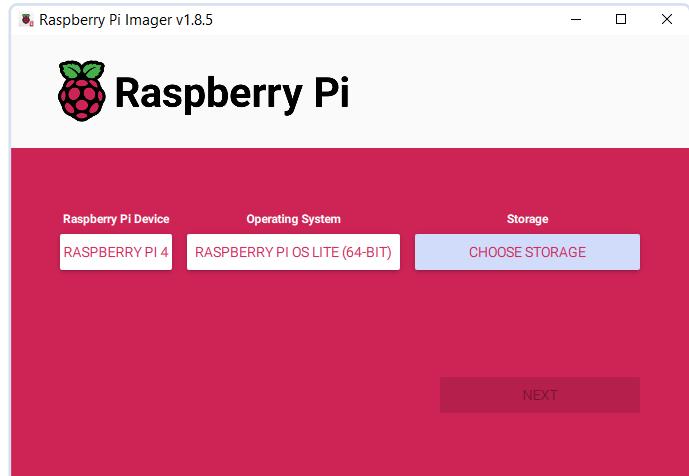


Figure 1. Raspberry Pi Imager dispose d'une interface simple et conviviale.

➤ **Raspberry Pi OS Lite (64-bit) :** Je ne suis pas vraiment fan de la complexité introduite par les interfaces graphiques, et je préfère le oui/non ou on/off rassurant que j'obtiens de l'interface texte, il s'agit d'un portage de Debian Bookworm sans environnement de bureau. Ayant longtemps travaillé avec des serveurs web, je n'ai jamais vraiment trouvé d'utilité à une interface graphique sous Linux. Faites-moi savoir si je devrais me mettre à la page !

J'ai installé la version 1.8.5 de Raspberry Pi Imager, la version disponible au moment de la rédaction de cet article depuis [2]. Il suffit de télécharger la version correspondant à votre système d'exploitation et de l'exécuter. C'est vraiment génial pour simplifier le processus d'installation. Il suffit de choisir la version de votre carte Raspberry Pi, le système d'exploitation que vous souhaitez, et la carte que vous avez connectée au lecteur de carte de votre ordinateur. L'interface utilisateur est vraiment simple (**figure 1**). Mieux encore, cela fonctionne. J'ai vérifié mon numéro de version une fois installé, en utilisant `cat /etc/os-release`. Le vôtre peut être différent, mais voici le résultat que j'ai obtenu :

```
briantw@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

En ce qui concerne les accessoires, j'ai simplement un clavier (pas besoin de souris - youpi !), une alimentation Raspberry Pi 4, et les deux sorties Micro HDMI connectées (**figure 2**). Actuellement, l'une est reliée à un petit moniteur portable et l'autre à un périphérique de capture sur mon PC, que, pour l'instant, j'utilise pour quelques captures d'écran. L'idée à terme est d'avoir un code qui s'exécute sur un écran et une sortie vidéo (par exemple d'un script de détection d'objet) visible sur l'autre.

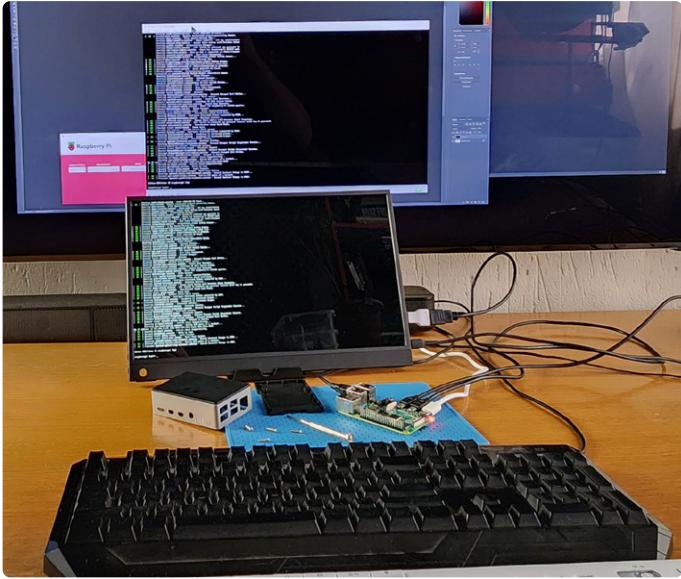


Figure 2. La configuration de base du Raspberry Pi que j'utilise pour ce projet.

## Installation de TensorFlow Lite

Maintenant que nous avons tout ce qu'il faut pour commencer, exécutons TensorFlow Lite sur le Raspberry Pi.

Tout d'abord, quelques réglages. Lançons un cycle de mise à jour/mise à niveau sur le Raspberry Pi, juste pour nous assurer que nous avons toutes les dernières versions de tout ce qui est installé. Commençons par :

