

**Implémentation de DenseWeight et LMFLoss  
afin de traiter des données déséquilibrées**

**Simon DURAND**

## **Introduction**

Le développement d'algorithmes de machine learning pour la classification d'images de plantes et l'analyse de leurs caractéristiques est un domaine en constante évolution. Cela est renforcé par la publication de nouveaux jeux de données, comme le récent CWD30 [1], qui contient des centaines de milliers images haute résolution de différentes espèces de plantes. Nous allons utiliser une partie de ce jeu de données pour améliorer un algorithme de machine learning basé sur le modèle MobileNetV3 [2]. Face au déséquilibre naturel des données de croissance, et à celui que nous avons intentionnellement introduit parmi les espèces, nous appliquons les méthodes DenseWeight [3] et LMFLoss [4] pour traiter ces deux problématiques.

## **Thématique**

L'analyse d'images en agriculture, en particulier la reconnaissance des espèces de plantes et l'analyse de leur croissance, est un aspect crucial de l'agriculture de précision, un domaine qui utilise des technologies avancées pour optimiser la qualité et le rendement des cultures. Cependant, ce domaine, comme de très nombreux domaines de recherche, fait face à un défi majeur : le traitement de données déséquilibrées. Ce déséquilibre peut être un frein à l'amélioration des performances des modèles de machine learning. Ce projet vise donc à optimiser un algorithme de machine learning en gérant efficacement le déséquilibre des données.

## **État de l'art**

Pour traiter le déséquilibre des données en classification et régression, plusieurs méthodes ont été suggérées [5] [6]. Parmi ces méthodes, DenseWeight a été proposée pour gérer le déséquilibre dans les problèmes de régression, en pondérant les erreurs pour donner plus d'importance aux données sous-représentées. La fonction de perte LMFLoss, d'autre part, est utilisée pour ajuster les marges en fonction de la distribution des espèces, dans le but de réduire l'impact de l'inégalité entre les classes majoritaires et minoritaires.

## Présentation du jeu de données

CWD30 est un ensemble de données contenant des images haute résolution de différentes espèces de plantes. Avec plus de 219770 images, présentant une diversité de conditions environnementales, de stades de croissance et de perspectives, cet ensemble offre une grande variété de caractéristiques pour l'étude et la classification des plantes.

Nous nous concentrons sur l'étude des espèces de plantes suivantes : maïs<sup>1</sup> (corn), arachide<sup>2</sup> (peanut), périlla<sup>3</sup> (perilla), millet proso<sup>4</sup> (proso-millet) et sésame<sup>5</sup> (sesame), à l'aide de près de 17000 images. Ces espèces de plantes ont été sélectionnées en raison de leur importance agricole.

Exemples d'images des plantes sélectionnées



Exemples d'images aux divers moments de croissance d'une arachide



<sup>1</sup>Maïs : plante herbacée tropicale annuelle, céréale la plus cultivée dans le monde

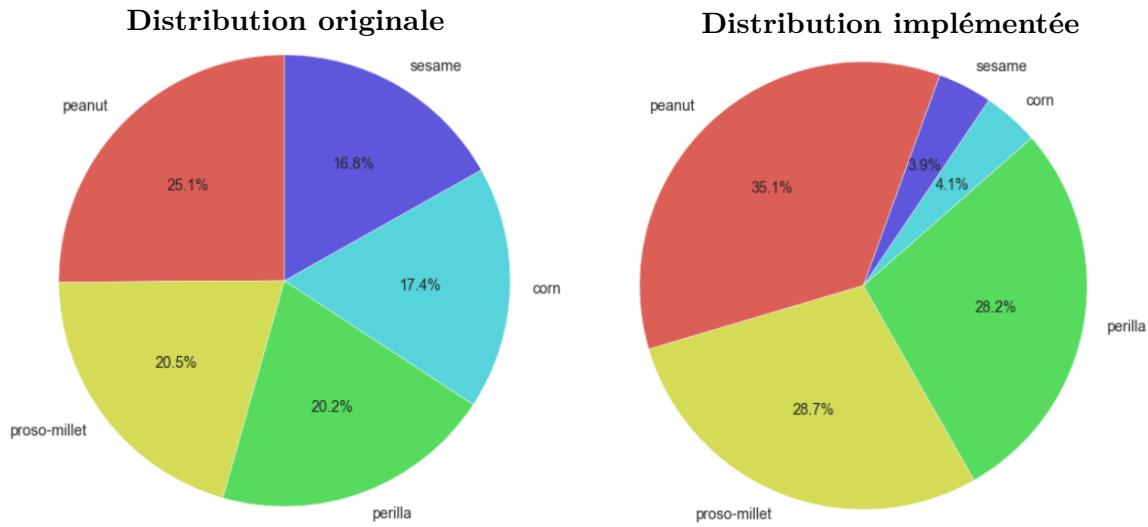
<sup>2</sup>Arachide : plante qui apprécie les sols fertiles, a besoin de chaleur constante et plutôt de sécheresse.

