

Indexation automatique d'images

Thomas Weber

Introduction

- Une association de protection des chiens souhaite classer par race l'ensemble de ses pensionnaires.
- Données disponibles: Stanford Dogs Dataset – 20 580 images pour 120 races.
- Objectif: Réaliser un algorithme de détection de la race du chien à partir d'une photo, afin d'accélérer le travail d'indexation.

Deux approches étudiées

Approche classique

- Pré-traitement des images (filtres)
- Extraction de features (SIFT)
- Réduction de dimensions (clustering)
- Visual bag-of-words
- Classification

Approche CNN (réseaux de neurones convolutionnels)

- CNN basique à partir de zéro (from scratch)
 - Intérêt de la data augmentation
- Utilisation du transfer learning:
 - 1/ En bloquant la base convolutionnelle et sans data augmentation
 - 2/ En bloquant la base convolutionnelle et avec data augmentation
 - 3/ En bloquant uniquement les premières couches de la base convolutionnelle et avec data augmentation

Approche classique – Pré-traitement

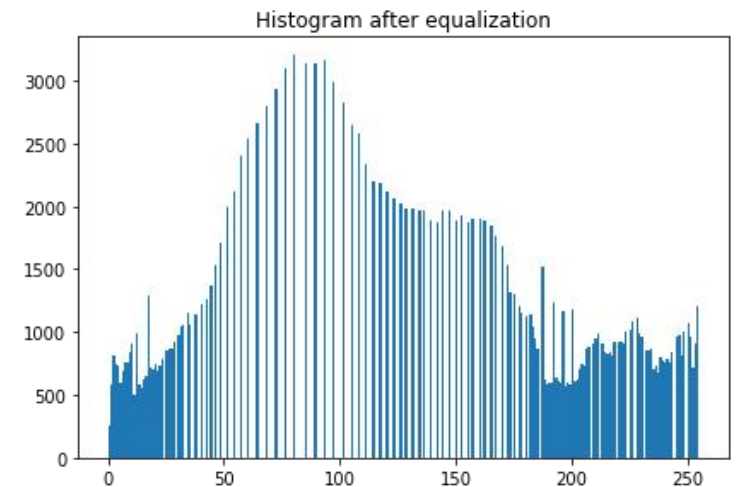
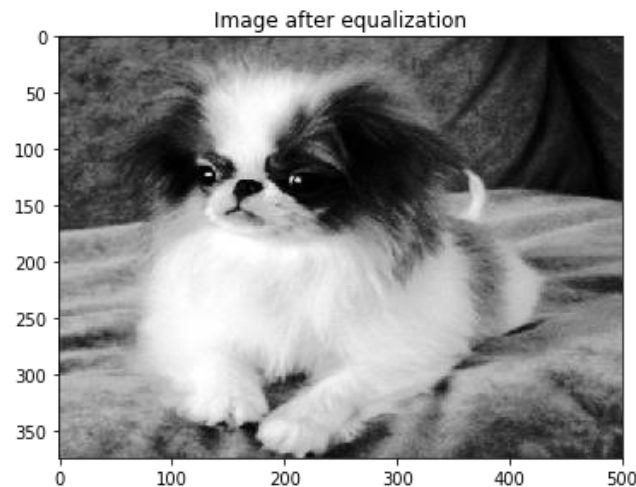
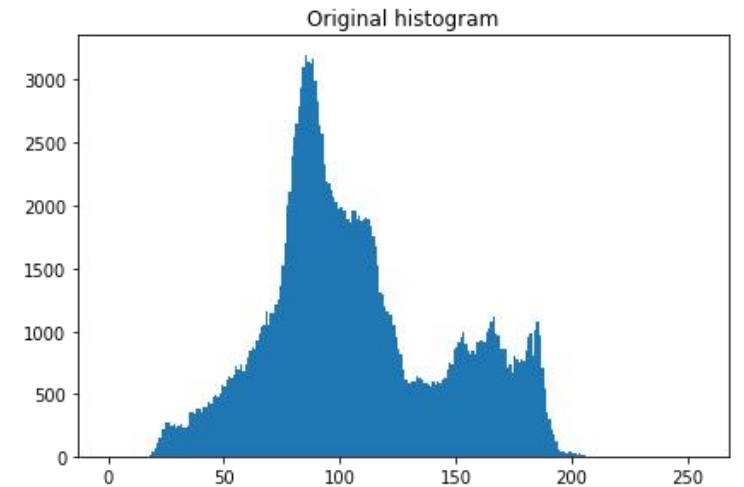
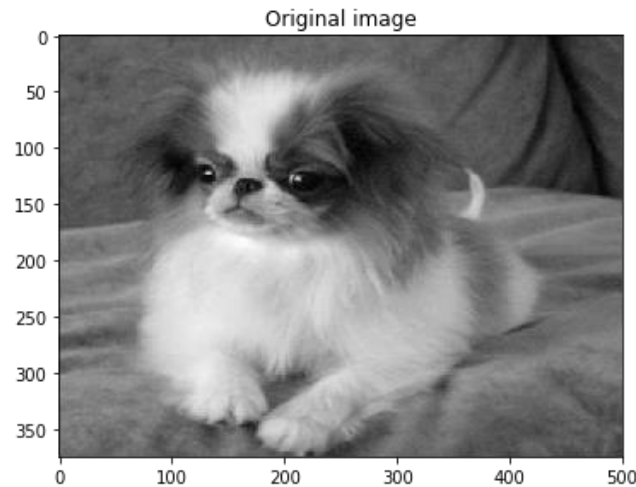
- Redimensionnement: 500x375 px
- Passage en niveaux de gris
- Test de plusieurs filtres:
 - Egalisation d'histogrammes
 - Filtres moyennneur, gaussien, médian, « non-local means »
 - Whitening

Pré-traitement - Equalization

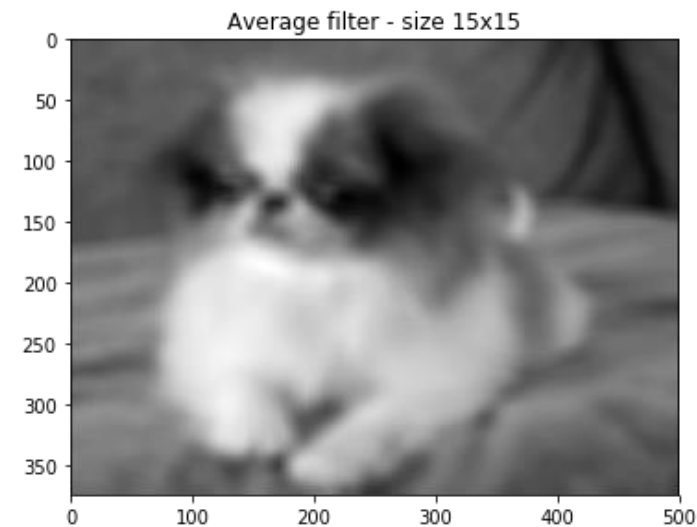
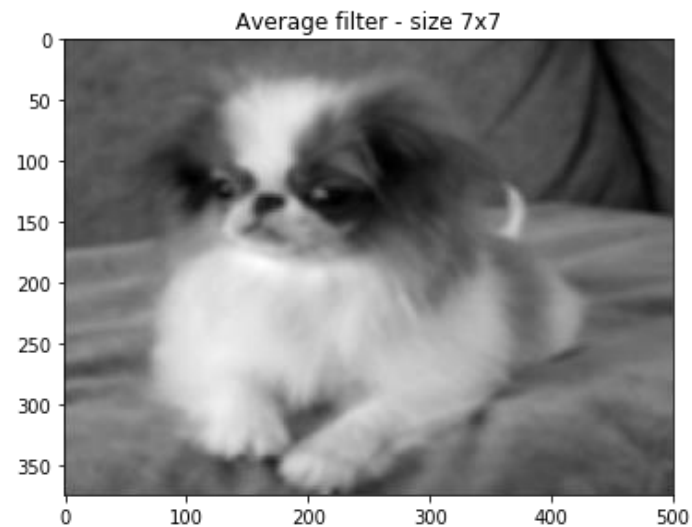
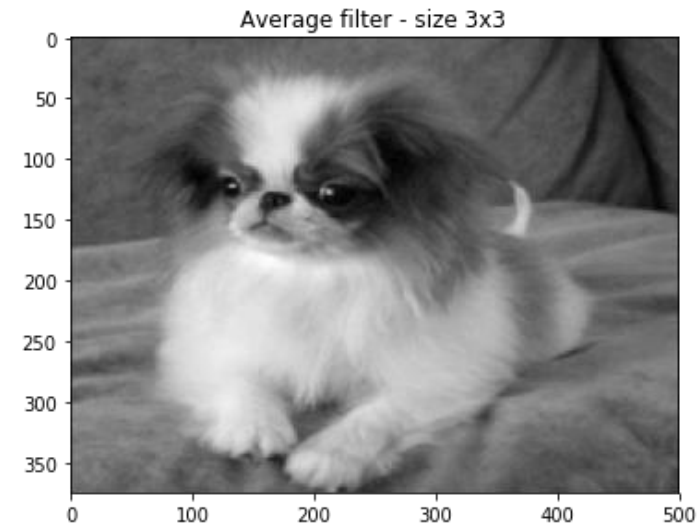
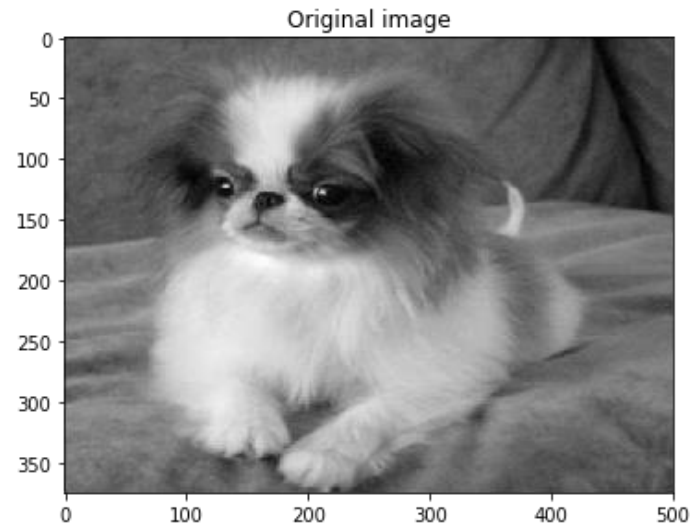
- Egalisation d'histogrammes:

