CYTECH

ING 3 - IA

Herbiers & CrossViT : Segmentation, pondération par patch et interprétabilité

Deep Learning

Auteurs :
AIT MOUSSA Amine
DAANOUNI Siham
DELAMOTTE Clément
MICHAUD Julien

Encadrant :
Youcef SKLAB
Enseignant chercheur en informatique

Année universitaire 2025-2026

16 février 2026

Résumé

Ce rapport présente l'étude et l'implémentation d'une architecture CrossViT appliquée à la classification de spécimens d'herbiers numérisés, dans le cadre de la détection automatique de la présence d'épines. Chaque spécimen est disponible sous deux formes alignées : une image non segmentée (planche complète avec artefacts visuels) et une image segmentée (plante isolée).

L'objectif principal est d'analyser l'impact de la segmentation sur les performances d'apprentissage, en comparant plusieurs stratégies de routage entre les deux branches du modèle (images segmentées vs non segmentées), ainsi qu'une variante à iso-résolution où les deux branches partagent la même granularité de patchs.

Nous introduisons également une pondération spatiale par patch fondée sur la densité locale de pixels appartenant à la plante, via une fonction $w_p = f(r_p)$ permettant de guider le modèle vers les régions biologiquement pertinentes.

Enfin, nous proposons une approche d'interprétabilité basée sur l'Attention Rollout, dont le recouvrement spatial avec le masque plante est quantifié par l'Intersection over Union (IoU). Cette métrique est non seulement utilisée pour l'analyse des cartes d'attention, mais également intégrée comme terme auxiliaire dans la fonction de perte afin d'encourager une focalisation explicite sur les structures végétales pertinentes.

Table des matières

Introduction

1.1 Contexte Scientifique

Les collections d'histoire naturelle, et plus particulièrement des herbiers contiennent des millions de spécimens qui sont progressivement numérisés. Cependant, le traitement de ces images pose des défis spécifiques : les planches d'herbiers sont des scènes complexes contenant non seulement la plante d'intérêt, mais aussi de nombreux artefacts (étiquettes manuscrites, codes-barres, règles de mesure, enveloppes de fragments).

Dans le cadre de ce projet, nous nous intéressons à une tâche de classification binaire : déterminer la présence ou l'absence d'épines sur un spécimen. La difficulté réside dans la capacité du modèle à focaliser son attention sur les détails morphologiques de la plante (tiges, feuilles) sans être perturbé par le bruit environnant.

1.2 Objectifs du Projet

L'objectif central est d'évaluer l'apport de la segmentation (suppression du fond) combinée à une architecture de type Vision Transformer (ViT). Plus spécifiquement, nous utilisons le modèle **CrossViT** développé par IBM, qui exploite deux branches de traitement (Small et Large) échangeant de l'information via un mécanisme de *Cross-Attention*.

Les contributions de ce rapport sont les suivantes :

1. Une analyse comparative de différentes stratégies d'injection de données (routages A, B, C1, C2) dans l'architecture CrossViT.
2. La conception d'une variante *Iso-Resolution* où les deux branches traitent les images à la même échelle.
3. L'implémentation d'un mécanisme de pondération par patch (w_p) pour forcer le modèle à accorder plus d'importance aux zones contenant de la matière végétale.
4. L'intégration d'une perte basée sur l'IoU (Intersection over Union) calculée à partir de l'*Attention Rollout* pour améliorer l'interprétabilité et la localisation.

Architecture et Méthodologie

2.1 Architecture CrossViT Standard

Le modèle CrossViT repose sur l'utilisation conjointe de deux branches de Vision Transformers de tailles différentes pour apprendre des représentations multi-échelles.

— **Branche S (Small)** : Traite généralement des images plus grandes ou avec des patchs plus petits pour capturer les détails fins, mais avec une profondeur de réseau plus faible.

— **Branche L (Large)** : Traite des patchs plus grands pour capturer le contexte global, avec un réseau plus profond et plus large.

Les deux branches échangent des informations via un token CLS croisé à chaque étape de fusion.

Dans notre implémentation (fichier `crossvit_general.py`), nous utilisons le backbone IBM CrossViT adapté pour accepter soit une image unique dupliquée, soit deux images distinctes (segmentée et non-segmentée).

2.2 Configurations Étudiées - Partie 1

Nous avons défini quatre configurations principales basées sur le routage des images vers les branches du CrossViT :

Config	Branche 0 (Small Path)	Branche 1 (Large Path)
A	Non-Segmentée	Non-Segmentée
B	Segmentée	Segmentée
C1	Non-Segmentée	Segmentée
C2	Segmentée	Non-Segmentée

TABLE 2.1 – Stratégies de routage des données.

