

DÉTECTION DE TUMEURS CÉRÉBRALES

CNN

Elaboré par: Sirine Ben Fadhel

Encadré par : Rym Sessi



Introduction

- **Intérêt médical**

Grâce à une analyse automatisée, on peut améliorer la précision du diagnostic, mais surtout, intervenir plus tôt pour sauver des vies.

- **Méthodes classiques vs intelligence artificielle**

Une nouvelle ère dans l'analyse des images médicales.

- **Approche**

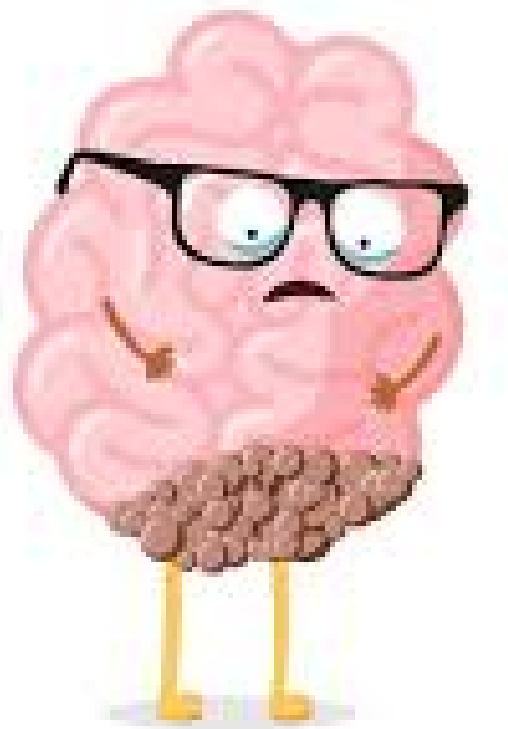
Comparaison entre modèles SVM, CNN et LeNet-5.

Ainsi que channel_attention, spatial attention et
CBAM



Objectif

Notre objectif principal de ce projet est de construire des modèles CNN capable de classer si un sujet présente ou non une tumeur à partir des image d'IRM.





Plan :

1.Présentation de la base de données

2.Description et prétraitement des données

3.Bibliothèques utilisées et Définition des métriques

4.Modèle 1 – SVM

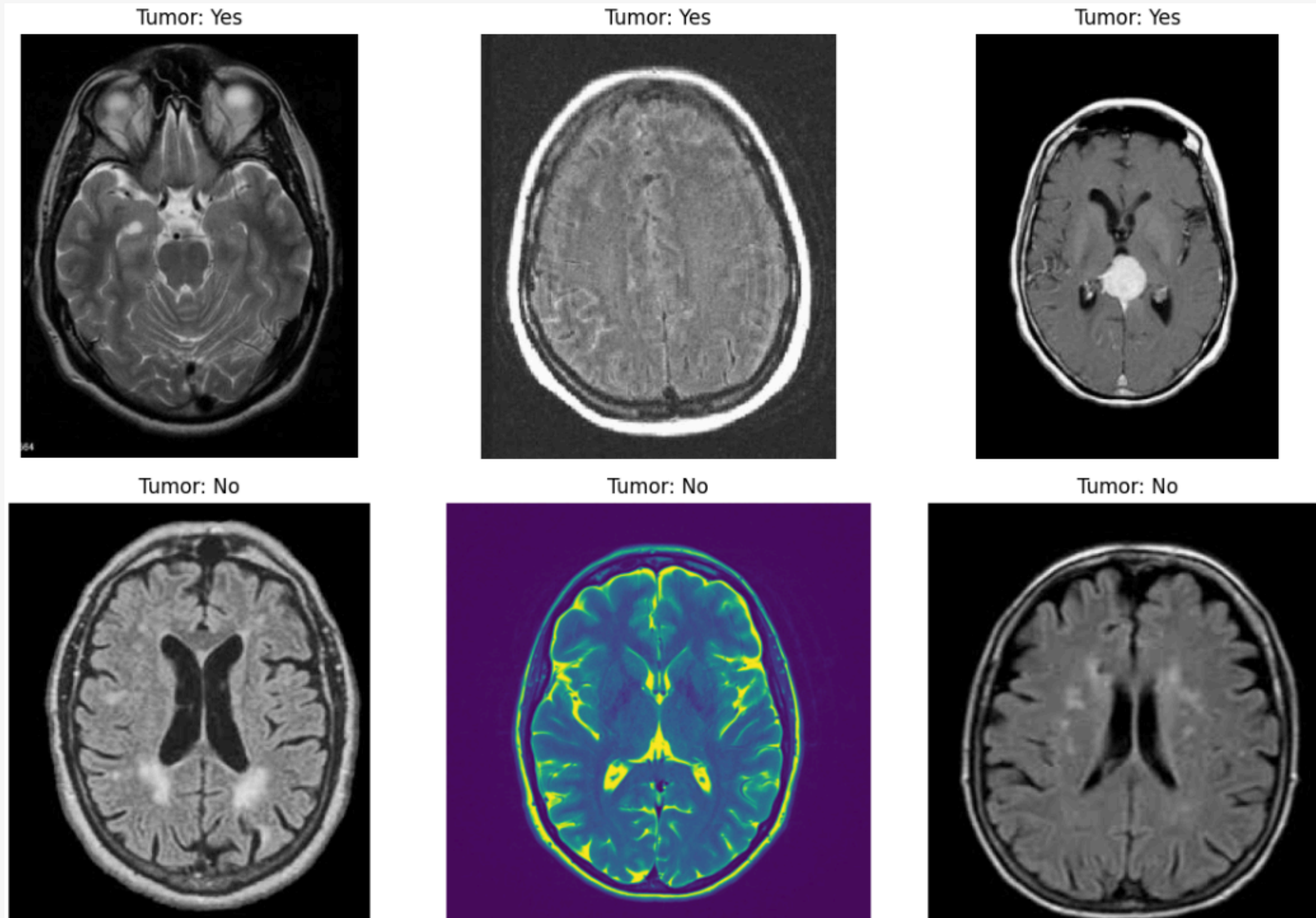
5.Modèle 2 – CNN personnalisé

6.Modèle 3 – LeNet-5

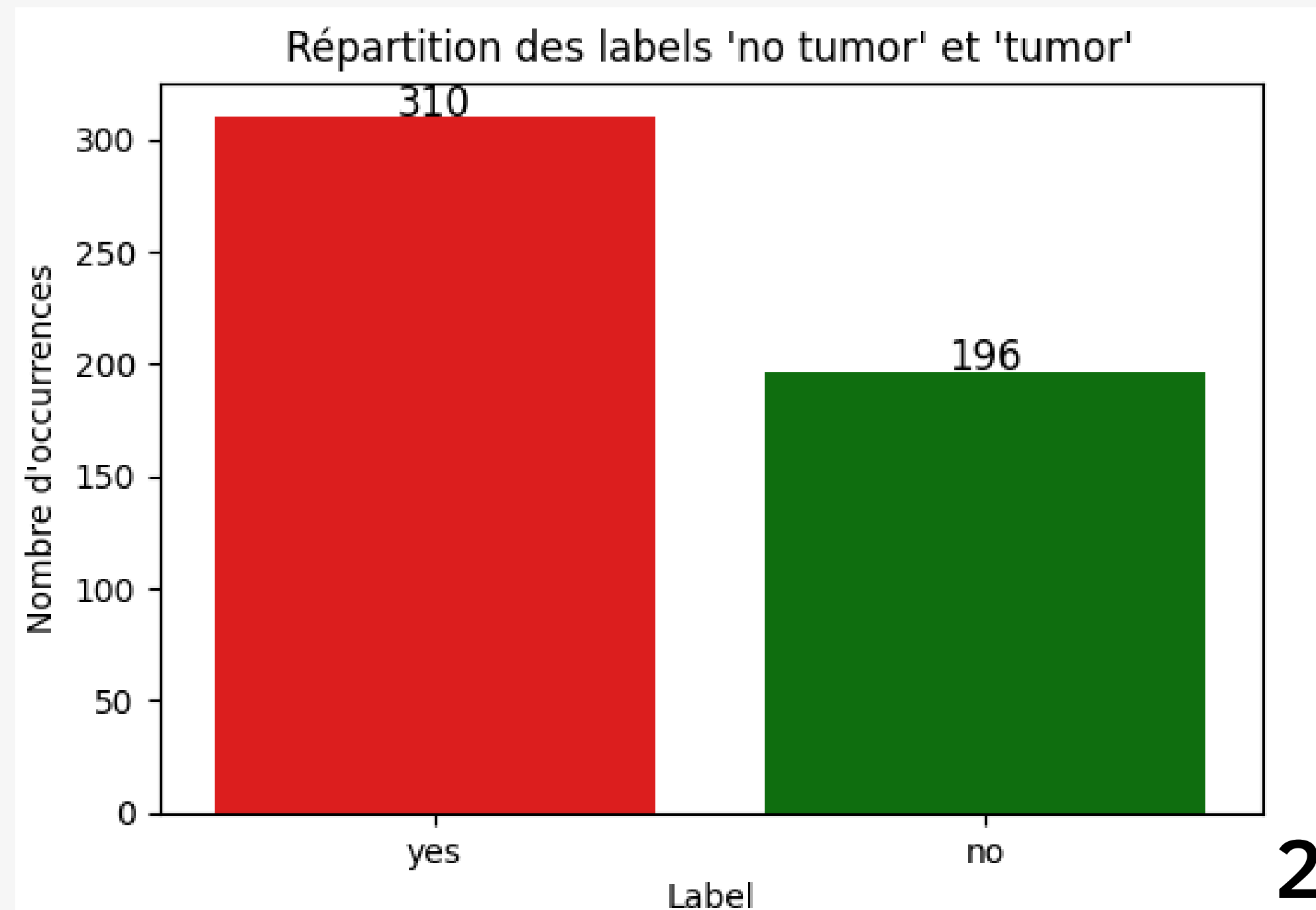
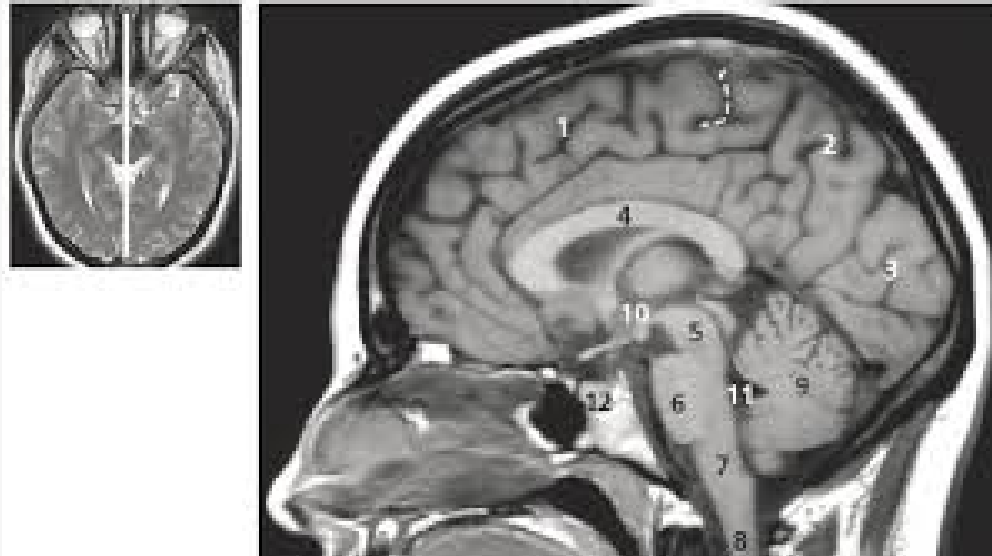
7.Comparaison des performances

