

# CLASSIFICATION D'IMAGES DE CHIENS VIA DES MÉTHODES DE DEEP LEARNING

Cécile Guillot, Ingénieur en  
Machine Learning

# sommaire

1

contexte

2

1er essais

3

généralisation

4

conclusion

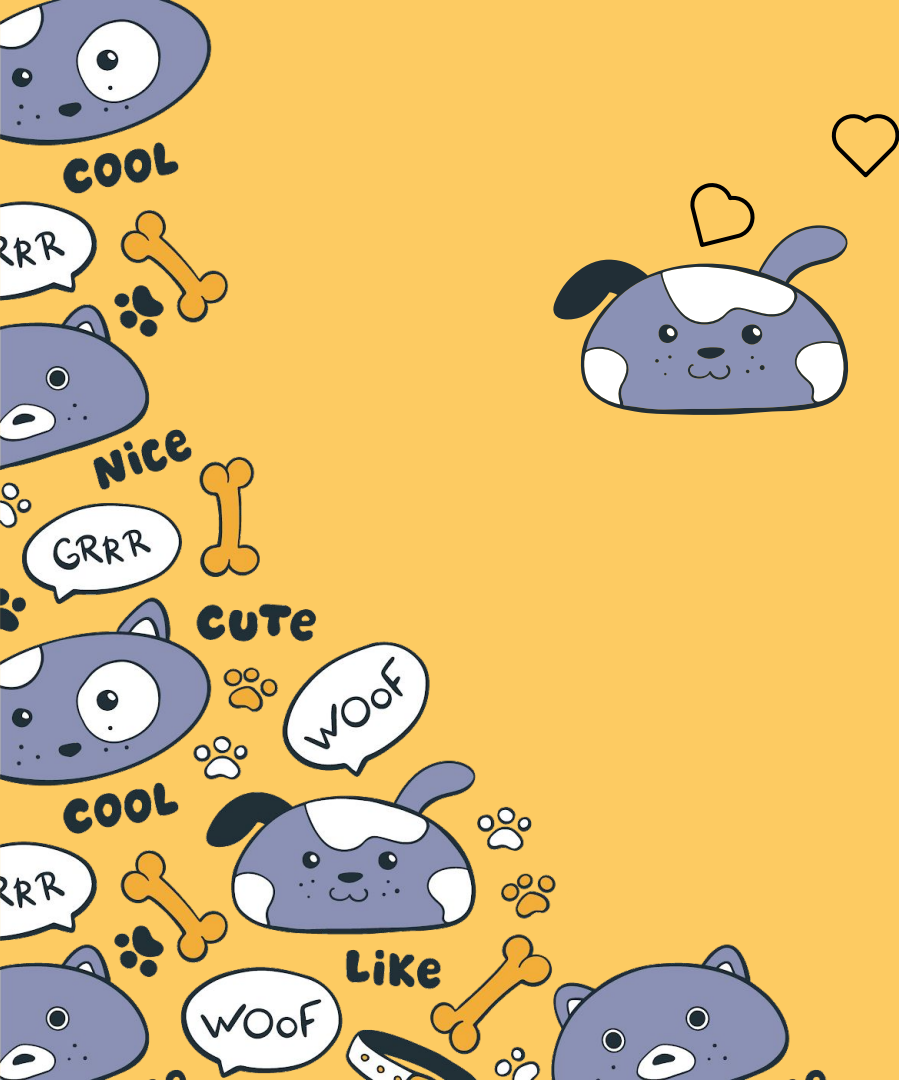




1



INTRO



# Le contexte

Agrandissement de la base  
de données et difficulté à  
référencer les nouveaux  
pensionnaires



# PROBLÉMATIQUE

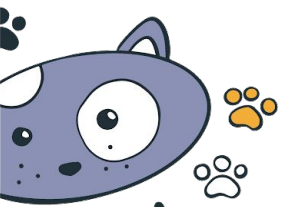
Comment automatiser la  
reconnaissance des  
pensionnaires de  
l'association grâce au Deep  
Learning ?





