

Classez des images à l'aide d'algorithmes de Deep Learning

- Le Refuge - Classification de races de chien

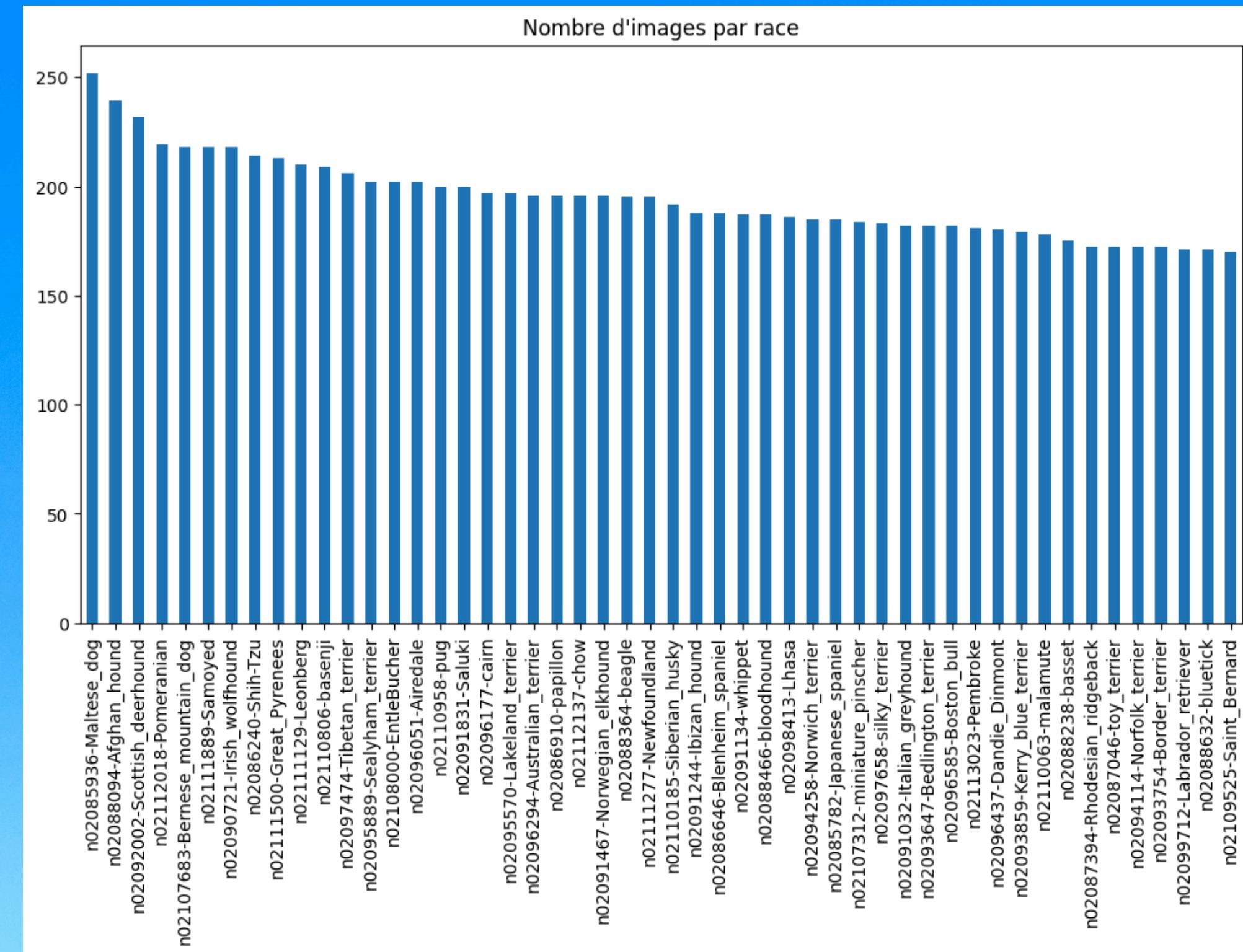
Objectifs du projet

- Développer un modèle CNN capable d'identifier automatiquement la race d'un chien à partir d'une image.
- Utiliser le transfer learning pour utiliser des modèles pré-entraînés avec de meilleurs résultats.
- Déployer le modèle sur une API Streamlit locale.

