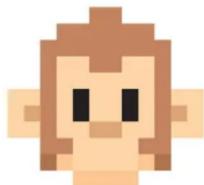


# DS P7 – Réalisez des indexations automatiques d'images



# Sommaire

1. Contexte
2. Exploration
3. Modélisation
4. API
5. Conclusion

# Contexte - Objectif

## Données

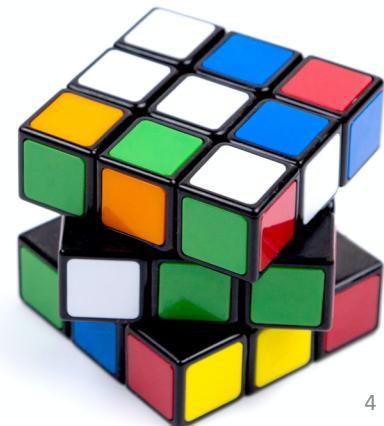
Dataset: StanfordDogsDataset (collection d'images de chiens, 20580 photos de tailles différentes, 120 races)

## Moteur de catégorisation de photo de chien

Modèle de détection de la race de chien à partir d'une photo

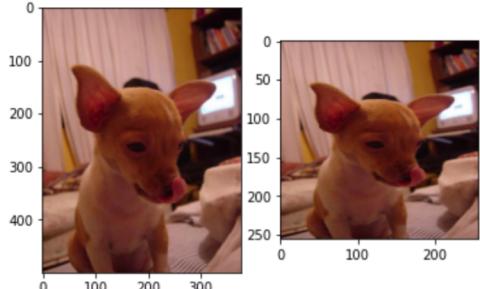
# Contexte - Problématique

- Quelles variables utiliser pour déterminer les races de chien?
- Quel méthode choisir pour identifier la race d'un chien sur une photo?
- Comment mesurer la performance du modèle?
- Comment gérer une volumétrie conséquente de photos?

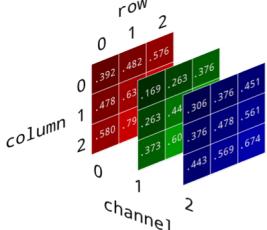


# Exploration - Prétraitement

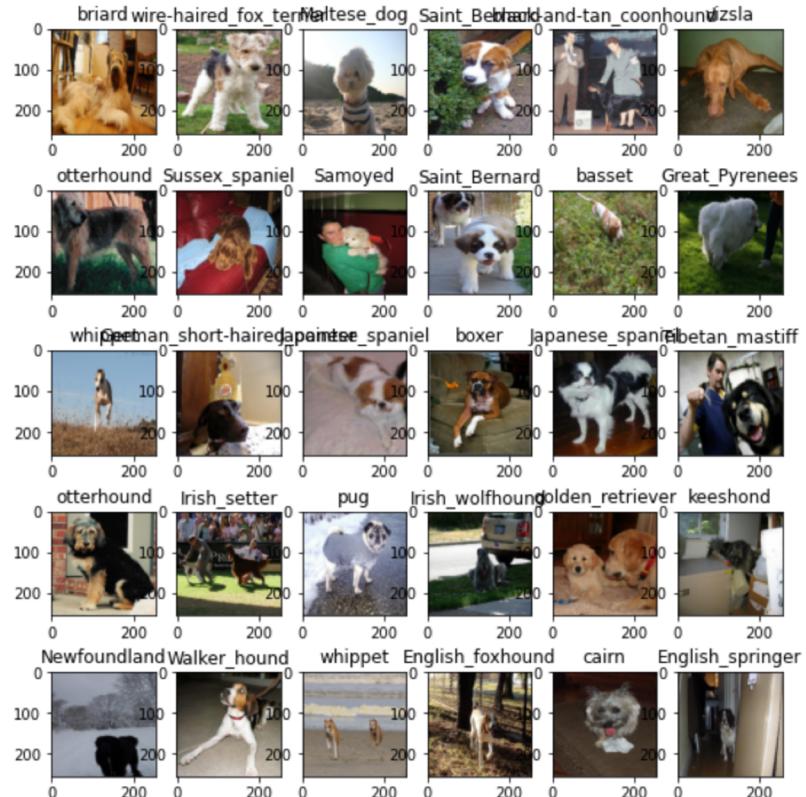
- Homogénéisation des images l: 256 x h:256



