

Projet 6 : *Classez des images à l'aide d'algorithmes de Deep Learning*

Matthieu MARQUER – 20 mars 2024



Sommaire

01 – Le Contexte

02 – Exploration

03 – Les modèles

04 – Les Résultats

01

Le Contexte

L'objectif de ce projet est de développer un modèle capable de prédire la race des chiens à partir de photos.

Caractéristiques des données

Le jeu de données utilisé provient du Stanford Dogs Dataset:
<http://vision.stanford.edu/aditya86/ImageNetDogs/>

Les données initial contiennent 120 races de chiens, mais j'ai limité l'analyse à seulement 3 races pour la comparaison des modèles.

Après avoir sélectionné le meilleur modèle, j'ai ré-entraîné ce modèle en utilisant un ensemble de données élargi contenant 50 races pour une prédiction plus large.

Données initiales :

- images.tar

A series of thin, dark, wavy lines that flow across the bottom of the slide, creating a sense of movement and depth.

02

Exploration des données

1. Création du répertoire de test avec déplacement des images (20% des photos)
2. Récupération des noms de races à partir des sous dossier
3. Encodage des labels via **LabelEncoder**
4. Vérification du nombre de races et du nombre de photo
5. Réduction de dimension PCA et T-SNE

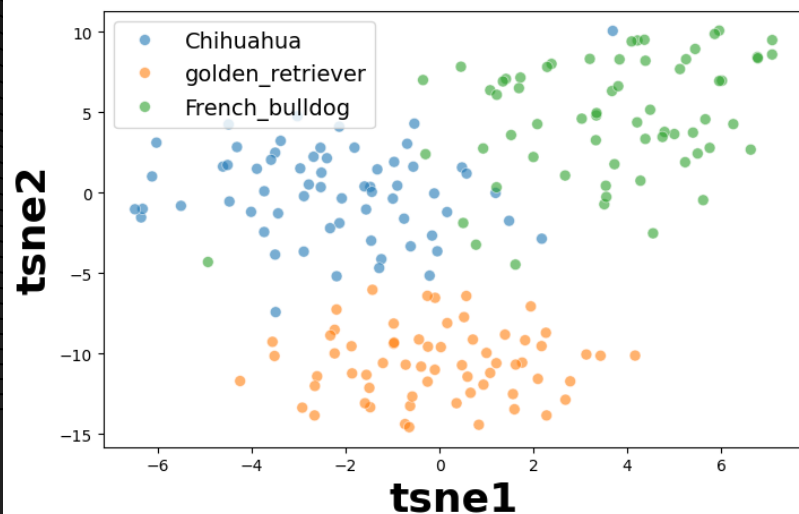
Cluster selon k-means:

0: Bulldog

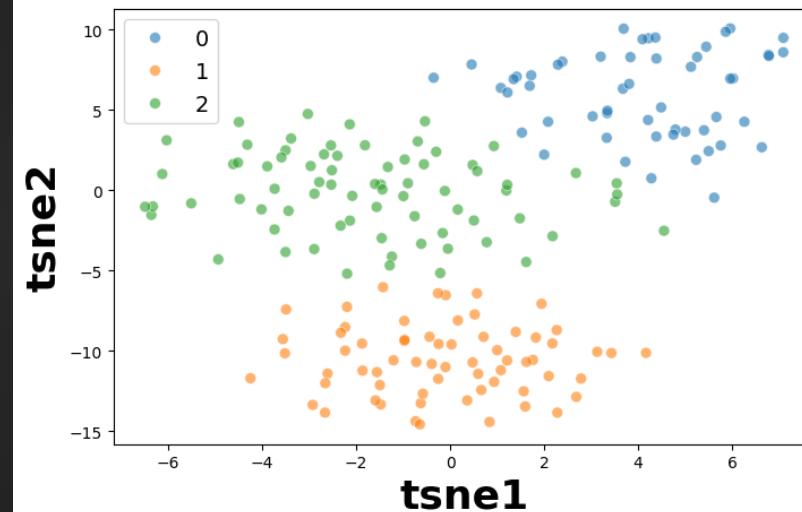
1: Chihuahua

2: Golden retriever

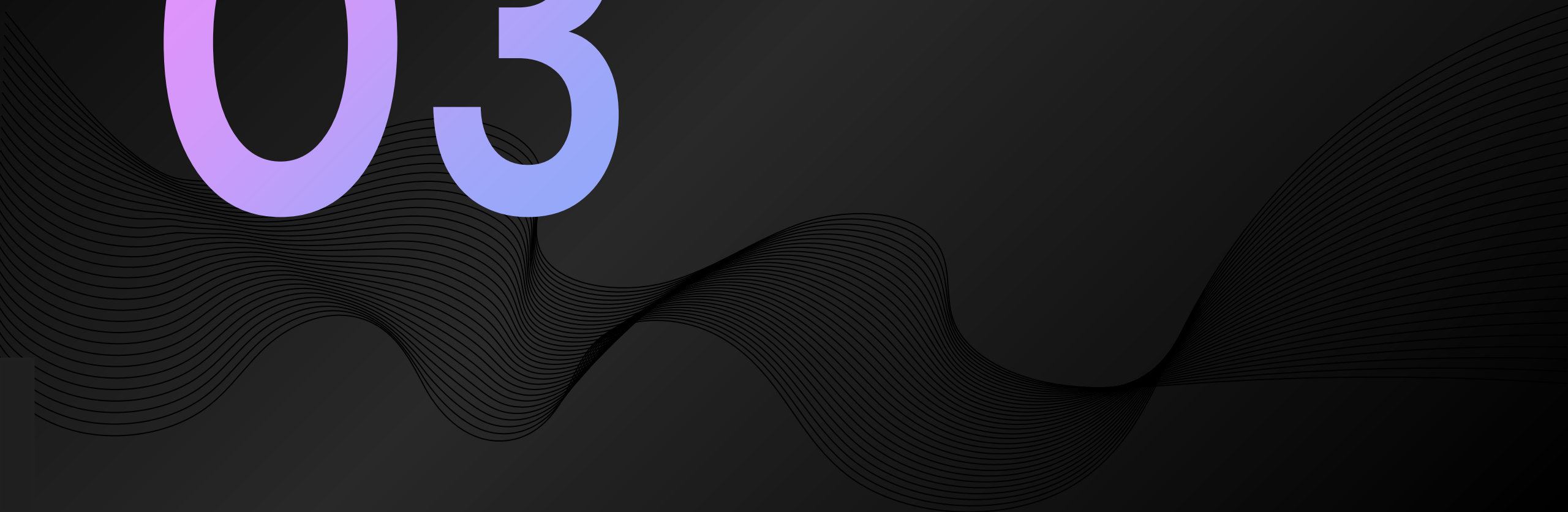
TSNE selon les vraies classes



TSNE selon les clusters



ARI: 0.80

A series of thin, dark grey wavy lines that flow across the bottom of the slide, creating a sense of movement and depth.

03

Les Modèles

Les modèles custom:

Custom model v0

Ce modèle a été conçu avec une approche itérative d'optimisation des hyperparamètres. Il comporte une architecture en couches convolutionnelles et entièrement connectées pour extraire les caractéristiques des images et effectuer la classification.

