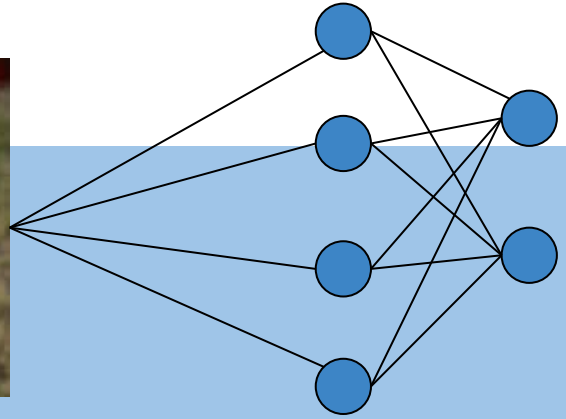
Classez des images à l'aide d'algorithmes de Deep Learning

par Ana Bernal

mentor: Samir Tanfous

Avril 2023



Repo github pour ce projet:

github.com/ana-bernal/P6_img-classif

Programme

1. Rappel mission
2. Exploration et aperçu des données
3. Tests pour classification: *from scratch* et *transfer learning*
   Meilleur modèle + inférence
4. Conclusions

1

Rappel mission

Rappel mission

Association de protection d'animaux

base des données grandit

Mission
algorithme pour automatiquement:

chihuahua

2

Exploration des données

Aperçu des données

Source des données

Stanford Dogs Dataset

Aditya Khosla Nityananda Jayadevaprakash Bangpeng Yao Li Fei-Fei

Stanford University

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. Contents of this dataset:

Quelques statistiques

Nombre total d'images	20581
Nombre de classes (races de chien)	120

```
Sample of 10 breed names:
['Chihuahua',
 'Japanese_spaniel',
 'Maltese_dog',
 'Pekinese',
 'Shih',
 'Blenheim_spaniel',
 'papillon',
 'toy_terrier',
 'Rhodesian_ridgeback',
 'Afghan_hound']
```

Aperçu des données

Source des données

Stanford Dogs Dataset

Aditya Khosla Nityananda Jayadevaprakash Bangpeng Yao Li Fei-Fei

Stanford University

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. Contents of this dataset:

Quelques statistiques

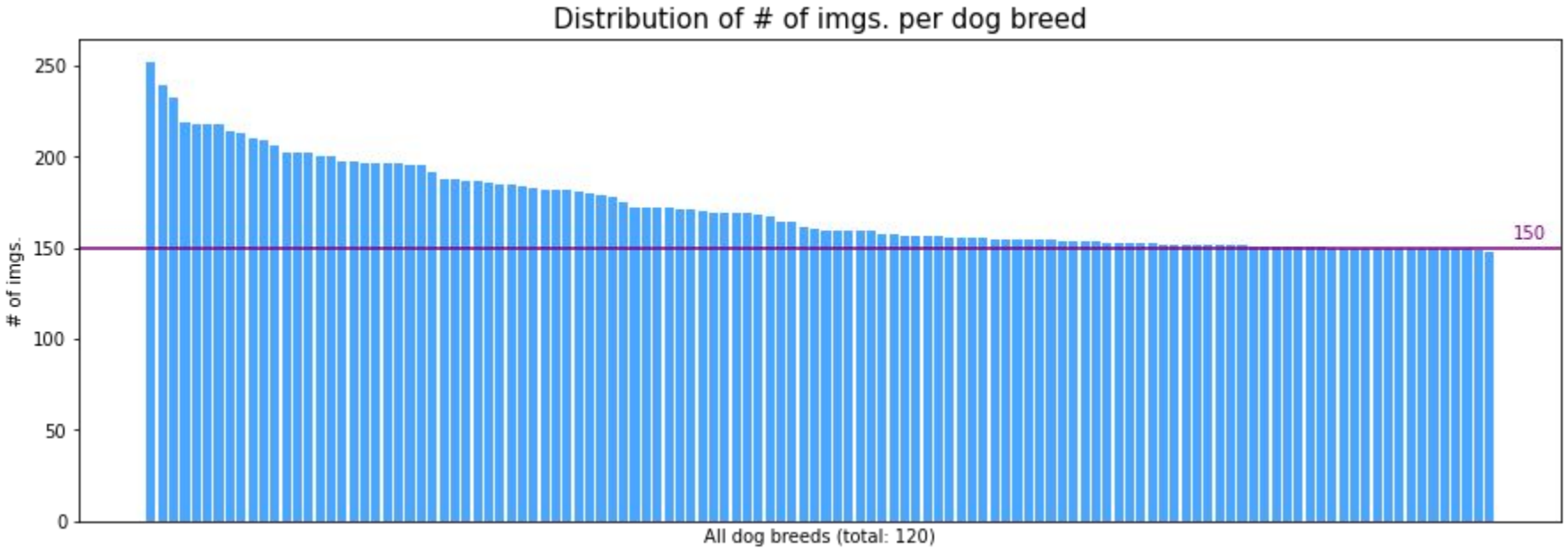
Nombre total d'images	20581
Nombre de classes (races de chien)	120

Pour classification après	
Nombre total d'images	2537 = 2030 + 507
Nombre de classes (races de chien)	15

```
Sample of 10 breed names:
['Chihuahua',
 'Japanese_spaniel',
 'Maltese_dog',
 'Pekinese',
 'Shih',
 'Blenheim_spaniel',
 'papillon',
 'toy_terrier',
 'Rhodesian_ridgeback',
 'Afghan_hound']
```

Aperçu des données

Distribution du # d'images par race de chien



Aperçu des données

Aperçu de quelques images et leurs labels

3

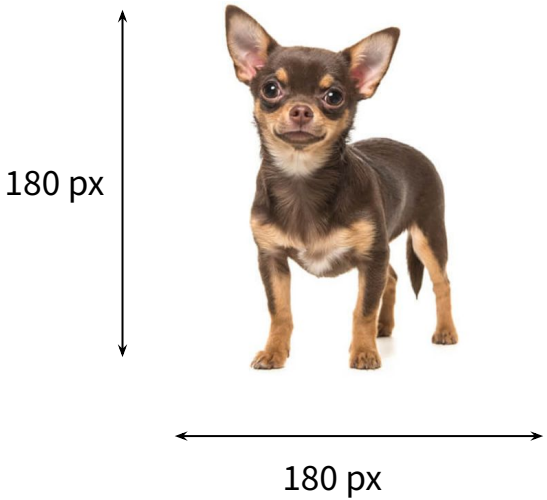
Tests de classification

Classification

- Classification d'images ➜ label.
- Un seul label possible.
- Réseaux des neurones

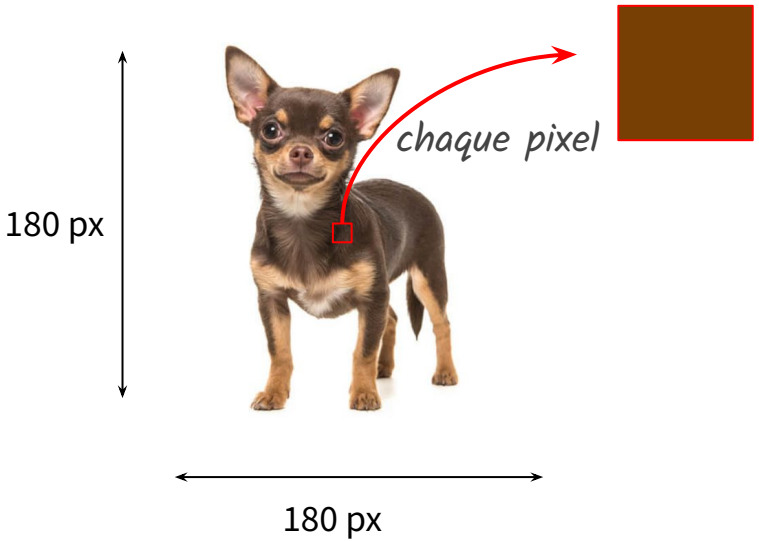
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