<sup>3</sup>Périlla : plante qui pousse naturellement dans une zone s'étendant de l'Himalaya jusqu'en Birmanie

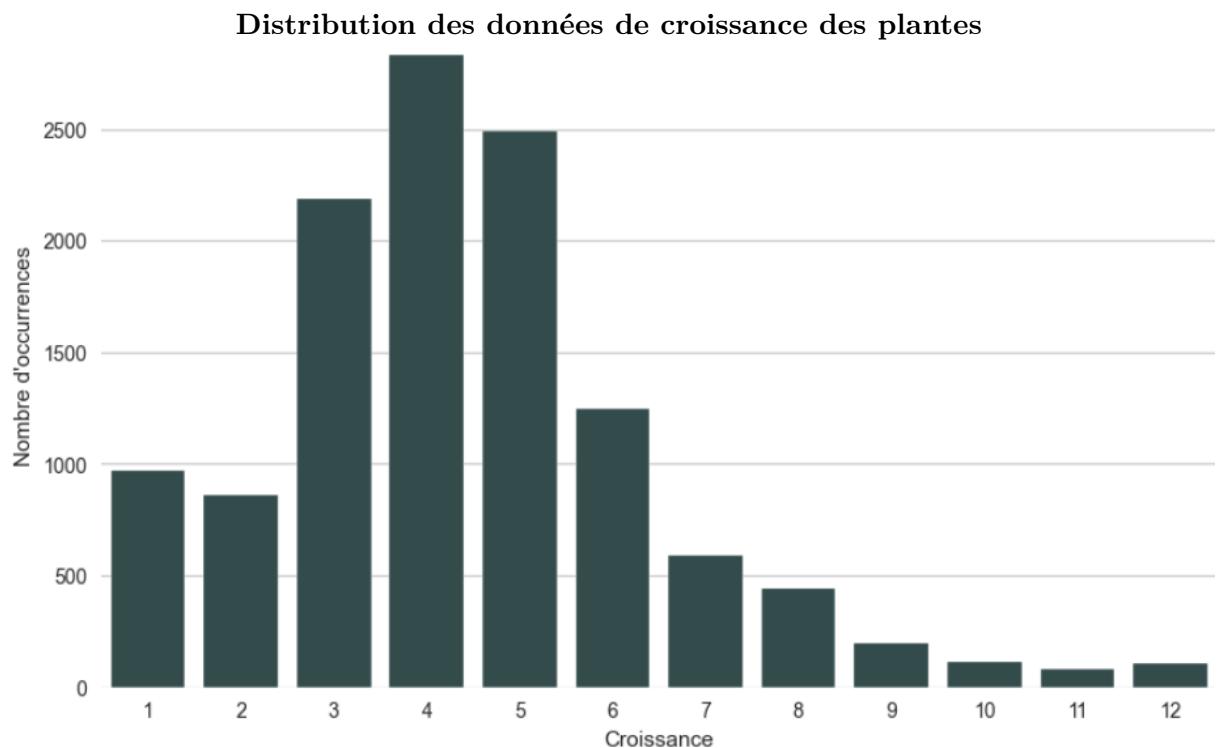
<sup>4</sup>Millet proso : plante robuste étant la céréale ayant le moins d'exigences en termes de consommation d'eau

<sup>5</sup>Sésame : plante annuelle produisant des graines très riches en matières grasses

Ce jeu de données présente un bon équilibre entre les différentes espèces. Afin d'implémenter un déséquilibre dans la répartition des données, nous décidons d'effectuer un sous-échantillonnage sur les espèces « sesame » et « corn », en utilisant la bibliothèque imbalanced-learn de sklearn. Le sous-échantillonnage est réalisé en supprimant 5/6ème des observations de ces classes, ne gardant ainsi qu'un sixième des observations initiales.



