

# Projet Machine Learning

---

Erwan Gouhier, Sixtine Delage de Luget et Emilie Fraysse

# SOMMAIRE

- \* Introduction
- \* Matériel et méthodes
- \* Résultats
- \* Discussion
- \* Conclusion



# INTRODUCTION

## Présentation des données

- \* Données issues du jeu de données PlantDoc (Singh et al., 2020)
- \* Jeu de données constitué de 2598 images réparties dans 27 classes (17-10).
- \* Jeu de données constitué d'images récupérées sur internet et annotées à la main.
- \* Jeu de données constitué dans le but d'identifier les maladies qui impactent les cultures en Inde.

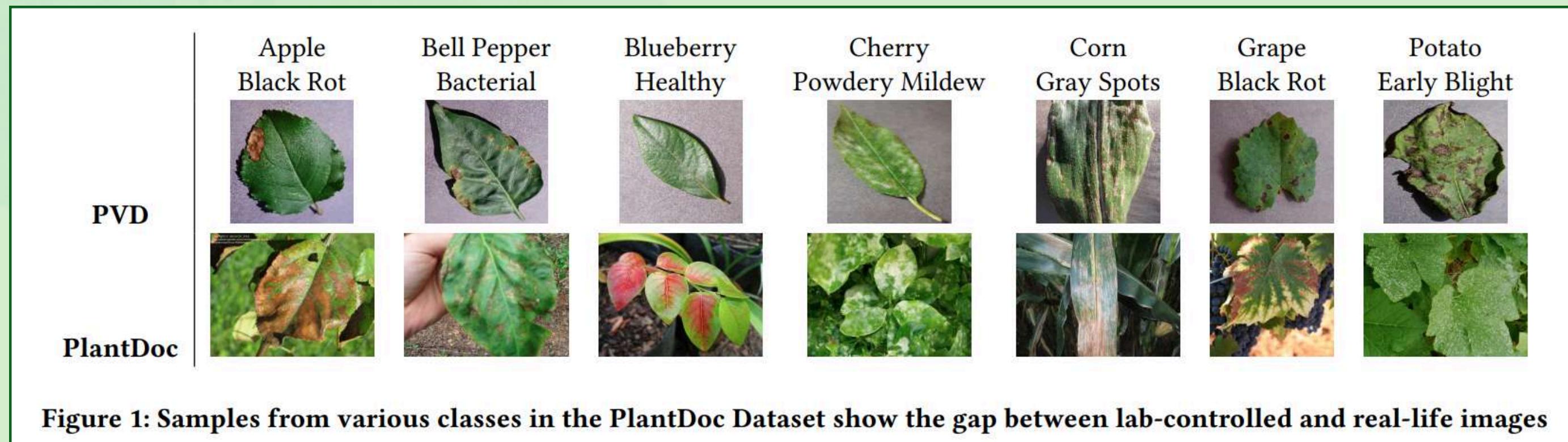


Figure d'après Singh et al., 2020

# INTRODUCTION

## Problématique

- \* Jeu de données volontairement constitué d'images de faible qualité
- \* Entraîner un modèle sur PlantDoc pour classifier les images :
  - Selon si elles sont malades ou pas (rouille, black rot, ...)
  - Selon l'espèce
  - Selon l'espèce et la maladie

Entraîner un modèle ViT  
en Transfer Learning

# INTRODUCTION

## Problématique

### \* Intérêt de choisir un Vit :

- Permet de prendre en compte la position de l'objet d'intérêt dans l'image
- Meilleure reconnaissance de la maladie lorsque les signes sont dispersés sur la feuille (taches, bordures, veinures)
- Moins sensible aux textures parasites
- Reconnaît mieux une classe même si les tâches de la maladie ne sont pas exactement au même endroit sur la feuille
- Plus interprétable



# MATÉRIEL ET MÉTHODES

## Préprocessing des données

- \* Nettoyer les noms des images
- \* Séparer les données d'entraînement en jeux de données de validation