8.Amélioration du modèle avec les modules d'attention

Présentation de la base de données



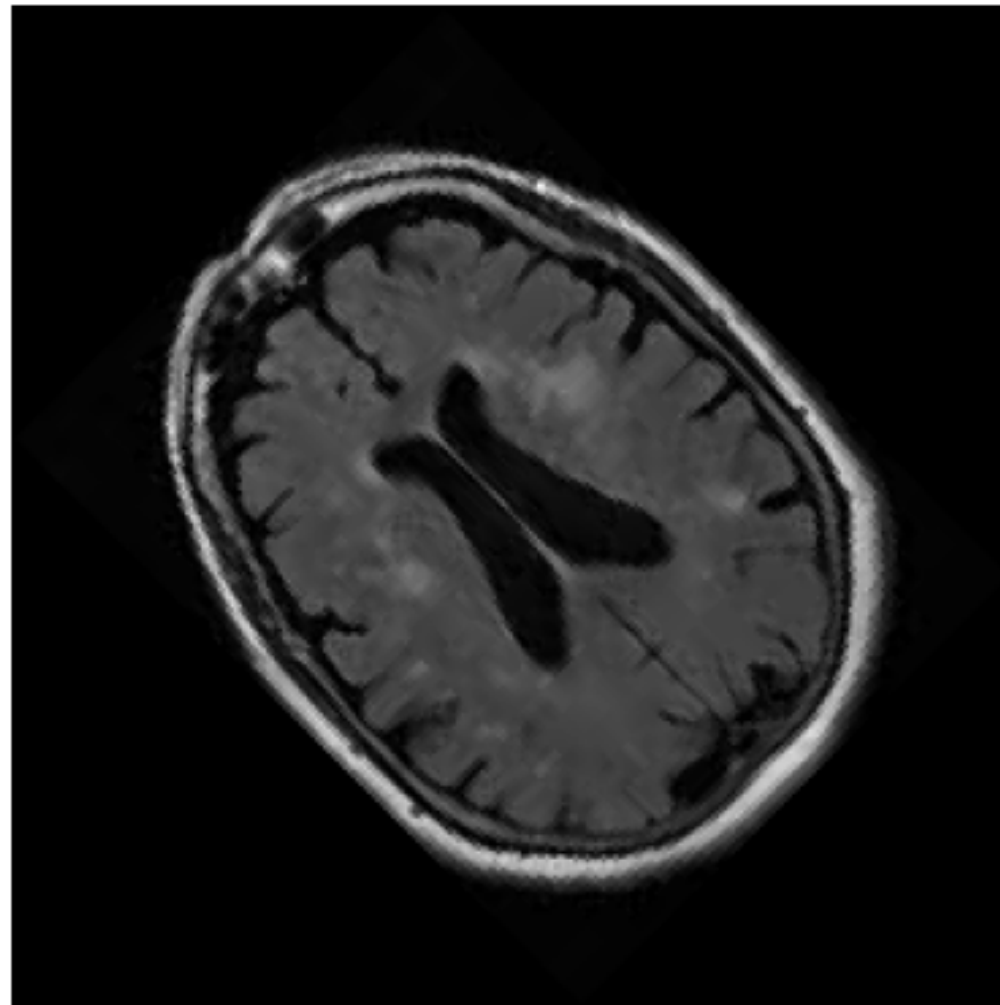
```
Nombre de classes : 2
Nombre d'éléments par classe
label
yes      310
no       196
```



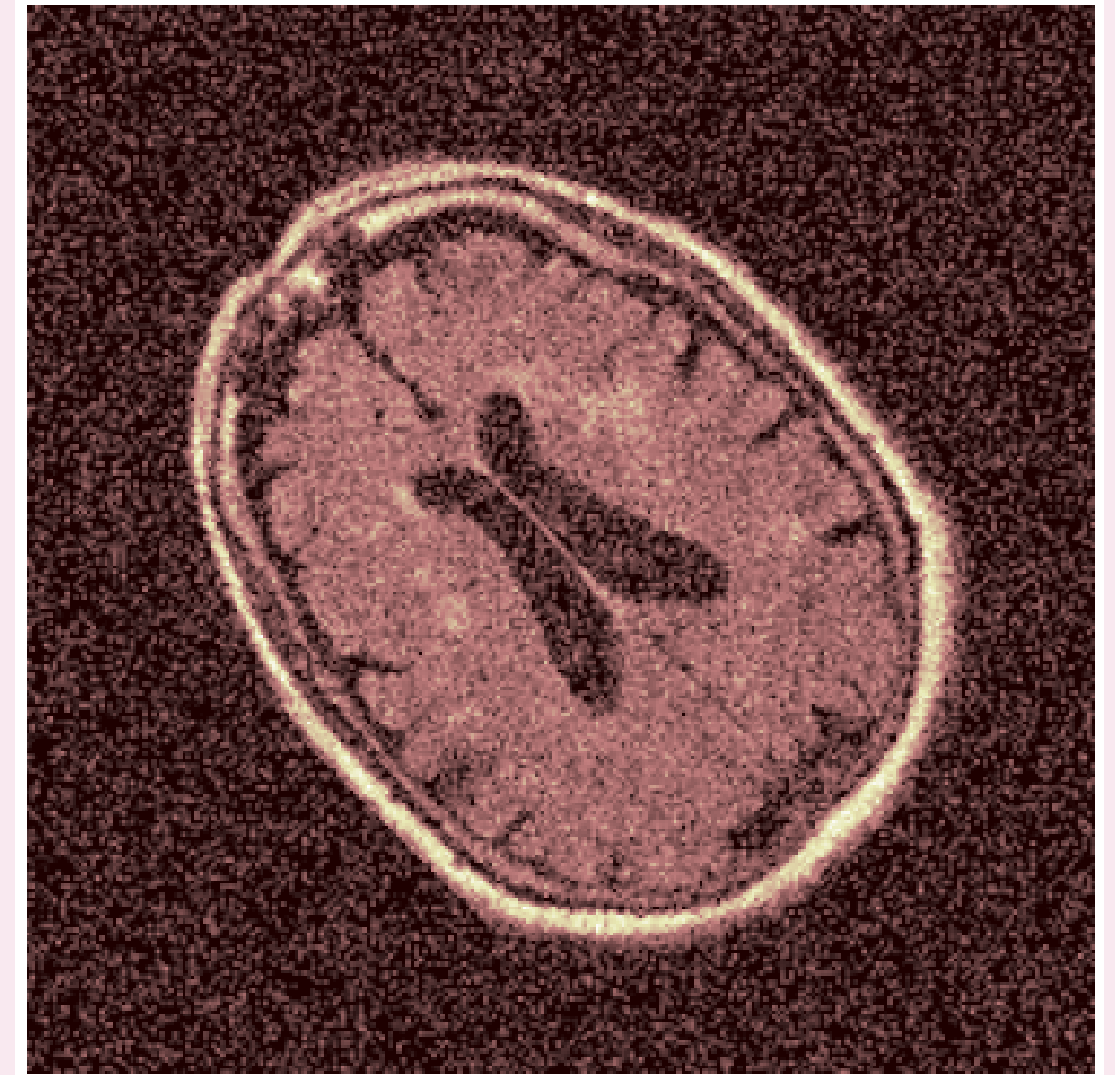
Prétraitement des données

Générer des images modifiées pour augmenter la diversité des données et éviter le surapprentissage