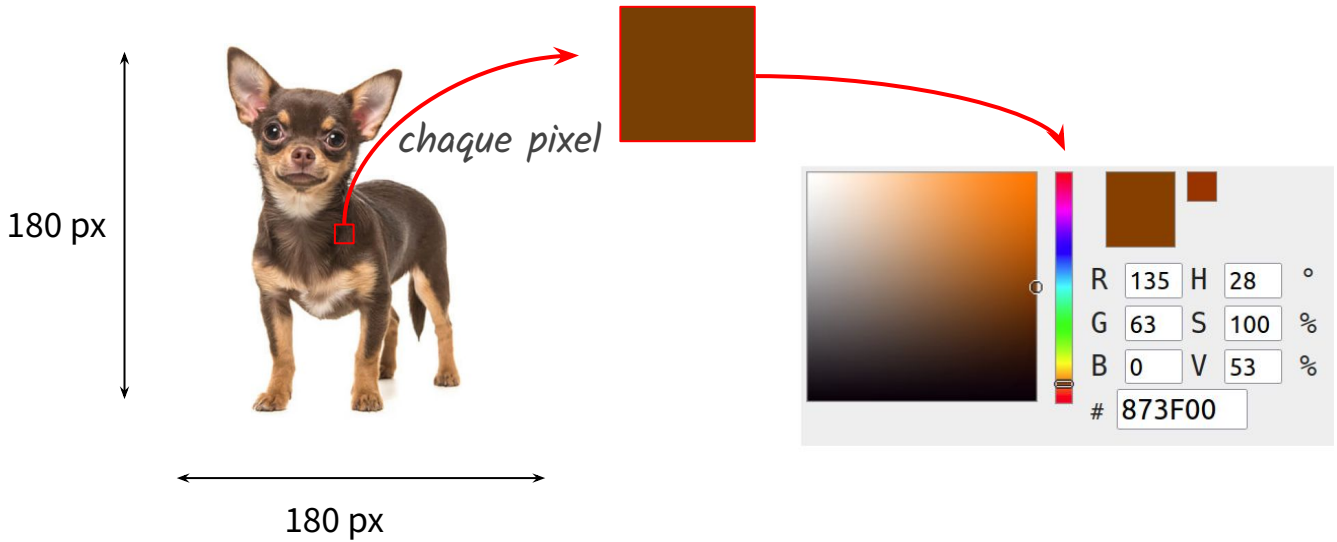
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



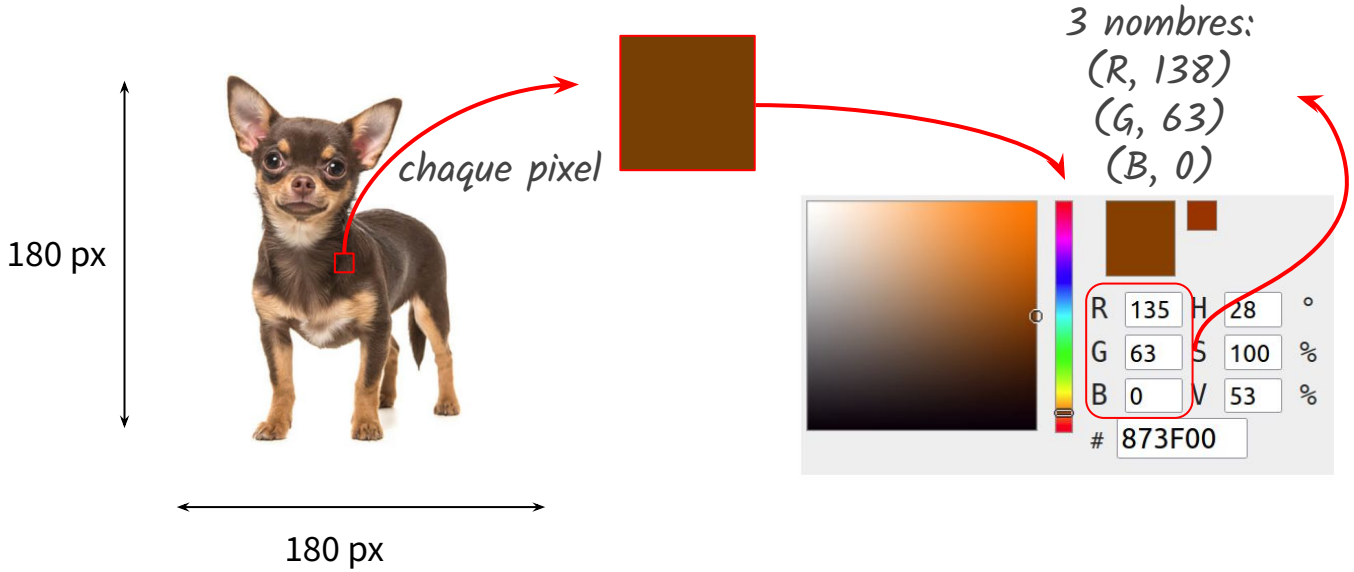
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



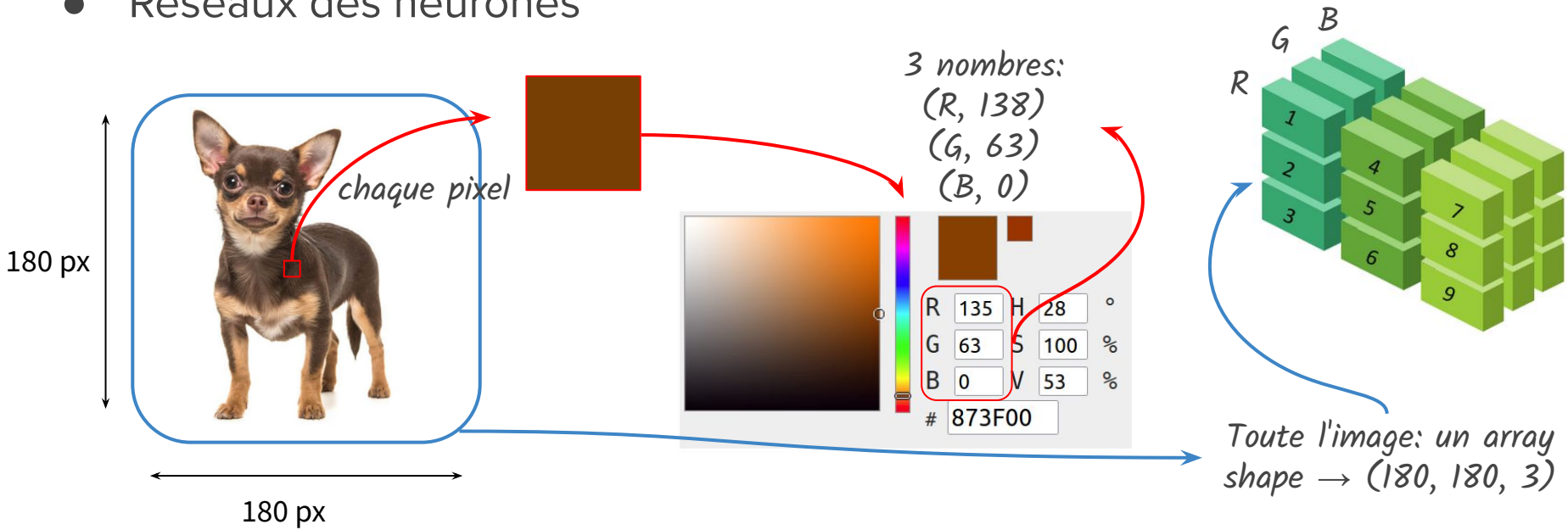
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



Classification

- Classification d'images → label
- Un seul label possible
- Réseaux des neurones