# MATÉRIEL ET MÉTHODES

## Présentation des modèles utilisés

### Visual Transformer

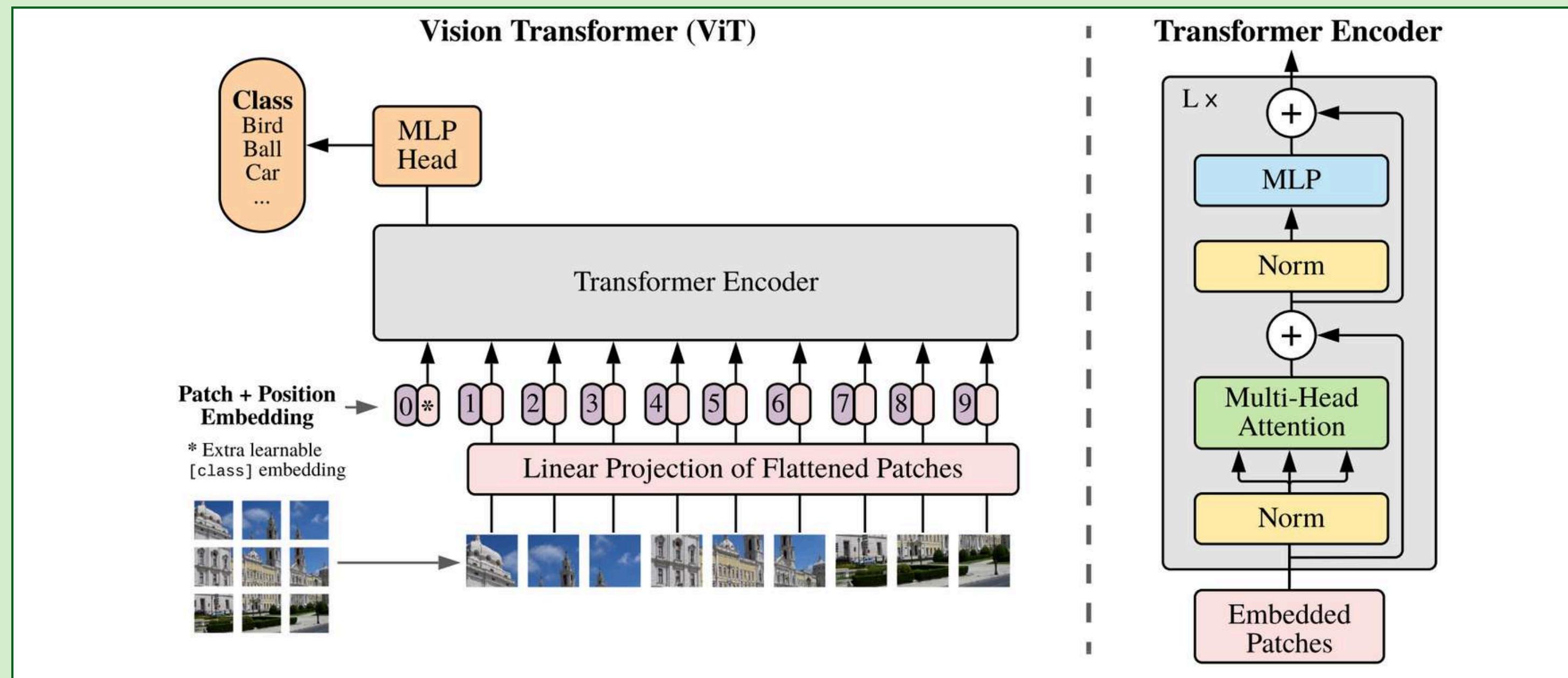


Figure d'après Dosovitskiy et al., 2020

# MATÉRIEL ET MÉTHODES

## Présentation des modèles utilisés

### Visual Transformer

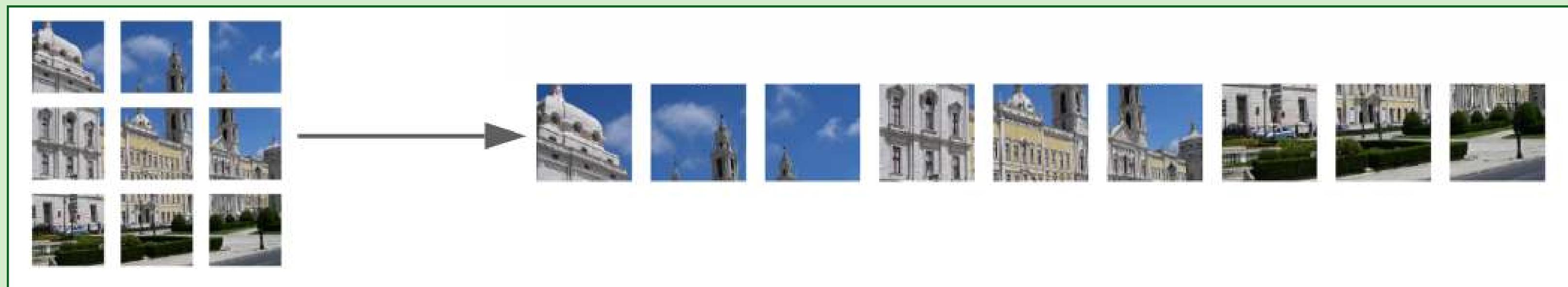


Figure d'après Dosovitskiy et al., 2020)

- Chaque patch est “écrasé” en un seul long vecteur
- Chaque patch est attaché à un vecteur de position
- Chaque vecteur est “embed” dans une matrice  $N \times D$

# MATÉRIEL ET MÉTHODES

## Présentation des modèles utilisés

### Visual Transformer



Figure d'après Dosovitskiy et al., 2020)



