

CLASSEZ DES IMAGES À L'AIDE  
D'ALGORITHMES DE DEEP LEARNING

---

PROJET IML 6

# SOMMAIRE

- ▶ La problématique et son Interprétation
- ▶ Cleaning effectué
- ▶ Exploration
- ▶ Feature engineering
- ▶ Modèle VGG-16 en « transfert learning »
- ▶ Modèle ResNet50 en « transfert learning »
- ▶ Mon Modèle CNN
- ▶ Data augmentation appliqué à mon modèle
- ▶ Conclusions
- ▶ Axes d'améliorations

# LA PROBLÉMATIQUE ET SON INTERPRÉTATION

- ▶ Problématique d'une association de protection des animaux
  - ▶ Base de données des pensionnaires trop lourde à gérer manuellement
- ▶ Besoin
  - ▶ Pouvoir référencer les images des pensionnaires automatiquement par race de chien,
- ▶ Comment
  - ▶ réaliser un algorithme de détection de la race du chien sur une photo, afin d'accélérer leur travail d'indexation
  - ▶ Avec réseaux de neurones convolutifs CNN: reconnu comme efficace et certains pré-entraînés



CNN

BOSTON\_BULL

# CLEANING EFFECTUÉ

## ► Base de données : ImageNetDogs Stanford Dogs Dataset

- Train : 100 images / race
- Test : 100 à 150 images / race

## ► Sélection de 3 et 10 races

### ► 3 races

- Bernese Mountain dog →



- Afghan Hound →



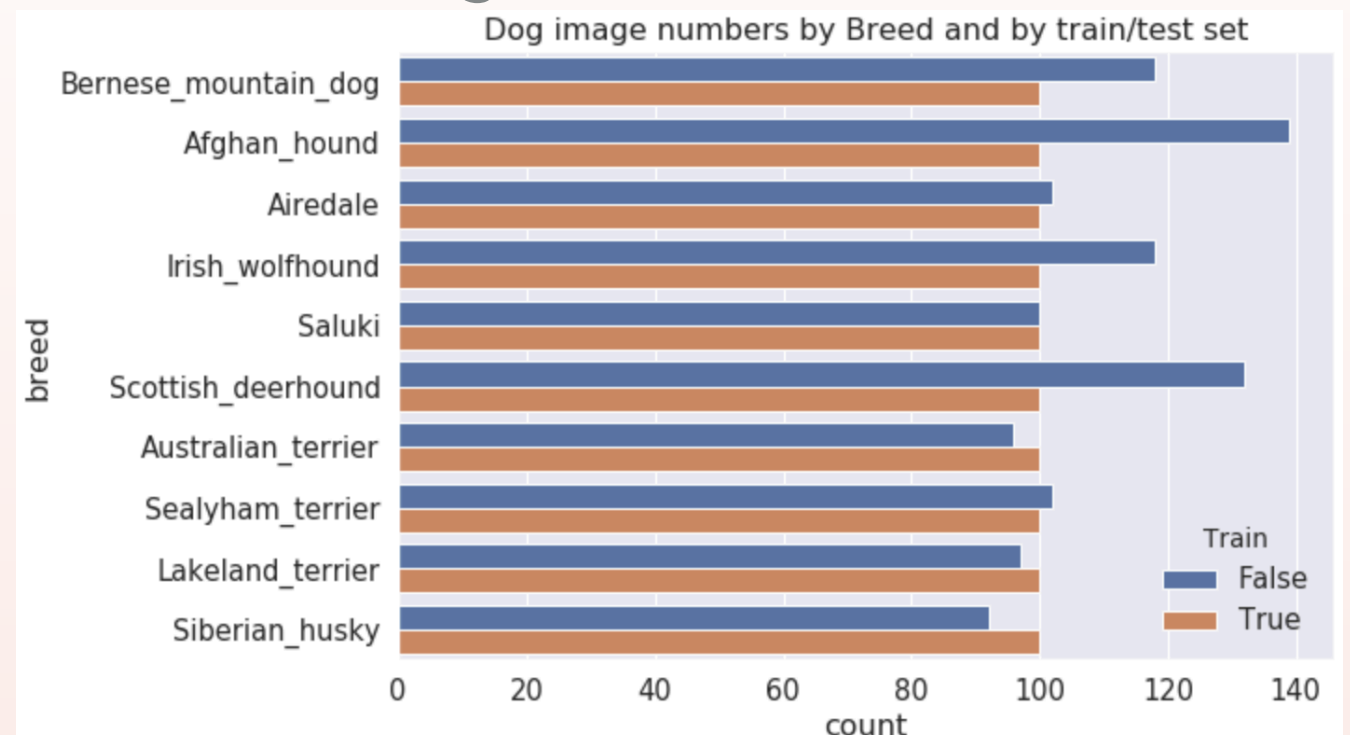
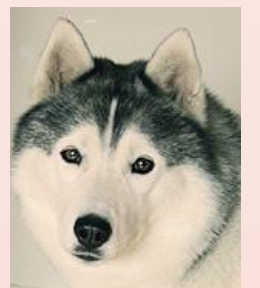
- Airedale →



### ► 10 races

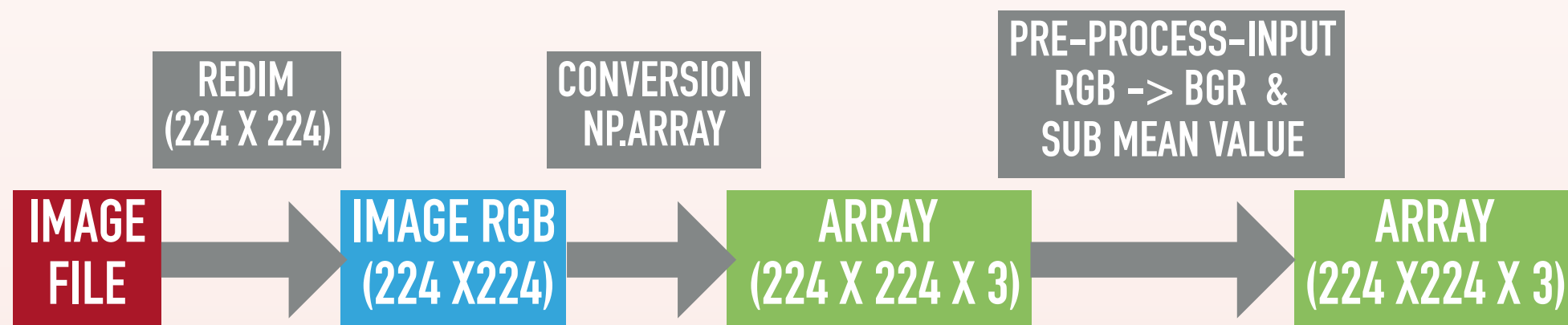
- + Irish Wolfhound, Saluki, Scottish Deerhound, Australian Terrier, Sealyham Terrier, Lakeland Terrier, Siberian Husky

►



## FEATURE ENGINEERING

- ▶ CNN => recherche de feature non nécessaire
- ▶ Pre-processing



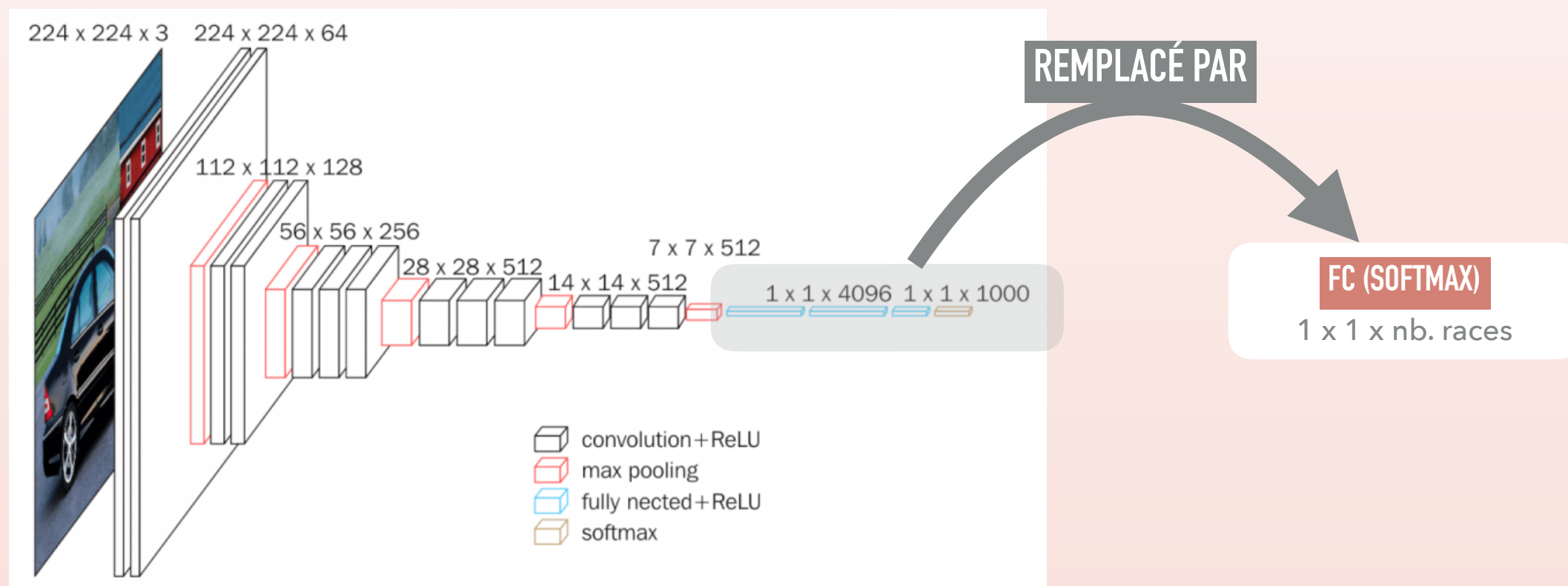
- ▶ 3 architectures de CNN testés
  - ▶ 2 modèles pré-entraînés sur ImageNet VGG-16 et ResNet50
  - ▶ 2 modèles « transfert learning » en ré-entraînant leur dernière couche (FC)
  - ▶ 1 modèle « from scratch » à entraîner en totalité

# MODÈLE VGG-16

- ▶ Transfert learning pré-entraîné sur ImageNet
  - ▶ Remplacement des 3 derniers layers Fully connected par : 1 layer FC
    - ▶ Nb. unités = Nb race de chien à prédire
    - ▶ Activation : SOFTMAX
    - ▶ Ré-entraînement dernier layer (75 000 à 250 000 param.)

