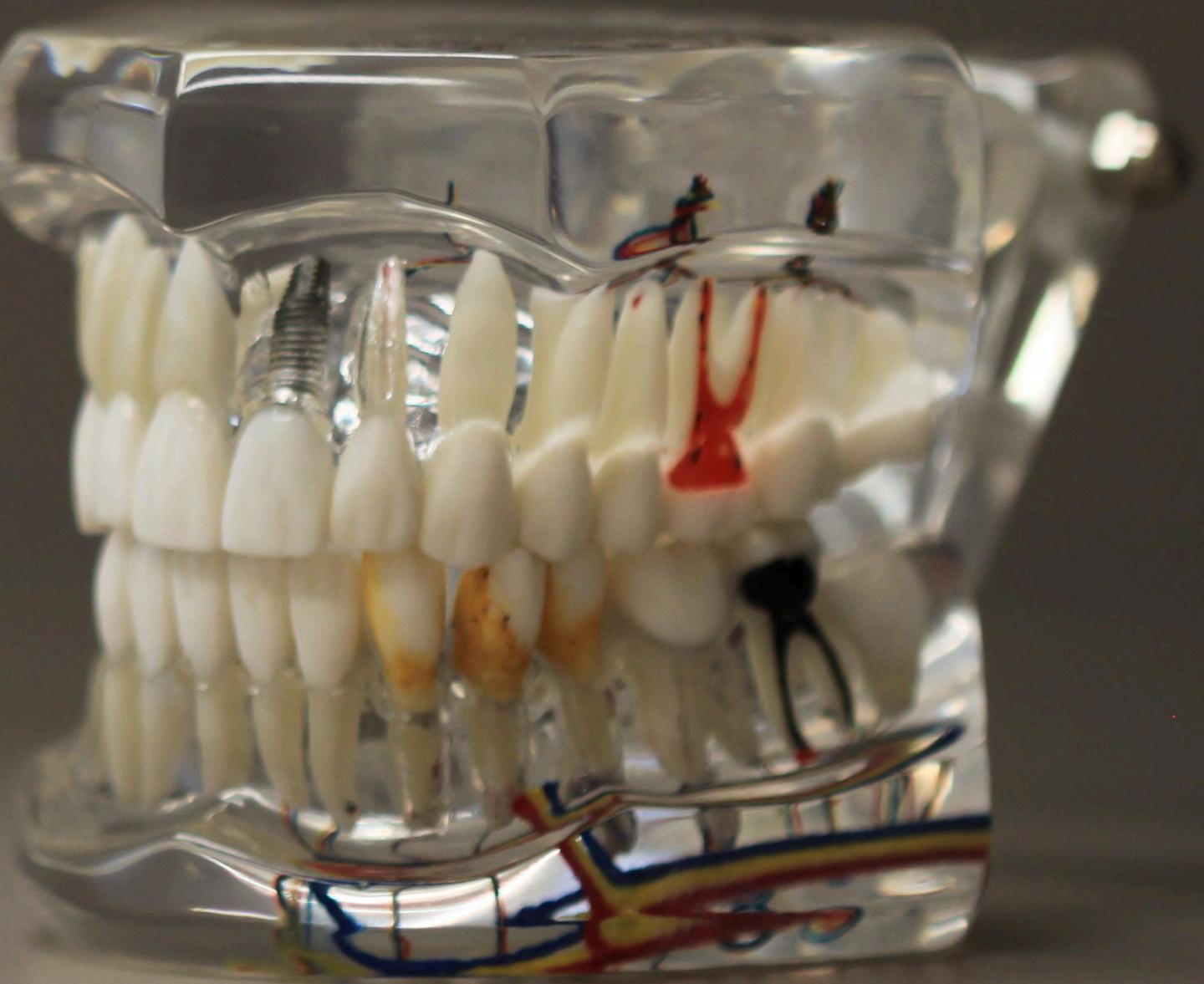


Projet HDDL

 *Linda CHIKHAOUI, Victor LIN,
Emeline CASTELLO*





Contexte

Constat sur le terrain

- Les dentistes partagent des images chaque jour sur les groupes Facebook pour demander un avis
- Création de groupes d'entraides informels entre confrères

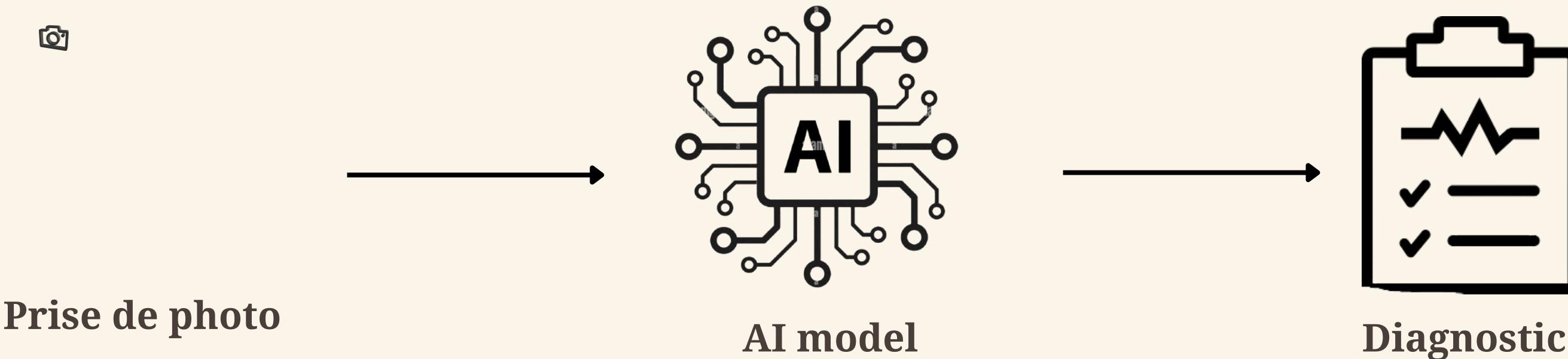
Objectif

---> Transformer cette logique en un outil structuré, éthique et intelligent.

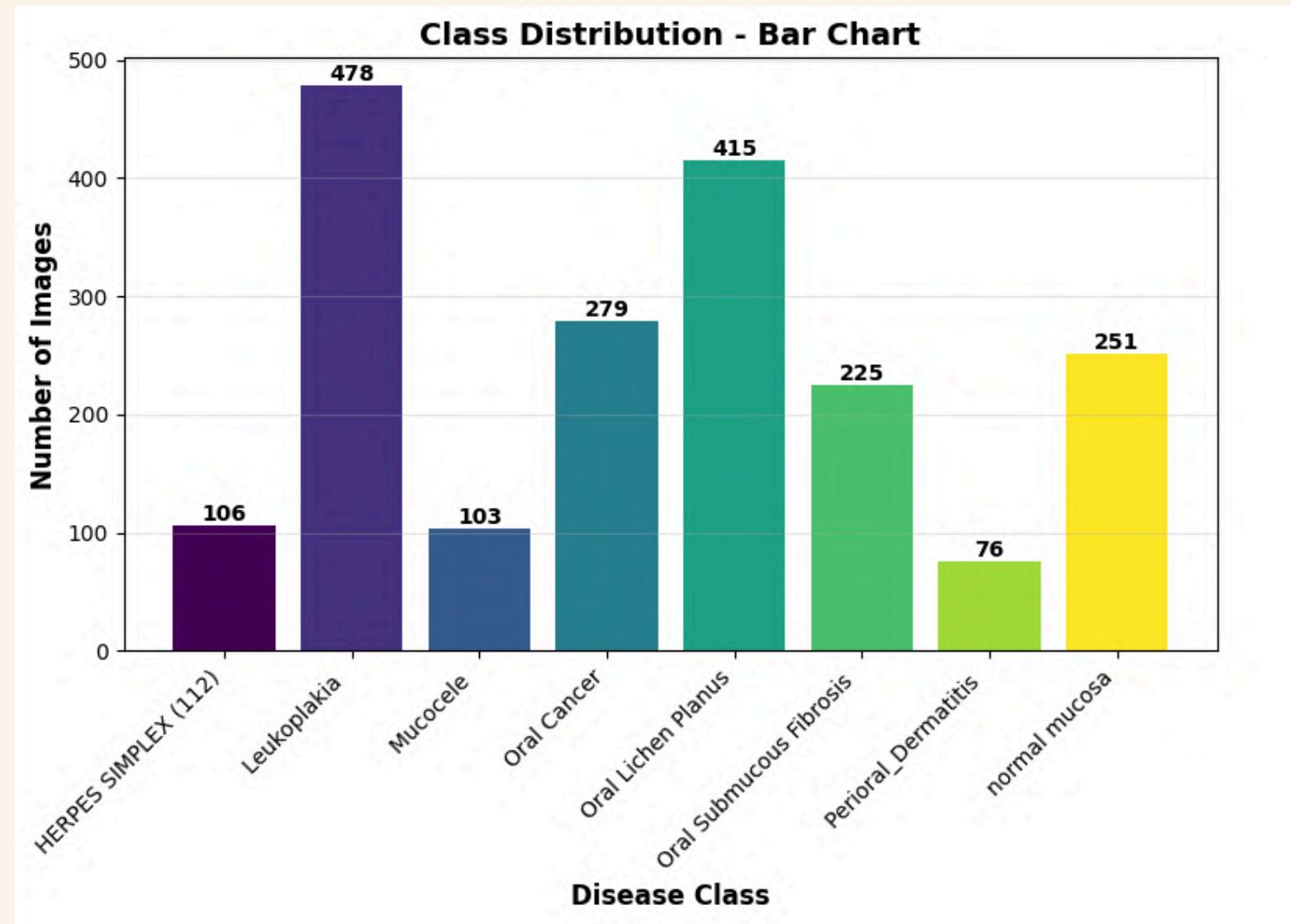
Solution ORALYZE

Application mobile destinée aux:

- Professionnels de santé (dentistes et spécialistes en chirurgie orale)
- Médecins généralistes
- Etudiants...



Présentation de données



- 8 pathologies
- 1933 images labelisées au total

Sommaire

1

CNN

2

VIT

3

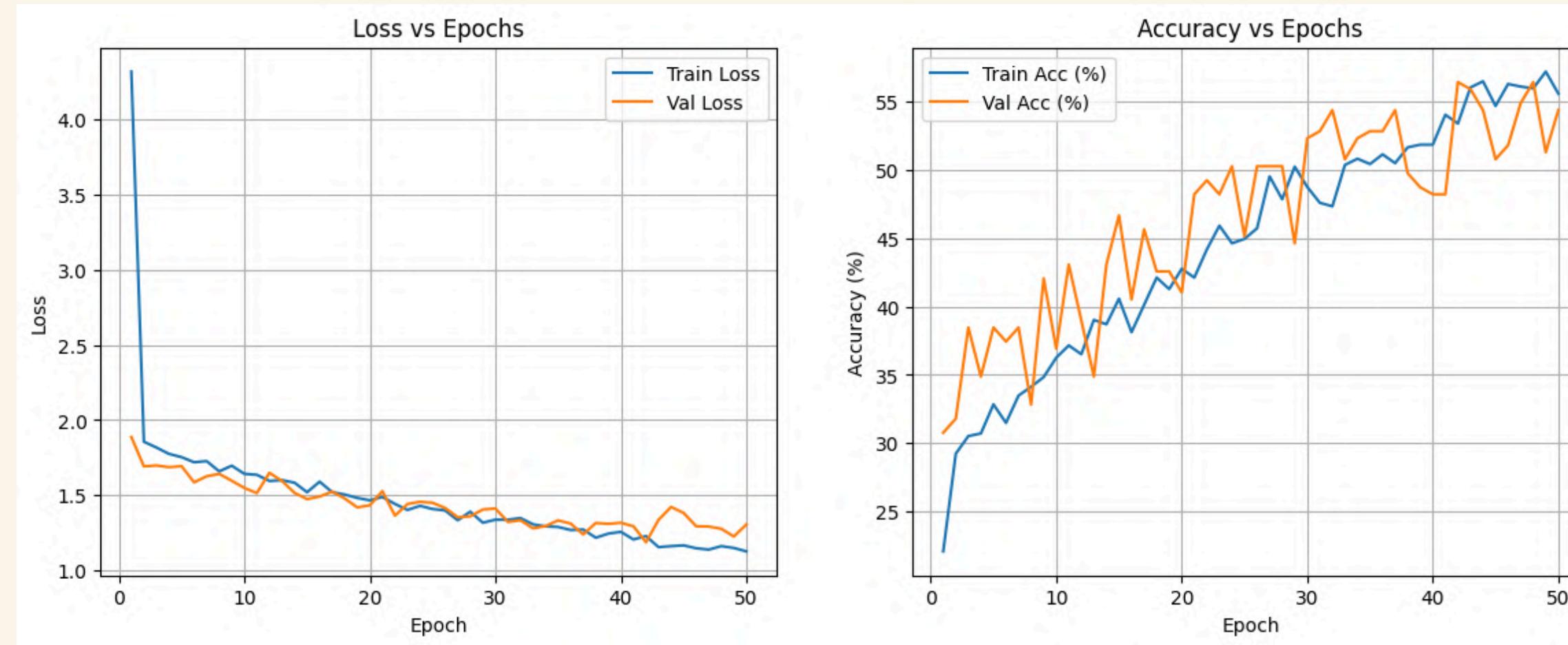
Conclusion



Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN)

Architecture (3 blocs convolutionnels)



Chaque bloc contient :

- 2 couches de convolution
- Chaque couche est suivie d'une fonction d'activation **ReLU** et d'une **batch normalisation**
- À la fin de chaque bloc, un **max pooling** est appliqué pour réduire la dimension spatiale
- Une couche **Fully Connected** pour faire la classification finale

Self-Supervised Learning (SSL)

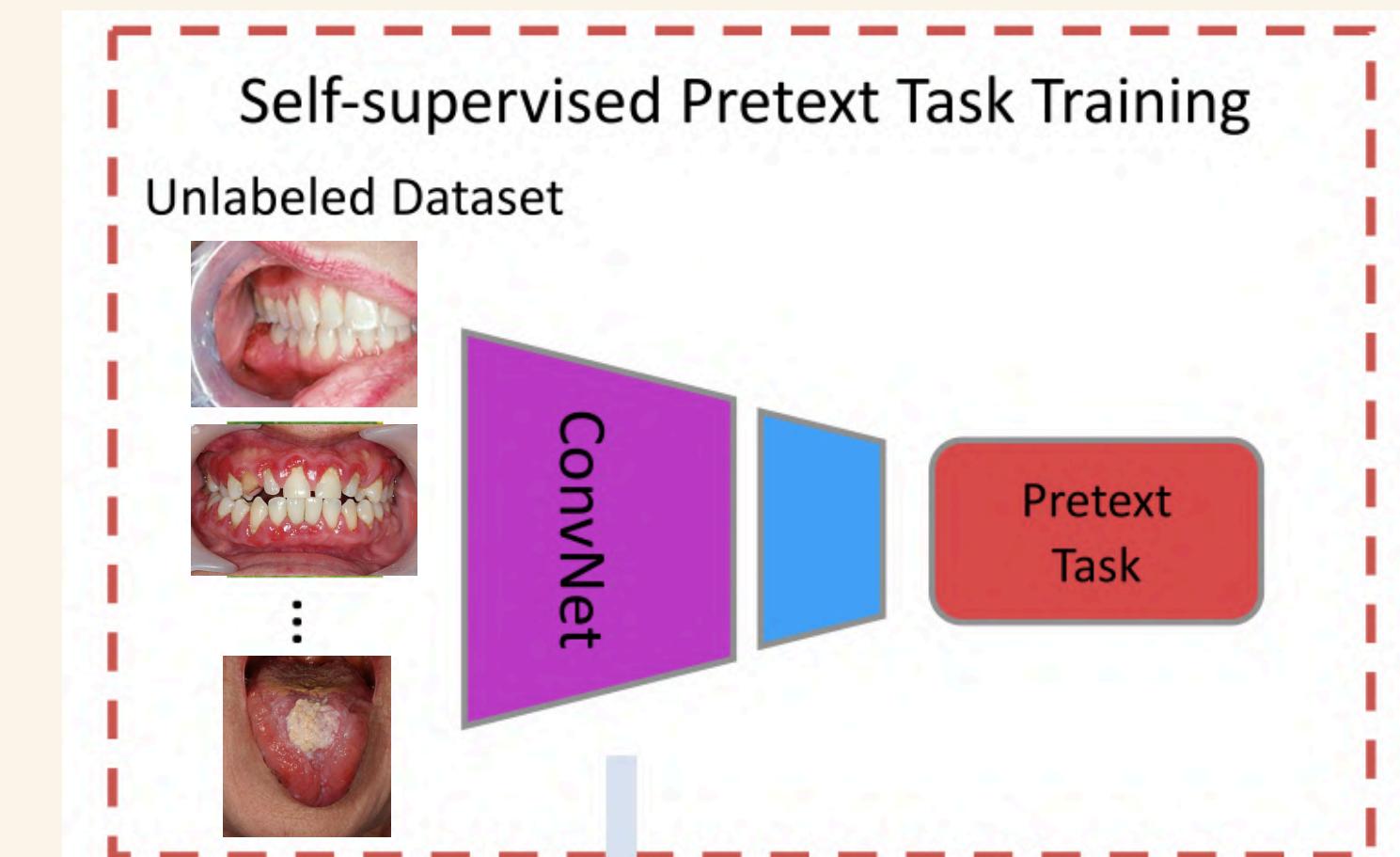
Taille du dataset: 1333 images non labelisées