**La matrice est ensuite passée au Transformer**

# MATÉRIEL ET MÉTHODES

## Présentation des modèles utilisés

- \* Modèle : **Vit patch16 224** issu du package **timm**, pré-entraîné sur **ImageNet21k**
  - \* Modèle entraîné avec le package **PyTorch**
  - \* Fonction de loss **Cross-Entropy** et optimiseur **Adam**
  - \* Chaque modèle a été entraîné pendant **10** epochs avec un learning rate à **1e-4**
- 

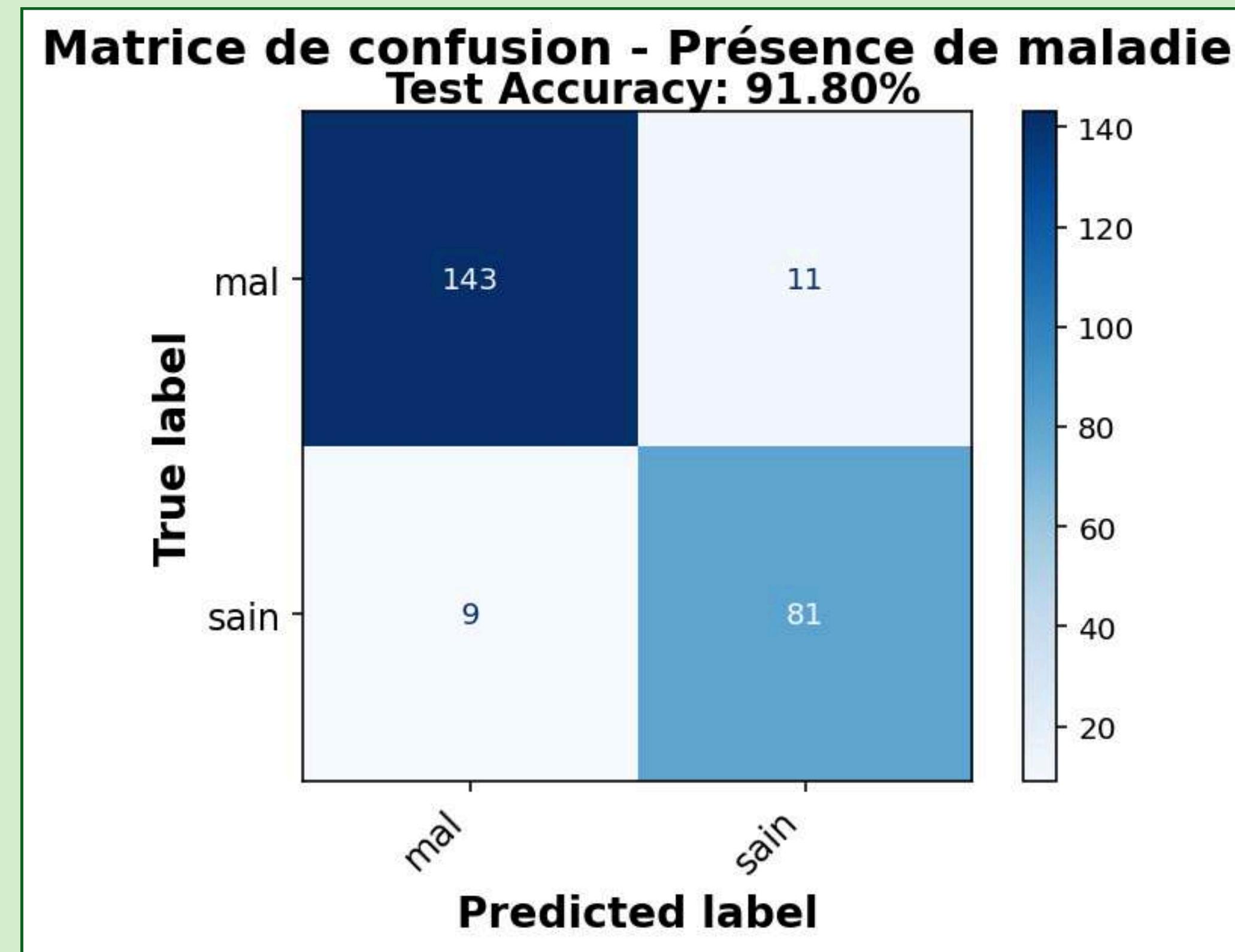
\* **Modèle 1 :** 2 classes

\* **Modèle 2 :** 13 classes

\* **Modèle 3 :** 27 classes

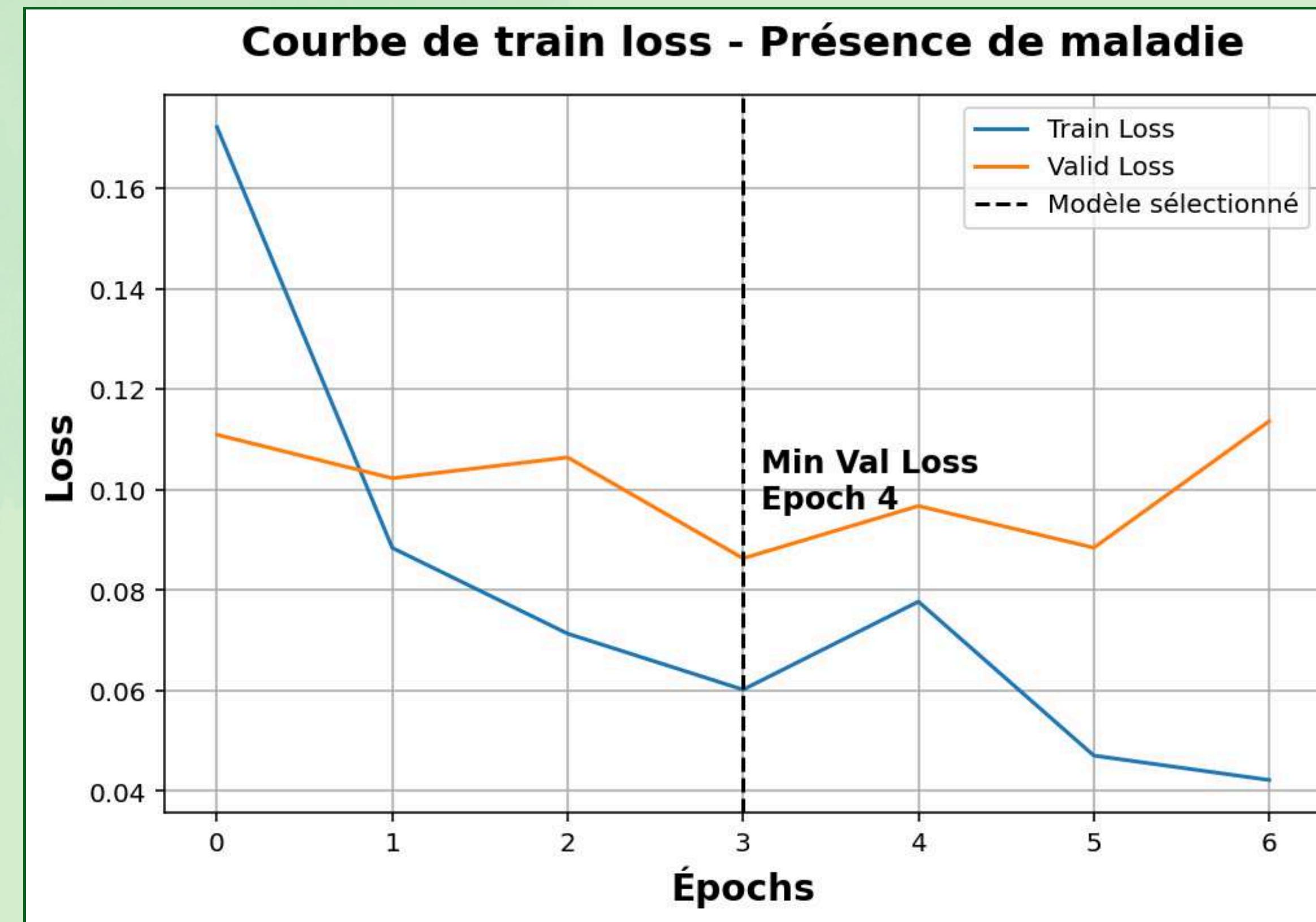
# RÉSULTATS

## Modèle 1 - Malade VS. Sain



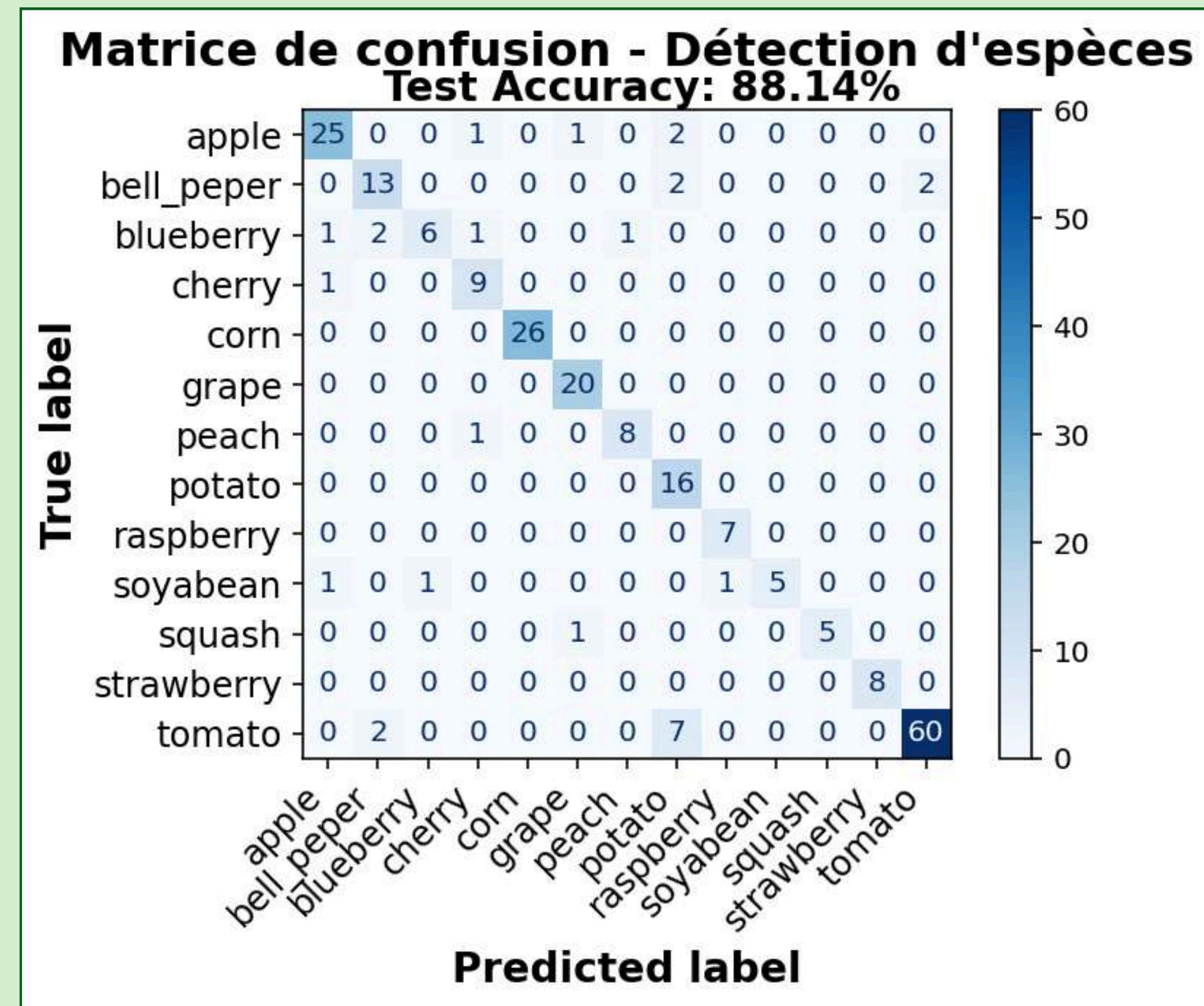
# RÉSULTATS

## Modèle 1 - Malade VS. Sain



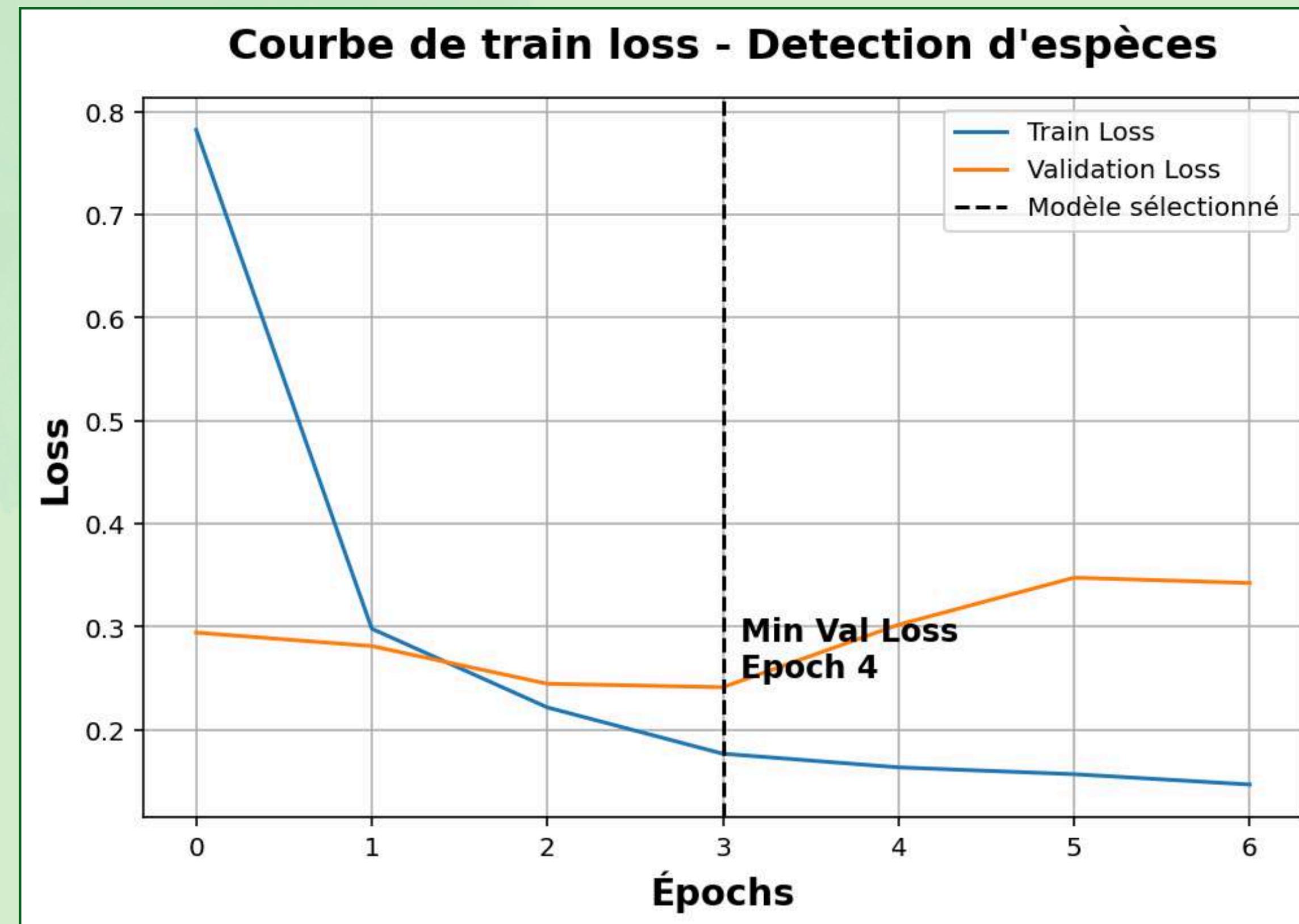
# RÉSULTATS

## Modèle 2 - Détection des espèces



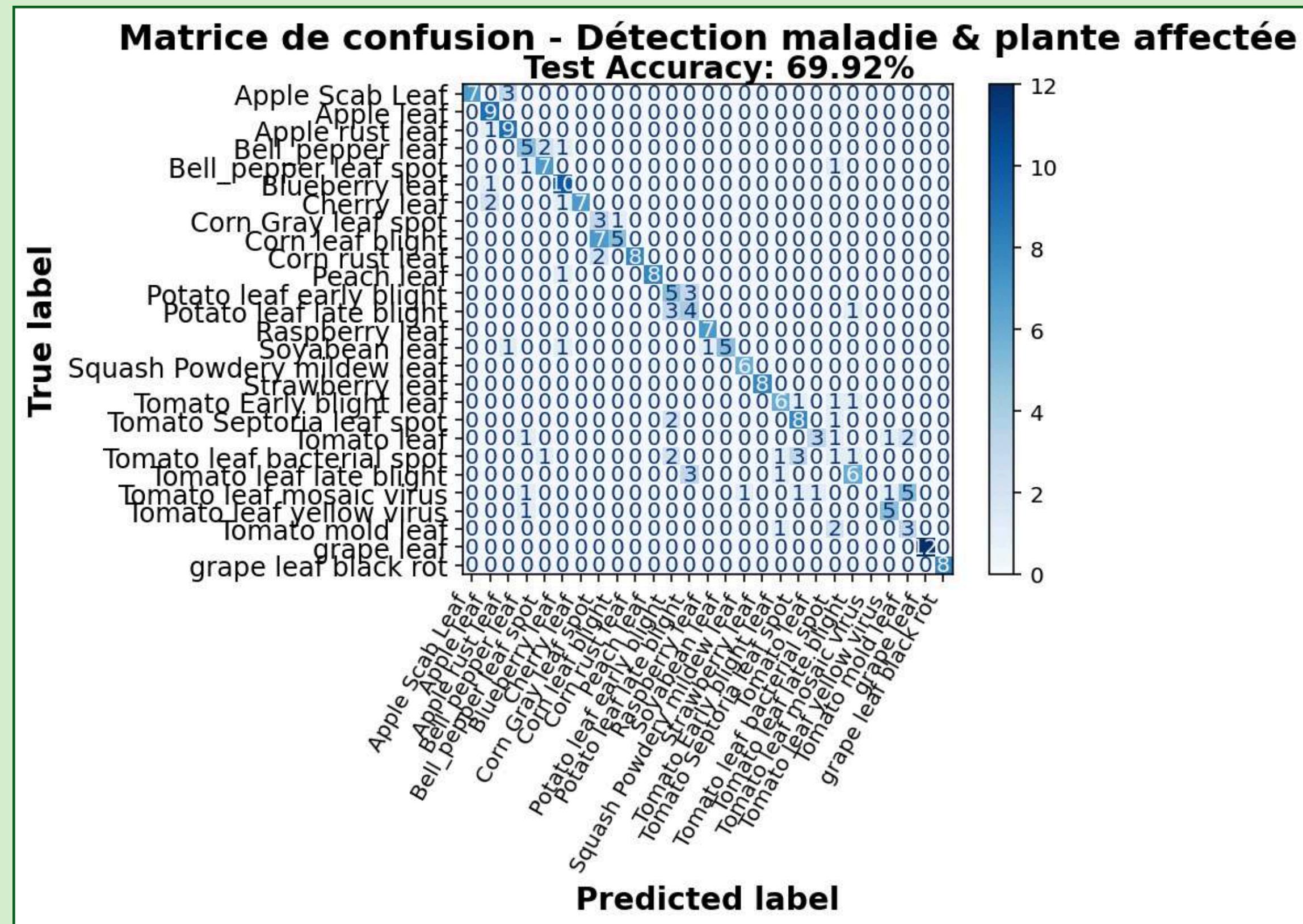
