

Développez une preuve de concept

- DataSpace - Réaliser une veille

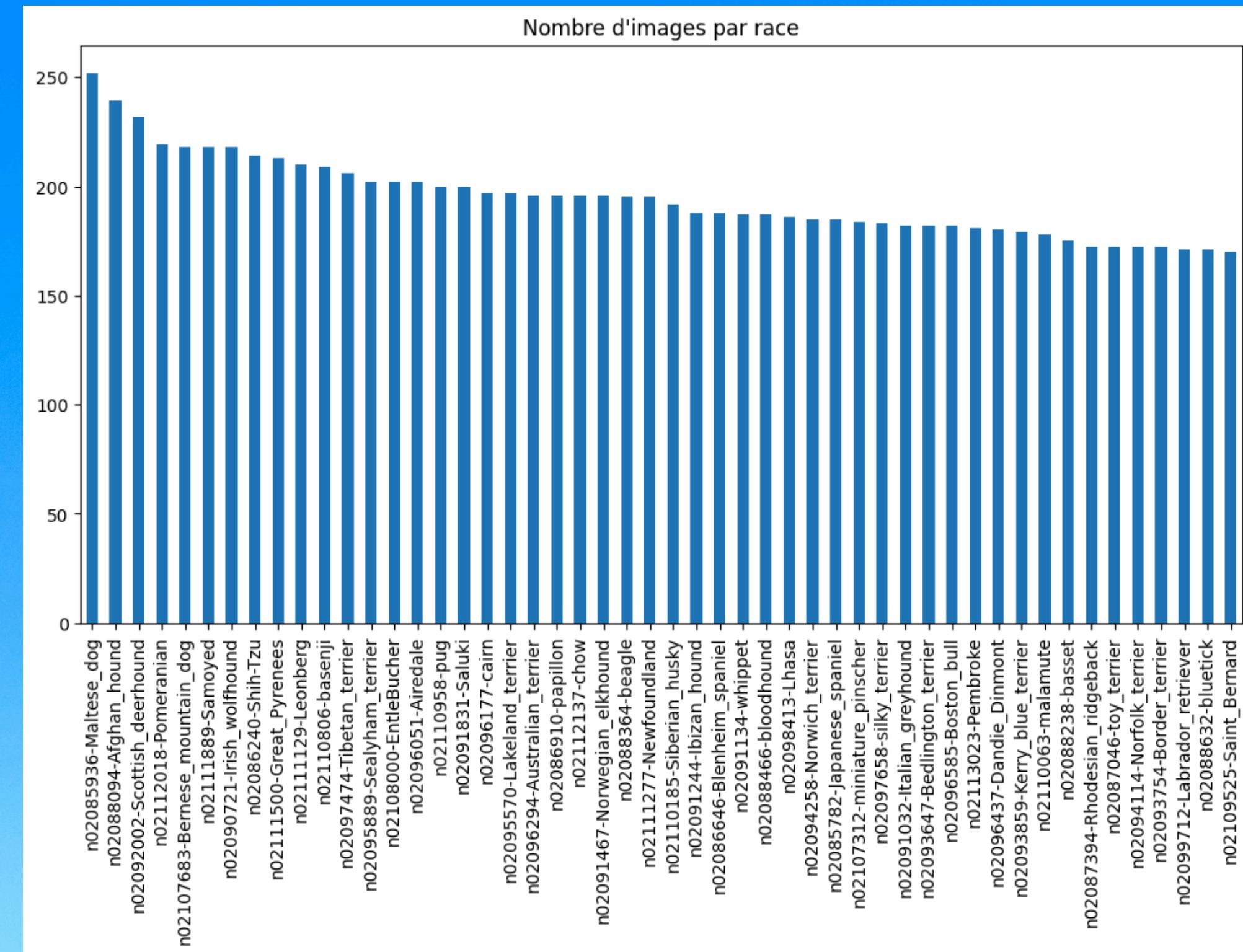
Objectifs du projet

- Mener une veille sur les techniques et outils récents en data science & IA.
- Mettre en place une preuve de concept.
- Développer un dashboard hébergé sur le cloud.

