

# Développez une preuve de concept

## Objectif

- Identifier et implémenter un algorithme SOTA pour améliorer les performances d'une méthode en production existante

**Problématique :** Gestion du déséquilibre des classes

**Optimisation de la classification des plantes**

# Présentation du jeu de données CWD30

Ensemble de données de 219770 images de plantes

Sélection de 5 espèces (~17000 images) :

- **maïs (corn)**
- **arachide (peanut)**
- **périlla (perilla)**
- **millet proso (proso-millet)**
- **sésame (sesame)**

Exemples d'images des plantes sélectionnées



# Présentation du jeu de données CWD30

## Données de croissance : 1 à 12 semaines

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

Semaine : 1



Semaine : 2



Semaine : 3



Semaine : 4



Semaine : 5



Semaine : 6



Semaine : 7



Semaine : 8



Semaine : 9



Semaine : 10



Semaine : 11



Semaine : 12



## Différents angles : 0°, 45°, 90°

Exemples d'images avec les 3 angles de prise de vue

Angle : 0



Angle : 0



Angle : 45



Angle : 45



Angle : 90



Angle : 90

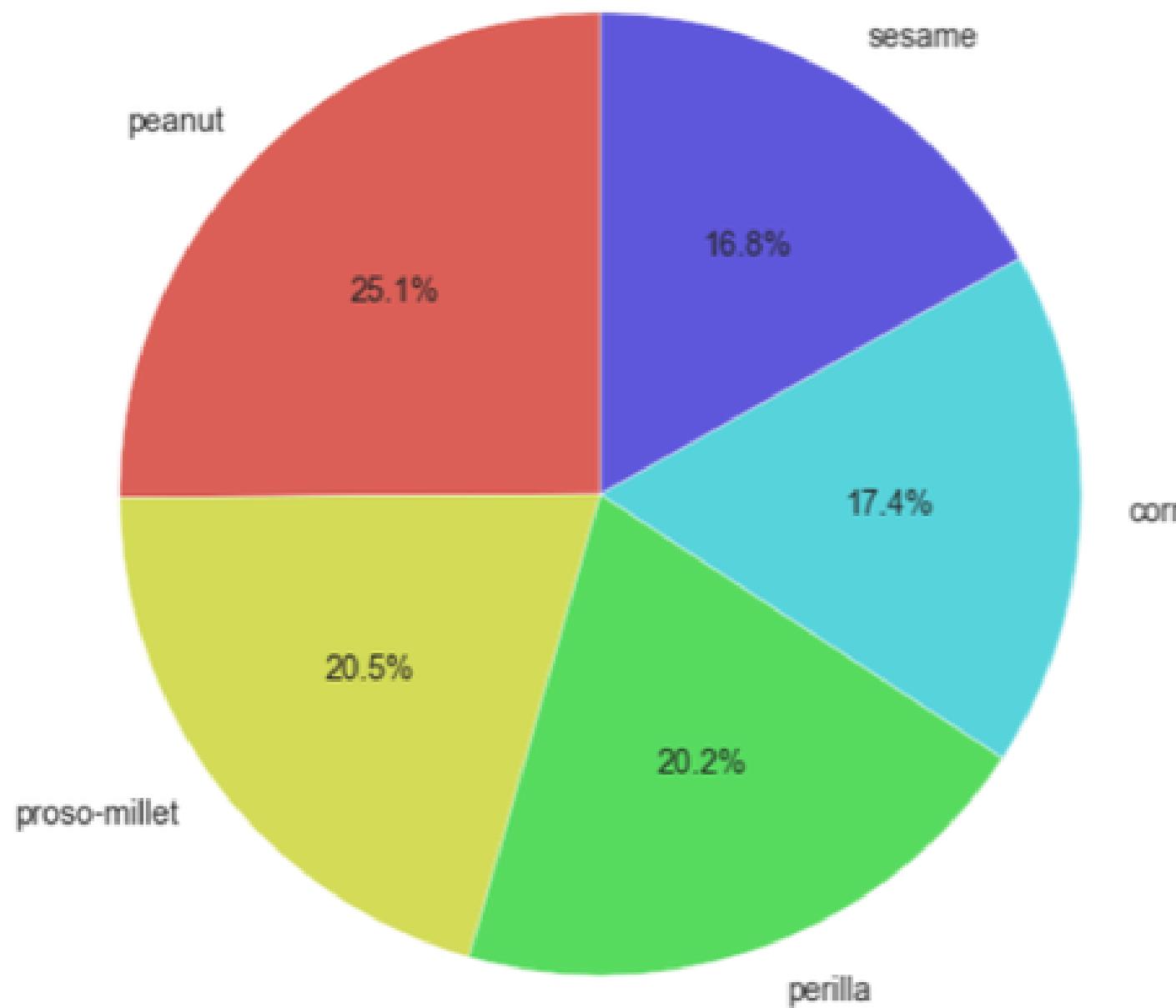




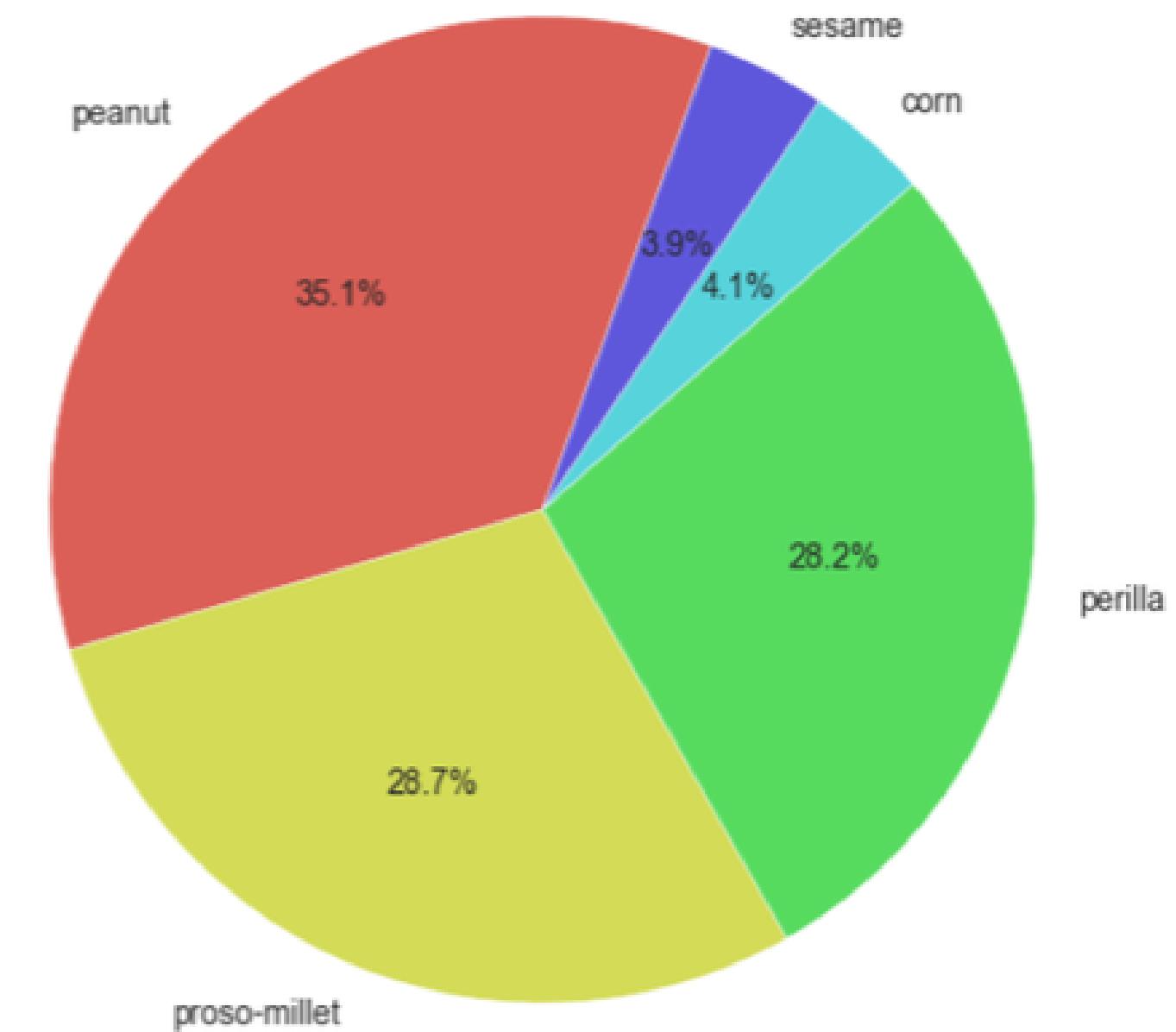
# Prétraitement des données

# Prétraitement des données

Distribution originale

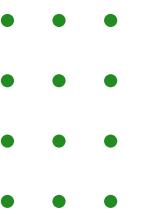


Distribution implémentée



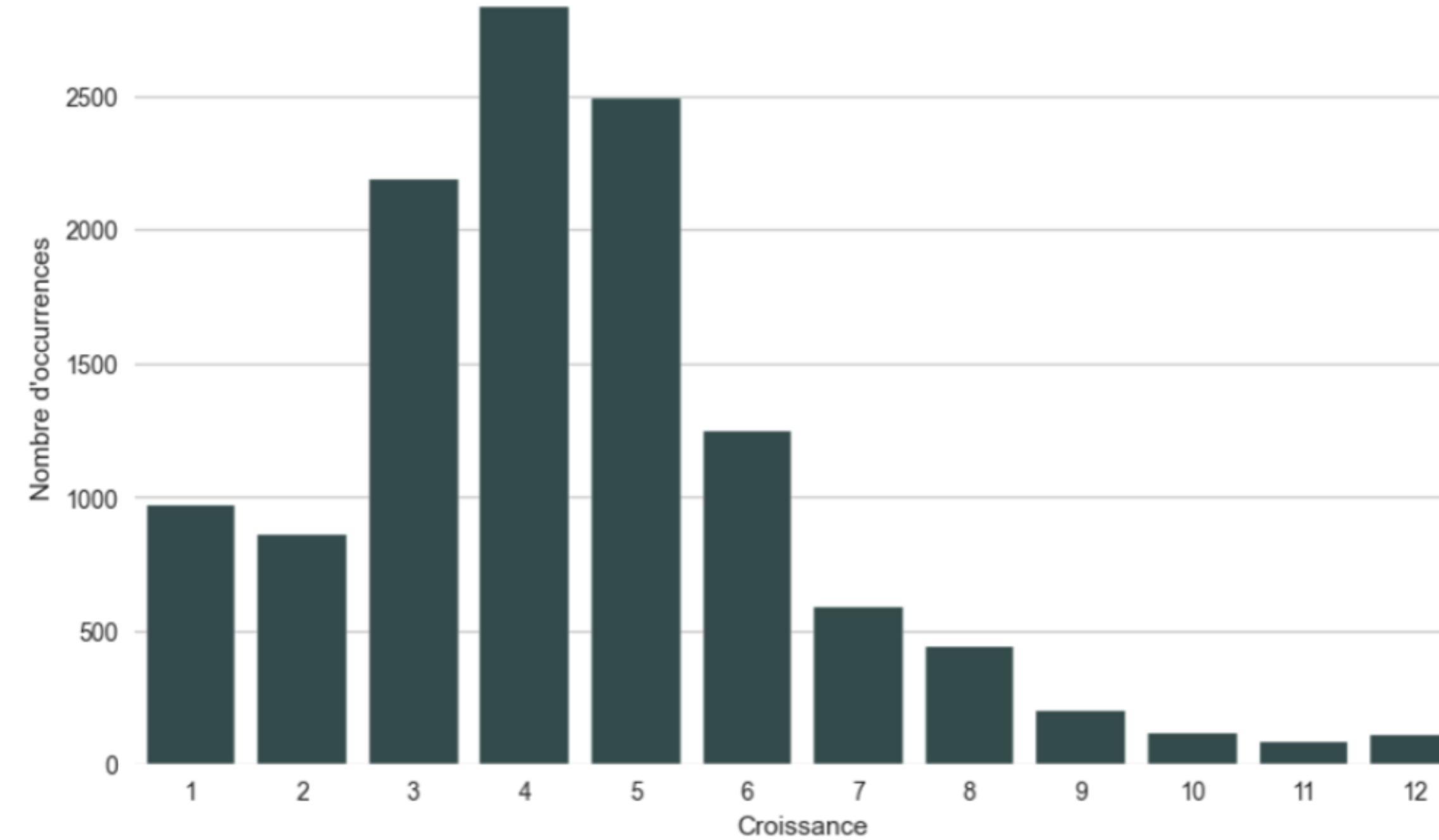
Déséquilibre implémenté :

- suppression de 5/6ème des images des classes sésame et maïs

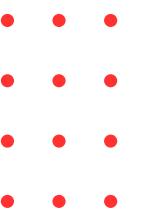


# Prétraitement des données

Distribution des données de croissance des plantes



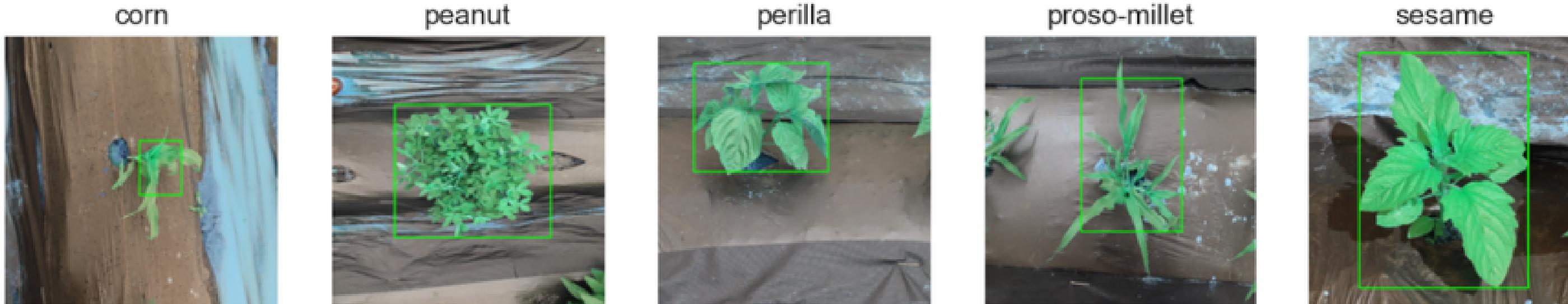