# GRRR

- Entraîner un modèle pour pouvoir identifier la race d'appartenance d'un chien mis à l'adoption
- Données de départ : Stanford Dogs dataset  
<http://vision.stanford.edu/aditya86/ImageNetDogs/>



# 2

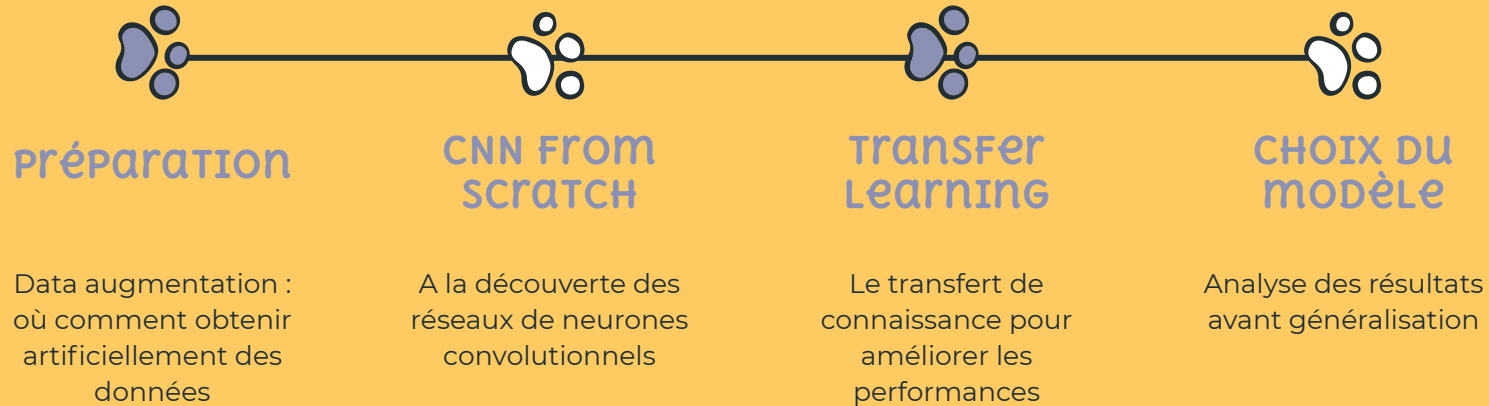


## Premiers essais

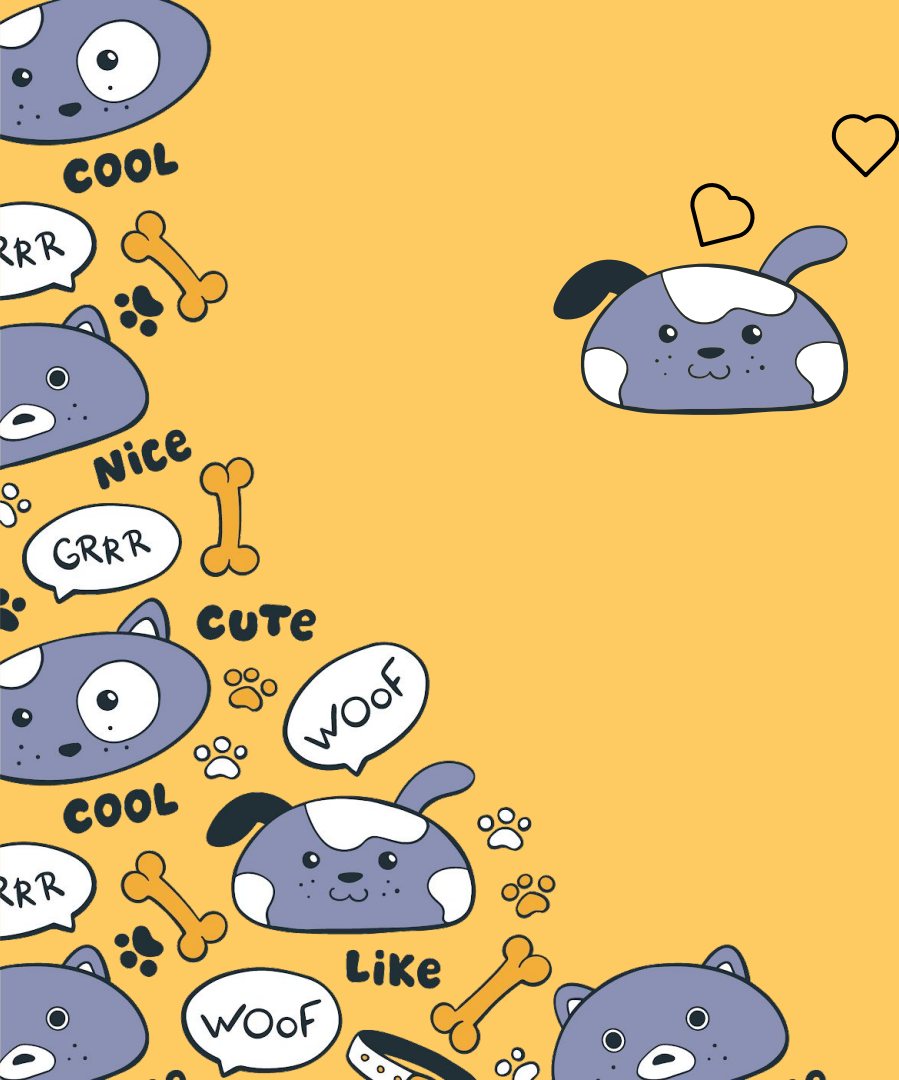
D'un CNN from scratch au  
transfer learning



