ChampySeed Etape 3 Iteration 2

Benoit Maxence Bastien

September 18, 2021

0.1 Rappel Iteration 1

Nous avons lancé le téléchargement de notre base de donnée. Celle ci étant très importante(plus de 2 millions), nous avons opté pour un téléchargement qui redimensionne en 224*224 directement après celui ci pour rendre le stockage physique possible. C'est plus long mais du coup cela reste possible.

Mise en place d'un premier workflow basé sur le modèle pré-entrainé VGG16 . Test sur un jeu réduit de 4 familles de champignons avec environ 500 photos par famille.(Ces choix étaient purement pratique pour un premier essai) Les images étaient présentent en local sur le HDD.

Dans le notebook livré, comprenant le cheminement complet, se trouve:

- Définition des répertoires contenant les images des 4 familles de champignons
- Affichage de quelques images
- Paramétrage d'un générateur d'image
- Entraînement avec un VGG16 sur un set de 2000 images et 4 familles présentent sur le HDD
- Graphique montrant la perte et la précision
- Test de prédiction avec 1 image de chaque famille

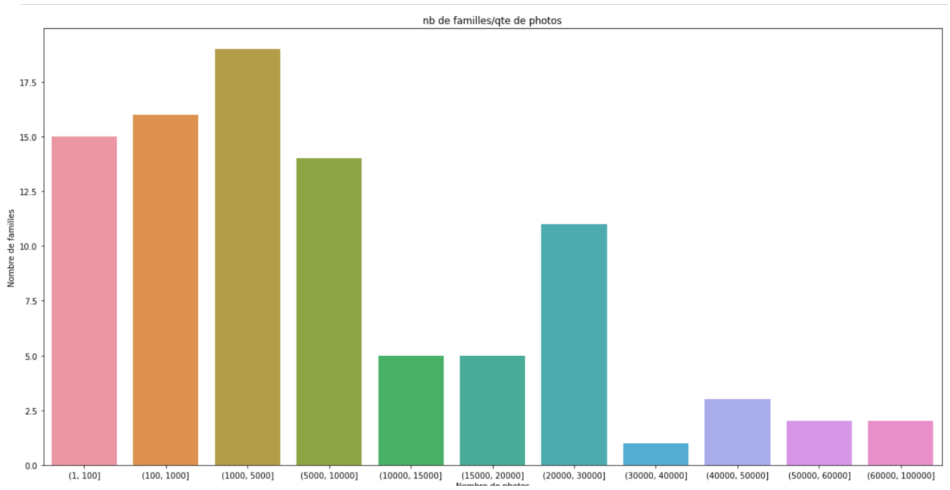

Figure 1: Demande de Théo pour mieux voir la répartition des familles selon le nombre de photos disponibles au sein de celles-ci

1 Iteration 2: Mise en place

1.1 Identification de photos à éliminer

Suite au téléchargement intensif nous avons pu trouver un point commun avec des photos non pertinentes comme celle-ci.(Figure 2)

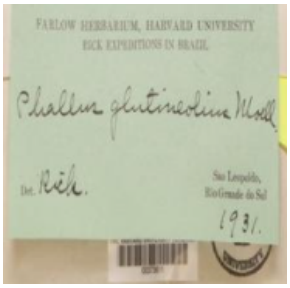

Figure 2

Nous avons pu vérifier que la source `http:\sweemgum.nygd.org` correspondait à ces images. Etant donné que cette source représente moins de 5% des images, nous avons décidé de les écarter pour éviter de perturber l'apprentissage. Il semblerait que ce site ne montre que des spécimens, des échantillons séchés.