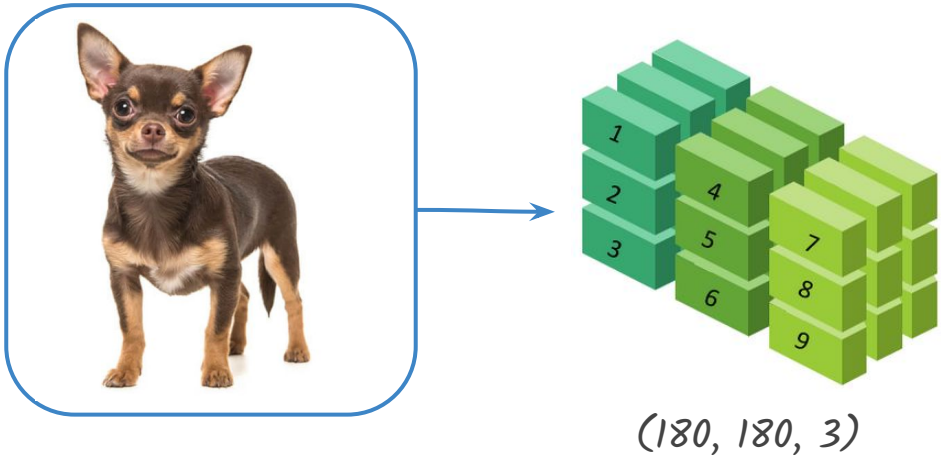


Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones

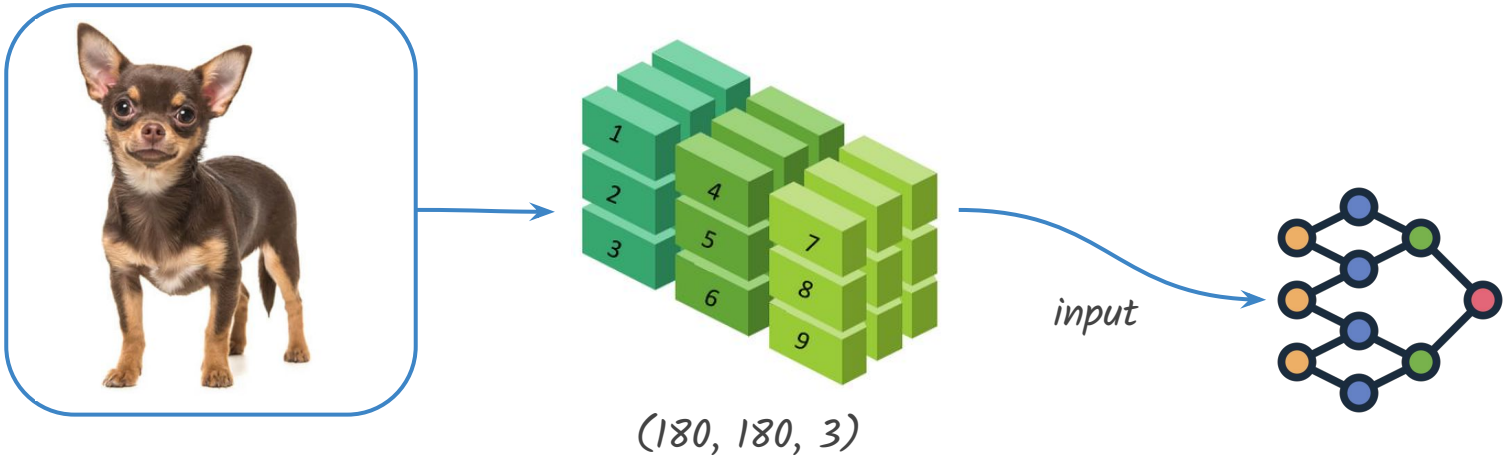
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



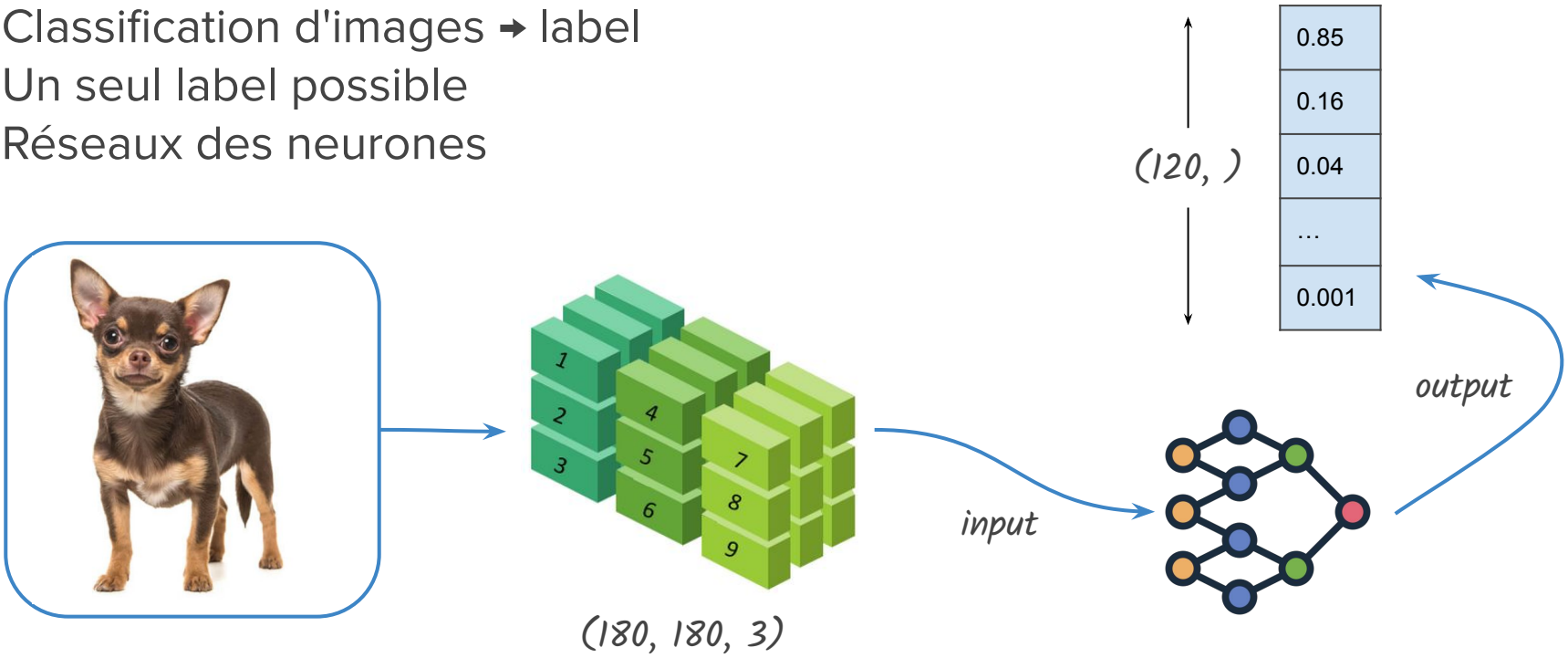
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



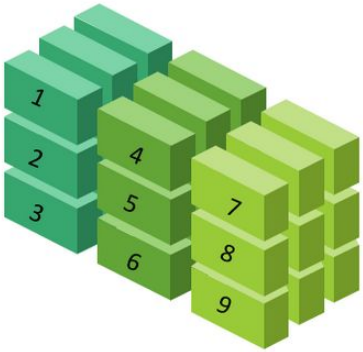
Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



Classification

- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



(180, 180, 3)

input

output

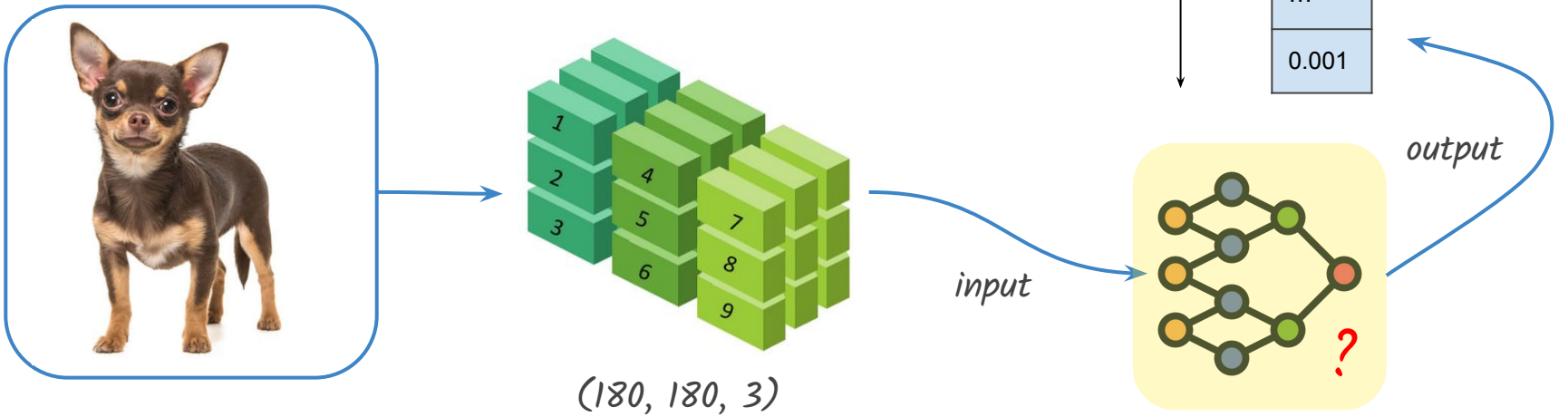
probabilités pour chaque classe

(120,)