# ETAPE par étape







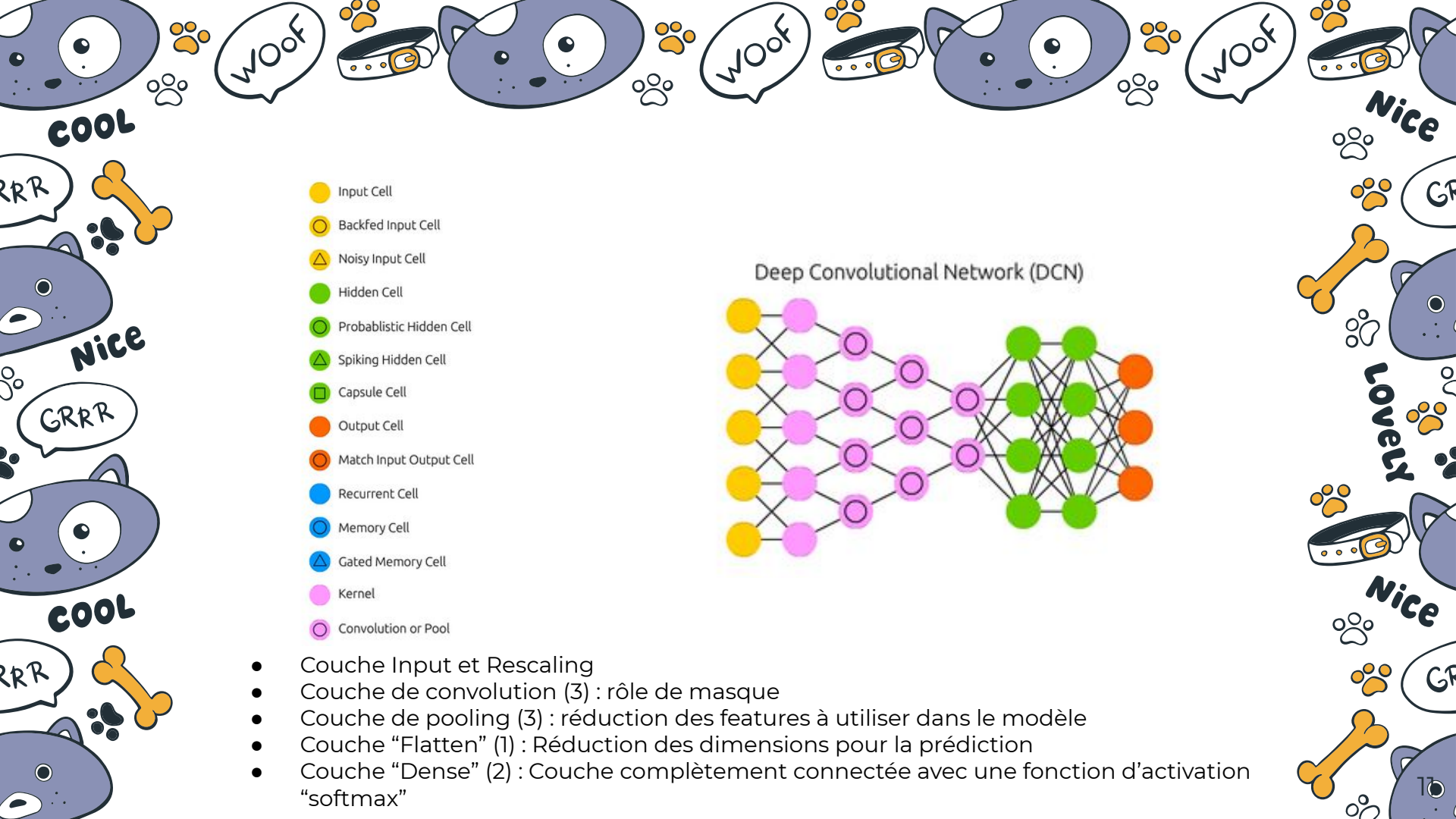
## DATA AUGMENTATION

- Génération (artificielle) d'un nombre plus importants d'images pour l'apprentissage
- Gestion des cas où l'image n'a pas été uploadé dans le bon sens