Améliore le contraste

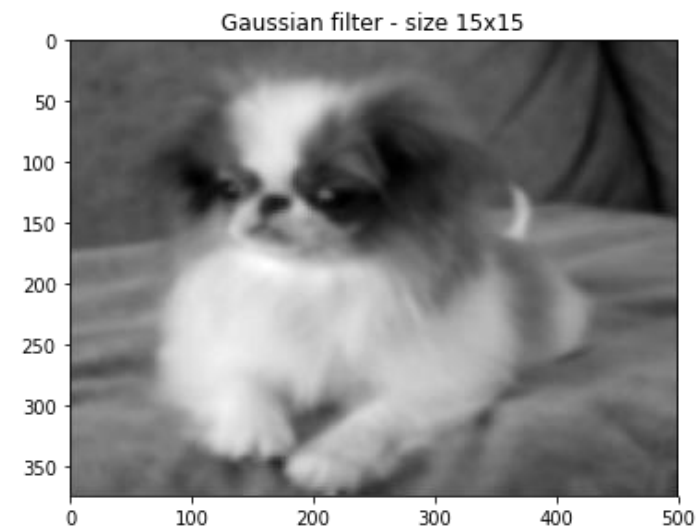
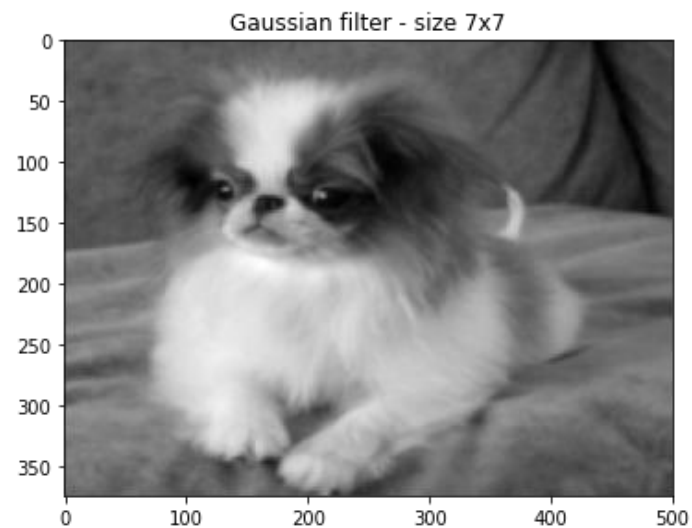
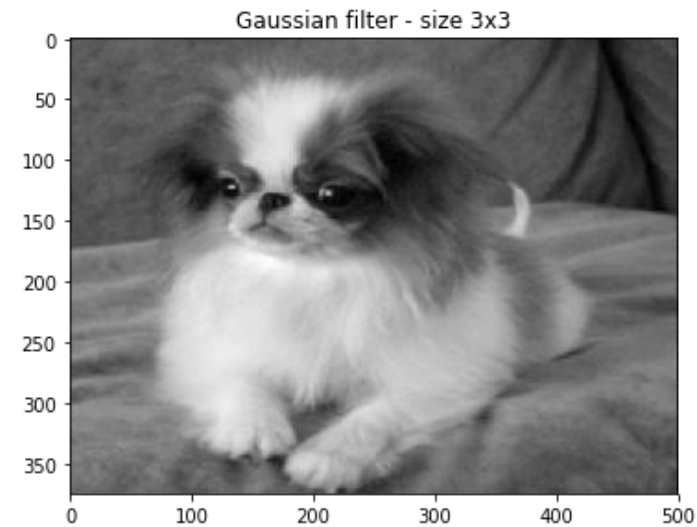
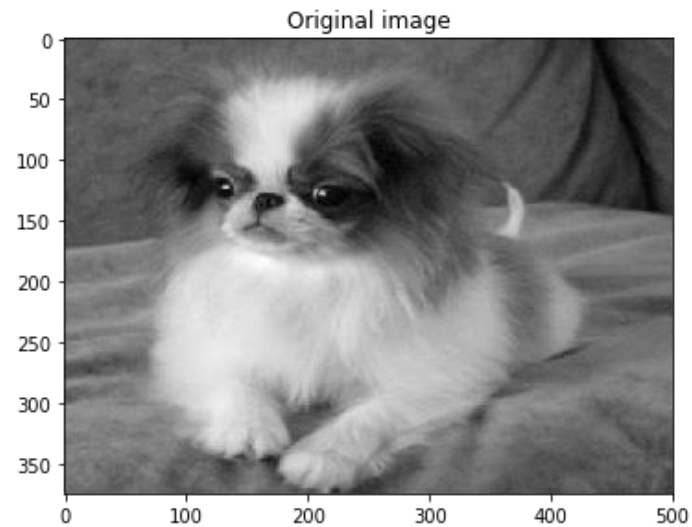
Donne de la clarté aux détails



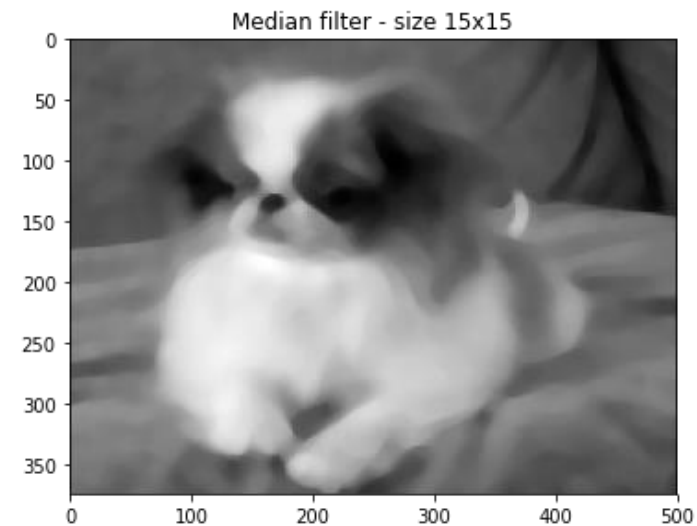
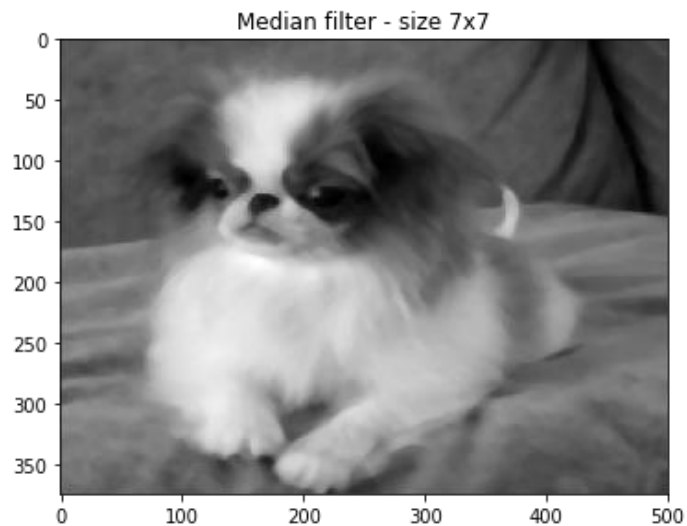
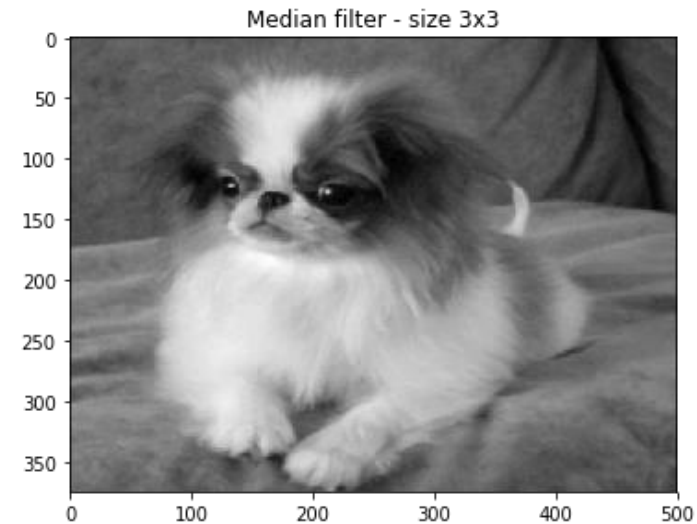
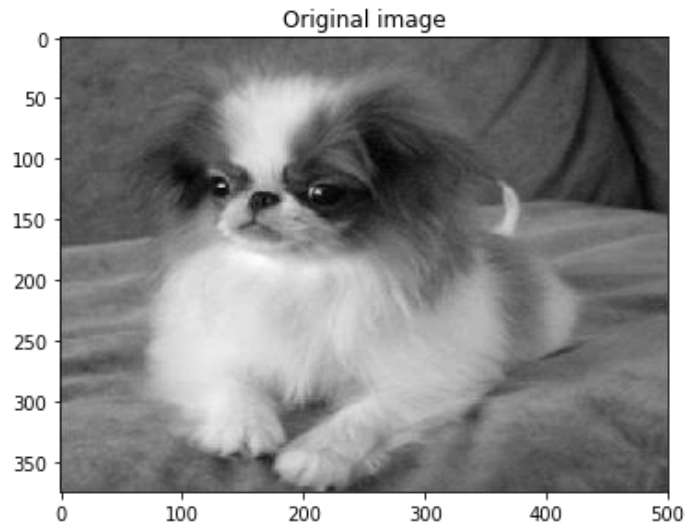
Pré-traitement - Filtre moyenneur



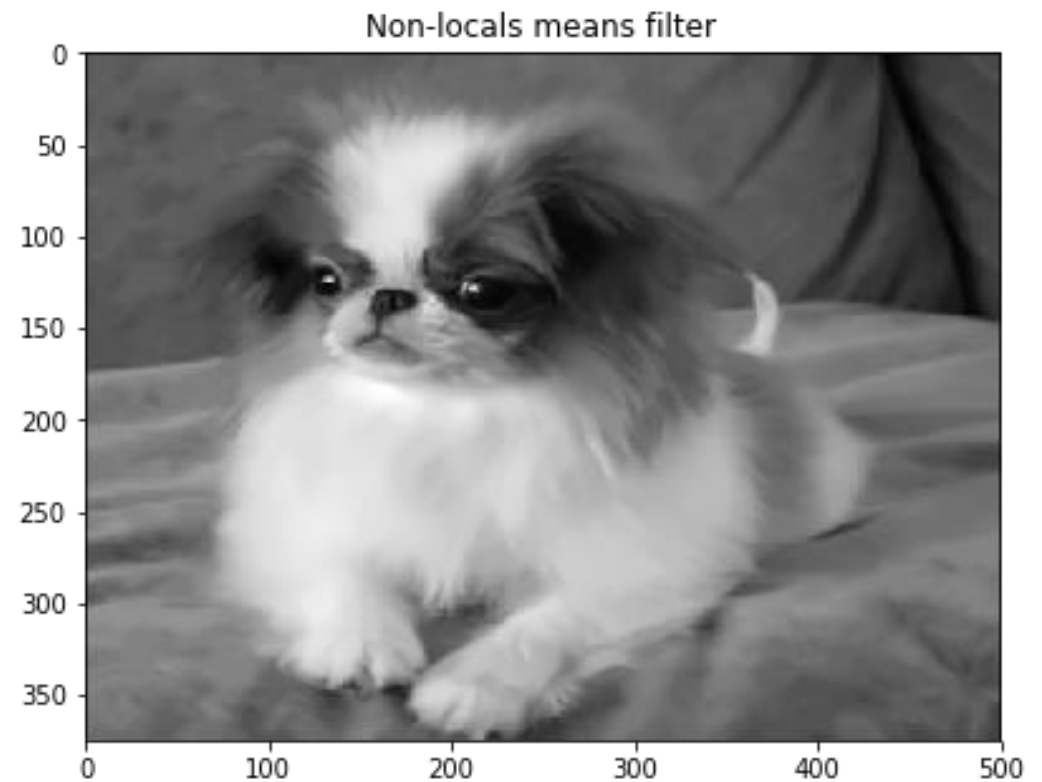
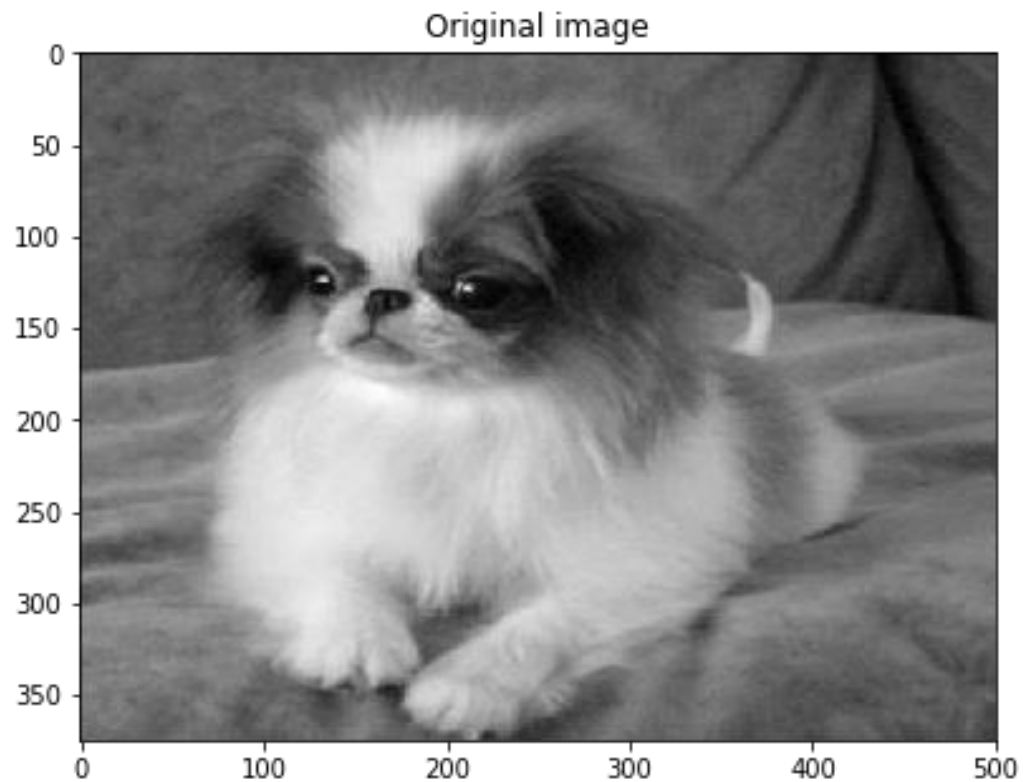
Pré-traitement - Filtre gaussien