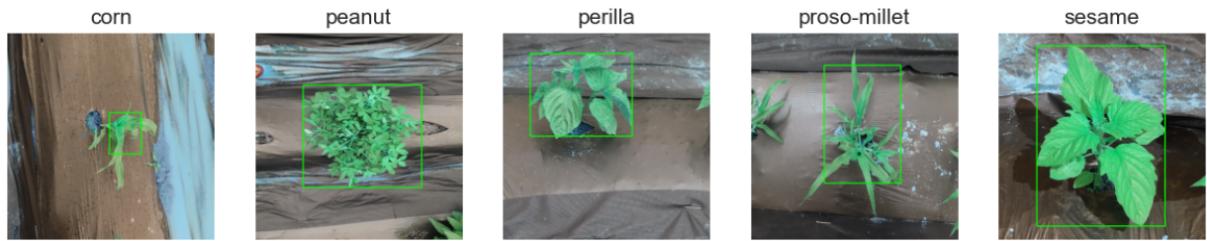
Le jeu de données CWD30 présente également un déséquilibre en termes de distribution des stades de croissance. Un nombre significatif d'images concerne des plantes se trouvant dans des phases de croissance allant de 3 à 5 semaines, ce qui peut représenter un défi pour l'apprentissage automatique et la prédiction précise des autres stades de croissance.



## Préparation des données

La préparation des données est une étape importante pour assurer la performance des modèles de deep learning. MobileNetV3 nécessitant des valeurs de pixels entre 0 et 255, nous n'avons pas normalisé nos données à l'échelle [0,1] comme cela se fait parfois dans d'autres contextes. Nous avons prétraité l'ensemble des images en créant des « bounding boxes », ou boîtes d'encadrement. Ce processus a été réalisé en identifiant et en isolant les régions présentant des nuances de vert spécifiques. Ce travail a été facilité par l'utilisation du package OpenCV.

### Exemples d'images avec sélection des contours des plantes



Les images ont ensuite été rognées en fonction de ces boîtes englobantes pour se concentrer uniquement sur la plante, augmentant ainsi la pertinence de nos analyses ultérieures.

Nous avons ensuite procédé à l'encodage des labels à l'aide de LabelEncoder, ce qui nous a permis de convertir nos cinq espèces de plantes en numéros allant de 0 à 4.

Nous avons divisé notre jeu de données en trois ensembles distincts. Pour ce faire, nous avons utilisé la méthode `train_test_split` de la bibliothèque `sklearn`, qui nous a permis de répartir 70% des images dans l'ensemble d'entraînement, 15% dans l'ensemble de validation et 15% dans l'ensemble de test.

Nous avons créé des générateurs d'images distincts pour chacun de nos ensembles de données en utilisant la fonction `flow_from_dataframe` de la classe `ImageDataGenerator` de la bibliothèque Keras. Ces générateurs nous ont permis non seulement de charger les images directement à partir des chemins spécifiés dans nos dataframes respectifs, mais aussi de les redimensionner automatiquement en 224 x 224 pixels et de les regrouper par lots de 32 images pour optimiser l'apprentissage. Pour l'ensemble d'entraînement, nous avons également intégré une série de légères transformations d'images, telles que des rotations, des zooms et des décalages, afin de renforcer la robustesse de notre modèle face à la variabilité des données réelles.

