

INDEXATIONS AUTOMATIQUES D'IMAGES

Projet 7

Azim Makboulhousen
09 Juillet 2018



Sommaire

- Introduction
- Les données
- Préparation pour modélisation
- Apprentissage avec approche classique de machine learning
- Apprentissage avec approche basée sur réseaux de neurones
- Outil de prédiction
- Conclusion

Introduction

Objectif du projet

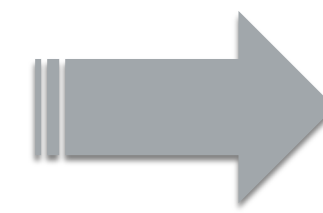
- Aider une association de protection d'animaux dans son travail de référencement de photos
- Réaliser un programme de reconnaissance automatique de la race d'un chien à partir de sa photo
- Solution basée sur les approches classiques de classification
- Solution basée sur l'utilisation de CNN



Les données

Stanford Dog Dataset

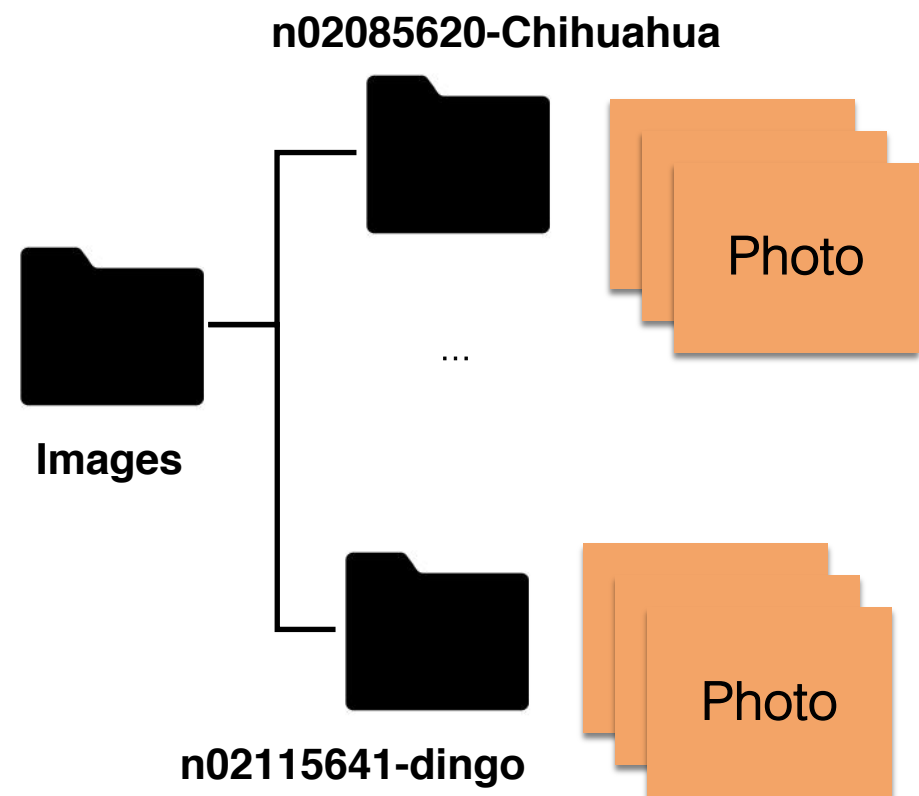
- ❑ Données qui seront utilisées pour l'apprentissage
- ❑ Un peu plus de **20 000 photos** de chiens
- ❑ **120 races** de chiens



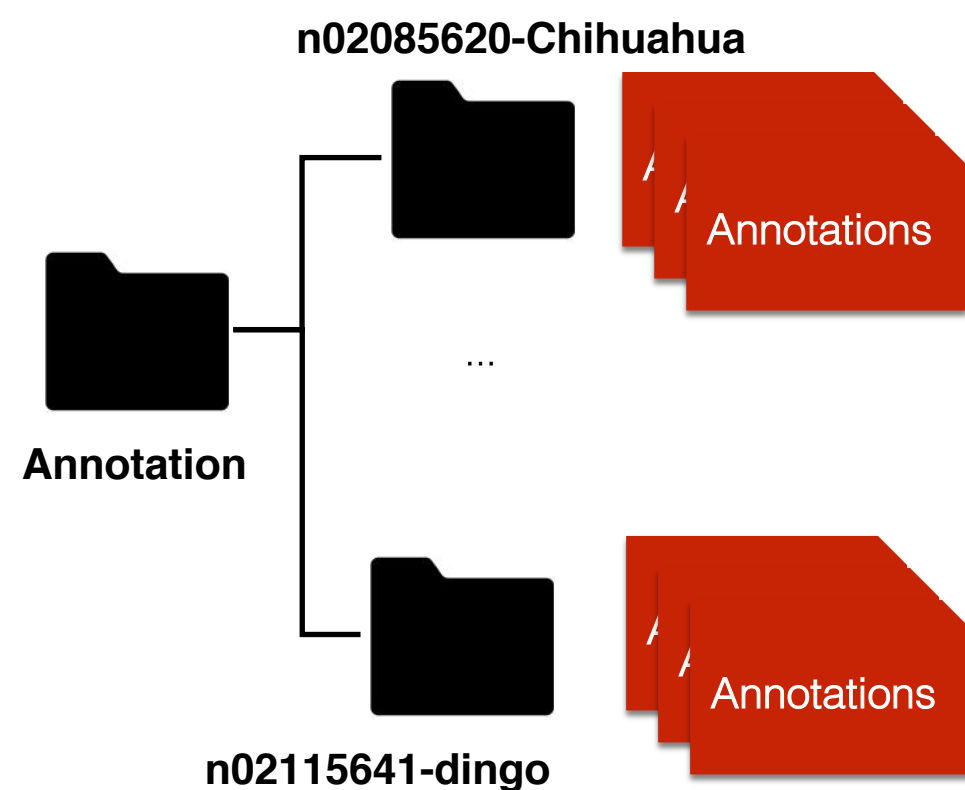
En moyenne **180 photos**
par **race** de chiens



