

Classification supervisée d'images d'éoliennes

Auteurs : Théo Lahontâa, Elias Aboukacem

Encadrant : Camille Castera

UE : Introduction à l'apprentissage profond

1 Premières expérimentations sur le petit jeu de données

1.1 Modèles simples et limites initiales

Nous avons commencé par entraîner un modèle composé de couches denses précédées d’une opération de *flatten*. Cette approche supprimant l’information spatiale des images, les performances restaient faibles, avec une validation accuracy autour de 0.2. L’ajout de couches convolutives améliorait partiellement les résultats (environ 0.4), ce qui a confirmé la nécessité d’utiliser des architectures adaptées au traitement d’images. Nous avons donc abandonné les modèles basés sur des vecteurs pour passer à des modèles pré-définis.

1.2 Passage à des architectures convolutionnelles

Nous avons testé plusieurs architectures convolutionnelles standards. ResNet18, associé à l’optimiseur AdamW, obtenait environ 0.85 de validation accuracy sur le petit jeu de données. Étant donné la taille réduite de l’ensemble d’entraînement, nous avons intégré de la *data augmentation* (rotations, contrastes) afin de limiter le surapprentissage. Le jeu étant équilibré, nous n’avons pas utilisé de pondération des classes. ResNet34, EfficientNet et DenseNet ont également été testés, sans amélioration notable par rapport à ResNet18.

L’évaluation reposait sur la matrice de confusion, l’évolution des courbes de loss et d’accuracy, les scores de confiance supérieurs à 0.85 dans l’erreur, ainsi que la courbe de calibration. Plusieurs images mal classées étaient très sombres ou très lumineuses ; même avec une augmentation ciblée sur le contraste, le modèle ne généralisait pas correctement ces cas. Compte tenu de l’ensemble des résultats, nous avons retenu **ResNet18** comme architecture de référence pour le passage au grand jeu de données.

2 Entraînement à grande échelle et modèle final

2.1 Montée en volume et réduction de résolution

Pour observer la dynamique d’apprentissage avec davantage de données, nous avons d’abord utilisé un sous-ensemble de 2 000 images, redimensionnées en 64×64 afin de réduire le temps d’entraînement. Avec ResNet18 et AdamW, la validation accuracy atteignait environ 0.94 en trois époques.

2.2 Transfert learning et régularisation

Nous avons ensuite appliqué le transfert learning avec les poids pré-entraînés sur ImageNet. La validation accuracy montait autour de 0.98, mais la validation loss devenait instable, ce qui révélait un surapprentissage. De plus un biais était introduit par l'utilisation répétée du même jeu de test et la recherche du minimum d'erreurs de prédiction sur ce dernier.

Pour y remédier, nous avons intégré plusieurs techniques de régularisation :

- *label smoothing* ;
- *dropout* ;
- *early stopping*.

Nous avons également divisé le grand jeu de données en 80 % d'entraînement et 20 % de validation, les deux ensembles étant équilibrés. La data augmentation était appliquée uniquement aux données d'entraînement.

2.3 Recherche d'hyperparamètres et choix définitifs

Nous avons testé différents niveaux de *label smoothing*, de *dropout*, ainsi que de *weight decay*. Nous avons comparé AdamW et SGD, avec ou sans scheduler. Les résultats montraient que :

- AdamW offrait une convergence plus rapide ;
- l'ajout d'un scheduler ne donnait pas de gain notable ;
- un taux de *label smoothing* et de *dropout* autour de 0.15 stabilisait bien l'apprentissage.

Avec ces réglages, la validation accuracy dépassait 0.97 à l'époque 14 avant l'apparition du surapprentissage. Comme l'early stopping n'était pas assez restrictif pour explorer le plus loin possible, on a réduit le nombre d'epoch à 14 pour éviter le surapprentissage et en même temps avoir une meilleure convergence.

2.4 Entraînement final

À partir des hyperparamètres retenus, nous avons ré-entraîné ResNet18 sur les images en taille réelle afin de produire le modèle final utilisé pour la classification d'éoliennes et avons retenu les poids de l'epoch 12.