0.85	chihuahua
0.16	doberman
0.04	beagle
…	
0.001	

Classification

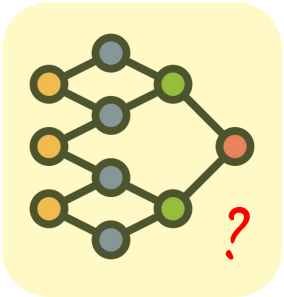
- Classification d'images ➜ label
- Un seul label possible
- Réseaux des neurones



Classification

Convolutional Neural Network

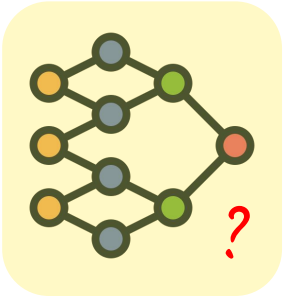
Parenthèse:
couche convolution

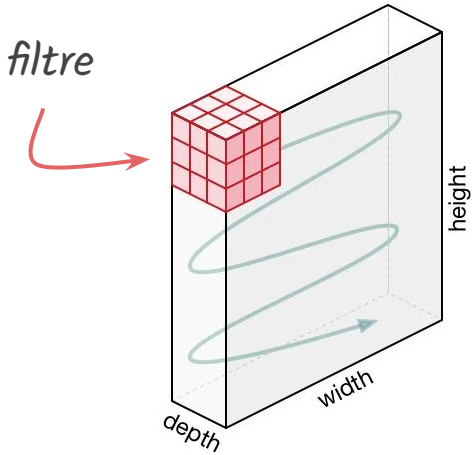


Classification

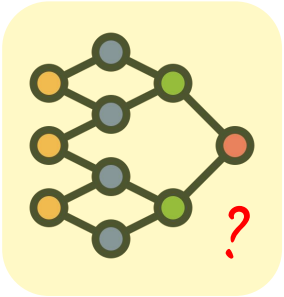
Convolutional Neural Network

Parenthèse:
couche convolution



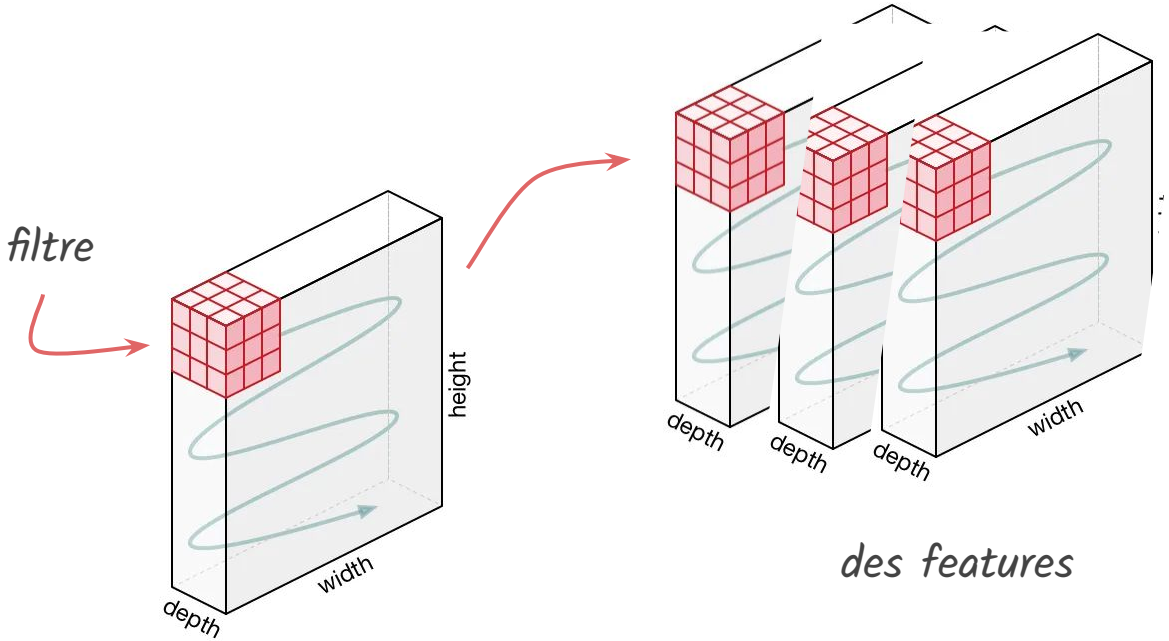


Classification



Convolutional Neural Network

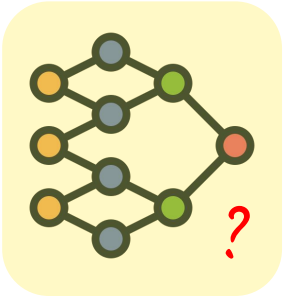
Parenthèse:
couche convolution



Classification

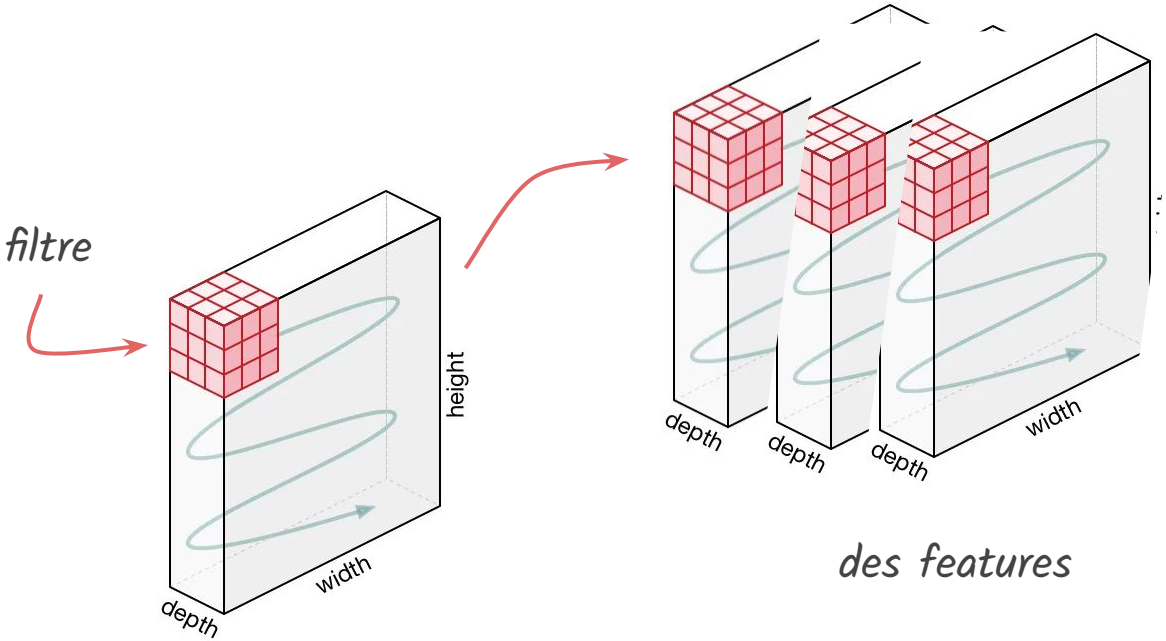
Convolutional Neural Network

Parenthèse:
couche convolution



Beaucoup d'hyper paramètres

filtres, taille filtres
stride
couches conv.
...