1) *Pretext Task*

→ Apprendre les features à partir des images non labelisées

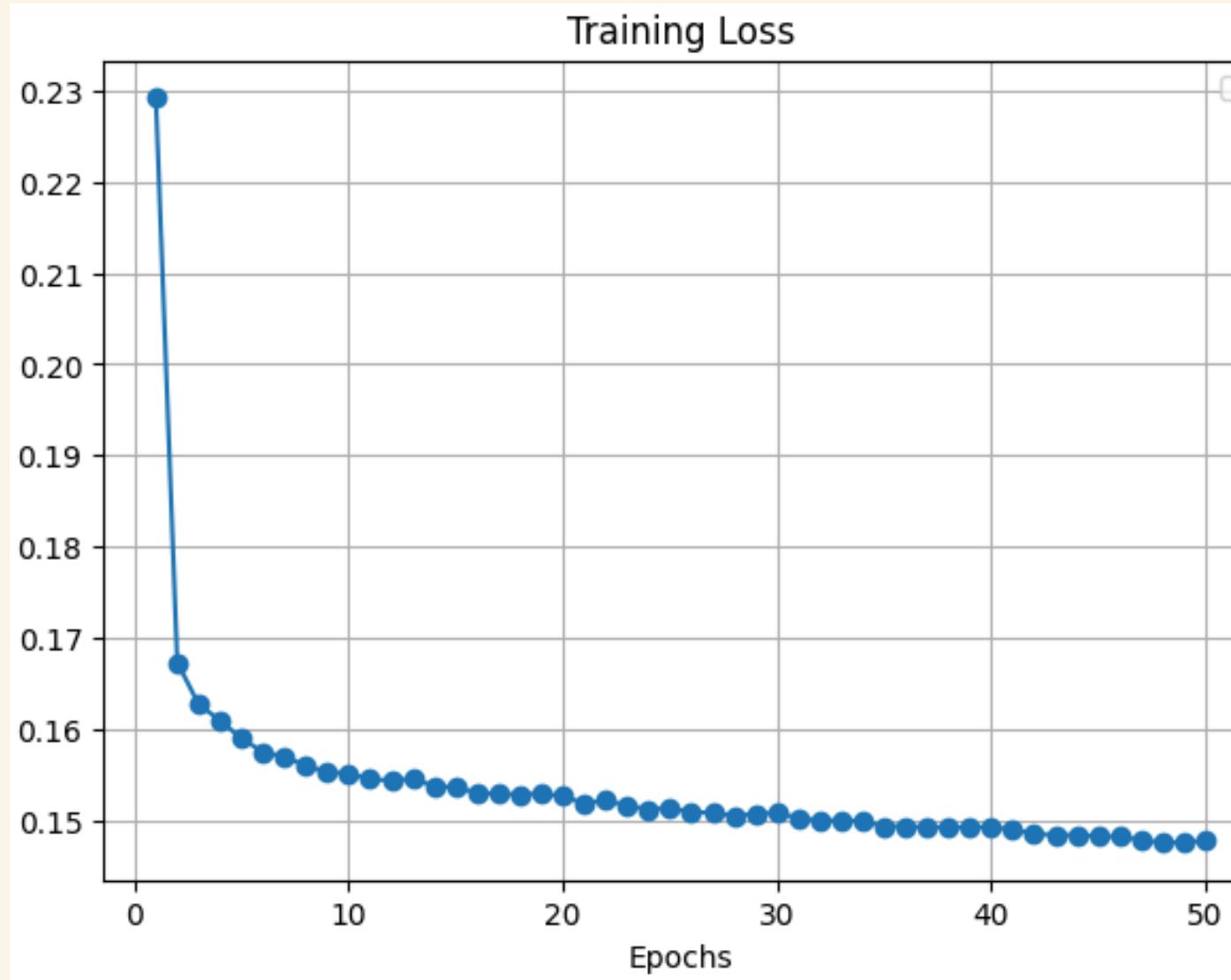
2) *Downstream Task*

→ Classification des pathologies



Tâche de prétexte 1 : Colorization

- Phase d'entraînement



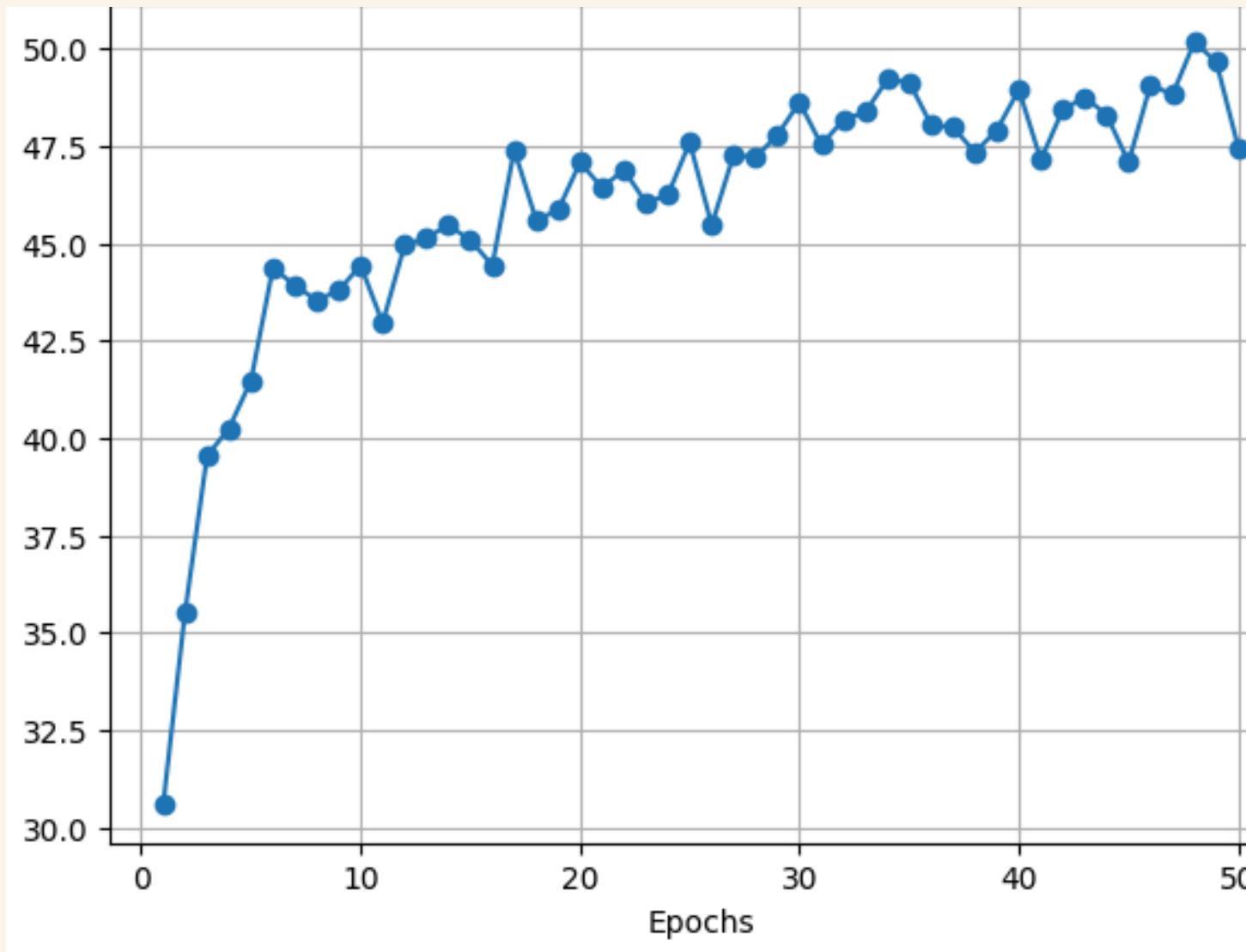
Évolution de la perte d'entraînement
sur 50 époques



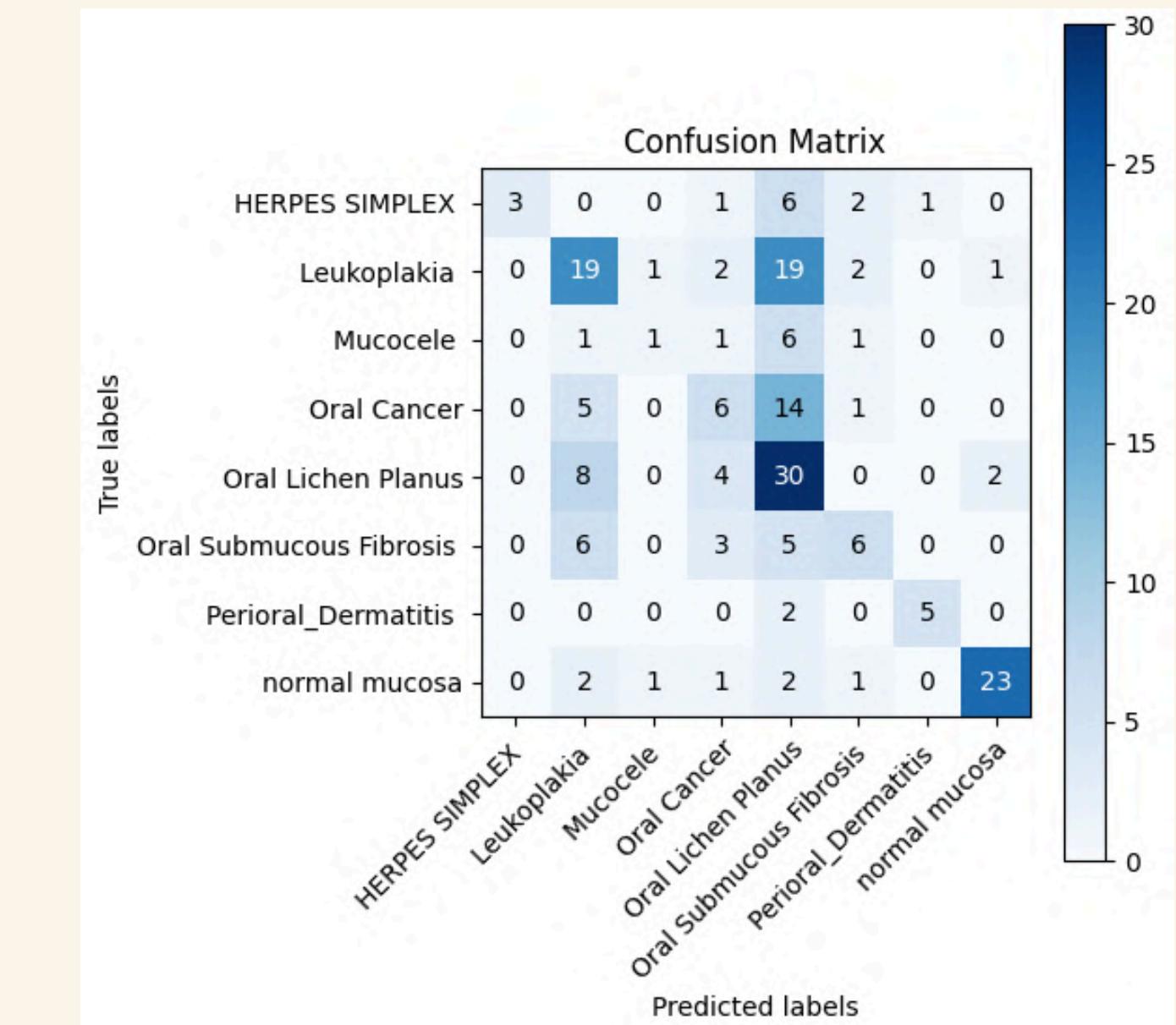
Résultats de la tâche de prétexte de
colorization

Tâche de prétexte 1 : Colorization

- Fine Tuning

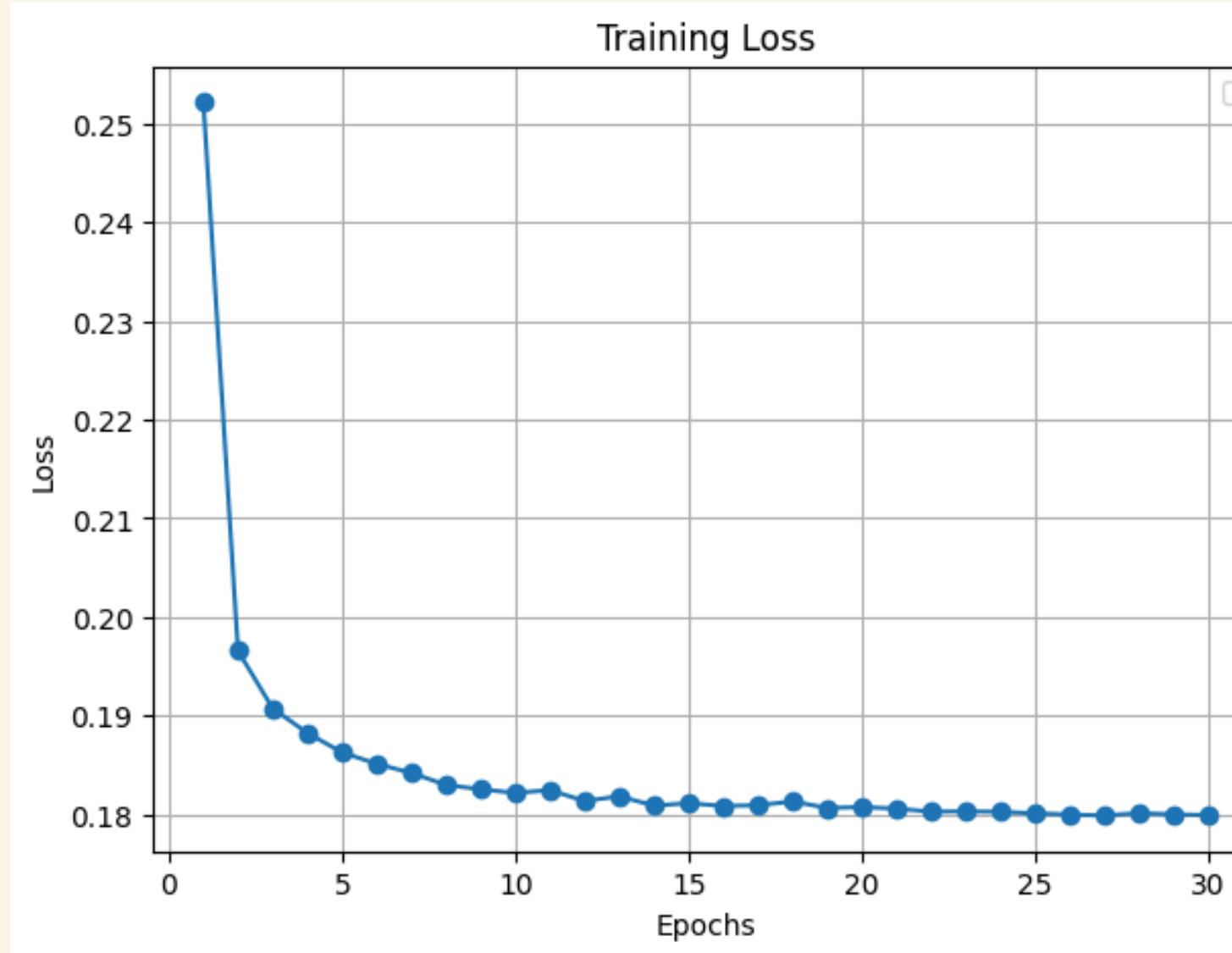


Training Accuracy : 47.46%

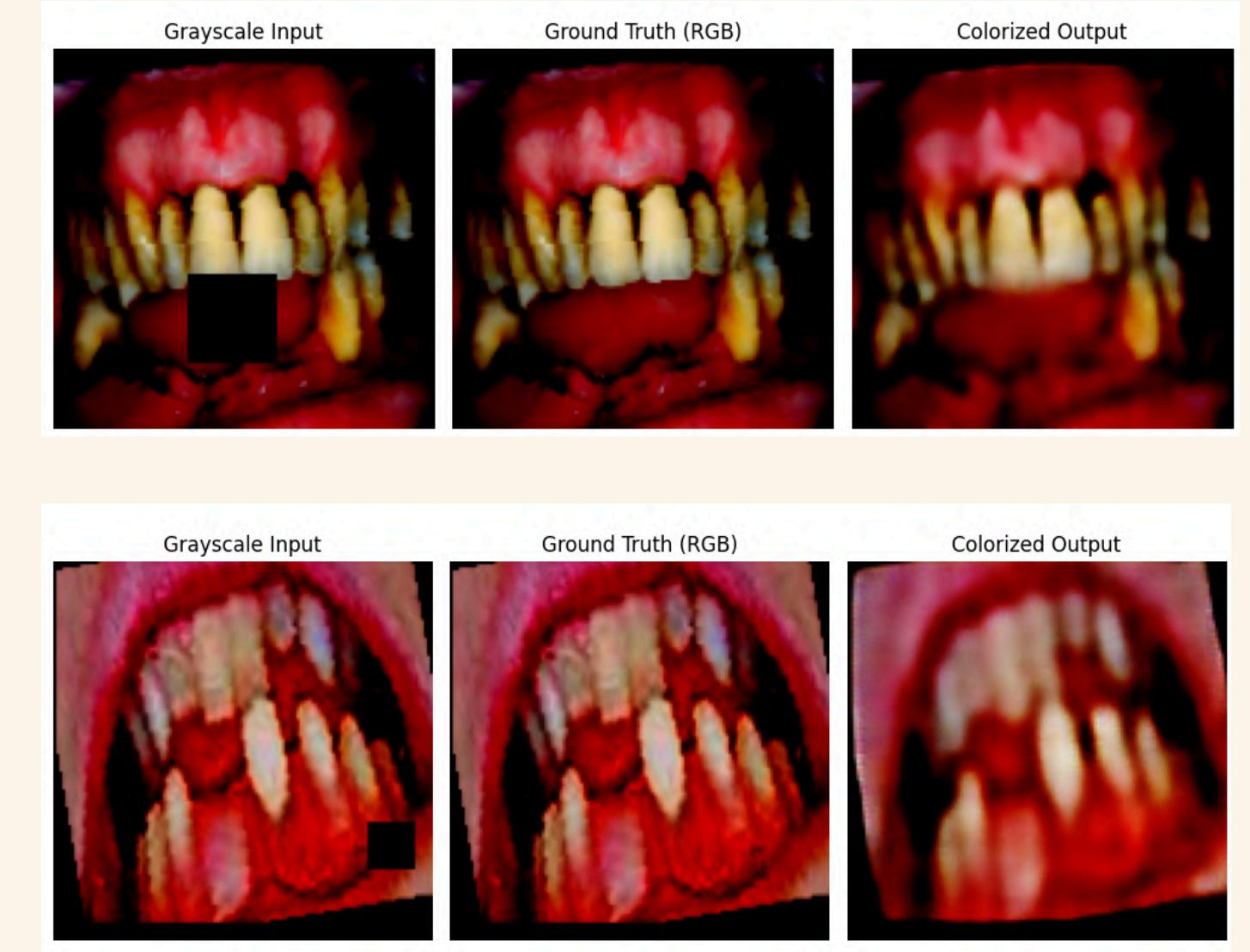


Accuracy Test: 47.94%

Tâche de prétexte 2 : Inpainting



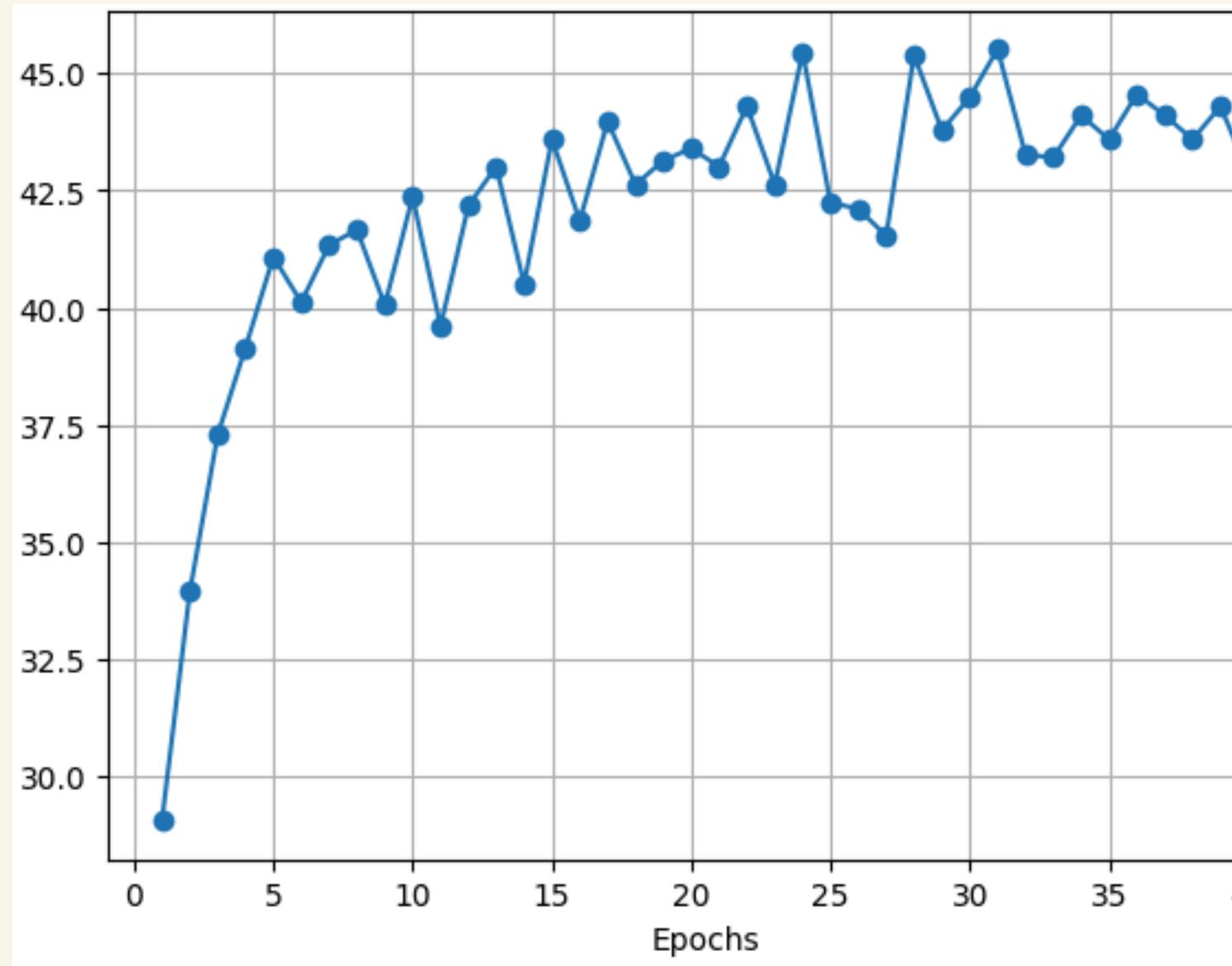
Évolution de la perte d'entraînement
sur 30 époques