1.2 Identification des "Genus" que nous souhaitons identifier

Nous avons souhaité sélectionner 10 classes pour notre projet. Après avoir hésité entre l'identification de Family ou de Genus, nous avons vérifié que nous avions suffisamment de photos au niveau des Genus pour en sélectionner 10 ayant entre 5000-10000 photos. Notre choix est basé sur le nombre et aussi sur une diversité de style (Visionnage des photos pour voir si ils avaient des caractéristiques spécifiques) Voici la liste des Genus choisis:

Ordre	Famille	Genus	Nombre de fichiers
Boletales	Paxillaceae	Paxillus	7619
Phallales	Phallaceae	Clathrus	8676
Geastrales	Geastraceae	Geastrum	13534
Boletales	Sclerodermataceae	Scleroderma	10509
Polyporales	Phanerochaetaceae	Byssomerulius	5387
Polyporales	Fomitopsidaceae	Phaeolus	10284
Agaricales	Agaricaceae	Agaricus	21809
Agaricales	Amanitaceae	Amanita	123681
Auriculariales	Auriculariaceae	Auricularia	20434
Cantharellales	Cantharellaceae	Craterellus	8219

1.3 Téléchargement des images

Dans notre récolte de photos, beaucoup d'erreurs(au début, et de moins en moins maintenant) doivent être anticipés dans notre boucle afin que celle ci ne s'arrête pas.Que cela soit du format de la photo, du blacklistage de certain site, des problèmes de réponse du serveur distant, il y a un certain nombres de problèmes qui étaient susceptibles de couper notre téléchargement. Nous avons petit à petit étoffé notre boucle afin que celle ci ,"gère" ces erreurs.(Ou du moins que cela n'arrête pas notre boucle)

Nous avons également ajouté la possibilité de télécharger des images en donnant la "family" ou le "genus" en paramètre. Ainsi nous avons pu concentrer notre téléchargement sur les "genus" sélectionnés.

2 Iteration 2: Côté modèle

2.0.1 Passage de flow_from_directory en flow_from_dataframe

Nous créons un sous-dataset qui sélectionne au hasard 4000 images par genus afin d'avoir un nombre de photos équivalent par classe. Le même tableau est ensuite utilisé pour les différents modèles testés.

Notebook sur le GitHub

2.1 Test de 3 autres modèles pré-entraînés

Nous nous sommes partagé les 3 nouveaux modèles pré-entraînés suivants :

- EfficientNetB0
- Inception V3
- RestNet50

Le notebook VGG16 est également mis à niveau pour refaire les calculs sur les mêmes données.
Voir les Notebooks dans GitHub pour le code

2.2 Comparaison des performances

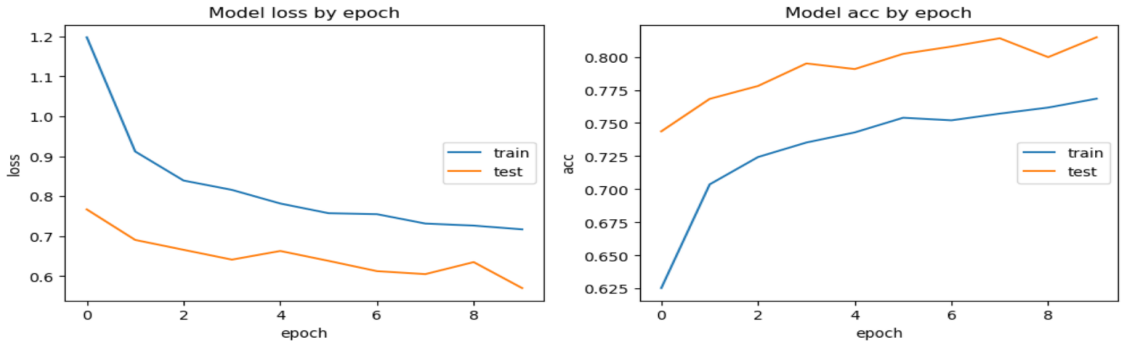

Figure 3: VGG 16(première itération)