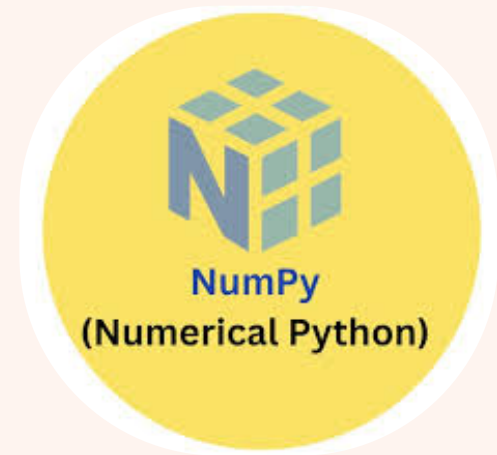
Rotated Image



Noisy Image

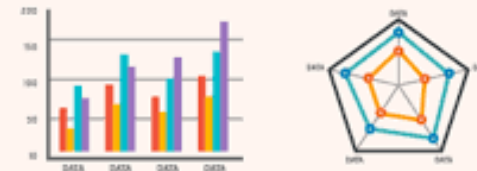


Bibliothèques utilisées et Définition des métriques



numpy

matplotlib



sklearn

os

OS Module
>>> import os

pandas



seaborn

tensorflow

PIL



1

2

3

4

5

6

7

8

- Les métriques utilisées sont:



accuracy

Proportion de
bonnes
prédictions
parmi toutes les
prédictions.

precision

Proportion de
vrais positifs
parmi tous les
éléments
prédits positifs.

recall

La capacité
du modèle à
ne pas rater
de cas
positifs.

f1_score

Moyenne
harmonique
entre
précision et
rappel.

Modèle 1 - SVM

Le SVM est un algorithme de classification supervisée qui cherche à trouver la meilleure frontière séparant les classes tout en maximisant la marge entre elles.

1. Accuracy

Médiane : 65% de bonnes prédictions en moyenne.

Variabilité modérée à élevée : certaines itérations sont très bonnes (~ 1.0), d'autres sont faibles.

2. Precision

Médiane élevée (~ 0.85), donc le modèle fait peu de fausses alertes quand il prédit "tumeur".

Faible variabilité en général. Très bon signe pour éviter les faux positifs.

3. Recall

Médiane autour de 0.6 mais grande variabilité (allant de ~ 0.1 à 1.0).

Ce modèle manque parfois des cas de tumeurs réelles.

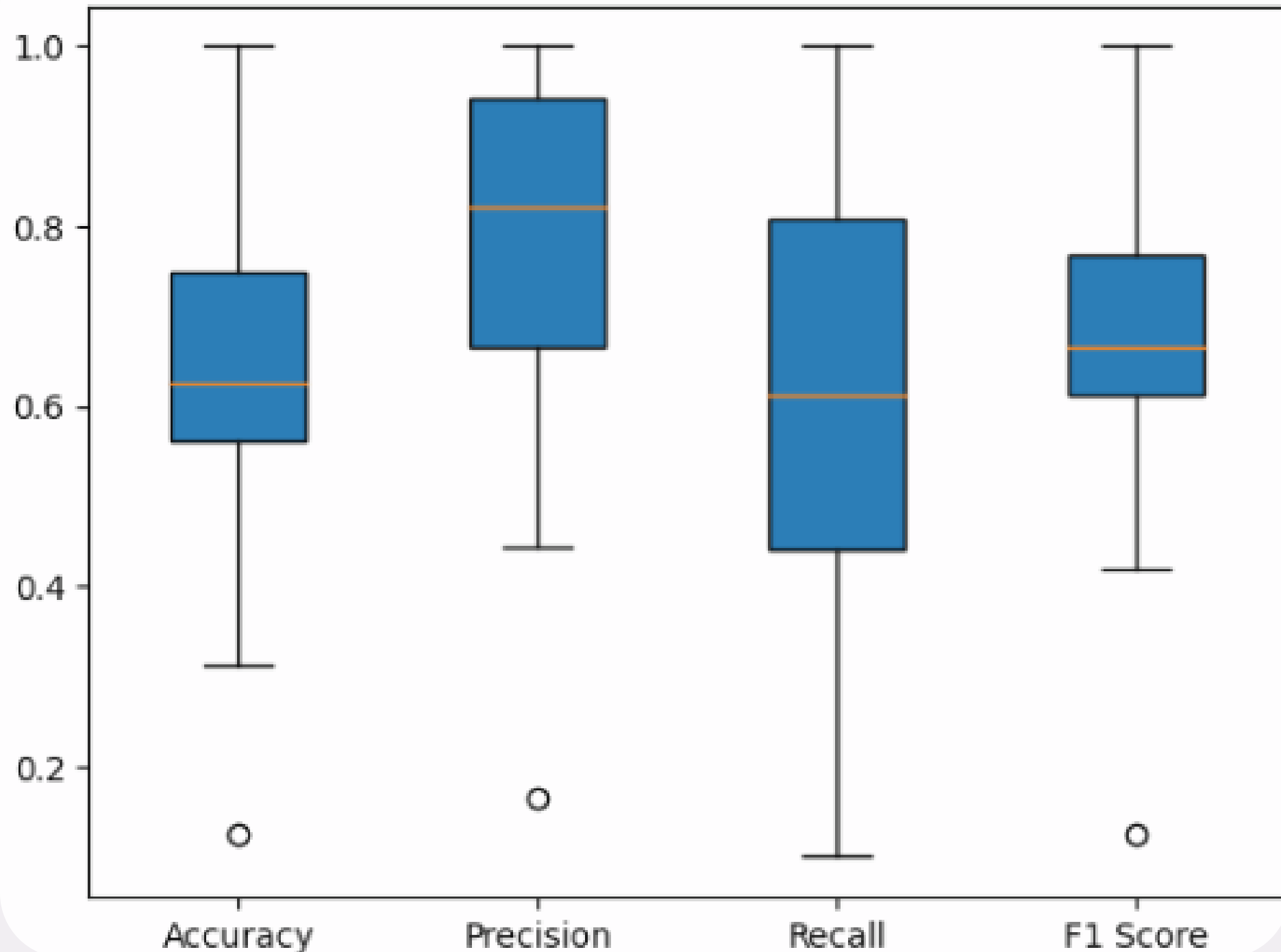
4. F1 Score

Médiane autour de 0.7, ce qui est un compromis entre précision et rappel.

Variabilité modérée.