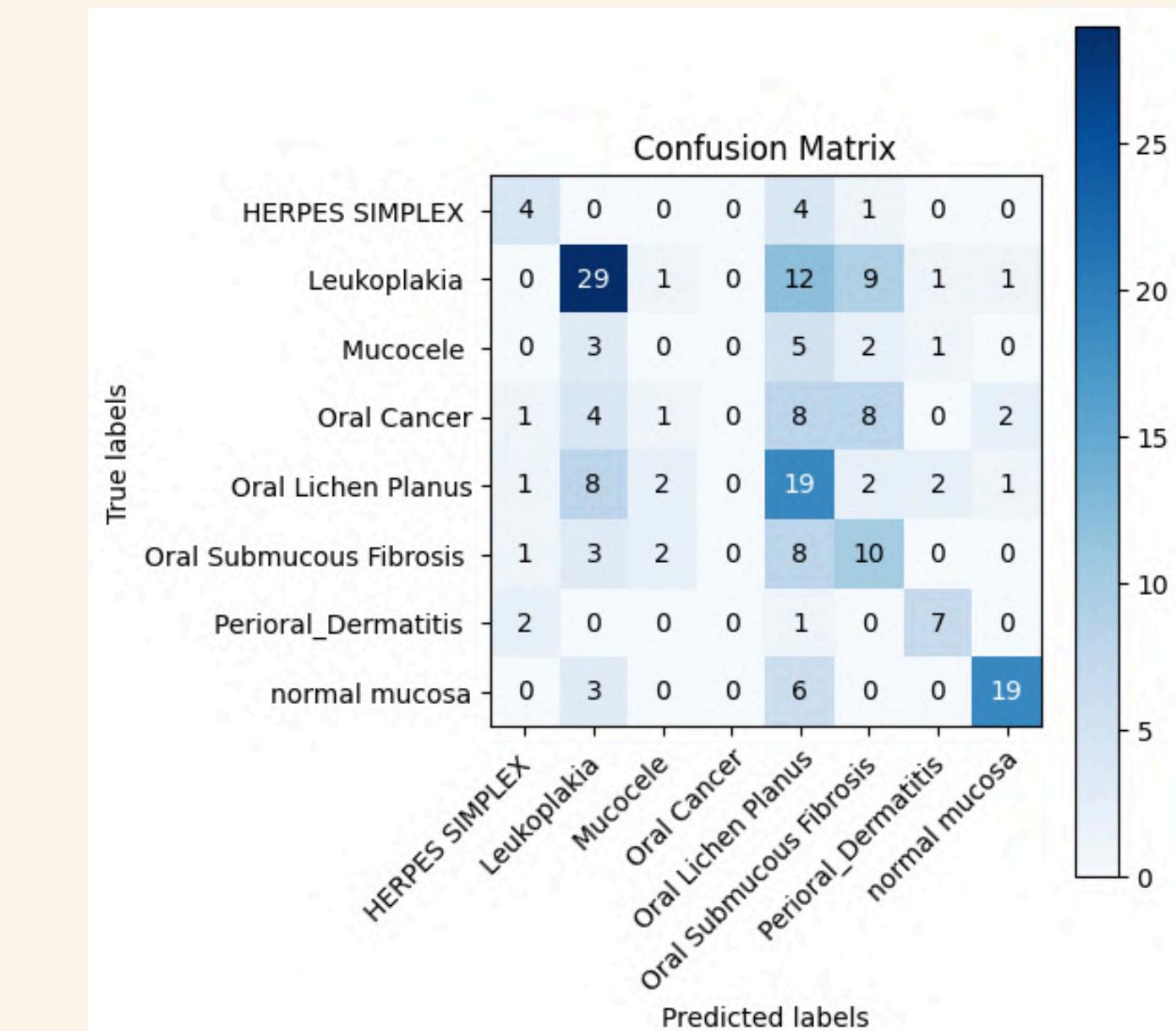
Résultats de la tâche de prétexte de
Inpainting

Tâche de prétexte 2 : Inpainting

- Fine Tuning

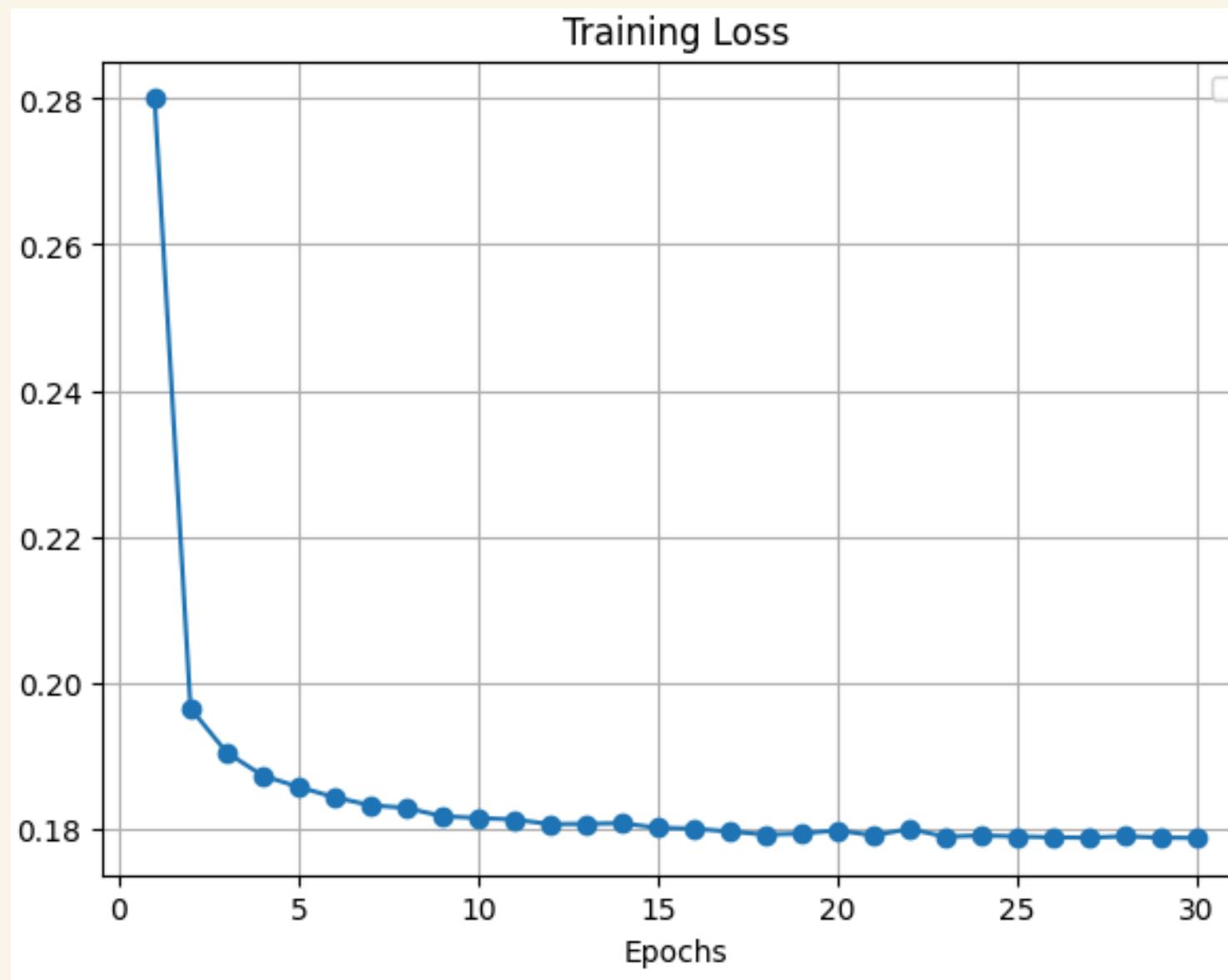


Training Accuracy : 42.88%

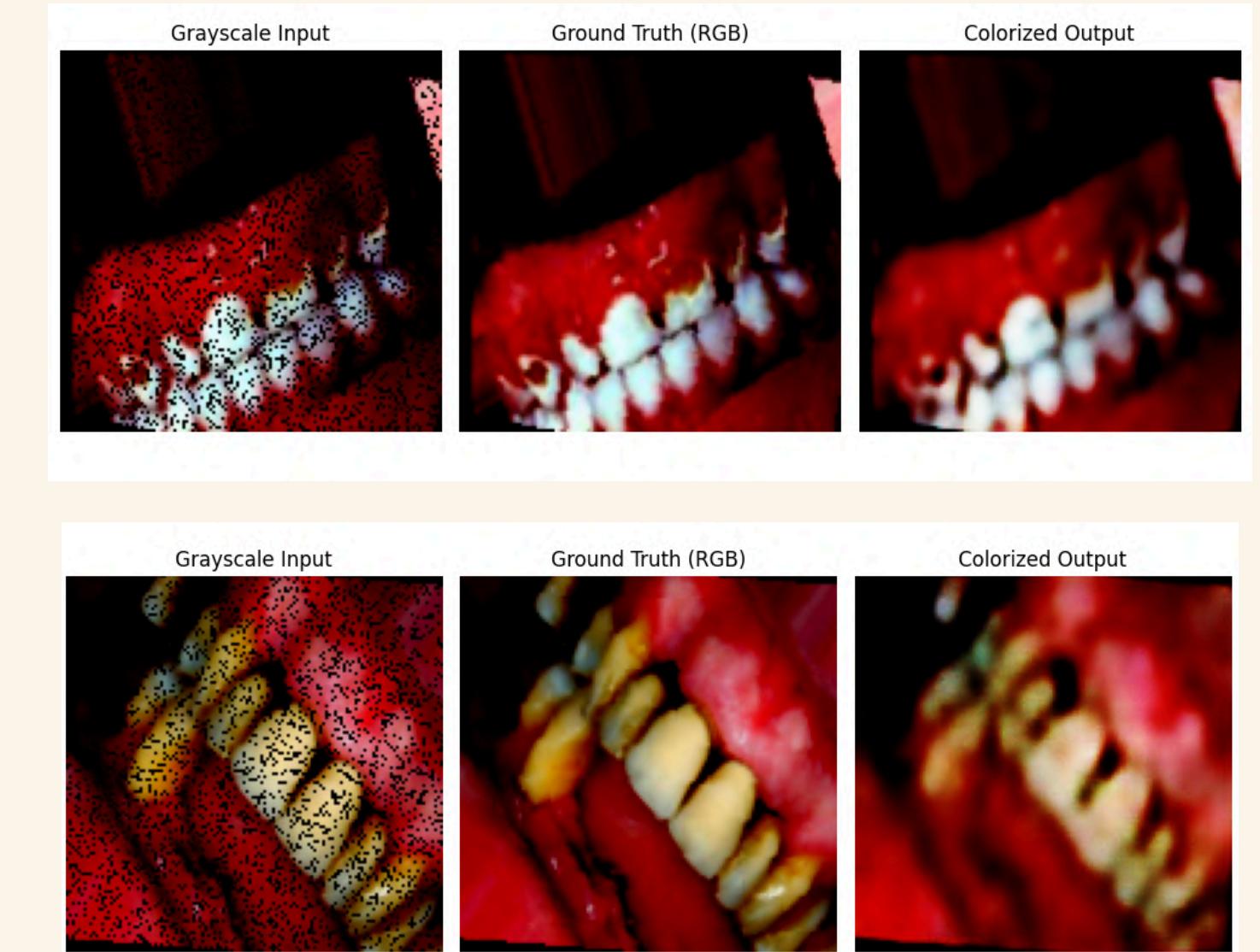


Accuracy Test: 45.36%

Tâche de prétexte 3 : Masked Autoencoder



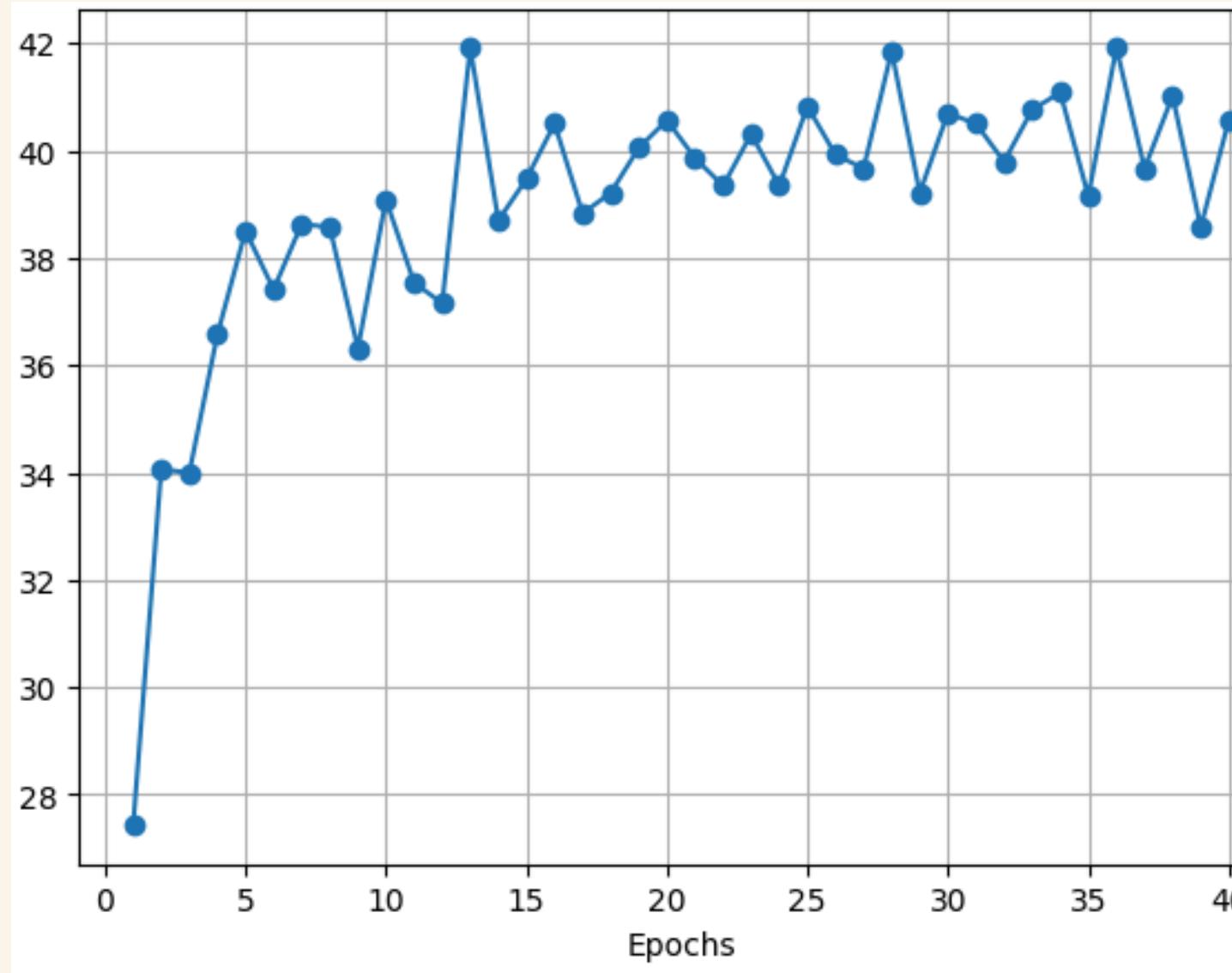
Évolution de la perte d'entraînement
sur 30 époques



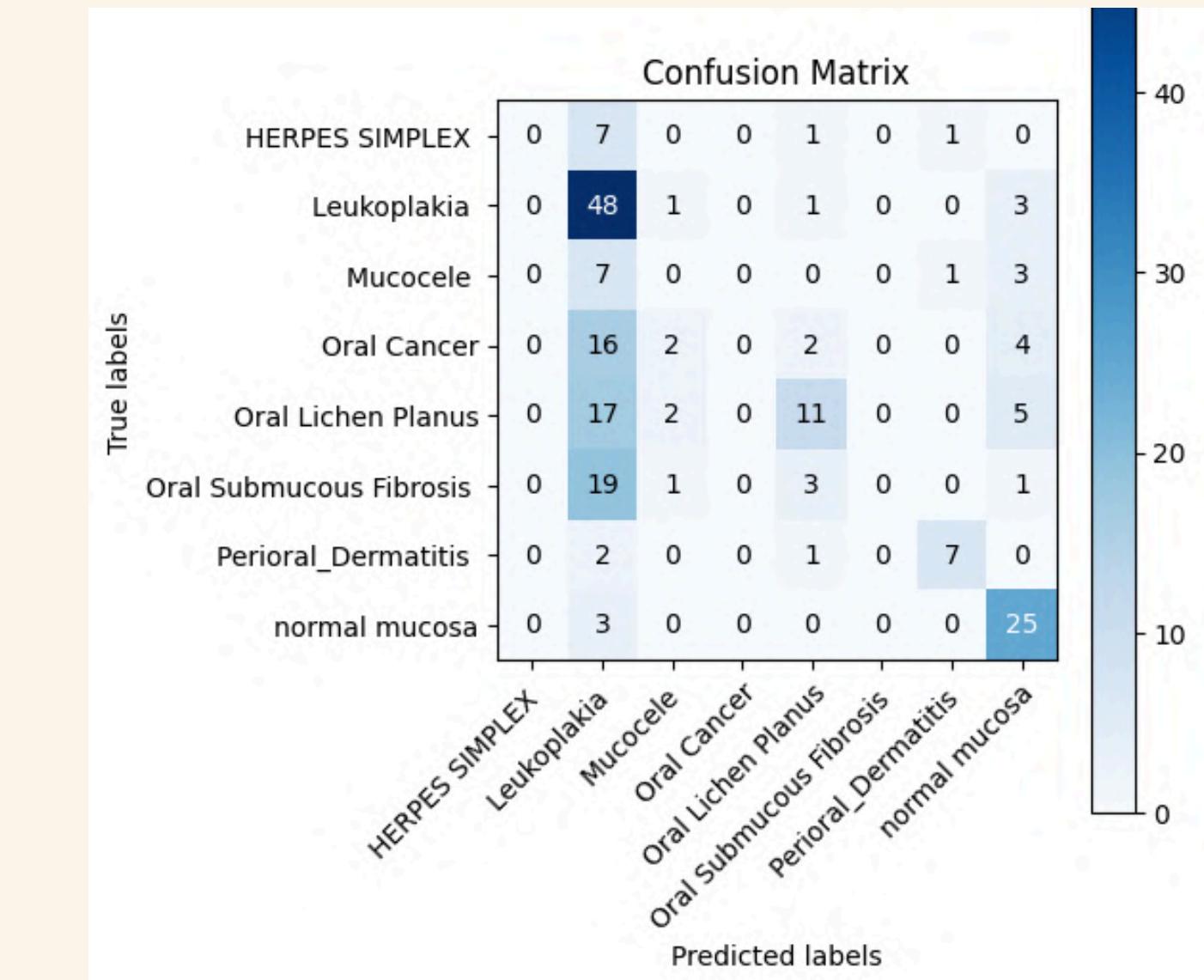
Résultats de la tâche de prétexte de
masquage

Tâche de prétexte 3 : Masked Autoencoder

- Fine Tuning



Training Accuracy : 40.57%



Accuracy Test: 46.91%

CNN Fine-Tuning

VGG16

Empilement de petites convolutions 3×3 + ReLU

Architecture régulière : blocs convolutionnels + max pooling
→ réduction progressive de la résolution

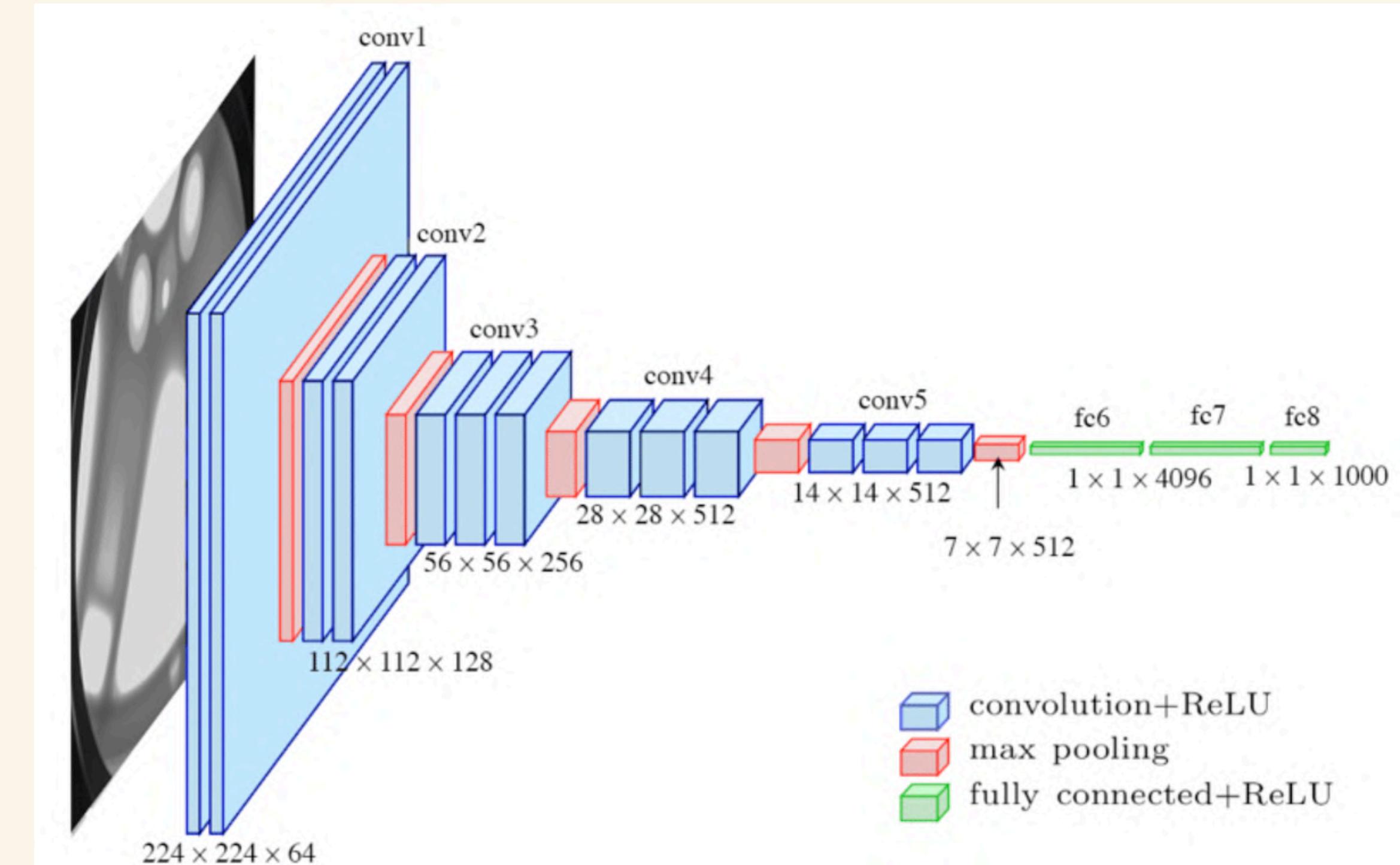
Profondeur croissante : canaux croissants