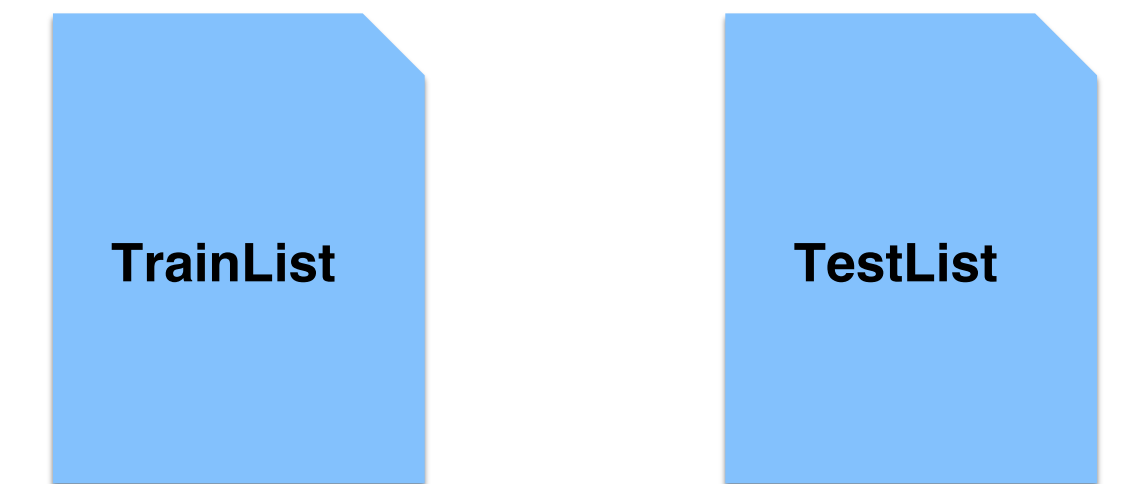
Description des données



- ❑ 1 dossier par race de chien
- ❑ Chaque dossier contient les photos de chiens

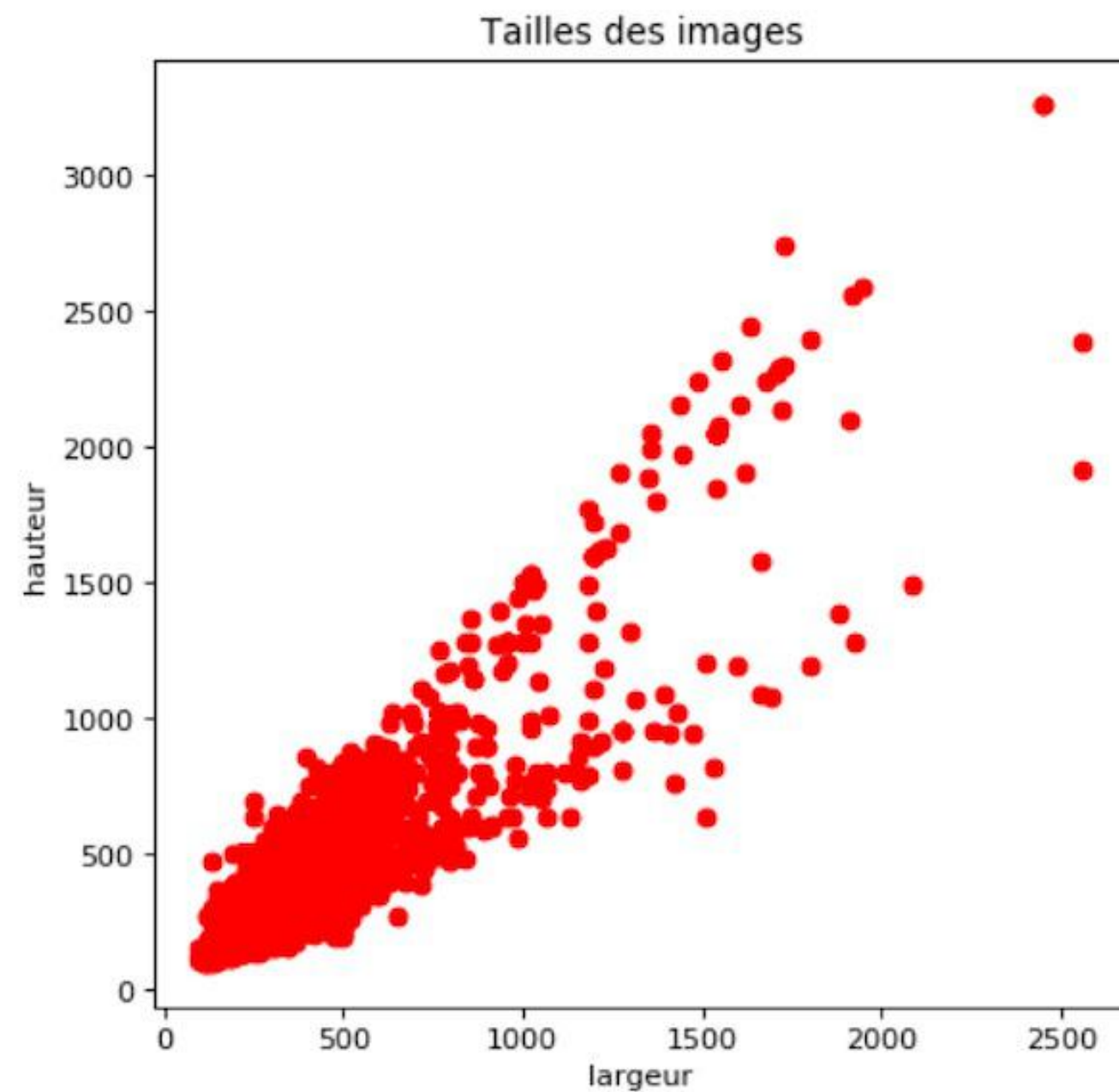


- ❑ 1 fichier ***Annotations*** par photo
- ❑ Contient descriptions image:
 - Largeur
 - Hauteur
 - Nom de la race de chien
 - Rectangle de délimitation centré sur le chien



- ❑ Liste des images à utiliser pour l'entraînement
- ❑ Liste des images à utiliser pour les tests

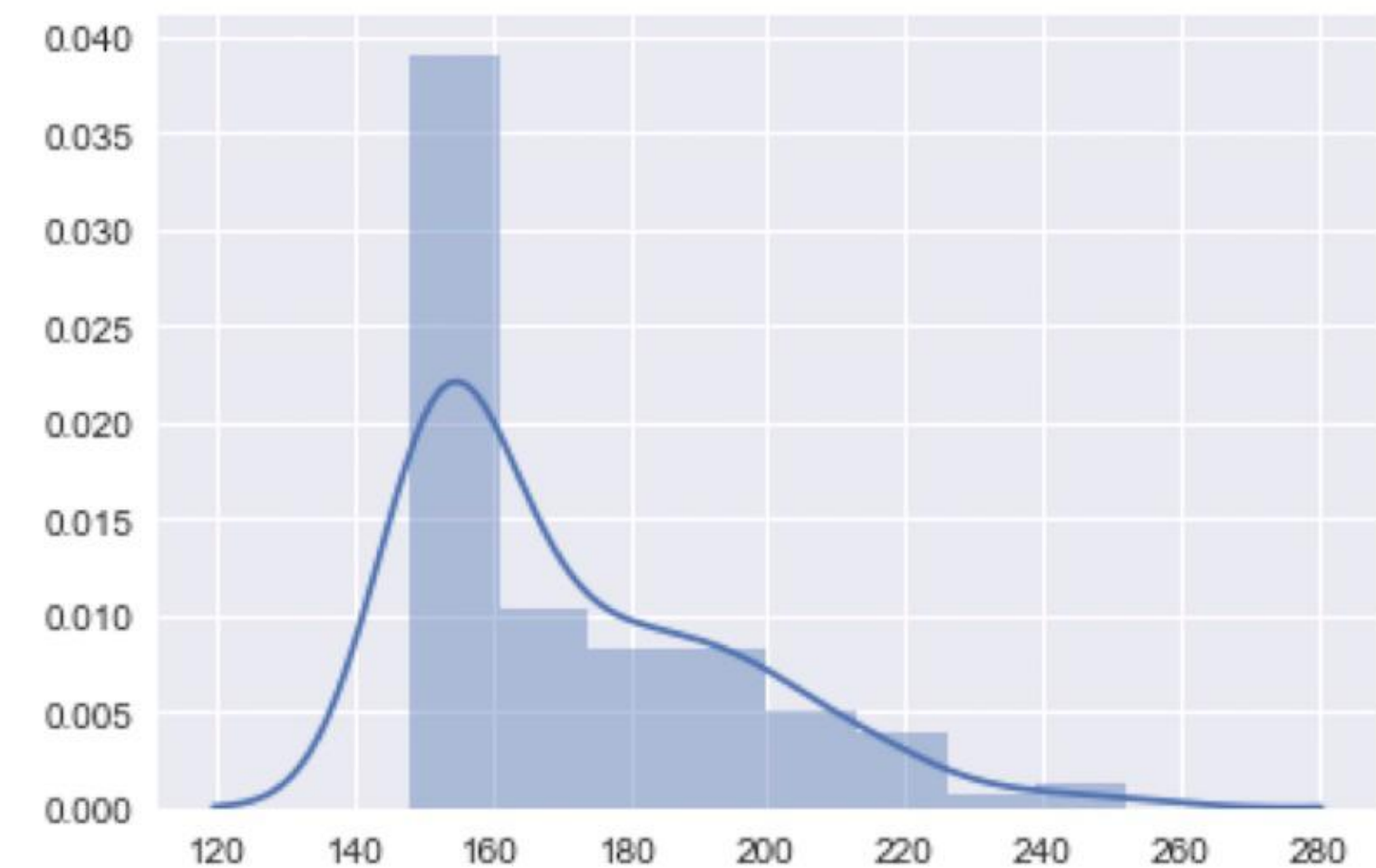
Les photos



- ❑ Photos en couleur
- ❑ Taille des images variables
- ❑ Majorité avec hauteur/largeur inférieures à 500 pixels



- ❑ Entre 150 et 250 images par race
- ❑ Essentiellement 150 images par race

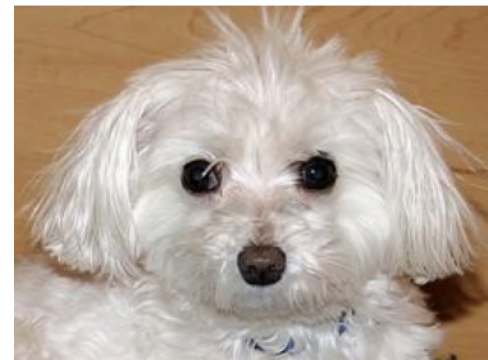
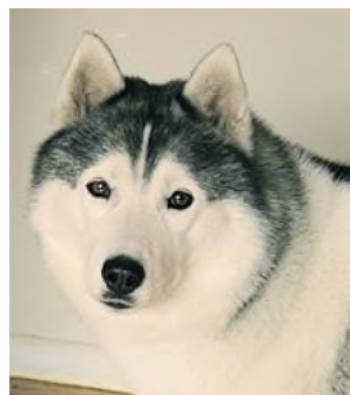


