



# Connexionisme

Deeplab

**FELIX Jérémy**  
**MOUKIN Kévin**  
**REGALADE Davy**  
**2018 - 2019**

## **Sommaire**

I.	Objectifs et points du projet.....	3
II.	Explications.....	3
a.	TensorFlow.....	3
b.	DeepLab .....	4
c.	DCNN et Markov .....	4
III.	Installation de l'application.....	4
IV.	Modifications apportés dans le code.....	5
V.	Résumé/Conclusion .....	5
VI.	Webographie.....	6

## I. Objectifs et points du projet

Le but de ce projet était de comprendre le deep learning, à travers un sujet choisis ou imposé. Dans notre cas, cela s'est fait sur le DeepLab. Le DeepLab est une technique de segmentation d'images avec une reconnaissance d'objets dans un plan grâce à son apprentissage préalable par tensorflow.

Pour ce projet, nous disposons d'un jeu de données fournies par DeepLab, disponibles sur leur dépôt Github.

Le jeu de données peut avoir deux axes très différents dans l'apprentissage :

- Entraîner le modèle à partir de zéro : liberté d'avoir un nombre illimité de classes d'objets (nombre d'étiquettes) pour la segmentation. Cela nécessite un temps de formation très long.
- Le modèle préformé : liberté d'avoir un nombre quelconque de classes d'objets pour la segmentation. Mettre à jour que les poids du classificateur avec l'apprentissage par transfert. Cela prendra beaucoup moins de temps pour l'entraînement que pour le scénario précédent.

## II. Explications

### a. TensorFlow

TensorFlow est un Framework Open Source de programmation pour le calcul numérique développé Google.

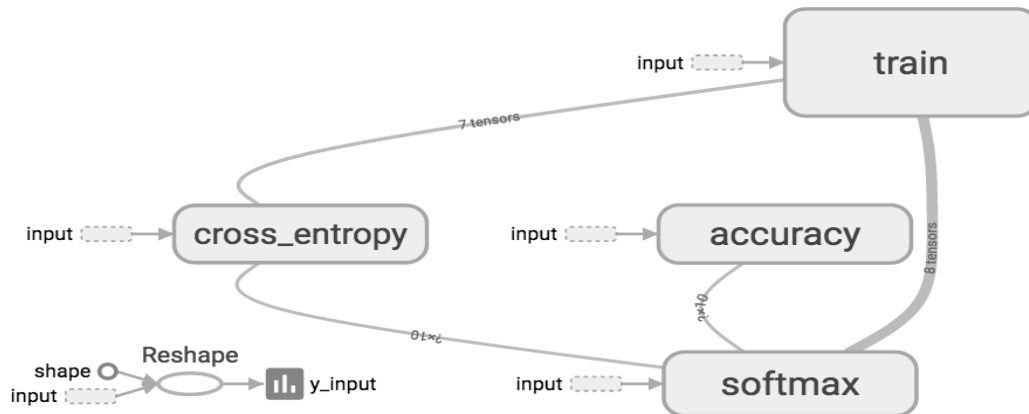
TensorFlow représente les calculs sous la forme d'un graphe d'exécution: les différents nœuds représentent les « *Operations* » à réaliser, et les liens représentent les « *Tensors* ». Une « *Operation* » peut aller d'une simple addition à une fonction complexe de différenciation matricielle.

Chaque « *Operation* » prend en entrée zéro, un ou plusieurs « *Tensor* », effectue un calcul, et retourne zéro, un ou plusieurs « *Tensor* ». Un exemple typique de « *Tensor* » est un batch d'images. Un batch d'images est représenté par un « *Tensor* » à 4 dimensions: taille du batch (nombre d'images dans le batch), hauteur, largeur et nombre de canaux de représentation (3 pour une image en couleurs représentée en RGB).

La création du graphe est automatiquement gérée par TensorFlow une fois les « *Tensors* » et « *Operation* » implémentés et instanciés. Cela permet une optimisation et parallélisation du code et de l'exécution lors du lancement.