Extraction de features grossières au début et fines à la fin

Fully connected à la fin

Avantage : backbone efficace

Inconvénient : modèle simple mais lourd



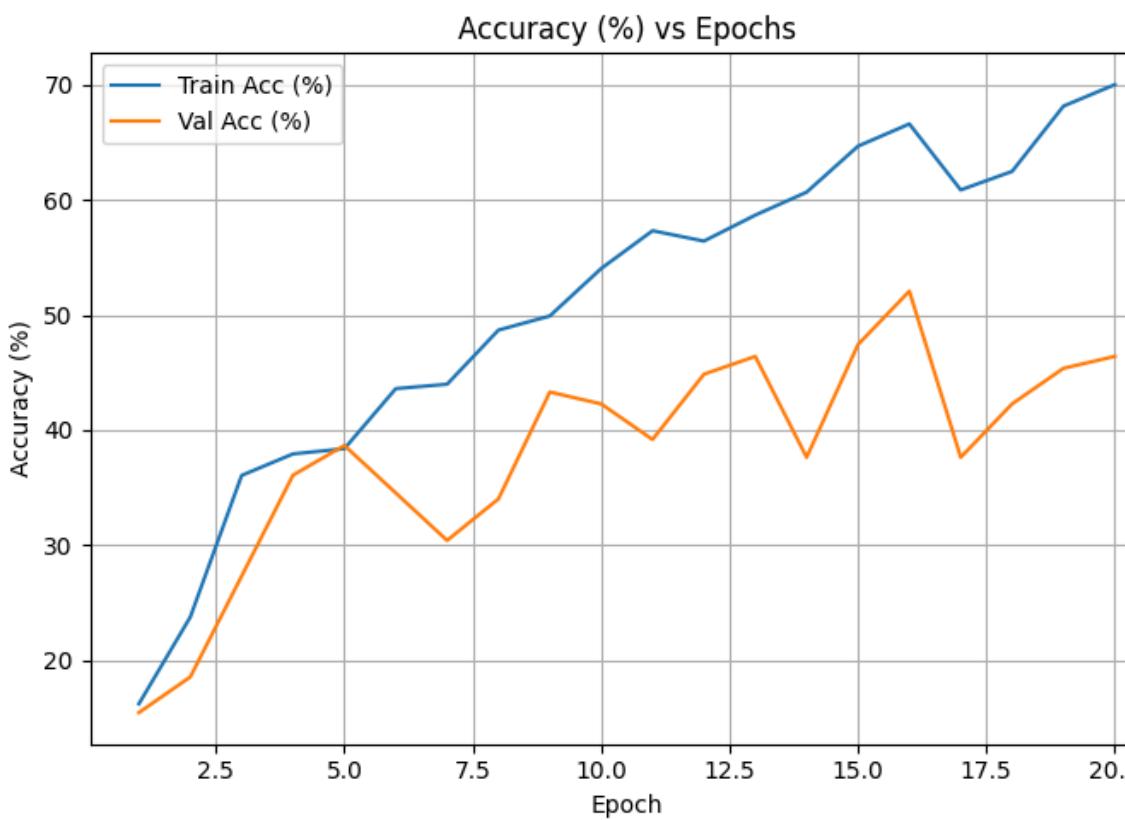
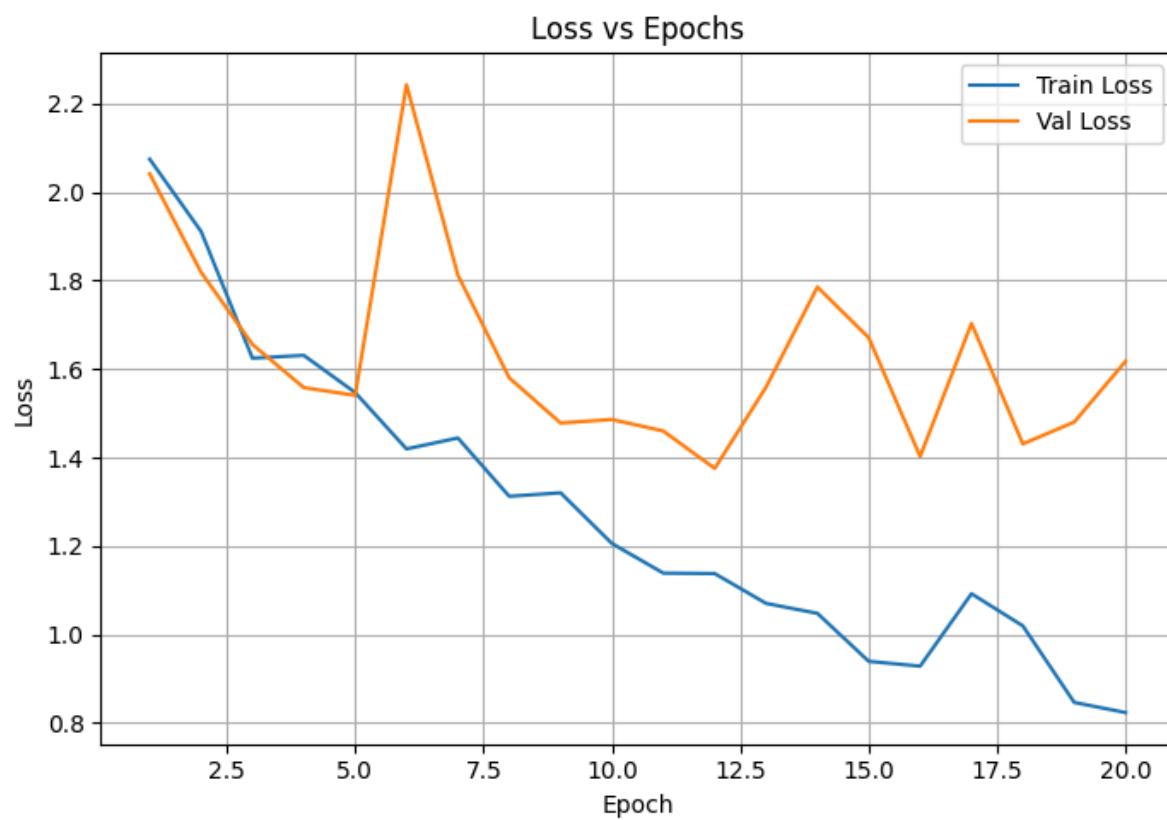
Architecture de VGG16

CNN Fine-Tuning

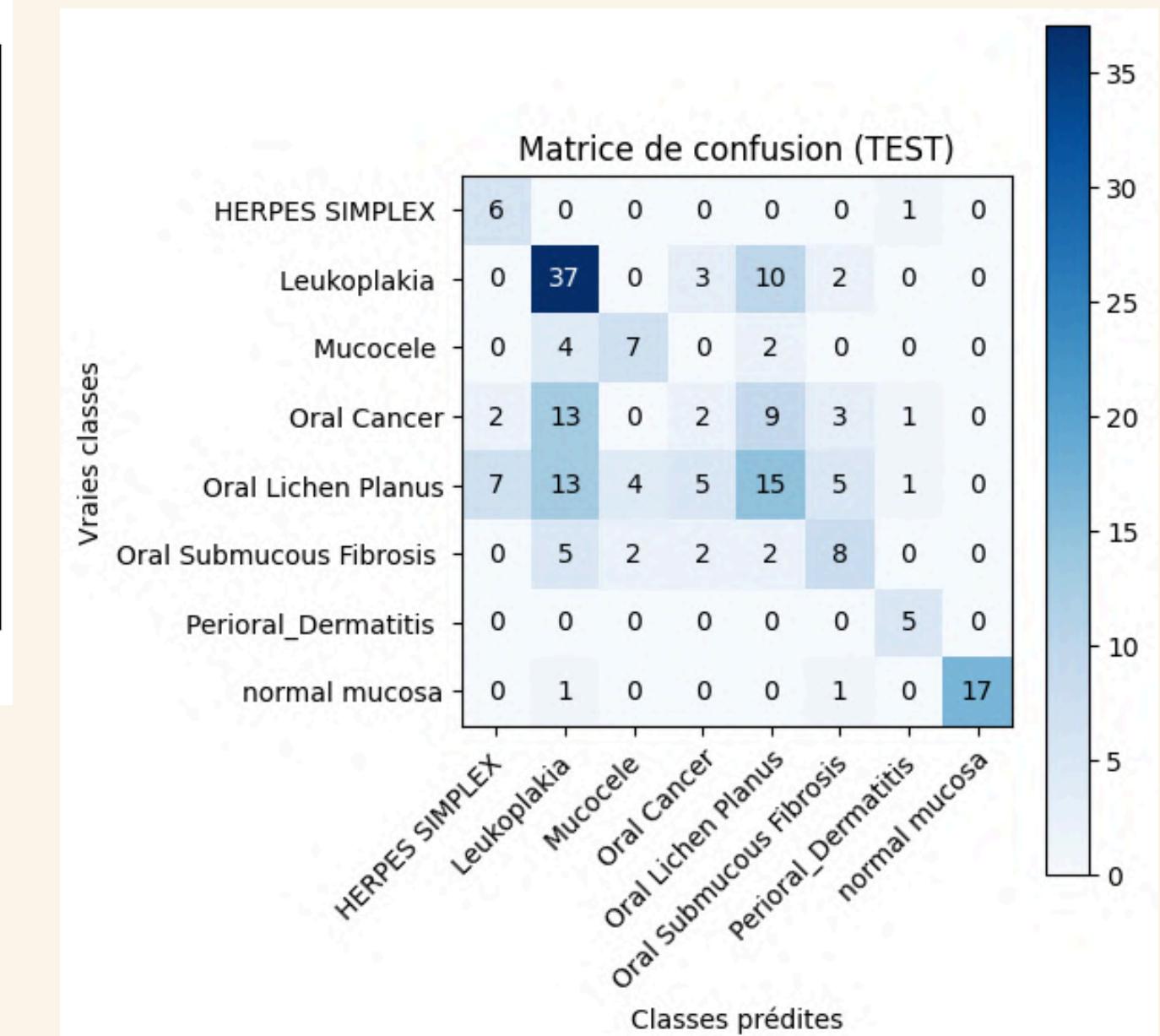
VGG16

Fine-tuning complet avec early stop

Evolution des losses et accuracies



Global Accuracy Score sur Test : 49.74%

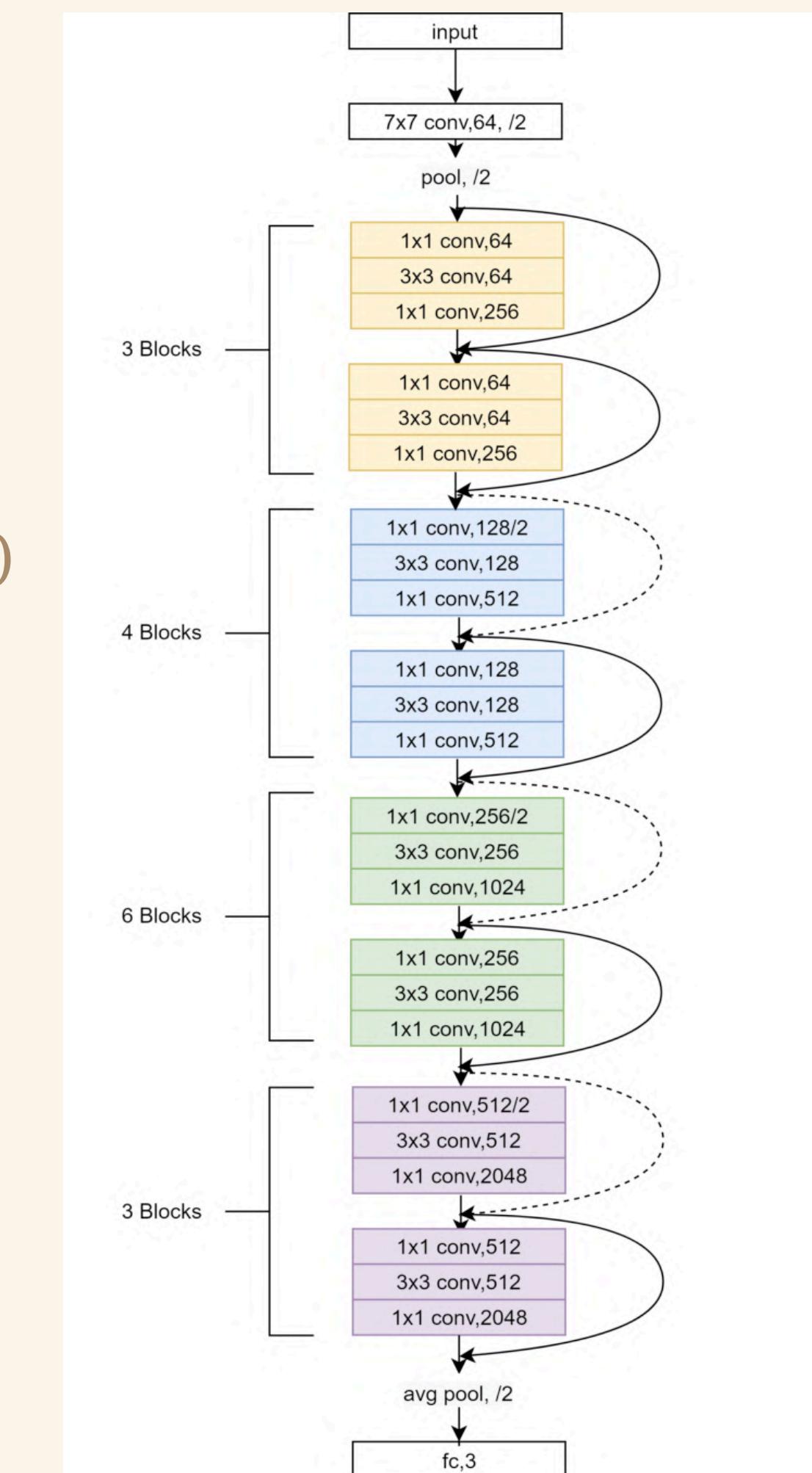


- Sur-apprentissage, meilleur modèle retenu à la 16ème epoch
- Classification correcte pour les classes dominantes

CNN Fine-Tuning

ResNet50

- Modèle plus profond
- Profondeur stabilisée par les Residual Block en exploitant les connections résiduelles (**apprentissage sur la différence entrée-sortie**)