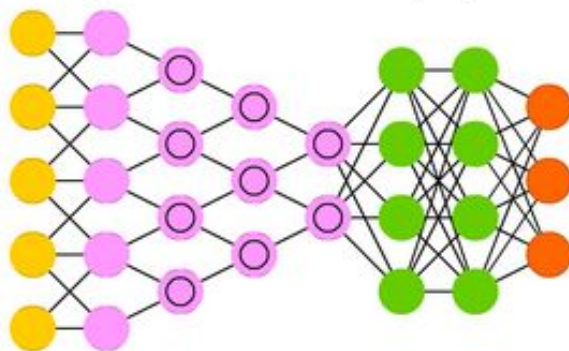
Exemple de data augmentation





- Input Cell
- Backfed Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Gated Memory Cell
- Kernel
- Convolution or Pool

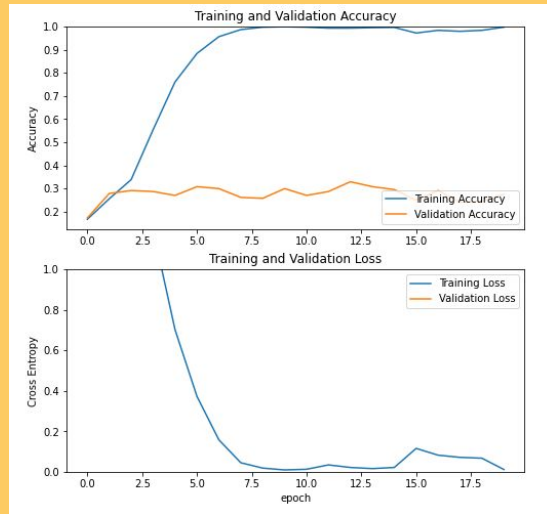
Deep Convolutional Network (DCN)



- Couche Input et Rescaling
- Couche de convolution (3) : rôle de masque
- Couche de pooling (3) : réduction des features à utiliser dans le modèle
- Couche "Flatten" (1) : Réduction des dimensions pour la prédiction
- Couche "Dense" (2) : Couche complètement connectée avec une fonction d'activation "softmax"