Collecte des données

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

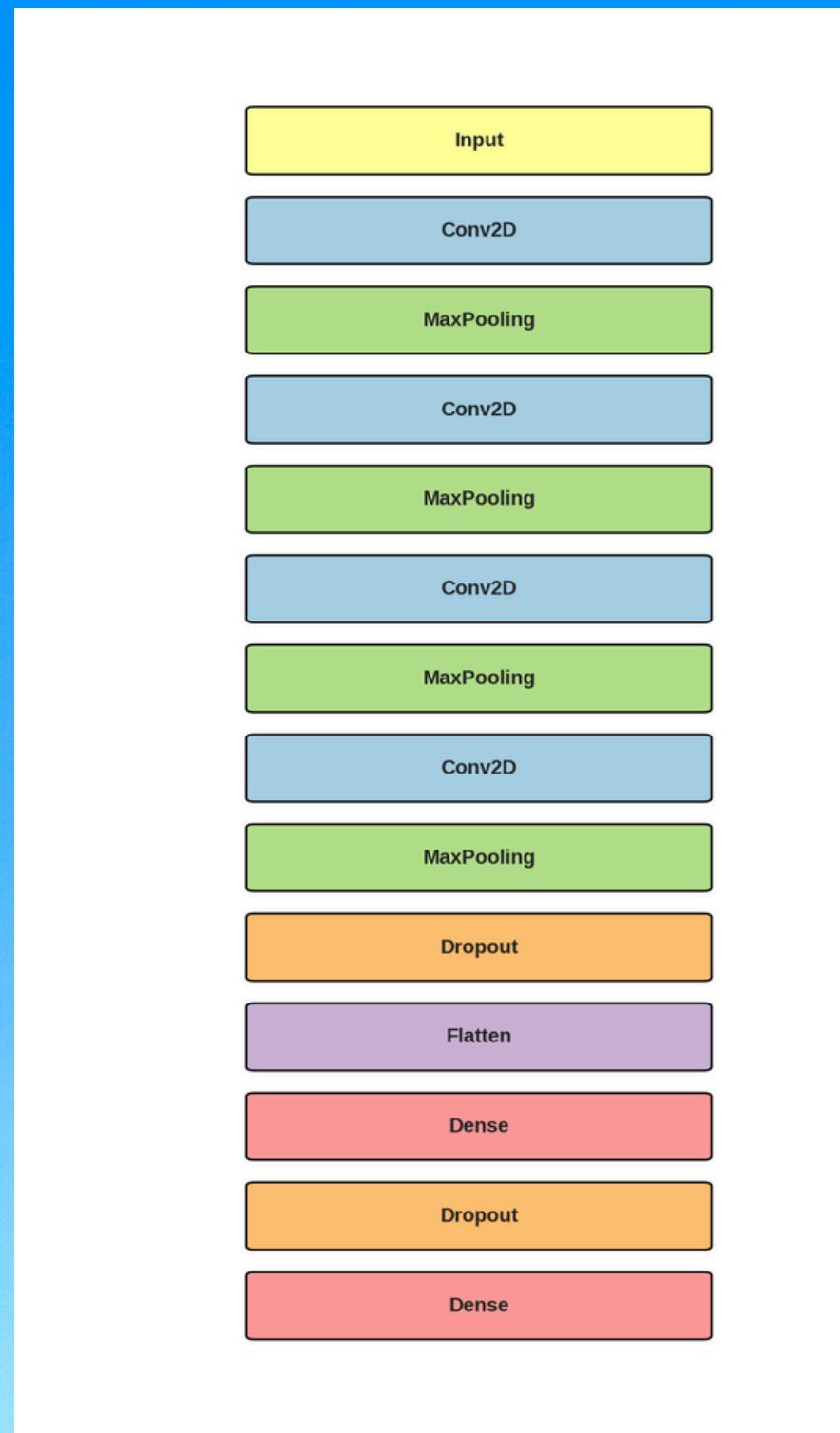
Analyse exploratoire



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
	HorizontalFlip	Apprendre l'invariance gauche/droite
	RandomBrightnessContrast	Simuler des conditions d'éclairage différentes
	MedianBlur	Rendre le modèle robuste au bruit