Dataset retenu

- Utilisation du site Stanford Dogs qui contient plus de 20000 images de chien.
- Récupération des images par race (1 dossier/race)
- Respect des règles RGPD

Dataset retenu



Exemple de données

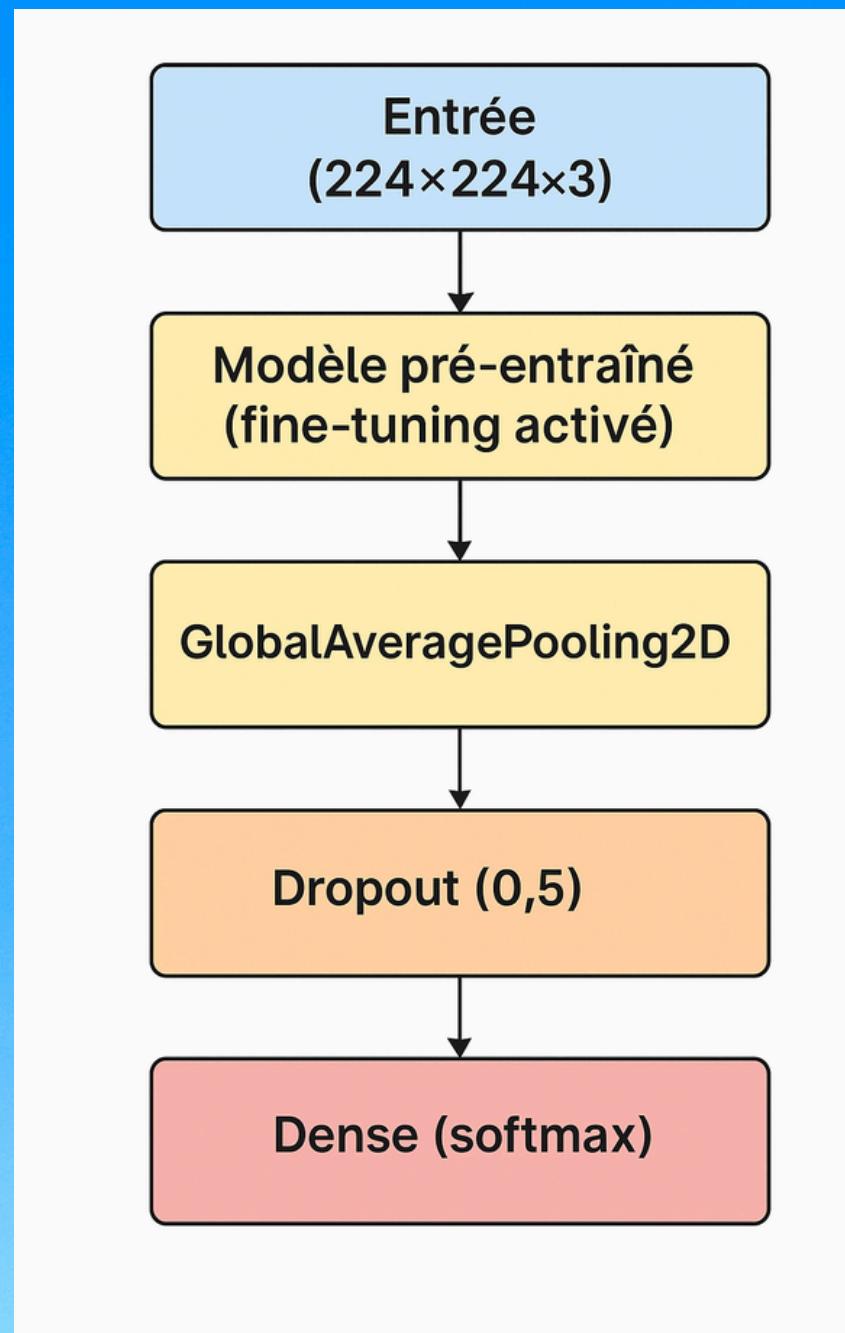
Données très diverses



Prétraitement et Data Augmentation

Type	Nom	Utilité
Transformations	Resize	Uniformiser la taille des entrées du modèle
	Equalization	Améliorer localement le contraste
	Normalisation	Centrer/Réduire les pixels
Augmentation	ShiftScaleRotate	Déplacer / Zoomer / Rotationner les images
	Vertical/HorizontalFlip	Apprendre l'invariance gauche/droite, haut/bas
	RandomBrightnessContrast	Simuler des conditions d'éclairage différentes
	MedianBlur	Rendre le modèle robuste au bruit

Baseline : ResNet-50 pré-entraîné



Hyperparamètres :

Optimizer : Adam

Learning rate : 1e-5

Batch size : 32

Epochs : 30

Dropout rate : 0.5

Modèle retenu : EfficientNetV2

Article pivot :

“EfficientNetV2: Smaller Models and Faster Training” de
Mingxing Tan et Quoc V. Le.