## NB. PARAMS (EX: 3 RACES DE CHIEN)

Total params: 14,789,955  
Trainable params: 75,267  
Non-trainable params: 14,714,688





# MODÈLE RESNET-50

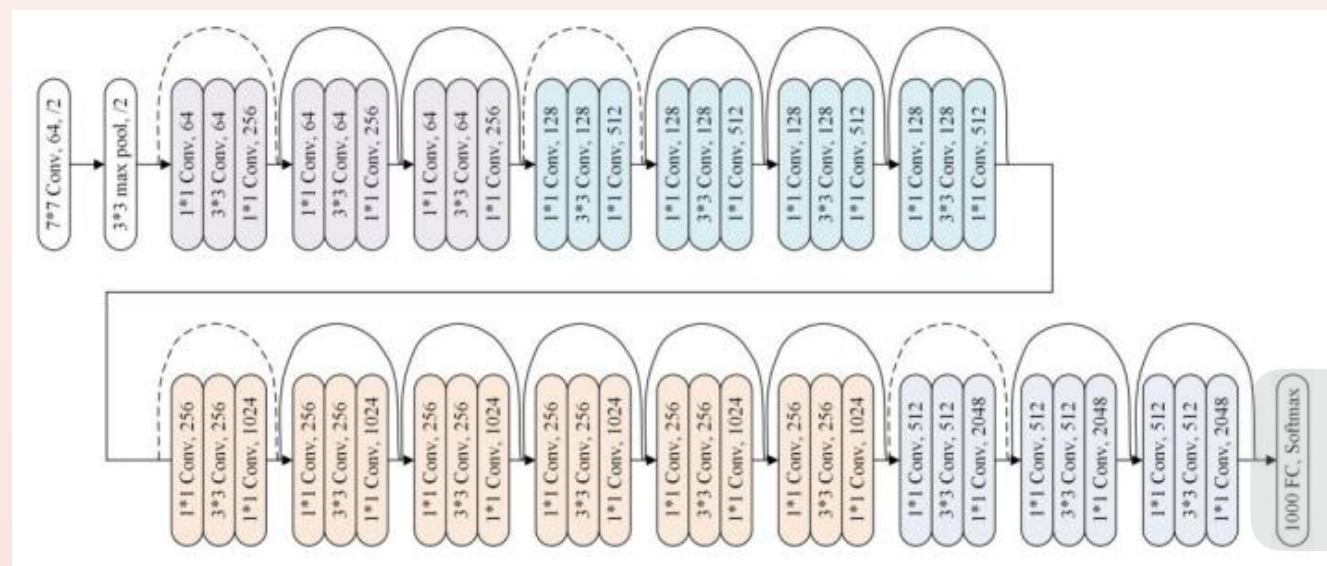
- ▶ Transfert learning pré-entraîné sur ImageNet
  - ▶ Remplacement dernier layer Fully connected par : 1 layer FC
- ▶ Nb. unités = Nb race de chien à prédire
- ▶ Activation : SOFTMAX
- ▶ Ré-entraînement dernier layer (300 000 à 1 000 000 param.)

## NB. PARAMS (EX. 3 RACES DE CHIEN)

Total params: 23,888,771

Trainable params: 301,059

Non-trainable params: 23,587,712



REPLACÉ PAR

FC (SOFTMAX)

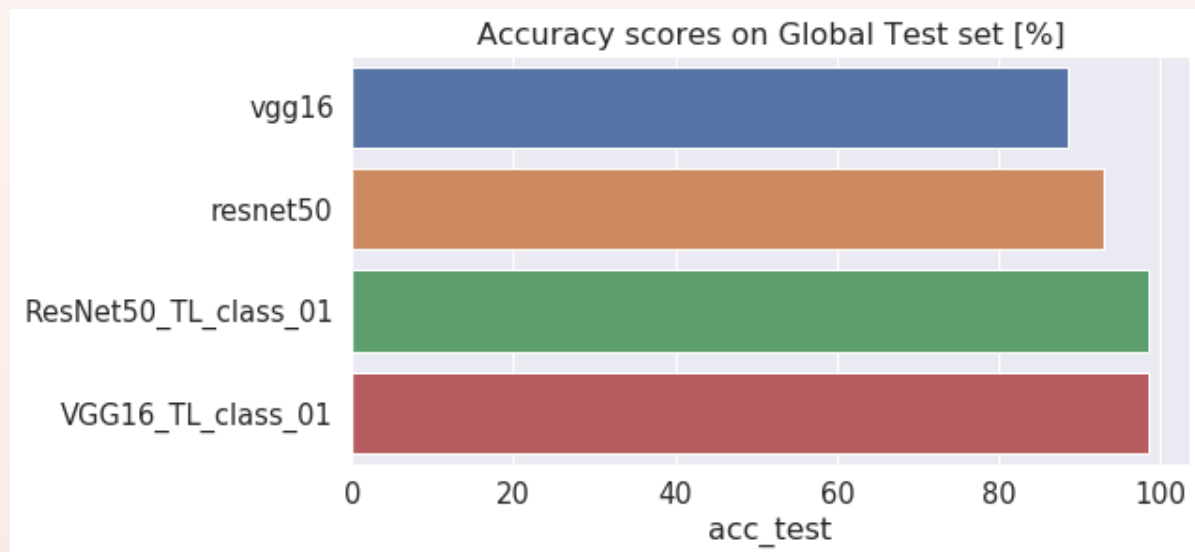
1 x 1 x nb. races

# COMPARAISON TRANSFERT LEARNING SUR TEST SET

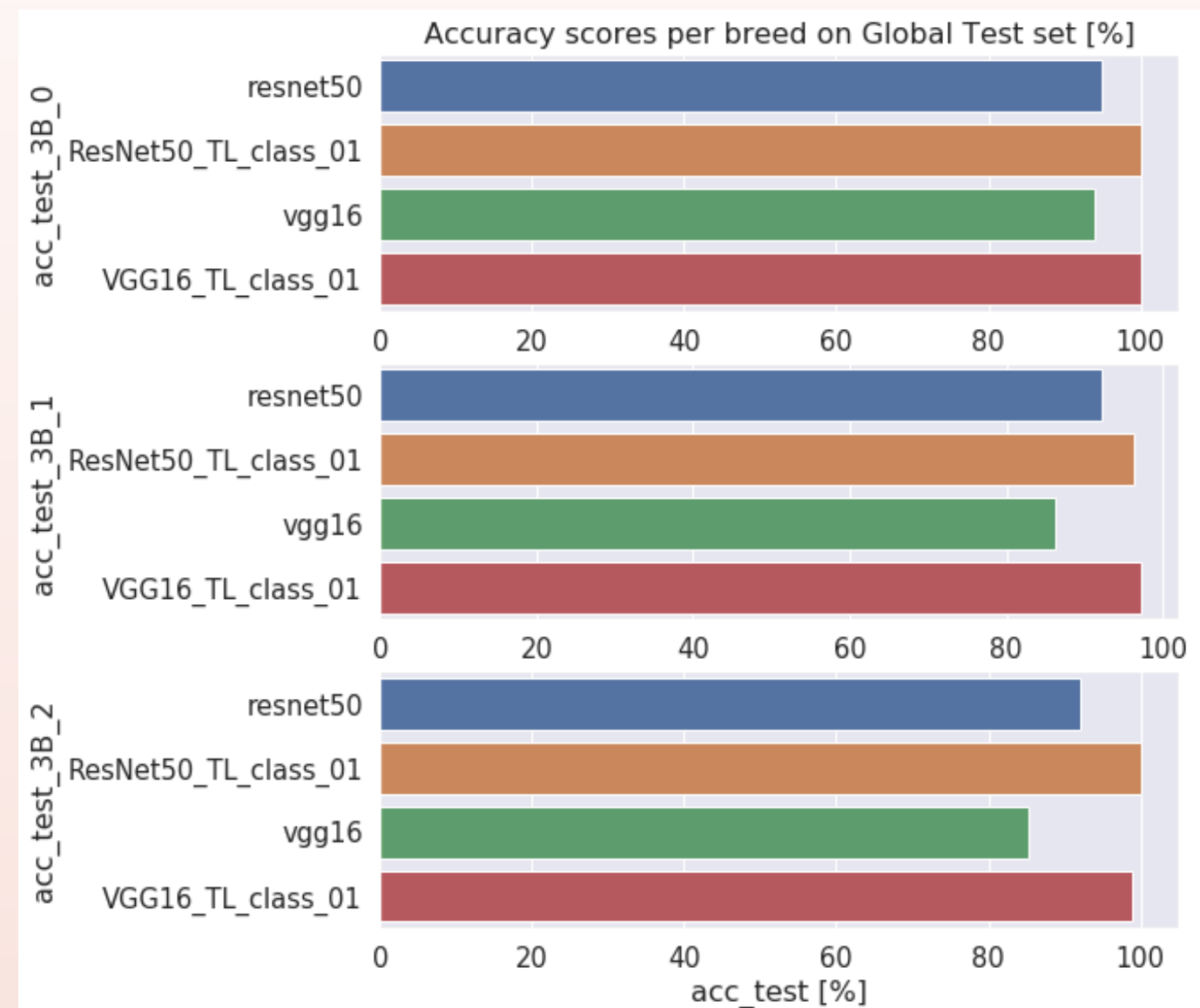
## ► 3 races : meilleur accuracy en transfert learning