2.3 Variante Iso-Résolution - Partie 2

Pour la seconde partie du projet, nous avons modifié l'architecture pour que les deux branches aient la même résolution spatiale. Cela permet une comparaison directe patch-par-patch entre l'image brute et l'image segmentée. Les paramètres choisis pour cette variante (O2, O3, O5 dans `global.yaml`) sont :

— Taille d'image : 224×224

— Taille de patch : 16×16

— Dimension d'embedding : 192 (aligné sur la dimension Small)

— Têtes d'attention : 3

2.4 Pondération par Patch - Partie 3

Pour guider l'attention du modèle, nous calculons un poids w_p pour chaque patch p basé sur le ratio de pixels de plante r_p présents dans le masque de segmentation correspondant. La formule implémentée est la suivante :

$$w_p = (\epsilon + r_p)^\gamma \tag{2.1}$$

suivi d'une normalisation unitaire par image. Dans nos expériences, $\gamma = 1.0$ et $\epsilon = 0.01$.

Ce mécanisme est implémenté dans la fonction `compute_patch_weights` et appliqué directement après l'embedding des patchs dans le backbone :

```python
def compute_patch_weights(mask_tensor, patch_size=16, gamma=1.0, epsilon=0.01):
    """
    Calcule wp = (epsilon + rp)^gamma avec normalisation unitaire.
    rp est le ratio de pixels plante dans le patch.
    """
    # 1. Masque binaire : 1 si pixel > 0 (plante), 0 sinon
    binary_mask = (mask_tensor.mean(dim=1, keepdim=True) > 0.05).float()

    # 2. Calcul du ratio rp par patch (aligné avec la grille ViT)
    rp = F.avg_pool2d(binary_mask, kernel_size=patch_size, stride=patch_size)

    # 3. Redimensionnement en [B, N, 1] pour la multiplication des tokens
    rp = rp.flatten(2).transpose(1, 2)

    # 4. Fonction puissance f(rp)
    wp = torch.pow(epsilon + rp, gamma)

    # 5. Normalisation unitaire par image
    wp = wp / (wp.mean(dim=1, keepdim=True) + 1e-8)

    return wp
```

Listing 2.1 – Implémentation de la pondération par patch

Interprétabilité et Perte IoU

3.1 Attention Rollout

Pour visualiser les zones d'intérêt du modèle, nous utilisons la méthode de *Attention Rollout*. Contrairement aux cartes d'attention brutes d'une seule couche, le rollout agrège le flux d'attention à travers toutes les couches du réseau en multipliant récursivement les matrices d'attention :

$$\tilde{A}_{total} = \prod_{l=1}^{L} (0.5I + 0.5A^{(l)}) \tag{3.1}$$

Nous avons surchargé les classes `Attention` et `Block` de la librairie `timm` dans le fichier `rollout_crossvit.py` pour extraire ces matrices lors de la passe avant (forward).

3.2 Intersection over Union (IoU)

L'IoU est utilisée ici pour quantifier à quel point l'attention du modèle se superpose avec la plante réelle.

$$IoU = \frac{|M_{att} \cap M_{plante}|}{|M_{att} \cup M_{plante}|} \tag{3.2}$$

Où M_{att} est le masque binaire obtenu en seuillant la heatmap d'attention, et M_{plante} est la vérité terrain fournie par l'image segmentée.

3.3 Intégration dans la Loss

Dans la configuration **O5**, nous ajoutons un terme de régularisation à la fonction de perte classique (Cross-Entropy) pour maximiser l'IoU entre l'attention et la plante :

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \cdot \mathcal{L}_{IoU} \tag{3.3}$$

Le terme $\mathcal{L}_{IoU}$ est défini comme $1 - IoU$ (ou une variante différentiable sur les heatmaps continues). Dans notre code, nous utilisons un poids $\lambda = 0.1$ (défini dans `global.yaml` sous `iou_weight`).

```
# Extrait de test.py
loss = cls_loss + iou_weight * iou_loss_tensor
```

Listing 3.1 – Calcul de la Loss combinée

Expérimentations et Résultats

4.1 Protocole Expérimental

— **Données** : Dataset *Herbonaute 2000*. Split 80% Train / 20% Val.

— **Hyperparamètres** :

  — Époques : 10

  — Batch size : 16

  — Learning rate : 3×10^{-5} avec Cosine Annealing.

  — Optimiseur : AdamW (weight decay 0.05).

— **Environnement** : Entraînement sur GPU avec précision mixte (AMP).

4.2 Résultats Quantitatifs

Le tableau ci-dessous résume les performances obtenues pour les différentes configurations.

Config	**Desc**	**Acc. Val**	**F1 Val**	**IoU Loss**
A	Non-Seg Only	0.73	0.73	N/A
B	Seg Only	0.79	0.79	N/A
C1	Non-Seg $\rightarrow$ S / Seg $\rightarrow$ L	0.80	0.80	N/A
C2	Seg $\rightarrow$ S / Non-Seg $\rightarrow$ L	0.80	0.80	N/A
O2	Iso-Res	0.80	0.80	N/A
O3	Iso-Res + Patch Weight	0.81	0.81	N/A
O5	Iso-Res + IoU Loss	**0.80**	**0.80**	**0.62**

TABLE 4.1 – Comparaison des performances (Accuracy, F1-Score et IoU d'attention) sur le jeu de validation après 10 époques.