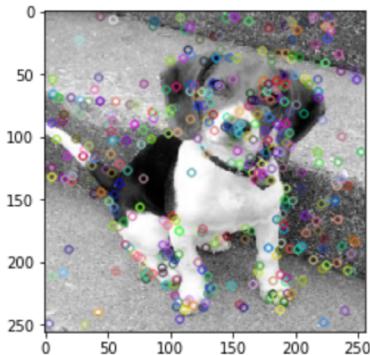
- Transformation image jpg en tableau numpy à 4 dimensions (index, largeur, hauteur, canal rgb)



- Indexation aléatoire des images dans le tableau
- Labellisation en fonction des noms de répertoire

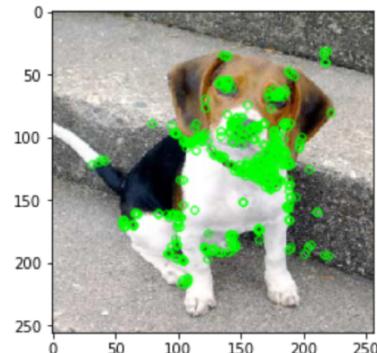


# Modélisation – SIFT vs ORB



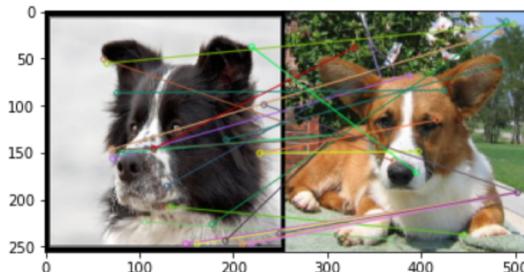
SIFT (scale invariant feature transform)

- Plus sensible au bruit
- Invariant aux rotation et mise à échelle
- Licence



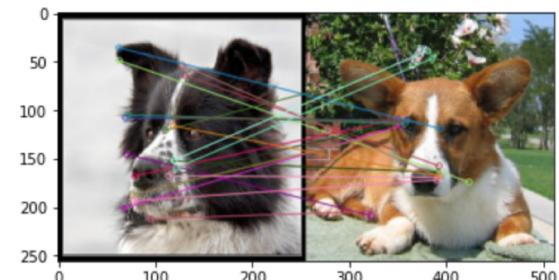
ORB (oriented fast and rotated brief)

- Coût faible, adapté pour du temps réel
- Moins sensible au bruit
- Plus sensible à la rotation
- Gratuit