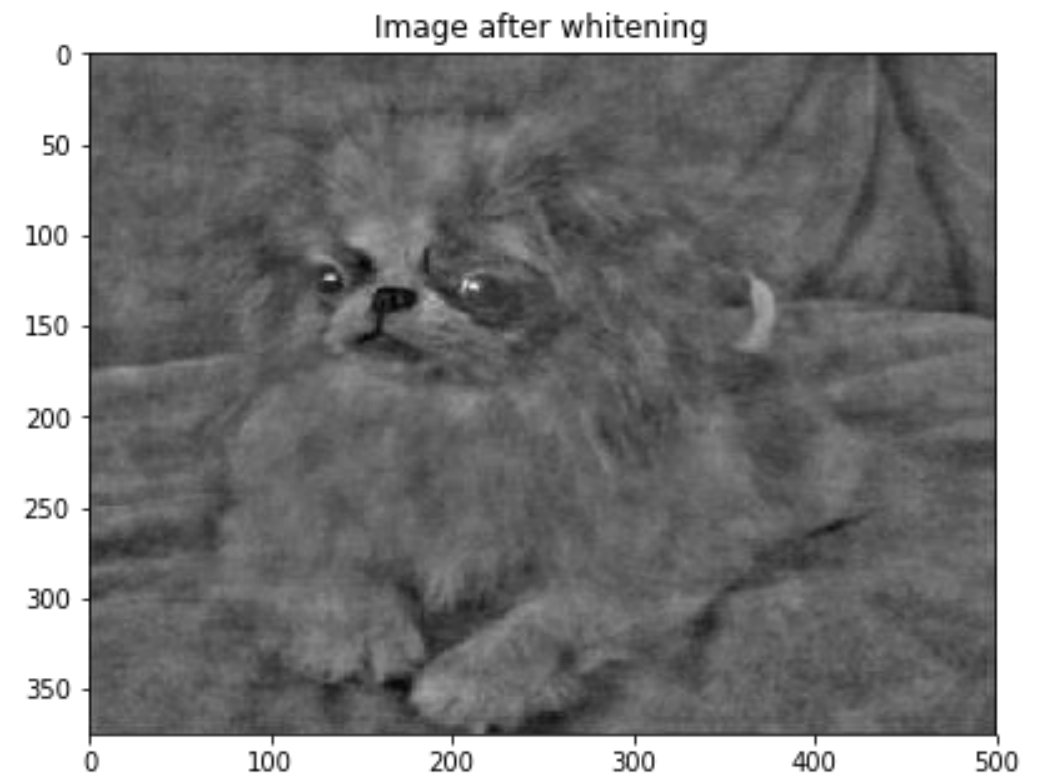
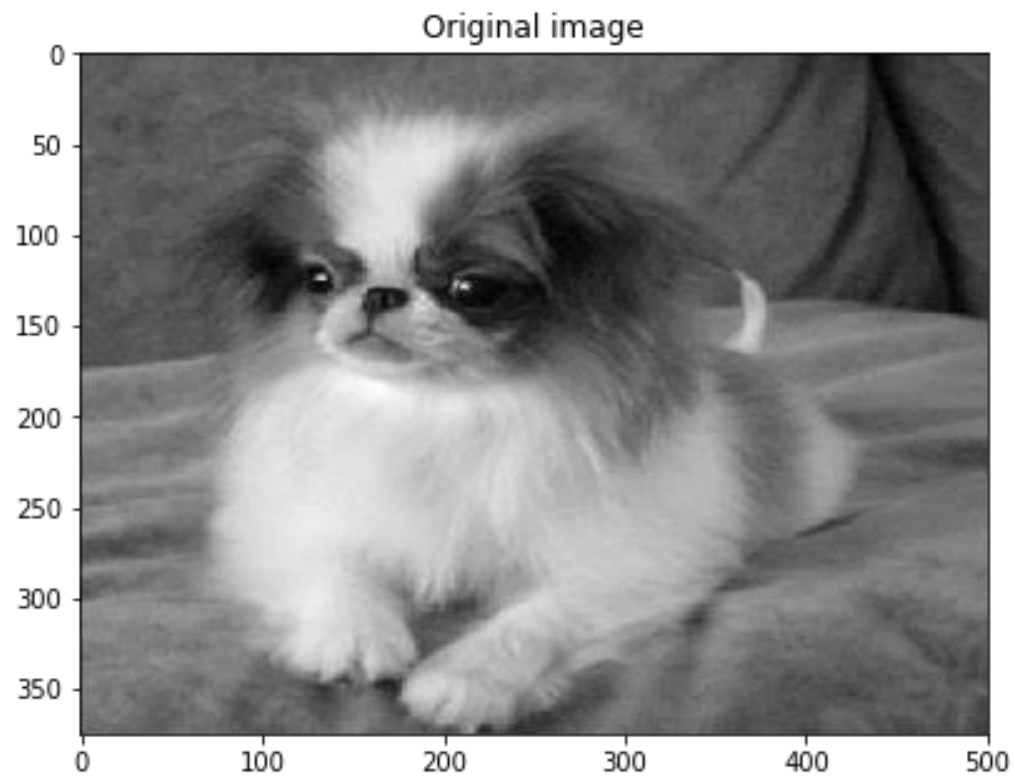
Pré-traitement - Filtre médian



Pré-traitement – Filtre « Non-local means »