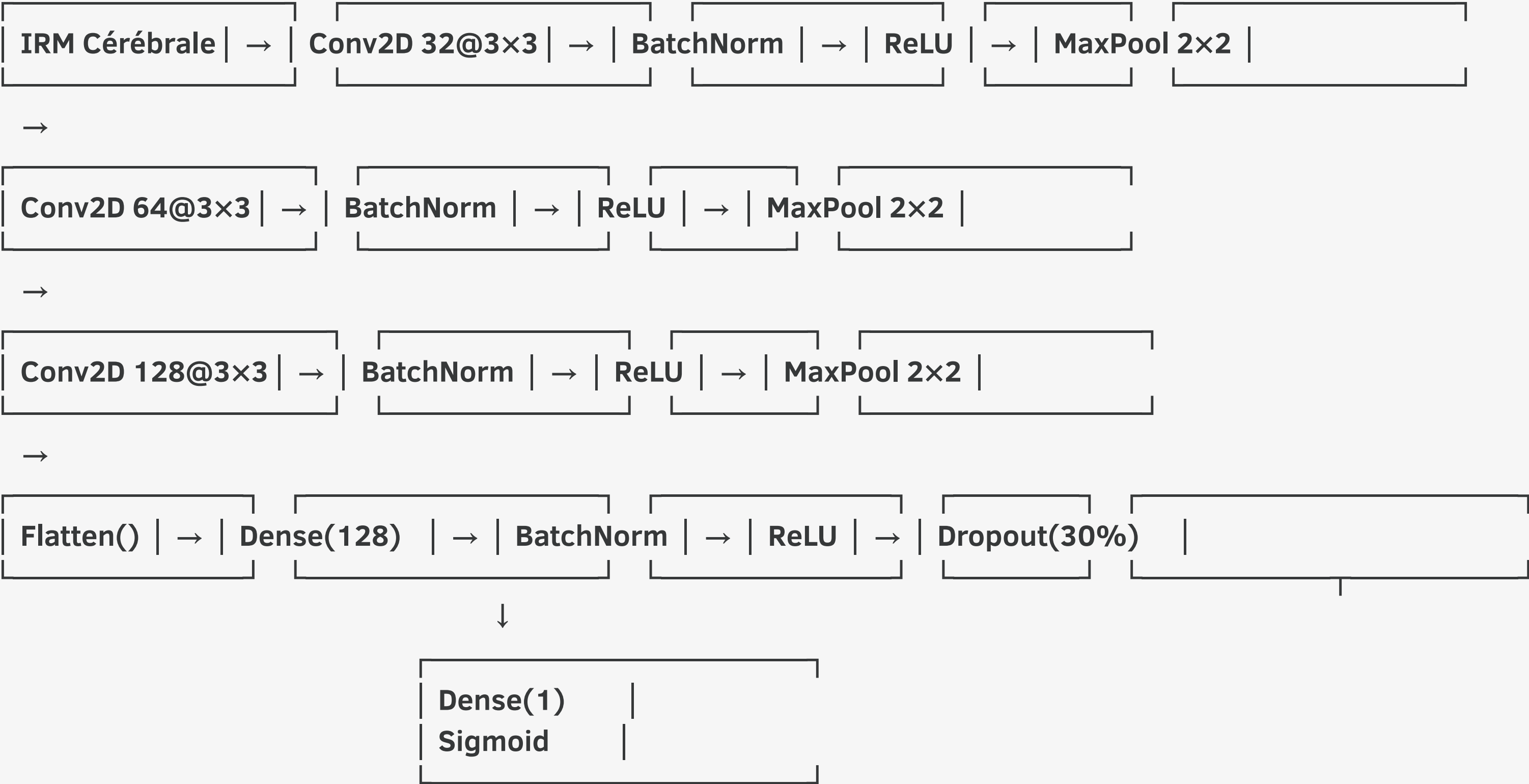
Le modèle est globalement correct, mais pas toujours équilibré entre précision et rappel.

Distribution des métriques du modèle SVM sur 10 itérations



Modèle 2 - CNN personnalisé

Architecture utilisée



Résultat et Interprétation :

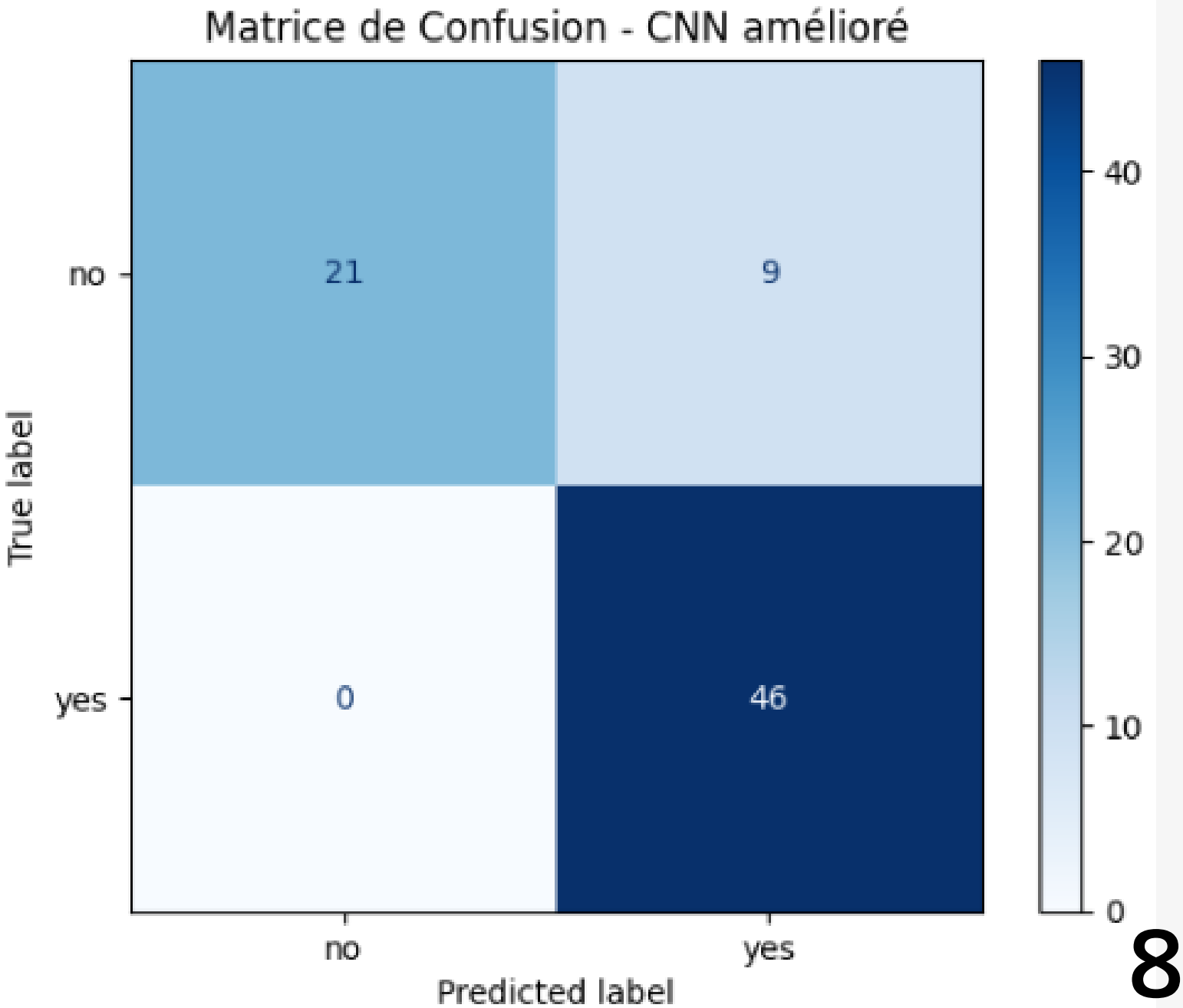
Classe 0 :

Précision = 1.00 : toutes les prédictions faites comme "classe 0" sont correctes.
Rappel = 0.70 : le modèle identifie 70% des vrais cas de classe 0, mais en rate encore 30%.
F1-score = 0.82 : bon équilibre, mais le rappel pourrait être amélioré.

Classe 1 :

Précision = 0.84 : 84% des prédictions en classe 1 sont justes.
Rappel = 1.00 : le modèle identifie tous les cas de classe 1.
F1-score = 0.91 : excellente performance globale sur cette classe.

Classification Report - CNN amélioré					
	precision	recall	f1-score	support	
0	1.00	0.70	0.82	30	
1	0.84	1.00	0.91	46	
accuracy			0.88	76	
macro avg	0.92	0.85	0.87	76	
weighted avg	0.90	0.88	0.88	76	



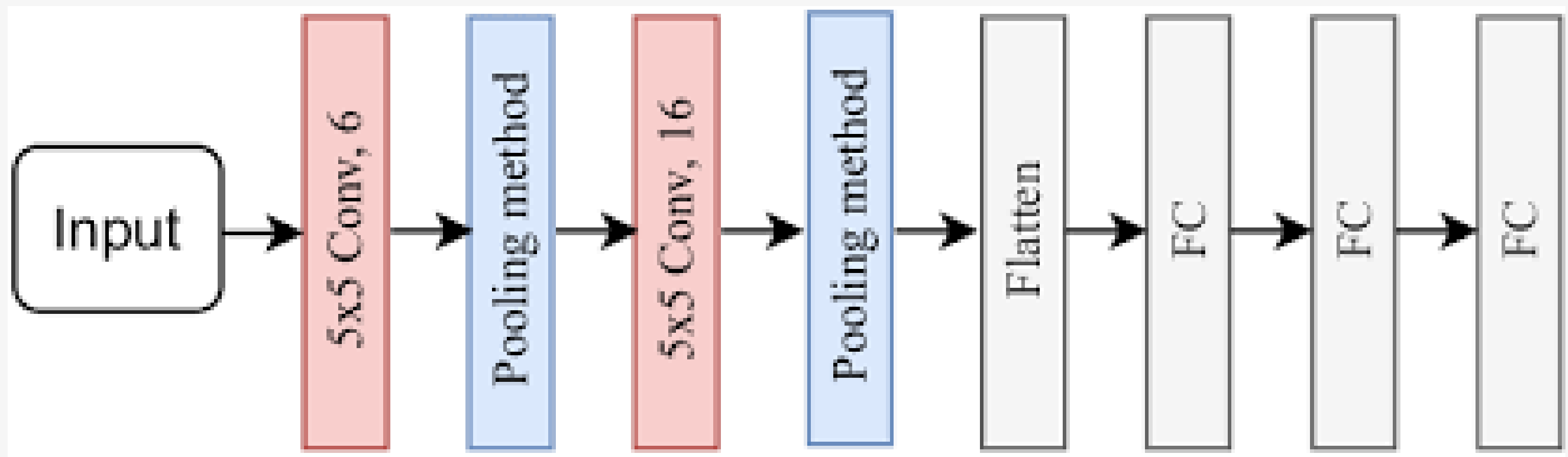
Conclusion :

Le CNN amélioré présente des performances excellentes sur la classe 1, avec un rappel parfait, ce qui est crucial pour ne rater aucun cas positif. Il est également très précis sur la classe 0, mais il lui arrive de manquer des cas réels de cette classe. Globalement, c'est un modèle très performant, particulièrement bien équilibré entre précision et rappel.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
batch_normalization (BatchNormalization)	(None, 222, 222, 32)	128
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 109, 109, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 52, 52, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11,075,712
batch_normalization_3 (BatchNormalization)	(None, 128)	512
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

Modèle 3 - LeNet-5

Architecture utilisée



LeNet-5 Test Accuracy: 0.8289

Résultat et Interprétation :

Classe 0 :

Précision = 0.90 : parmi les prédictions faites en classe 0, 90% étaient correctes.