<https://arxiv.org/pdf/2104.00298v3>

Modèle retenu : EfficientNetV2

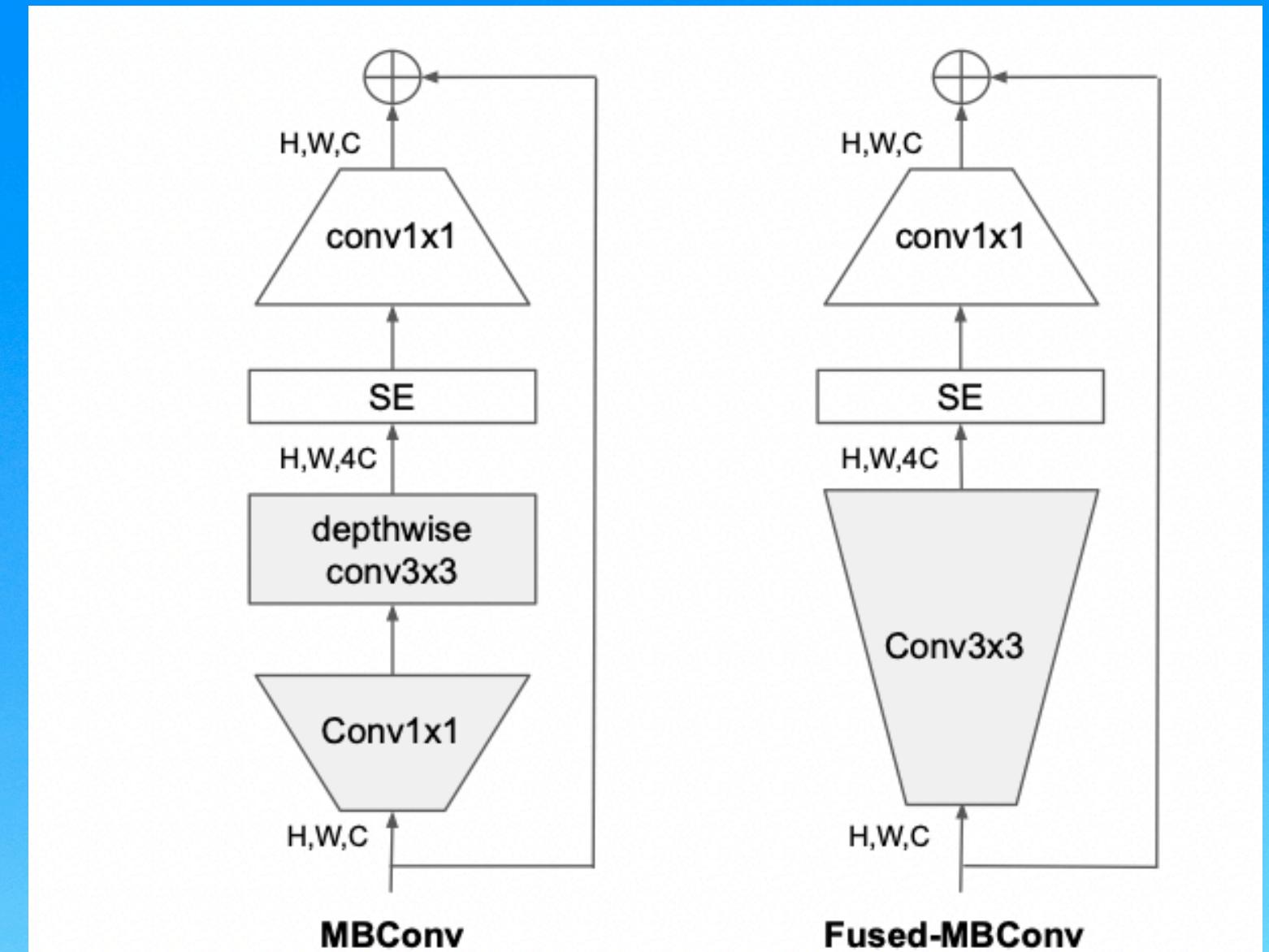
Pourquoi ce modèle :

- Plus adapté à ce petit dataset qu'un Vision Transformer.
(ViT classique ou Swim)
- Gain de précision attendu grâce a son architecture.
- Réduction du temps d'inférence grâce à des couches moins coûteuses et un nombre de FLOPs réduit.