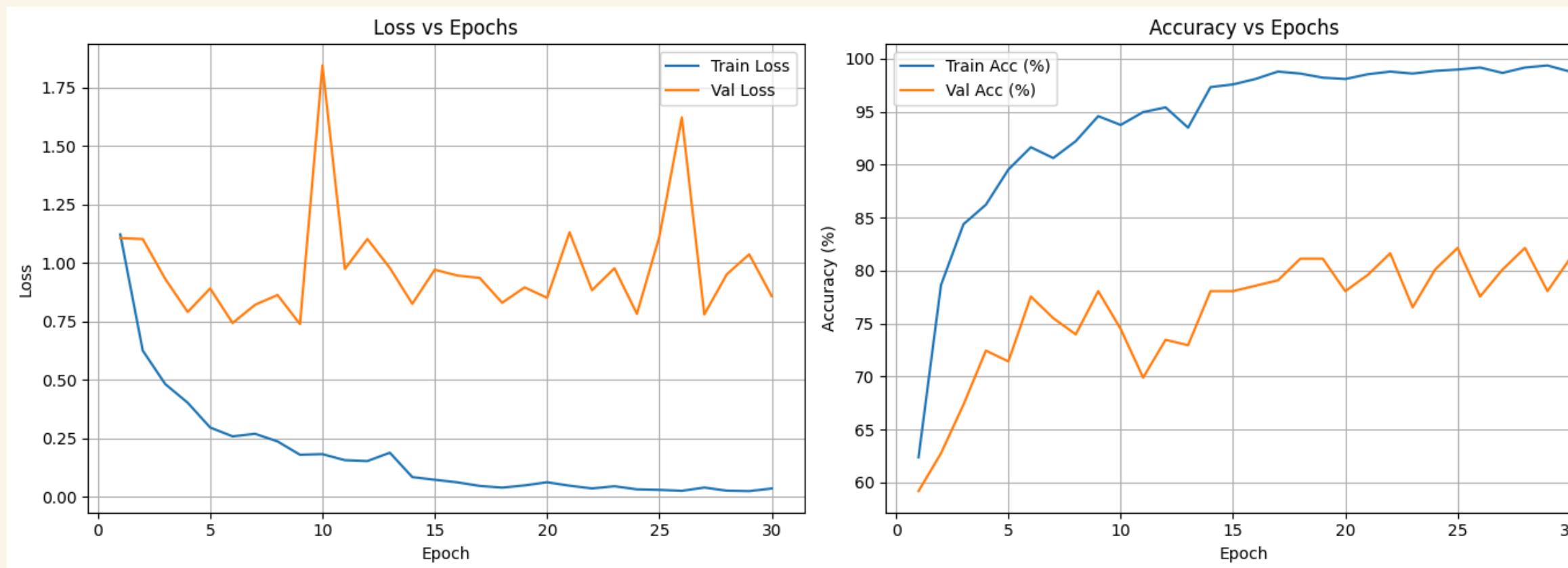
Architecture de ResNet50

CNN Fine-Tuning

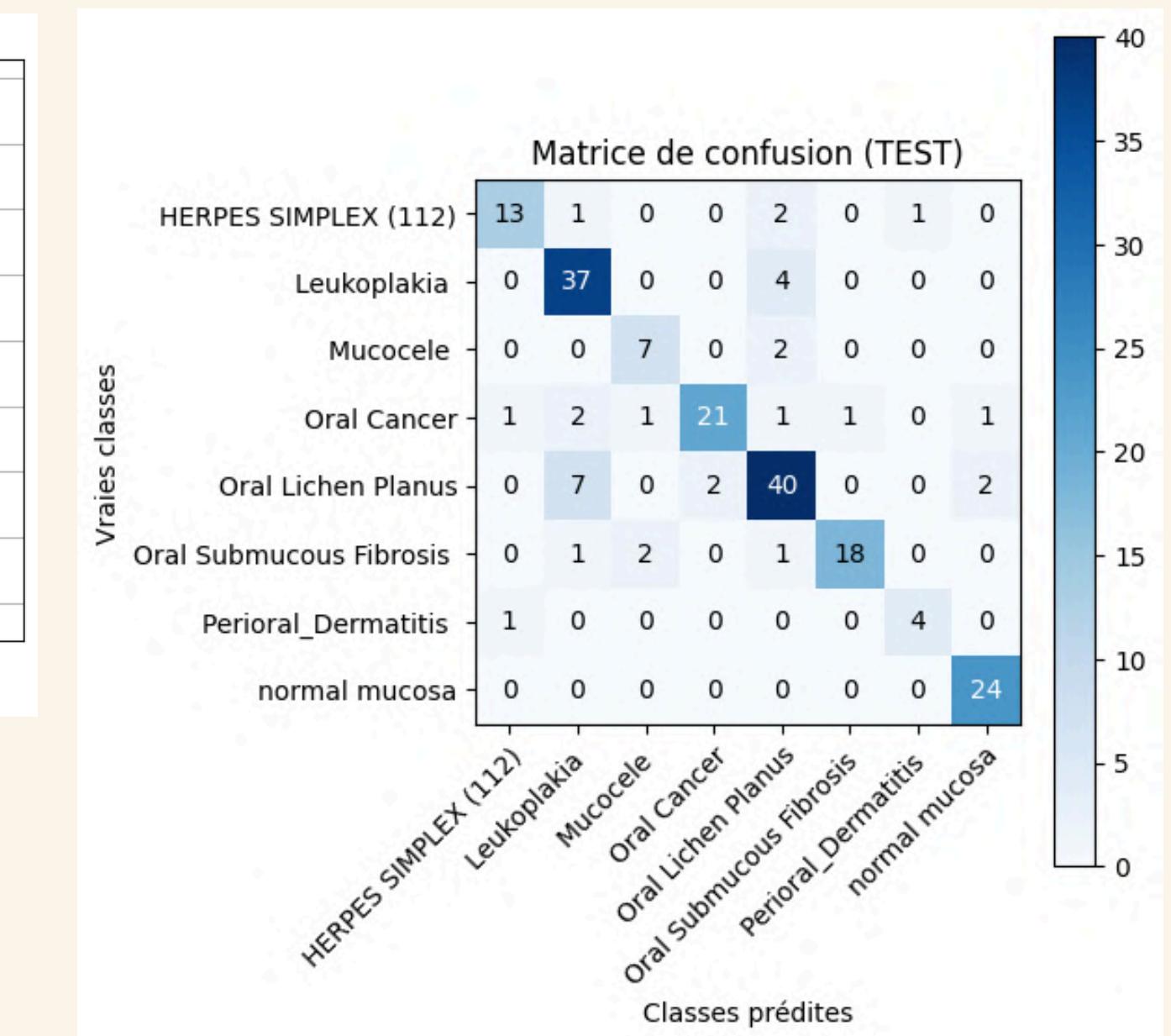
ResNet50

Fine-tuning complet avec early stop

Evolution des losses et accuracies



Global Accuracy Score sur Test : 83.25%

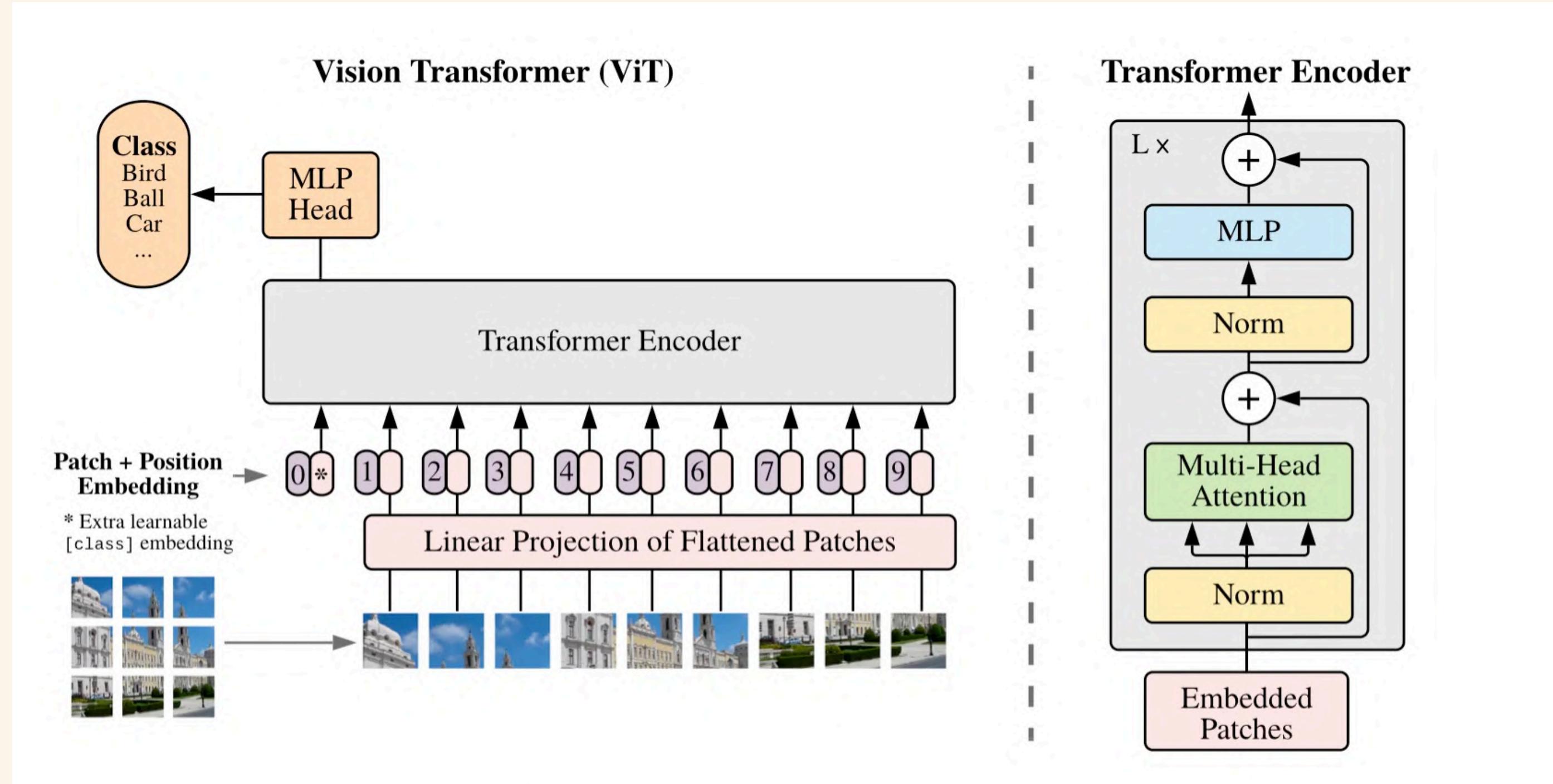


- Sur-apprentissage, meilleur modèle retenu à la 25ème epoch
- Classifications mauvaises pour les classes minoritaires

Vision Transformer (ViT)

Vision Transformer - Architecture

Introduit en 2020 par Google Research



Vision Transformer - Architecture

ViT : Un encoder de Transformer adapté à la classification d'images

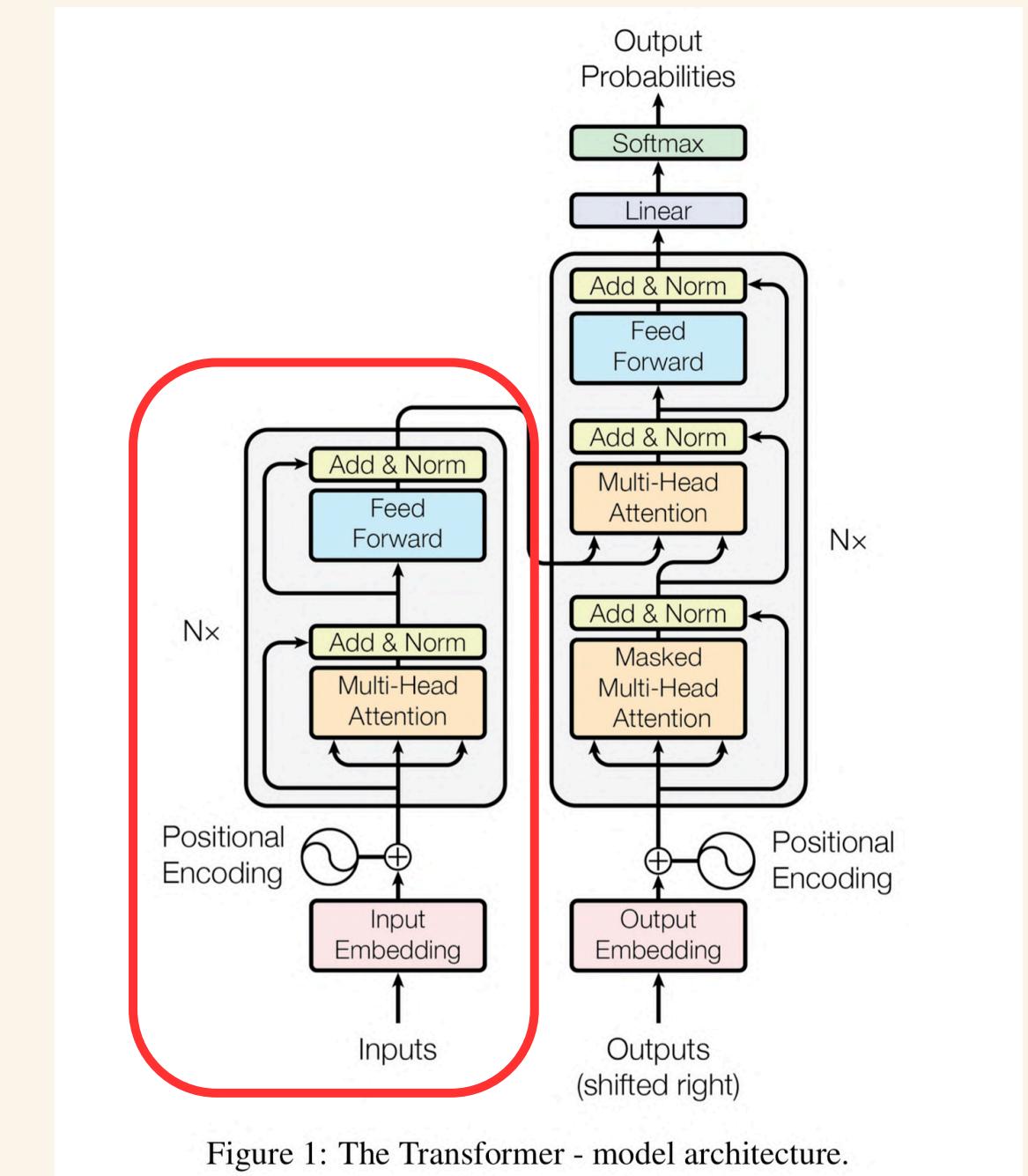
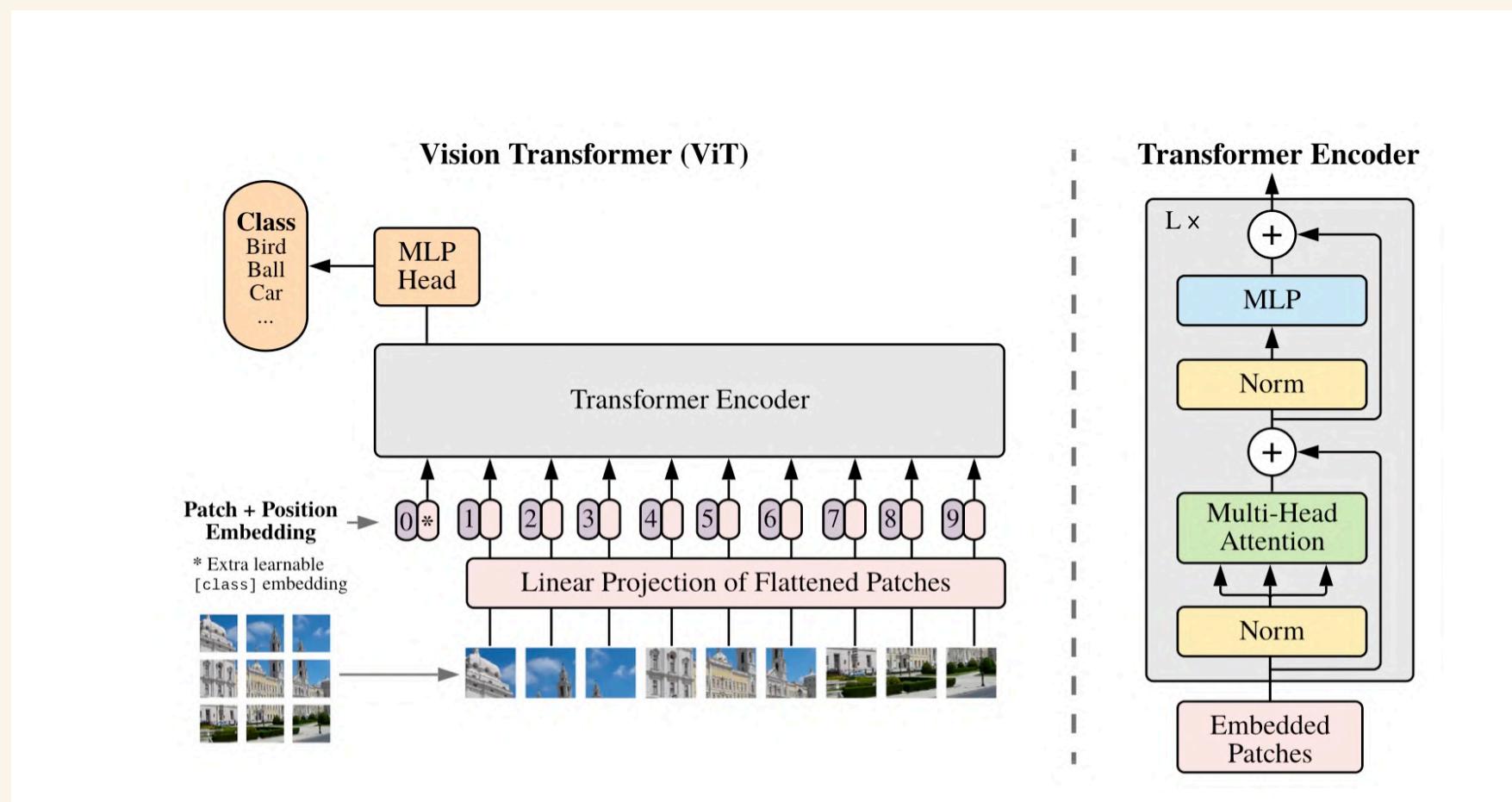
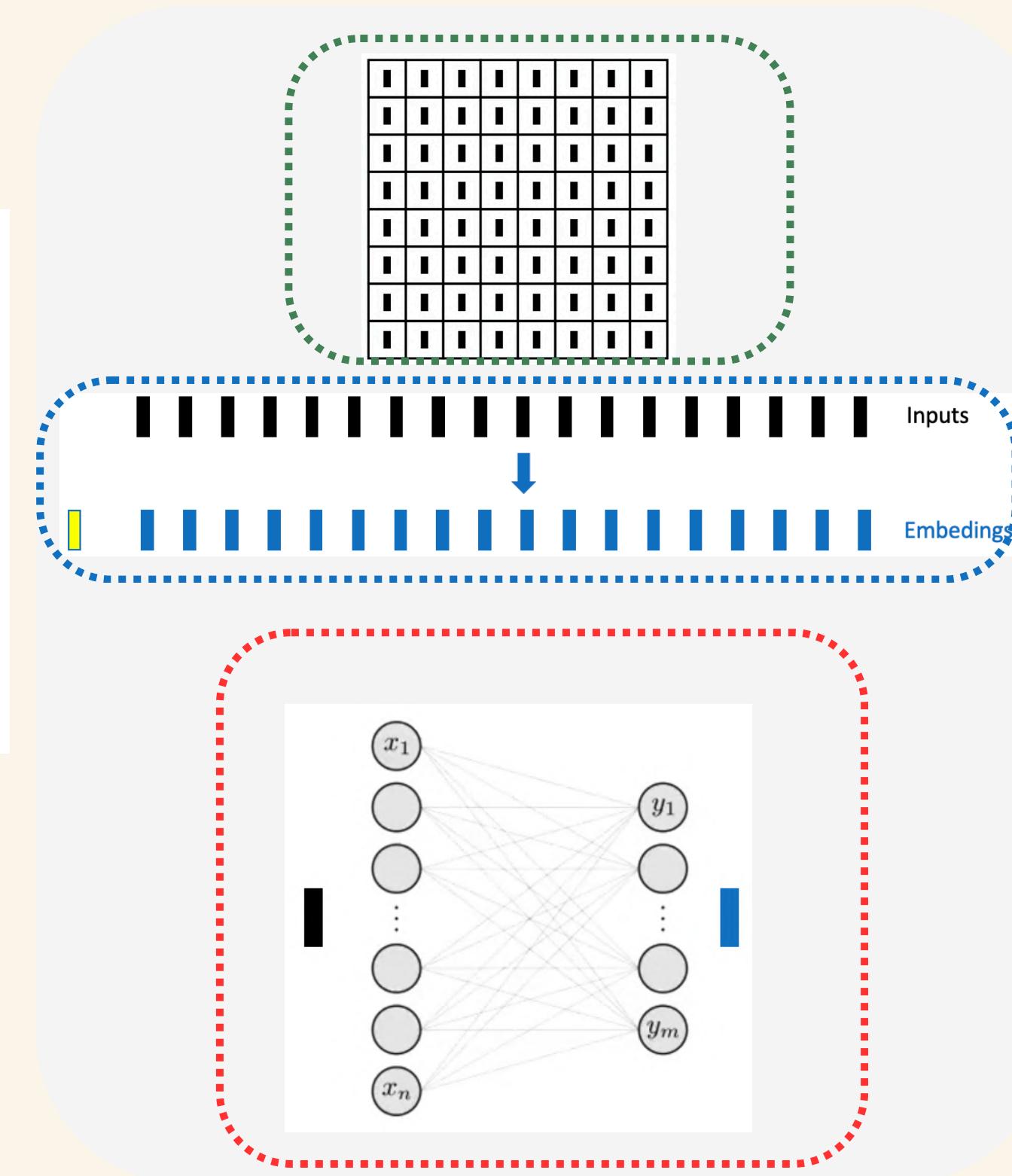
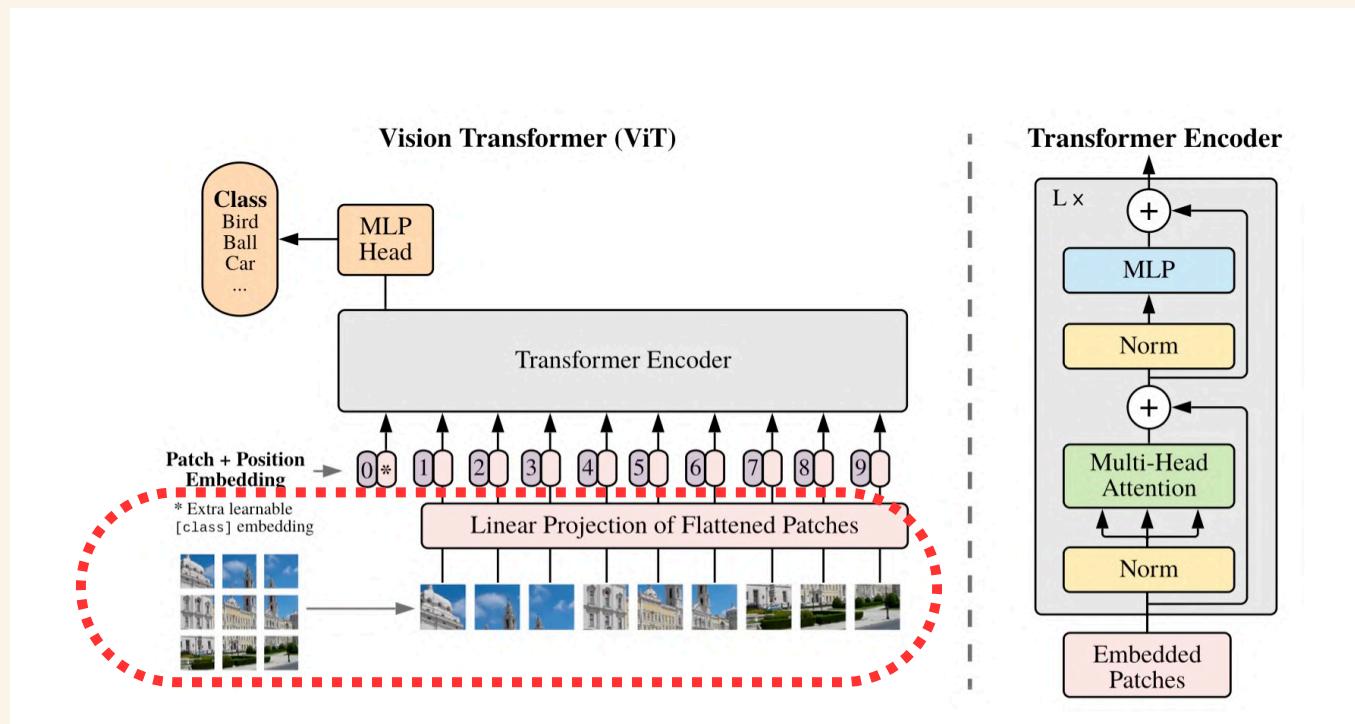


Figure 1: The Transformer - model architecture.