Rappel = 0.63 : le modèle a détecté 63% des vrais cas de classe 0.

F1-score = 0.75 : bon compromis entre précision élevée et rappel modéré.

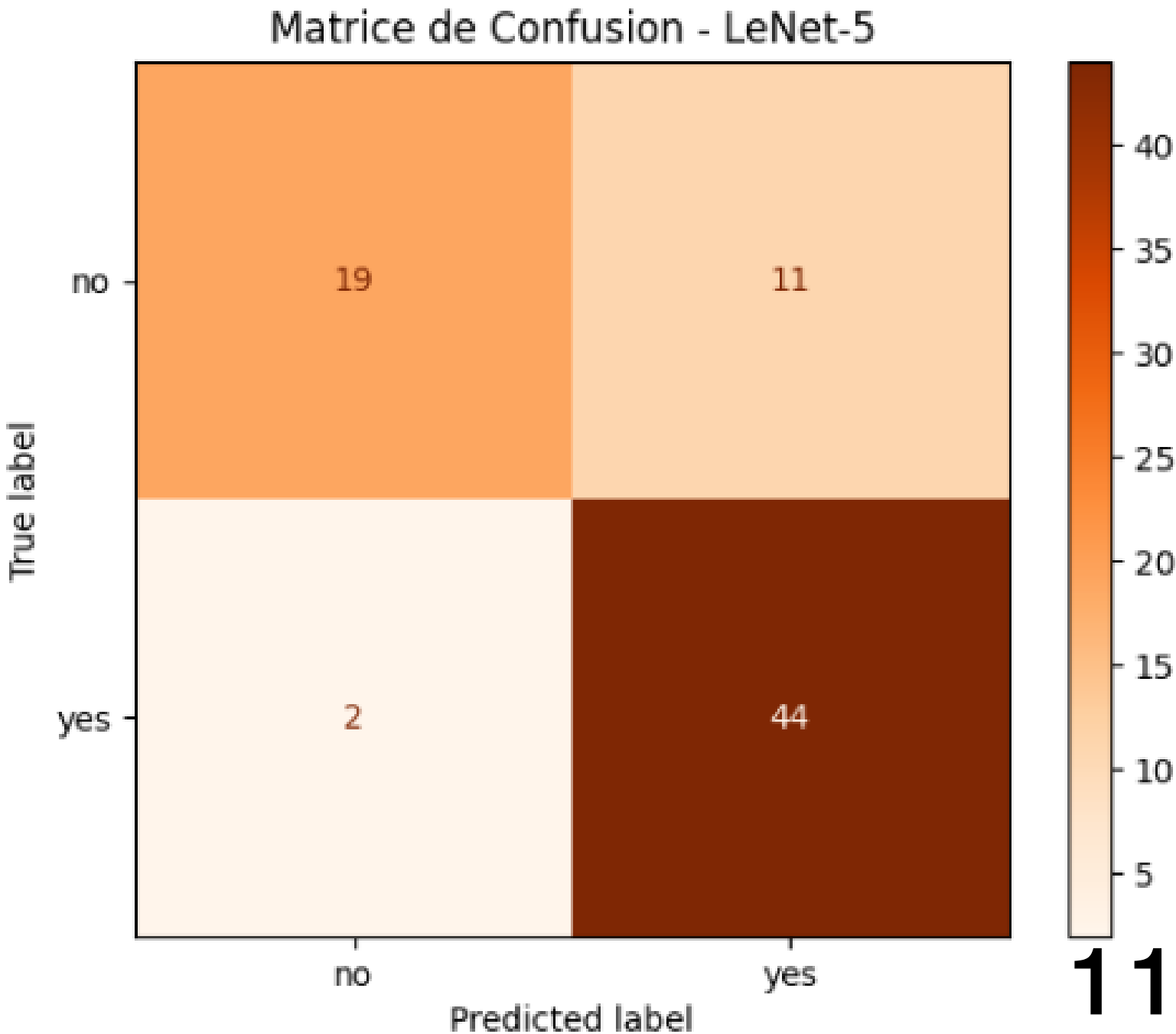
Classe 1 :

Précision = 0.80 : 80% des prédictions en classe 1 sont correctes.

Rappel = 0.96 : presque tous les cas réels de classe 1 ont été correctement identifiés.

F1-score = 0.87 : très bonne performance globale sur cette classe.

Classification Report - LeNet-5					
		precision	recall	f1-score	support
	0	0.90	0.63	0.75	30
	1	0.80	0.96	0.87	46
accuracy				0.83	76
macro avg		0.85	0.79	0.81	76
weighted avg		0.84	0.83	0.82	76



Conclusion :

LeNet-5 montre des performances équilibrées, avec :

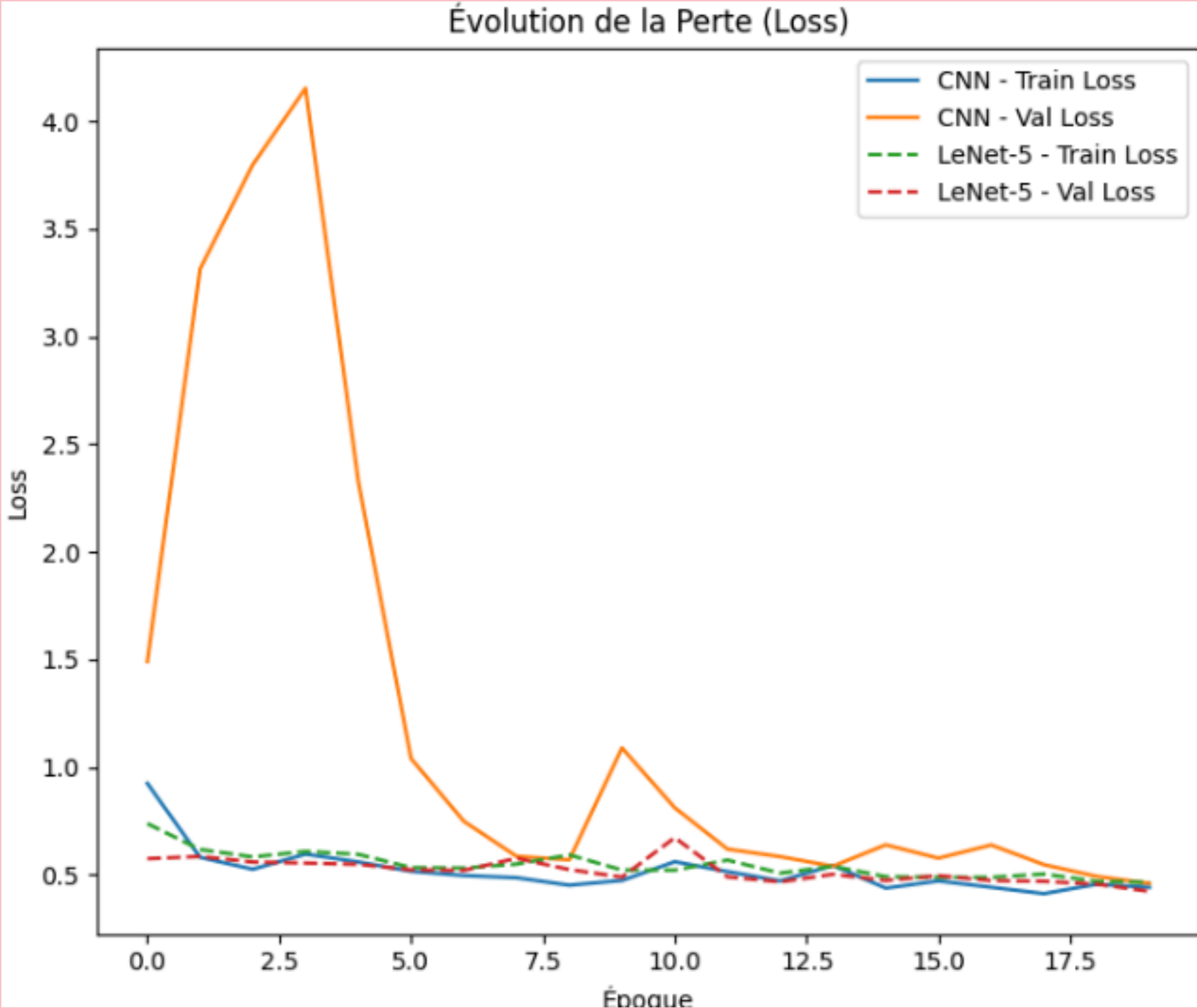
Une excellente détection de la classe majoritaire (1).