Architecture de mon modèle CNN personnalisé



- Entrée : image $224 \times 224 \times 3$
- 4 blocs Conv2D + MaxPooling
- Dropout
- Flatten
- Dense (200 neurones)
- Dropout
- Dense finale (Softmax)

Entraînement et test du CNN personnel

Stratégie d'entraînement :

- loss = sparse_categorical_crossentropy
- metrics = accuracy
- validation_split = 0.2

Grille de paramètres testés :

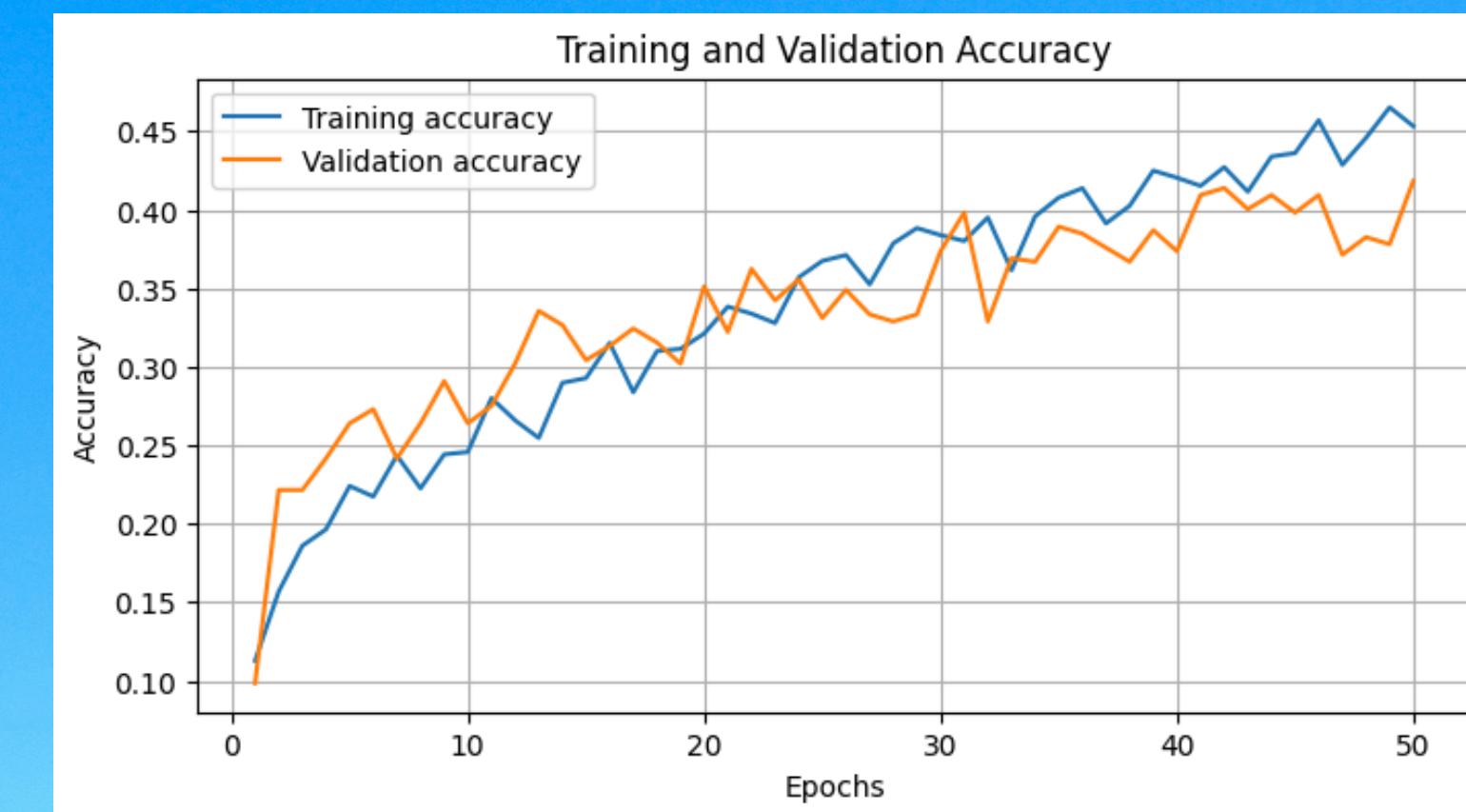
- Learning rate
- Dropout rate
- Fonction d'activation
- Optimizers
- Epochs