Préparation pour modélisation

Notre démarche

❑ Sélection de **4 races** de chiens en raison de la limitation des ressources matérielles

- 'Siberian_husky', 'Chihuahua', 'Maltese_dog', 'Scotch_terrier'



❑ Utilisation des fichiers **TrainList** et **TestList** pour se constituer les jeux de données d'entraînement et de test

❑ 2 approches pour l'apprentissage :

- Technique de **Bag of Visual Words** puis application d'un algorithme de classification
- **Réseaux de neurones à convolution**

Configuration des données – Data Augmentation

➡ Augmentation du nombre d'images pour l'apprentissage



Entraînement :

- ☐ Image d'origine
- ☐ Image centrée sur le chien
(délimiteurs du fichier annotation)
- ☐ Image pivotée
- ☐ Image pivotée centrée sur le chien
- ☐ Rotation aléatoire de l'image

Test :

- ☐ Image centrée sur le chien
(délimiteurs du fichier annotation)

Train

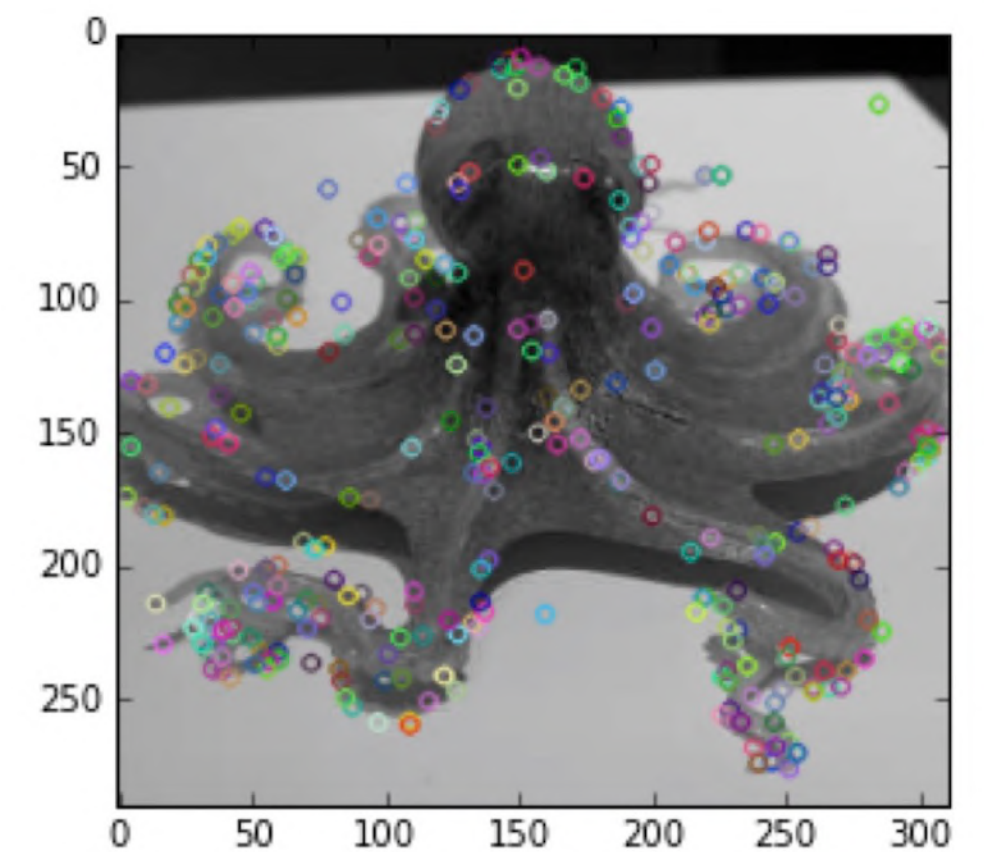
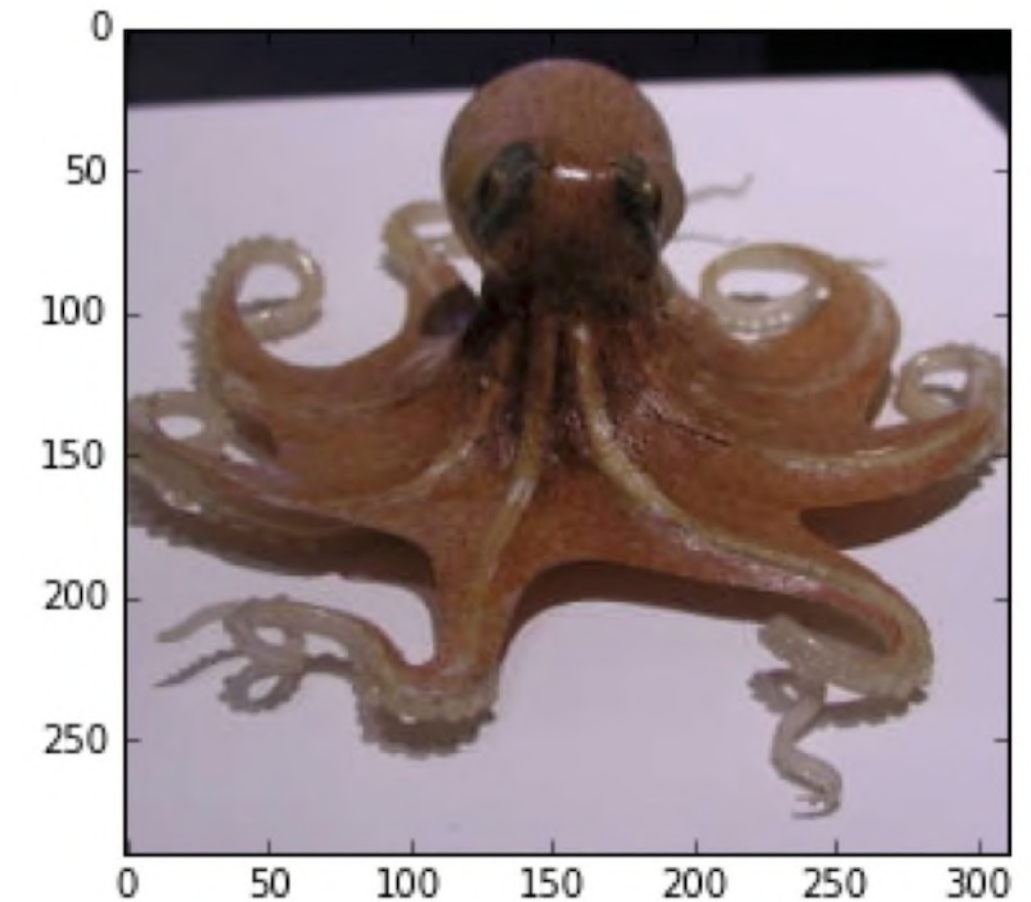
X 5 (2000 photos)

