# Rapport jour 4

**Problème :**

**Référence :**

## 1. Création de la base des donnés

J’ai écrit une script .m pour créer plusieurs hologrammes en utilisant Matlab / Octave. L’objectif est de générer plusieurs hologrammes avec une quantité différente des sources dans la scène 3D, on commence avec 1 source ponctuelle dans une position 3D calculé de manière aléatoire et on monte jusqu’à 5 sources ponctuelles.

Par rapport à la génération aléatoire des positions 3D des particules, les positions x, y respectent les limites du plan d’hologramme et la position z respect une profondeur maximale et minimale.

Il est important de dire qu’on enregistre trois fichiers, un fichier avec les hologrammes, un fichier avec les images restituées dans une profondeur fixée et un fichier avec toutes les positions des sources.

Exemple : pour un numéro de classes égal à 5 et avec 200 hologrammes de dimension 2mm x 2 mm et de résolution 200 pixels x 200 pixels par classe, les tailles des fichiers avec les hologrammes et les images restituées ont environ 600 Mo. L’image suivante détaille l’exemple précédent :

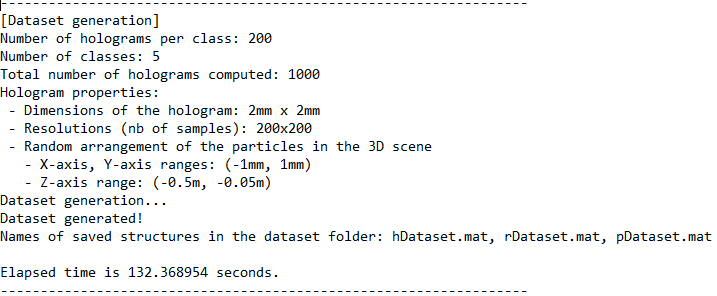


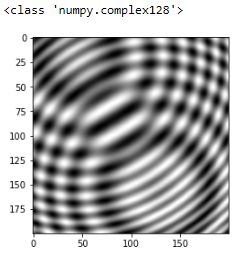
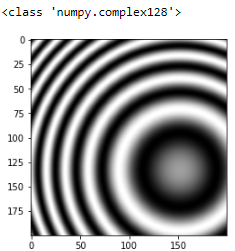
Figure 1 : sortie de la fenêtre de commande du script

#### Exemples :

Exemple des hologrammes générés :



Figure 2: code pour load la base de données



*Figure X : examples des hologrammes avec 1 source (à gauche) et deux sources (à droite)*

#### Problèmes résolues :

Le script a généré des valeurs réelles pour les hologrammes et les images restituées, par contre ce problème a été déjà résolu et maintenant on a des valeurs complexes comme on peut vérifier avec l’image ci-dessus.

#### Problèmes à résoudre :

Quelques améliorations à faire :

1. Pour l’instant, la restituion de tous les hologrammes est fait pour une même pronfondeur definie au début du code. De cette façon, les images restitués ne sont pas utiles, car la function de restituion d’image focalise la restituion seulement pour une distance dans l’axe z, donc on n’arrive pas à trouver le source avant de faire la restituion. Peut-être, il n’est pas necessaire cette donné dans l’étude.

[insérer des images ?]

#### Observations :

Quelques remarques à faire :

1. Quand j’ai utilisé l’Octave pour exécuter le script pour la génération de la base de données, j’ai trouvé quelques limitations et problèmes dans le processus de sauvegarde. Si on veut faire, par exemple, une base des données avec 5 classes et avec plus de 25 hologrammes par classe, on a des erreus pour sauvegarder les dictionnaires dans le format Matlab. Je remarque aussi que les fichiers sont plus gros quand ils sont générés par l’Octave. Donc, c’est intéressant d’éviter l’Octave.

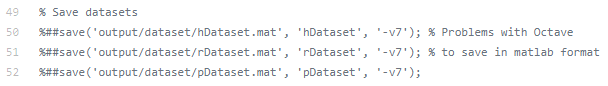


Figure 3 : code exemple pour sauvegarder en format Matlab

## 2. Convolutional Neural Network (CNN)

Pour essayer de résoudre le problème de classification, j’ai utilisé une réséau de néurones. J’ai décidé utilisér l’architecture d’une réseau neural convolutif car cette réseau est beaucoup utilisé pour les applications avec des images.Par rapport à la bibliothéque choisie pour faire l’implementation, j’ai choisi utiliser la biblioheque Keras, car elle est une API facile à utiliser et est le plus populaire *framework* pour l’apprentissage profond (*Deep learning*).

Il faut rappeler que les réseau des néurones sont des algorithmes stochastiques, c’est-à-dire que le même algorithme sur les mêmes données peut entraîner un modèle différent à chaque fois que le code est exécuté.

#### Tests :

Pour une base de donnés avec 125 hologrammes, où pour chaque classe ( 5 classes au totale) on a 25 hologrammes, j’ai crée une CNN avec les parametres suivantes, il faut remarquer que j’ai utilisé les mêmes parametres de la réference utilisé [2].

#### Problèmes à resoudre :

- wd results a 8 images par hologramme

- epochs, parameters

#### Reférences :

Nodes, epochs, batch

#### Reférences :

[ 1] Bibliotheque keras :

<https://keras.io/>

[2] Building a Convolutional Neural Network (CNN) in Keras :

<https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5>

[3] Your First Deep Learning Project in Python with Keras Step-By-Step :

<https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>