# RÉSULTATS

## Modèle 2 - Détection des espèces



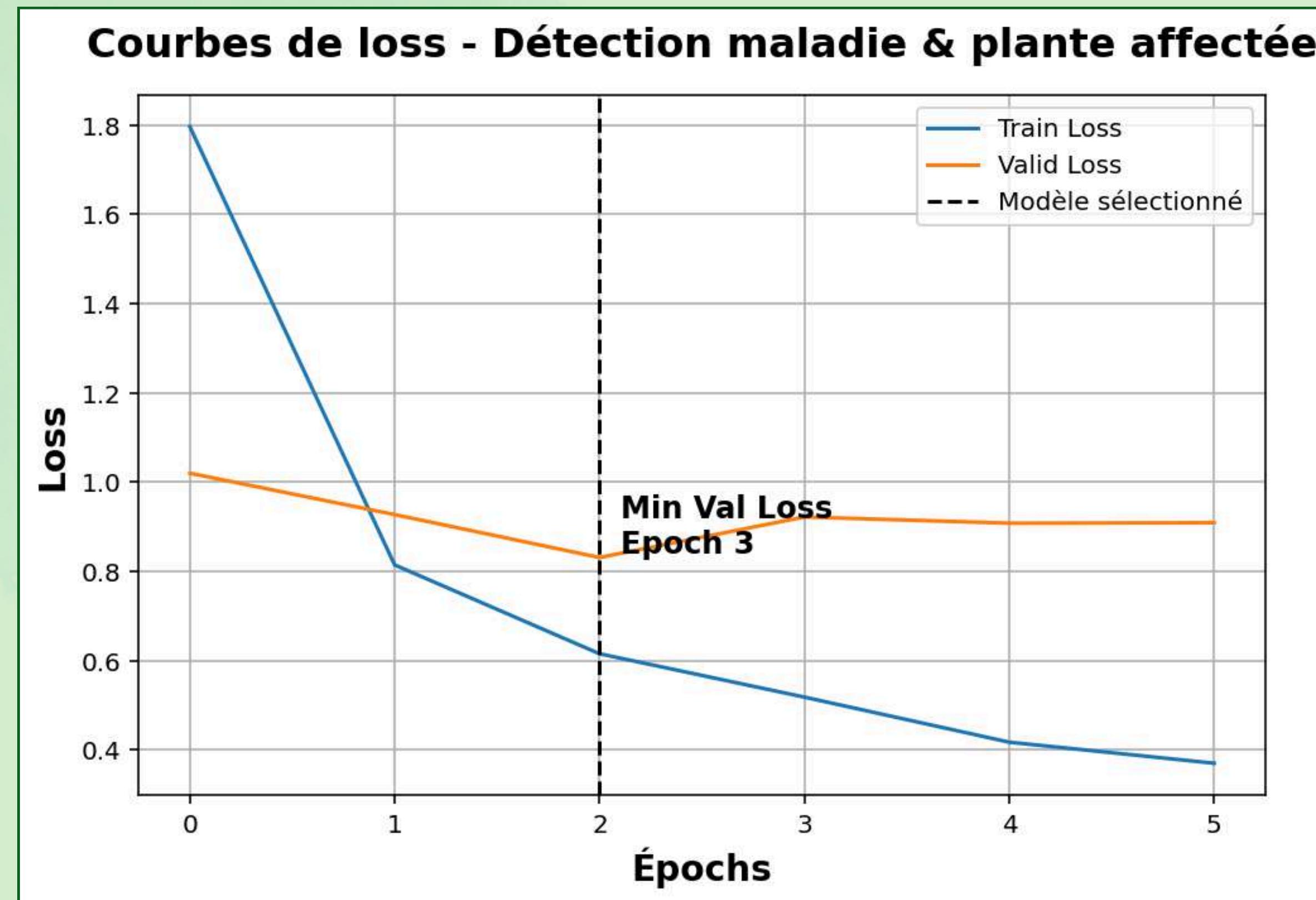
# RÉSULTATS

# Modèle 3 - Espèces et Maladies



# RÉSULTATS

## Modèle 3 - Espèces et Maladies



# DISCUSSION

## \* Résultats globaux à première vue peu satisfaisants

- Accuracy sur modèle **3 = 69.92%**

## \* Paramètres à modifier pour l'optimisation :

- Gel des paramètres de toutes les couches sauf la tête de classification  
*/!\ Geler les couches peut provoquer de l'overfitting*
- Agrandir le modèle (nombre de couches entraînées, réduire la taille des patchs)  
→ **81 millions** de paramètres

## \* Limites du ViT dans notre cas de figure :

- Jeu de données **très petit** pour un ViT
- Pas d'augmentation des données (limite de temps de calcul)
- Très peu d'epochs (**10**)

# DISCUSSION

## \* Comparaison avec CNN dans la littérature - data augmentation en plus

| Model           | Accuracy (Without Noise) | Accuracy (With Noise) |