### ► ResNet-50 TL & VGG-16 TL équivalents

**ACCURACY 98.6%**



**TRÈS BONNE ACCURACY PAR CLASSE**

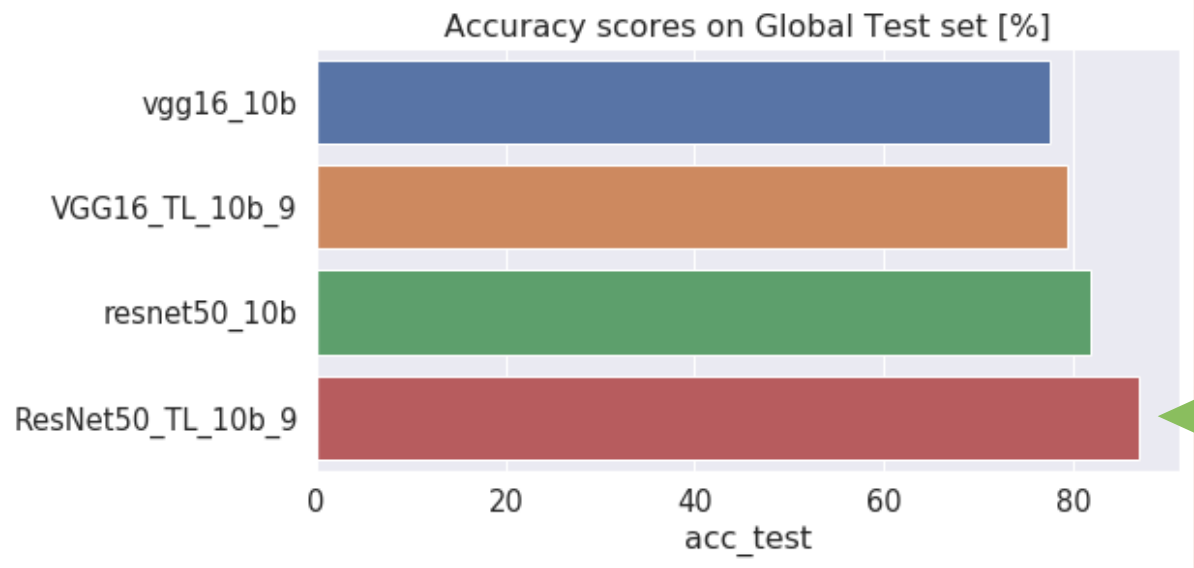




# COMPARAISON TRANSFERT LEARNING SUR TEST SET

► 10 races

► ResNet-50 TL meilleur que VGG-16 TL

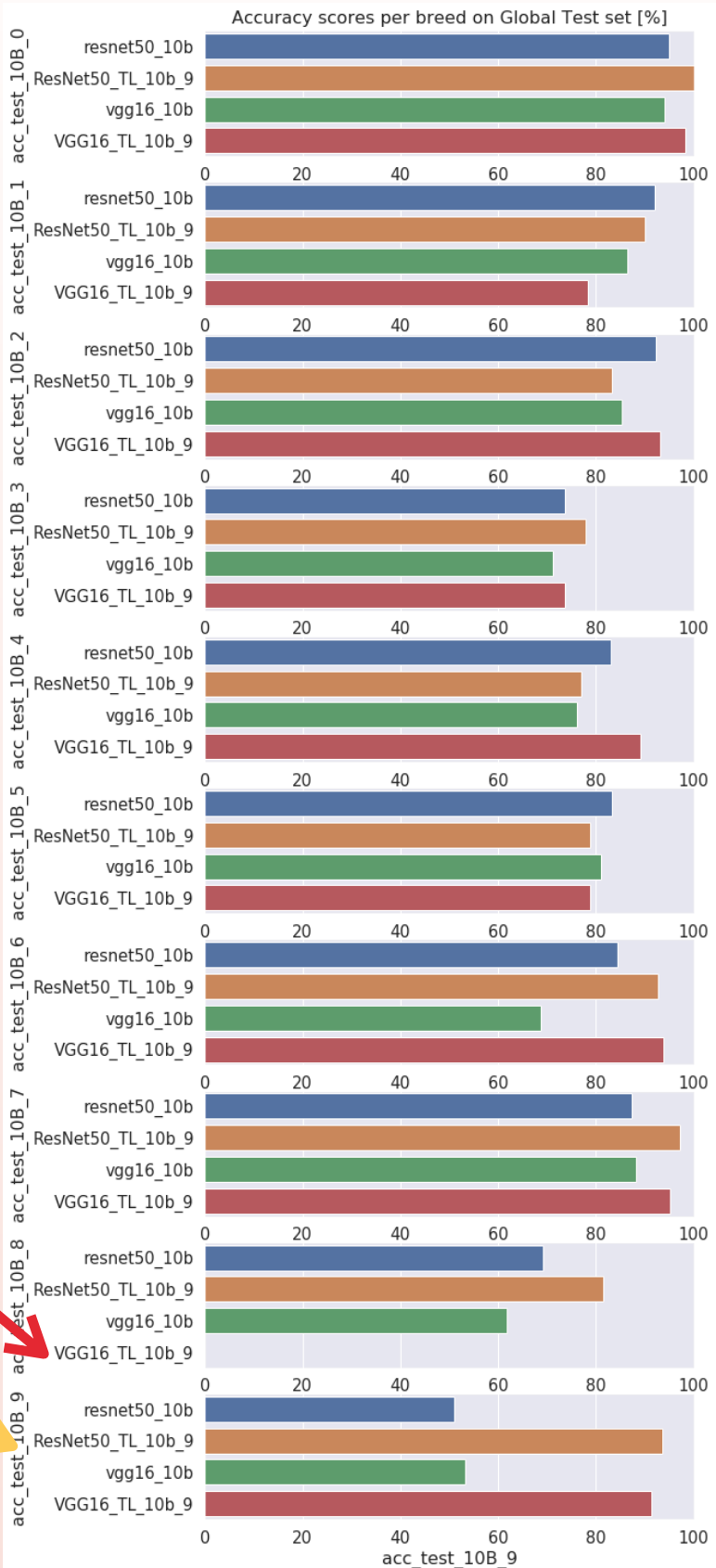


← ACCURACY 87%

CONFUSION :  
LAKELAND TERRIER (8) -> AIREDALE (2)

