

# Synthèse Etape 1

---

## Définition des objectifs

- Définition des objectifs
- Synthèse du tuto
  - Algorithmes
  - Formats des Images
- Environnements
  - Gestion de projet
  - Développement
- Critique du code existant
- Sources

# Définition des objectifs

Le but est de réaliser un algorithme permettant de reconnaître la forêt, l'eau, la mer, les habitations, la route, etc. Le tout, en utilisant les algorithmes de Deep Learning. Nous attendons plus de précisions sur les données qui seront entrées au final, et la limite de reconnaissance d'objet pour le moment.

## Synthèse des tutos

### *Algorithmes*

Utilisation d'un algorithme de deep learning de classification. Celui-ci possédera des entrées relatives aux formats d'images. Il possédera n couches cachées de neurones (définies de manière arbitraire). Il y aura autant de sorties que de classes définies (forêt, la mer, la route ...).

On se basera sur le principe de convolution qui permet d'analyser de grandes images à moindre coût. Nous utiliserons aussi du dropout pour réguler le nombre de neurones en cours d'exécution (ce qui permet d'éviter de calculer le nombre de neurones optimal).

On peut coder l'algorithme de génération de réseau de neurones de manière générique, il pourra ensuite nous resservir pour reconnaître des objets plus précis. On peut aussi mettre en place une hiérarchie (de fichier, de classe ou de configuration) permettant de créer des sous réseaux de neurones. (ex: eau -> mer, rivière, lac )

### *Formats des Images*

On utilisera le format TIFF (Tagged Image File format).

Ce format permet de labelliser les images en premier lieu pour l'entraînement du réseau de neurones. Il permet de stocker des images bitmap, celle-ci permettent de ne pas perdre de qualité d'images. Chacun des pixels de l'image bitmap possède une ou plusieurs valeurs décrivant sa couleur.

Au niveau des couleurs les formats possibles suivant nous intéressent :

Le format RGB qu'on utilisera en priorité pour le début du projet. Ce format est en "3 bandes", qui correspondent respectivement, au rouge, bleu et vert. Chaque pixel aura alors un certain pourcentage de chaque couleur.

Le format d'image "16 bandes"

# Critique de l'existant

Le code utilise Keras et nous pensons utiliser Tensorflow. Cas nous voulons nous baser sur le tutoriel (cité dans les sources), qui nous paraît plus simple et à jour. Cependant nous utiliserons les données utilisées dans celui-ci, certaines fonctions ainsi que les premiers labels.

## Environnements

### *Gestion de projet :*

Gitlab

Slack

Taiga

### *Développement :*

Python

TensorFlow

Pytorch

# Sources

Code existant :

<https://www.kaggle.com/khushhall/dstl-test/data>

Installation Environnement Python :

[deepllearning.fr/cours-pratiques-deep-learning/installation-de-lenvironnement-python](https://deepllearning.fr/cours-pratiques-deep-learning/installation-de-lenvironnement-python)

<https://deepllearning.fr/cours-pratiques-deep-learning/installation-de-lapi-de-haut-niveau-keras/>

<https://opencv.org/>

Reconnaissance de véhicule :

[blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee](http://blog.kaggle.com/2017/04/26/dstl-satellite-imagery-competition-1st-place-winners-interview-kyle-lee)

Vidéo explicative (deep learning, convolution ....)

[youtube.com/watch?v=BtAVBeLuigI](https://youtube.com/watch?v=BtAVBeLuigI)

Convolution : [natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-reseaux-de-neurones-convolutifs](http://natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-reseaux-de-neurones-convolutifs)

<https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicité/>

Conversion et composition image :

[deepllearning.fr/cours-pratiques-deep-learning/composition-et-conversion-dune-image](https://deepllearning.fr/cours-pratiques-deep-learning/composition-et-conversion-dune-image)

[deepllearning.fr/cours-pratiques-deep-learning/reconnaissance-dimages](https://deepllearning.fr/cours-pratiques-deep-learning/reconnaissance-dimages)

[github.com/priya-dwivedi/Deep-Learning/blob/master/Mask\\_RCNN/Mask\\_RCNN\\_Videos.ipynb](https://github.com/priya-dwivedi/Deep-Learning/blob/master/Mask_RCNN/Mask_RCNN_Videos.ipynb)

algo de classification :

[rfiap2018.ign.fr/sites/default/files/ARTICLES/CFPT2018/Posters/CFPT2018\\_paper\\_postadjian.pdf](http://rfiap2018.ign.fr/sites/default/files/ARTICLES/CFPT2018/Posters/CFPT2018_paper_postadjian.pdf)

Sauvegarder :

[tensorflow.org/guide/saved\\_model](https://tensorflow.org/guide/saved_model)

Format tiff :

<https://deepsense.ai/deep-learning-for-satellite-imagery-via-image-segmentation/>

Différence du nombre de bandes :

<https://support.geocue.com/wp-content/uploads/2014/09/Multi-band-Imagery-and-Bit-Depth.pdf>