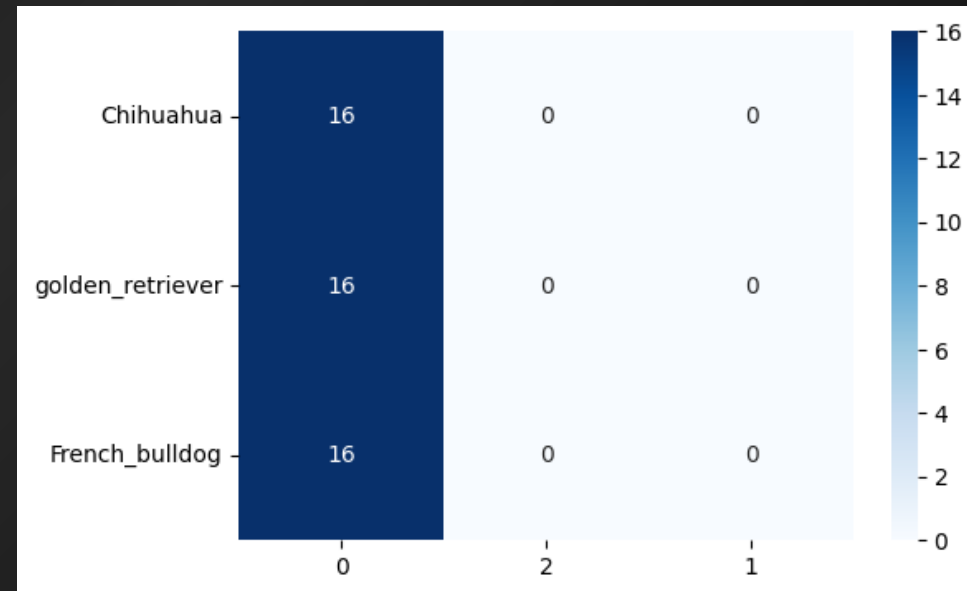
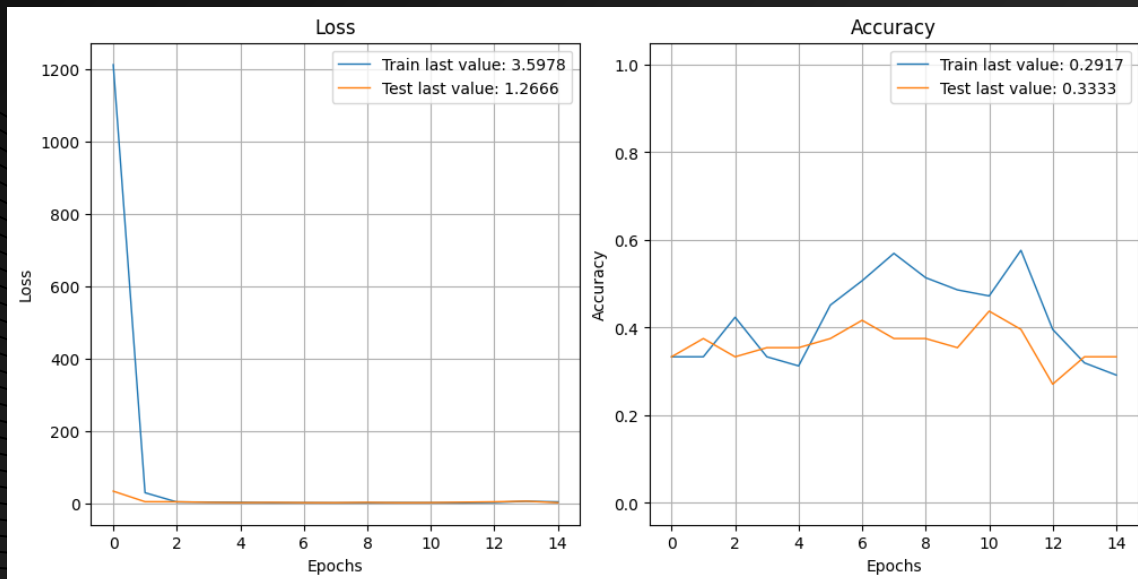
La démarche d'optimisation a porté sur des hyperparamètres tels que le nombre de couches, les tailles de filtres, et le taux de dropout pour régulariser le réseau et éviter le surapprentissage.

Le modèle a été compilé avec une perte "categorical_crossentropy" adaptée à la classification multi-classe, et l'optimiseur RMSprop a été sélectionné pour l'entraînement.

Les paramètres d'exécution tels que la taille de lot (batch_size) et le nombre d'époques ont été ajustés pour obtenir de meilleures performances.

Les modèles custom:

Custom model v0



Les modèles custom:

Custom model v1

La version v1 comprend une architecture plus complexe avec des couches convolutionnelles supplémentaires et l'ajout de couches de normalisation par lots pour accélérer l'apprentissage. De plus, elle utilise un taux de dropout configurable pour régulariser le réseau et éviter le surapprentissage.

Le modèle est compilé avec un optimiseur RMSprop et une perte "categorical_crossentropy" pour la classification multi-classe. Le réglage des hyperparamètres tels que le taux d'apprentissage et le taux de dropout permet d'optimiser les performances du modèle.

Sur la v1 plusieurs hyperparamètres ont été testés:

- `learning_rates = [0.001, 0.002, 0.005]`
- `dropouts = [0.2, 0.4, 0.5]`

Une version étendue de la V1 intègre des techniques de data augmentation, comme le retournement horizontal aléatoire, la rotation aléatoire et le zoom aléatoire, pour améliorer la capacité du modèle à généraliser à partir des données d'entraînement.