**Sur-représentation de plantes en croissance de 3 à 5 semaines**



# Préparation des données

**Création de boîtes d'encadrement afin de segmenter uniquement la plante**

Exemples d'images avec sélection des contours des plantes



**Encodage des labels (noms des espèces) à l'aide de LabelEncoder**

**Séparation des 12132 images restantes en trois jeux de données**

- **70% pour l'entraînement (8492)**
- **15% pour la validation (1820)**
- **15% pour le test (1820)**

# Préparation des données

## Utilisation d'un `ImageDataGenerator` pour augmenter le jeu d'entraînement

### Utilisation du générateur `flow_from_dataframe`

- **traitement des 3 ensembles préparés séparément**
- **redimensionnement des images à 224x224**
- **chargement par lots de 32 images**
- **sorties multiples pour le label et la croissance**

### Exemples d'images recadrées avec data augmentation



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



# Présentation du modèle de référence

# Modèle de base

## Utilisation du modèle MobileNetV3Small



**Poids 'ImageNet' avec dégel de l'ensemble des couches**

**Tenseur d'entrée : (224, 224, 3)**

**Version minimaliste**

**Ajout de couches denses (activation 'relu') et de régularisation par dropout (hp)**

**Ajout de 2 couches en sortie du modèle :**

- label : 5 sorties avec une activation 'softmax'**
- croissance : 1 sortie avec une activation 'relu'**

## Compilation du modèle

**Optimiseur : Adam, le taux d'apprentissage est un hyperparamètre à régler**

Paramètre	Espèce (label)	Croissance (growth)
Perte	sparse_categorical_crossentropy	MAE
Métriques	F1-Score Macro	MAE
Objectif (KT)	Max	Min