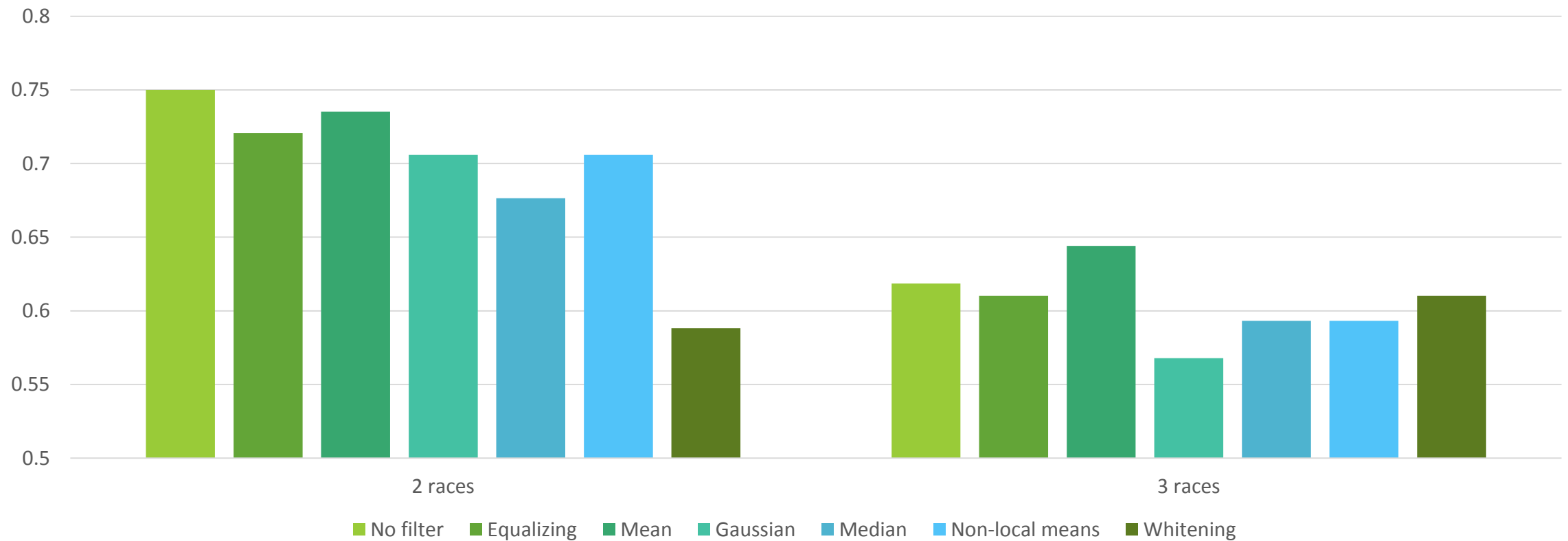


Pré-traitement – Whitening

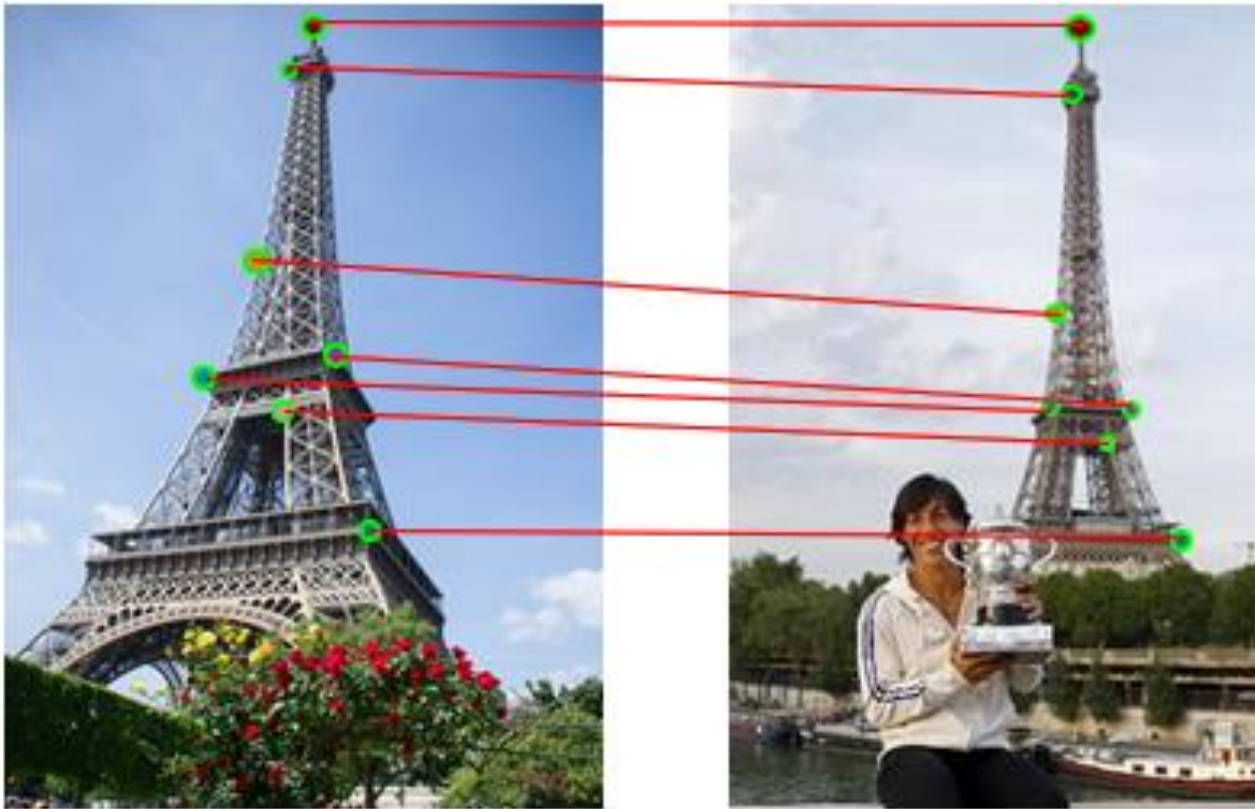


Choix des filtres

Accuracy en fonction des filtres utilisés



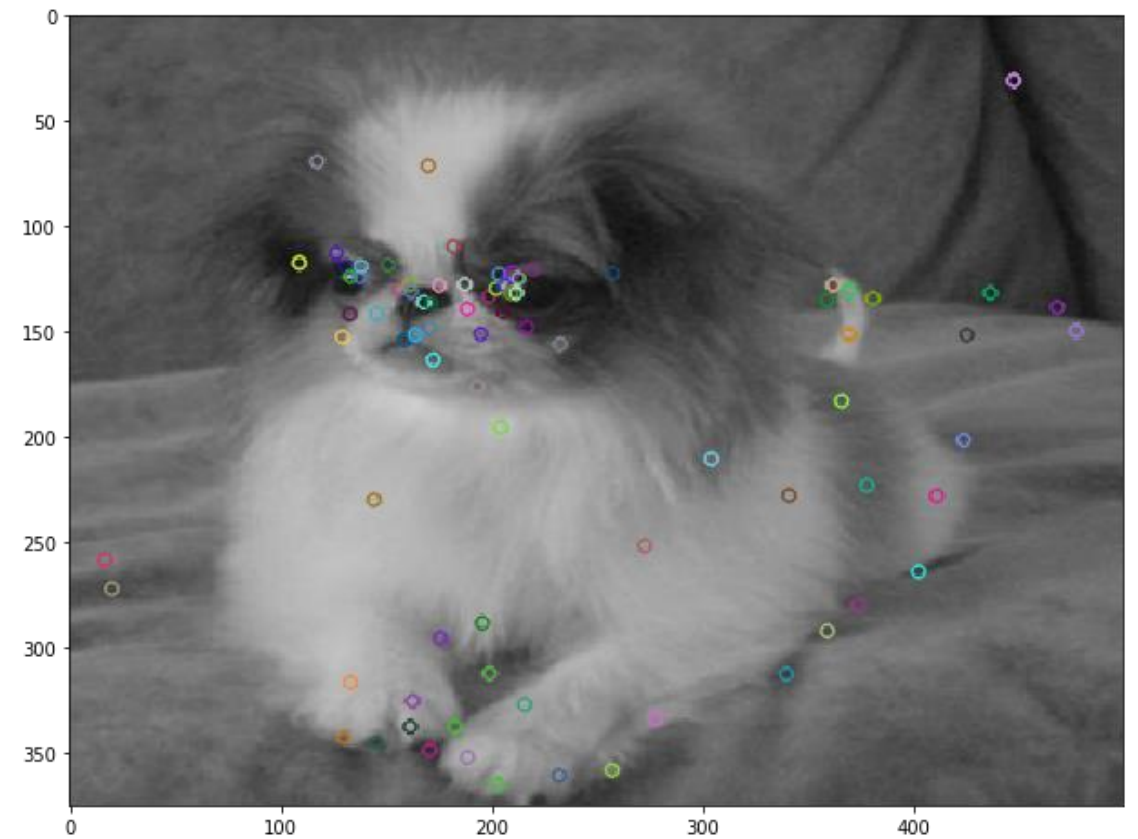
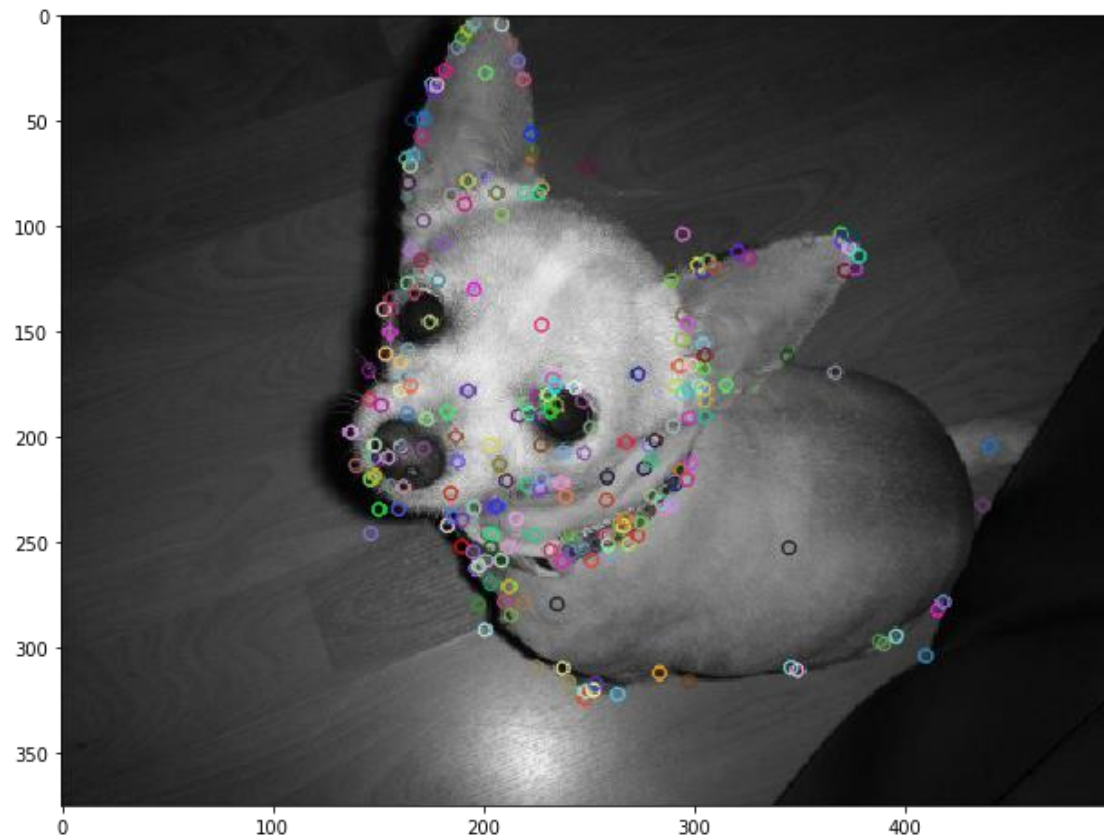
Extraction de features



Caractéristiques d'une « bonne » feature:

- Répétable
- Distinctive
- Locale

Extraction de features - SIFT

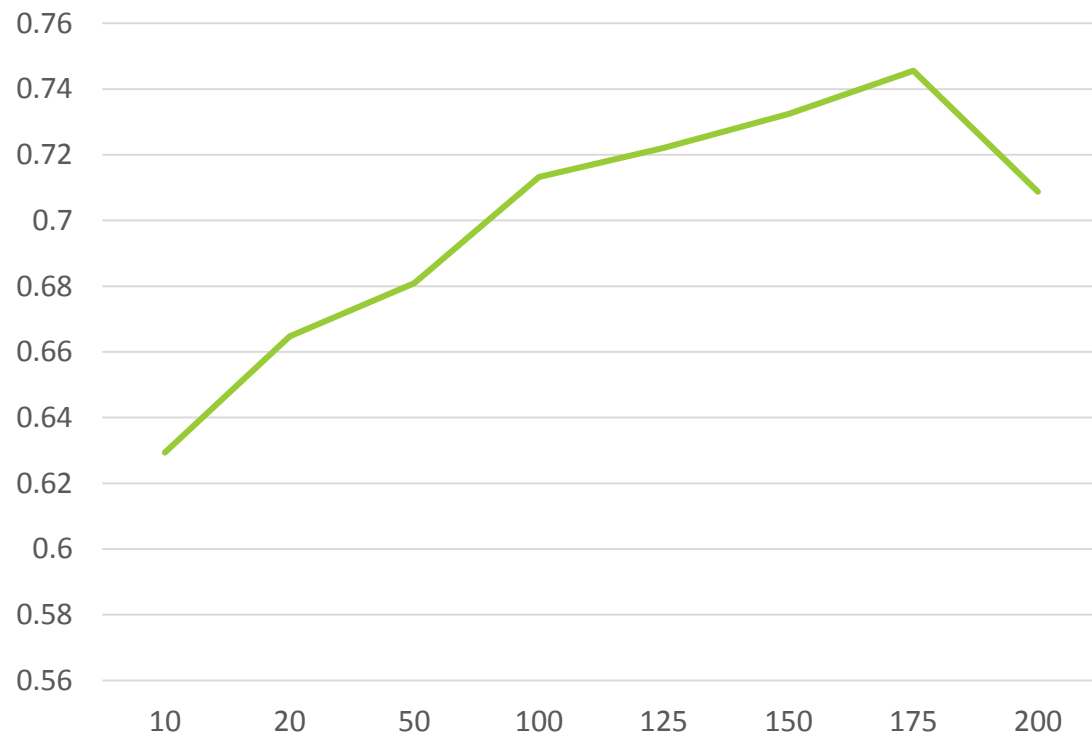


Clustering - KMeans

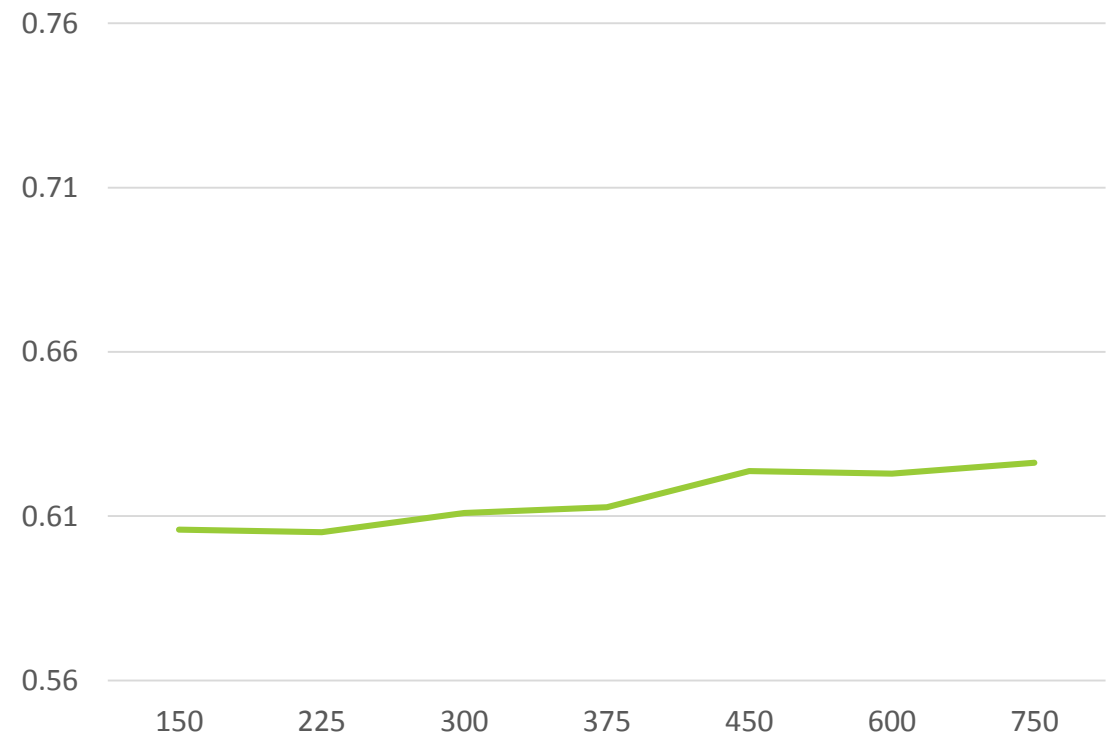
- Nombre de descripteurs SIFT élevé:
 - 2 races: 272 515 descripteurs
 - 3 races: 400 402 descripteurs
 - 5 races: 659 758 descripteurs
- Utilisation de KMeans pour réduire la dimension
- Nombre de clusters testés:
 - Pour 2 races: [10, 20, 50, 100, 125, 150, 175, 200]
 - Pour 3 races: [150, 225, 300, 375, 450, 600, 750]
- Pour chaque image, on détermine ensuite le nombre de descripteurs dans chaque cluster: bag-of-words visuels