Courbes du CNN personnel

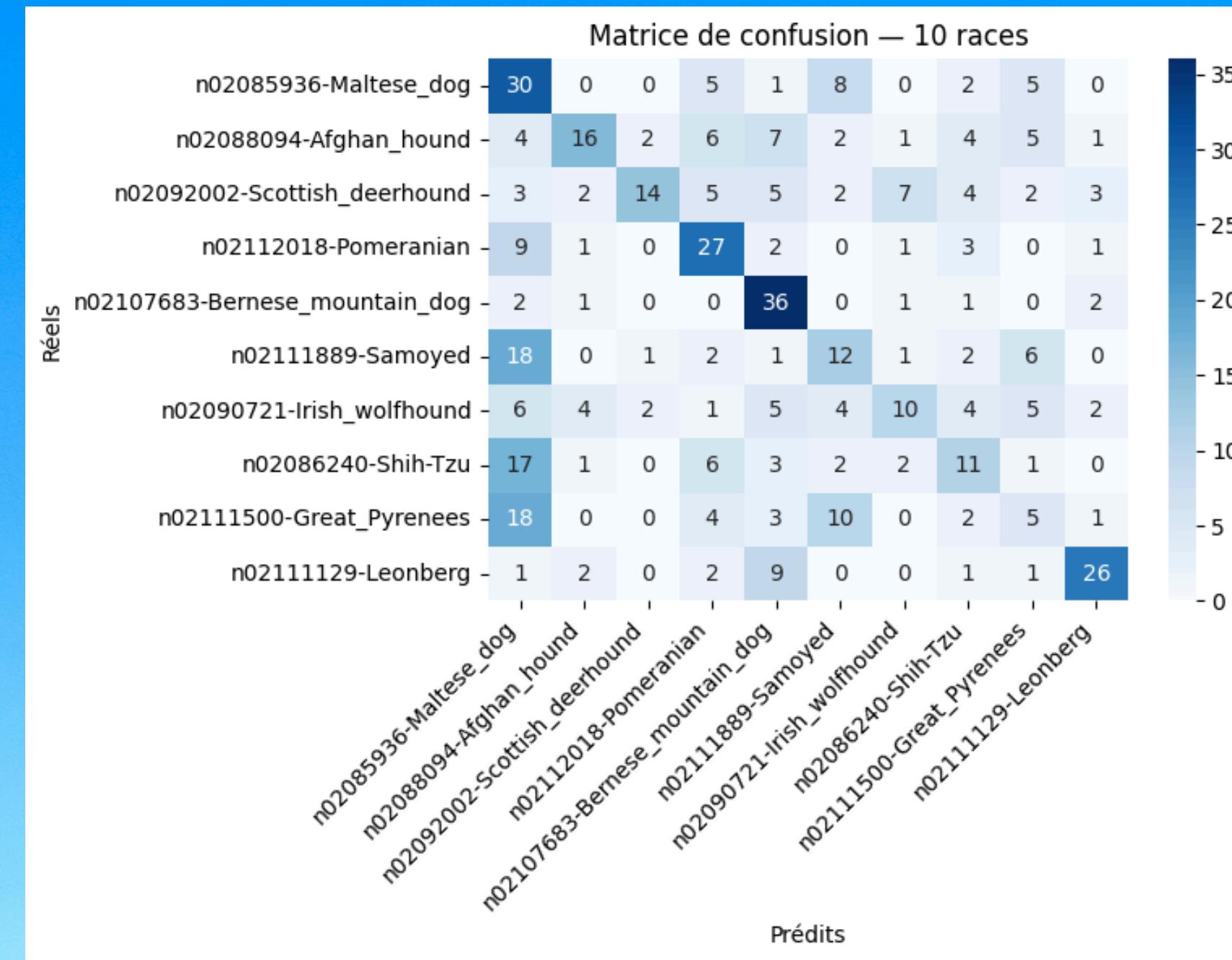
- Loss :



