**Rapport de Projet : Classification des Traits Botaniques**

~

Réalisé par AKUESON Julian, TRAN Michael et MAHENDRAN Sujeeban

**Introduction**

Ce projet réalisé vise à classifier différents traits botaniques à partir d'images d'herbier en utilisant plusieurs modèles de Deep Learning. Quatre modèles avancés pré entrainé sur le dataset ImageNet ont été testés pour évaluer leurs performances :

* **CoAtNet**
* **Davit**
* **SwinV2**
* **Apple MobileViT v2**

L'objectif principal était de comparer ces modèles sur des métriques clés telles que la perte, l'accuracy, le F1-score et le recall.

**Préparation des données**

**1. Structure du dataset**

Le dataset contient des images avec les labels suivants :

* Épines : Présence ou absence.
* Bordure des feuilles (BF) : Dentée ou lisse.
* Taille des feuilles (TF) : Petites, moyennes ou grandes.

Chaque classe a été prétraitée pour garantir une structure compatible avec les modèles pré-entrainés sur le dataset ImageNet.

**2. Splitting des données**

Les données ont été divisées en trois ensembles distincts (60/20/20) :

* **Entraînement** : 281 images.
* **Validation** : 108 images.
* **Test** : 149 images.

Les images ont été organisées en dossiers selon leur étiquette et un fichier CSV a été généré pour référencer les annotations. Cette approche a permis une gestion plus claire et une intégration fluide dans les DataLoaders de PyTorch.

**3. Problèmes rencontrés**

* **Qualité des images**: Certaines images ne sont pas du tout exploitables et nuise au bon entrainement du modèle, prenons l’exemple de l’image ci-dessous

**Une image contenant texte, capture d’écran, lettre, écriture manuscrite

Description générée automatiquement**

(ref : ALF035290.jpg)

Une segmentation de l’image ne serait pas suffisante pour permettre au modèle d’extraire les caractéristiques de l’exemple ci-dessus.

* **Déséquilibre des classes** : Comme l'illustre le graphique ci-dessous, il y a un fort déséquilibre intra classe. Ce déséquilibre a biaisé le modèle en faveur des attribut majoritaires de chaque classe rendant les résultats moins fiables dans des contextes réels.

Une image contenant texte, capture d’écran, diagramme, Tracé

Description générée automatiquement

* **Nettoyage des données** : Le fichier d’annotation contient, énormément de ligne inutile à l’entrainement du modèle, de plus il a fallu normaliser et nettoyer les éléments du dataset afin de pouvoir réaliser une jointure entre les images et les annotations.
* **Problème** **multi classe/multi-label** : Conflit du dataset multi-classe et multi-label, le problème a été provoqué par la non-binarité de la classe TF.

**4. Transformations appliquées**

* Redimensionnement à 256x256 pixels.
* Normalisation adaptée aux spécifications des modèles préentrainés.
* Data augmentation pour le dataset d’entraînement

**Description des modèles préentrainés :**

Les modèles suivants ont été choisis pour leurs caractéristiques avancées et leur capacité à traiter des données complexes :

1. **CoAtNet :** « *Marrying Convolution and Attention for All Data Sizes* ». Ce modèle combine les avantages des convolutions et des transformeurs pour capturer à la fois les détails locaux et les relations globales dans les images. Son architecture hiérarchique le rend idéal pour des données visuelles complexes.
2. **Davit :** « *Dual Attention Vision Transformers* ». Doté d'une efficacité optimisée, ce modèle est conçu pour offrir un équilibre entre précision et rapidité. Sa compatibilité avec le framework timm permet une intégration facile et rapide.
3. **SwinV2 :** « *Scaling Up Capacity and Resolution* ».Reconnu pour ses performances exceptionnelles, SwinV2 repose sur des fenêtres glissantes qui divisent l'image en patchs, offrant ainsi une meilleure efficacité et une résolution adaptée aux grandes images.
4. **Apple MobileViT v2 :** Conçu pour les dispositifs mobiles, ce modèle léger offre des performances impressionnantes tout en maintenant une efficacité énergétique. Il convient parfaitement aux systèmes embarqués.