Vision Transformer - Architecture



Découpage de l'image en séquence de patchs

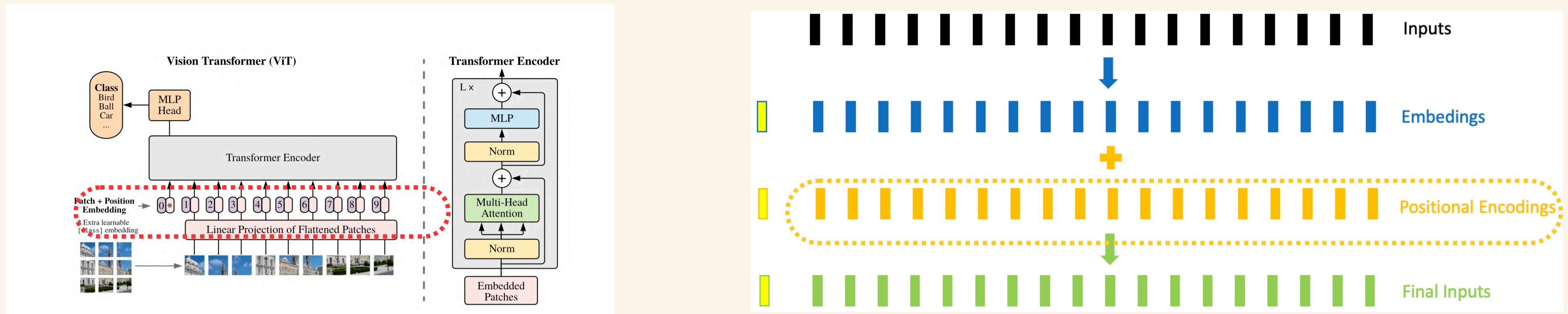
--> tokens = sous-images régulières mis en Flatten Vector

Ajout d'un token Class learnable capturant le contexte globale de l'image

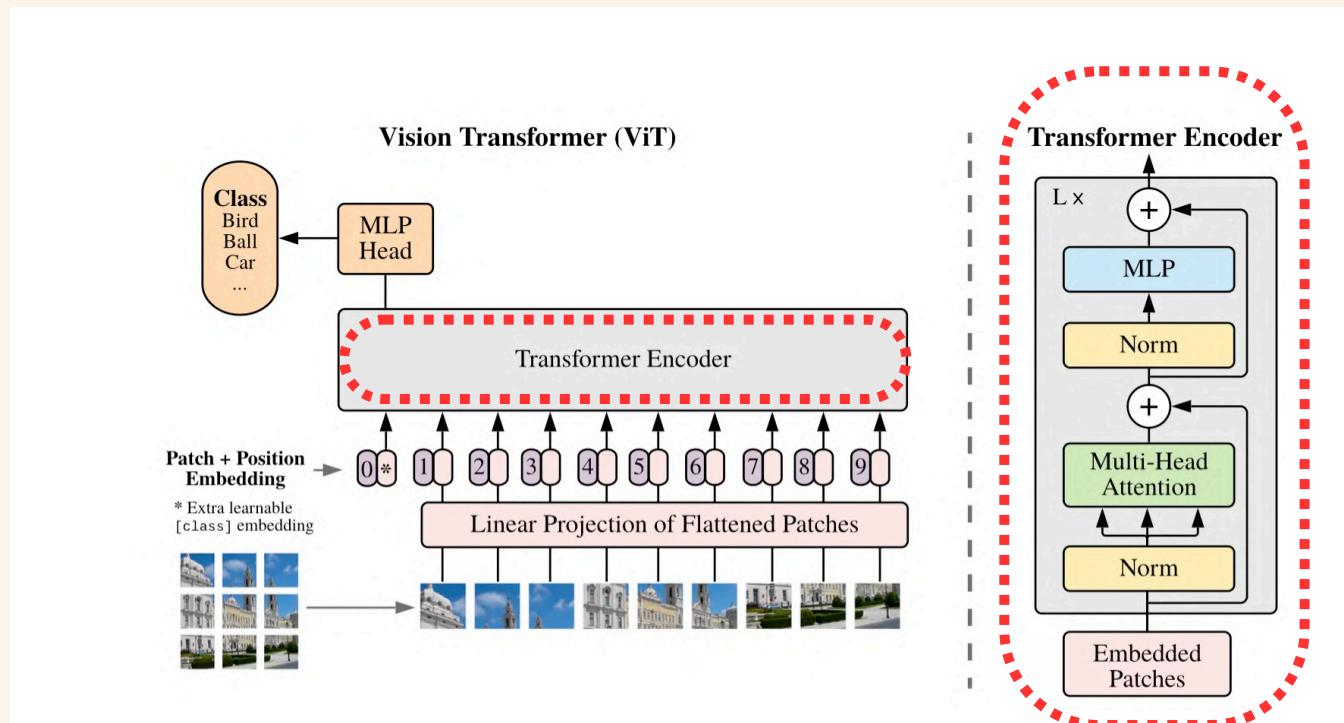
Embedding par passage dans une couche dense Linéaire

Vision Transformer - Architecture

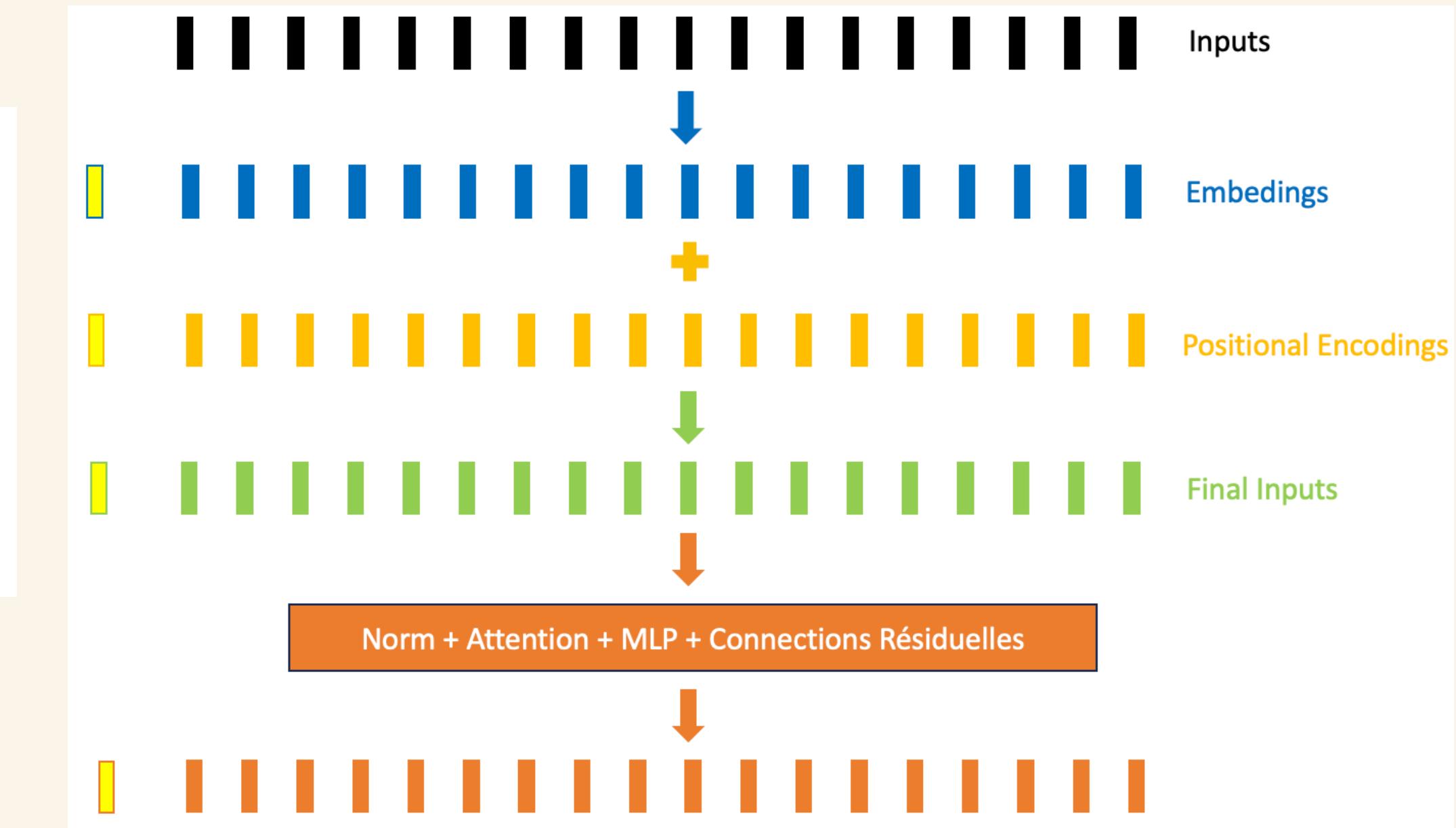
- Ajout de Postional Encodings pour injecter l'information spatiale des positions entre tokens



Vision Transformer - Architecture



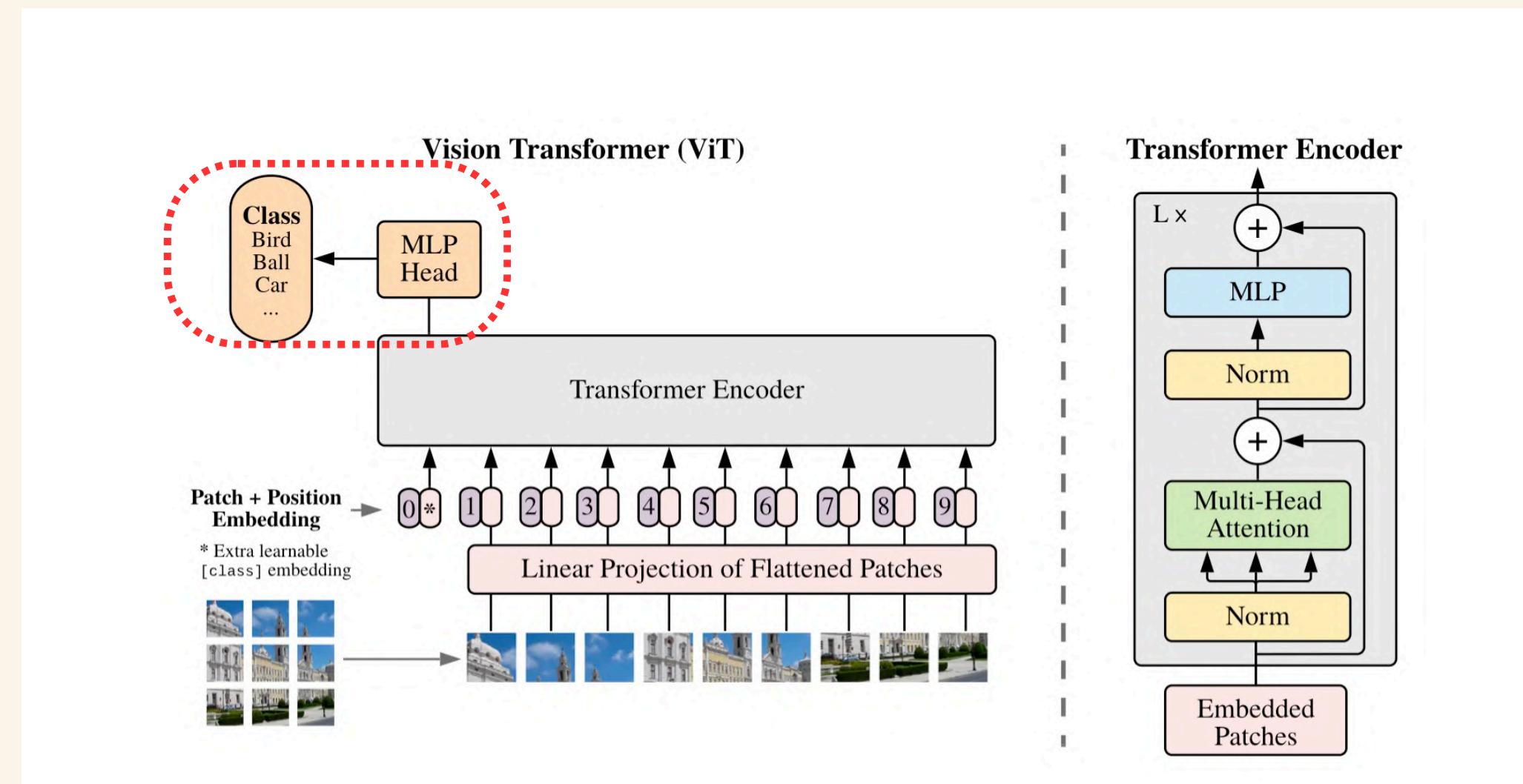
Le MHA permet d'incorporer les relations contextuelles
parmi les différents patchs en particulier pour le token Class



Normalisation pour stabiliser
et accélérer l'entraînement

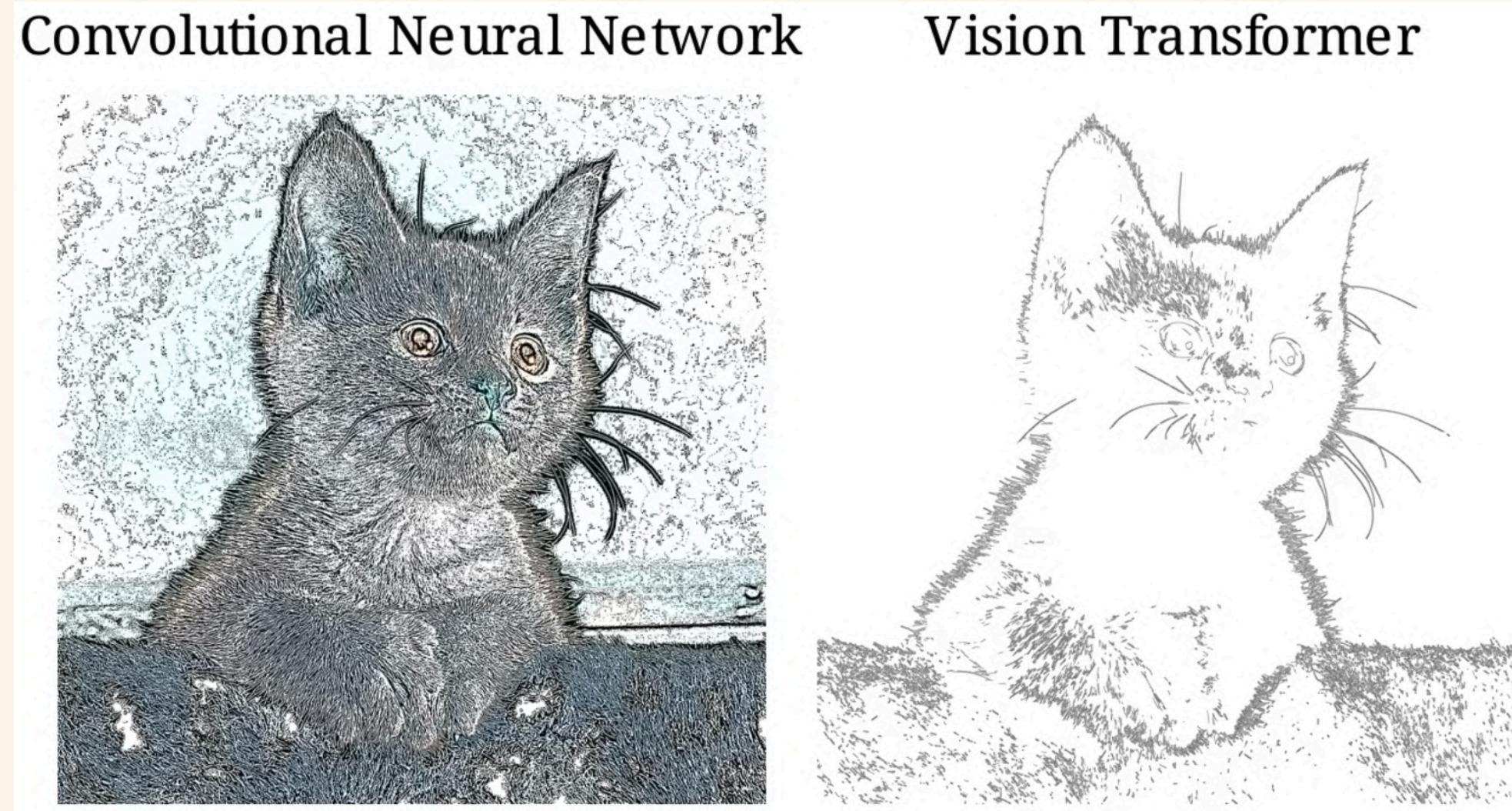
Vision Transformer - Architecture

- Classification par prédiction sur le vecteur final issu du token Class



Vision Transformer - Architecture

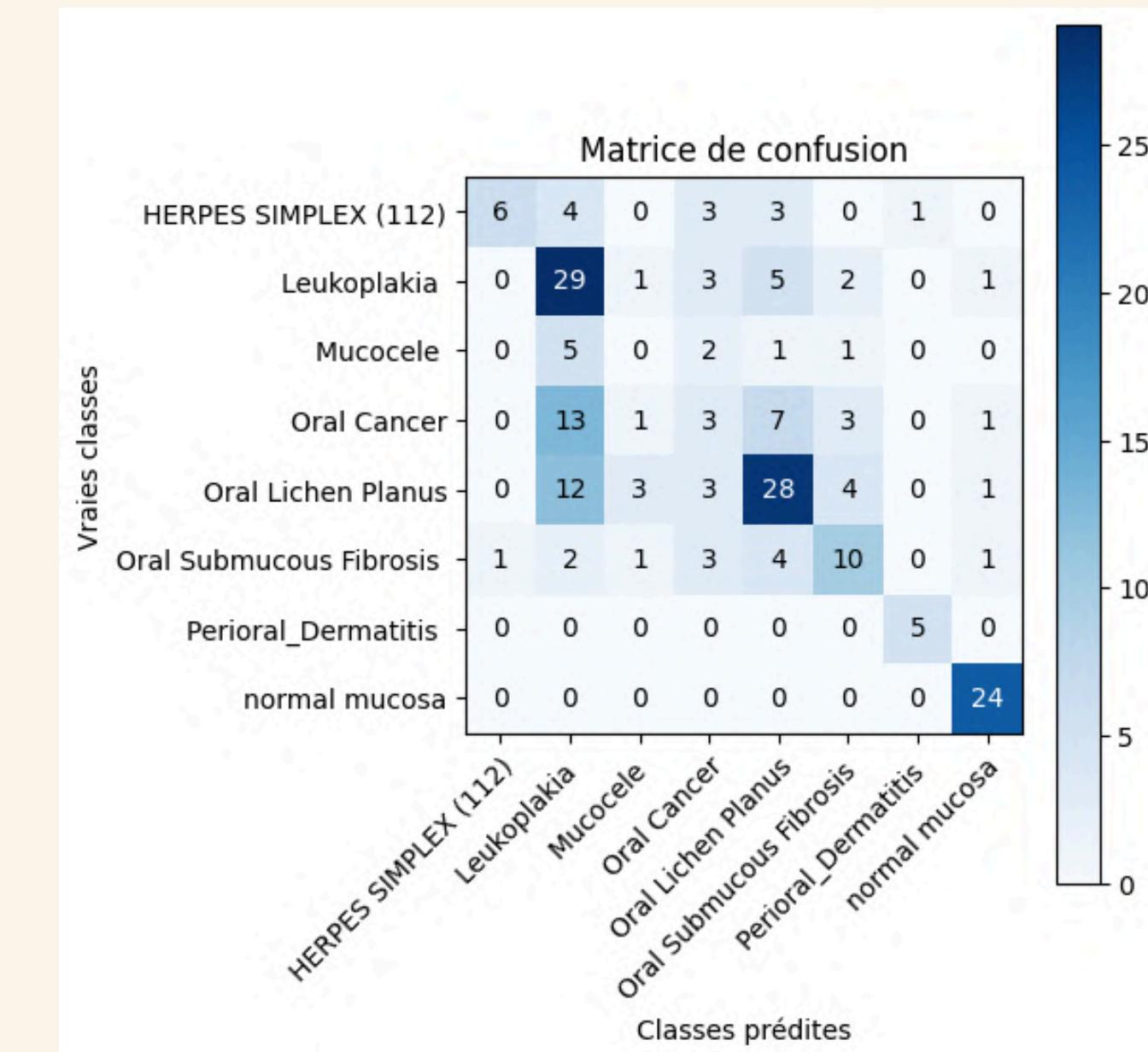
- 1) **CNN**: vision locale du contexte
- 2) **ViT(mécanisme d'attention)**: vision globale du contexte



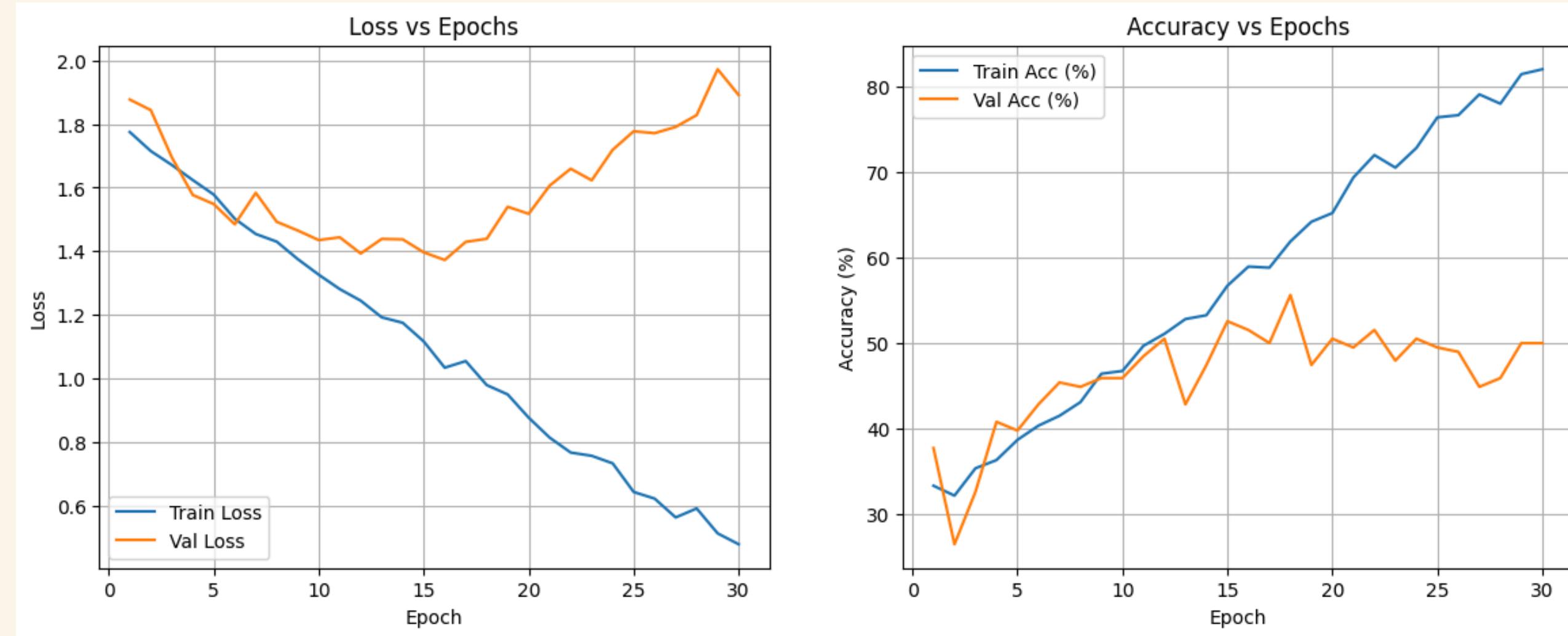
Vision Transformer - Résultats

--- Classification report ---				
	precision	recall	f1-score	support
HERPES SIMPLEX (112)	0.86	0.35	0.50	17
Leukoplakia	0.45	0.71	0.55	41
Mucocele	0.00	0.00	0.00	9
Oral Cancer	0.18	0.11	0.13	28
Oral Lichen Planus	0.58	0.55	0.57	51
Oral Submucous Fibrosis	0.50	0.45	0.48	22
Perioral_Dermatitis	0.83	1.00	0.91	5
normal mucosa	0.86	1.00	0.92	24
accuracy			0.53	197
macro avg	0.53	0.52	0.51	197
weighted avg	0.52	0.53	0.51	197

Test accuracy : 53.30%



Vision Transformer - Loss et Accuracy



Ce qui pourrait apporter de meilleurs résultats:

- On observe du sur-apprentissage -> utilisation de méthodes pour réduire le **sur-apprentissage**
- On remarque que la loss n'a pas fini de converger -> on peut **augmenter le nombre d'epoch**, cependant on aura un entraînement très long (ici 49 min déjà)
- Le nombre de données est trop faible malgré la data augmentation -> utilisation du **transfert learning**

Vision Transformer - Résolution du sur-apprentissage

Problème : sur-apprentissage

Comment le résoudre?