|-----------------|--------------------------|-----------------------|
| ResNet50        | 0.5508                   | 0.6229                |
| EfficientNet-B3 | 0.7163                   | 0.7119                |
| DenseNet201     | 0.7034                   | 0.6695                |
| EfficientNet-B0 | 0.7203                   | 0.6737                |

Manjunatha Shettigere Krishna et al., 2025

→ Résultats similaires aux nôtres

## \* Comparaison avec d'autres ViT issus de la littérature

| Paper | Dataset(s)                                      | Accuracy | Precision | Recall | F1     | Main Contribution   |
|-------|---|----------|-----------|--------|--------|---|
| [89]  | PlantVillage                                    | 99.92%   | 0.996     | 0.996  | 0.995  | Proposing LeafyGAN, a GAN-based augmentation framework for synthetic leaf disease data generation, enabling lightweight MobileViT models to achieve high diagnosis accuracy even with limited real data |
|       | PlantDoc  | 75.72%   | 0.75      | 0.74   | 0.72   |   |
| [63]  | Combination of PlantVillage, PlantDoc, and CD&S | 99.24%   | 0.9915    | 0.9937 | 0.9926 | Adapting MaxViT structure with SE blocks and implementing Global Response Normalization in MLP layers for 4-class maize data  |

Modifié d'après Mehdipour et al., 2025

# CONCLUSION

- \* Modèles **ViT** sont très efficaces pour la classification d'images, mais ils nécessitent une grande quantité de données
  - \* Nous avons testé **3** versions du modèle pour observer la précision des prédictions
  - \* Meilleur modèle est le **Modèle 1**, car c'est celui qui a le moins de classes.
- 

Résultats à première vue faibles, qui correspondent au final bien à ceux observés dans la littérature.