Une bonne précision sur la classe minoritaire (0), bien qu'avec un rappel plus faible.

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 220, 220, 6)	456
max_pooling2d_3 (MaxPooling2D)	(None, 110, 110, 6)	0
conv2d_4 (Conv2D)	(None, 106, 106, 16)	2,416
max_pooling2d_4 (MaxPooling2D)	(None, 53, 53, 16)	0
flatten_1 (Flatten)	(None, 44944)	0
dense_2 (Dense)	(None, 120)	5,393,400
dense_3 (Dense)	(None, 84)	10,164
dense_4 (Dense)	(None, 1)	85

Comparaison des performances

	Métrique	CNN amélioré	LeNet-5
	Accuracy	0.88	0.83
Precision	- Classe 0	1.00	0.90
	Recall - Classe 0	0.70	0.63
	F1-score - Classe 0	0.82	0.75
Precision	- Classe 1	0.84	0.80
	Recall - Classe 1	1.00	0.96
	F1-score - Classe 1	0.91	0.87
	Macro F1-score	0.87	0.81
	Weighted F1-score	0.88	0.82



Interprétation Générale :

Le modèle CNN amélioré affiche une accuracy globale supérieure (0.88) par rapport à LeNet-5 (0.83), indiquant une meilleure performance générale sur l'ensemble du jeu de test.

Pour la classe 1 (majoritaire (310)) :

- Le CNN amélioré obtient un rappel parfait (1.00), ce qui signifie qu'il détecte tous les cas positifs réels.
- LeNet-5 a un rappel légèrement inférieur (0.96) mais une précision un peu plus faible (0.80 contre 0.84 pour le CNN).
- Le F1-score est donc légèrement meilleur pour le CNN (0.91 vs 0.87), montrant un bon équilibre entre précision et rappel.

Pour la classe 0 (minoritaire (196)) :

- Le CNN amélioré obtient une précision parfaite (1.00), ce qui signifie que toutes ses prédictions de classe 0 sont correctes. Rappel est plus faible (0.70), ce qui montre qu'il rate certains vrais cas de classe 0.
- LeNet-5 détecte plus de cas de classe 0 (rappel = 0.63) mais avec une précision moindre (0.90).
- Le F1-score est donc plus élevé pour le CNN (0.82 vs 0.75), montrant une meilleure performance globale sur cette classe aussi.

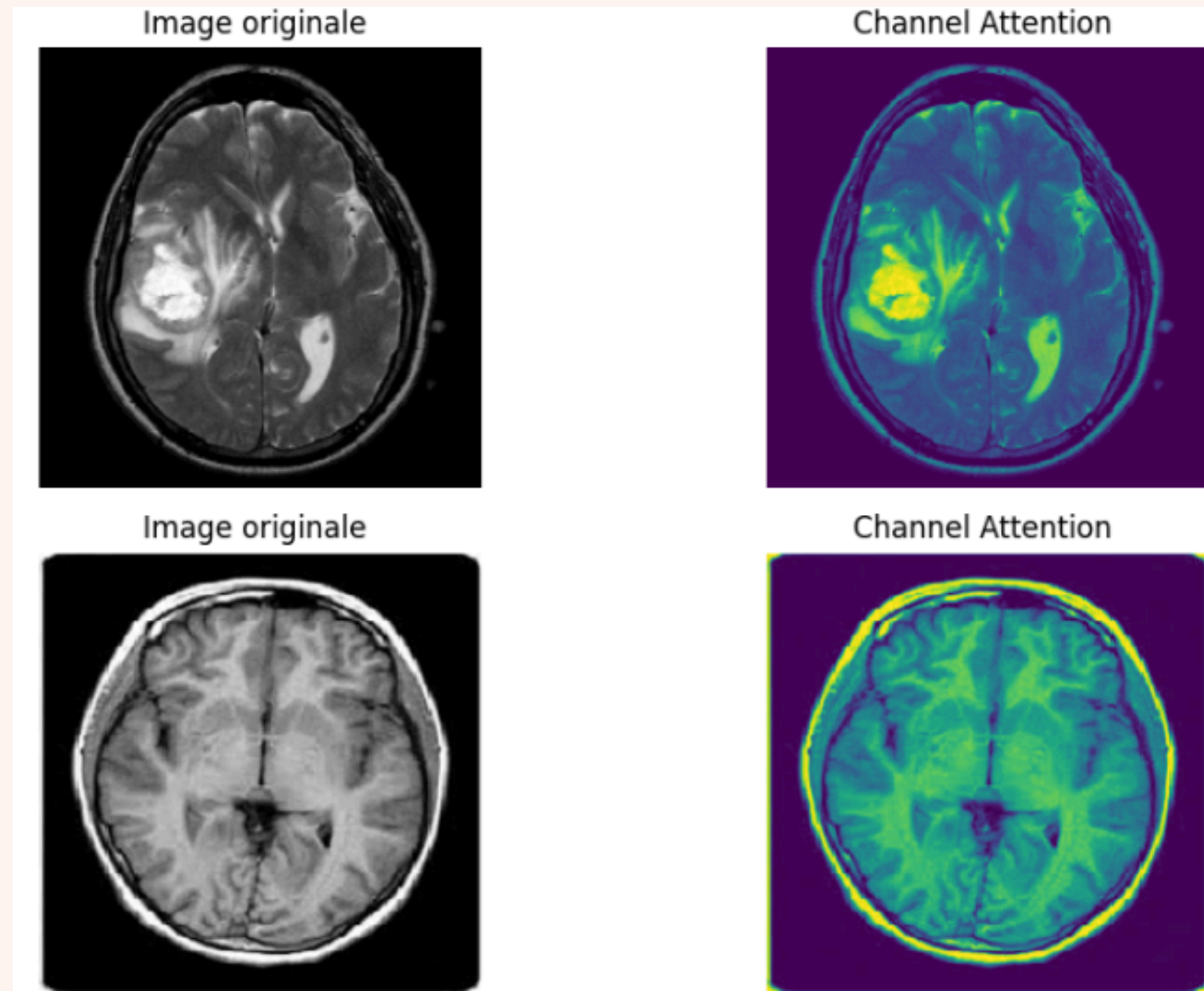
Le modèle CNN amélioré surpasse LeNet-5 sur presque toutes les métriques, notamment avec :

- Une meilleure accuracy globale.
- Un meilleur équilibre entre précision et rappel pour les deux classes.
- Des performances solides aussi bien sur la classe majoritaire (1) que la classe minoritaire (0).

Amélioration du modèle avec les modules d'attention :

1.Channel Attention Module

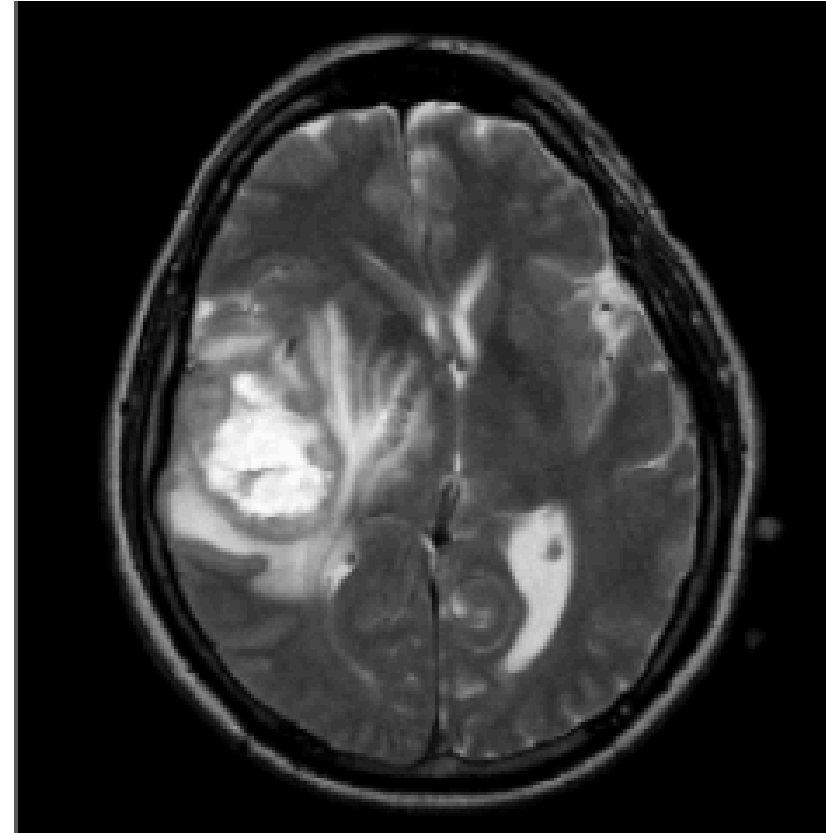
Accuracy sur le jeu de test avec channel_attention : 0.8289