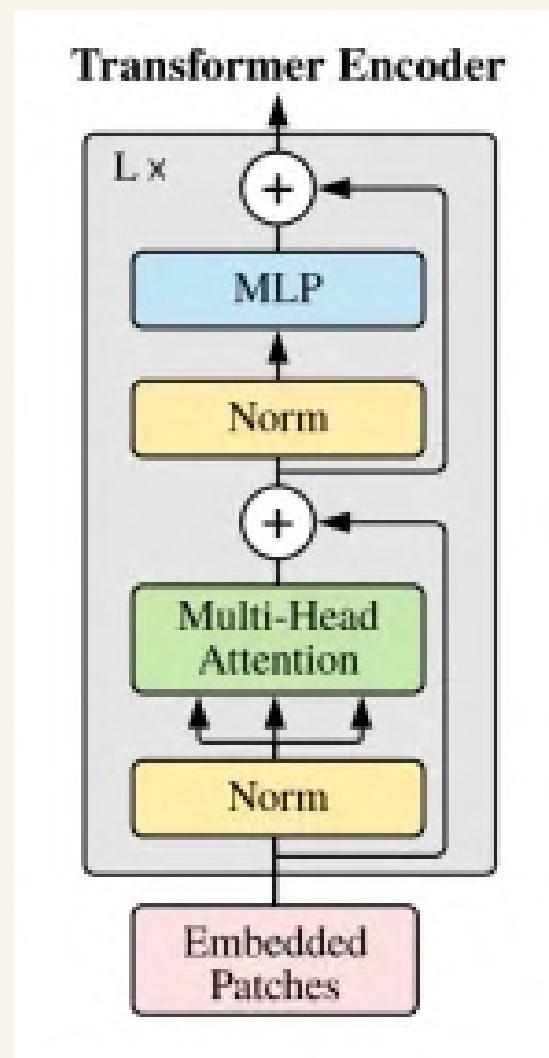
- Plus grande Data augmentation → pas possible dans notre cas
- Dropout plus fort : 0.25
- Utilisation de DropPath:0.1

Qu'est ce que le DropPath ?

Drop Path (aussi appelé Stochastic Depth) est une technique de régularisation utilisée surtout dans les Transformers et les ResNets profonds.

→ On supprime aléatoirement des chemins entiers du réseau, c'est-à-dire des blocs complets, au lieu de supprimer des neurones individuellement (comme le dropout).

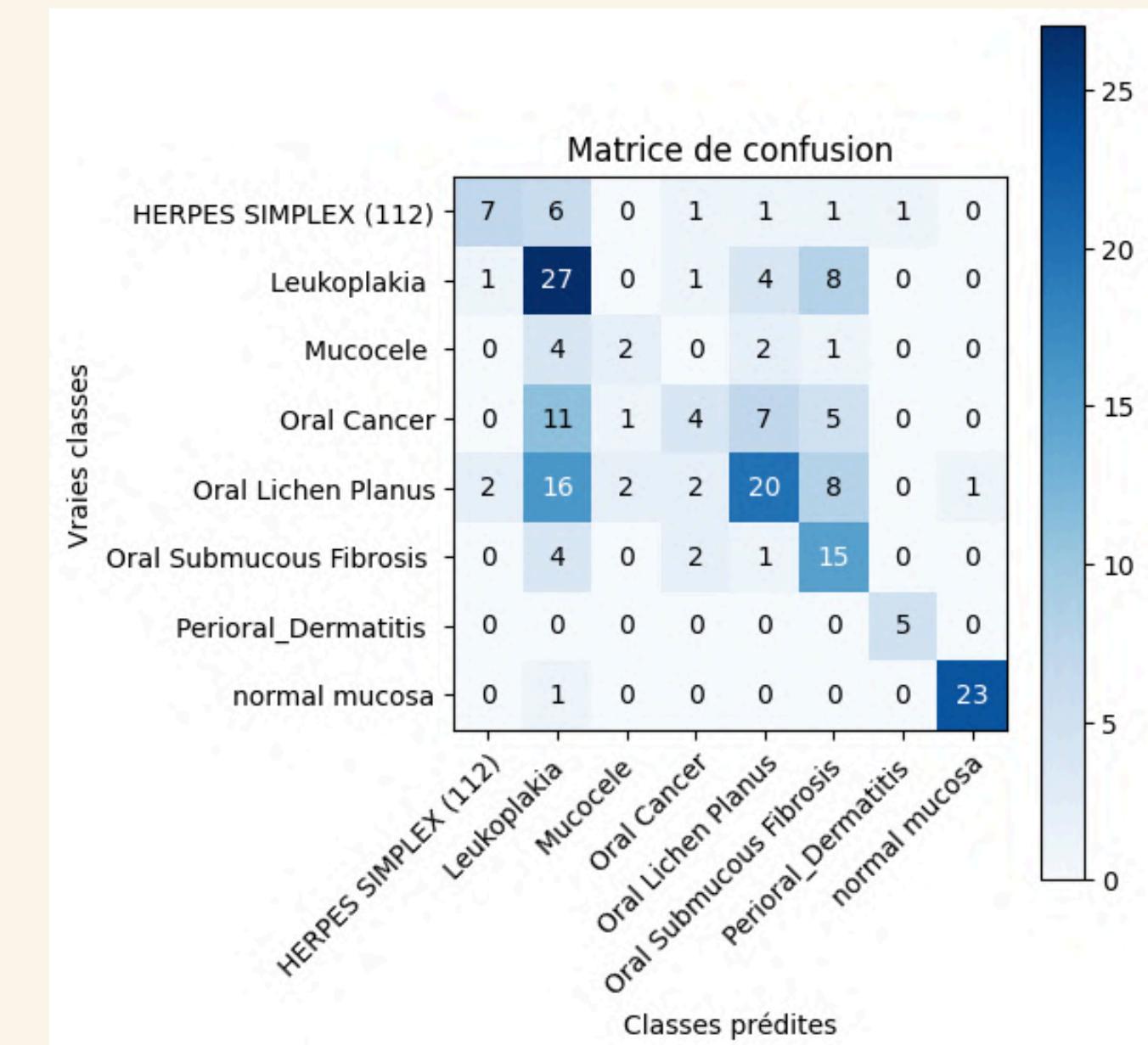
DropPath → avec une certaine probabilité p (0.1 standard), on ignore complètement le résultat et on ne prend que x



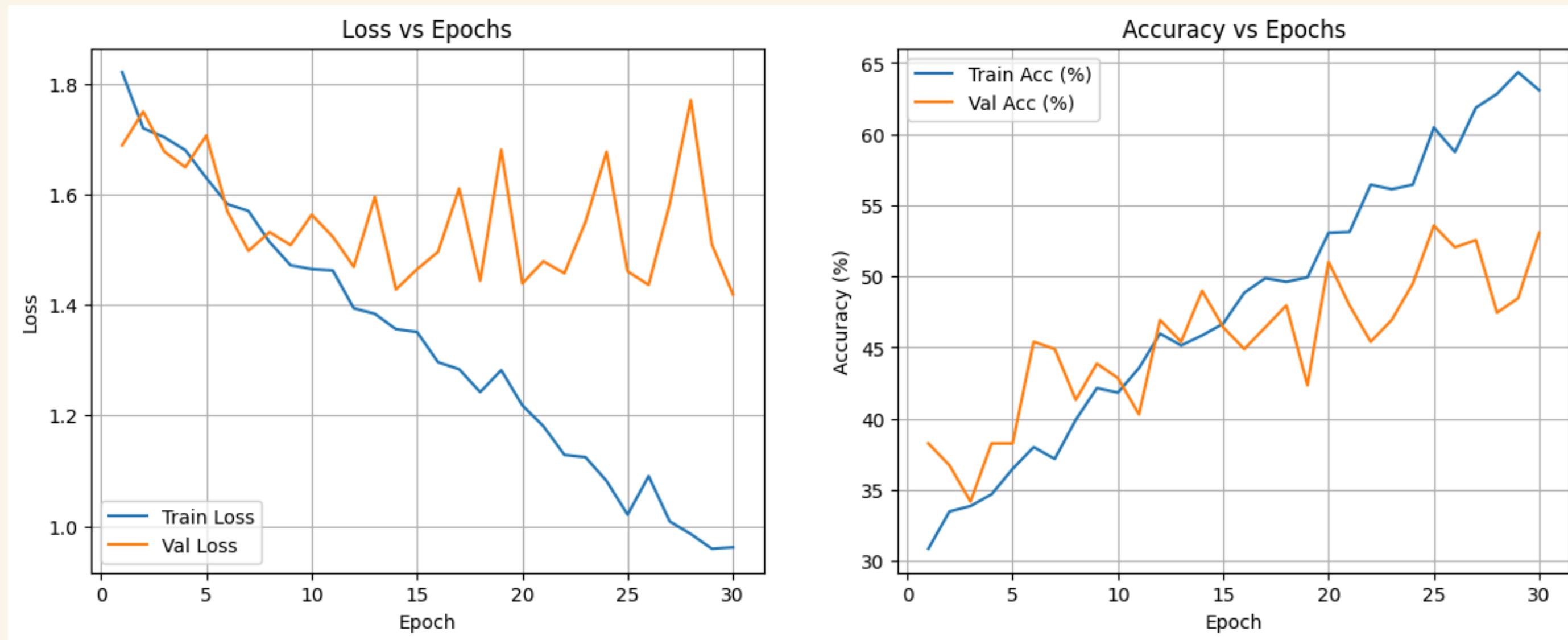
Vision Transformer - Résultats

--- Classification report ---				
	precision	recall	f1-score	support
HERPES SIMPLEX (112)	0.70	0.41	0.52	17
Leukoplakia	0.39	0.66	0.49	41
Mucocele	0.40	0.22	0.29	9
Oral Cancer	0.40	0.14	0.21	28
Oral Lichen Planus	0.57	0.39	0.47	51
Oral Submucous Fibrosis	0.39	0.68	0.50	22
Perioral_Dermatitis	0.83	1.00	0.91	5
normal mucosa	0.96	0.96	0.96	24
accuracy			0.52	197
macro avg	0.58	0.56	0.54	197
weighted avg	0.55	0.52	0.51	197

Test accuracy : 52.28%



Vision Transformer - Loss et Accuracy



Sur-apprentissage encore visible + loss non convergente
→ Problème potentiel : Base de données trop petite
On va donc utiliser l'idée du transfert learning

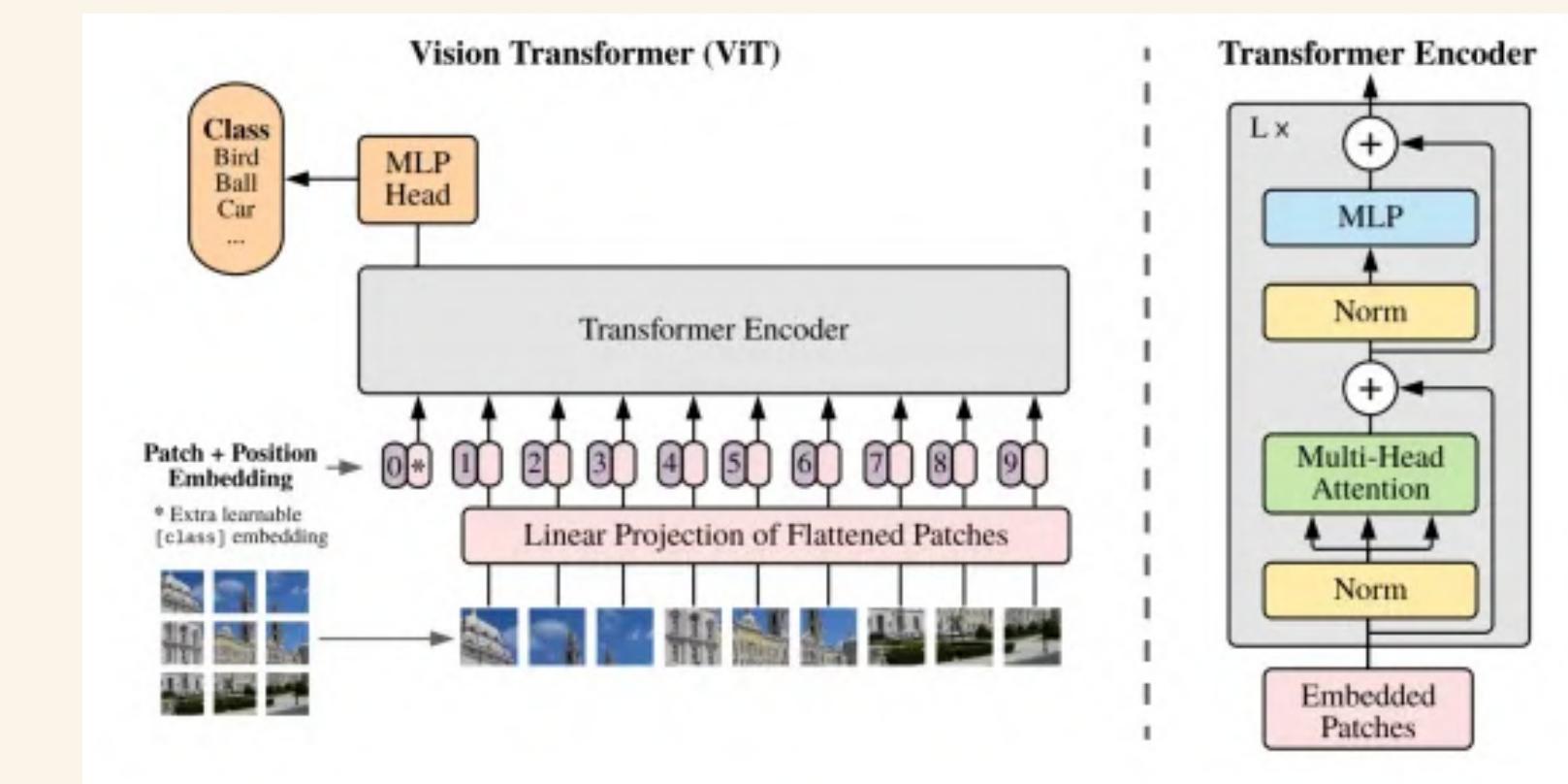
Vision Transformer avec Transfert Learning

- **Vision transformer sur MedMNIST**

- Utilisation de la base de données MedMNIST
- Entrainer notre modèle avec la base de données MedMNIST

- **Transfert Learning sur notre base de données médicales**

- Enlever la head et la remplacer
- Freeze les paramètres liés au modèle entraîné précédemment
- Utilisation d'un learning rate plus important
- Entrainer le modèle sur notre base de données médicales