Nombre de clusters optimal

Accuracy en fonction du nombre de clusters
2 races

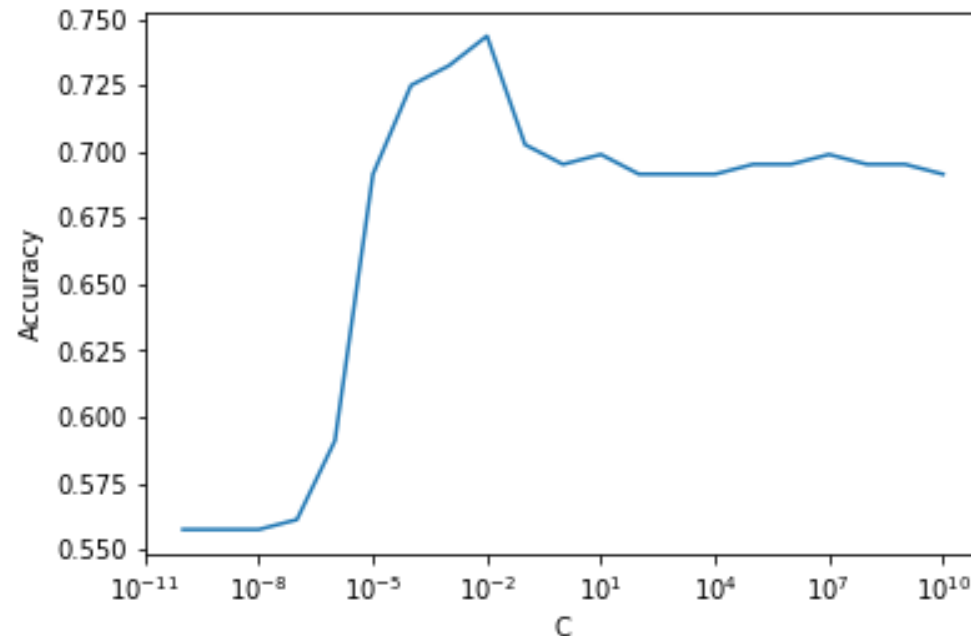


Accuracy en fonction du nombre de clusters
3 races



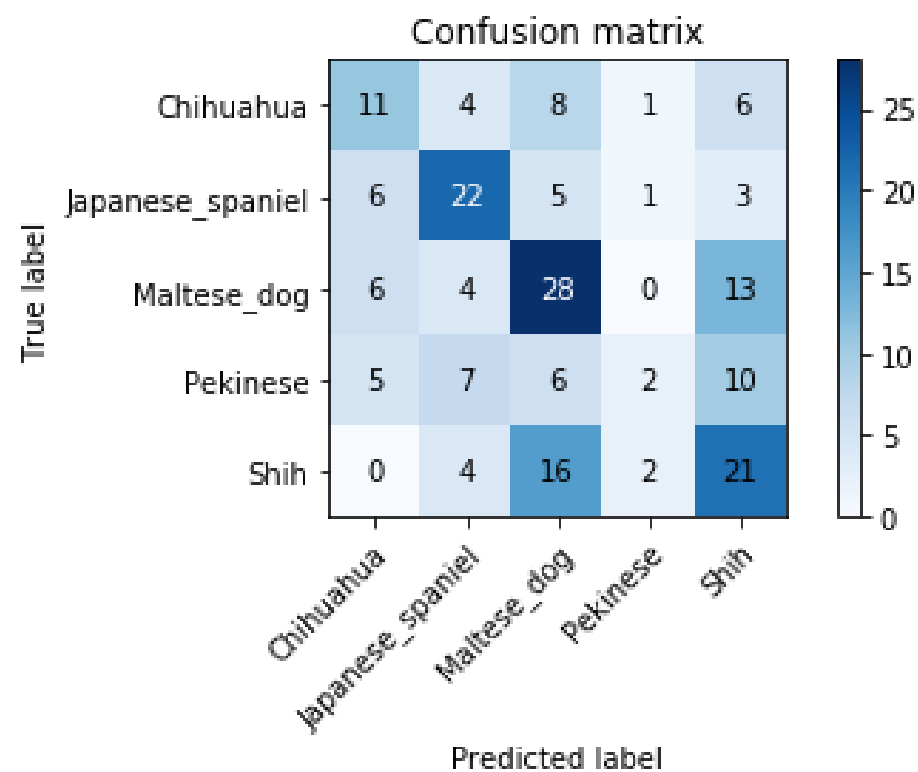
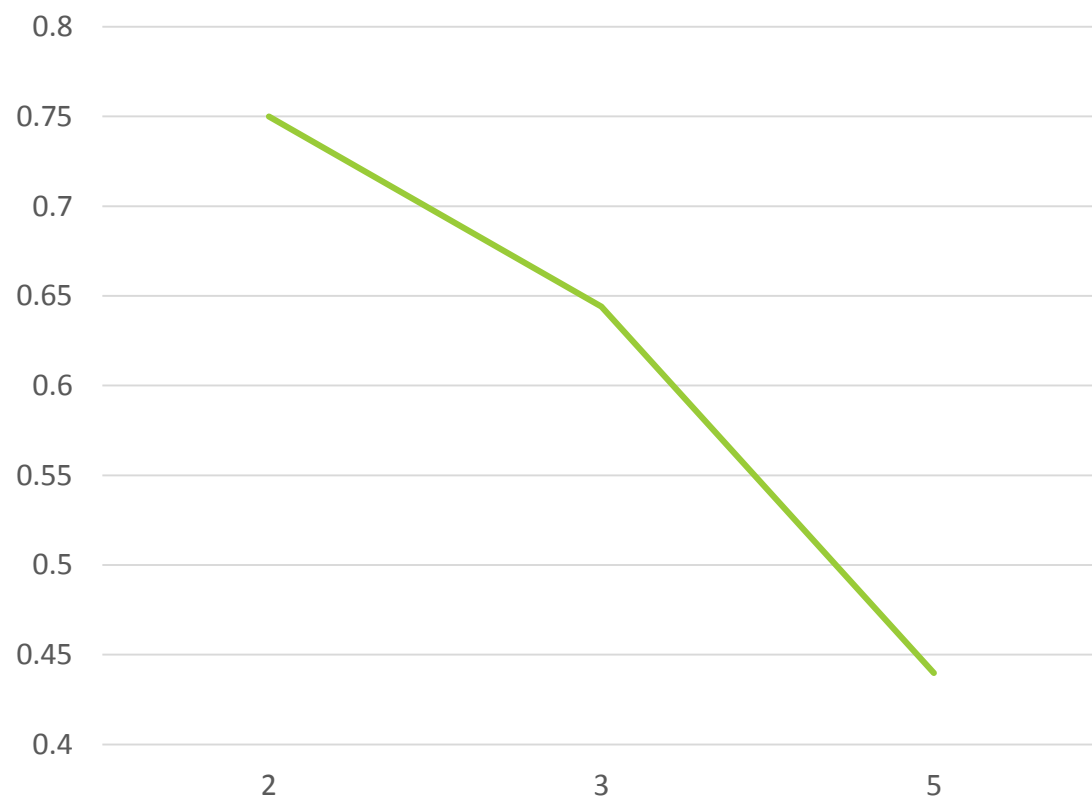
Classification – Régression logistique

- Grid Search pour trouver la meilleure valeur de C
- Par exemple, pour 2 races:



Résultats

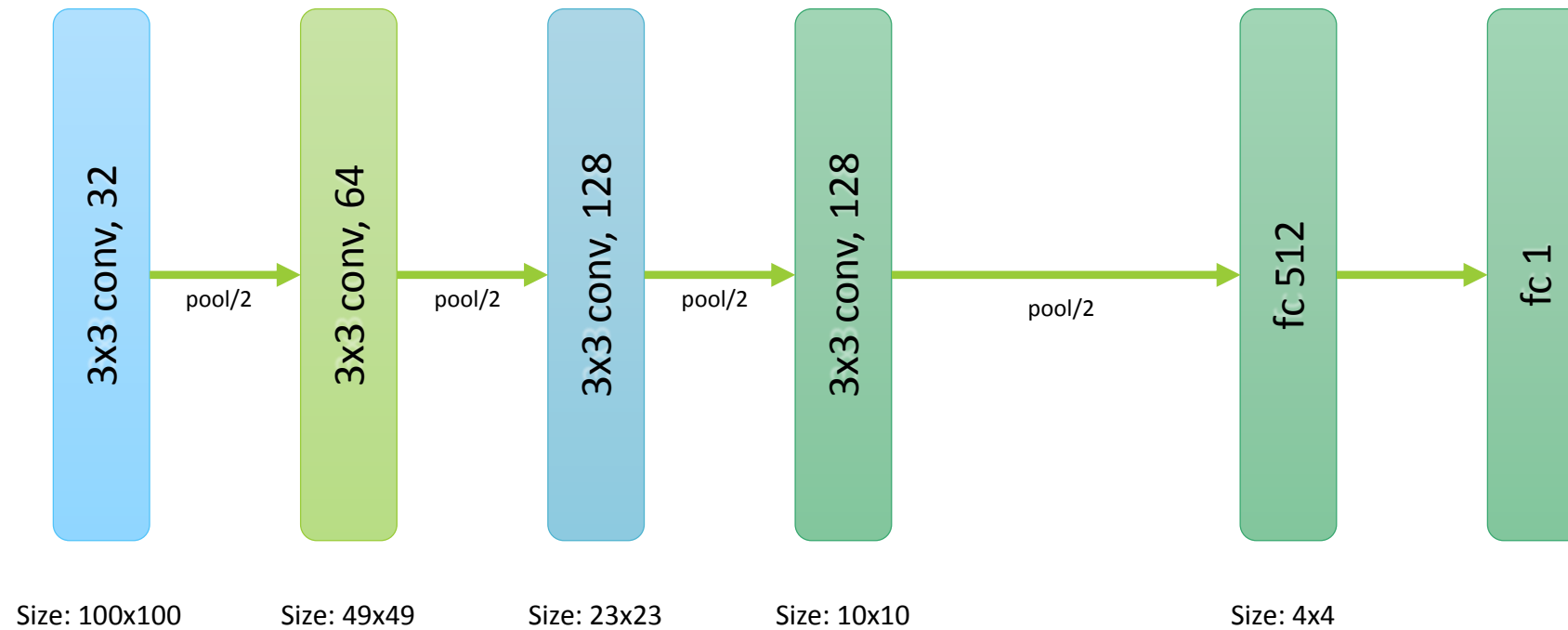
Accuracy en fonction du nombre de races



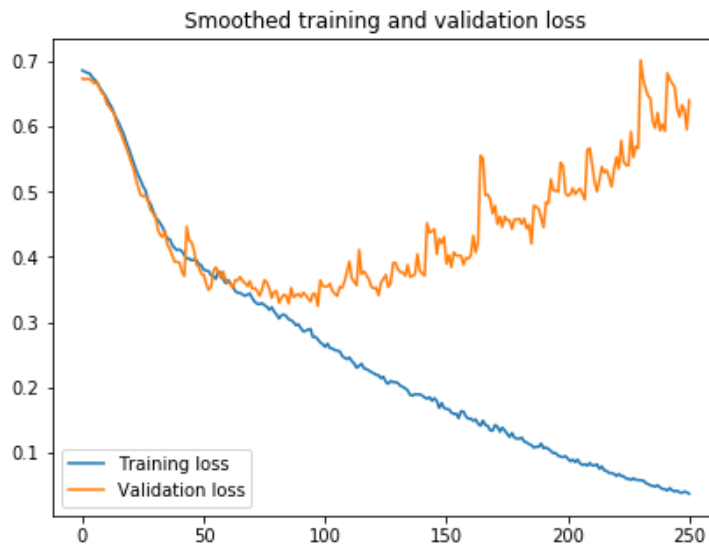
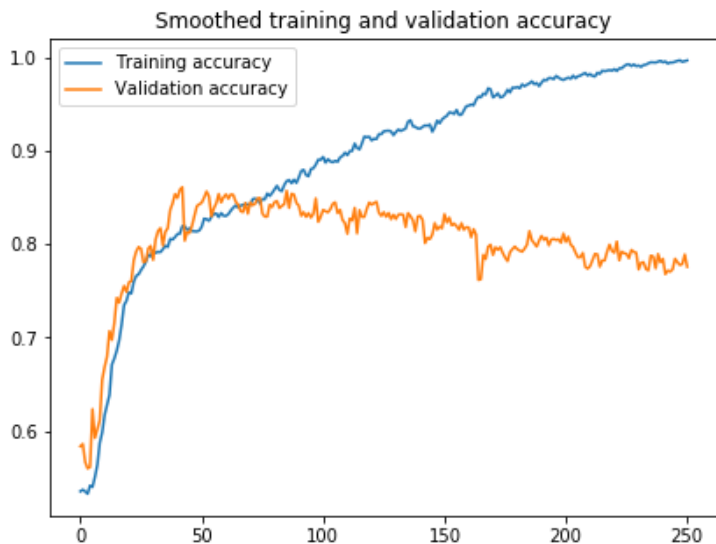
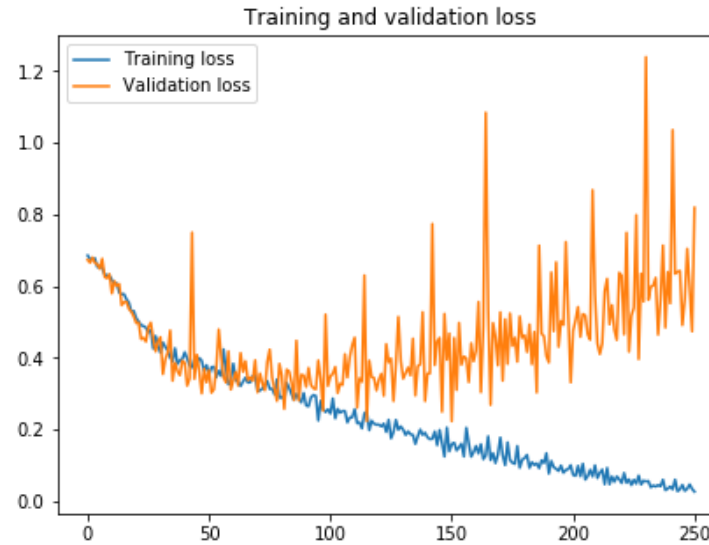
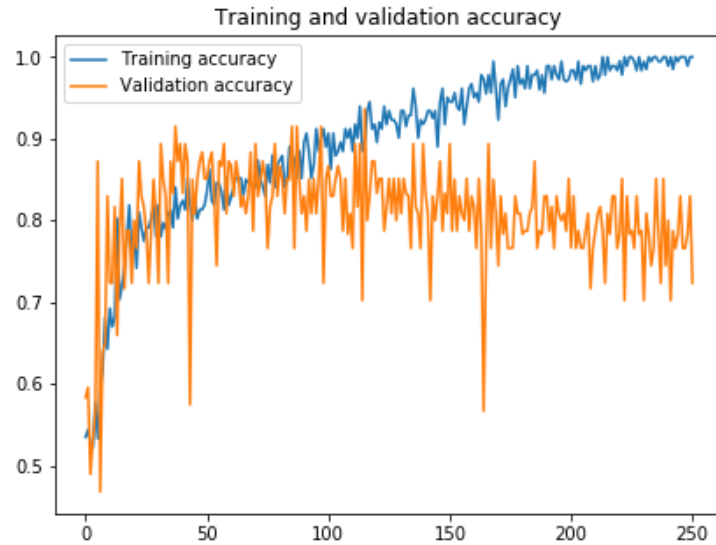
Approche CNN

- 2 stratégies:
 - CNN from scratch
 - Transfer learning
- Redimensionnement des images en 100x100
- Utilisation des images en couleur
- Séparation train/validation/test: 60 %/20 %/20 %

CNN from scratch



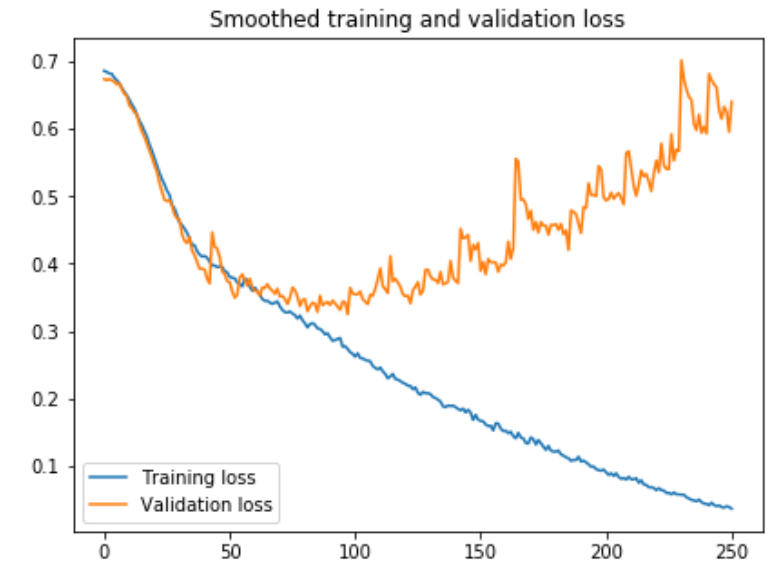
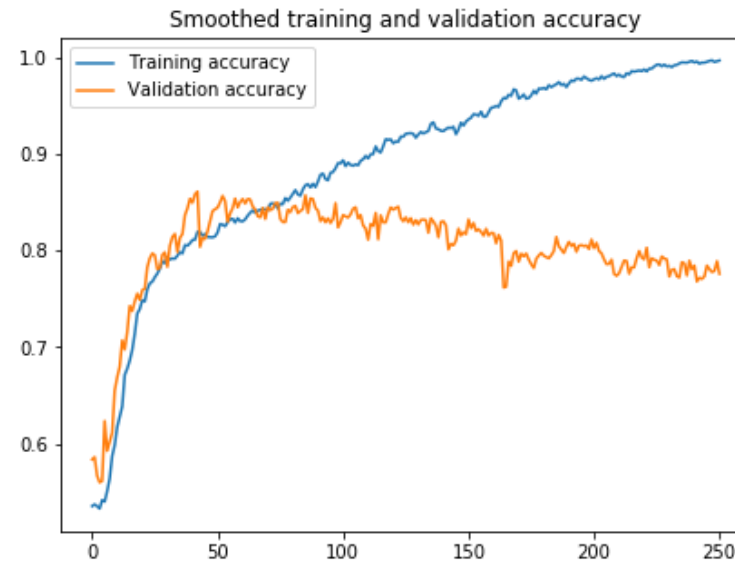
CNN from scratch – 2 races



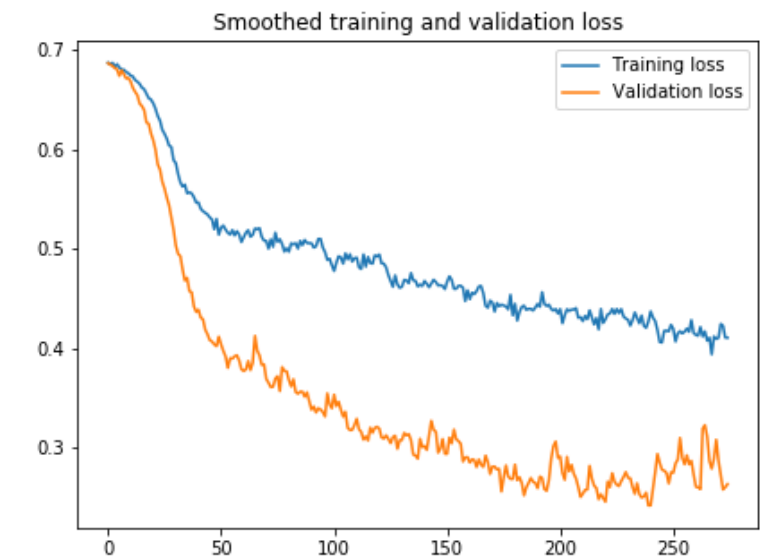
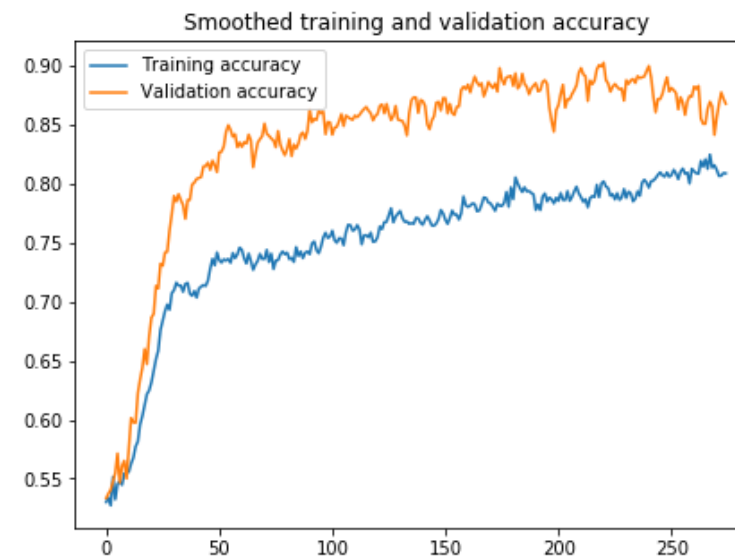
Accuracy sur jeu de test:
82.35%