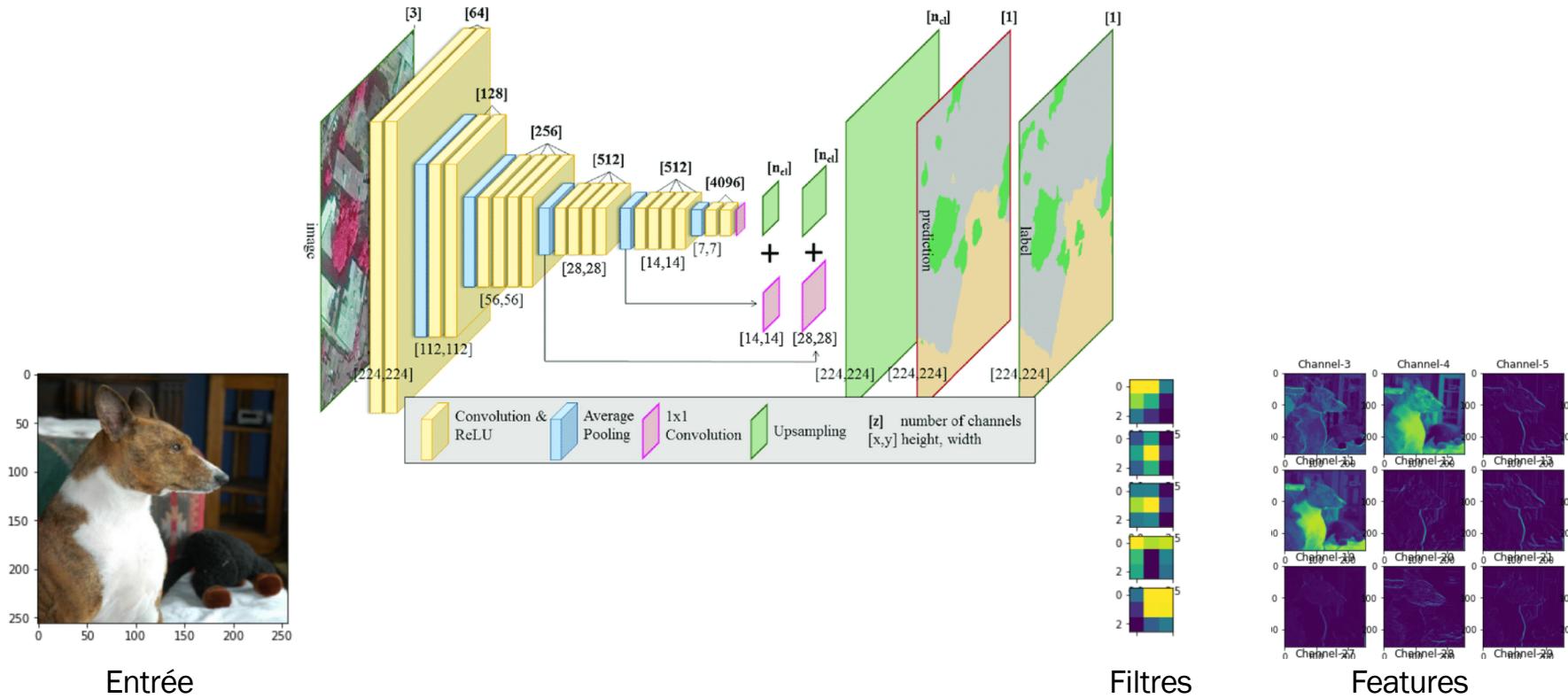


Méthode *Bag Of Features*

- Extraction des features (avec ORB)
- Clustering Kmeans
- Matrice des distances
- Classifier les clusters avec SVM

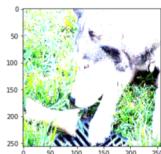


# Modélisation – CNN Principes

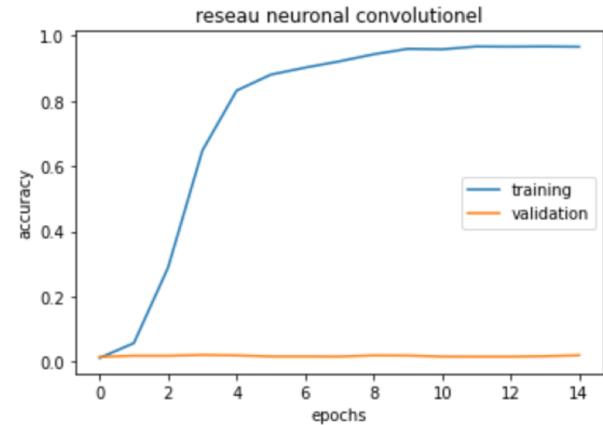
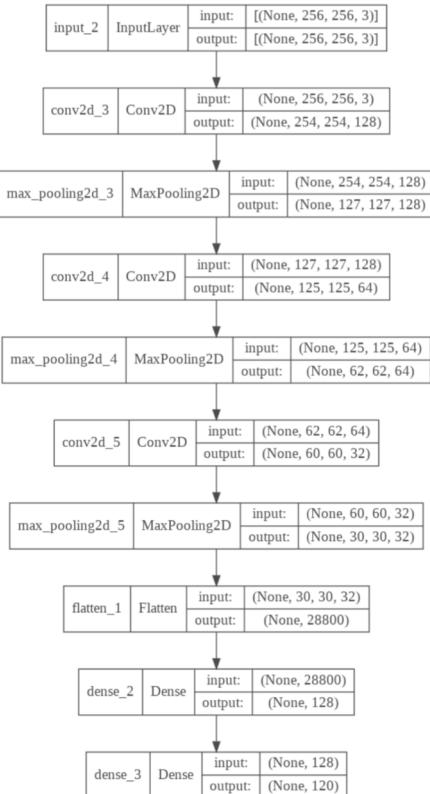