Classification

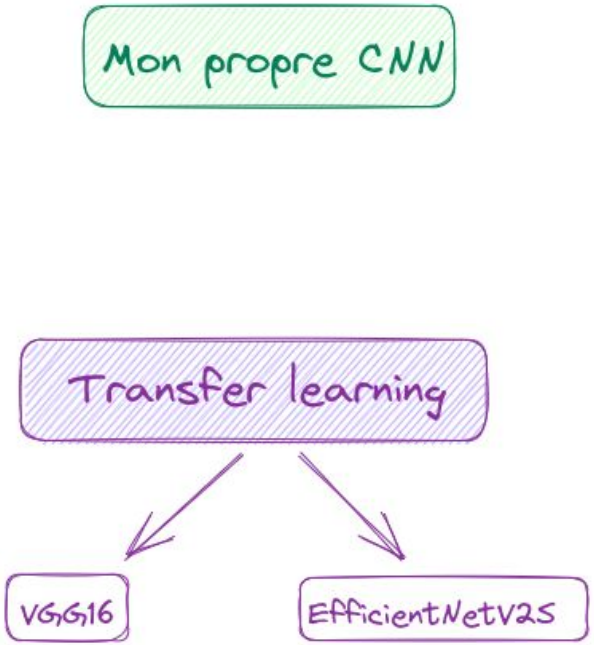
Démarche

Classification

Démarche

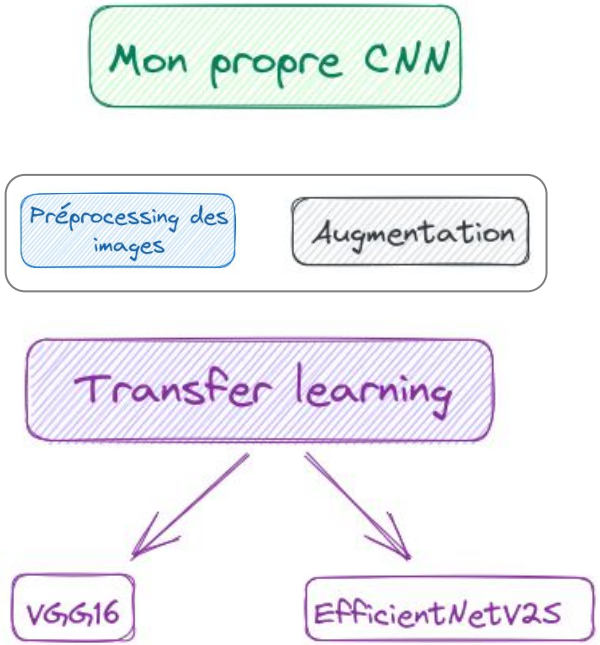
Classification

Démarche



Classification

Démarche



Classification

Classification

VGG16

The images are converted from RGB to BGR, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling.

EfficientNetV2S

Included in the model

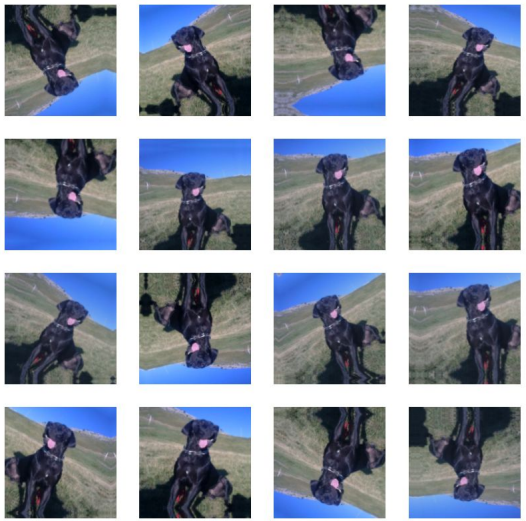
Classification

Préprocessing des images

Augmentation

VGG16

The images are converted from RGB to BGR, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling.