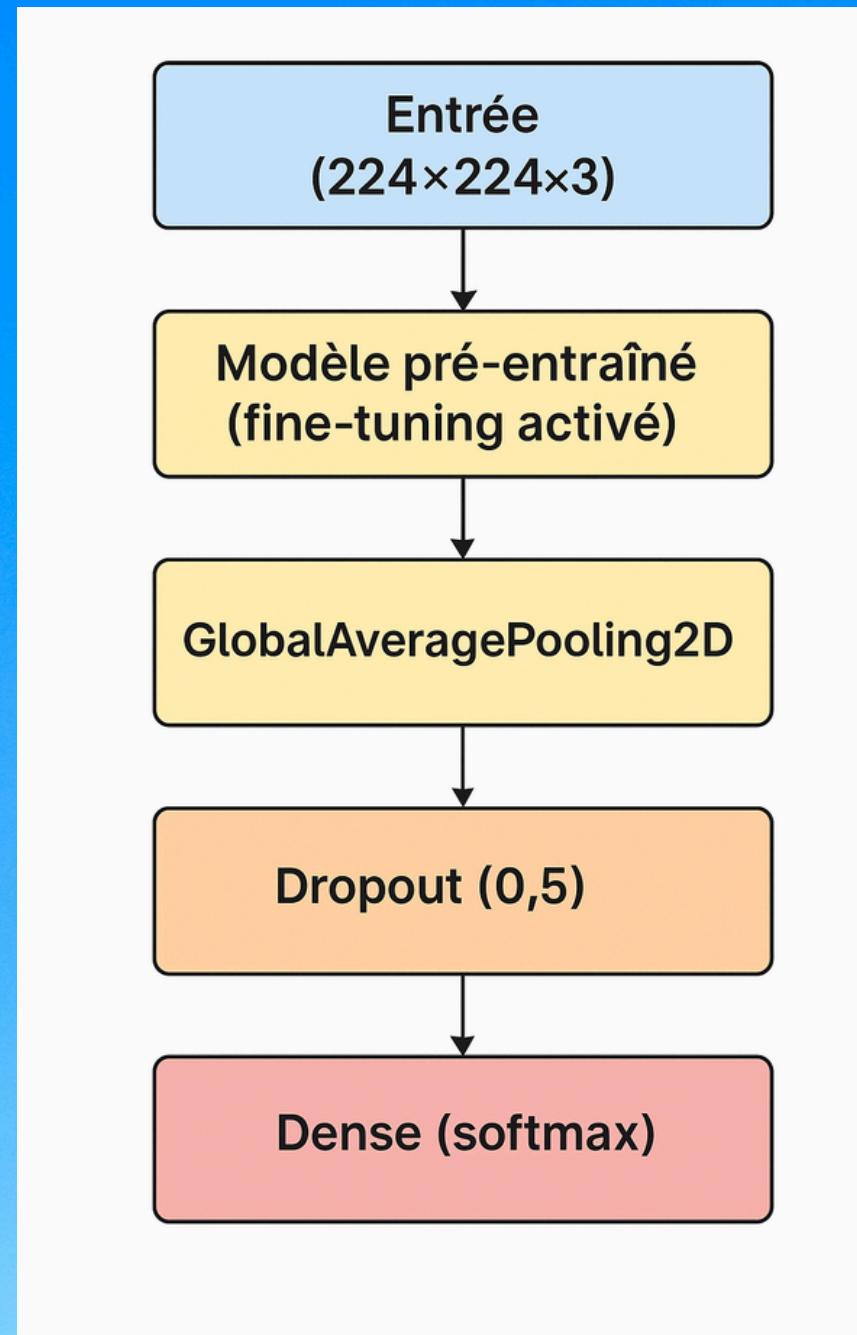
- Accuracy :



Matrice de confusion du CNN Personnel



Transfer Learning



Modèles pré-entraînés testés :