Test

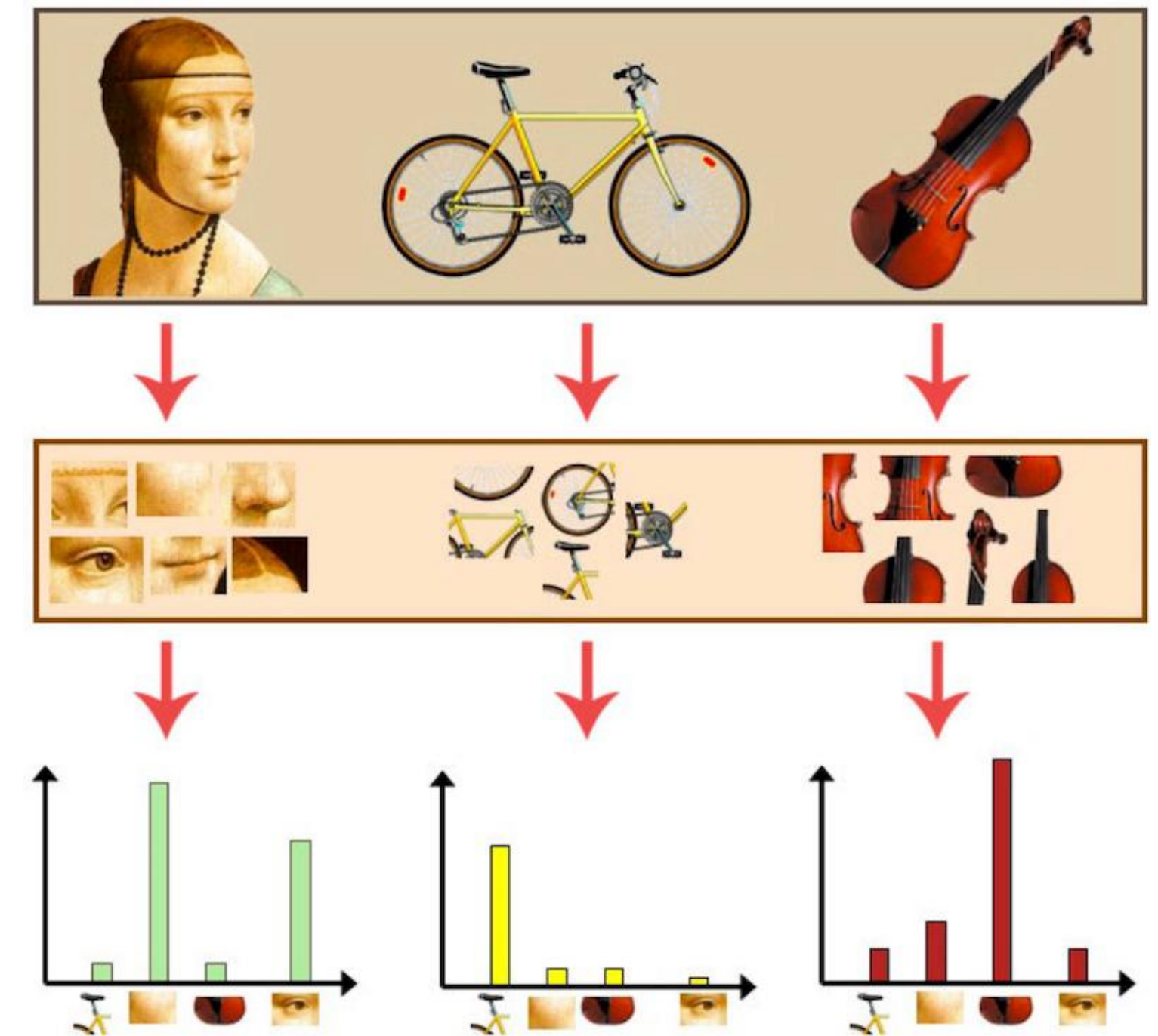
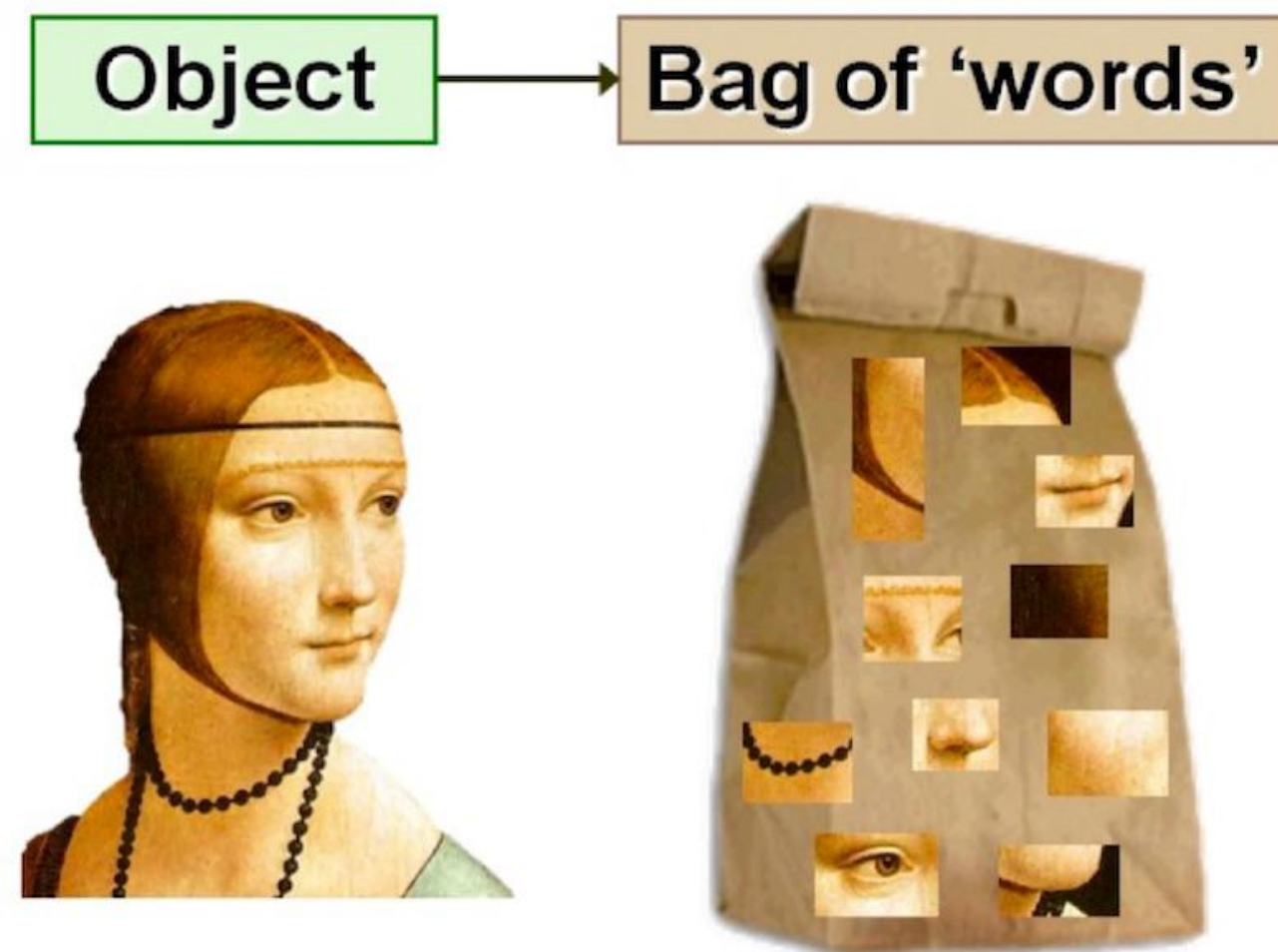
Apprentissage avec méthodes supervisées

Décrire une image

- ❑ Calculer la similarité entre images est complexe pour une machine.
- ❑ Notre système de visualisation se base sur des points intéressants d'une image
- ❑ Capacité à reconnaître la même image même si elle a changé
- ❑ La machine va utiliser le même principe et extraire les points intéressants d'une image