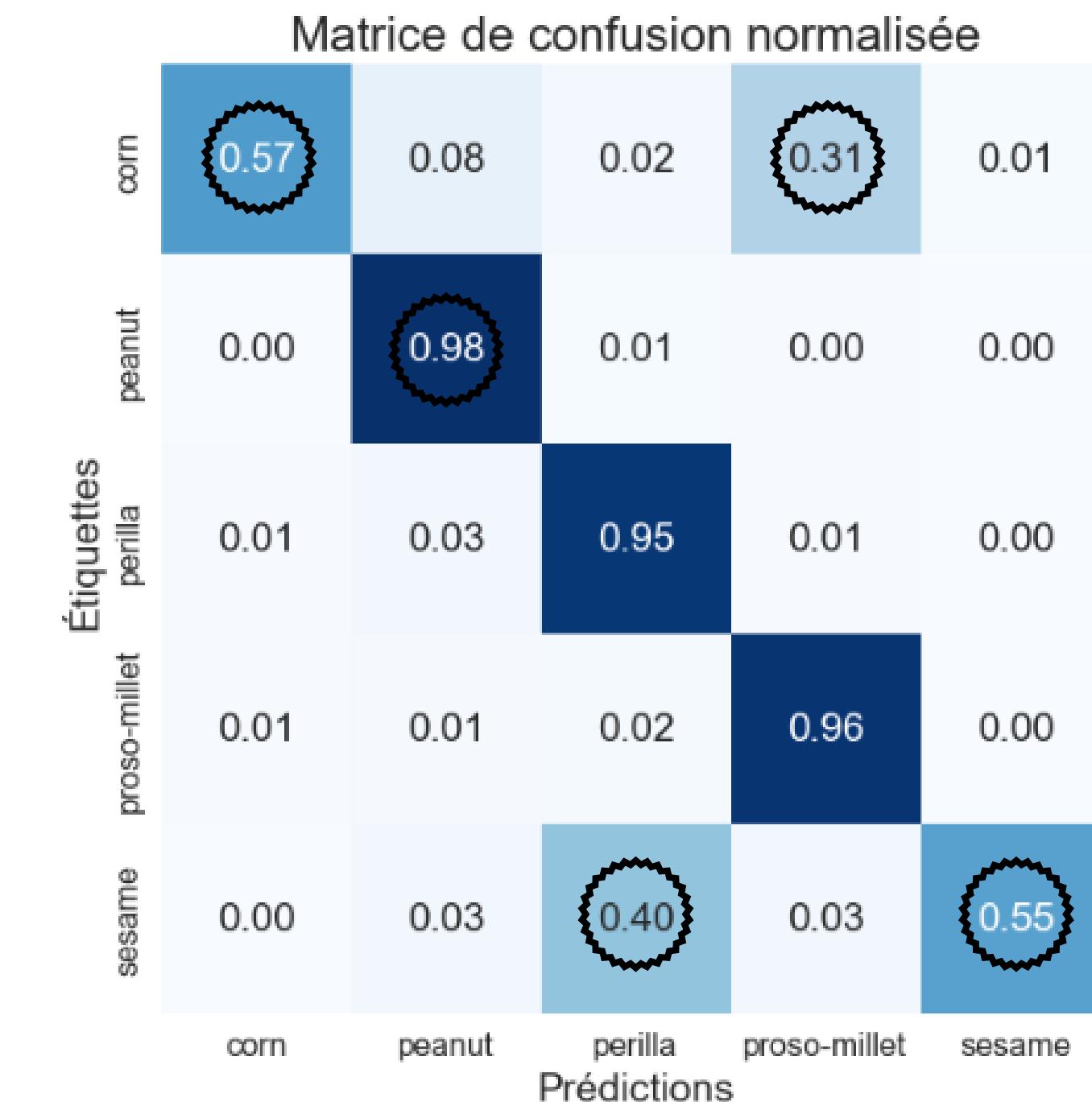
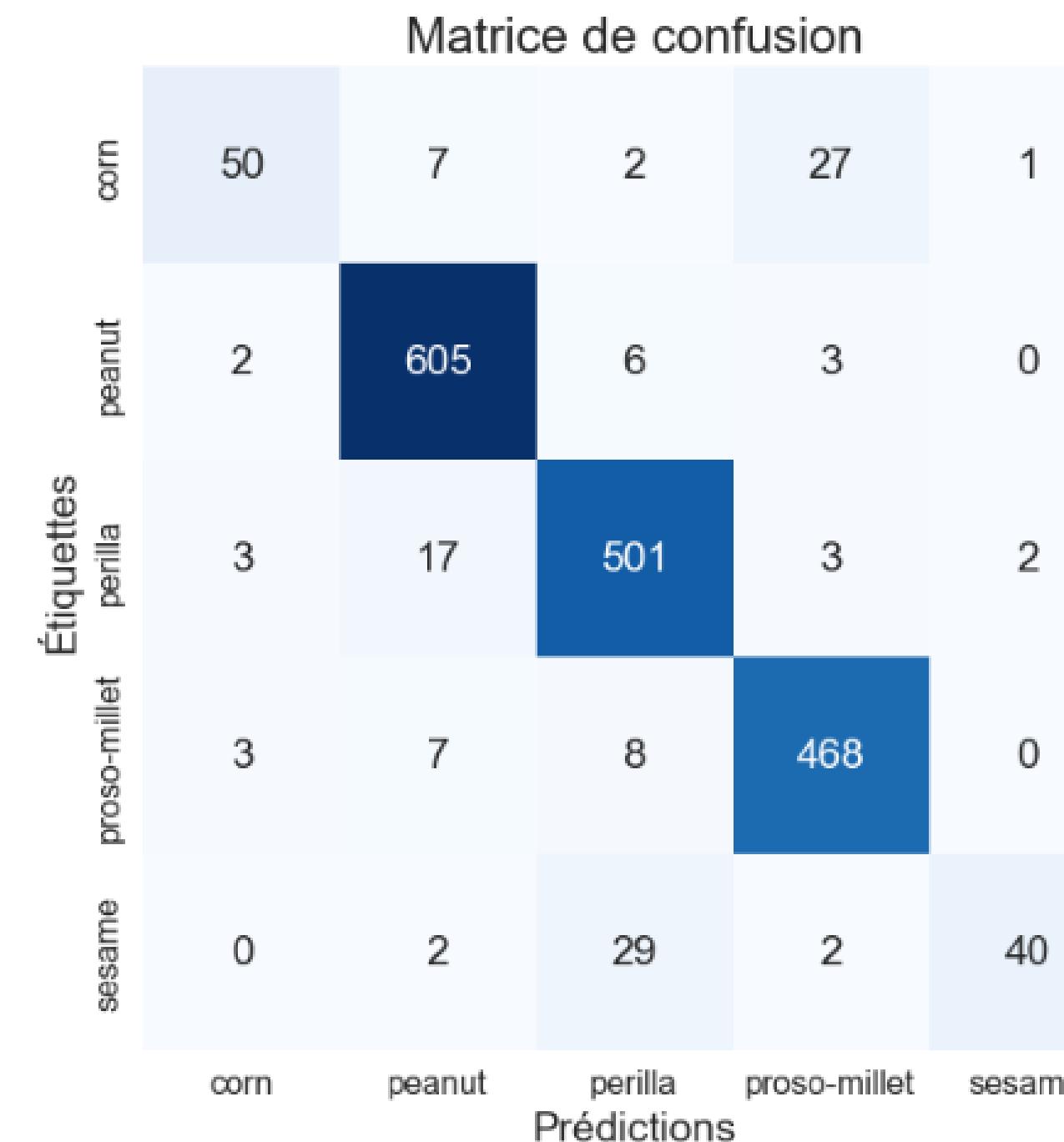
**Création d'un rappel (EarlyStopping) sur la perte de la validation**

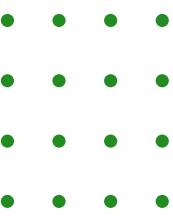
**Utilisation de la méthode BayesianOptimization de KerasTuner**