Nouveaux concepts & techniques

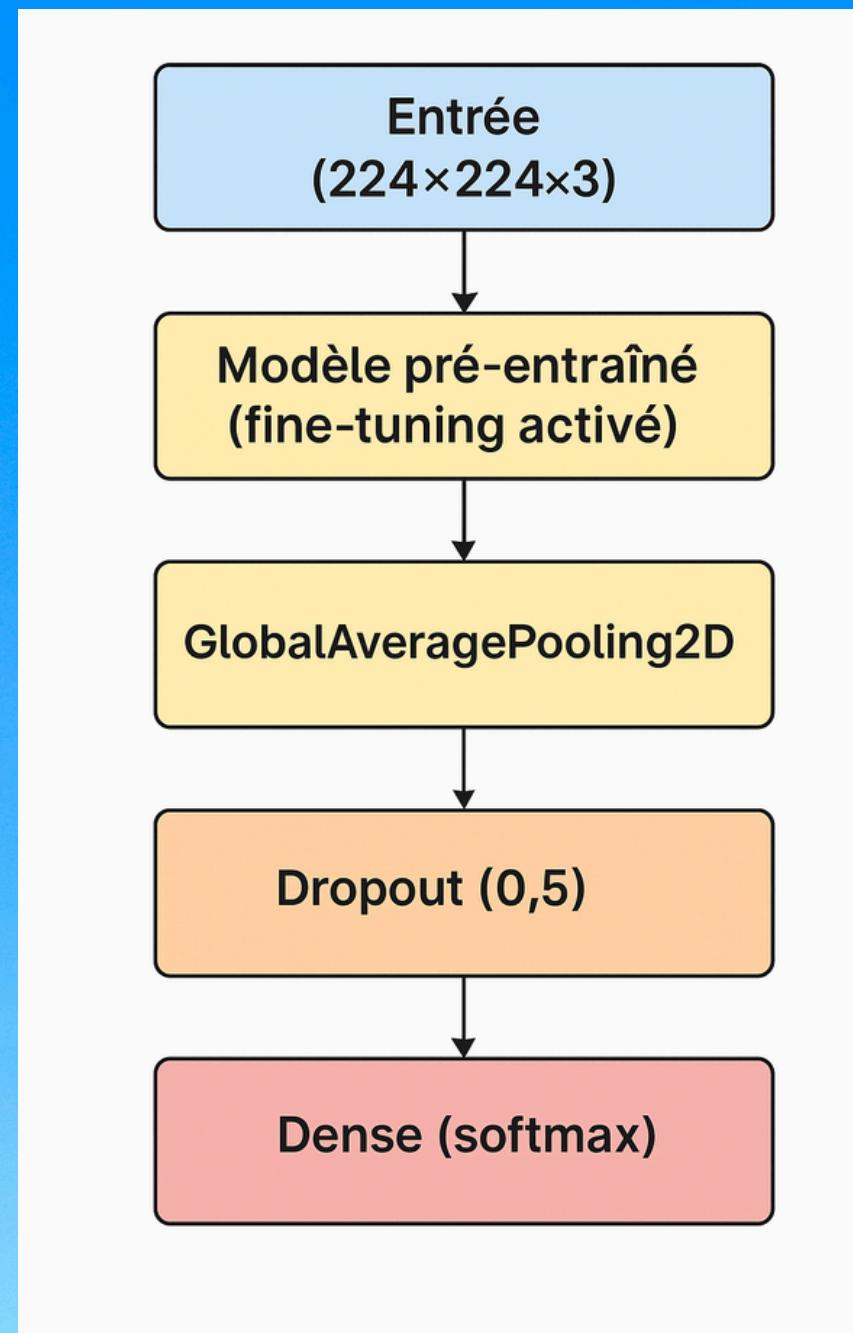
Principes du modèle :

- Nouveaux blocs Fused-MBConv pour remplacer les blocs MBConv
- Découvert par Training-Aware NAS



	Params (M)	FLOPs (B)	Top-1 Acc.	TPU imgs/sec/core	V100 imgs/sec/gpu
No fused	19.3	4.5	82.8%	262	155
Fused stage1-3	20.0	7.5	83.1%	362	216
Fused stage1-5	43.4	21.3	83.1%	327	223
Fused stage1-7	132.0	34.4	81.7%	254	206

Modèle : EfficientNetV2S



Hyperparamètres :

Optimizer : Adam
Learning rate : 1e-4
Batch size : 32
Epochs : 30
Dropout rate : 0.2