### Exemples d'images recadrées avec data augmentation



**Labels :** 0 - corn, 1 - peanut, 2 - perilla, 3 - proso-millet, 4 - sesame

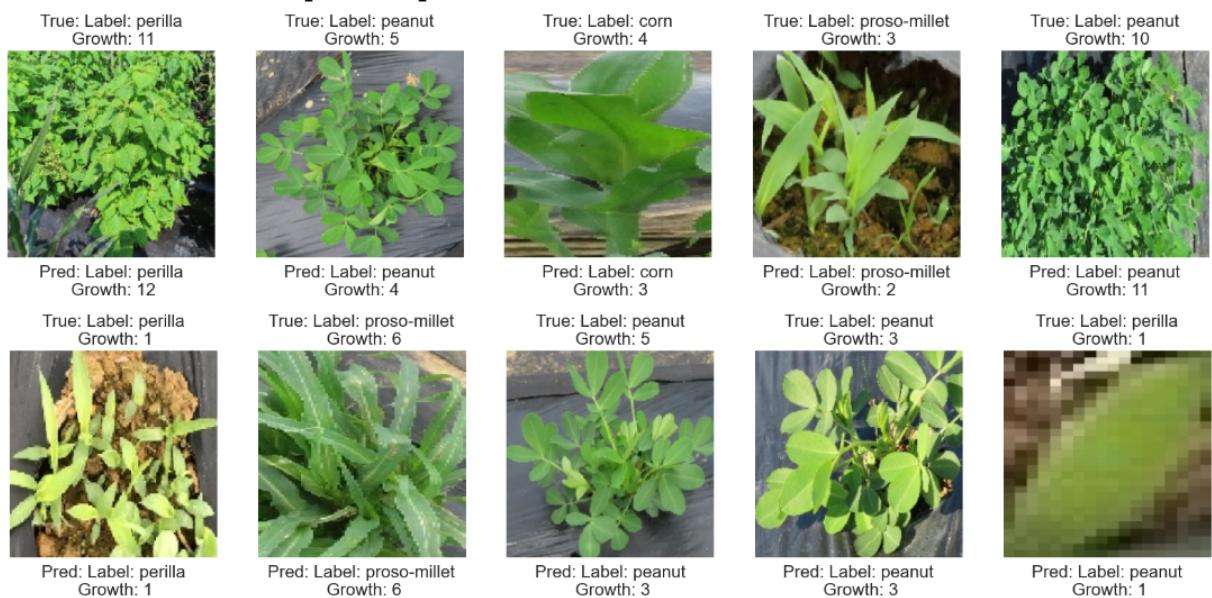
## Modèle MobileNetV3

Nous avons opté pour le modèle MobileNetV3, un choix stratégique pour bénéficier d'une extraction de caractéristiques d'images efficace, accélérer l'apprentissage et minimiser la complexité computationnelle. Nous avons rendu toutes les couches du modèle disponibles à l'apprentissage et avons ajouté des couches de sortie pour prédire les différentes caractéristiques. Pour notre modèle, les pertes ont été calculées par différentes méthodes selon la caractéristique à prédire. Une erreur absolue moyenne a été utilisée pour prédire le stade de croissance des plantes, et l'entropie croisée catégorielle pour en prédire l'espèce.

Pour la prédiction de la croissance, nous avons utilisé l'erreur absolue moyenne (MAE). Ce mode de régression plutôt que de classification pour la croissance des plantes a été choisi car les valeurs suivent un ordre naturel. La régression prend en compte cet ordre intrinsèque, tandis qu'une approche de classification traiterait chaque classe comme distincte et indépendante des autres. Cela nous permet de capturer la relation ordonnée entre les différentes valeurs de croissance. Ensuite, pour le type de plante, qui est une tâche de classification multiclasse, nous avons utilisé le F1-Score macro, une métrique équilibrée pour les tâches de classification multiclasse. Afin d'éviter le surapprentissage, nous avons mis en œuvre une technique d'arrêt précoce, retenant le modèle avec les meilleures performances.

Pour optimiser davantage les performances de notre modèle, nous avons fait appel à KerasTuner pour trouver les meilleurs hyperparamètres à travers 20 époques. L'optimisation a pris en compte la taille des couches denses, le taux d'abandon (dropout rate) et le taux d'apprentissage (learning rate).

### Exemples de prédictions avec le modèle de référence

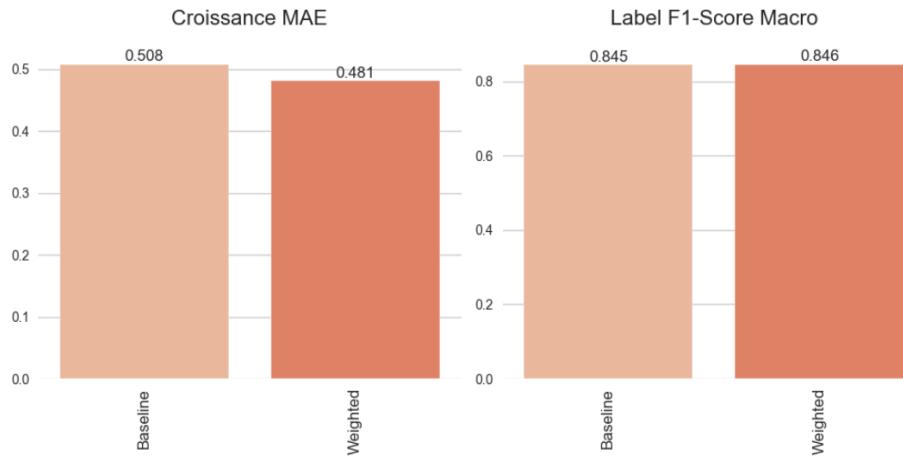


Notre modèle de référence a montré des performances solides, avec un MAE de 0.508 pour la prédiction de la croissance et un F1-score macro de 0.845 pour la classification des plantes.