- **Points initiaux : 5**
- **Nombre d'essais : 25**
- **Nombre d'epochs : 20**

# Performances sur le dataset de test

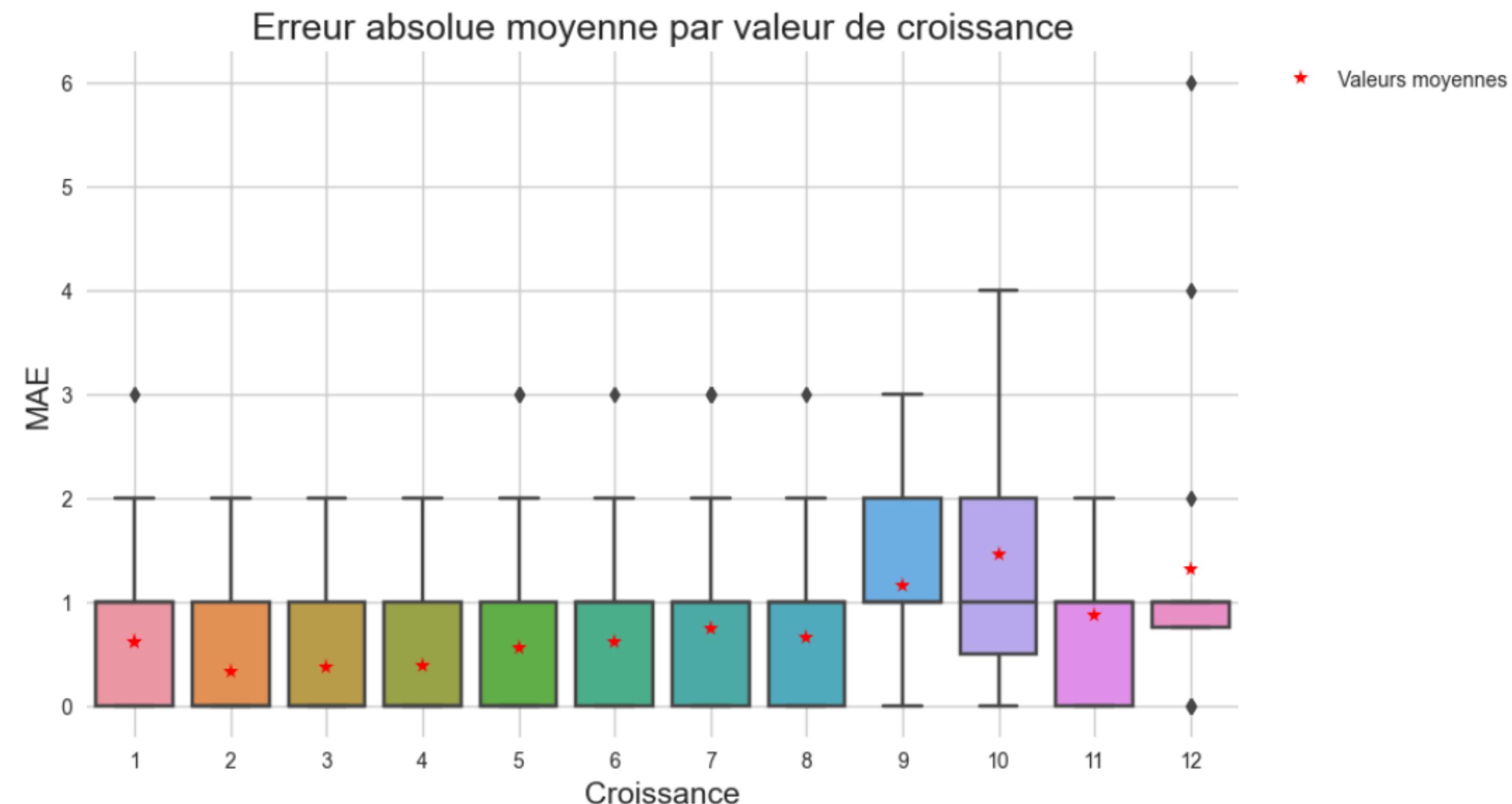
**Label F1-Score Macro : 0.845**





# Performances sur le dataset de test

**Growth MAE : 0.508**



# Performances sur le dataset de test

## Exemples de prédictions avec le modèle de référence



Pred: Label: perilla  
Growth: 12



Pred: Label: perilla  
Growth: 1



Pred: Label: peanut  
Growth: 4



Pred: Label: proso-millet  
Growth: 6



Pred: Label: corn  
Growth: 3



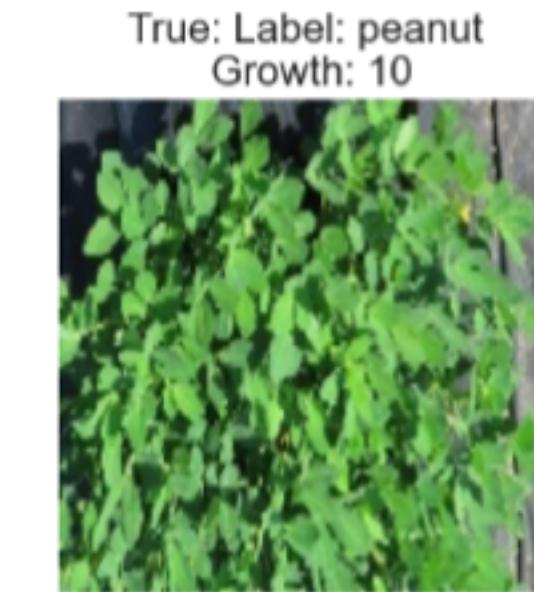
Pred: Label: peanut  
Growth: 3



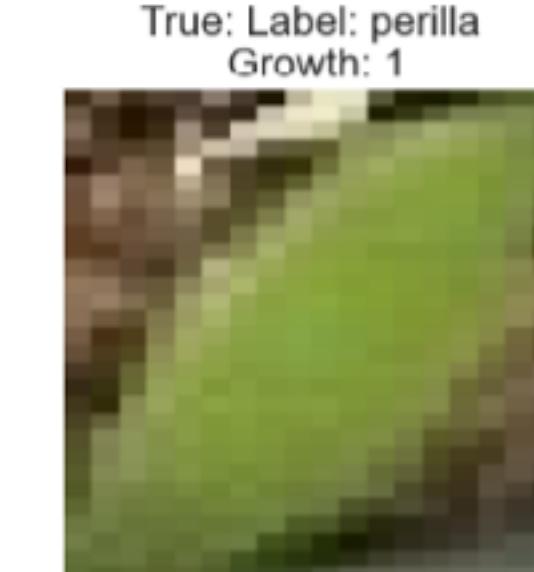
Pred: Label: proso-millet  
Growth: 2