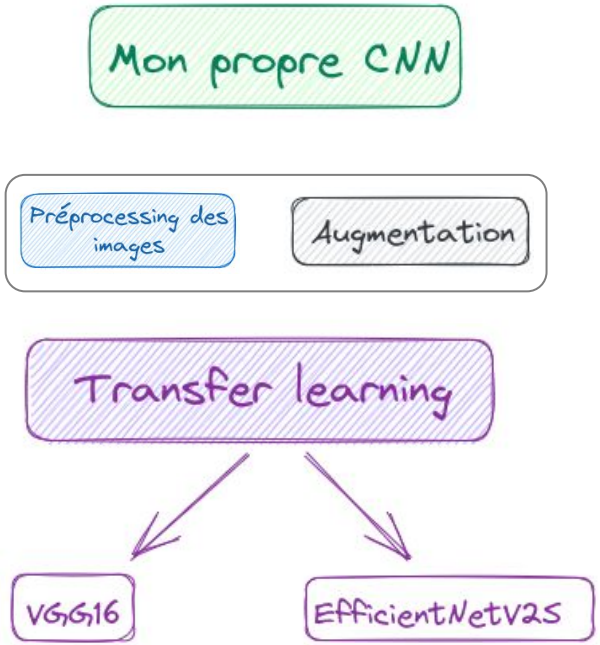


EfficientNetV2S

Included in the model

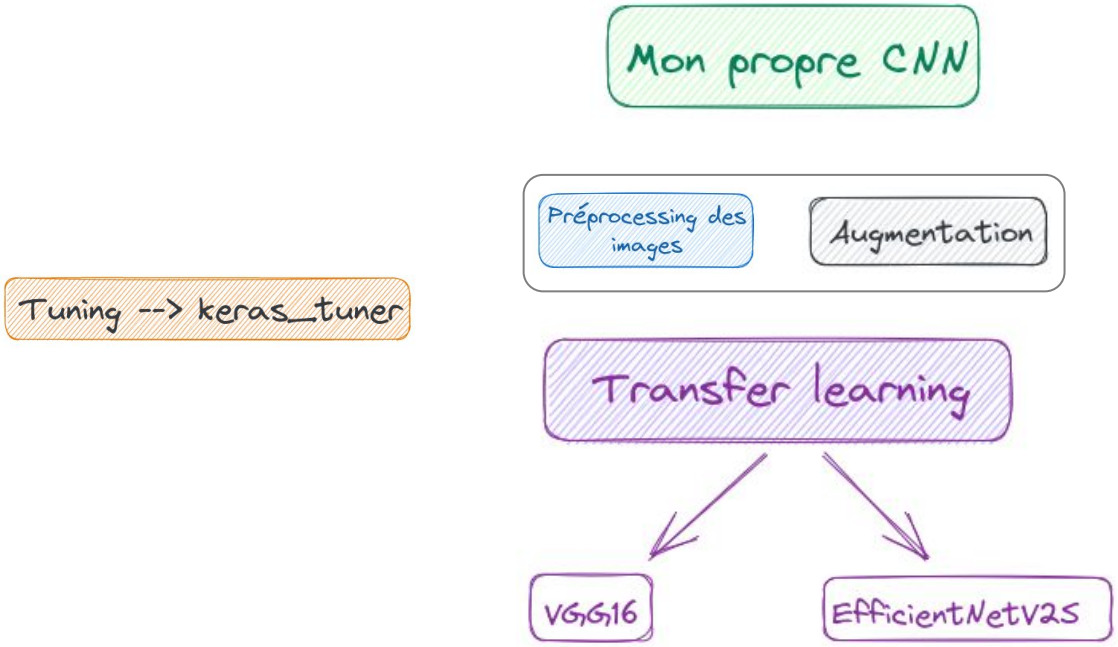
Classification

Démarche



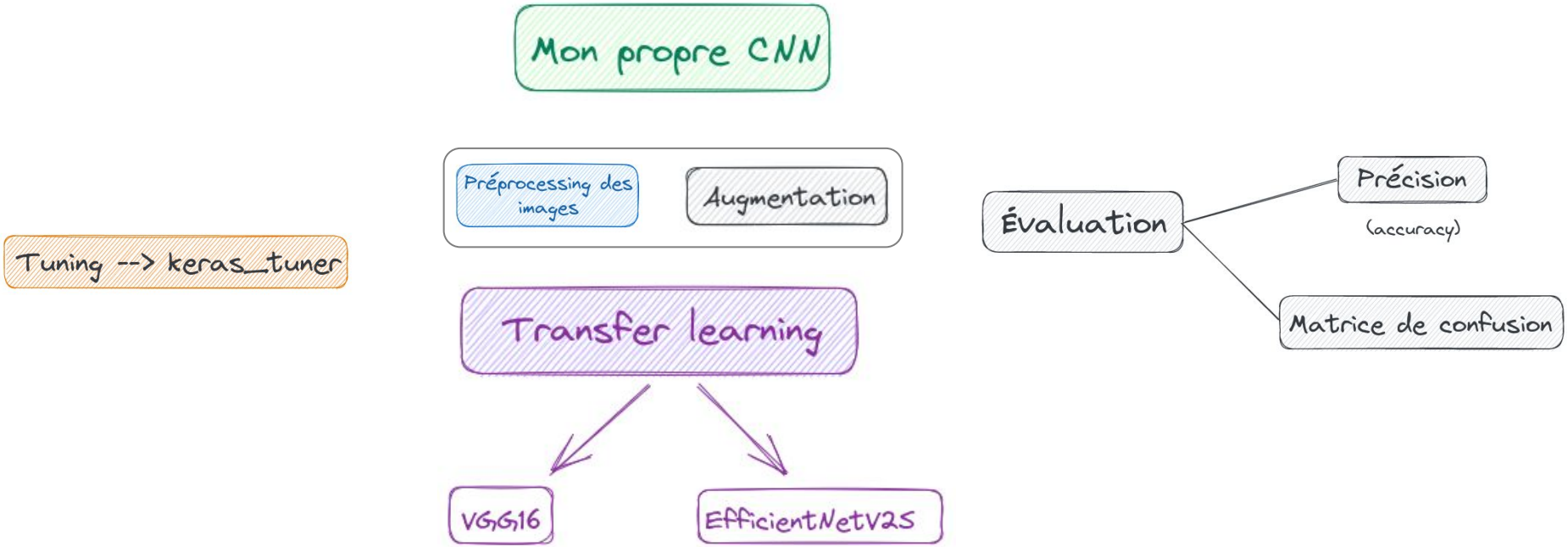
Classification

Démarche



Classification

Démarche



Classification

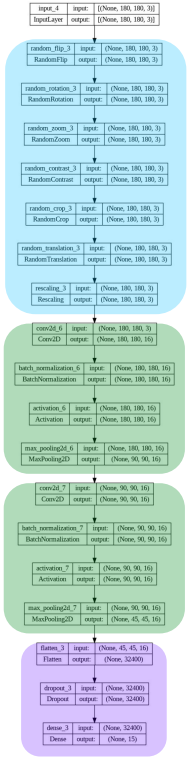
Architectures

Classification

Classification

Mon propre CNN → S / L

CNN - S



augmentation

Bloc convolution:
Conv2D
batch norm
ReLU
MaxPooling

bloc convolution

Bloc classification:
flatten
dropout
fully connected + softmax

CNN - L

augmentation

bloc convolution

bloc convolution

bloc convolution

bloc convolution

bloc convolution

Bloc classification:
flatten
fully connected
dropout
fully connected
dropout
fully connected + softmax

Classification

Mon propre CNN

- S
- L

CNN - S

params = 491 167

augmentation

Bloc convolution:
Conv2D
batch norm
ReLU
MaxPooling, strides = (2,2)

K = 16
F = 10

bloc convolution

K = 16
F = 1

Bloc classification:
flatten
dropout rate = 0.3
fully connected + softmax

CNN - L

params = 1 270 143

augmentation

K = 64
F = 10

bloc convolution

K = 64
F = 5

bloc convolution

K = 32
F = 5

bloc convolution

K = 32
F = 10

bloc convolution

K = 16
F = 10

bloc convolution

Bloc classification:
flatten
fully connected, u = 1023
dropout = 0.2
fully connected, u = 512
dropout = 0.5
fully connected + softmax