Les modèles custom:

Custom model v1

Tableau des résultats:

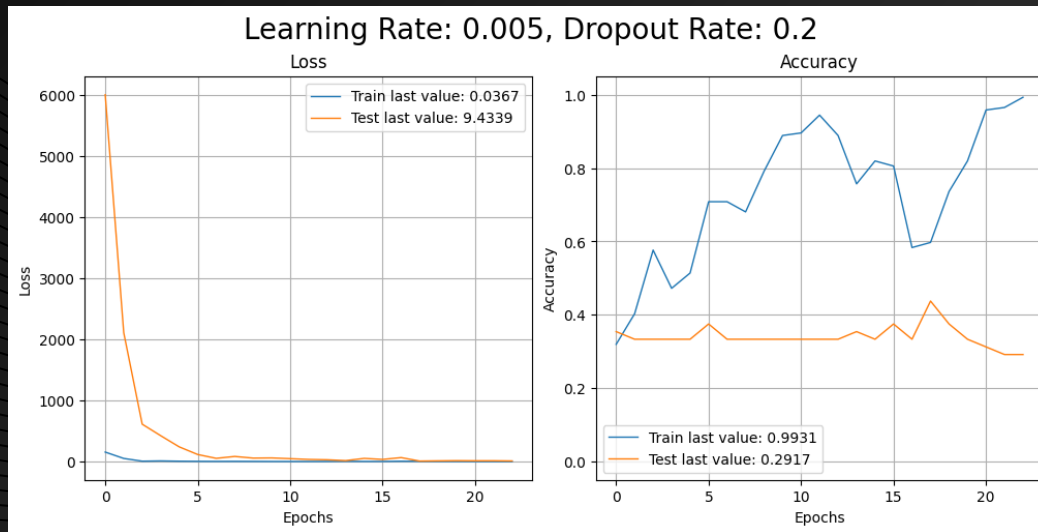
Model	Data Augmentation	Validation Loss Optimal	Validation Accuracy Optimal	Test Accuracy Optimal	Nombre de paramètres	Learning Rate	Dropout Rate	
18	v1	avec	2.054702	0.291667	0.437500	6840323	0.005	0.5
8	v1	sans	138.285706	0.333333	0.354167	5723907	0.005	0.4
6	v1	sans	12.159389	0.333333	0.437500	5723907	0.002	0.5
0	V0	sans	1.266567	0.333333	0.354167	1442947	NaN	NaN
4	v1	sans	6.242844	0.354167	0.354167	5723907	0.002	0.2
14	v1	avec	14.865101	0.354167	0.395833	6840323	0.002	0.4
2	v1	sans	39.916843	0.375000	0.437500	5723907	0.001	0.4
10	v1	avec	2.565145	0.375000	0.395833	6840323	0.001	0.2
13	v1	avec	2.728584	0.375000	0.354167	6840323	0.002	0.2
5	v1	sans	12.342187	0.395833	0.333333	5723907	0.002	0.4
1	v1	sans	9.182384	0.395833	0.291667	5723907	0.001	0.2
17	v1	avec	2.159613	0.395833	0.333333	6840323	0.005	0.4
3	v1	sans	8.300183	0.416667	0.479167	5723907	0.001	0.5
9	v1	sans	30.117920	0.416667	0.312500	5723907	0.005	0.5
7	v1	sans	7.499078	0.437500	0.395833	5723907	0.005	0.2
11	v1	avec	2.181337	0.458333	0.437500	6840323	0.001	0.4
15	v1	avec	3.863509	0.458333	0.375000	6840323	0.002	0.5
12	v1	avec	3.273990	0.479167	0.479167	6840323	0.001	0.5
16	v1	avec	2.292186	0.479167	0.416667	6840323	0.005	0.2

Suivant les résultats obtenu on voit que pour la v1 sans data augmentation (lr: 0.005 et dp: 0.2) est le meilleur model, et que si on lui ajoute la data augmentation ce même model a un accuracy un peu plus élevé: 0.48 contre 0.44 pour la version sans data augmentation

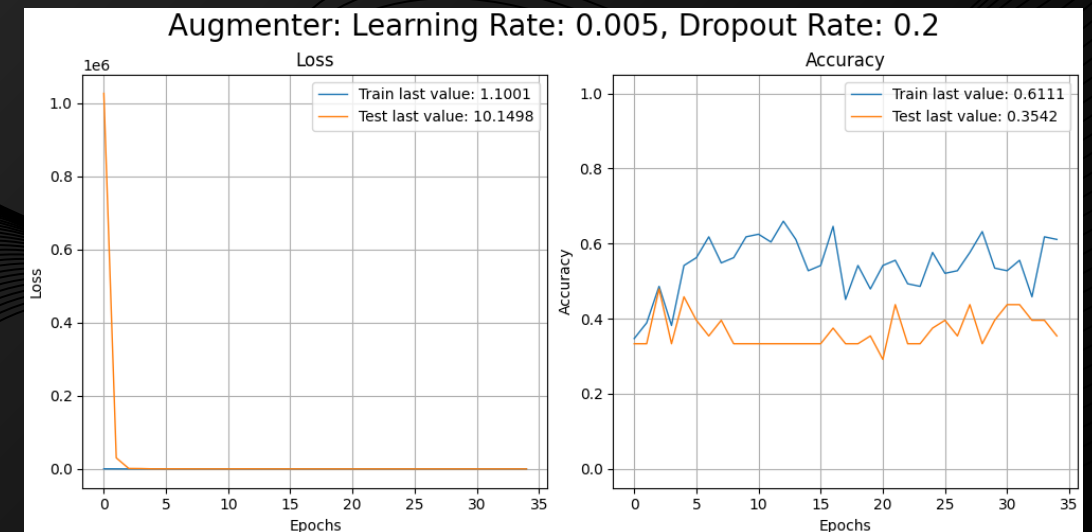
Les modèles custom:

Custom model v1

Le modèle avec un learning rate à 0.005 et un dropout à 0.2, est celui ayant les meilleurs résultats pour la partie custom v1 sans data augmentation.

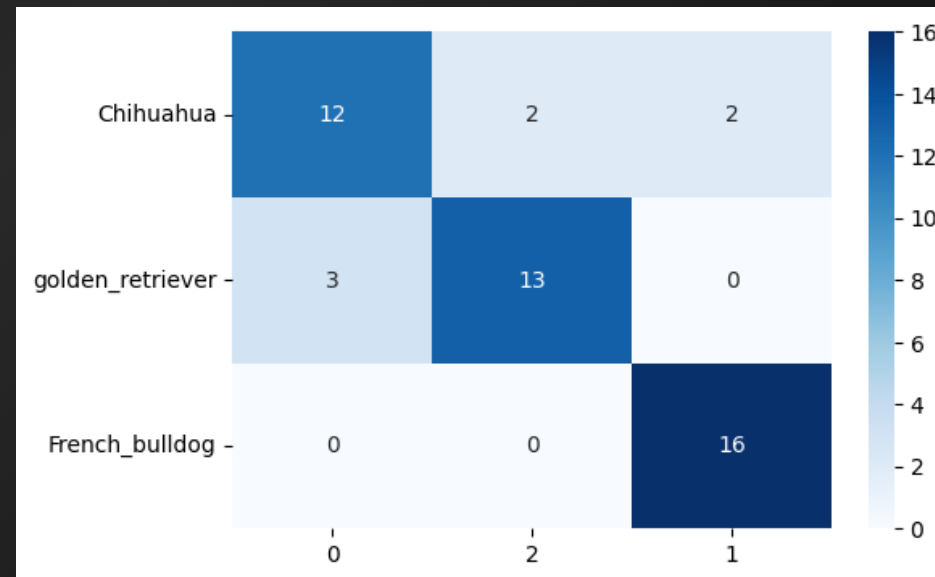
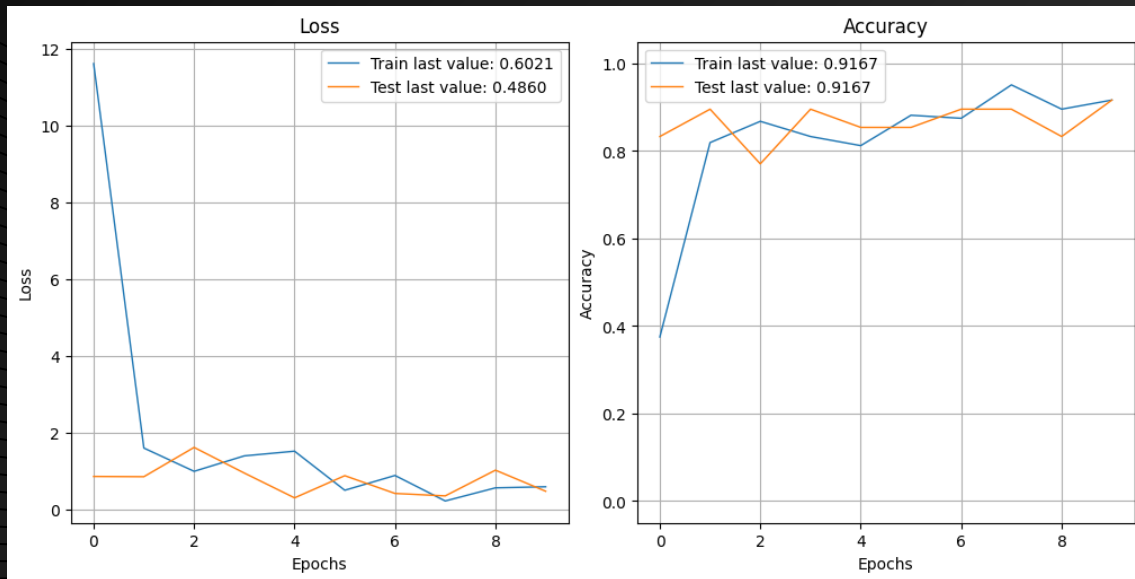


Avec data augmentation.