Pred: Label: peanut  
Growth: 3



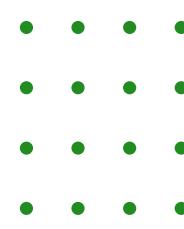
Pred: Label: peanut  
Growth: 11



Pred: Label: peanut  
Growth: 1



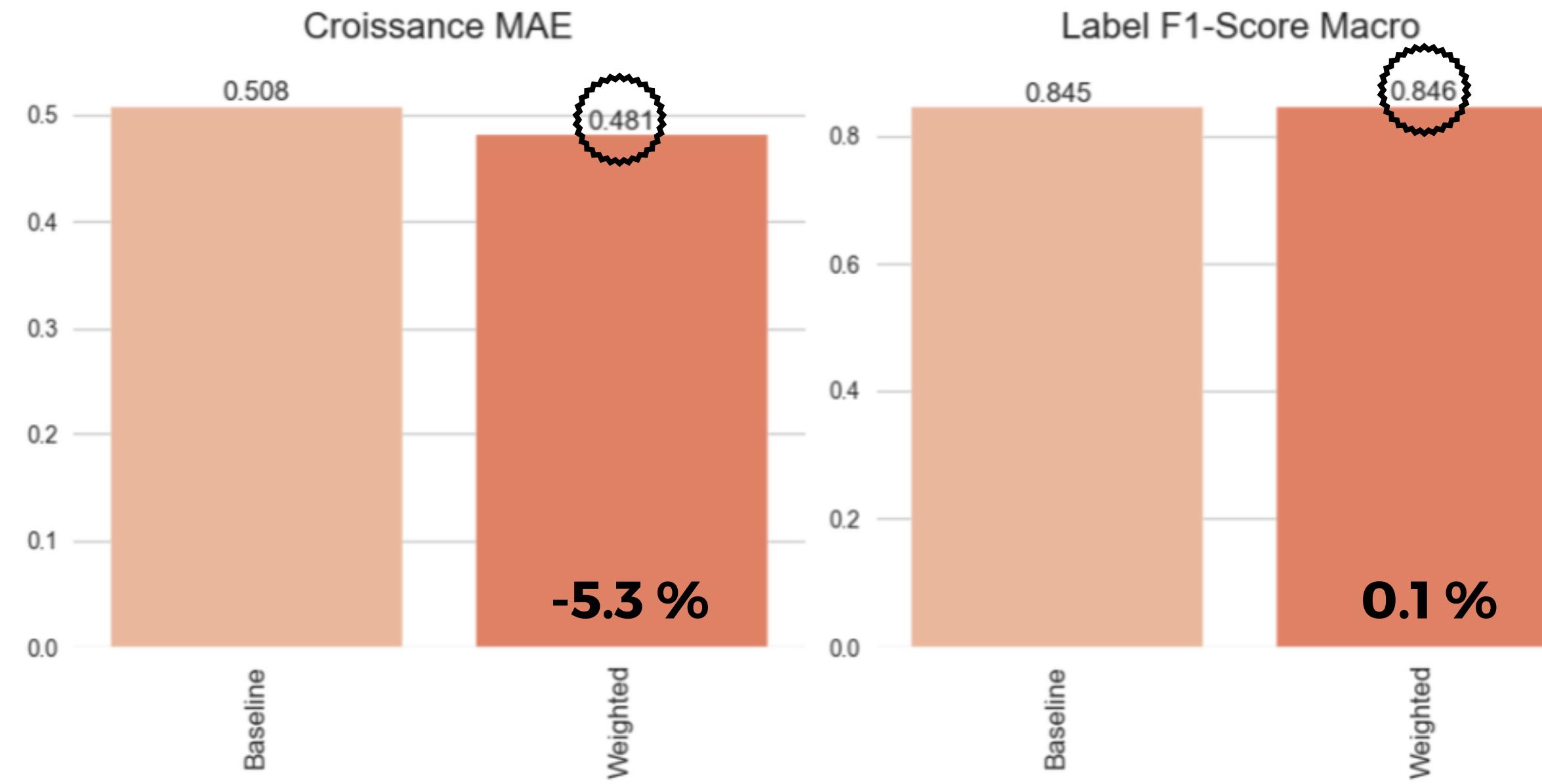
# Présentation des méthodes usuelles



# Pondération des échantillons

**Utilisation de la méthode Inverse du Nombre d'Échantillons (INS)**  
**Application des poids INS au générateur de données d'entraînement**

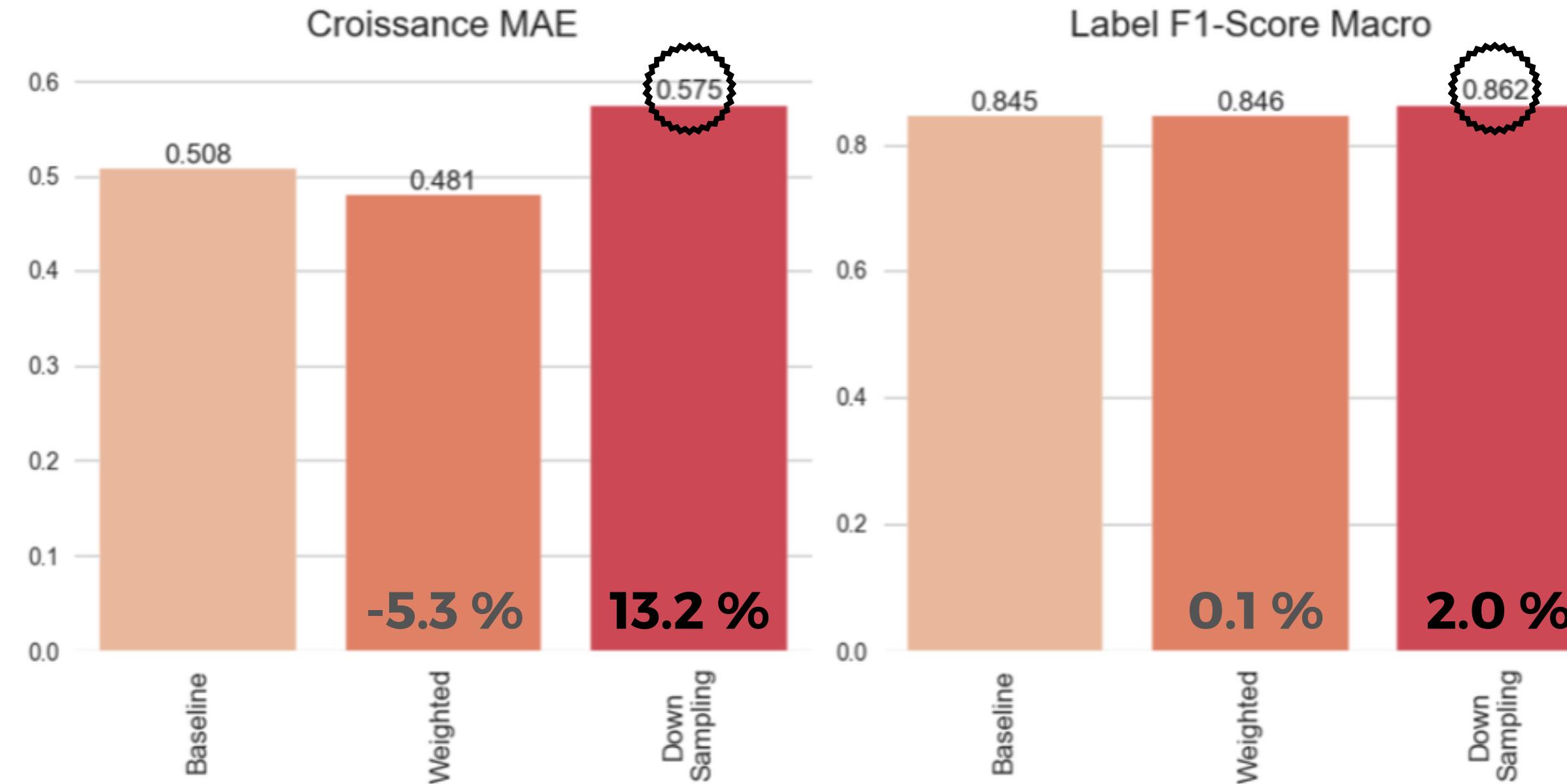
Comparaison des performances sur le dataset de test



# Sous-échantillonage des classes majoritaires

**Conservation de la moitié des échantillons des trois classes majoritaires**  
**Entraînement du modèle avec les données rééchantillonnées**