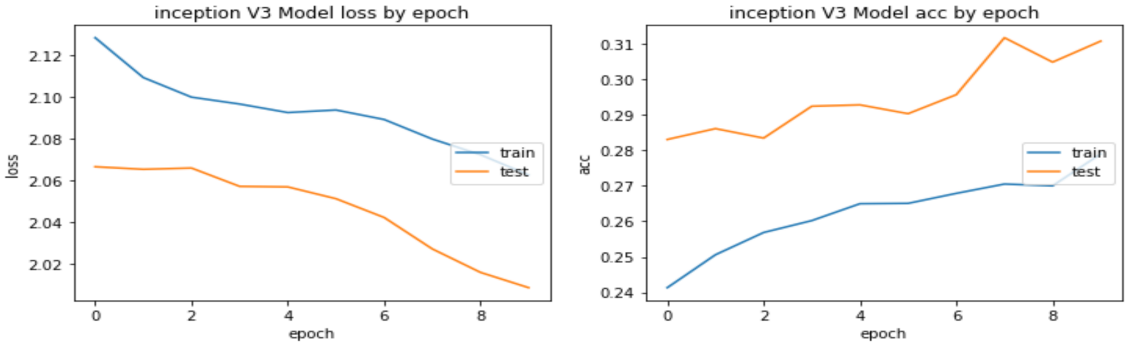

Figure 4: Inception V3

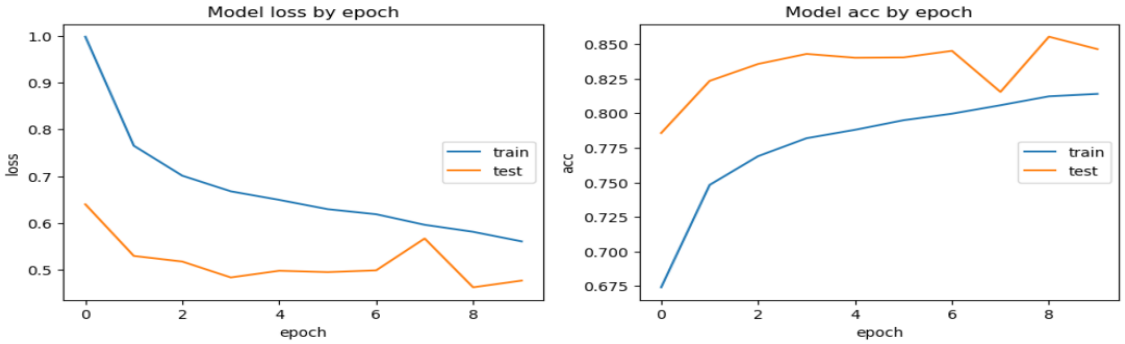

Figure 5: ResNet 50

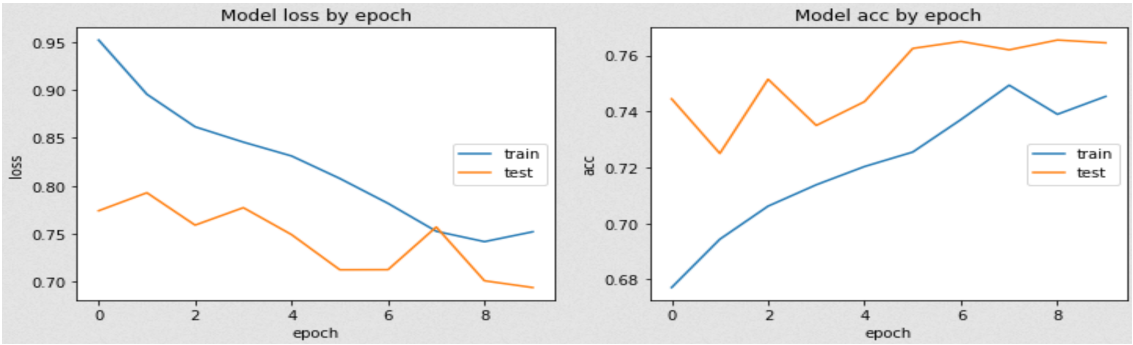

Figure 6: EfficientnetB0

Figure 7: Comparaison des performances

3 Conclusions et axes d'améliorations

3.1 Conclusion

Nous avons différents modèles qui fonctionnent. Il est possible que nous recherchions encore un modèle (il y a plein de différentes versions de ces modèles) qui se trouvera le mieux adapté à notre jeu de donnée. Voici une petite carte qui semble montrer les possibilités de toutes ces différentes versions. (Figure 8) Nous espérons avoir finalisé le choix complet de notre modèle(et de version) pour la prochaine itération. Une fois ce choix fait nous devrons commencer à déterminer quelles seront nos objectifs finaux à partir de notre jeu de données.

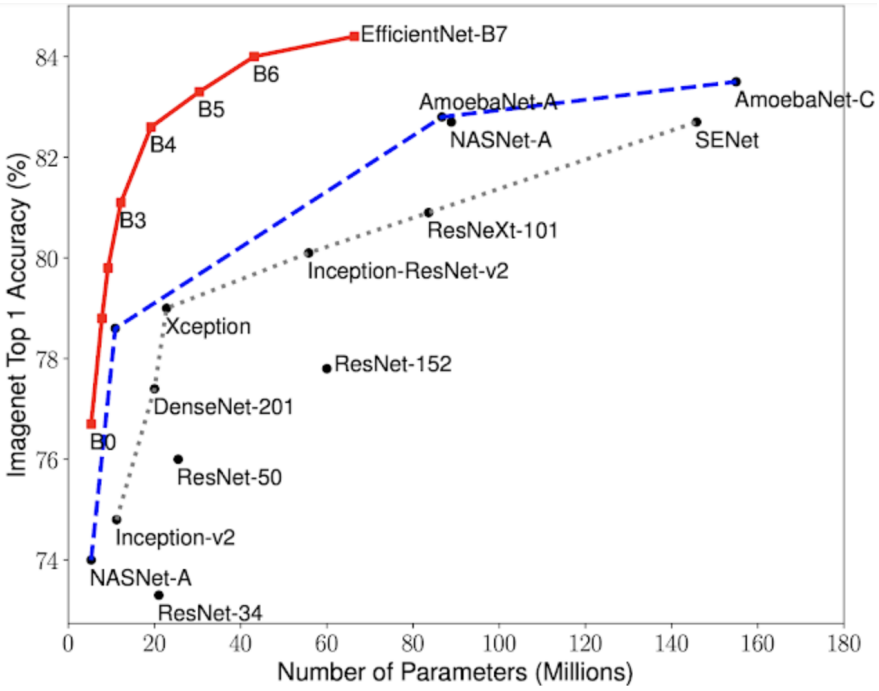

Figure 8: Carte des modèles

Nous aimerions aussi pouvoir parfaire ce modèle en dégelant certaines couches et de comprendre un peu mieux pourquoi tel choix entraine tel résultat.

Nous aimerions aussi réaliser un traitement des images pour essayer de retirer encore quelques images inutiles , corrompues, ou autre. Il semble aussi que dans une grande majorité des cas les champignons soient au centre de la photo. Faire un masque est peut être quelque chose qui est envisageable.

3.2 Difficulté rencontrées et gérées ou abandonnées

- Notre code prenait 4000 images au hasard dans les genres mais comme ces familles n'était pas completes, on obtenait des fichiers absents. Cela a été gérer en supprimant ces entrées avec un os.remove
- Le choix des batch_size , steps_per_epoch nous a donné pas mal de fil à retordre.

3.3 Difficultés rencontrées encore en cours

- Des images pas encore téléchargées
- Des difficultées à faire fonctionner son GPU pour Bastien
- Interprétabilitée des modèles
- La carte graphique de Max étant une Radeon il existe des incompatibilitées difficiles à gérer