Overfitting

Sans data-augmentation

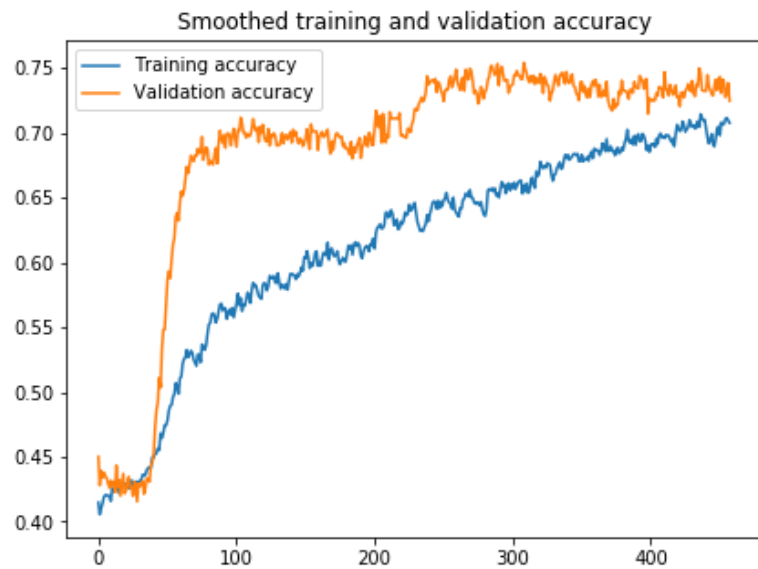


Avec data-augmentation

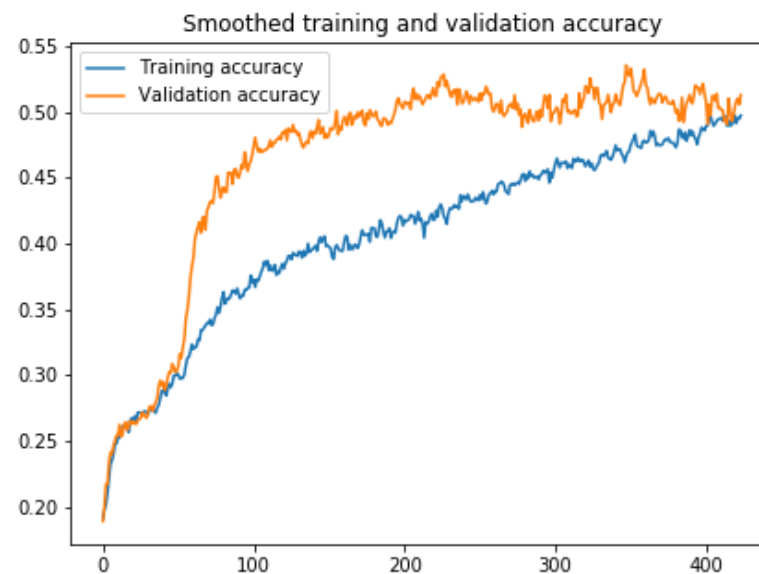


Résultats

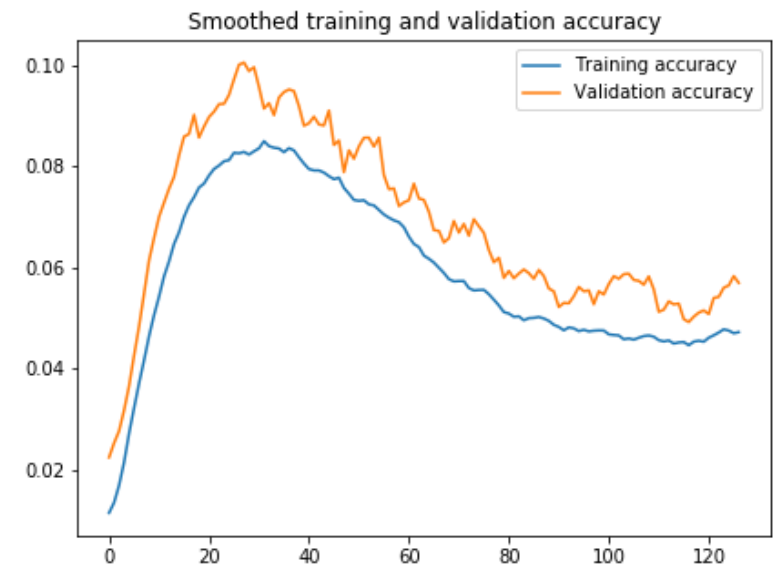
3 races: 70.59 %



5 races: 52.60 %

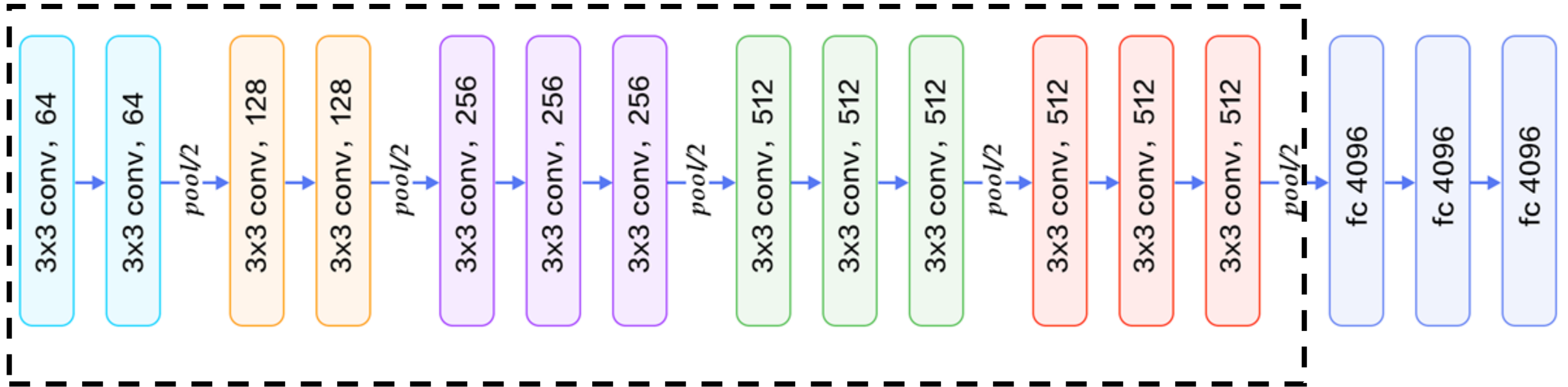


120 races: 12.95 %



Transfer Learning

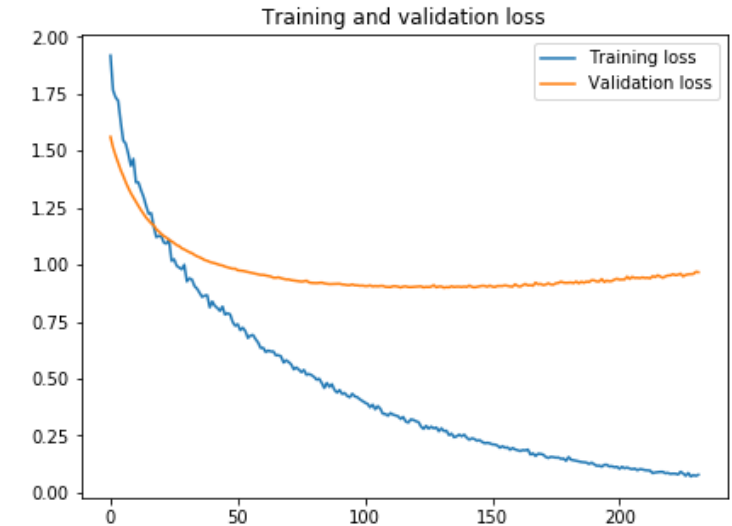
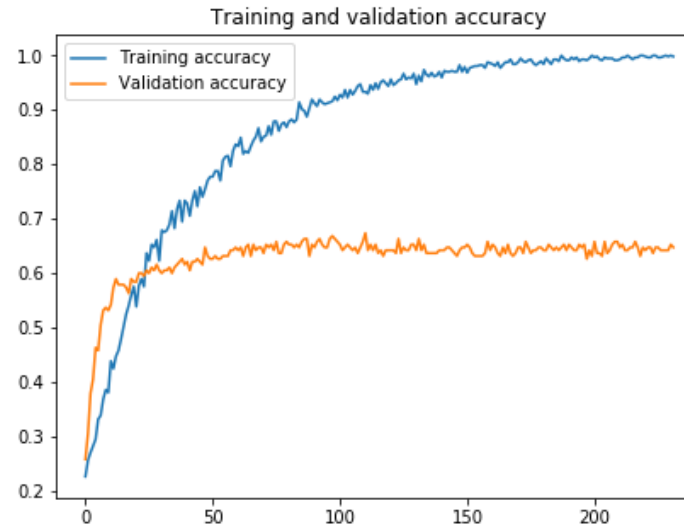
- Modèle pré-entraîné utilisé: VGG16



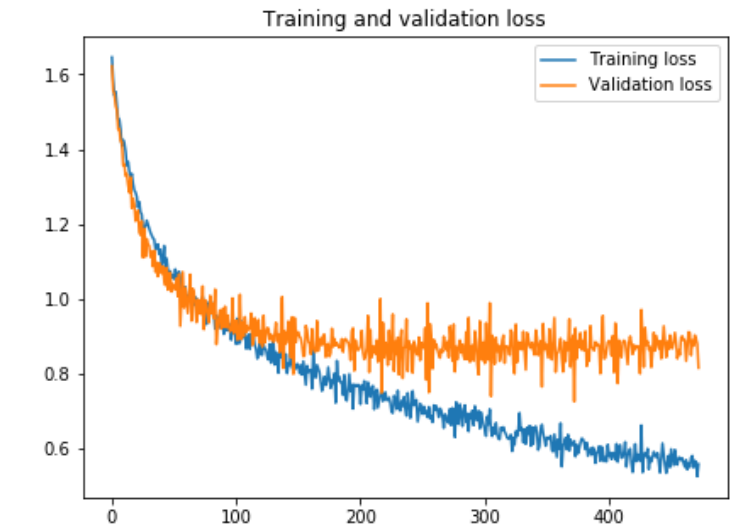
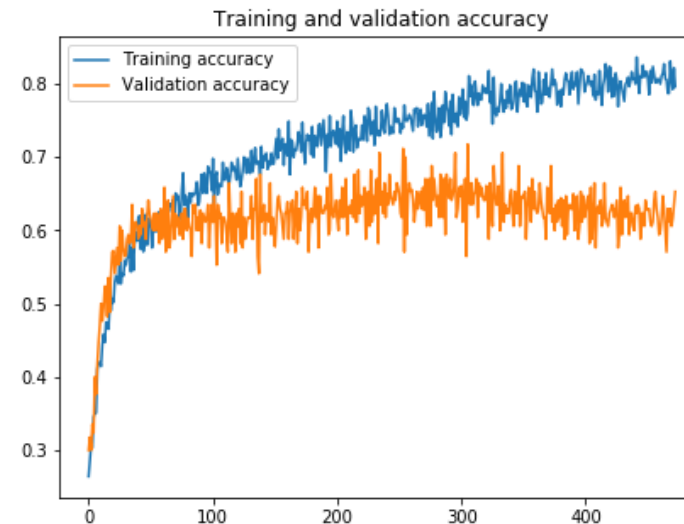
Couches de convolution

Transfer Learning – Extraction de features (5 races)