## Méthode usuelle : Pondération des échantillons lors de l'entraînement

Cette technique est efficace pour gérer les déséquilibres de classe dans les données d'apprentissage. La méthode de l'Inverse du Nombre d'Échantillons (INS) est utilisée pour attribuer un poids plus important aux espèces moins représentées, permettant ainsi au modèle d'apprendre plus efficacement ces classes. Ces poids sont appliqués au générateur de données d'entraînement. Ensuite, le modèle est construit et est entraîné avec ces nouvelles pondérations.

Comparaison des performances sur le dataset de test

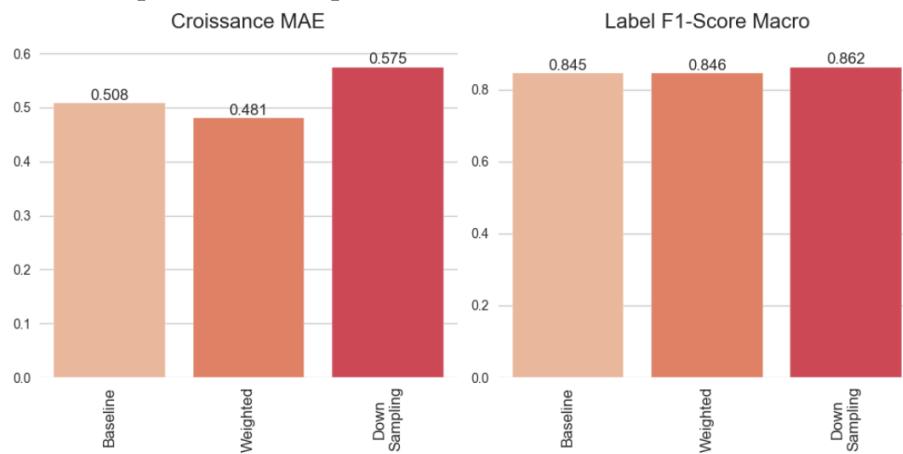


On observe une amélioration de 5.3% de la MAE sur la croissance et une très légère amélioration de 0.1% du F1 score macro pour la prédiction de l'espèce.

## Méthode usuelle : Sous-échantillonage des classes majoritaires

Une autre approche est essayée pour gérer le déséquilibre des classes : le sous-échantillonage. Cette technique réduit le nombre d'échantillons des classes majoritaires pour rééquilibrer leurs distributions. Ici, la moitié des échantillons des trois classes majoritaires (« peanut », « perilla », et « perso-millet ») est conservée. Le modèle est ensuite construit et est entraîné avec ces données rééchantillonées.

Comparaison des performances sur le dataset de test



On observe une dégradation de 13.2% de la MAE sur la croissance (due au plus faible nombre de données d'entraînement) et une amélioration de 2.0% du F1 score macro pour les espèces.

## Méthode implémentée : Large Margin aware Focal Loss (LMFLoss)

### Principe de la LMF Loss

$$L_{LMF} = \alpha L_{LDAM} + \beta L_{FL}$$

Cette fonction de perte combine linéairement la Focal Loss [7] et la LDAMLoss [8] pour atténuer le déséquilibre de la distribution entre les classes dans les tâches de classification. Deux nouveaux paramètres sont introduits ici,  $\alpha$  et  $\beta$ , qui serviront à contrôler l'intensité de l'attention portée aux deux fonctions de pertes.

### Principe de la LDAM Loss

$$L_{LDAM}((x, y), f) = -\log \frac{u}{u + \sum_{j \neq y} e^{z_j}}$$

$$u = e^{z_y - \Delta_y}$$

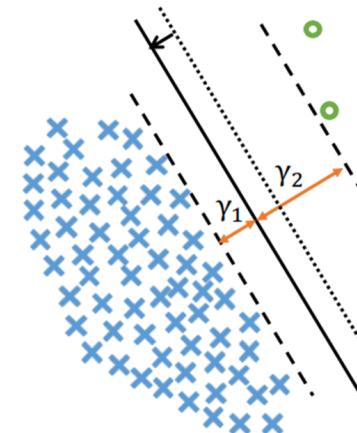
$$z_y = f(x)_y$$

$$\Delta_j = \frac{C}{n_j^{1/4}}$$

$C$  : constante

$n_j$  : nombre d'observations dans la classe

La LDAM Loss augmente la marge entre les scores de la classe majoritaire et les scores des autres classes.