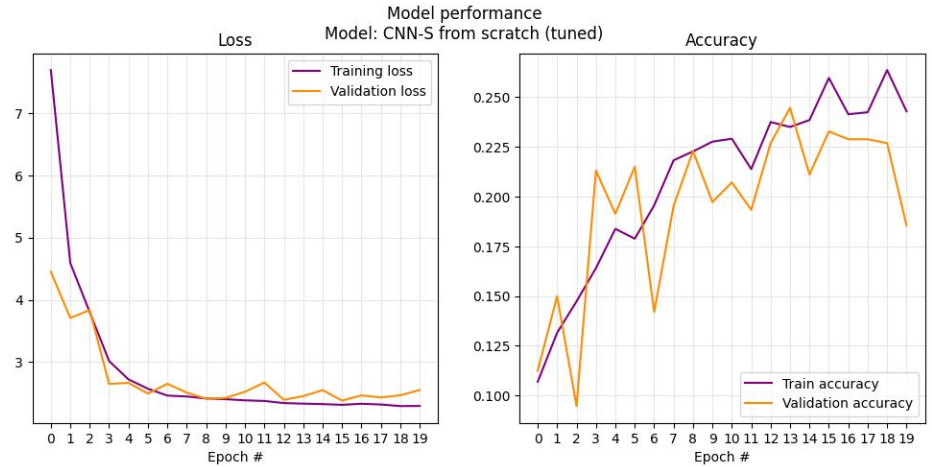
Classification

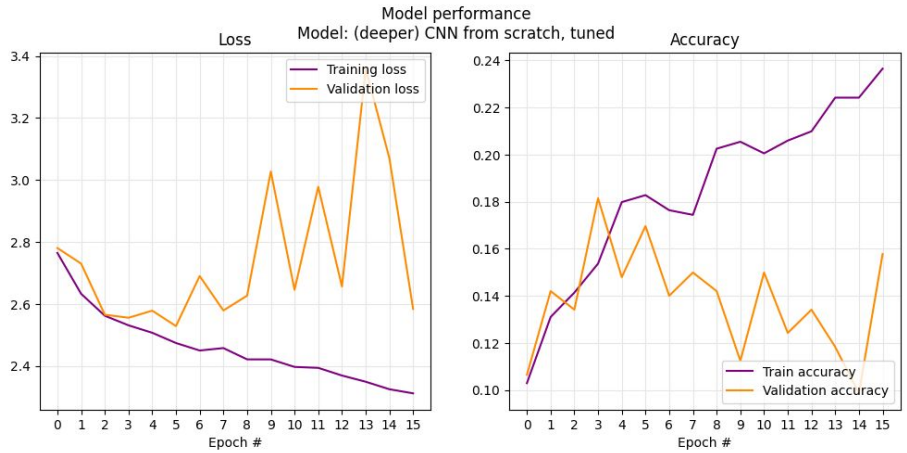
Mon propre CNN

- S
- L

CNN - S

CNN - L





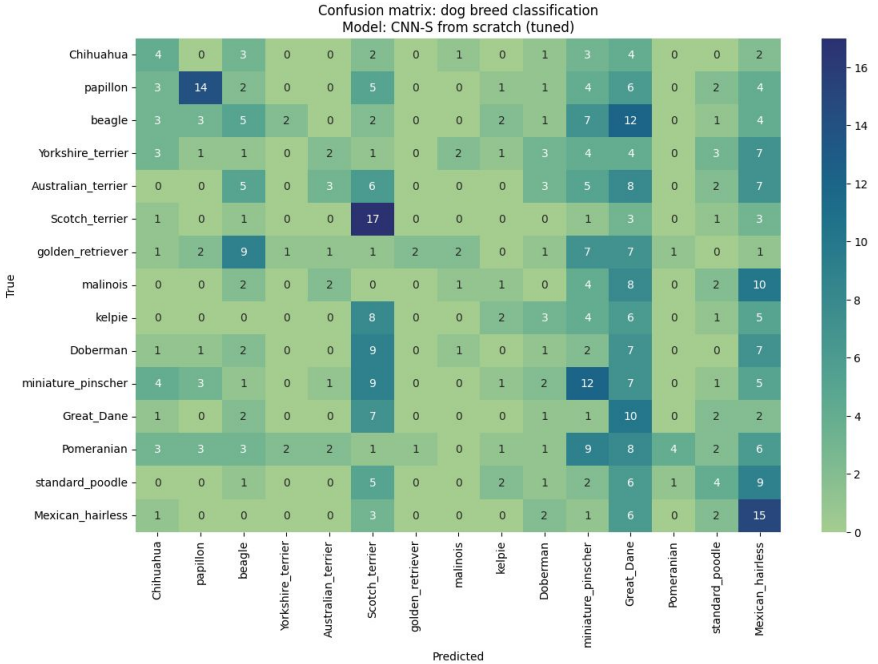
Classification

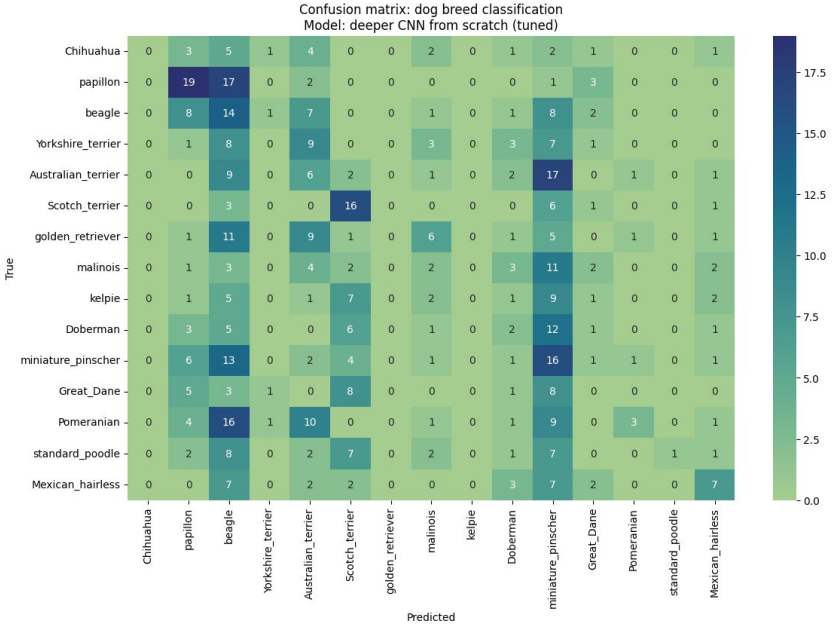
Mon propre CNN

- S
- L

CNN - S

CNN - L

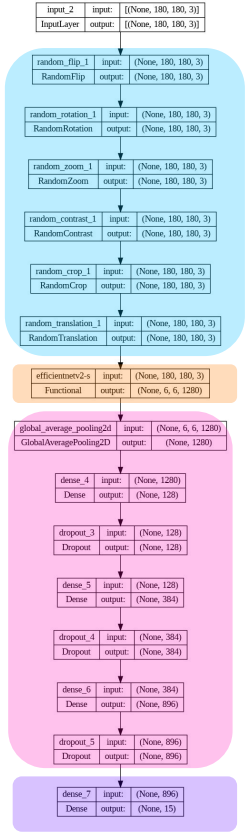




Classification

Transfer learning

Architecture



augmentation

VGG16 ou EfficientNetV2S

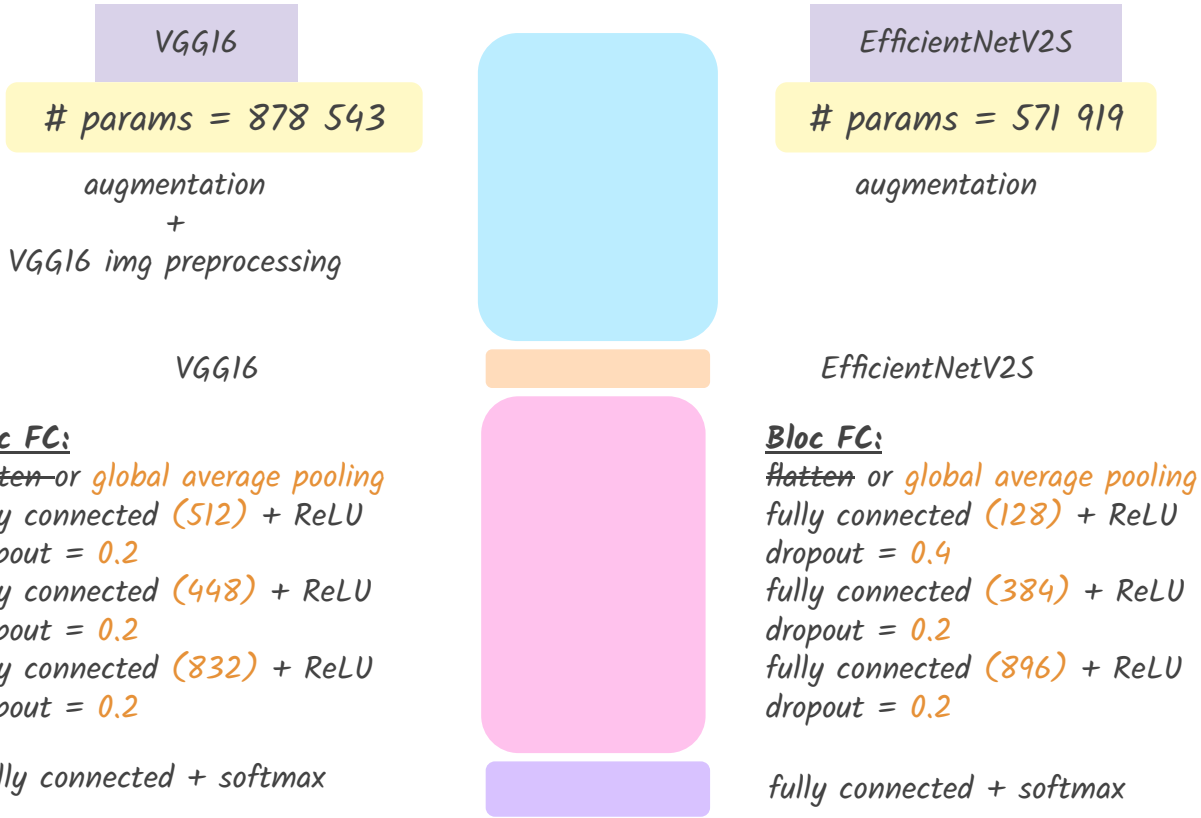
Bloc FC:

flatten or global average pooling
fully connected + ReLU
dropout
fully connected + ReLU
dropout
fully connected + ReLU
dropout