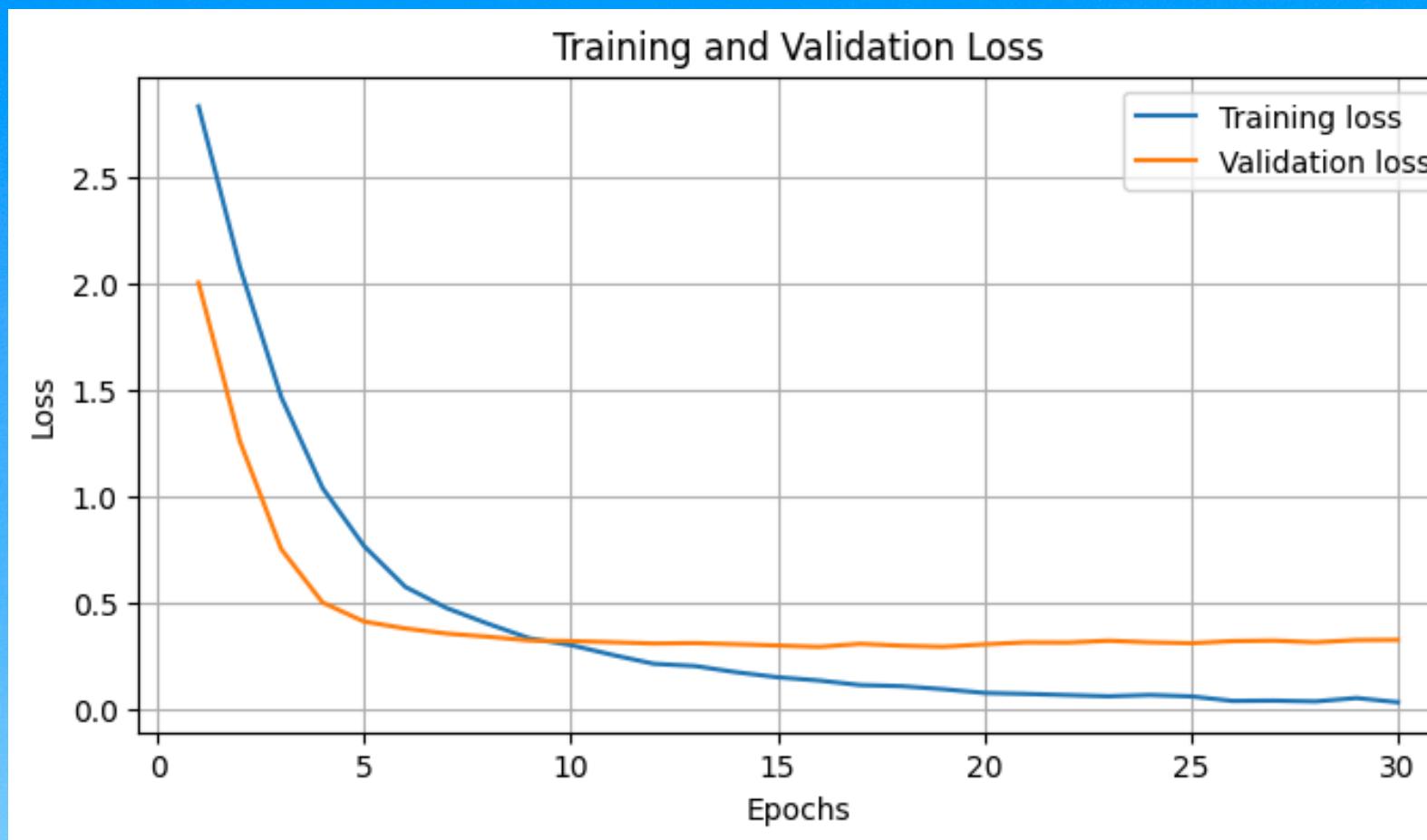
- **VGG16**
- **ResNet50**
- **InceptionV3**