Sans data-augmentation :
60.42 %

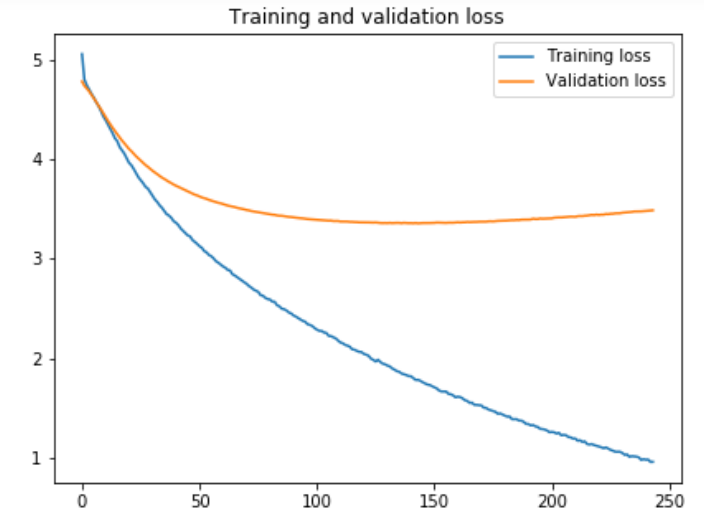
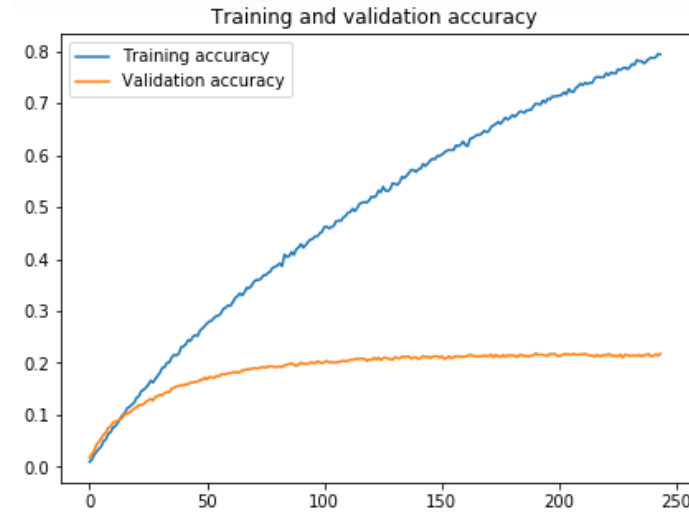


Avec data-augmentation :
62.50 %

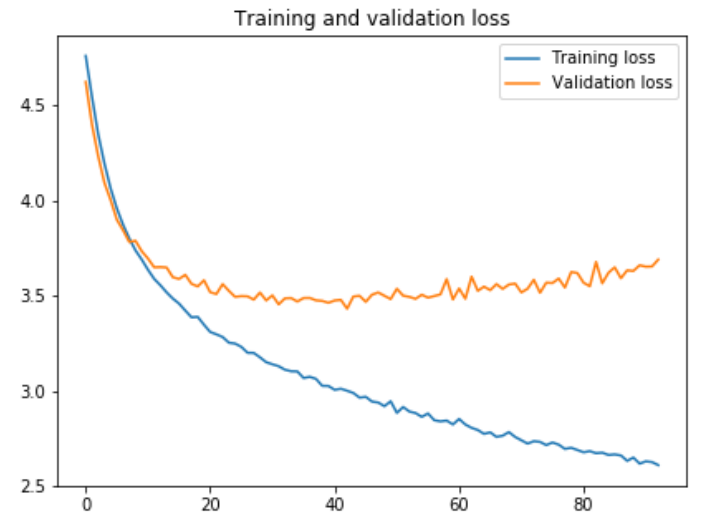
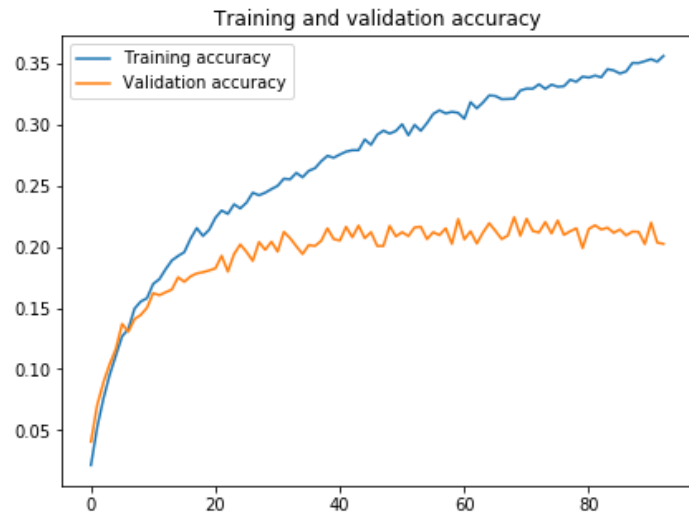


Transfer Learning – Extraction de features (120 races)

Sans data-augmentation :
21.58 %

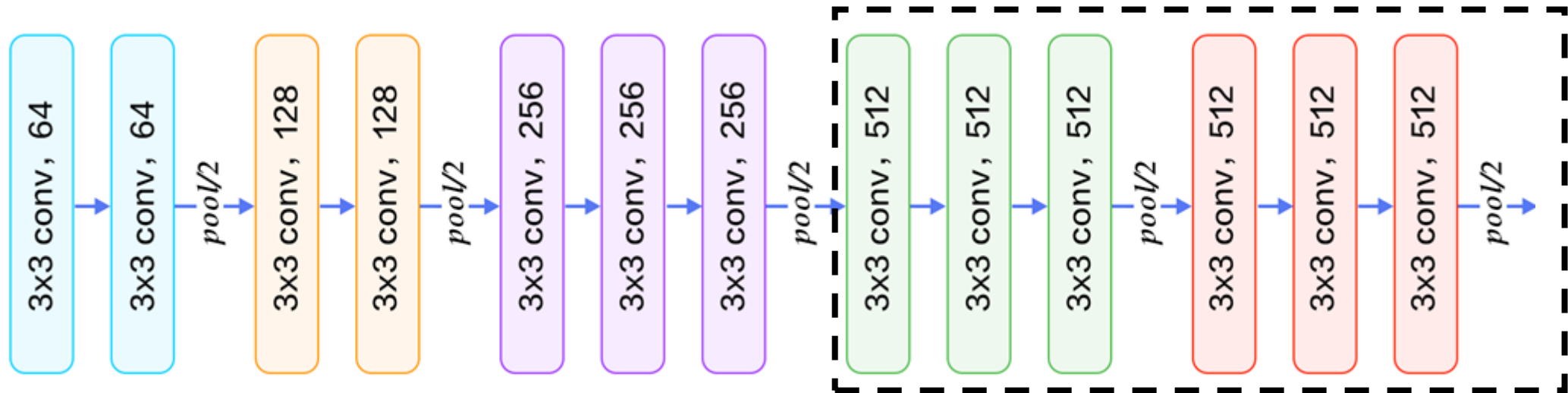


Avec data-augmentation :
21.94 %



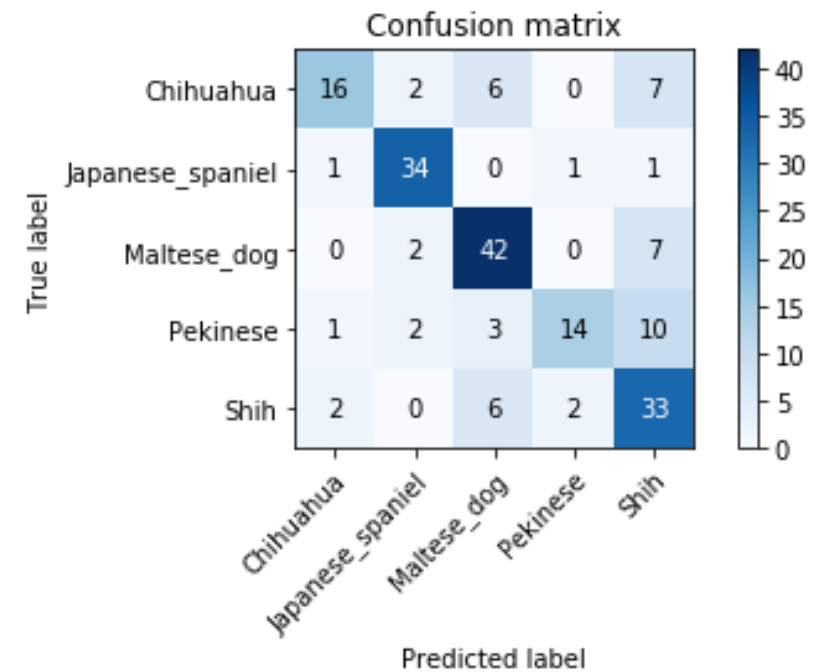
Transfer Learning – Fine tuning

- On repart des poids obtenus précédemment avec l'extraction de features
- On re-entraîne aussi les dernières couches de convolution (6 sur 13)

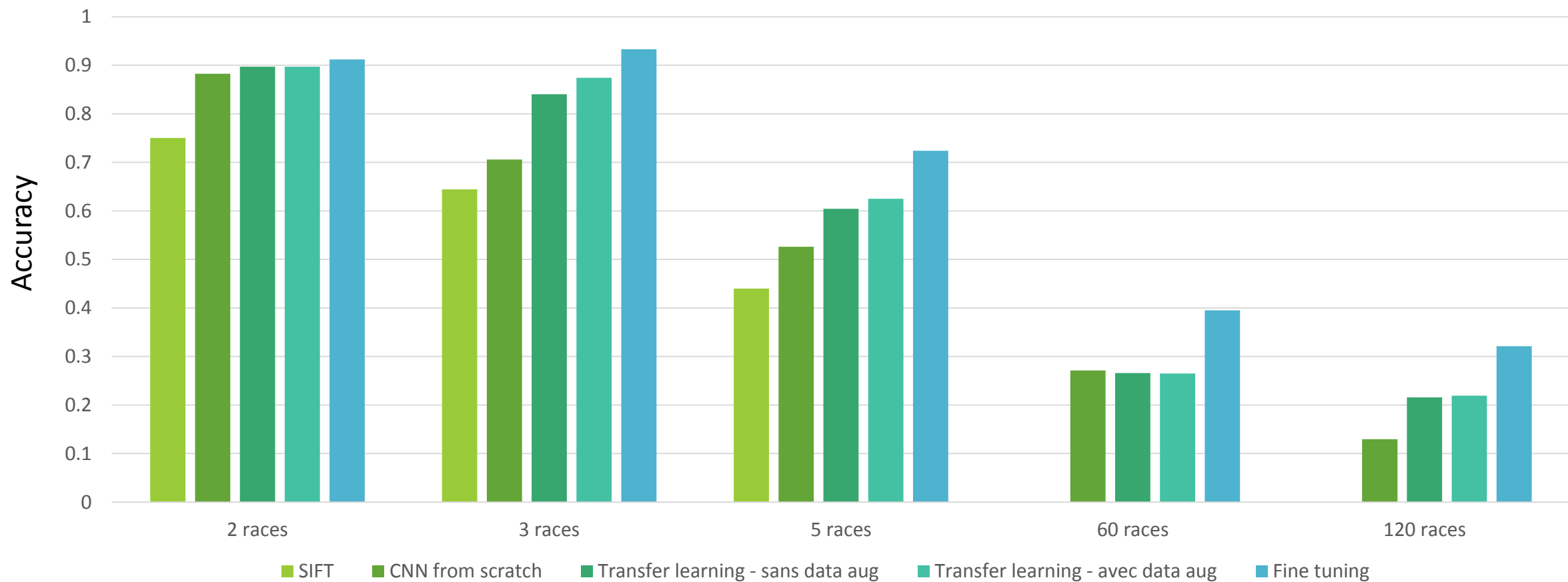


Résultats

Nombre de races	Accuracy
2	91.18 %
3	93.28 %
5	72.40 %
60	39.49 %
120	32.10 %



Comparaison des performances



Conclusions

- Les résultats illustrent l'efficacité des réseaux de neurones convolutionnels
- Points importants:
 - Data augmentation pour éviter l'over-fitting si le nombre de données est faible
 - Fine tuning pour améliorer rapidement les performances
- Pistes pour améliorer le modèle:
 - Aller plus loin dans le fine tuning en bloquant moins de couches
 - Obtenir plus de données