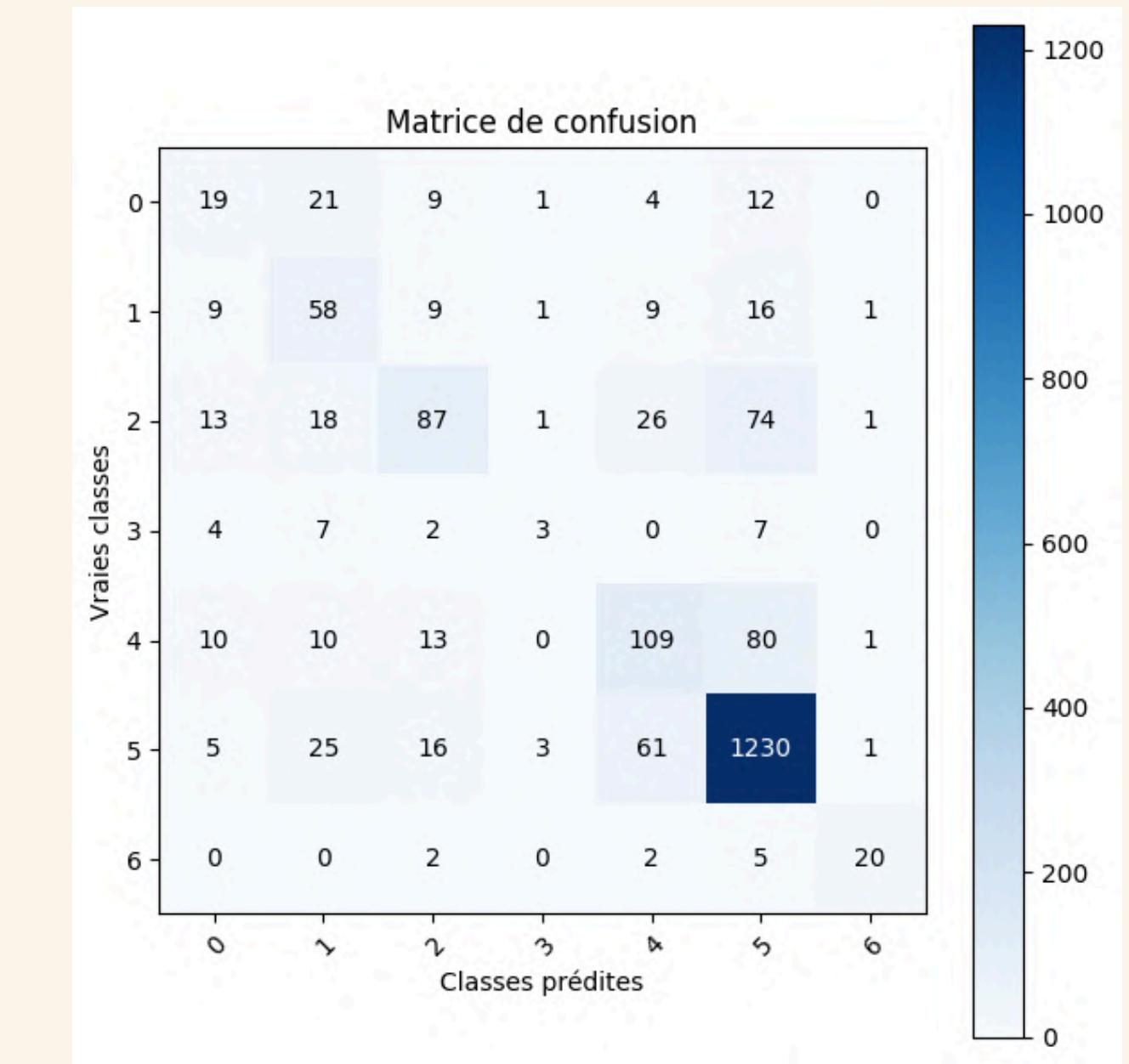


Vision Transformer avec Transfert Learning

- Résultats avec MedMNIST

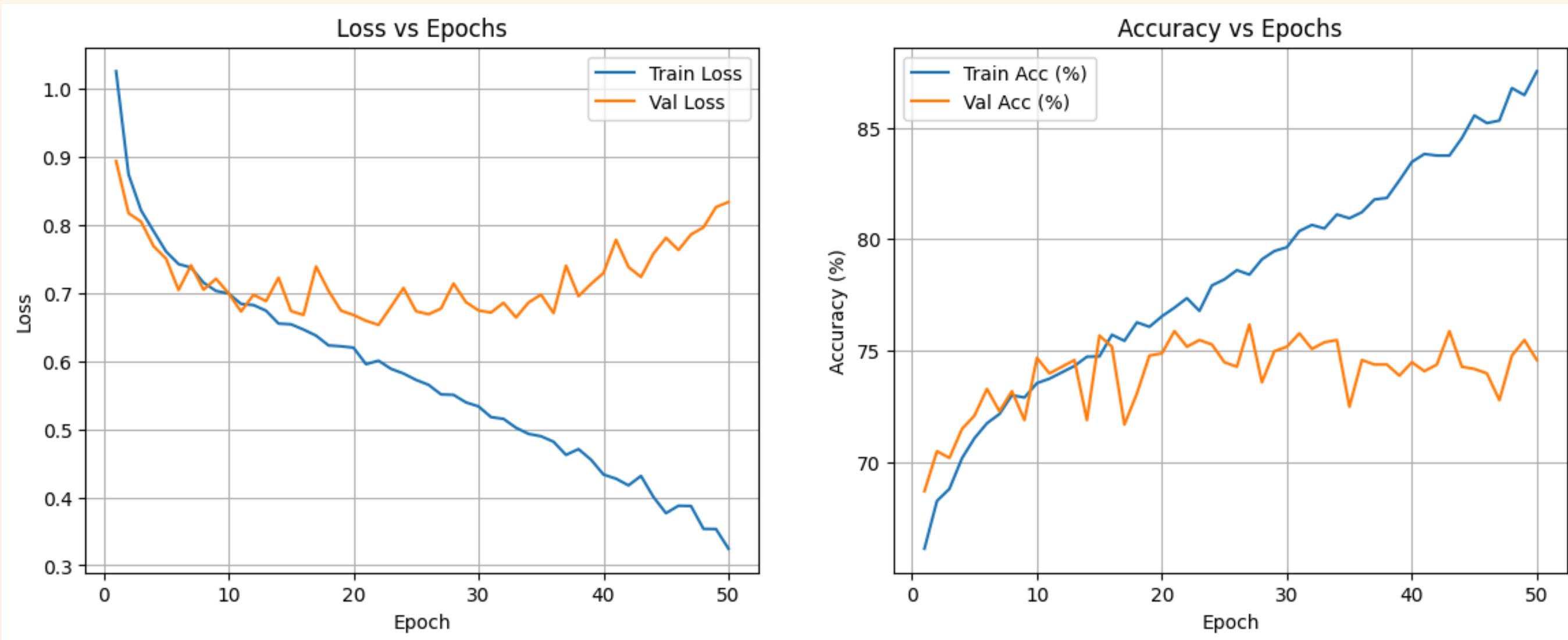
--- Classification report ---				
	precision	recall	f1-score	support
0	0.32	0.29	0.30	66
1	0.42	0.56	0.48	103
2	0.63	0.40	0.49	220
3	0.33	0.13	0.19	23
4	0.52	0.49	0.50	223
5	0.86	0.92	0.89	1341
6	0.83	0.69	0.75	29
accuracy			0.76	2005
macro avg	0.56	0.50	0.51	2005
weighted avg	0.75	0.76	0.75	2005

Test accuracy : 76.11%



Vision Transformer avec Transfert Learning

- Loss et Accuracy avec MedMNIST



Remarques :

- l'exécution est beaucoup plus rapide (ici 21min pour 50 epoch).
- la loss n'a pas convergé → il faut augmenter le nombre d'epoch.
- la présence de sur-apprentissage.

Vision Transformer avec Transfert Learning

On observe du **sur-apprentissage**. Pour pallier à ce problème:

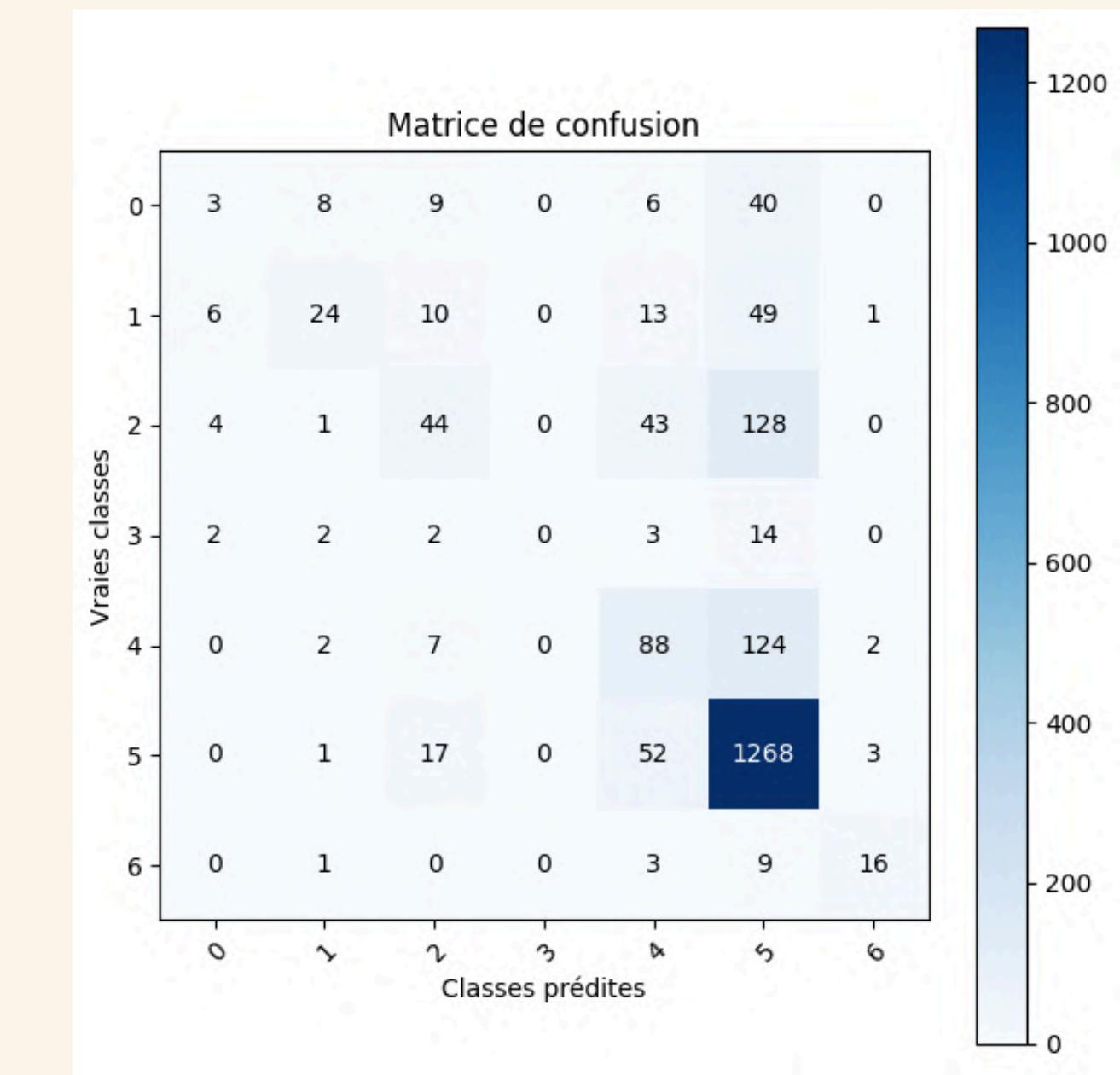
- Plus grande **Data augmentation**
- **Dropout plus fort** : 0.3
- Utilisation de **DropPath**: 0.1
- **weight_decay** = 0.1 → contrôle les poids trop grand
- Réduire la taille du ViT avec embed_dim=128, depth=4, dropout=0.3 et mlp_ratio=2

Vision Transformer avec Transfert Learning

- Résultats avec MedMNIST

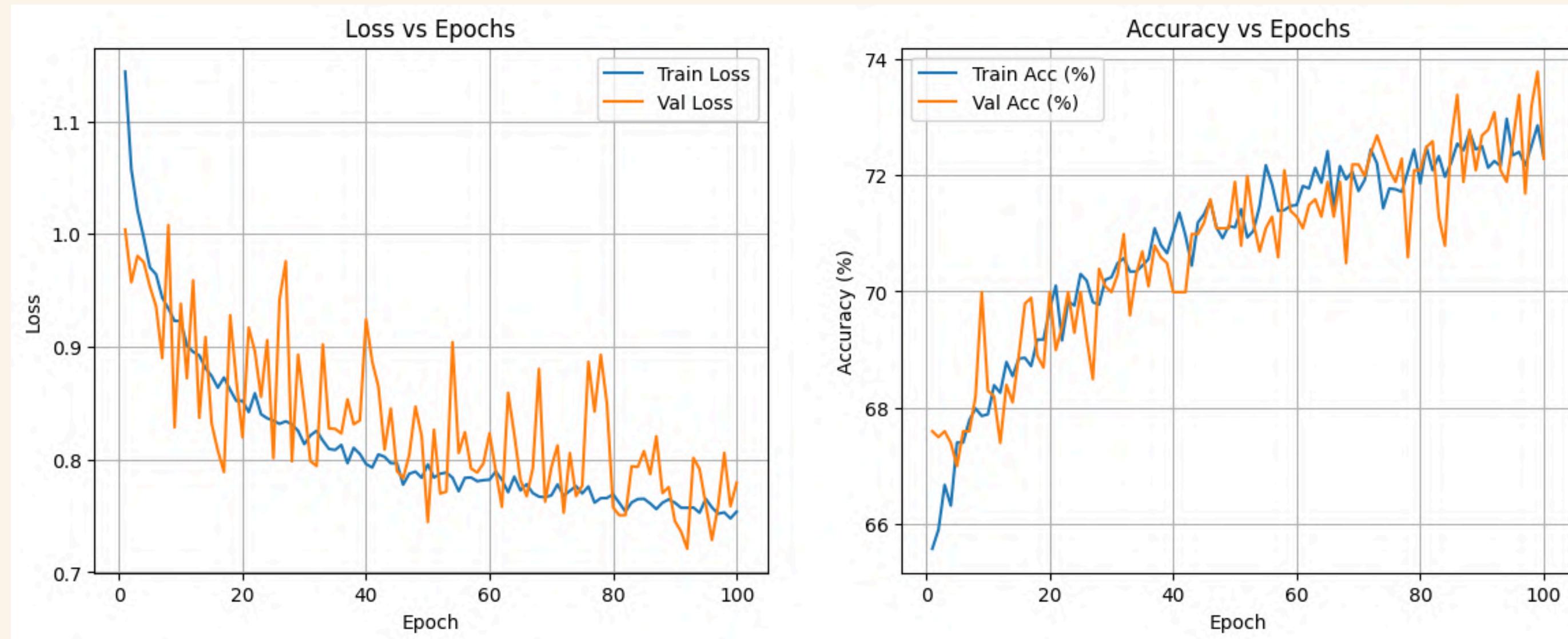
--- Classification report ---				
	precision	recall	f1-score	support
0	0.20	0.05	0.07	66
1	0.62	0.23	0.34	103
2	0.49	0.20	0.28	220
3	0.00	0.00	0.00	23
4	0.42	0.39	0.41	223
5	0.78	0.95	0.85	1341
6	0.73	0.55	0.63	29
accuracy			0.72	2005
macro avg	0.46	0.34	0.37	2005
weighted avg	0.67	0.72	0.68	2005

Test accuracy : 71.97%



Vision Transformer avec Transfert Learning

- Loss et Accuracy avec MedMNIST



On a réduit le sur-apprentissage

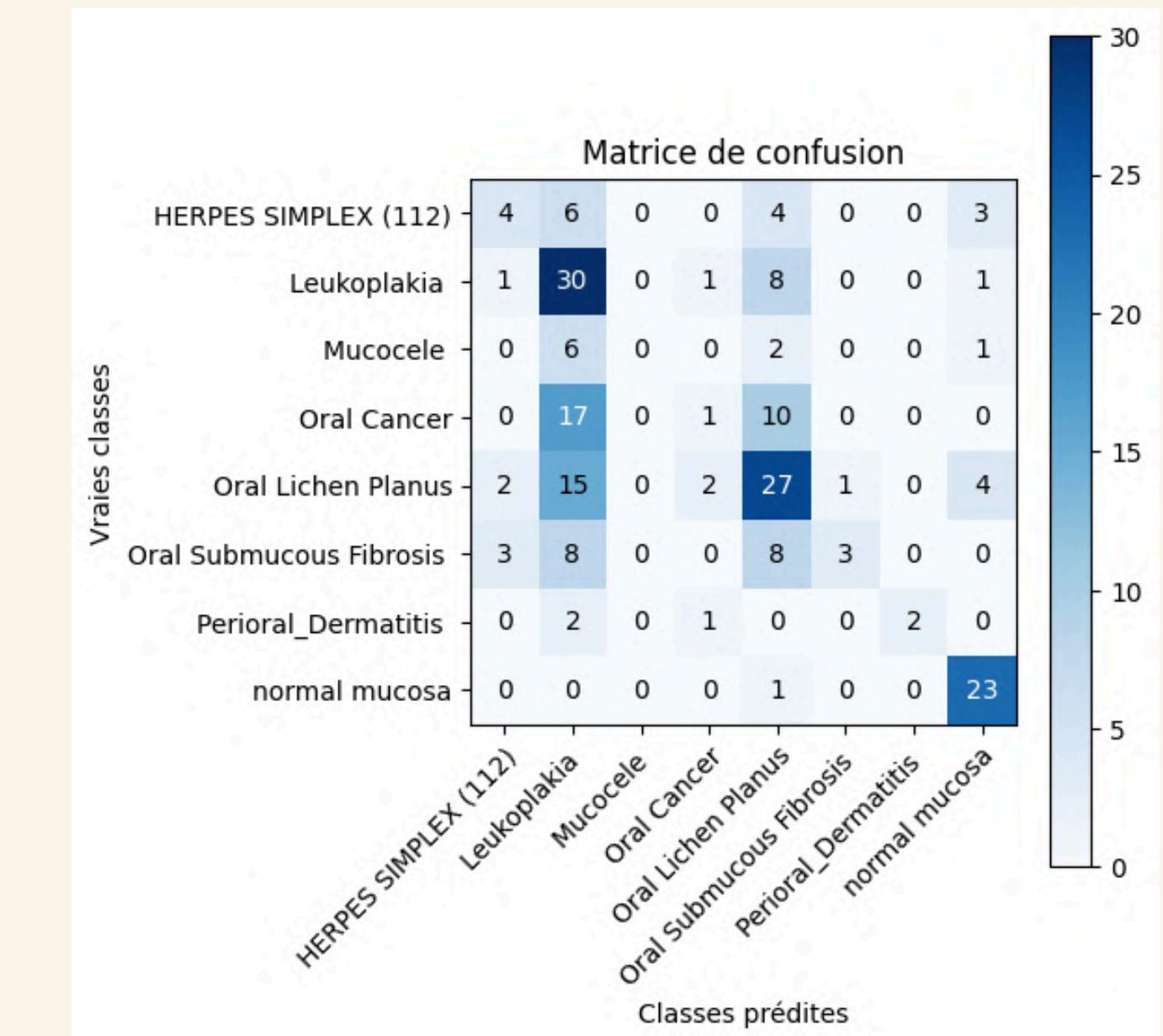
→ On peut passer à l'entraînement sur nos données médicales

Vision Transformer avec Transfert Learning

- Résultats

--- Classification report ---				
	precision	recall	f1-score	support
HERPES SIMPLEX (112)	0.40	0.24	0.30	17
Leukoplakia	0.36	0.73	0.48	41
Mucocele	0.00	0.00	0.00	9
Oral Cancer	0.20	0.04	0.06	28
Oral Lichen Planus	0.45	0.53	0.49	51
Oral Submucous Fibrosis	0.75	0.14	0.23	22
Perioral_Dermatitis	1.00	0.40	0.57	5
normal mucosa	0.72	0.96	0.82	24
accuracy			0.46	197
macro avg	0.48	0.38	0.37	197
weighted avg	0.45	0.46	0.40	197

Test accuracy : 45.69%



Vision Transformer avec Transfert Learning

- Loss et Accuracy

Epoch [1/10] - Train loss: 1.7996, acc: 32.46% - Val loss: 1.7474, acc: 42.86%

>> Nouveau meilleur modèle sauvegardé.

Epoch [2/10] - Train loss: 1.6756, acc: 36.54% - Val loss: 1.6498, acc: 40.82%

Epoch [3/10] - Train loss: 1.6311, acc: 37.12% - Val loss: 1.6581, acc: 36.73%

Epoch [4/10] - Train loss: 1.6052, acc: 37.50% - Val loss: 1.6670, acc: 40.31%

Epoch [5/10] - Train loss: 1.5948, acc: 39.41% - Val loss: 1.6410, acc: 36.22%

Epoch [6/10] - Train loss: 1.6010, acc: 38.58% - Val loss: 1.6566, acc: 36.73%

Epoch [7/10] - Train loss: 1.5631, acc: 39.35% - Val loss: 1.6127, acc: 39.80%

Epoch [8/10] - Train loss: 1.5770, acc: 38.58% - Val loss: 1.6082, acc: 42.86%

Epoch [9/10] - Train loss: 1.5687, acc: 38.84% - Val loss: 1.6251, acc: 44.90%

>> Nouveau meilleur modèle sauvegardé.

Epoch [10/10] - Train loss: 1.5603, acc: 39.86% - Val loss: 1.6147, acc: 46.43%

>> Nouveau meilleur modèle sauvegardé. Meilleure val acc: 46.43%

On observe que pour 10 epochs, la loss n'a pas encore convergé. Puis on pourrait améliorer:

- Augmenter le nombre d'epoch
- Trouver une base de données plus adaptée que MedMNIST

Performances ViT - CNN

Récapitulatif des accuracies	CNN	ViT
Architecture basique	53.94%	53.30%
Solution envisagée	<ul style="list-style-type: none">• Pretext Task Colorization : 47.94%• Inpainting : 45.36%• Masked AE : 46.91%	Transfert Learning 45.69%

- Le mécanisme d'attention permet certes au ViT d'être plus efficace en saisant le contexte global. Néanmoins, ViT nécessite une très grande base de données d'apprentissage.
- Pour une petite base de données comme la notre, il reste préférable de fine-tune un modèle avancé de CNN comme ResNet50 (83% d'accuracy) contre un modèle par transformers comme ViT B16 (48% d'accuracy).

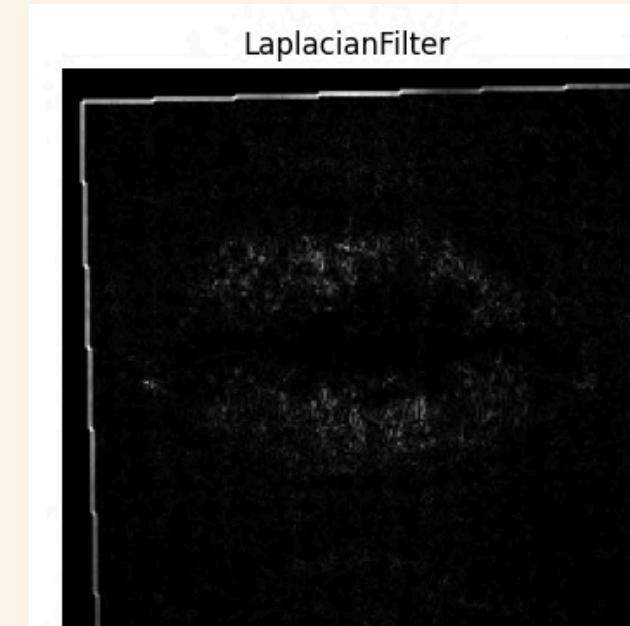
Remarques générales d'amélioration des performances

- Sensibilité à la data augmentation (par exemple: ajout du filtre Laplacien)

Accuracies à 20 epochs	CNN	ViT
sans Laplacien	45%	51%
avec Laplacien	49%	12%



Filtrage Laplacien



LaplacianFilter

- Retrait des classes minoritaires pour obtenir de meilleures performances
- Trouver une base de données plus adaptée que MedMNIST pour le Transfert Learning



*Merci pour
votre attention*