```
sudo apt-get update
```

Le temps d'attente dépend de votre retard côté mises à jour. Je n'ai eu que 12 éléments à mettre à jour. Ensuite :

```
sudo apt-get upgrade
```

Pour moi, cela a résulté en un écran entier rempli de mises à jour, ce qui me surprend toujours un peu puisque je viens juste d'installer la dernière version officielle du système d'exploitation. Soyez patient avec le processus - la durée peut varier.

En ce qui concerne l'installation de TensorFlow Lite, j'ai plongé dans divers résultats de recherche Google, et le guide le plus accessible que j'ai trouvé était celui d'EdjeElectronics sur GitHub [3]. Bien entendu, un certain temps s'est écoulé depuis la présentation initiale du processus par Edge et les nouvelles versions et mises à jour du système d'exploitation, et j'ai donc rencontré quelques difficultés mineures, que je vais documenter ici.

La première étape a été de cloner le dépôt GitHub en utilisant :

```
git clone https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git
```

La réponse encourageante de mon Raspberry Pi :

```
-bash: git: command not found
```

Ah oui, c'est une nouvelle installation. Oups. Installez `git` en utilisant :

```
sudo apt install git
```

Cela a fonctionné pour moi. Maintenant que `git` est installé, revenons à la commande `git clone...` mentionnée plus haut. Si vous réussissez, vous obtiendrez un sous-répertoire dans votre répertoire personnel appelé *TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi*. C'est un peu compliqué, alors renommez-le en *tflite1*, avec la commande `mv` (move) :

```
mv TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi tflite1
```

Ensuite, accédez à ce répertoire avec :

```
cd tflite1
```

EdjeElectronics suggère de créer un environnement virtuel, nous allons donc le suivre :

```
sudo pip3 install virtualenv
```

La réponse du Raspberry Pi, comme on pouvait s'y attendre :

```
-bash: pip3: command not found
```

L'installation de `pip3` s'est faite presque aussi facilement que celle de `git`. J'ai d'abord utilisé `sudo apt-get update`, où neuf mises à jour ont été effectuées, puis j'ai utilisé la commande :

```
sudo apt-get install python3-pip
```

De nombreuses installations ont eu lieu à ce stade, ce qui a abouti à un autre écran rempli de texte.

Ensuite, j'ai réessayé de lancer la commande `sudo pip3...` Debian n'a pas aimé. Après quelques recherches, j'ai découvert que je devais créer mon propre environnement virtuel. En supposant que vous êtes toujours dans le répertoire *tflite1*, utilisez :

```
python3 -m venv tflite1-env
```

Activez ensuite cet environnement virtuel :