2. Spatial Attention Module

Accuracy sur le jeu de test :
0.8553

Image originale



Avec Attention Spatiale

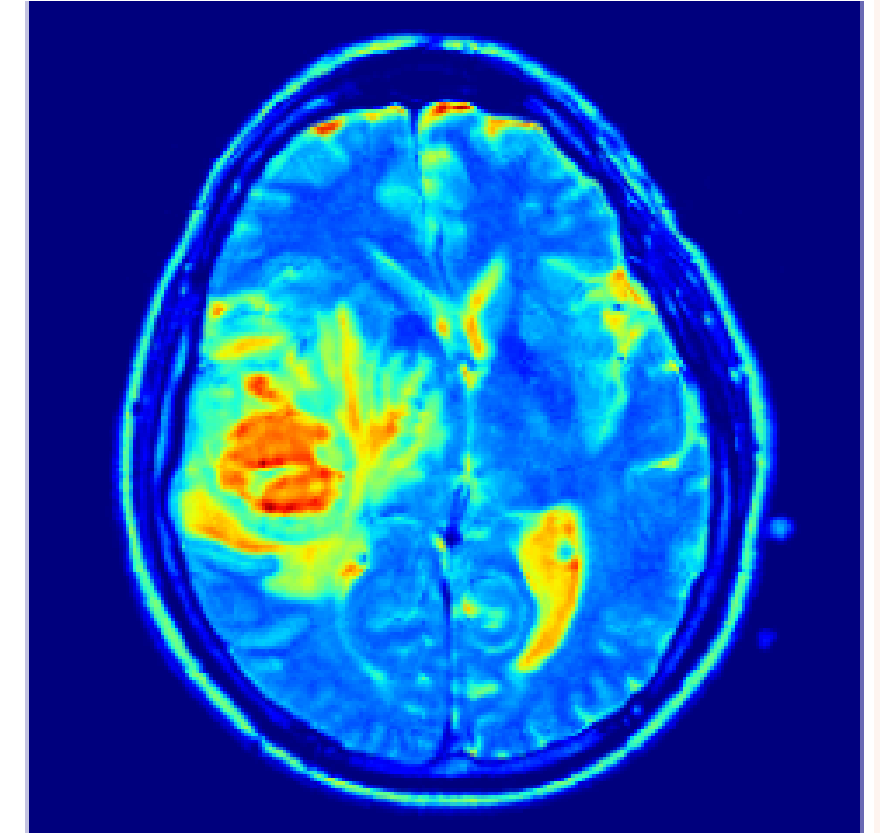
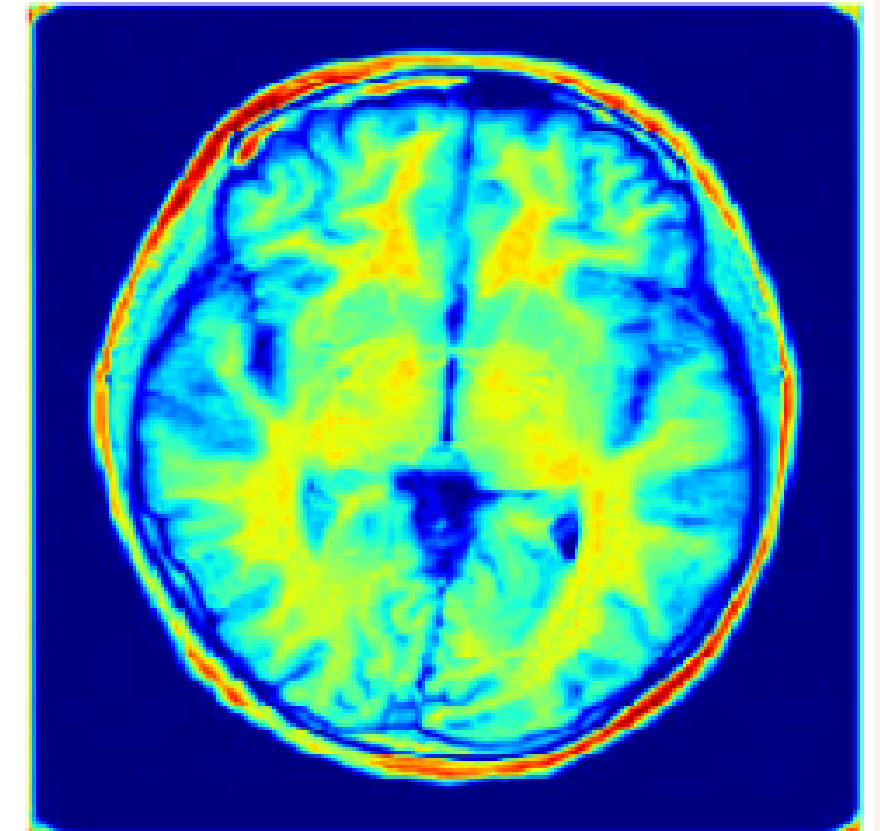