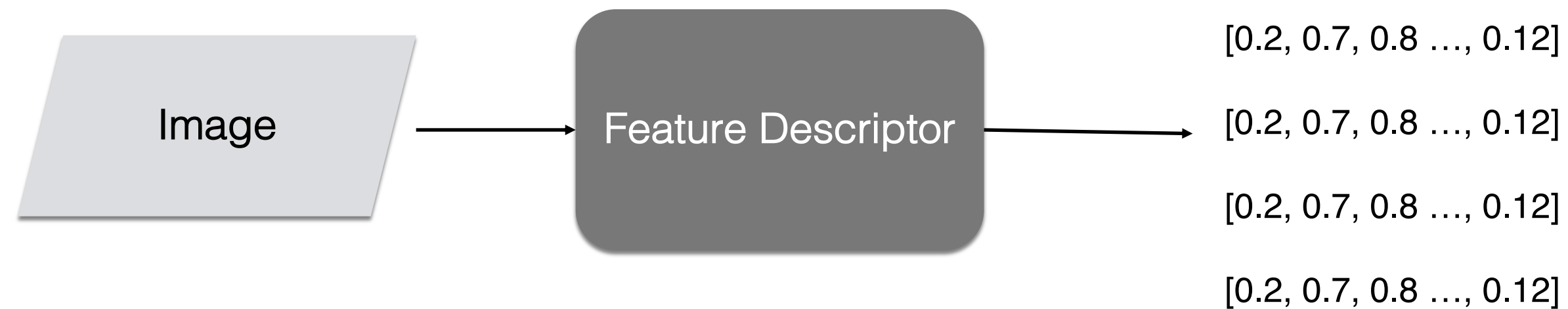


Bag of Visual Words



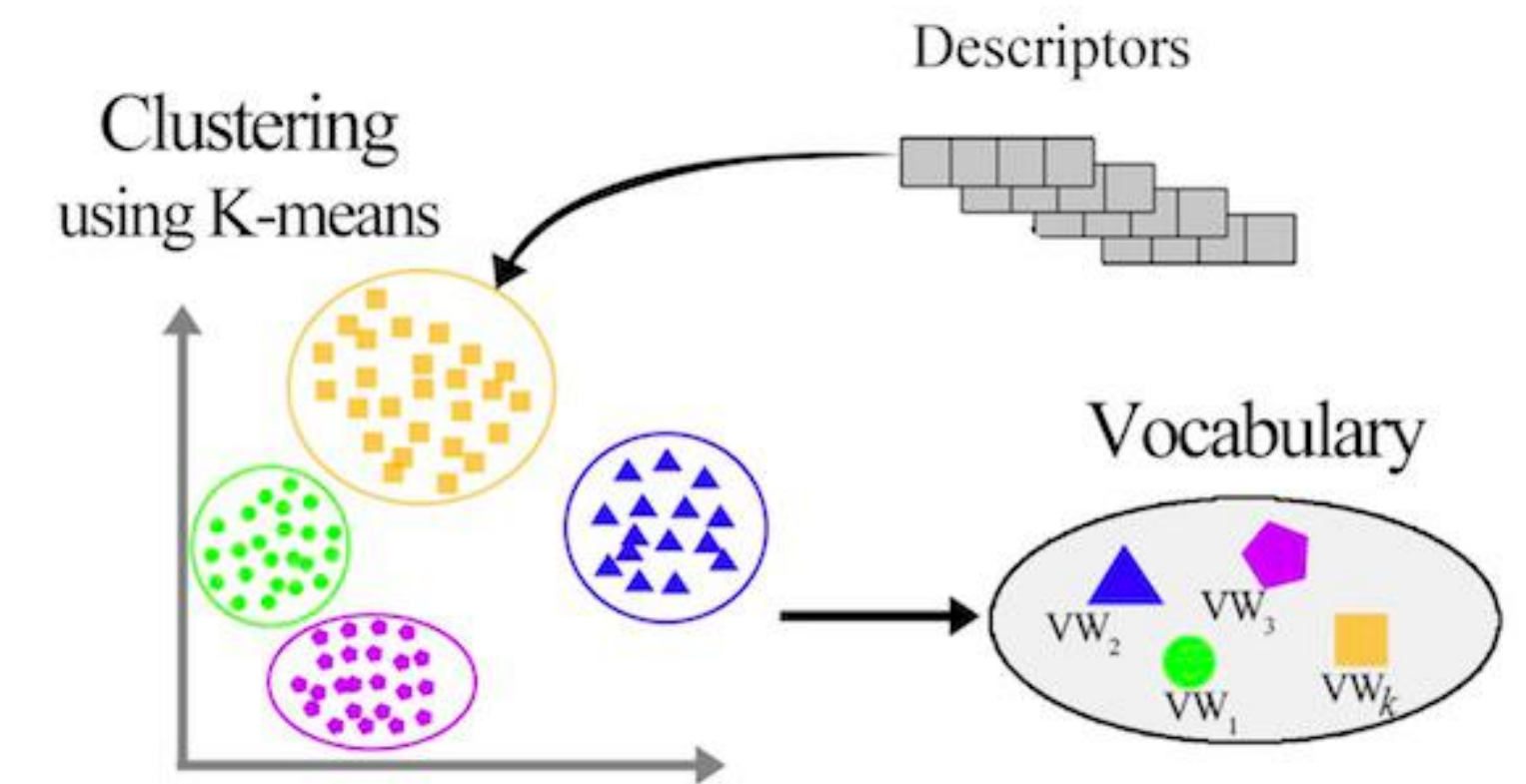
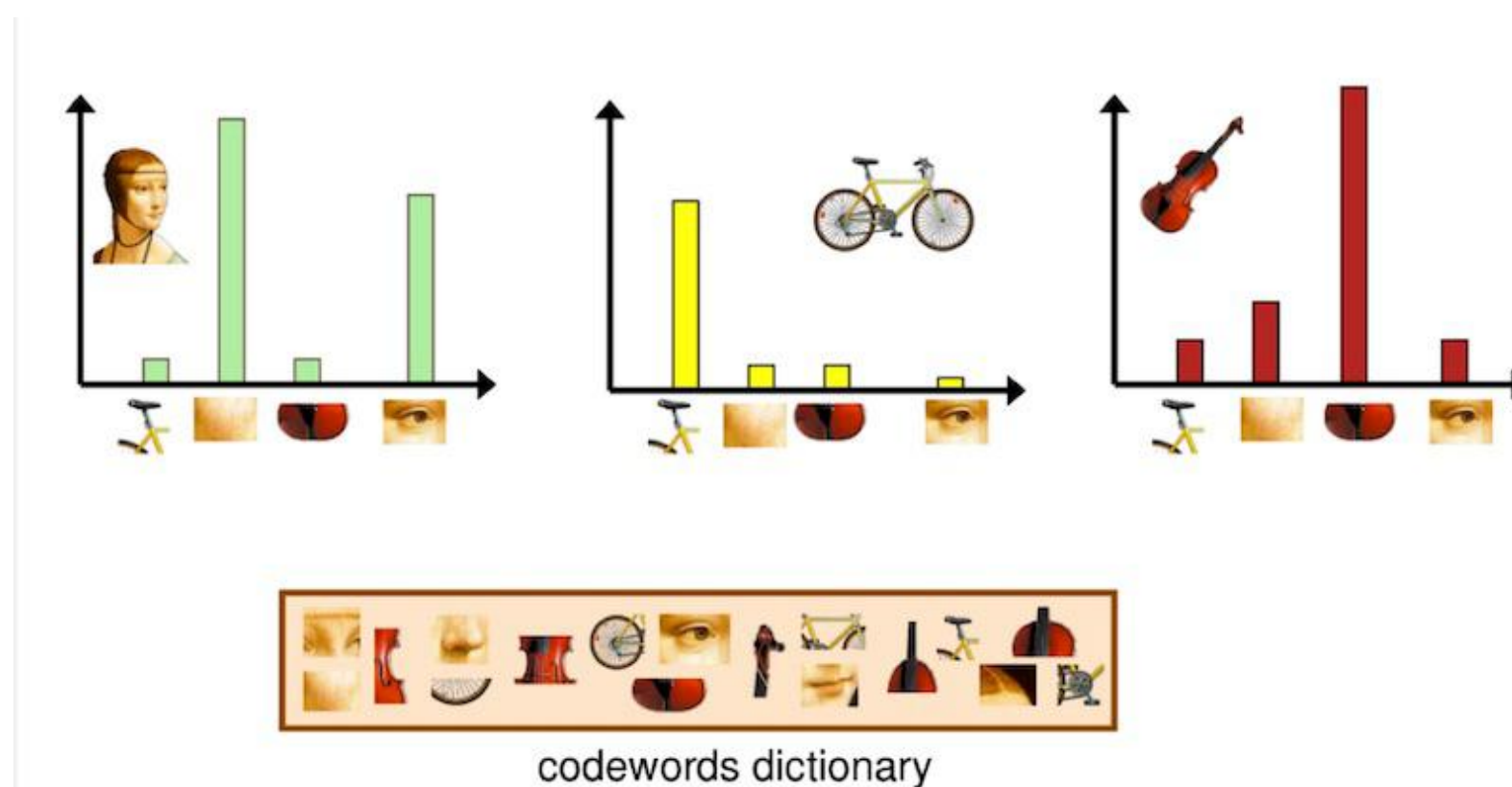
- ❑ Concept similaire au Bag Of Word appliqué aux documents textuels
- ❑ A la place de mots clés nous avons des blocs/morceau d'images (features)
- ❑ On construit un vocabulaire de « mots visuels » appelé codebook
- ❑ On peut alors représenter une image par un histogramme de « mots visuels » appelé Bag of Visual Words

Bag of Visual Words



1. Extraction des points d'intérêts (keypoints) d'une image → utilisation de l'algorithme SIFT

2. Construction d'un dictionnaire avec k-means en regroupant les features similaires



3. Pour une image, on détermine dans quelle partition se trouve chacun de ses keypoints et on construit un histogramme

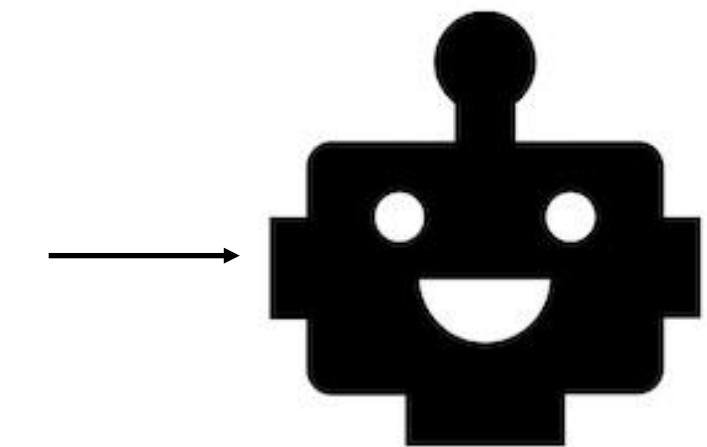
Classification des images

❑ Utilisation des algorithmes supervisés de classification sur la matrice Bag of Visual Words pour apprentissage :