Résultats comparés

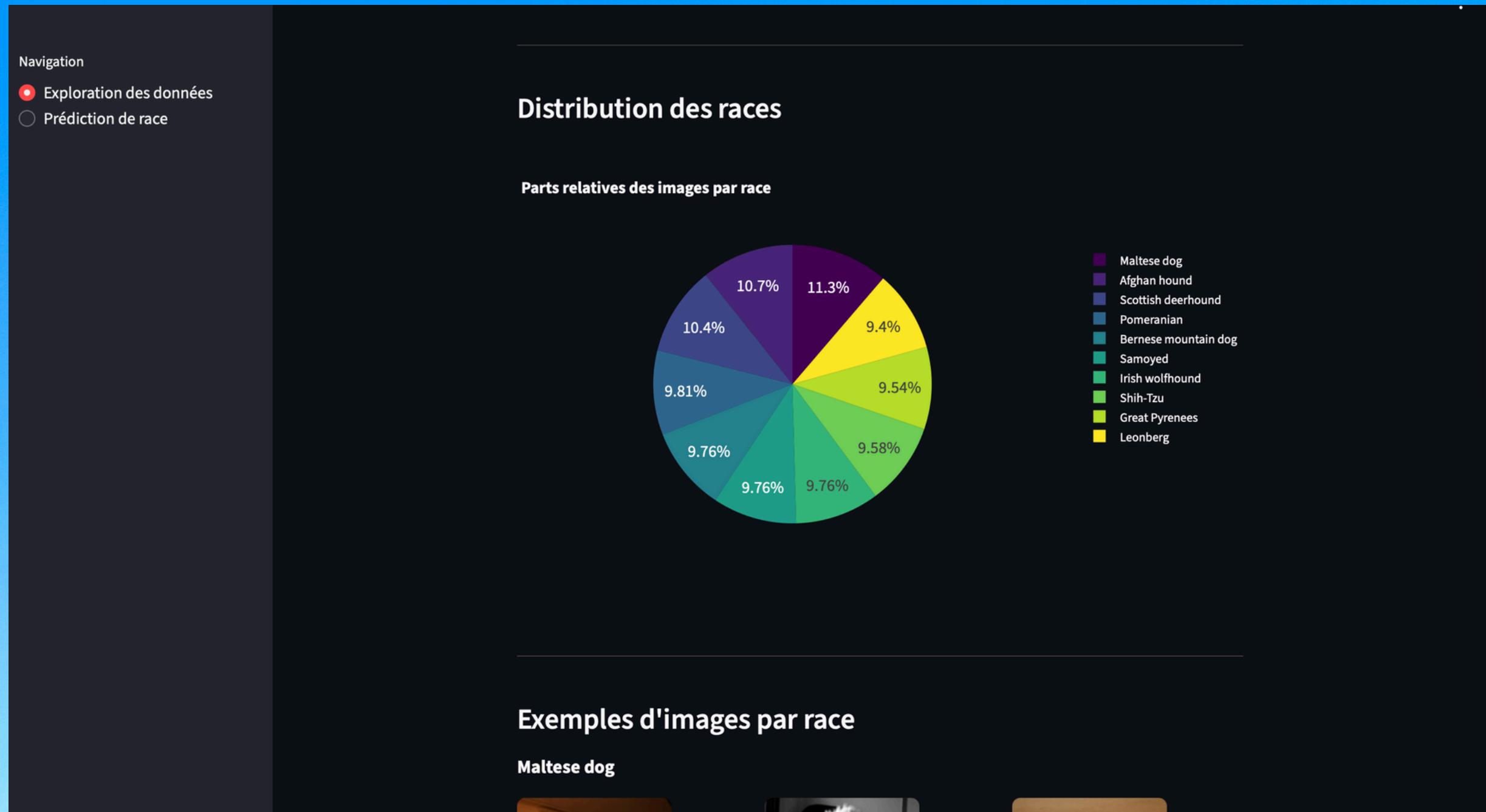
Modèle	ResNet-50	EfficientNetV2S
Accuracy Top-1	89.286%	95.536%
Accuracy Top-3	97.321%	99.554%
F1-macro score	0.8708	0.9321
Nombre de paramètres	23,6M	20,3M
Temps d'inférence moyen (GPU)	219,8 ms	168,3 ms

Dashboard

- Conçu et déployé avec le framework Streamlit pour une interface interactive.
- Déploiement manuel sur Azure App Service via une image Docker construite localement
- Image Docker hébergée sur DockerHub