```
source tflite1-env/bin/activate
```

Il faut maintenant installer les dépendances de TensorFlow Lite et OpenCV. Comme nous l'avons vu précédemment, TensorFlow Lite a de nombreux domaines d'application, mais cette installation va l'adapter à un domaine très populaire : la vision artificielle et la reconnaissance d'objets. Ainsi, OpenCV, une bibliothèque open-source destinée aux tâches de vision artificielle et d'apprentissage automatique, sera bien pratique.

Heureusement, vous n'aurez pas besoin de `curl`, `wget`, ou `git clone` ici, parce que le créateur du dépôt a écrit un script shell utile de 40 lignes,

appelé `get_pi_requirements.sh`. Si vous êtes curieux de découvrir ce que le script est censé faire avant de l'exécuter, vous pouvez voir son contenu en exécutant un `cat get_pi_requirements.sh`, mais nous allons l'exécuter en entrant :

```
bash get_pi_requirements.sh
```

J'ai fait l'erreur d'essayer ceci après avoir redémarré le Raspberry Pi, ce qui a généré des messages d'erreur et des avertissements. Pourquoi ? La commande source que nous avons exécutée plus tôt dans le répertoire `tflite1` doit être exécutée chaque fois que vous redémarrez ou que vous démarrez une nouvelle session du Terminal. De plus, cela a pris quelques minutes pour s'exécuter, alors soyez patient, encore une fois ! Maintenant, nous avons la possibilité d'utiliser un modèle TensorFlow Lite de Google, ou d'utiliser un modèle que nous avons développé nous-mêmes. En tant que débutant, j'ai choisi d'utiliser celui de Google :

```
wget https://storage.googleapis.com/download.tensorflow.org/models/tflite/coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip
```

Cette commande est longue et ne tient pas ici sans saut de ligne, mais notez que le seul espace que vous devez saisir est celui qui suit `wget` - il n'y en a pas à partir de `https`.

Ensuite, décompressez le fichier que vous venez de télécharger :

```
unzip coco_ssd_mobilenet_v1_1.0_quant_2018_06_29.zip -d Sample_TFLite_model
```

Et voilà, nous avons installé tout ce dont nous avons besoin pour commencer à nous amuser avec la détection d'objets et la classification d'images en utilisant TensorFlow Lite sur le Raspberry Pi.

Nous pouvons maintenant commencer à faire des essais.

Dans notre prochain épisode : en tant que collectionneur de données, en particulier d'images et de vidéos historiques qui ne sont souvent disponibles nulle part ailleurs, je souhaite partager ce que j'ai à ma disposition avec le monde entier. Mais je n'aurai jamais le temps de passer en revue chaque photo et chaque image vidéo pour ajouter des métadonnées et indiquer aux moteurs de recherche ce que j'ai. J'ai donc l'intention d'utiliser ces outils pour m'aider à analyser et à classer ma grande collection de fichiers multimédias. Je vous tiendrai au courant de mes succès et de mes échecs la prochaine fois ! ↗

230181-E-04

## Questions ou commentaires ?

Envoyez un courriel à l'auteur (brian.williams@elektor.com).



## À propos de l'auteur

Brian Tristam Williams est fasciné par les ordinateurs et l'électronique depuis qu'il a eu son premier « micro-ordinateur » à l'âge de 10 ans. Son aventure avec le magazine Elektor a commencé lorsqu'il a acheté son premier numéro à 16 ans, et depuis lors, il suit le monde de l'électronique et de l'informatique, ne cessant d'explorer et d'apprendre. Il a rejoint Elektor en 2010 et, aujourd'hui, il s'attache à suivre les dernières tendances technologiques, en se concentrant notamment sur l'intelligence artificielle et les ordinateurs monocartes tels que le Raspberry Pi.



## Produits

› **Raspberry Pi 4B (2 GB RAM)**  
[www.elektor.fr/18965](http://www.elektor.fr/18965)



## LIENS

- [1] Site officiel de TensorFlow Lite : <https://tensorflow.org/lite>
- [2] Téléchargement de Raspberry Pi Imager : <https://raspberrypi.com/software>
- [3] TensorFlow Lite Object Detection on Android and Raspberry Pi [GitHub] : <https://tinyurl.com/edjetflite>