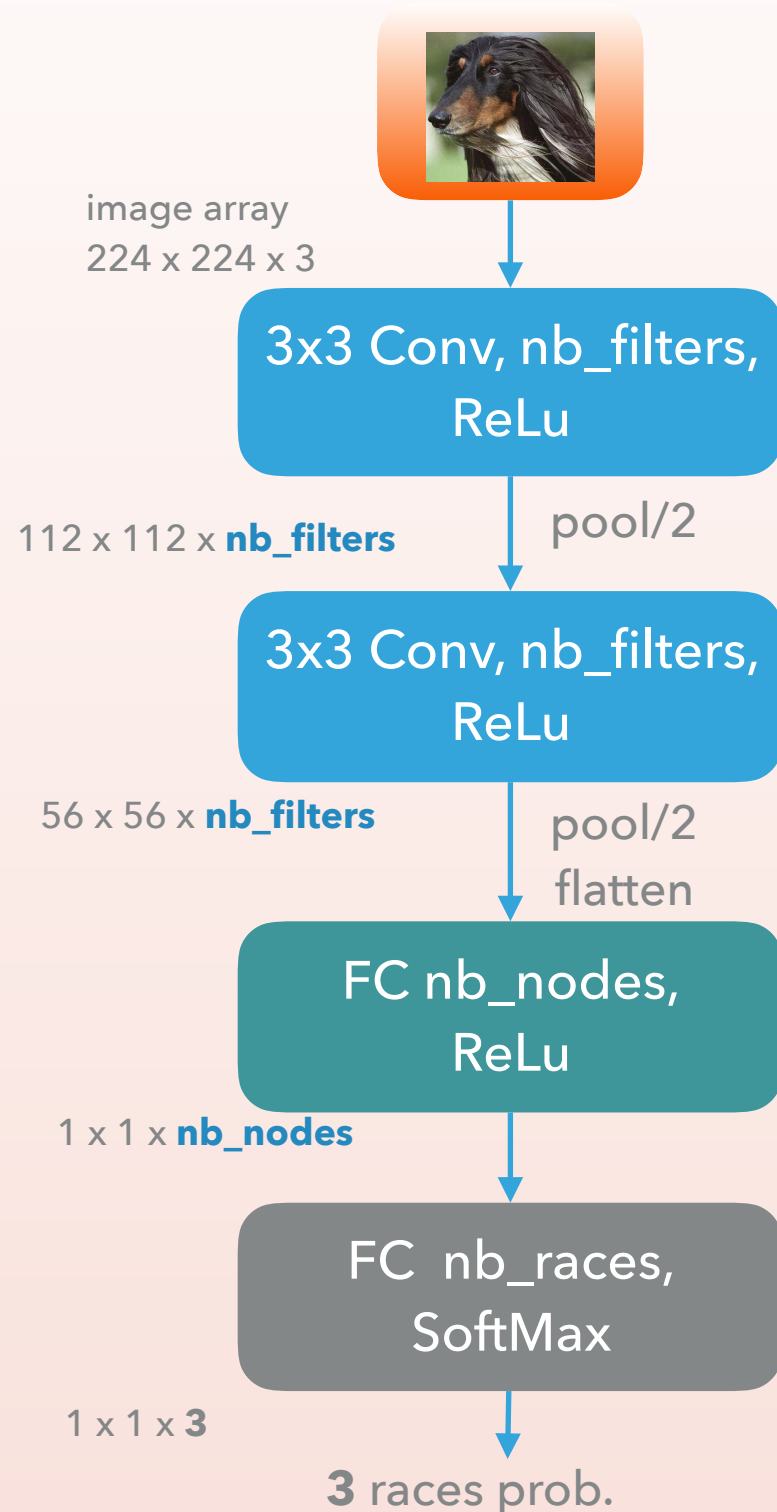


TL MEILLEUR SUR  
RACE TRÈS DIFFÉRENTE



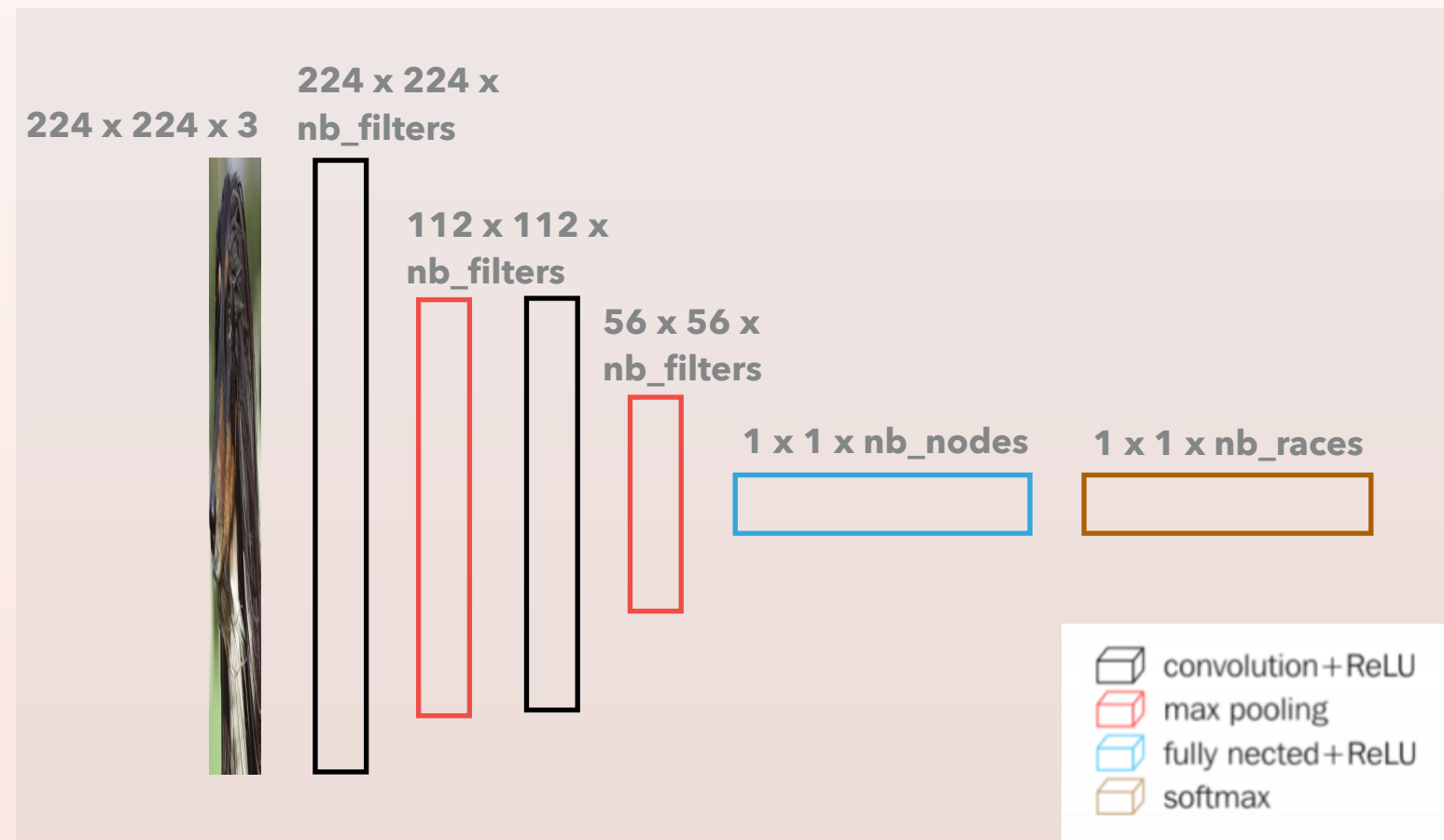
# MON MODÈLE CNN

## ► Architecture type VGG-16 simplifié



## 2 PARAMÈTRES D'ARCHITECTURE

- nb\_filters
- nb\_nodes

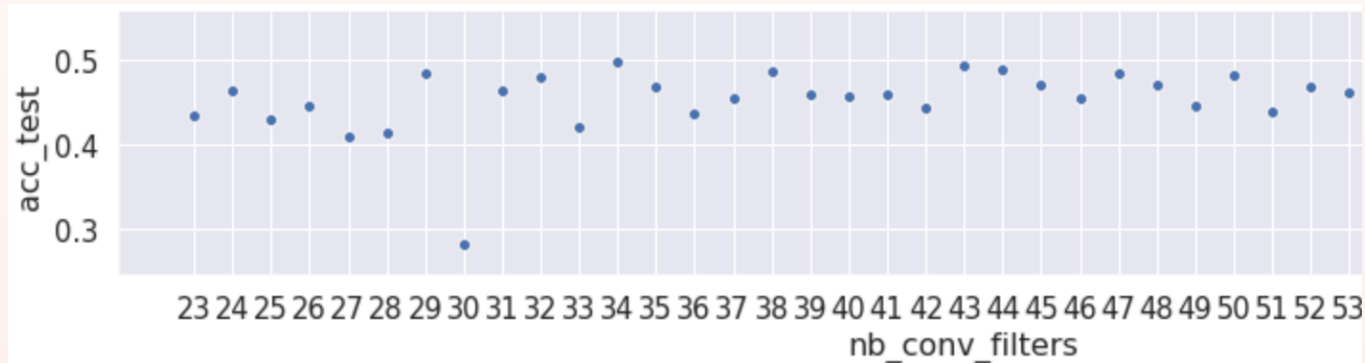


# MON MODÈLE CNN

- Initialisation aléatoire => Dimensionnement difficile

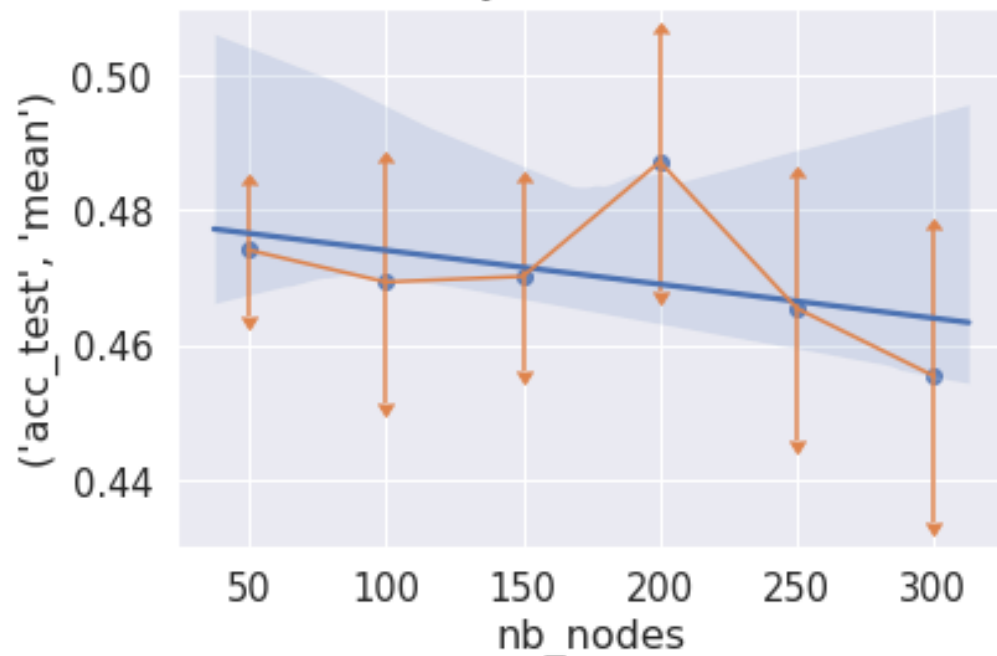
**INITIALISATION ALÉATOIRE => PLUS DE « RUN » NÉCESSAIRE**