**Analyse de l’entrainement des Modèles sur le dataset**

**1. CoAtNet**

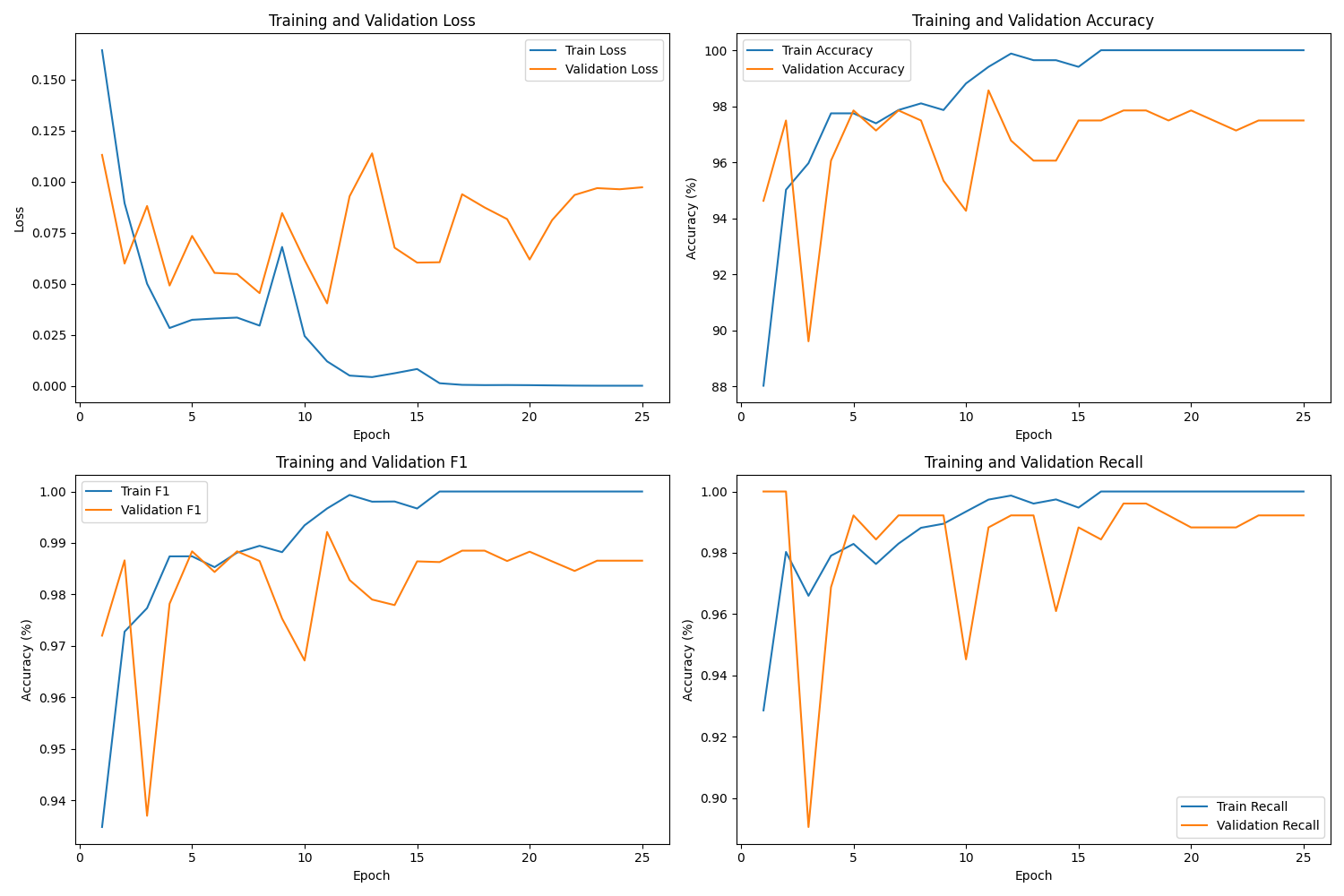
* **Observations :**
  + La convergence est visible après environ 15 époques.
  + L'accuracy de validation atteint 98%, mais des variations notables dans la perte indiquent problème de surapprentissage.

L'implémentation de CoAtNet avec le framework **timm** a grandement facilité l'utilisation du modèle et de ses transformations. Voici un exemple de code illustrant cette intégration :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Graphique d’entrainement :**

****

**2. Davit**

* **Observations :**
  + Des performances solides en validation avec une accuracy équilibrée.
  + Les variations importantes dans la perte de validation indiquent un problème de surapprentissage.

Comme pour CoAtNet, **timm** simplifie également l'implémentation de Davit, réduisant ainsi les efforts nécessaires pour adapter les transformations.

**Graphique d’entraînement :**

Une image contenant texte, diagramme, ligne, Parallèle

Description générée automatiquement

**3. Apple MobileViT v2**

* **Observations :**
  + Une convergence rapide avec des courbes stables.
  + Le modèle présente une performance équilibrée avec des pertes faibles et des scores de précision, de F1 et de recall proches de 99%.
* **Graphique d’entrainement :**

Une image contenant texte, diagramme, ligne, carte

Description générée automatiquement

**4. SwinV2**

* **Observations :**
  + Le modèle offre des performances très élevées en accuracy et F1-score.
  + Les variations importantes dans la perte de validation indiquent un problème de surapprentissage.
* **Graphique d’entrainement : Une image contenant texte, diagramme, ligne, Tracé

  Description générée automatiquement**

**Comparaison Globale**

**Tableau Récapitulatif des Performances**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modèle** | **Accuracy (%)** | **F1-score** | **Recall** |  |
| **CoAtNet** | **98.5** | **0.97** | **0.96** |  |
| **Davit** | **97.3** | **0.96** | **0.95** |  |
| **MobileViT v2** | **99.0** | **0.99** | **0.99** |  |
| **SwinV2** | **99.3** | **0.99** | **0.99** |  |

**Conclusion**

En vue des résultats non représentatif de la réalité il est difficile d’émettre un jugement sur le finetuning de ces modèles, cependant d’une manière générale voici ce que l’on peut en conclure.

* MobileViT v2 est idéal pour des dispositifs à faible puissance grâce à sa légèreté tout en maintenant des performances comparables aux modèles plus lourds.
* SwinV2 offre les meilleures performances globales, mais au coût d'une complexité et d'une taille de modèle accrues.
* L’utilisation du Framework **timm** offre un avantage non négligeable pour l’utilisation des modèles.

**Prochaines étapes**

1. Augmentation des données : Introduire de nouvelles images plus qualitatives et des annotations plus variés afin d’augmenter la diversité intra classe.