Comparaison des performances sur le dataset de test



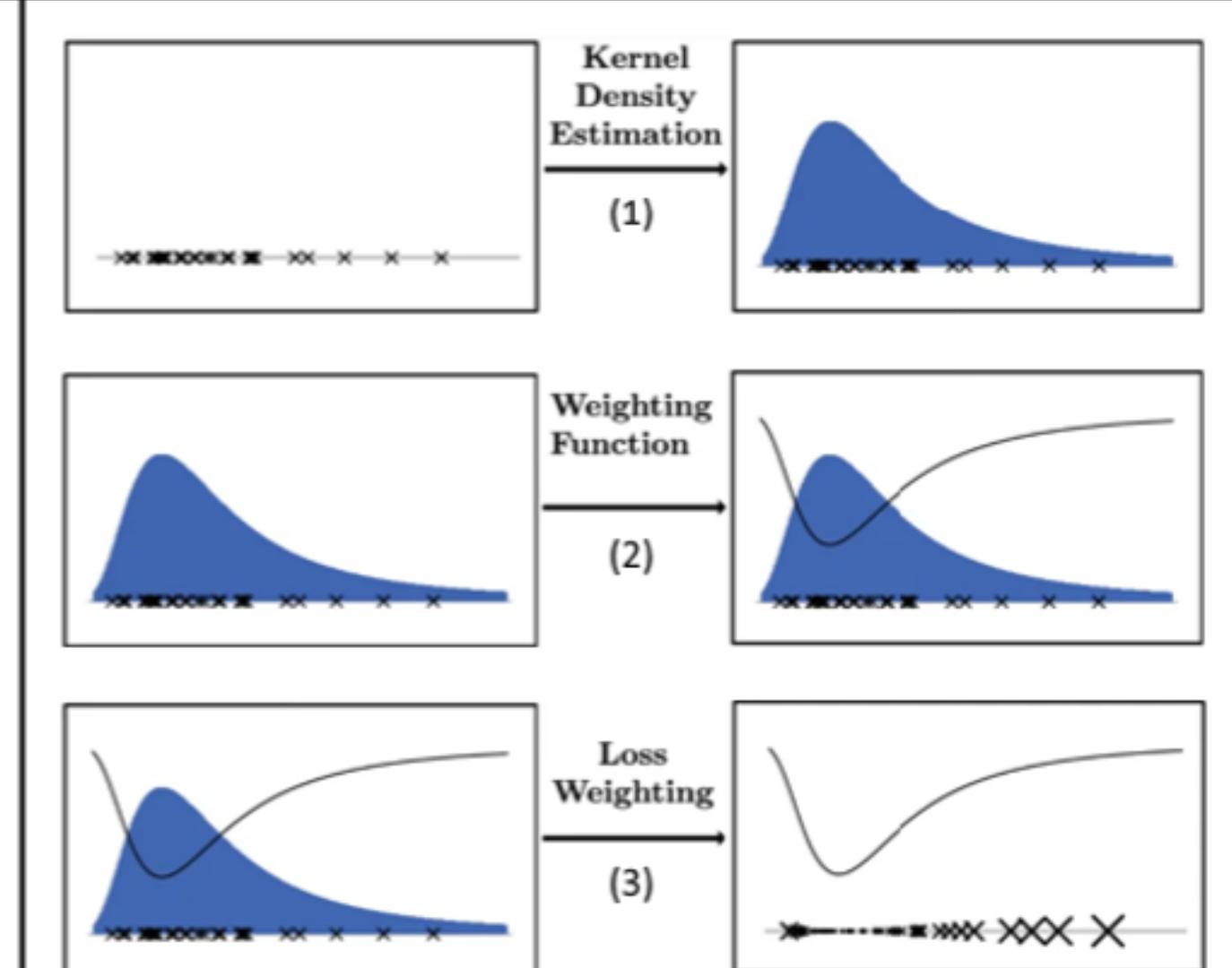


# Présentation des méthodes SOTA

# Présentation de DenseWeight

**Ajustement des poids et de la fonction de perte en contexte de régression  
Importance accrue accordée aux groupes sous-représentés**

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



**Hyperparamètre  $\alpha$  pour ajuster l'attention du modèle**

# Présentation de LMF Loss

## Combinaison linéaire de LDAM Loss et Focal Loss

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

### Principe de la LDAM Loss

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

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

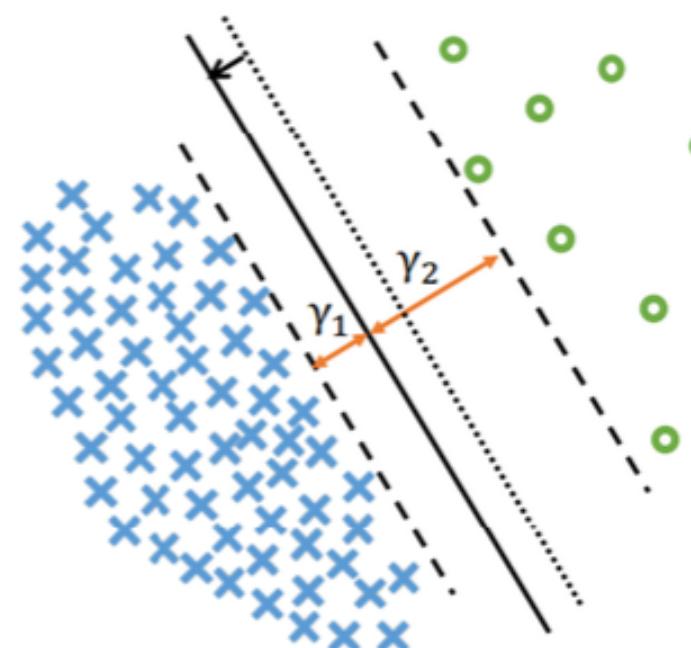
$$z_y = f(x)_y$$

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

$C$  : 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.



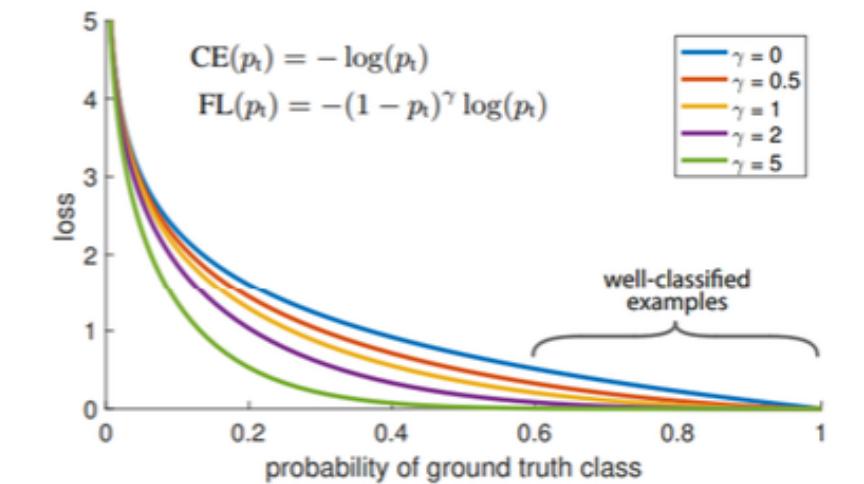
### Principe de la Focal Loss

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

$p_t$  : 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.