**PLUS DE FILTRES DE CONVOLUTION => PLUS D'ACCURACY**

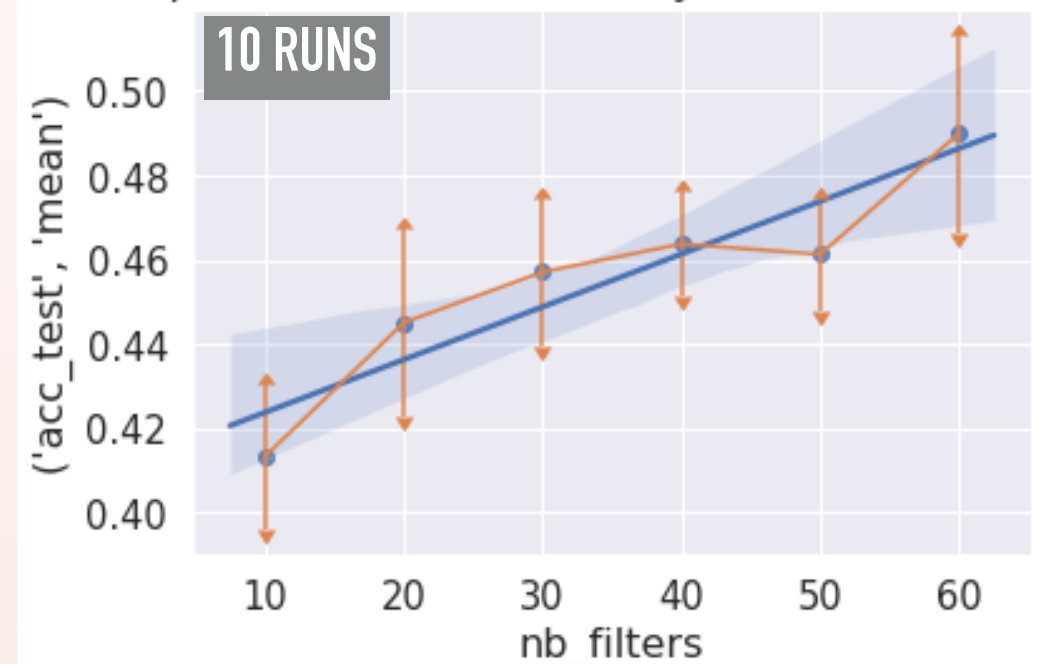


**PLUS DE NOEUD FC => MOINS D'ACCURACY**

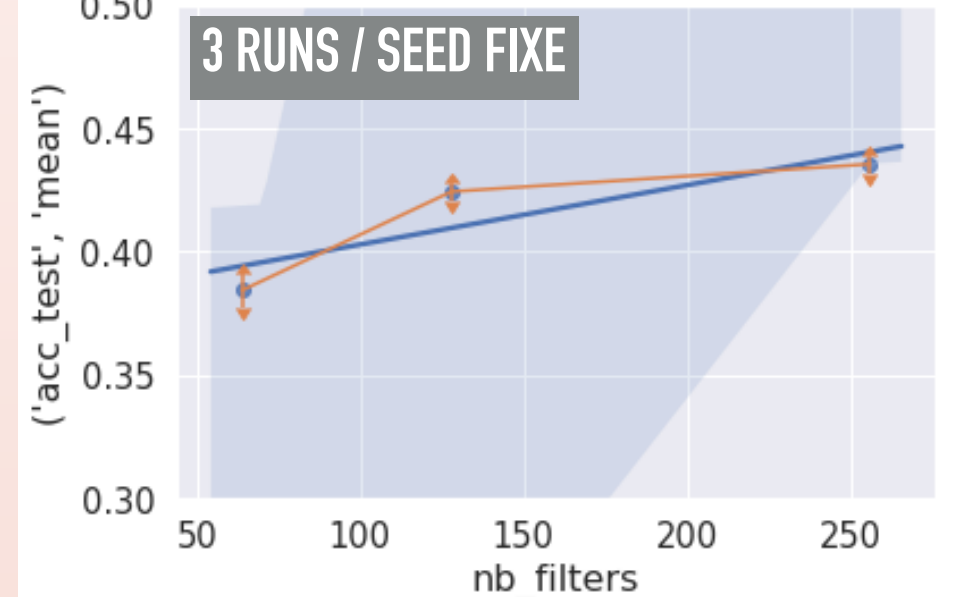
Improve mean test accuracy with nb nodes of before last FC layer



Improve mean test accuracy with nb conv. filters



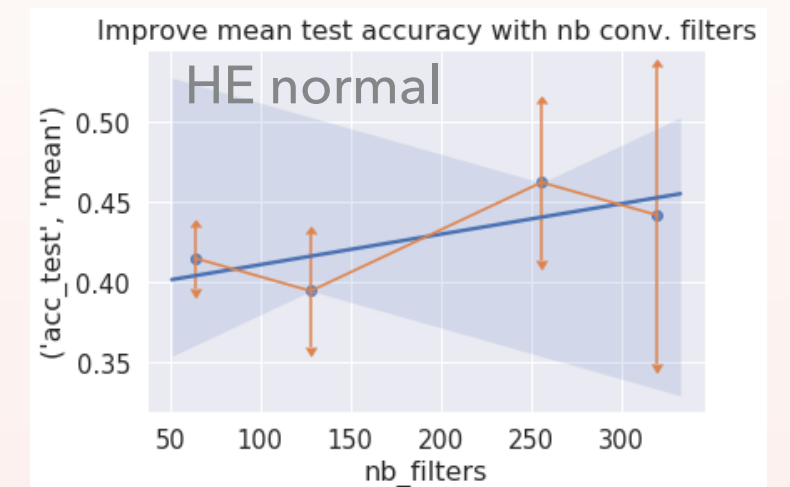
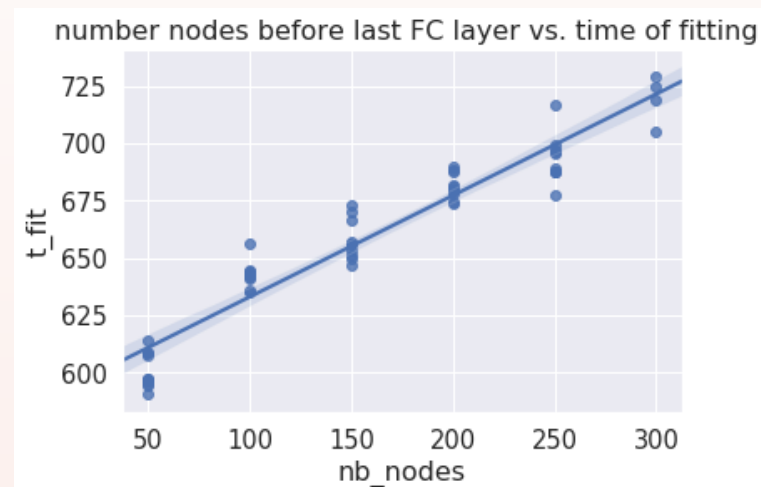
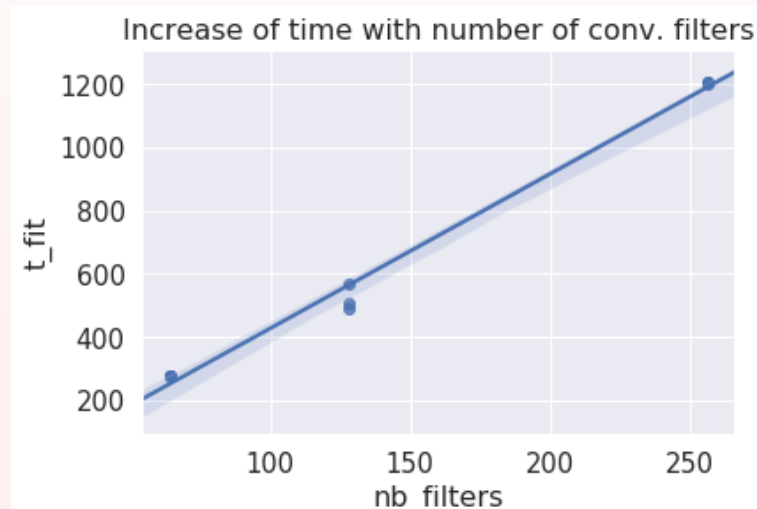
Improve mean test accuracy with nb conv. filters



# MON MODÈLE CNN

Total params: 40,738,251  
 Trainable params: 40,738,251  
 Non-trainable params: 0

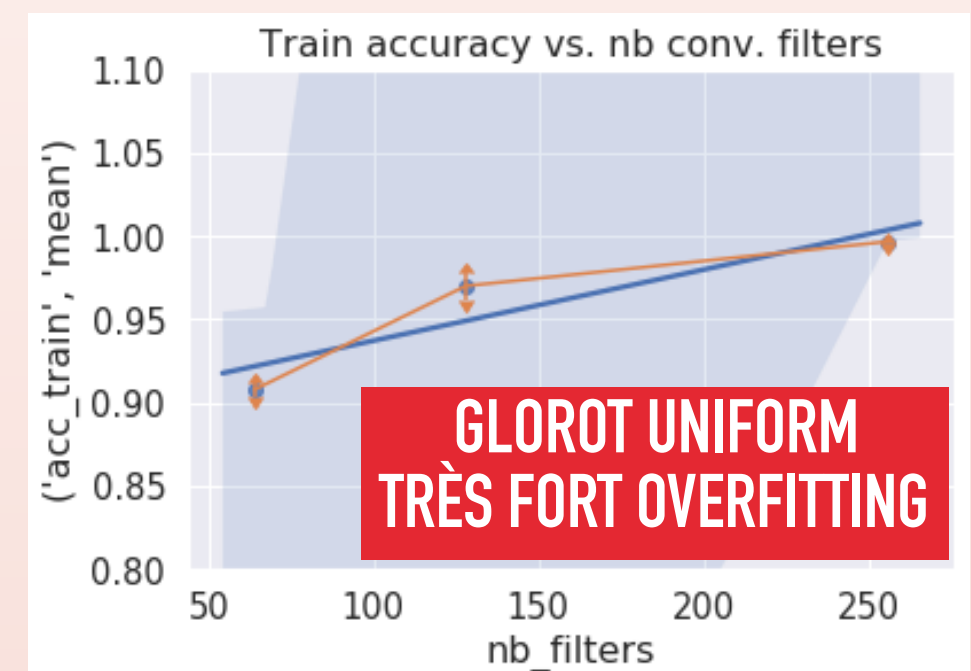
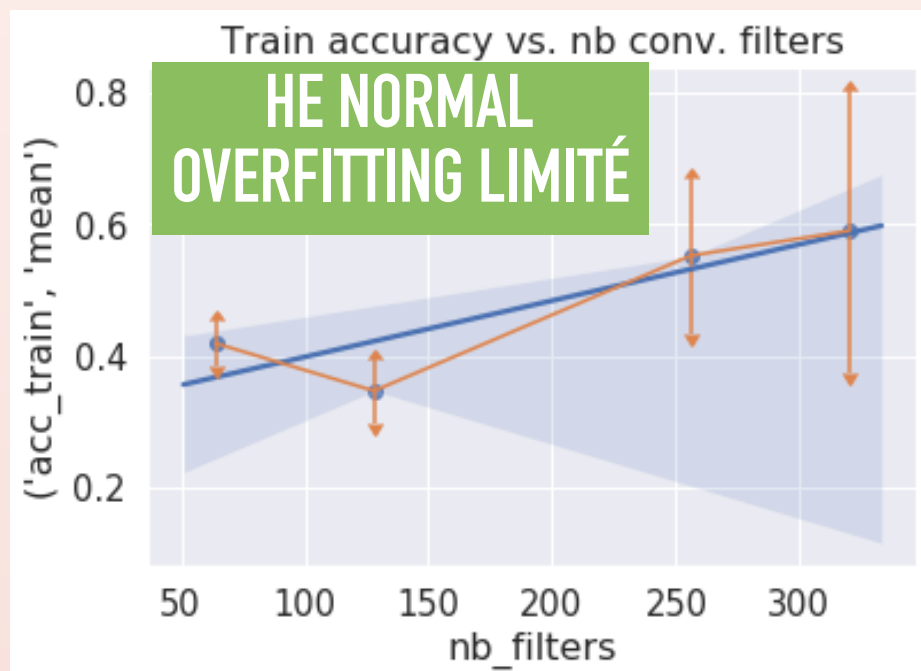
- Choix **256** filtres conv. et **50** noeuds FC pour limiter temps de calcul et dispersion