fully connected + softmax

Classification

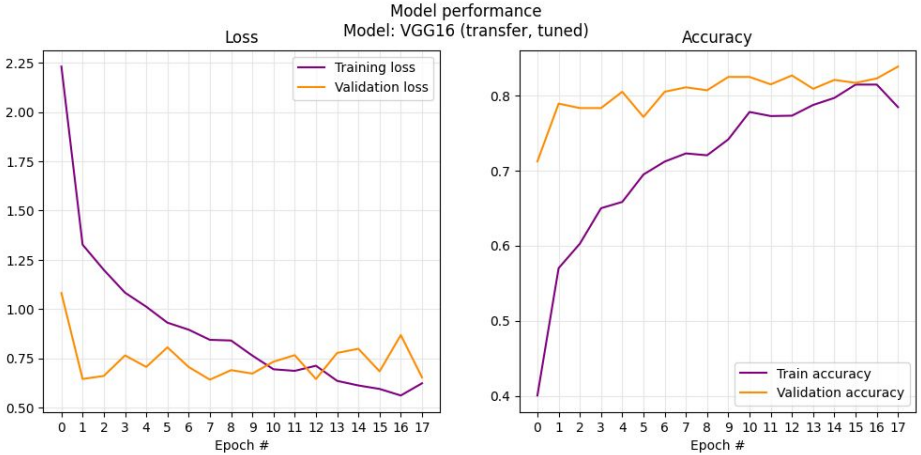
Transfer learning



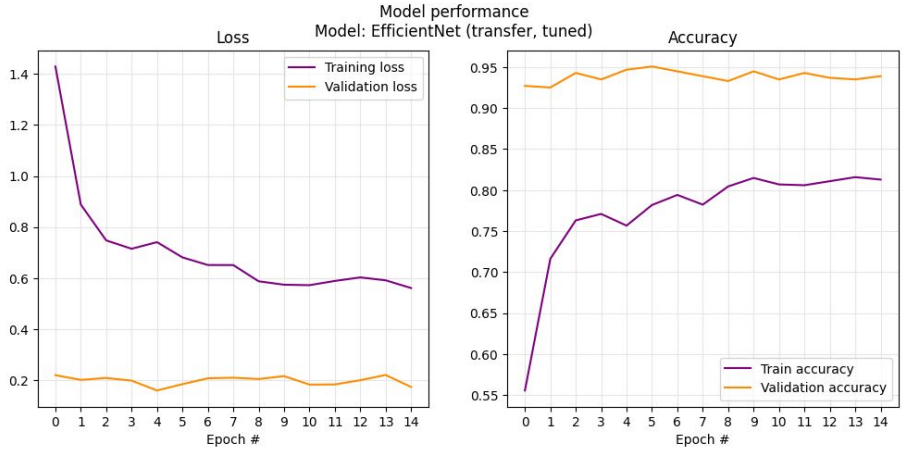
Classification

Transfer learning

VGG16



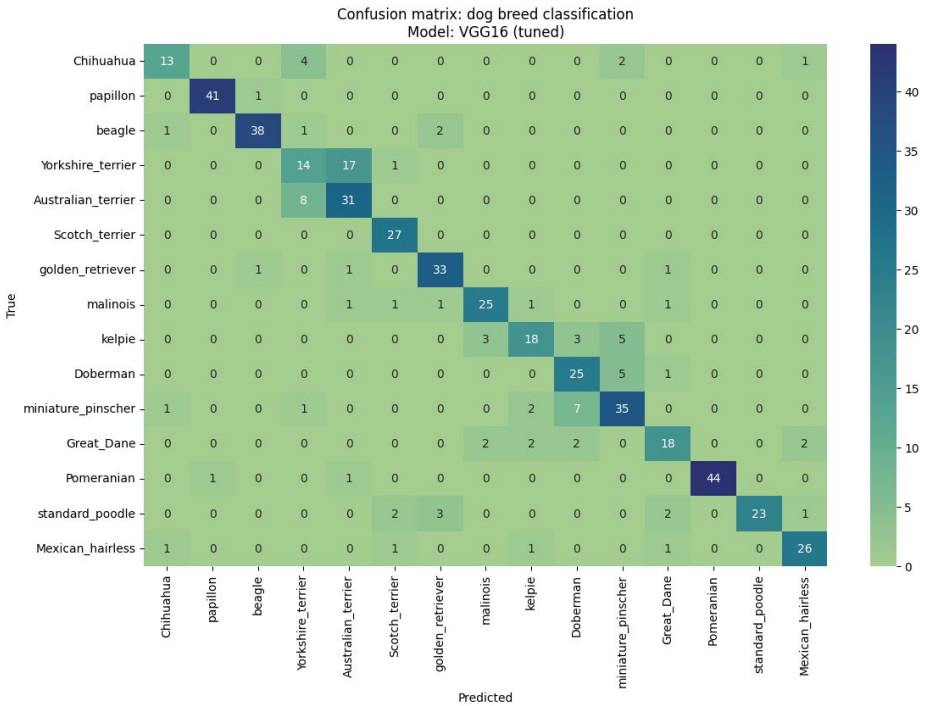
EfficientNetV2S



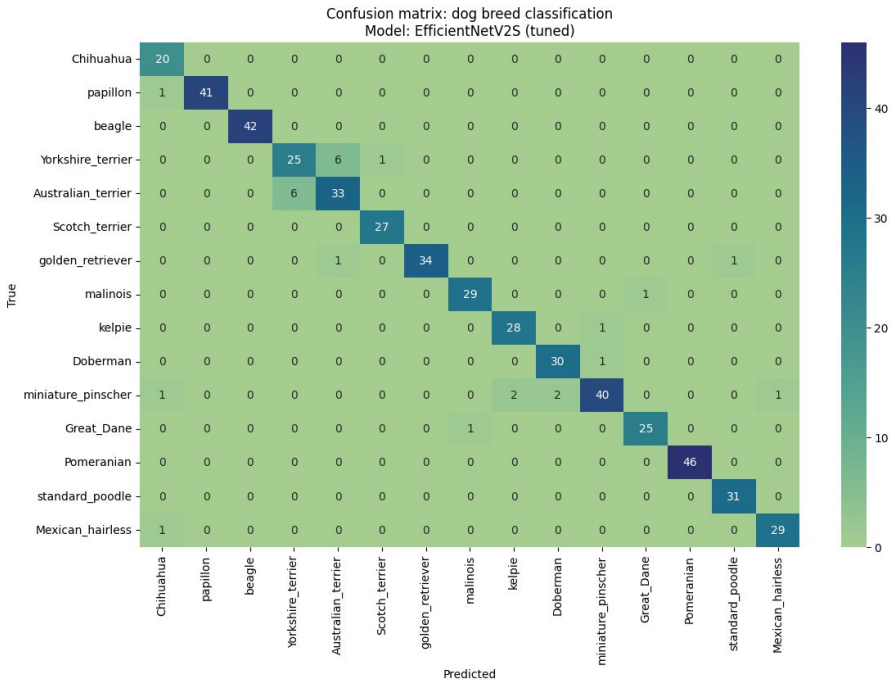
Classification

Transfer learning

VGG16



EfficientNetV2S



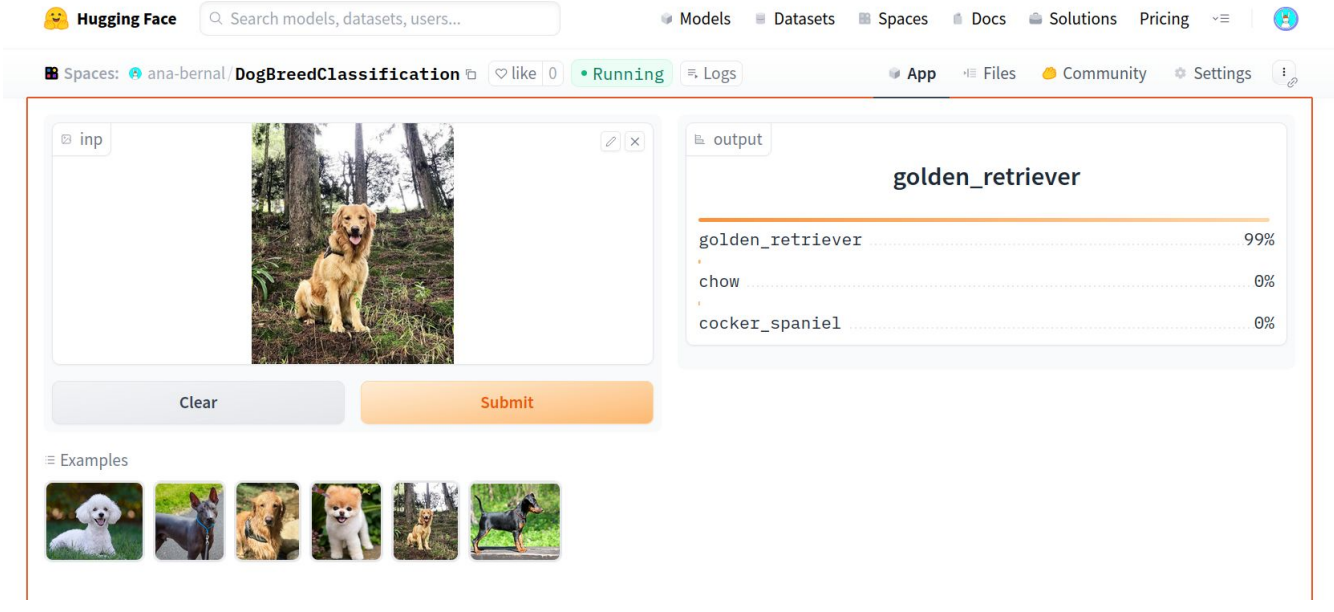
Classification

Modèle choisi

Transfer learning + EfficientNetV2S

Classification: inférence

Clic ici * *pour tester ce modèle en action*



* huggingface.co/spaces/ana-bernal/DogBreedClassification

4

Conclusions

Conclusions

- EfficientNetV2 ➜ les meilleurs performences sur tous les modèles.
- ImageNet base des données/labels assez efficace déjà pour classification.
- Modèles à disposition aussi très efficaces déjà.

Aller + loin

- 3 datasets: train, test, validation.
- Entraîner tous les paramètres de VGG16 et EfficientNet, ou plus de couches ➜ mieux comprendre les performances avec nos données.
- Appliquer les modèles tels quels avec une autre base des données.
- Forcer le # et tailles des filtres pour modèles from scratch et tuner autres paramètres.
- + d'Epochs.

Merci !