- SVM
- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- AdaBoost

	Cluster 1	Cluster 2	...	Cluster k
Image 1				
...				
Image N				

Chihuahua
...
Maltese Dog



❑ Choix de l'algorithme qui a donné les meilleurs résultats en terme de prédiction

❑ Validation croisée pour évaluer les algorithmes

❑ Utilisation des scores : accuracy, rappel et précision et affichage de la matrice de confusion

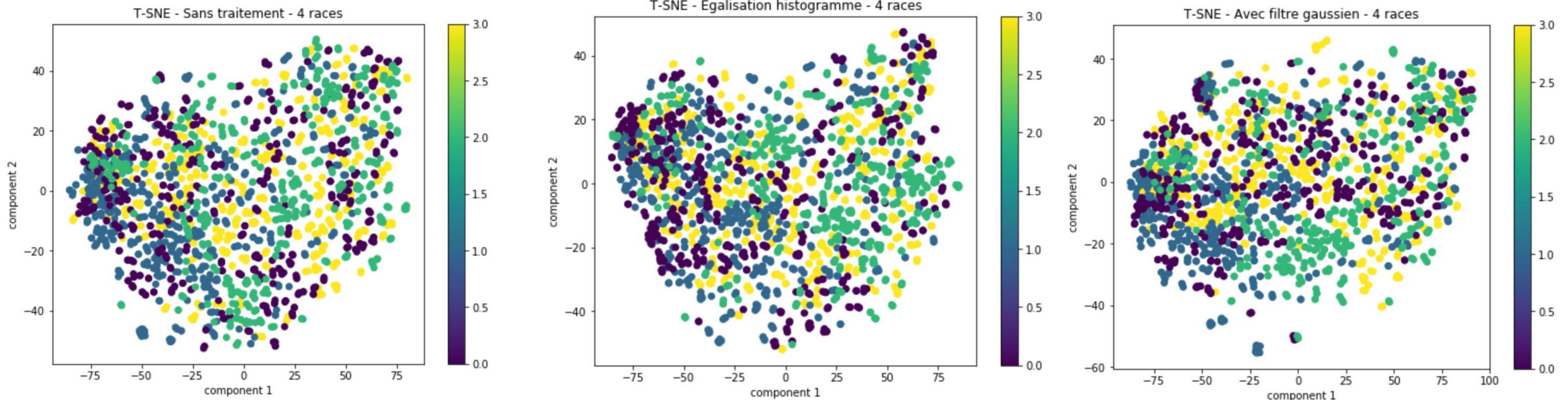
Résultats –Score prédiction jeu de test

	SVM	Logistic Regression	Decision Tree	Random Forest	Gradient Boosting	AdaBoost
accuracy	63,56%	58,19%	43,79%	61,02%	48,59%	55,37%
f1-score	62%	58%	45%	62%	49%	55%

- ❑ Le Support Vector Machine est celui qui donne le meilleur résultat
- ❑ C'est l'algorithme que nous sélectionnons et que nous allons essayer d'optimiser.

SVM – Prétraitement images

- ❑ Tests de pré-traitement des images avant extraction des features :
Egalisation histogramme, Filtre Gaussien et Suppression de l'arrière plan
- ❑ Le Filtre gaussien améliore légèrement la prédiction
- ❑ La suppression de l'arrière plan a été abandonné en raison du temps de calcul
- ❑ Réduction de dimension avec tSNE pour visualiser les features extraites



Tuning SVM

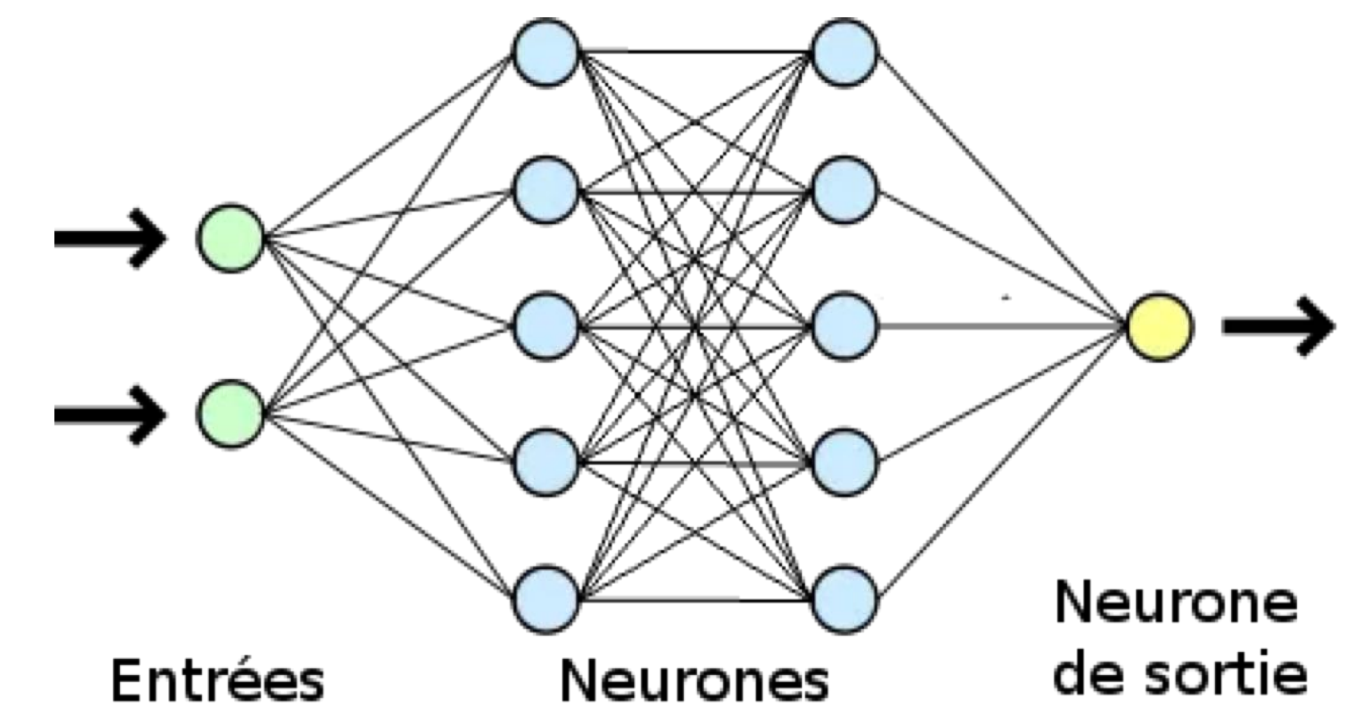


- ☐ Nombre de clusters K-Means donnant la meilleure performance : 100, 300, 500 et 1000
- ☐ On obtient le meilleur résultat sur les données de test avec 100 clusters
- ☐ Tuning de l'hyper-paramètre de pénalité
- ☐ Recherche sur grille pour évaluation
- ☐ Performance reste à 63%

Réseaux de neurones à convolution

CNN – Réseau neuronal convolutif

- ❑ Méthode d'apprentissage profond (Deep Learning)
- ❑ Sous catégorie des Réseaux de Neurones spécialement conçu pour l'apprentissage d'image.
- ❑ Réseau de Neurones est composé d'unités d'apprentissage appelées neurones.
- ❑ Les neurones apprennent à convertir un signal d'entrée (exemple : image) en un signal de sortie (exemple chien).
- ❑ Une fonction d'activation est utilisée pour déterminer le résultat de sortie du neurone.
- ❑ Plus le réseau est confronté à des données pour l'apprentissage meilleure il sera dans la prédiction mais plus long seront les calculs.