**+ DE PARAMÈTRES À ENTRAÎNER => + DE TPS CPU & MEMOIRE**

**+ DE FILTRES => + DE DISPERSION**

- Limiter l'overfitting avec une initialisation **HE normal** pour les couches avec f. activ. ReLu

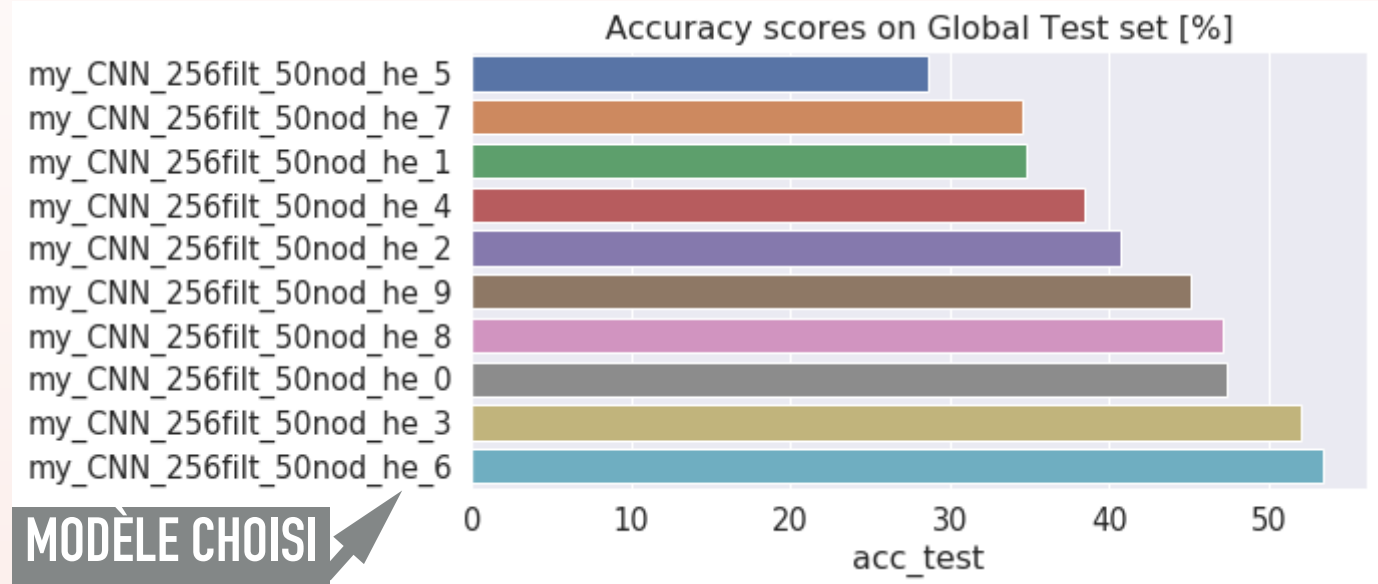


# MON MODÈLE CNN

- ▶ Choix meilleur modèle parmi les 10 runs aléatoires

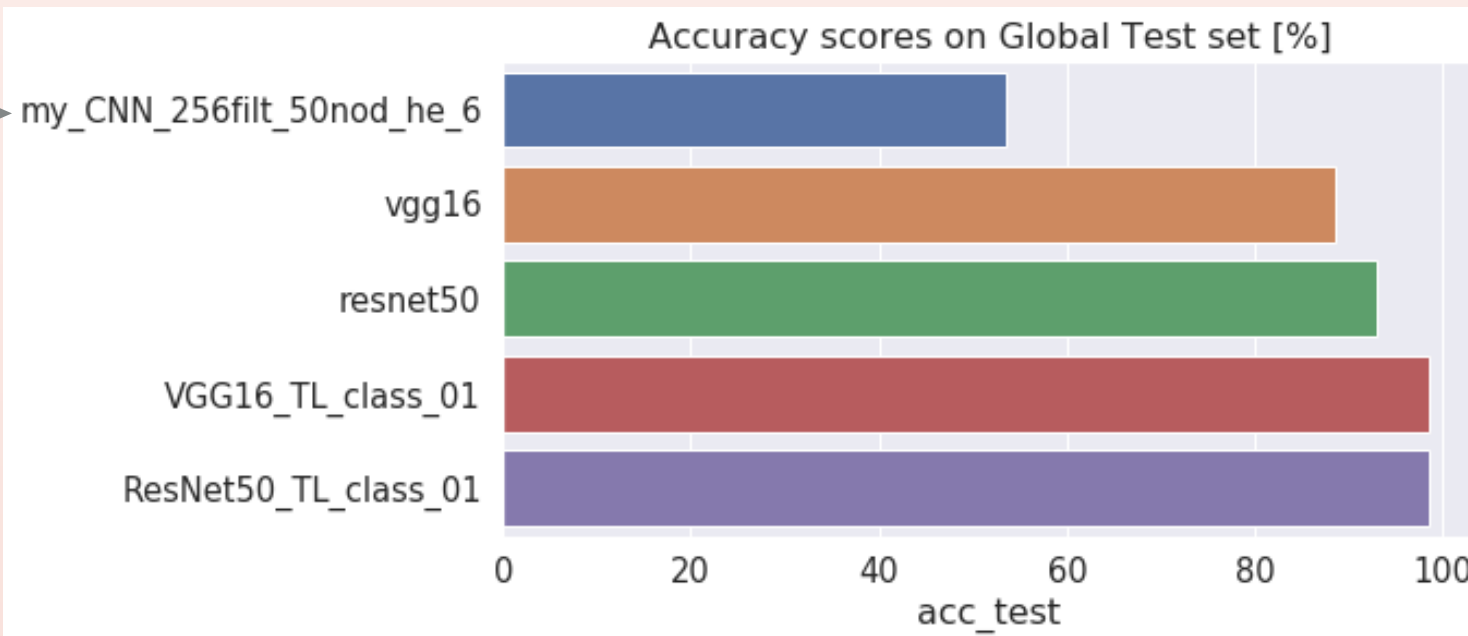
- ▶ **256 filtres** couches de convolution
- ▶ **50 noeuds** couche Fully Connected
- ▶ Optimizer : **SGD** avec learning rate: **3e-8**
  - ▶ decay: 1e-6 & momentum: 0.9
- ▶ Init. f. act. ReLu : **HE normal** random seed
- ▶ Init f. act. Softmax : **GLOROT Uniform**