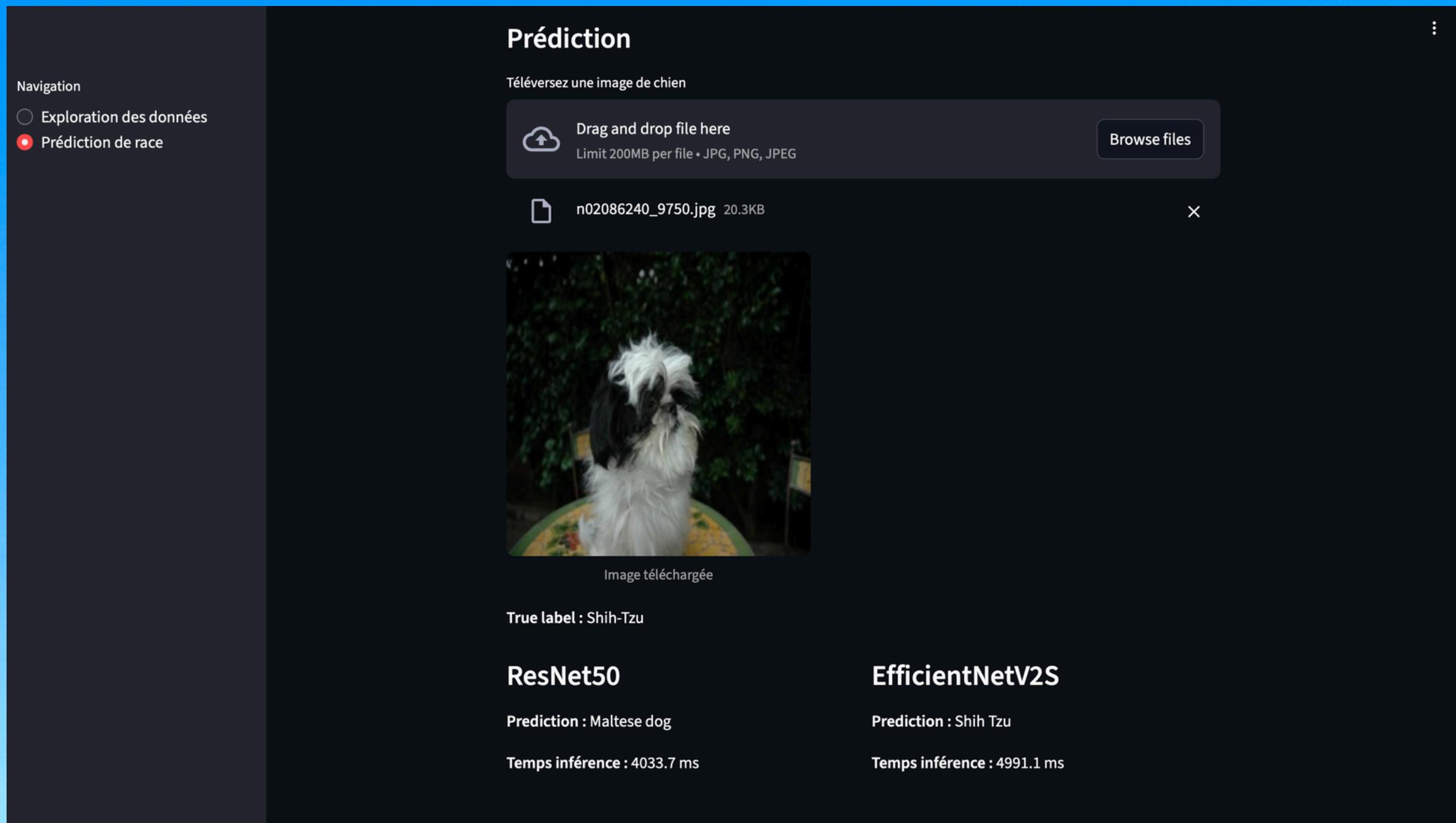
Dashboard

- L'interface d'exploration :



Dashboard

- L'interface de prédition :



Conclusion

- **Objectif** : Mettre en place une preuve de concept qui améliore le modèle d'un projet précédent.
- **Résultats** : Le modèle EfficientNetV2S plus performant et plus léger que ResNet-50, temps d'inférence plus court sur GPU
- **Limites & améliorations** : Temps d'inférence sur CPU plus long que ResNet-50. Progressive learning permettrait d'améliorer les performances et de rendre le modèle plus robuste