Entraîner les modèles plus longtemps avec une plus grande quantité de données pour obtenir des résultats plus concluants.

Jeu de données très intéressant qui pourrait être utilisé dans des perspectives de détermination de plantes à l'espèce.



MERCI DE VOTRE  
ATTENTION

# BIBLIOGRAPHIE

- \* Batool, A., & Byun, Y.-C. (2025). Revolutionizing plant disease diagnosis through vision-based intelligence and next-generation computing. *Computers and Electrical Engineering*, 128, 110695. <https://doi.org/10.1016/j.compeleceng.2025.110695>
- \* Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (No. arXiv:2010.11929). arXiv. <https://doi.org/10.48550/arXiv.2010.11929>
- \* Krishna, M. S., Machado, P., Otuka, R. I., Yahaya, S. W., Neves Dos Santos, F., & Ihianle, I. K. (2025). Plant Leaf Disease Detection Using Deep Learning: A Multi-Dataset Approach. *J*, 8(1), 4. <https://doi.org/10.3390/j8010004>
- \* Mehdipour, S., Mirroshandel, S. A., & Tabatabaei, S. A. (2025). Vision Transformers in Precision Agriculture: A Comprehensive Survey (Version 4). arXiv. <https://doi.org/10.48550/ARXIV.2504.21706>
- \* Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library (Version 1). arXiv. <https://doi.org/10.48550/ARXIV.1912.01703>
- \* Salman, Z., Muhammad, A., & Han, D. (2025). Plant disease classification in the wild using vision transformers and mixture of experts. *Frontiers in Plant Science*, 16, 1522985. <https://doi.org/10.3389/fpls.2025.1522985>
- \* Shoaib, M., Shah, B., El-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, 14, 1158933. <https://doi.org/10.3389/fpls.2023.1158933>
- \* Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc: A Dataset for Visual Plant Disease Detection. *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 249-253. <https://doi.org/10.1145/3371158.3371196>

# ANNEXES