# Modélisation – CNN Custom

- Entrées:
  - 10000 photos
  - taille 256x256x3 (rgb)
- Prétraitement:
  - /255, Centrage, Normalisation



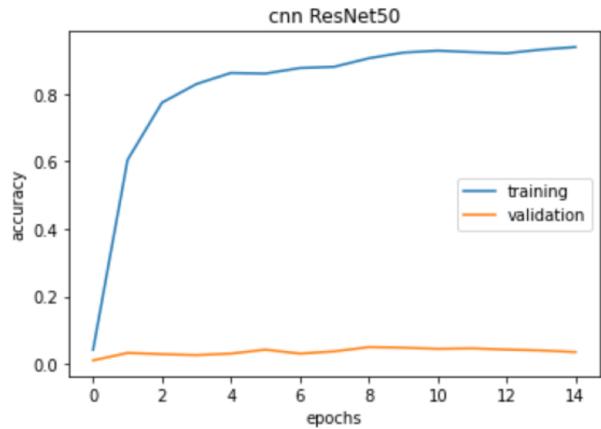
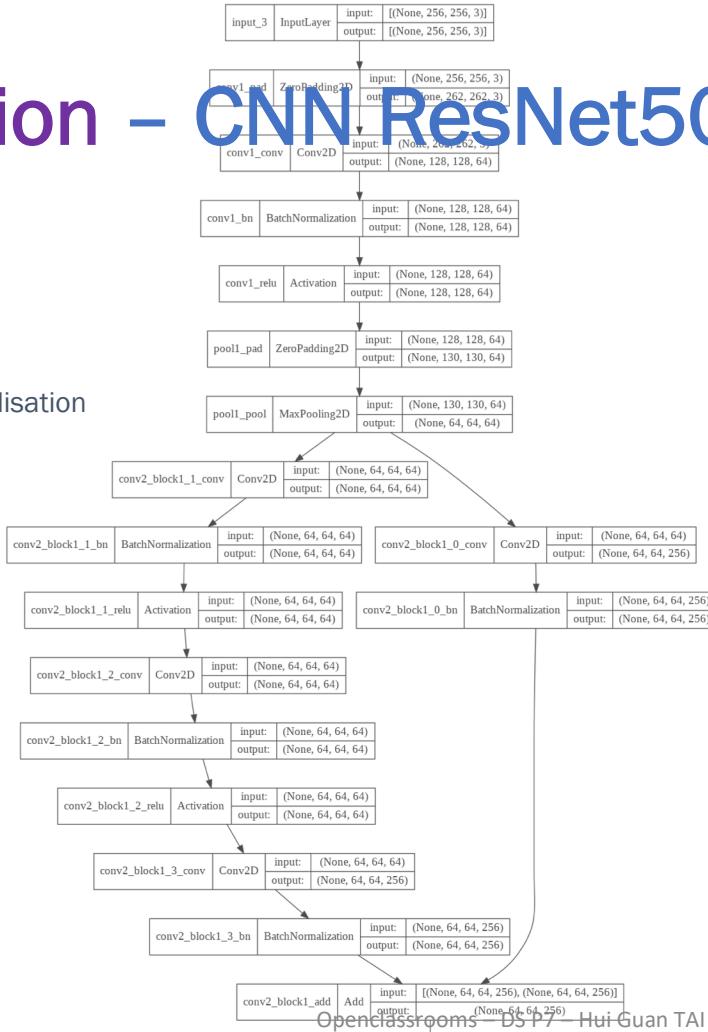
- Itérations: 15
- Couches: 8
- Sortie: 120 classes



- Exactitude:
  - Apprentissage: 0.97
  - Validation: 0.02
- L'exactitude est > 96% pendant l'apprentissage au bout de 8 itérations
- Très faible taux d'exactitude: le modèle identifie correctement au plus 2% des photos de chiens