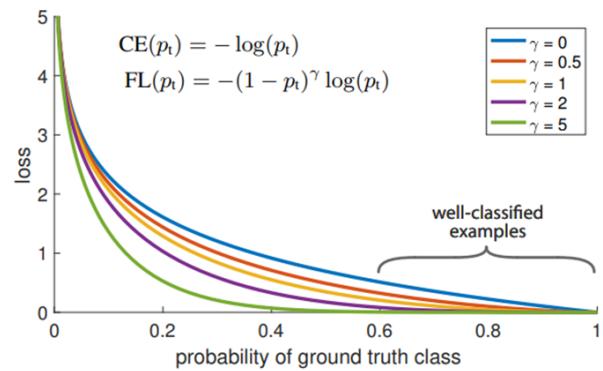
### Principe de la Focal Loss

$$L_{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

$p_t$  : probabilité prédite par le modèle

$\gamma$  : paramètre ajustable

La Focal Loss réduit le poids des échantillons facilement classifiables afin que le modèle se concentre sur les échantillons difficiles à classer.



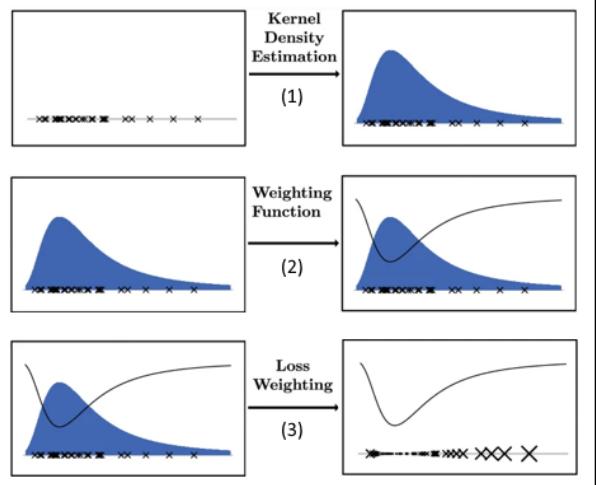
## Méthode implémentée : DenseWeight

Cette méthode, utilisée dans un contexte de régression, modifie les poids des échantillons lors de l'entraînement. Elle attribue une importance accrue aux échantillons sous-représentés, ce qui oriente l'apprentissage du modèle pour qu'il se focalise davantage sur les erreurs commises sur ces groupes. Utilisée en conjonction avec DenseLoss pour calculer l'erreur lors de l'apprentissage, cette technique permet de construire un modèle plus équilibré.

### Principe de DenseWeight

1. Estimation par noyau de la densité (KDE) pour approximer la fonction de densité des valeurs cibles d'entraînement
2. Utilisation de la fonction de densité résultante pour calculer la fonction de pondération de DenseWeight  

$$f'_w(\alpha, y) = 1 - \alpha p'(y)$$
3. Attribution à chaque point dans l'ensemble d'entraînement un poids augmentant ainsi l'influence des points rares sur la perte

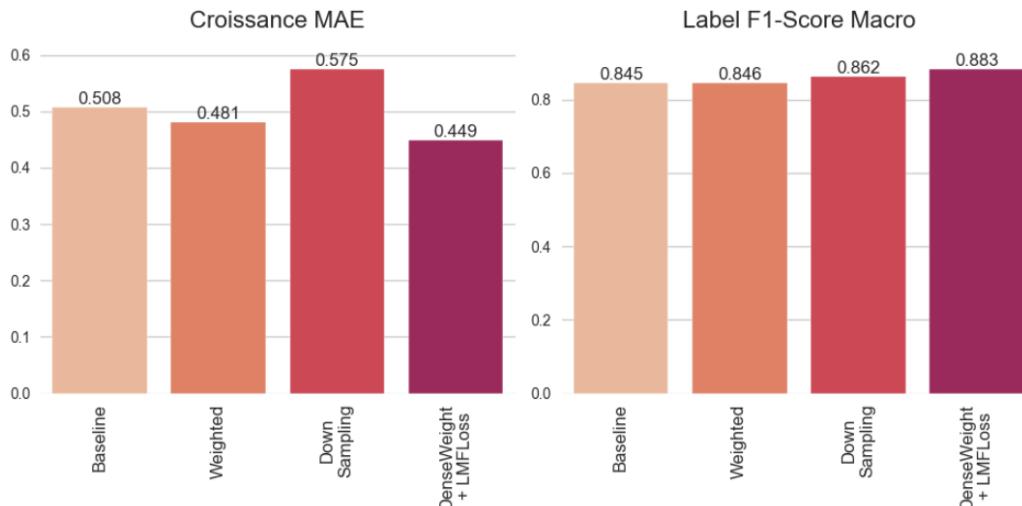


L'hyperparamètre  $\alpha$  permet de configurer l'ajustement de l'attention du modèle.