## Priorisation des classifications difficiles

Durcissement pour les classes minoritaires

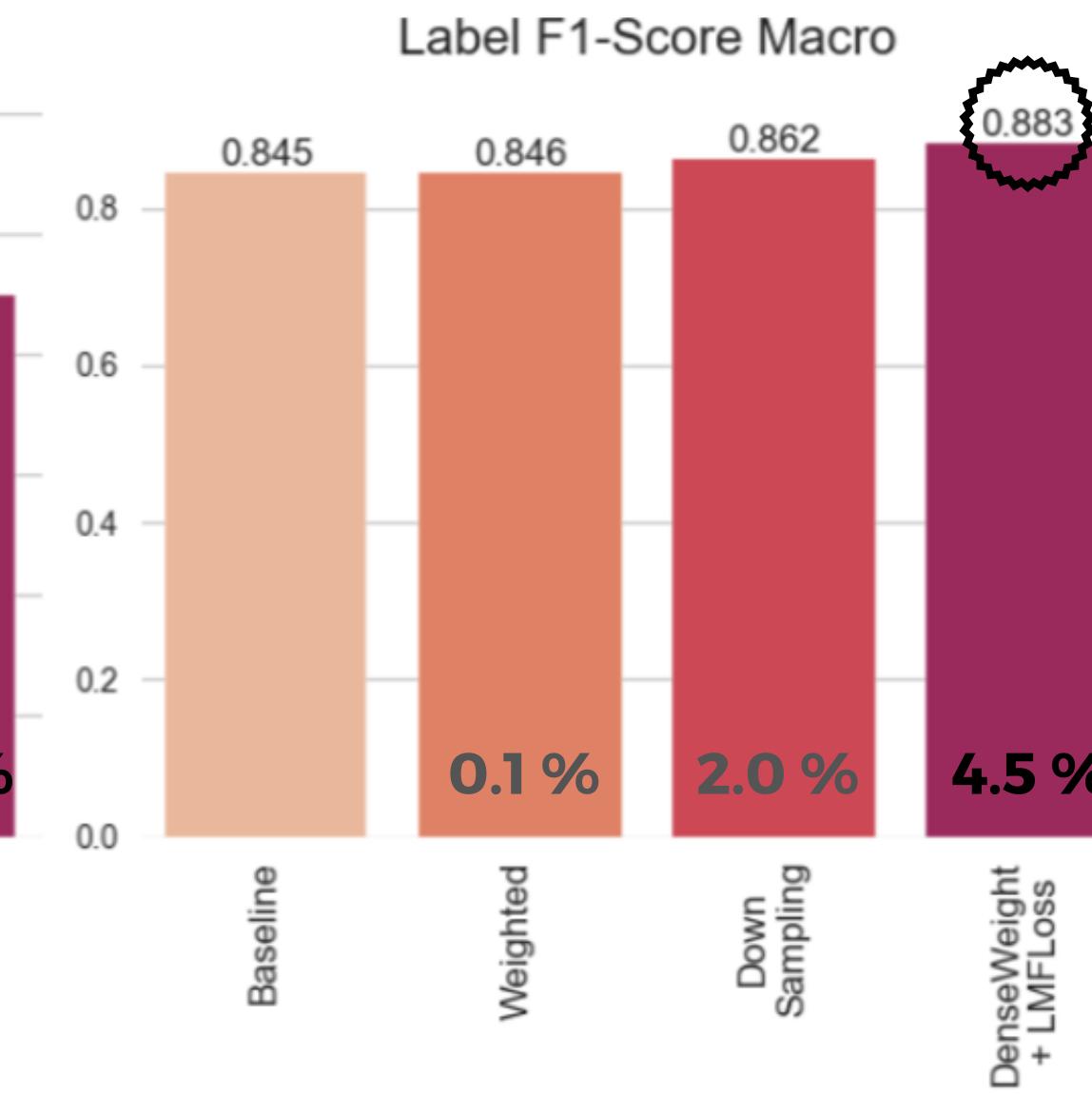
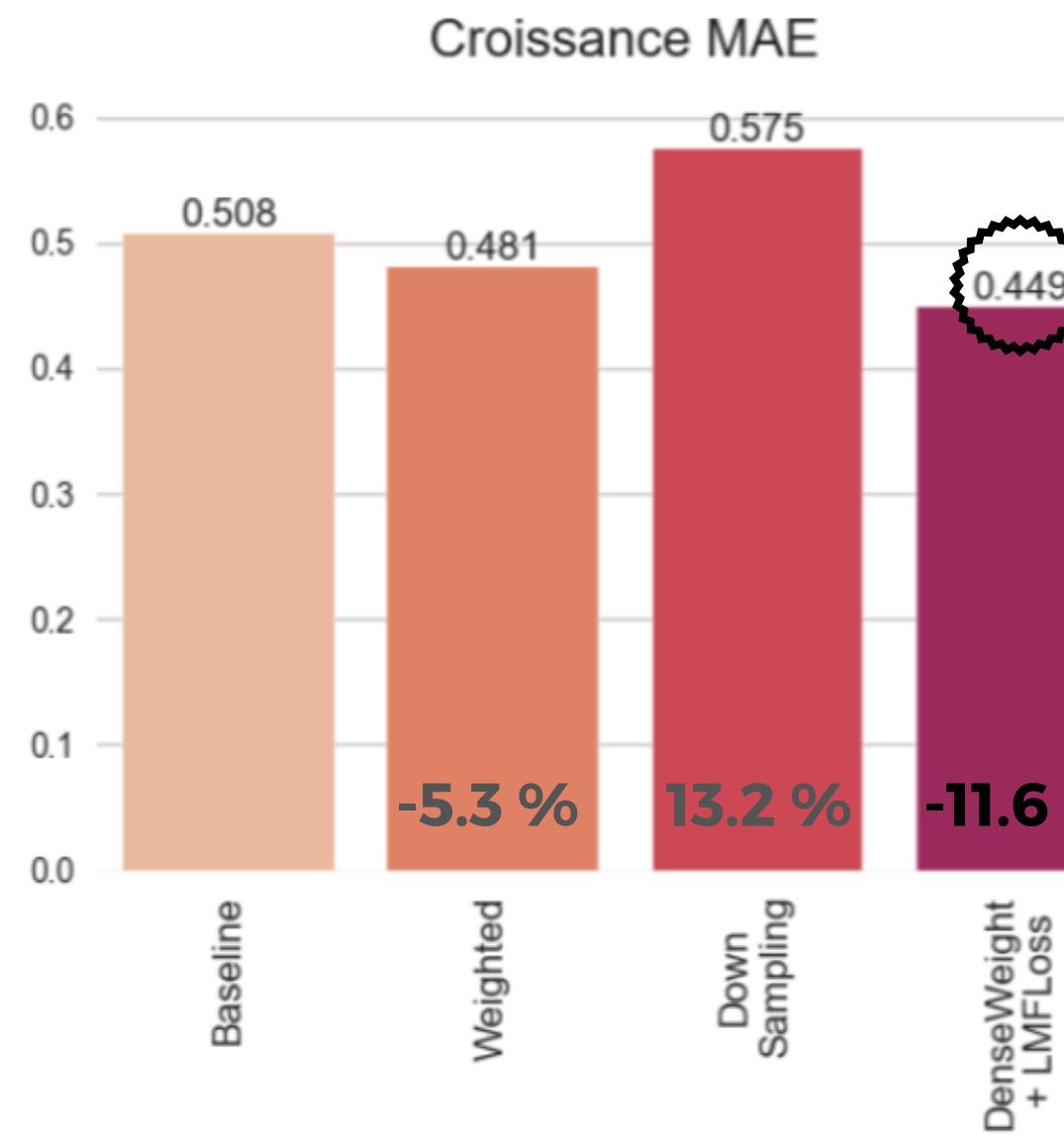
# Implémentation de DenseWeight & LMF Loss

DenseWeight:  $\alpha = 0.25$

DenseLoss :  $\alpha = 0.25$

LMF Loss :  $\alpha = 2.5$ ,  $\beta = 0.05$  et  $\gamma = 4.0$

## Comparaison des performances sur le dataset de test





# Présentation des résultats

# Prédiction de l'espèce

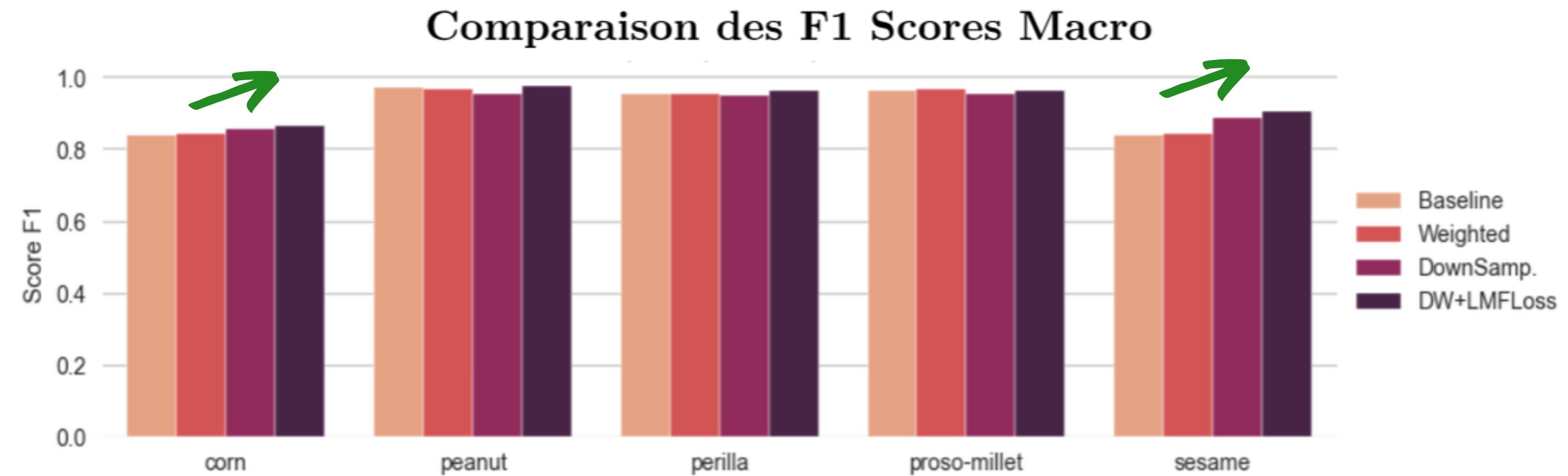
## Comparaison des matrices de confusion pour la classification des plantes