Comparaison des résultats

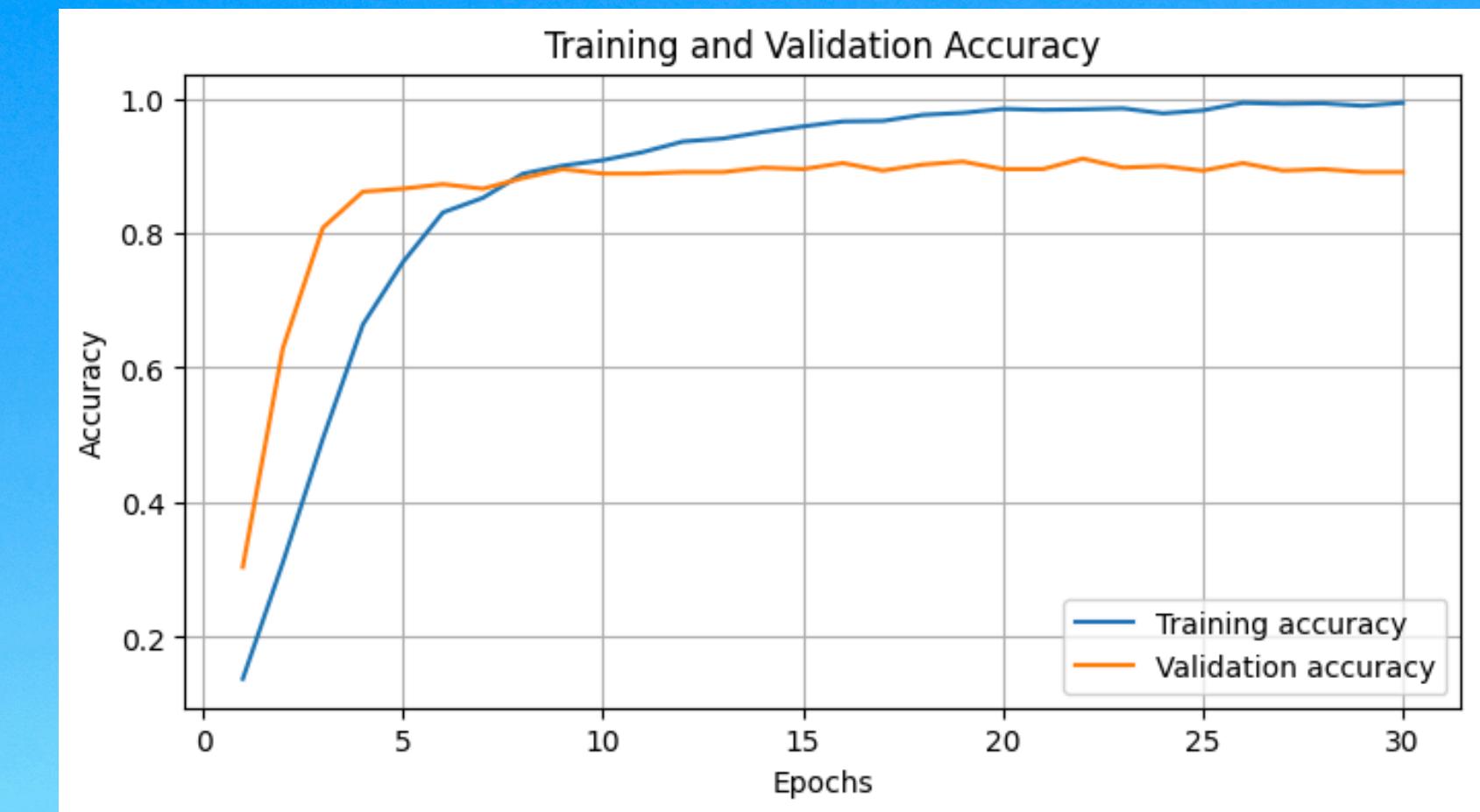
Modèle	Accuracy	Loss
Mon CNN Personnalisé	45.638%	1.651
ResNet50	91.499%	0.265
InceptionV3	88.143%	0.309
VGG16	74.944%	0.722