TensorFlow possède de plus un support très vaste pour la création d'opérations spécifiques au Deep Learning, et il devient donc facile de construire un réseau de neurones et d'utiliser les opérations mathématiques couramment associées pour l'entraîner avec les bons optimiseurs.

Voici, un exemple de graphe d'exécution :



Graphe d'exécution 1

## b. DeepLab

DeepLab est l'une des techniques les plus prometteuses pour **segmentation d'image sémantique** avec Deep Learning. La segmentation sémantique consiste à comprendre une image au niveau du pixel, puis à attribuer une étiquette à chaque pixel d'une image, de sorte que les pixels portant la même étiquette partagent certaines caractéristiques.

Qu'est-ce que la segmentation sémantique d'image ?

La **segmentation sémantique d'image** est une opération de traitement d'image qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond.

## c. DCNN et Markov

# III. Installation de l'application

Pour ce projet, nous nous sommes basé sur une application déjà existante, utilisant le Tensorflow et provenant d'un projet existant de reconnaissances d'objet dans une image/vidéo, DeepLab sur Android. (<https://github.com/pillarpond/image-segmenter-android>)

Afin de l'utiliser, il va falloir installer quelques pré-requis :

1. Installer Android Studio.
2. Installer Git sur votre machine
3. Configurer git dans Android studio
  - Sur Android Studio, aller dans File -> Settings -> Git
  - Mettre le chemin de git.exe existant dans votre Git au préalable installé,
  - Tester pour vérifier le path et Valider
  - Ensuite, cliquer sur VCS -> Check out from.... -> Git
  - Mettre l'url suivante : <https://github.com/lebenef/image-segmenter-android>
4. Git clone le projet.
5. Activer le mode développeur et le débogage de son téléphone.
6. Run l'application sur son téléphone.

## **IV. Modifications apportés dans le code**

Pour l'instant, nous n'avons pas effectué de modifications majeures sur l'application. Nous pouvons juste afficher les labels (noms) de chaque classe reconnue.

Par ailleurs, nous avons réfléchi à des améliorations que l'on pourrait apporter à cette application, dans un premier temps, augmenter la reconnaissance puis par exemple l'apprentissage de nouvelles classes d'objets ou encore un historique des objets reconnues et leur occurrence durant un laps de temps. L'ajout d'un mode vidéo ou photo pour la reconnaissance, c'est-à-dire, qu'on lui passe une vidéo et l'application arrive à détecter les objets de la vidéo avec leur temps d'apparitions ? De même pour une image qu'on lui passe en entrée.

Si l'on veut pousser le concept un peu plus loin et partir dans une idée assez folle. On pourrait s'inspirer de « l'œil de dieu » de Fast and Furious 7, et faire en sorte que le deeplab détecte des comportements dangereux ou d'objets grâce aux caméras. Intégrer le sujet de thèse de M.Nagau pour la détection de certaines plantes ?

## **V. Résumé/Conclusion**

## VI. Webographie

### DeepLab

- <https://ai.googleblog.com/2018/03/semantic-image-segmentation-with.html>
- <http://liangchiehchen.com/projects/DeepLab.html>
- <https://arxiv.org/pdf/1511.03339.pdf>
- <http://www.stat.ucla.edu/~xianjie.chen/paper/Chen14cvpr.pdf>
- <https://medium.freecodecamp.org/how-to-use-deeplab-in-tensorflow-for-object-segmentation-using-deep-learning-a5777290ab6b>

### CNN

- [https://fr.wikipedia.org/wiki/R%C3%A9seau\\_neuronal\\_convolutif#cite\\_note-mcdns-6](https://fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif#cite_note-mcdns-6)
- <https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-b288519dbcf8>
- <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>
- [https://www.researchgate.net/figure/A-general-DCNN-architecture\\_fig1\\_318332170](https://www.researchgate.net/figure/A-general-DCNN-architecture_fig1_318332170)