	Baseline					Weighted					DownSampling					DenseWeight + LMFLoss				
	corn	peanut	perilla	proso-millet	sesame	corn	peanut	perilla	proso-millet	sesame	corn	peanut	perilla	proso-millet	sesame	corn	peanut	perilla	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%
perilla	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%

Permet d'obtenir des performances satisfaisantes pour le maïs et le sésame

Meilleurs performances sur les autres espèces par rapport au DownSampling

# Prédiction de l'espèce



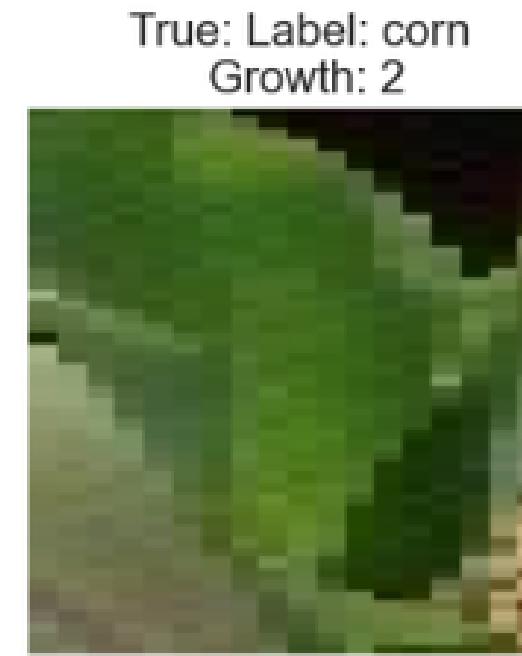
Amélioration des F1-Scores pour toutes les espèces

# Analyse qualitative des mauvaises performances

## Exemples d'erreurs de classification pour le maïs



Pred: Label: proso-millet  
Growth: 4



Pred: Label: proso-millet  
Growth: 3



Pred: Label: proso-millet  
Growth: 3



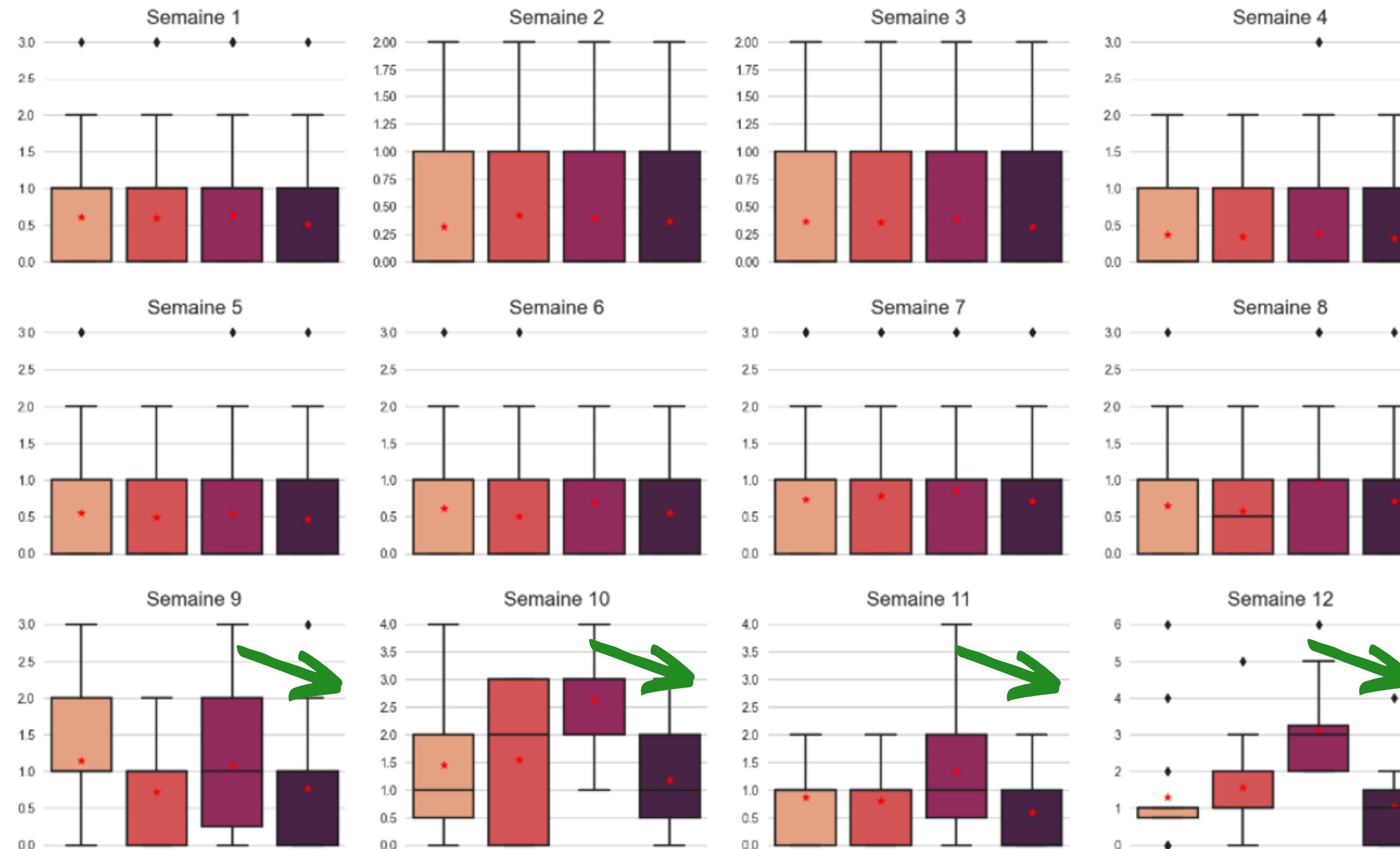
Pred: Label: proso-millet  
Growth: 5



Pred: Label: proso-millet  
Growth: 7

# Prédiction du stade de croissance

MAE des différents modèles pour chaque semaine de croissance



Amélioration notable des prédictions pour les tranches d' extrêmes

# Analyse qualitative des mauvaises performances

## Exemples d'erreurs dans les prédictions de croissance

True: Label: proso-millet  
Growth: 10



Pred: Label: proso-millet  
Growth: 7



True: Label: proso-millet  
Growth: 10



Pred: Label: proso-millet  
Growth: 10

True: Label: proso-millet  
Growth: 5



Pred: Label: proso-millet  
Growth: 8



True: Label: proso-millet  
Growth: 5



Pred: Label: proso-millet  
Growth: 5

True: Label: proso-millet  
Growth: 1



Pred: Label: peanut  
Growth: 4



True: Label: proso-millet  
Growth: 1



Pred: Label: proso-millet  
Growth: 1

True: Label: peanut  
Growth: 7



Pred: Label: peanut  
Growth: 4



True: Label: peanut  
Growth: 7



Pred: Label: peanut  
Growth: 7

True: Label: peanut  
Growth: 12



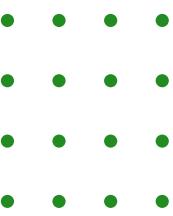
Pred: Label: peanut  
Growth: 8



True: Label: peanut  
Growth: 12



Pred: Label: peanut  
Growth: 12



# Rappel de l'objectif

- Identifier et implémenter un algorithme SOTA pour améliorer les performances d'une méthode en production existante

# Conclusion

- Modèle optimisé offrant une amélioration notable, autant pour la classification des plantes que pour la prédiction des stades de croissance, par rapport à MobileNetV3
- Importance de continuer à affiner ces techniques et de les adapter à une variété de problématiques dans le domaine de l'agriculture de précision