Courbes de ResNet50

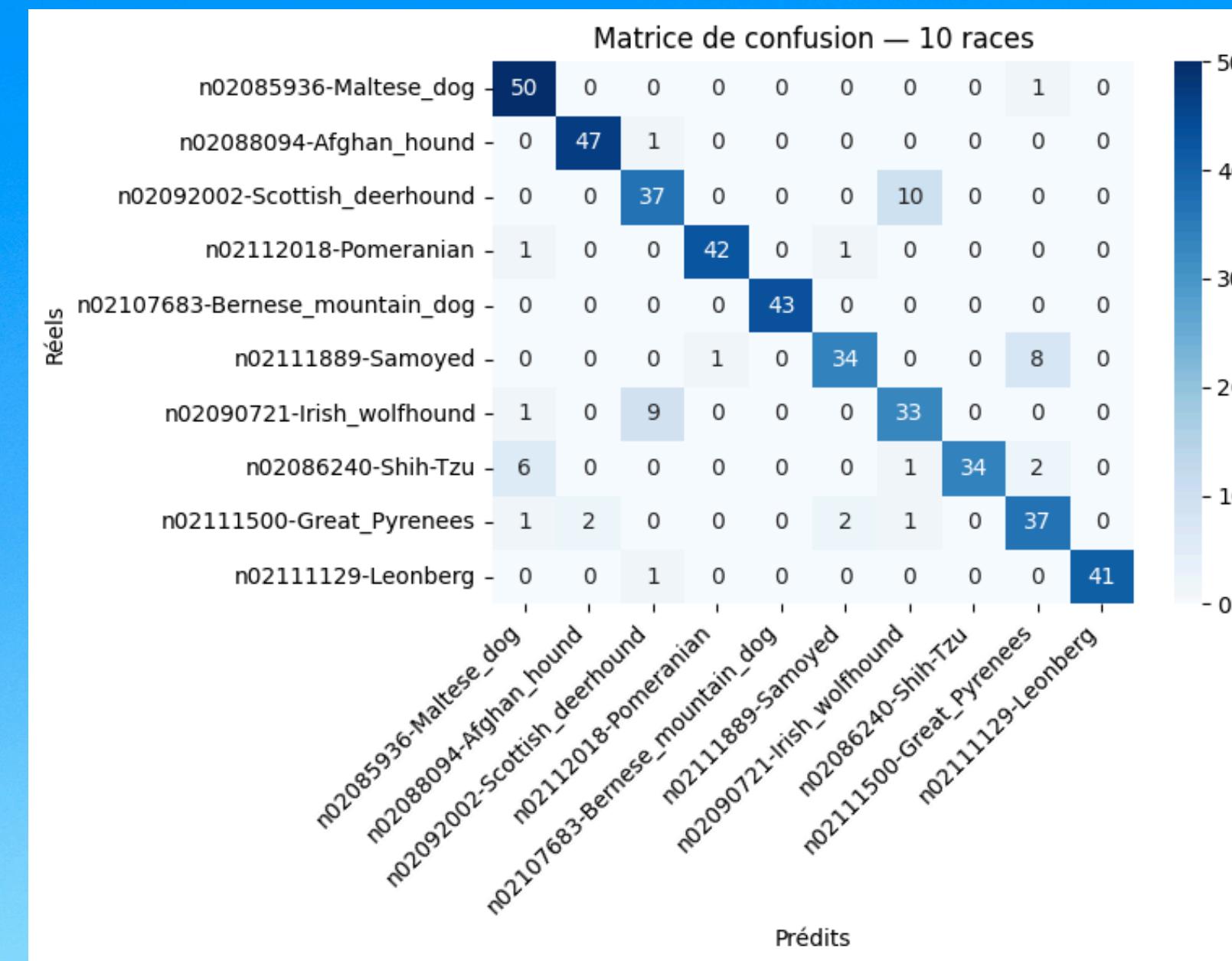
- **Loss**



- **Accuracy**



Matrice de confusion de ResNet50



Deploiement



The screenshot shows a Streamlit deployment interface with a dark theme. At the top right, there are buttons for "Deploy" and a menu. The main title is "Évaluation de la race de chiens" with a dog icon. Below the title, it says "Cette application Streamlit:" followed by a list of steps:

1. Charge votre modèle (`mon_modele.h5`).
2. Lit le CSV `test_labels.csv` (colonnes obligatoires : `filename` , `true_label`).
3. Parcourt automatiquement **toutes** les images contenues dans le dossier `test_images/` .
4. Pour chaque image, affiche :
 - o la vignette de la photo,
 - o la race réelle (depuis le CSV),
 - o la race prédictive (par le modèle).

1. Uploadez votre fichier CSV de labels réels

Fichier CSV (colonnes obligatoires : `filename` , `true_label`)

Drag and drop file here
Limit 200MB per file • CSV

Browse files

Commencez par uploader votre fichier CSV de labels réels (`test_labels.csv`).

Conclusion

- **Objectif** : Développer un modèle qui prédit la race d'un chien a partir d'une image
- **Résultats** : Approche par transfer learning bien plus performante qu'un CNN Personnel
- **Limites** : Confusion sur certaines races proches, léger overfitting sur mon transfer learning, réduction à 10 races.