## Implémentation de DenseWeight & LMF Loss

Ces méthodes ont été appliquées à notre modèle grâce à l'utilisation du package DenseWeight, par la création d'une classe DenseLoss et d'une classe LMF Loss, ainsi que par la recherche des meilleurs hyperparamètres pour obtenir une performance optimale.

### Comparaison des performances sur le dataset de test



Grâce à l'application de LMF Loss et DenseWeight, on constate une amélioration notable de la MAE sur la croissance de 11.6% et une progression de 4.5% du F1 score macro pour les espèces.

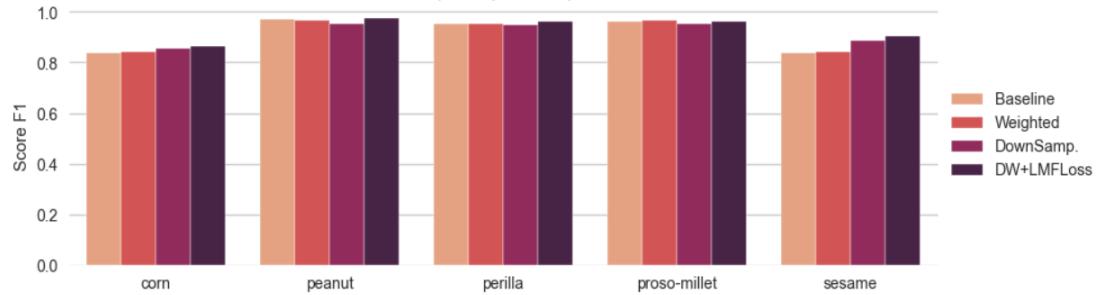
## Analyse des résultats

- Prédiction de l'espèce

### Comparaison des matrices de confusion pour la classification des plantes

	Baseline					Weighted					DownSampling					DenseWeight + LMFLoss				
	corn	peanut	peperilla	proso-millet	sesame	corn	peanut	peperilla	proso-millet	sesame	corn	peanut	peperilla	proso-millet	sesame	corn	peanut	peperilla	proso-millet	sesame
corn	57.5%	8.0%	2.3%	31.0%	1.1%	56.2%	9.0%	2.2%	32.6%	0.0%	77.9%	5.8%	1.2%	12.8%	2.3%	66.3%	3.4%	2.2%	28.1%	0.0%
peanut	0.3%	98.2%	1.0%	0.5%	0.0%	0.3%	98.4%	1.0%	0.3%	0.0%	0.3%	99.2%	0.5%	0.0%	0.0%	0.5%	97.9%	1.1%	0.5%	0.0%
peperilla	0.6%	3.2%	95.2%	0.6%	0.4%	0.4%	4.5%	93.4%	0.6%	1.1%	0.8%	7.8%	89.8%	0.4%	1.3%	0.8%	2.3%	94.9%	1.7%	0.4%
proso-millet	0.6%	1.4%	1.6%	96.3%	0.0%	0.0%	1.2%	1.2%	97.5%	0.0%	5.1%	3.9%	0.8%	89.8%	0.4%	0.8%	1.4%	0.8%	96.9%	0.0%
sesame	0.0%	2.7%	39.7%	2.7%	54.8%	1.4%	5.5%	31.5%	4.1%	57.5%	1.4%	6.8%	16.2%	1.4%	74.3%	0.0%	4.1%	20.3%	5.4%	70.3%