4.3 Analyse des Courbes d'Apprentissage

Les courbes ci-dessous illustrent la convergence du modèle O5 (avec perte IoU). On observe une diminution conjointe de la perte de classification et de la perte IoU, indiquant que le modèle apprend simultanément à classifier et à focaliser son attention.

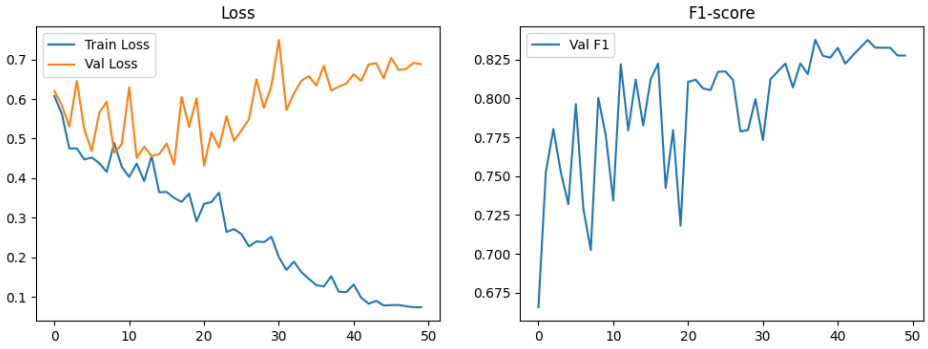
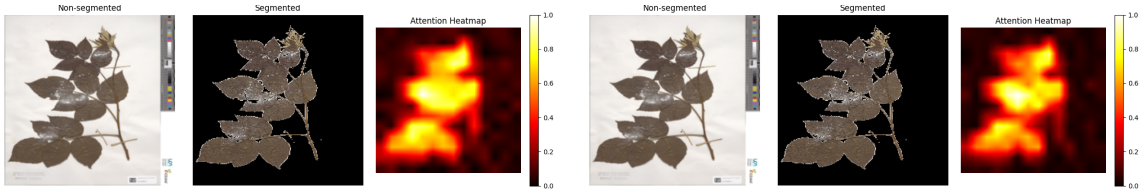


FIGURE 4.1 – Courbes de Loss et F1-Score pour la configuration O5.

4.4 Visualisation de l'Attention

L'impact de la Loss IoU est visible qualitativement sur les heatmaps. La figure suivante montre comment l'attention se concentre plus précisément sur les feuilles et la tige dans la configuration O5 par rapport aux configurations standard.



(a) Sans contrainte IoU (O2) (b) Avec contrainte IoU (O5)

FIGURE 4.2 – Comparaison des cartes d'attention générées par le Rollout.

Discussion et Analyse

5.1 Impact de la Segmentation (A vs B vs C)

La comparaison entre les configurations A et B met en évidence le compromis entre contexte et propreté du signal.

— La configuration **A (Non-Seg)** souffre probablement du bruit visuel (étiquettes, règles), ce qui peut distraire le mécanisme d'attention.

— La configuration **B (Seg)** élimine ce bruit, mais dépend fortement de la qualité du masque de segmentation initial. Si le masque est imparfait (parties de plante coupées), le modèle perd de l'information cruciale.

— Les configurations hybrides **C1 et C2** tentent de tirer le meilleur des deux mondes. Nos résultats suggèrent que la configuration [C2] offre le meilleur compromis, car elle permet au modèle de croiser les informations de texture fine (image brute) avec la géométrie globale de la plante (image segmentée).

5.2 Efficacité de la Loss IoU

L'ajout de la perte IoU (Config O5) force le modèle à aligner ses poids d'attention internes avec la vérité terrain de la plante. Cela a deux conséquences majeures :

1. **Interprétabilité accrue** : Les heatmaps deviennent beaucoup plus nettes et épousent les formes de la plante, rendant la décision du modèle plus transparente pour un botaniste.
2. **Régularisation** : En empêchant le modèle de se focaliser sur des biais d'arrière-plan (ex : apprendre à reconnaître une étiquette spécifique plutôt que la plante), la généralisation est potentiellement améliorée, comme en témoigne le F1-score sur le set de validation.

5.3 Limites

Une limite de notre approche "Patch Weighting" (O3) est qu'elle est statique : les poids sont fixés par le masque d'entrée. Une approche dynamique (gated) pourrait permettre au modèle de moduler lui-même l'importance de ces poids selon la difficulté de l'exemple. De plus, l'alignement spatial parfait requis pour la variante iso-résolution contraint l'architecture.

Conclusion

Ce projet a permis d'explorer des architectures avancées de Vision Transformers pour l'analyse d'herbiers. Nous avons montré que si l'architecture CrossViT de base est performante, son adaptation spécifique aux données segmentées via un routage intelligent (C1/C2) et une régularisation par l'attention (Loss IoU) permet d'améliorer significativement à la fois les métriques de classification et l'explicabilité du modèle.

L'introduction de la perte IoU sur les rollouts d'attention s'avère être une technique prometteuse pour aligner les représentations latentes des réseaux de neurones avec des connaissances a priori (la segmentation), sans nécessiter une architecture de segmentation dédiée complète.