CNN – Réseau neuronal convolutif

❑ Convolution

- Couches en charge de l'extraction des features
- Calcul dans toute l'image si une caractéristique est présente en faisant un filtrage (convolution)

❑ Pooling :

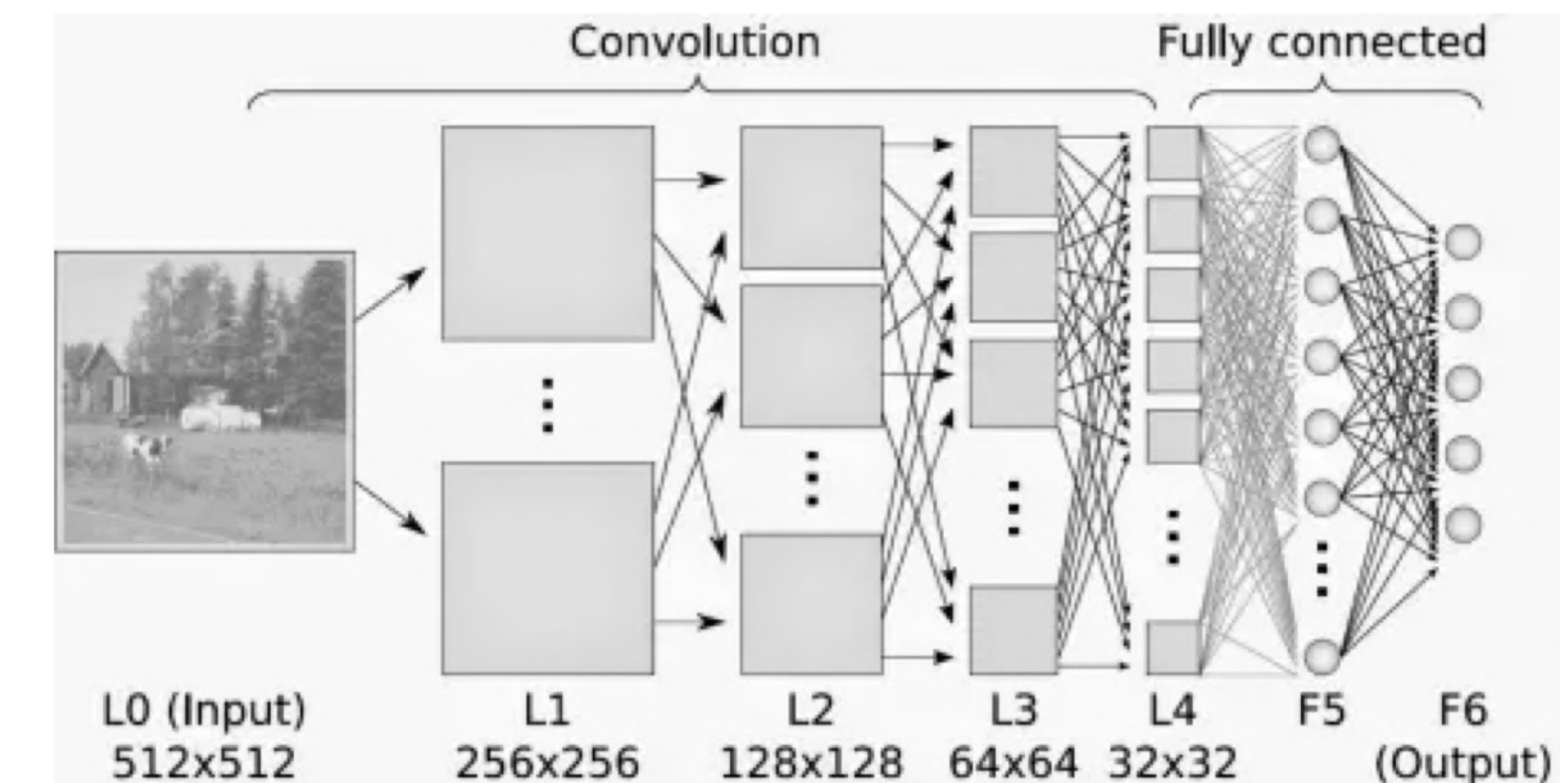
- Couche permettant de réduire la taille de l'image tout en préservant les informations les plus importantes

❑ Activation

- Contrôle comment est passé d'une couche à l'autre

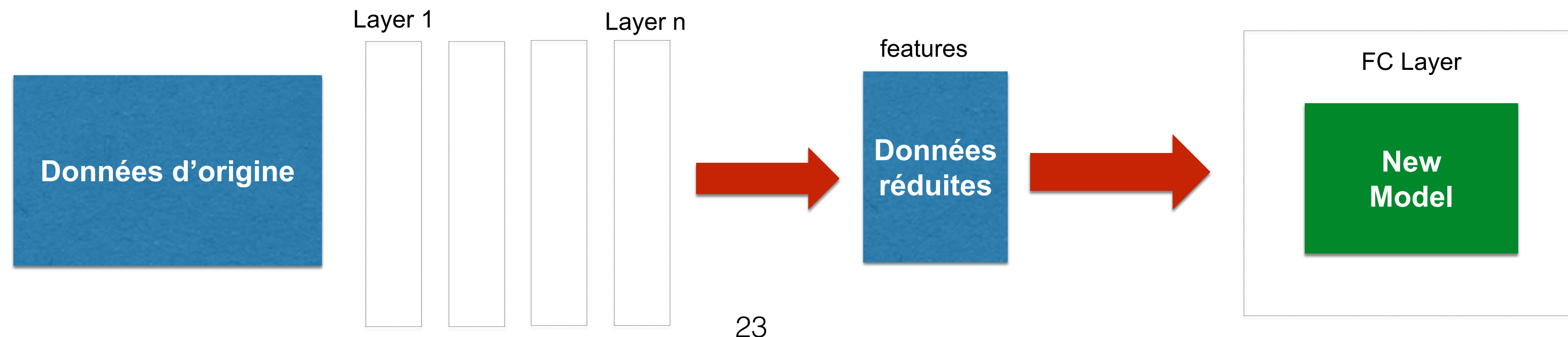
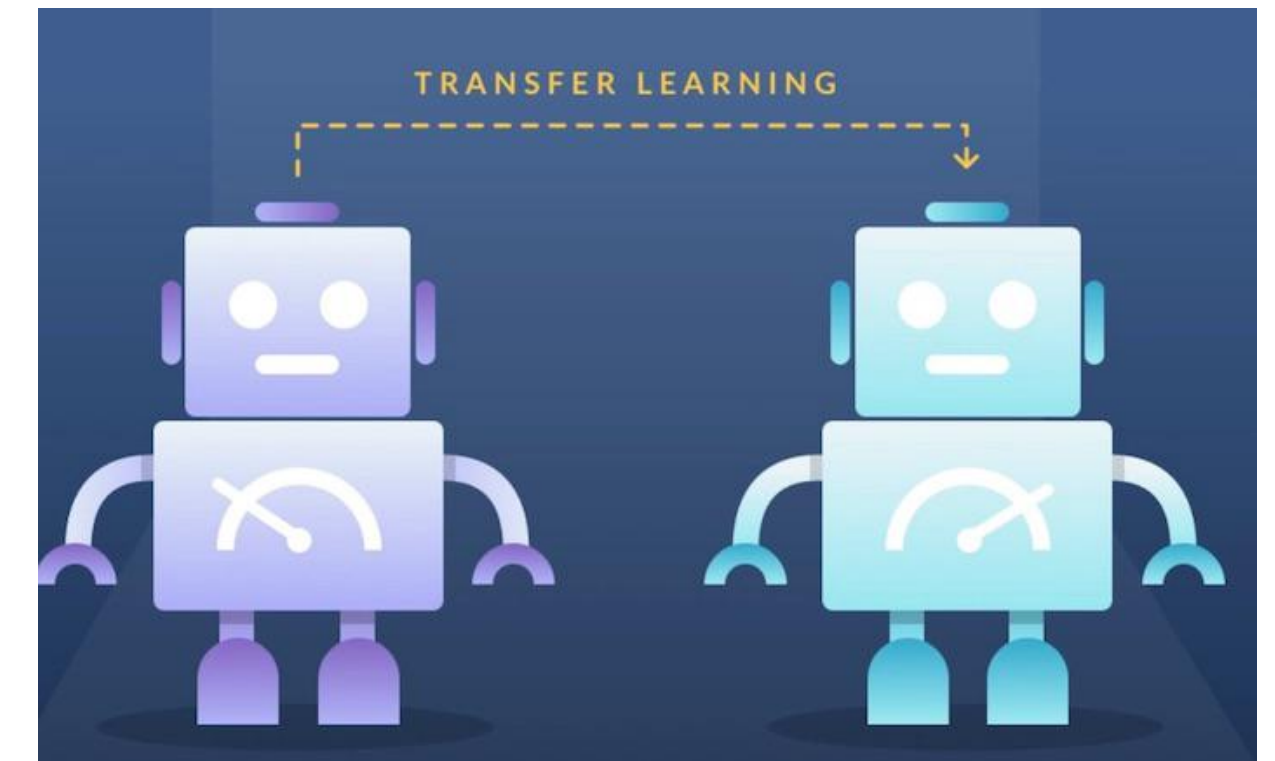
❑ Fully Connected :

- Couche de classification
- En sortie il va donner autant d'éléments qu'il y a de classes