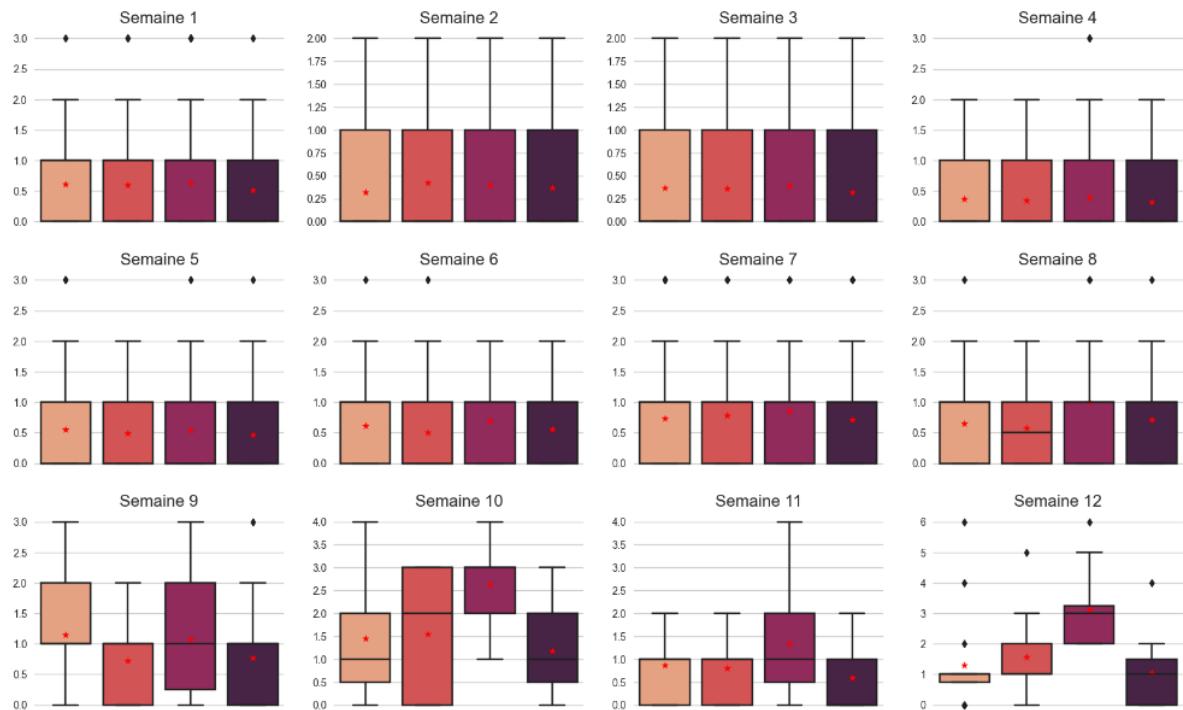
### Comparaison des F1 Scores Macro



La méthode implémentée a démontré une performance remarquable, réussissant à améliorer les F1 scores de toutes les espèces, avec une amélioration encore plus marquée pour la classe sous-représentée « *sesame* ».

- Prédiction des données de croissance

### MAE des différents modèles pour chaque semaine de croissance



Les valeurs de la MAE pour les semaines 1 à 8 restent relativement similaires, bien qu'on observe une légère baisse de l'erreur moyenne grâce à notre implémentation. Toutefois, l'amélioration devient beaucoup plus perceptible entre les semaines 9 et 12, témoignant de l'efficacité de notre méthode sur cette période, qui correspond à celle pour laquelle nous avions le moins d'observations.

## Conclusions

Ce projet a illustré l'importance d'une gestion adaptée du déséquilibre des données dans le contexte de la classification d'images de plantes et de l'analyse de leur croissance. Les améliorations significatives que nous avons constatées dans la précision de nos prédictions témoignent de l'efficacité des méthodes implémentées - LMFLoss et Denseweight. Au-delà de la progression notable des performances de notre modèle, ces résultats mettent en lumière la capacité de ces techniques à gérer le défi inhérent à la nature déséquilibrée de nombreuses données de terrain. Ce projet souligne l'importance de continuer à affiner ces techniques et de les adapter à une variété de problématiques similaires dans le domaine de l'agriculture de précision, mais aussi au-delà.

## Bibliographie

- [1] K. A. Y. C. J. O. W. J. H. L. H. K. Talha Ilyas Dewa Made Sri Arsa, “Cwd30: a comprehensive and holistic dataset for crop weed recognition in precision agriculture,” 2023. [Online]. Available: <https://arxiv.org/pdf/2305.10084.pdf>
- [2] A. Howard, M. Sandler, et al., “Searching for mobilenetv3,” 2019. [Online]. Available: <https://arxiv.org/pdf/1905.02244.pdf>
- [3] P. D. Michael Steininger Konstantin Kobs, “Density-based weighting for imbalanced regression,” 2021. [Online]. Available: <https://link.springer.com/article/10.1007/s10994-021-06023-5>
- [4] A. A. Sadi, L. Chowdhury, et al., “Lmfloss: a hybrid loss for imbalanced medical image classification,” 2022. [Online]. Available: <https://arxiv.org/pdf/2212.12741.pdf>
- [5] X. Long, H. Zhang, and T. Zhang, “Label-imbalanced and group-sensitive classification under overparameterization,” 2021. [Online]. Available: <https://arxiv.org/abs/2103.01550>
- [6] Y. Yang, K. Zha, Y.-C. Chen, H. Wang, and D. Katabi, “Delving into deep imbalanced regression,” 2021. [Online]. Available: <https://arxiv.org/abs/2102.09554>
- [7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2018. [Online]. Available: <https://arxiv.org/pdf/1708.02002.pdf>
- [8] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.07413>