100 À 150 EPOCHS



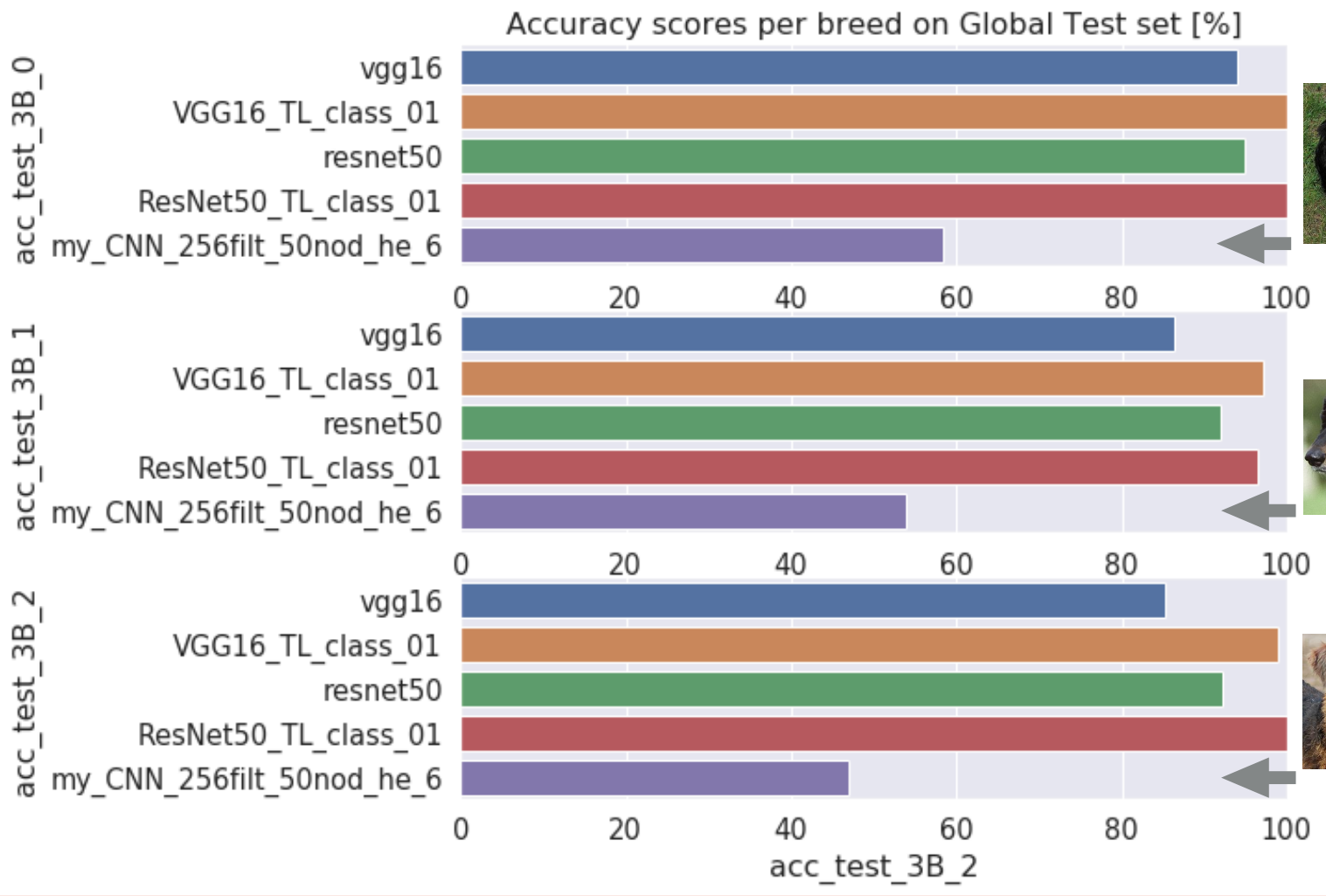
- ▶ Transfert Learning surclasse mon modèle

53%

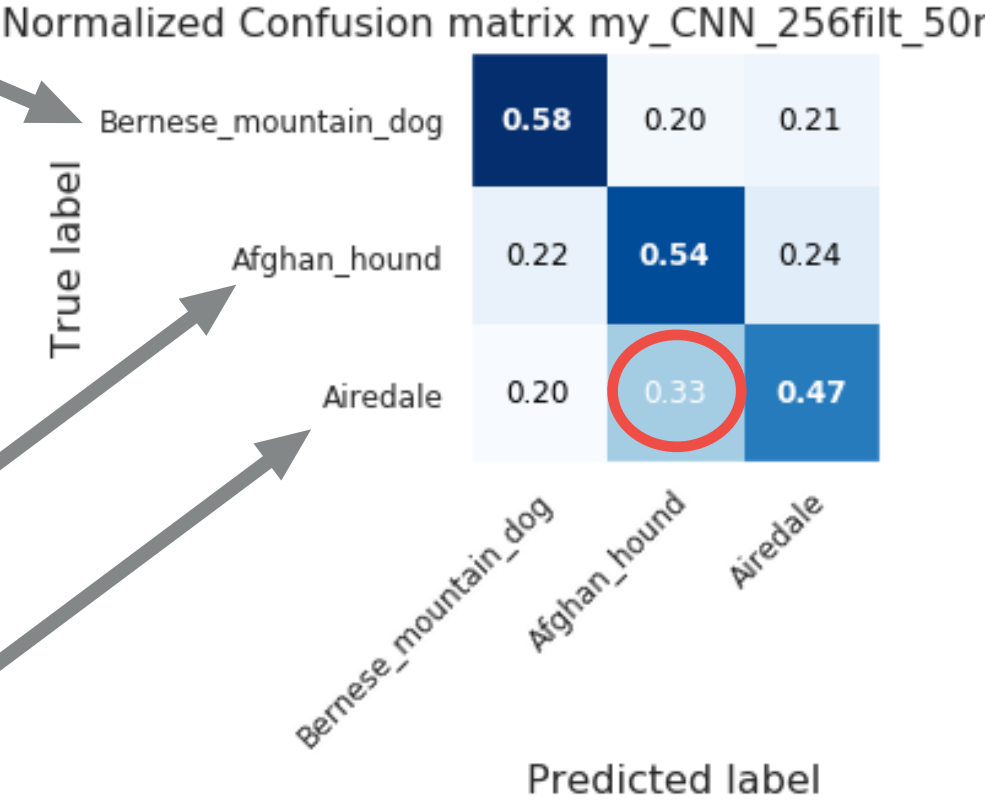


# MON MODÈLE CNN

► Léger déséquilibre par race



## CONFUSION AIREDALE/AFGHAN HOUND





# DATA AUGMENTATION SUR MON MODÈLE CNN

► Data générateur centré et normalisé

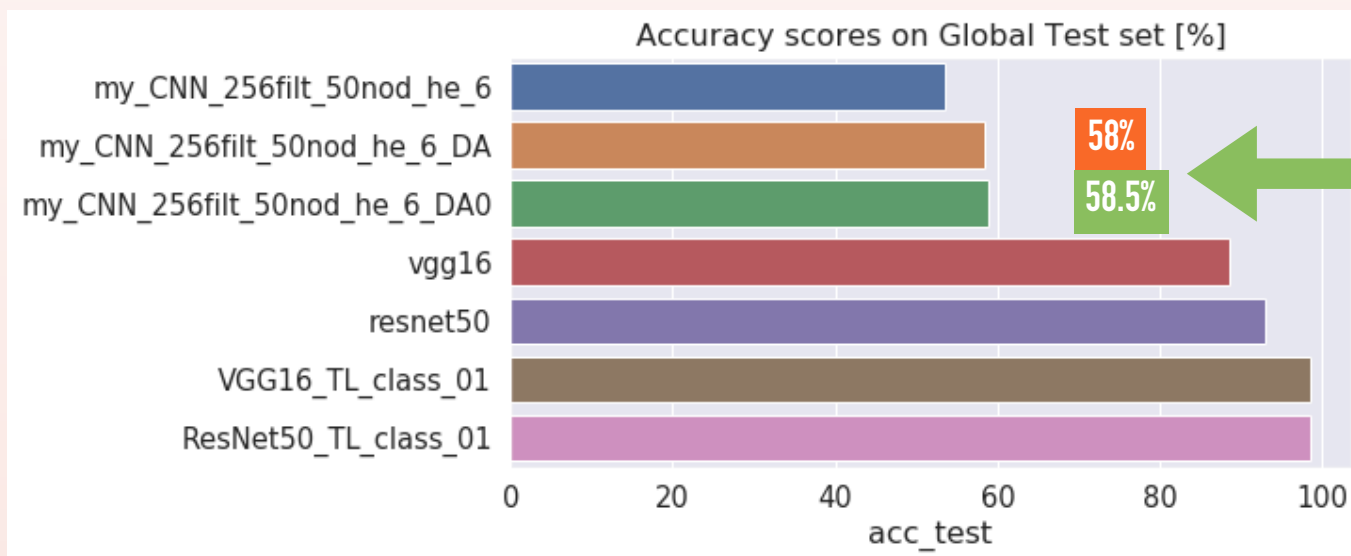
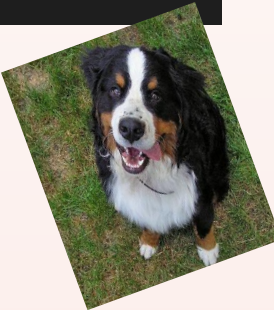
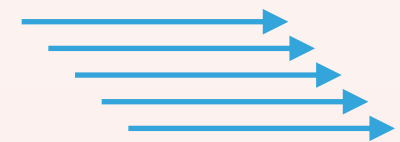
► 2 méthodes

► Ré-entraîné de zéro avec **5000** epochs (DA0)

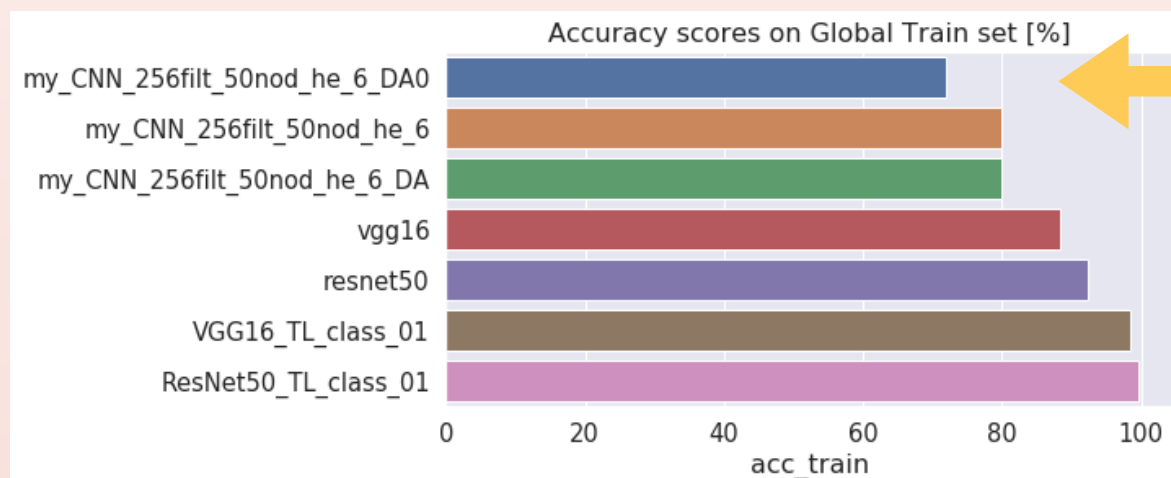
► Ré-entraîné depuis modèle sans D.A. +**3000** epochs : (DA)