# Modélisation - CNN ResNet50

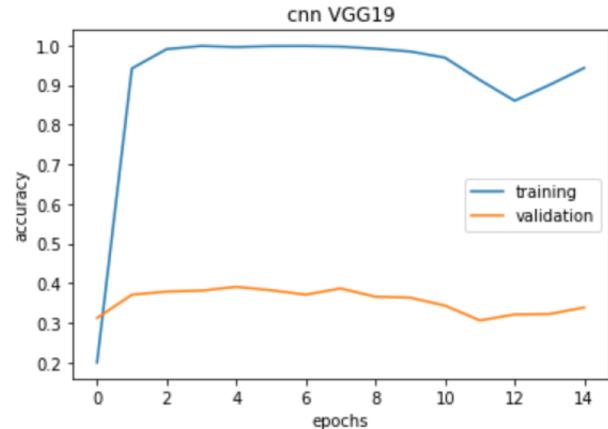
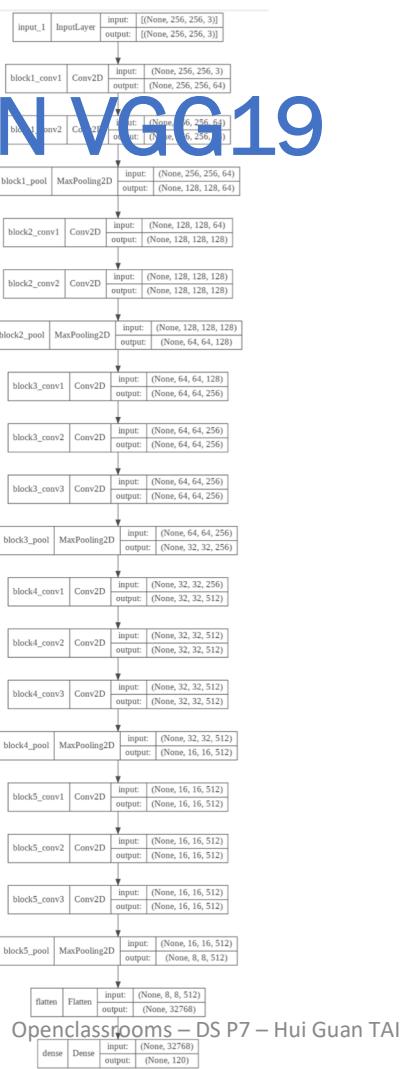
- Entrées:
  - 10000 photos
  - taille 256x256x3 (rgb)
- Prétraitement:
  - /255, Centrage, Normalisation
- Itérations: 15
- Couches: 50
- Sortie: 120 classes



- Exactitude:
  - Apprentissage: 0.94
  - Validation: 0.03
- L'exactitude est > 90% pendant l'apprentissage au bout de 9 itérations
- Très faible taux d'exactitude: le modèle identifie correctement au plus 3% des photos de chiens

# Modélisation – CNN VGG19

- Entrées:
  - 10000 photos
  - taille 256x256x3 (rgb)
- Prétraitement:
  - /255, Centrage, Normalisation
- Itérations: 15
- Couches: 19
- Sortie: 120 classes



- Exactitude:
  - Apprentissage: 0.94
  - Validation: 0.34
- L'exactitude est > 94% pendant l'apprentissage au bout de 3 itérations
- Exactitude de validation > 0,32%
- Infléchissement des scores d'apprentissage et de validation après 6 itérations
- Surapprentissage

# API – Déploiement



## FastAPI

- Tâches Asynchrones



## docker

- Isolation des composants
- Léger, rapide à déployer



## HEROKU

<https://dogtype.herokuapp.com>

- Gratuit
- 512 Mb Ram

## Quelle famille de chien?

Selectionner une image de chien à identifier

Choose File no file selected



identifier

Le chien sur l'image n02108915\_9899.jpg (117930 octets) correspond à la famille des French\_bulldog

[https://github.com/8huit/OC\\_DS\\_P7](https://github.com/8huit/OC_DS_P7)

# Conclusion

- Les +
  - Visualisation des couches de convolution
  - Légereté du modèle, déployable sur heroku
  - Site Graphique
- Les -
  - Faible exactitude sur jeu de test ~ environ 30%
  - Surapprentissage des modèles malgré la normalisation des images
  - Manque de connaissance pour optimiser un réseau neuronal de convolution
- Pour la suite
  - Ajout d'image (20000)
  - Image augmentation
  - Ajout de couches de convolutions sur des modèles transférés
  - Ajout de couches drop out et callback earlystop pour éviter le surapprentissage
  - Implémentation du modèle Bag of features