Image originale

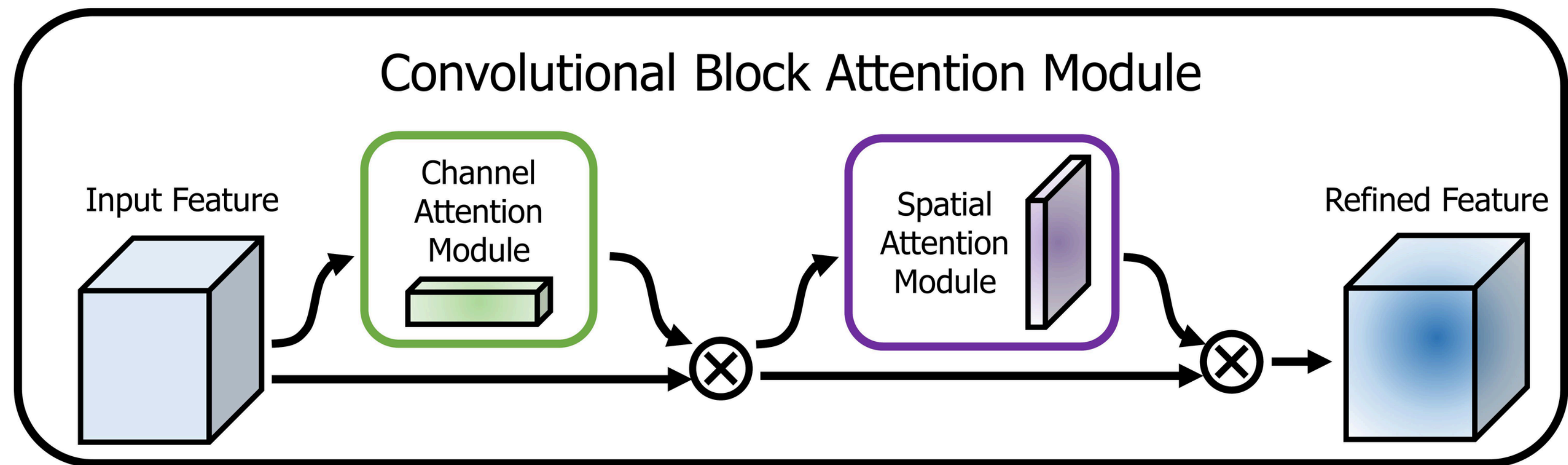


Avec Attention Spatiale



Module CBAM complet

Architecture utilisée



Accuracy sur le jeu de test : 0.7895
Accuracy sur le jeu de validation : 0.8026

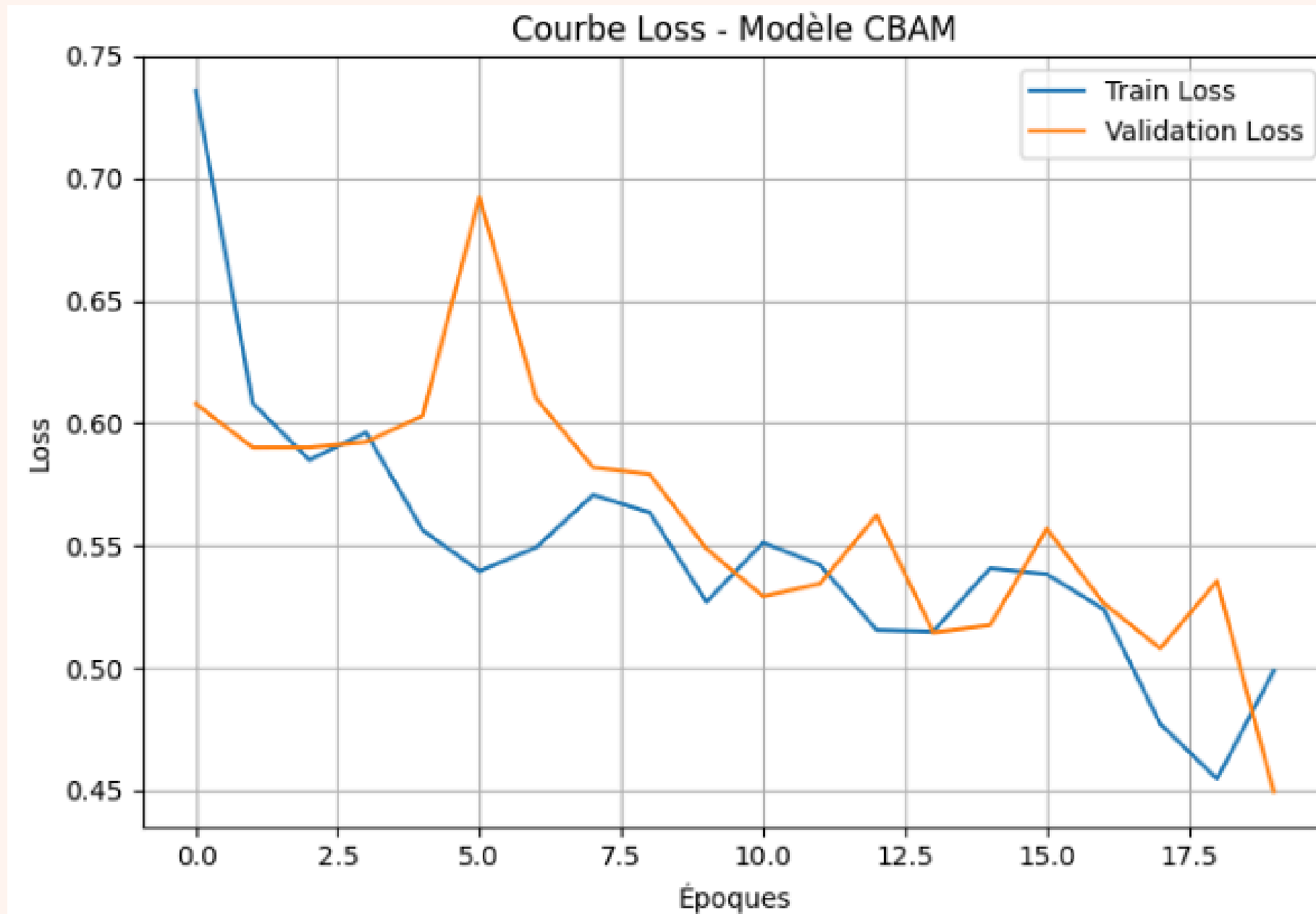
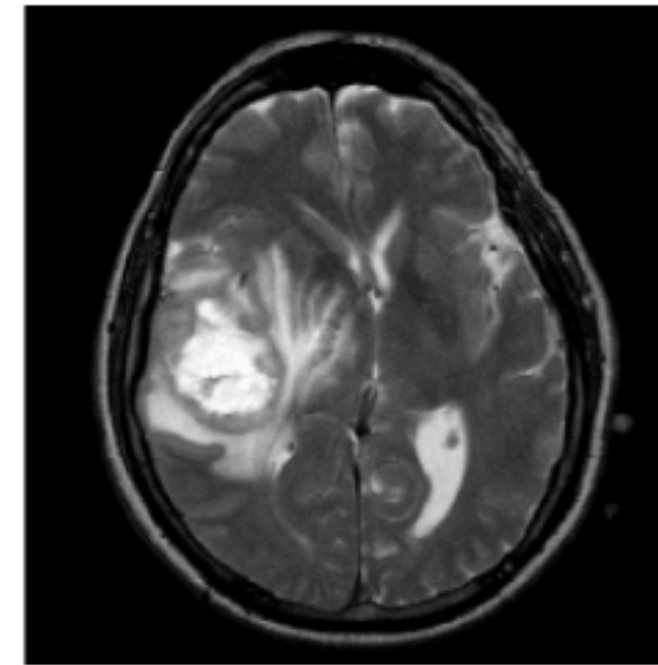


Image originale



Carte d'attention CBAM

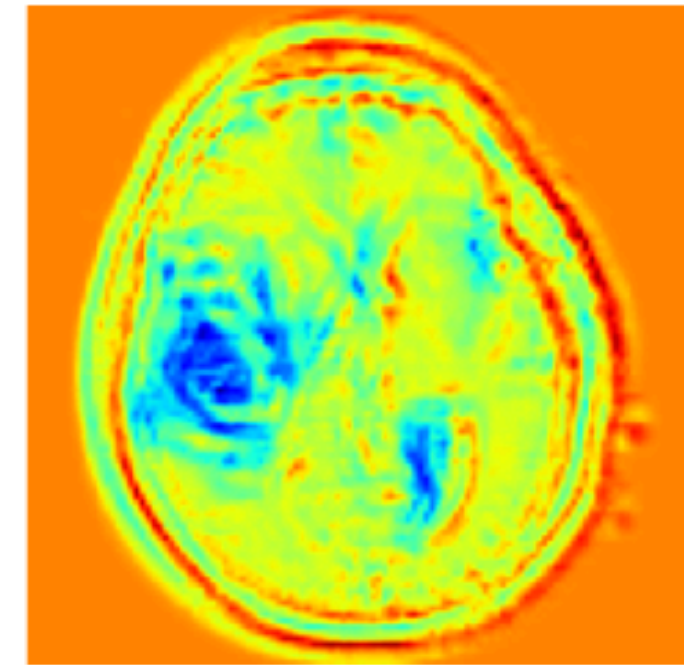
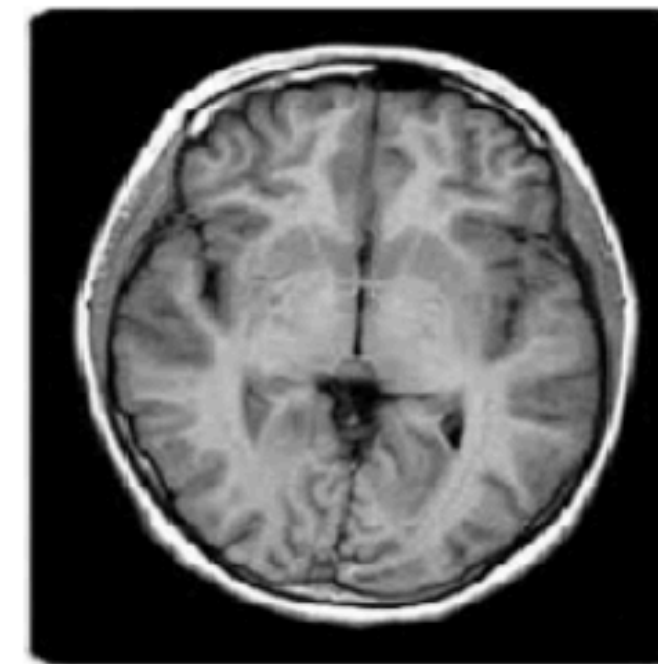
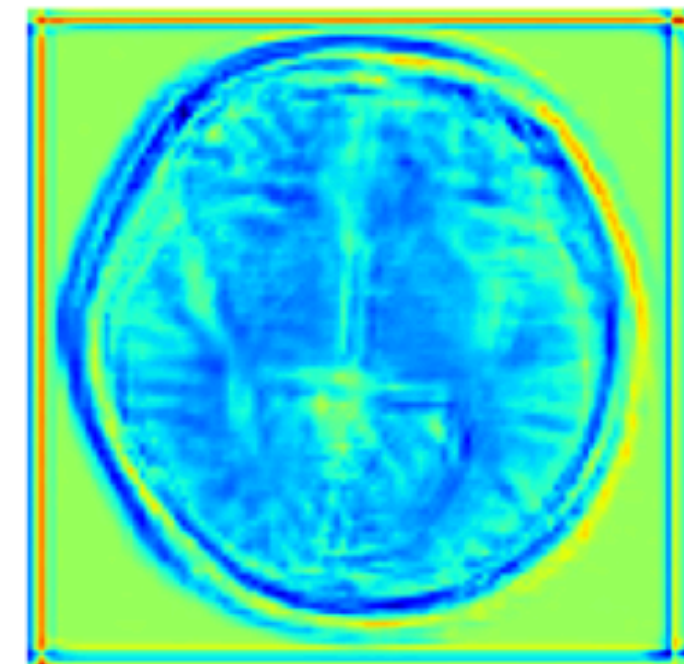


Image originale



Carte d'attention CBAM



END Conclusion

Nous avons exploré différentes approches de classification d'images IRM. Le modèle CNN personnalisé s'est distingué par la meilleure précision, démontrant l'efficacité des architectures adaptées aux données spécifiques.

Un tableau de bord interactif a été développé pour faciliter l'utilisation du modèle spatial_attention. Ces résultats ouvrent la voie à d'éventuelles améliorations, notamment par l'intégration de modules d'attention.

« Merci de votre attention !! »