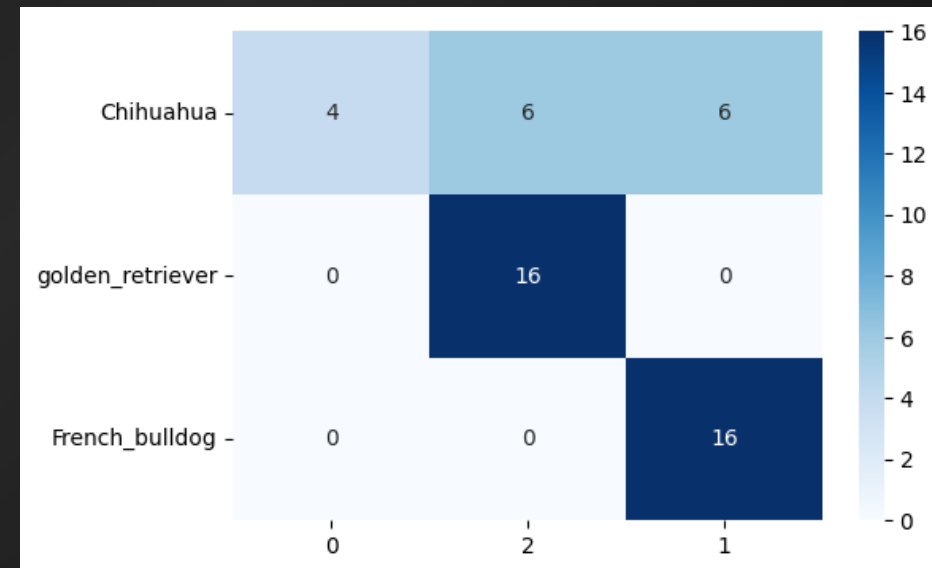
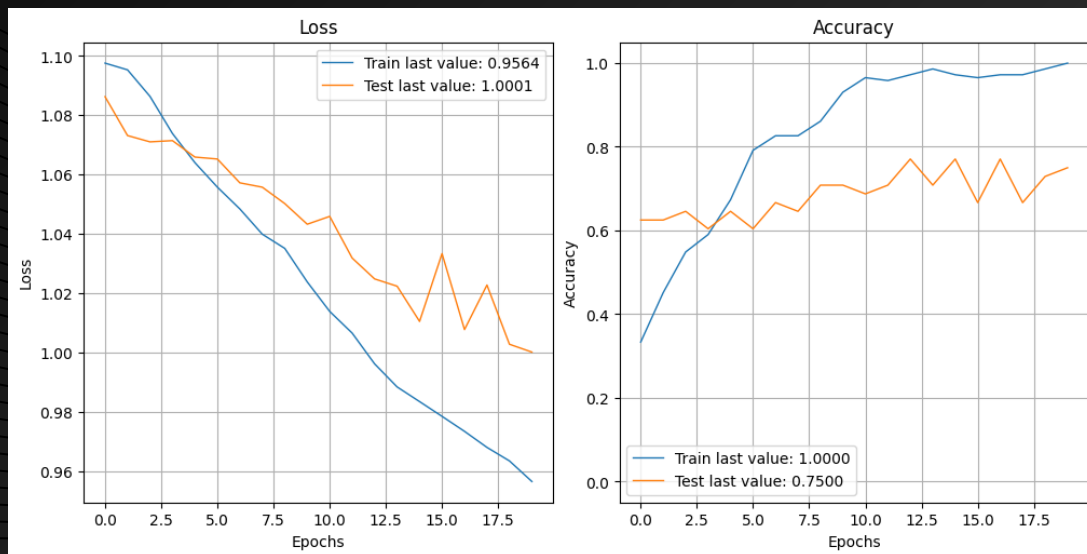
Modèles de transfer learning:

VGG16



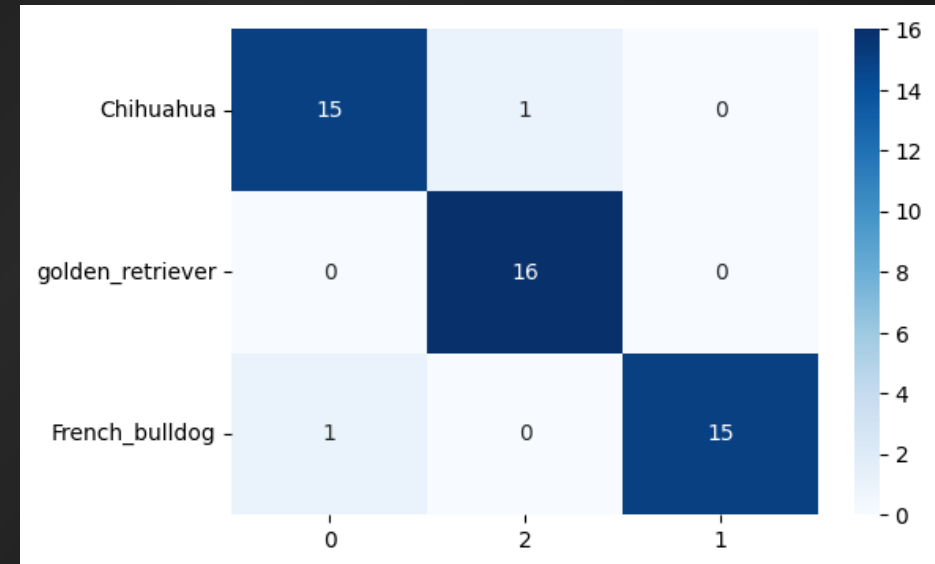
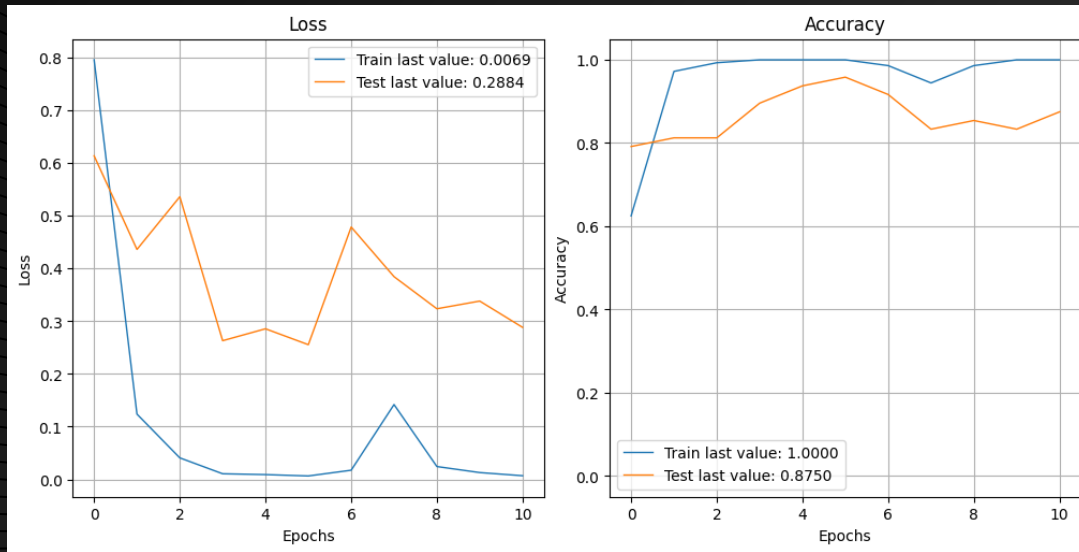
Modèles de transfer learning:

Mobile Net v3



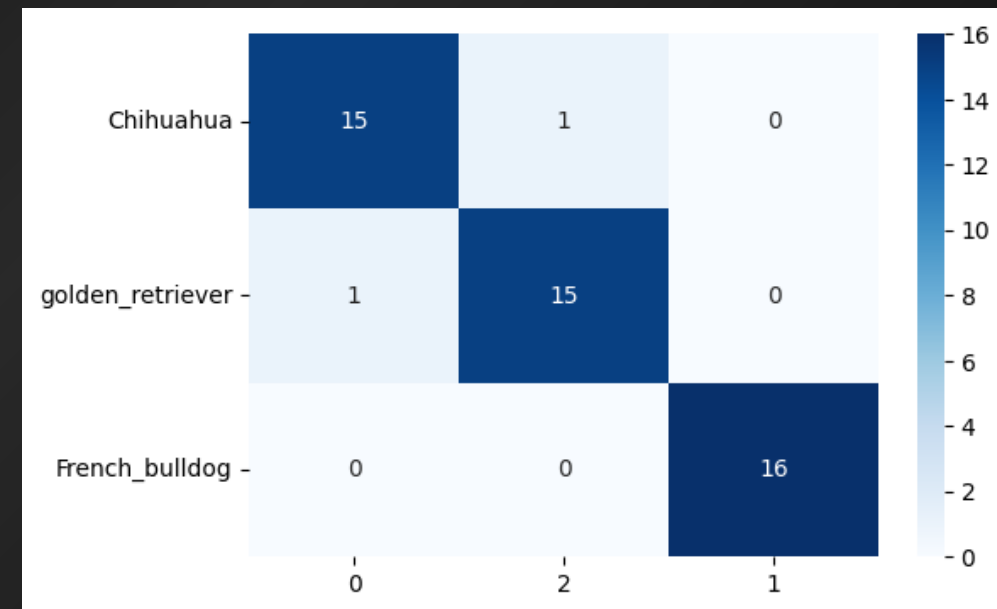
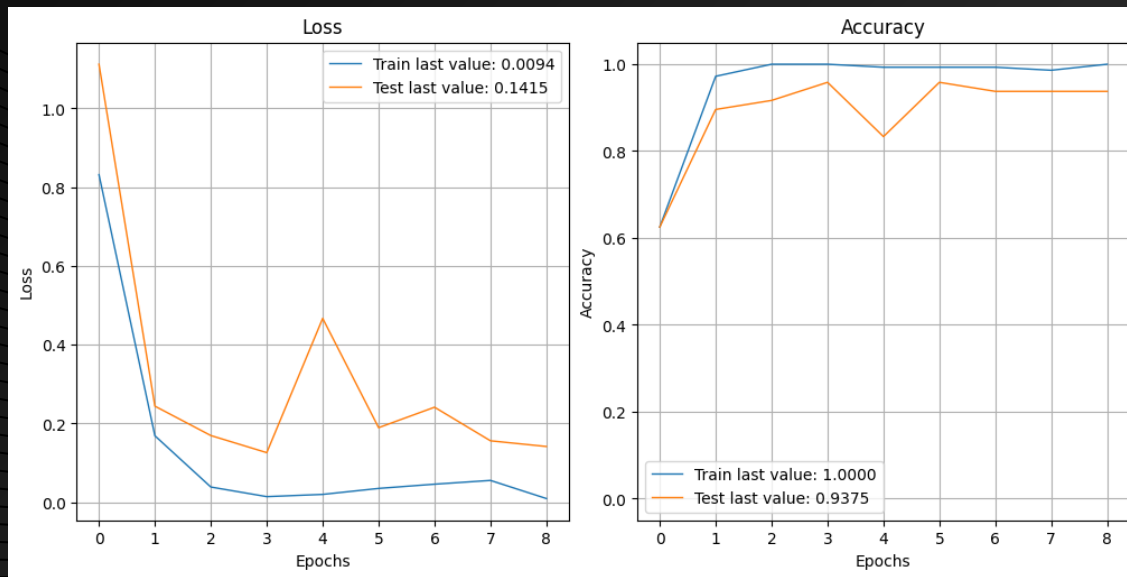
Modèles de transfer learning:

Efficient Net V2 B2



Modèles de transfer learning:

Efficient Net V2 B0:



Modèles de transfer learning:

Tableau des résultats:

	Model	Data Augmentation	Validation Loss Optimal	Validation Accuracy Optimal	Test Accuracy Optimal	Nombre de paramètres
3	Mobile Net v3s	sans	1.000100	0.750000	0.750000	2557971
0	VGG16	sans	0.313731	0.854167	0.937500	14846787
4	Efficient Net v2 B0	avec	0.210392	0.916667	0.916667	5923155
1	Efficient Net v2 B0	sans	0.125881	0.958333	0.875000	5923155
2	Efficient Net v2 B2	sans	0.255487	0.958333	0.958333	8773601

A series of thin, dark, wavy lines that create a sense of movement and depth, flowing across the bottom of the slide.

04

Les Résultats

Les résultats:

Après l'analyse des résultats obtenus et en tenant compte du temps de calcul, la décision finale s'est portée en faveur d'EfficientNet V2 B0.

Cette sélection s'est basée sur les performances de ce modèle par rapport aux autres candidats, ainsi que sur sa capacité à fournir des résultats satisfaisants.

Merci...

Avez-vous des questions?