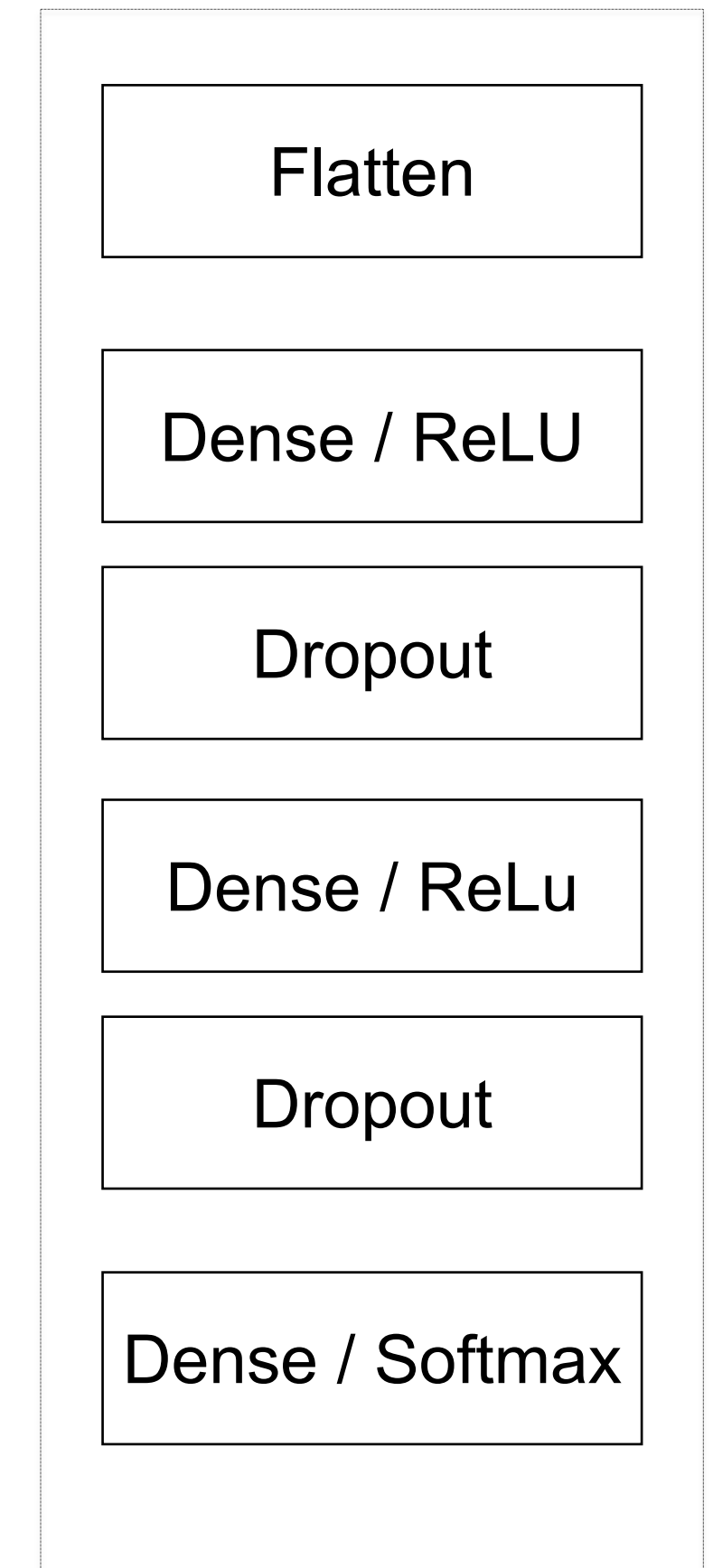
Le Transfer Learning

- ❑ Le Deep Learning est gourmand en données et temps de calcul.
- ❑ Une solution est de réutiliser un modèle existant et l'adapter à notre tâche
- ❑ Dans notre cas, on va utiliser les premières couches pour extraire les features de nos images
- ❑ Nous allons remplacer la dernière couche (Fully Connected) par notre propre modèle que nous allons entraîner sur nos données.



Entrainement de nos modèles

- ❑ Nous avons utilisé Keras une bibliothèque très intuitive de Deep Learning en python
- ❑ Nous nous sommes basés sur les réseaux pré-entraînés :
 - **VGG16, ResNet50 et Inception_V3.**
- ❑ Utilisation de Google Colaboratory et GPU
- ❑ Création d'un nouveau modèle pour la dernière couche et la classification
- ❑ Prédiction des 4 races de chiens
- ❑ Séparation des données en jeu d'entraînement, validation et test
- ❑ Tuning des hyper-paramètres avec recherche sur grilles



Résultats

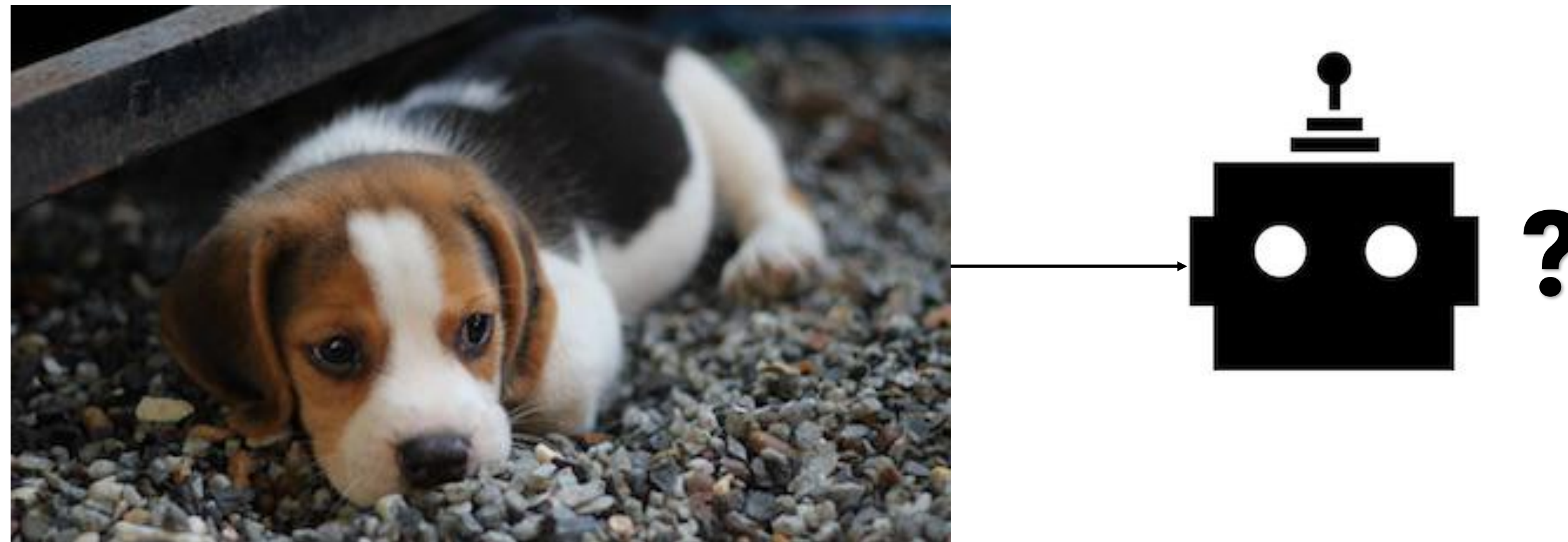
	VGG-16	ResNet50	Inception_V3
accuracy	96,33%	98,87%	99,15%
f1-score	96%	98%	99%

- ❑ Les résultats sur les données de tests sont très bons pour les 3 modèles
- ❑ Calculs particulièrement rapides sur Google Colaboratory
- ❑ Inception_V3 donne le meilleur score
- ❑ Meilleurs hyper-paramètres pour Inception_V3 :
 - dropout_rate': 0.5
 - 'optimizer': 'Ada

Implémentation

Outil de prédiction

- Mise en place d'un programme de prédiction en Python
- Il se base sur notre modèle basé sur le CNN Inception_V3
- Il prend en entrée une image de chien et retourne la prédiction de la race.
- Pourra être utilisé par l'association pour faciliter le travail d'indexation des photos



Conclusion

Conclusion

- Le projet a permis de tester 2 approches de catégorisation d'image : Bag of Visual Words et CNN
- La méthode basée sur les réseaux de neurones a donné des résultats impressionnants et elle est plutôt simple à mettre en œuvre
- Le traitement des images est assez gourmands en ressources et Google Colaboratory a facilité le projet
- Axes d'amélioration :
 - Mettre en place une solution sur l'ensemble des 120 races
 - Faire des tests supplémentaires de prétraitement des images et voir l'impact au niveau des performances
 - Techniques d'optimisation du CNN : couche de normalisation, ajout de données augmentés, ...
 - Comparer avec Google Cloud AutoML : construction de modèle de reconnaissance d'images

Merci à mon mentor Amine Abdaoui pour sa disponibilité, ses explications et ses précieux conseils