300 TRAINING DATA => BATCH SIZE =30

```
# create data generator
datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True)
```



ACCURACY +5% À 5.5% AVEC DA

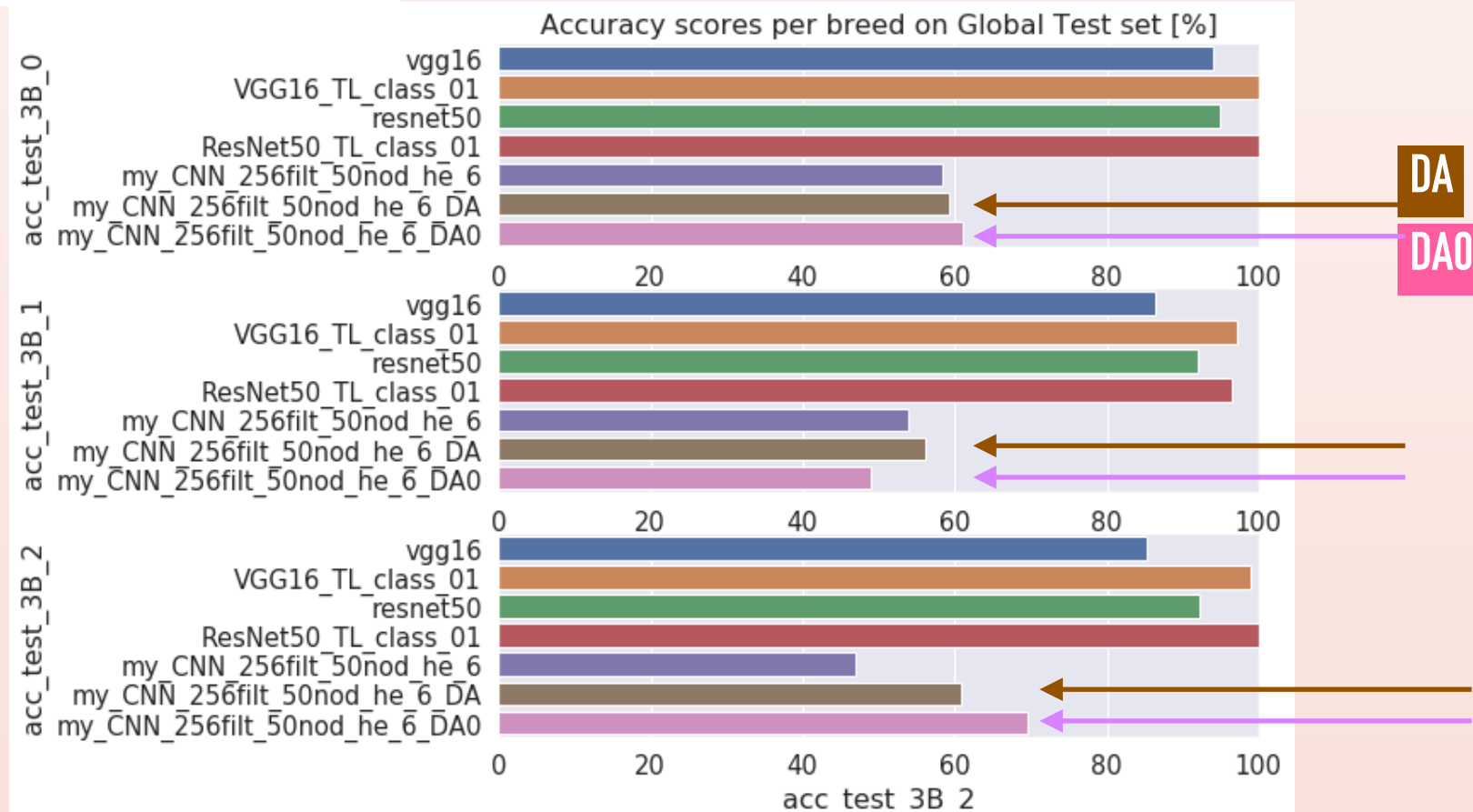
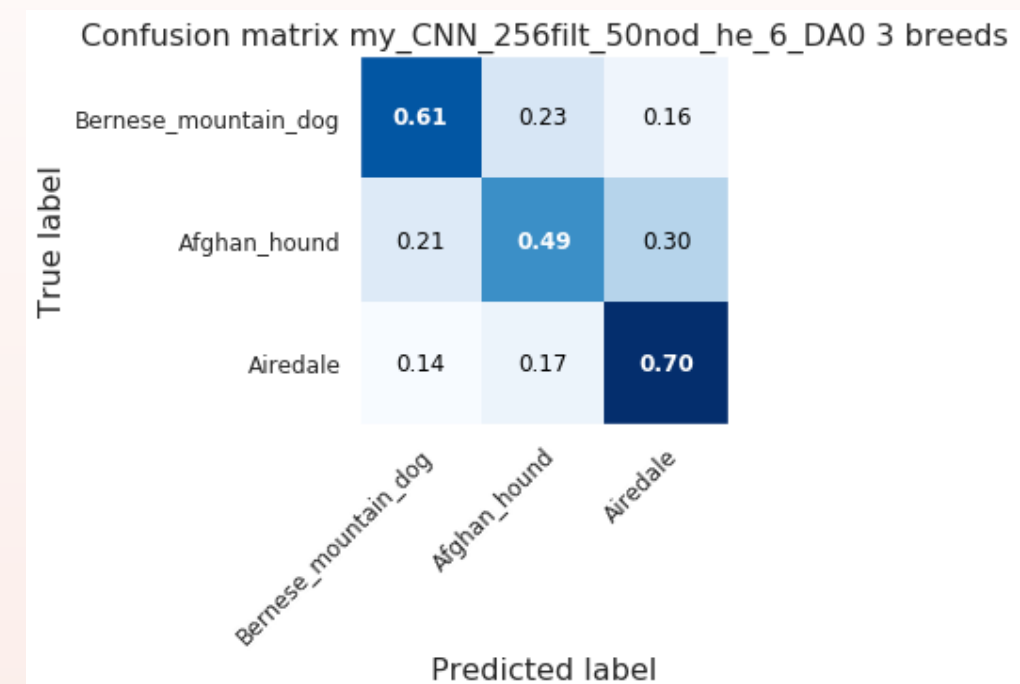
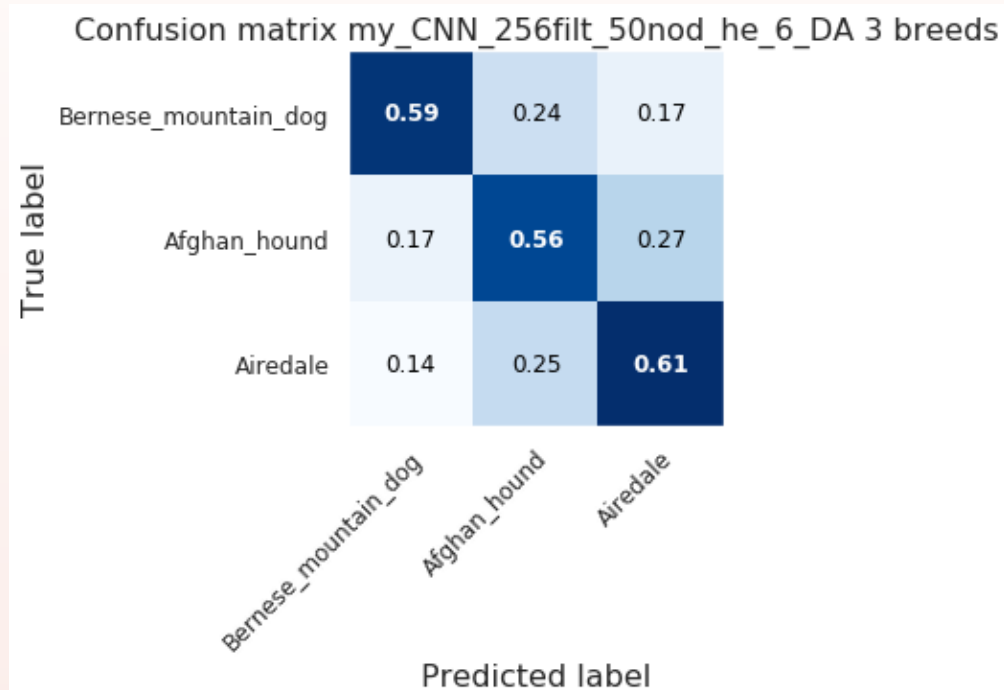


UN PEU MOINS D'OVERFITTING EN PARTANT DE ZÉRO (DA0)

# DATA AUGMENTATION SUR MON MODÈLE CNN

- Accuracy par race rééquilibrée

- Mais Déséquilibrée en partant de zéro (DA0)



## CONCLUSIONS

- ▶ Modèle choisi : Transfert Learning ResNet-50
  - ▶ Meilleur accuracy 10 classes => bonne capacité avec plus de races
- ▶ Pour mon modèle CNN
  - ▶ Limiter le nombre de filtres et noeud / nb. data en training
    - ▶ 256 filters / 50 nodes
  - ▶ Utiliser l'initialisation « He normal » pour moins d'overfitting
  - ▶ Utiliser Data augmentation pour gagner en accuracy
    - ▶ potentiel estimé à 5 à 10%
    - ▶ mais nécessite beaucoup de temps de calcul

## AXES D'AMELIORATION

- ▶ Mon modèle en D.A. améliorable
  - ▶ Relancer l'entraînement sur plus d'itérations
  - ▶ Ajouter des nouvelles photos d'entraînement
  - ▶ Cloud computing plus performant pour **augmenter le nombre de races** prédites
- ▶ Augmenter nombre de race avec ResNet-50 en transfert learning
- ▶ Transfert learning avec d'autres réseaux CNN pré-entraînés
- ▶ Evolution vers la détection des potentiels croisements de chiens