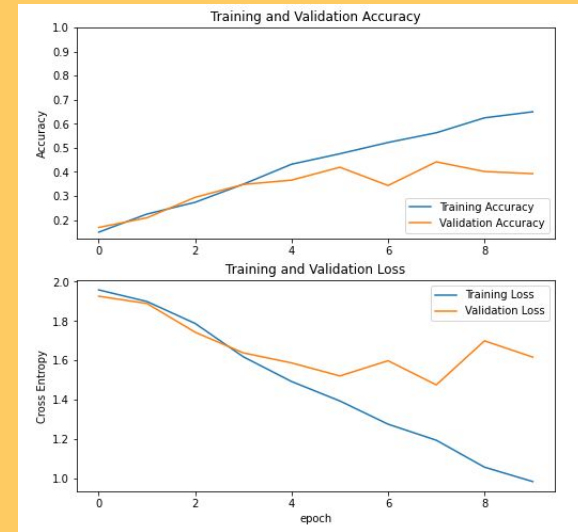
# convnet : RÉSULTATS

sans



53 sec.  
Précision : 33%

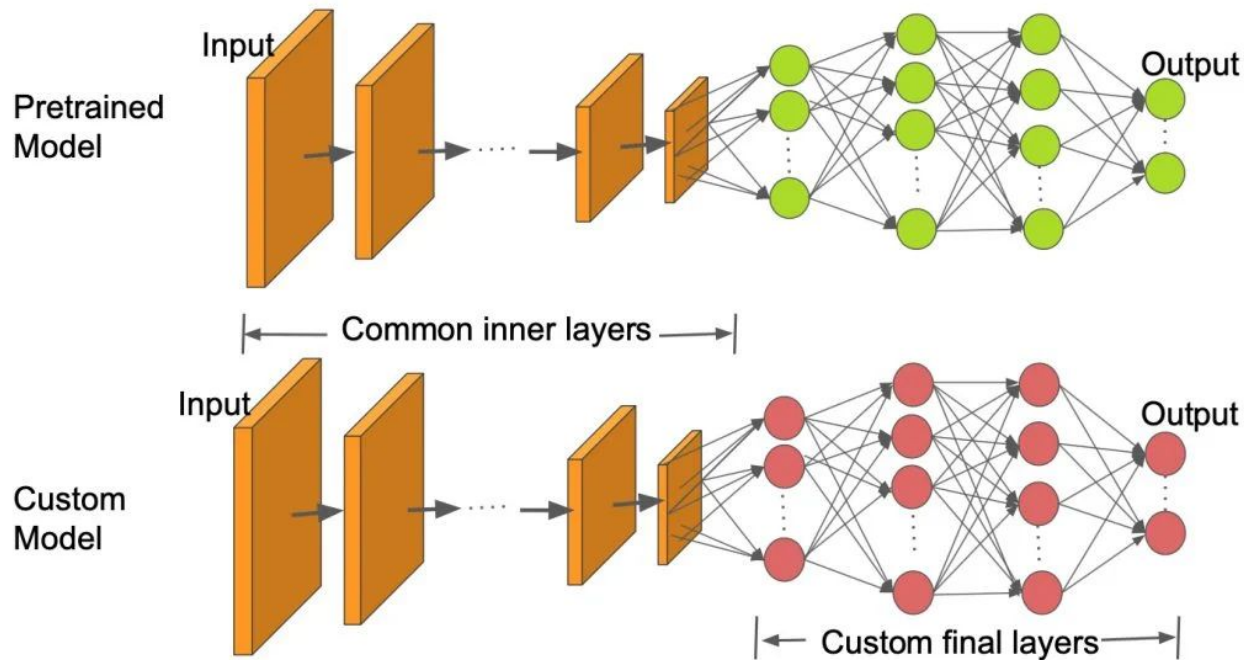
Avec



105 sec.  
Précision : 43%



# Transfer Learning : Théorie



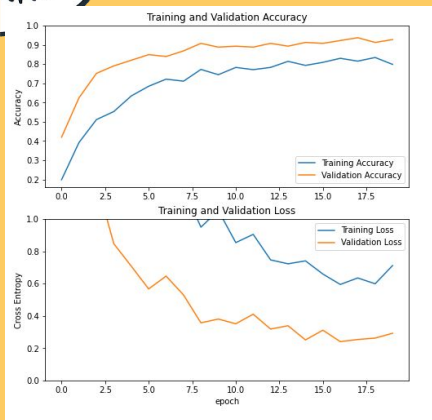




# Transfer Learning : Résultats

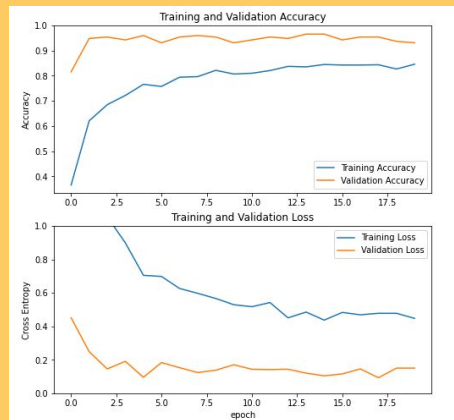


RRR



VGG19

76 sec.  
Précision : 94%

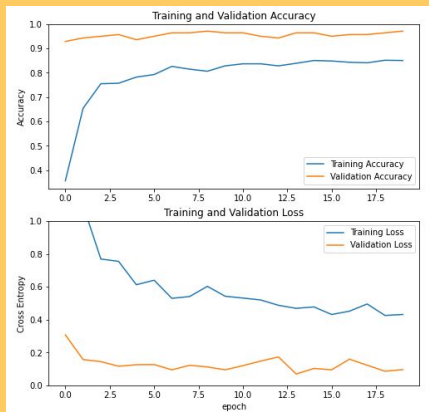


RESNET 50

57 sec.  
Précision : 90%

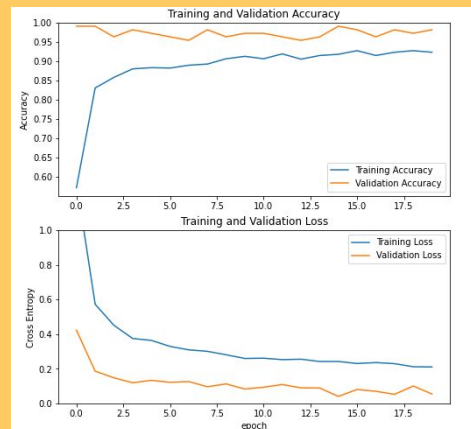
INCEPTION V3

54 sec.  
Précision : 94%



EFFICIENT NET  
B7

180 sec.  
Précision : 100%







## EN Bref...

- Amélioration des performances sur 7 races
- Temps d'entraînement légèrement supérieur à CNN from Scratch
- Peu d'overfitting/underfitting





3

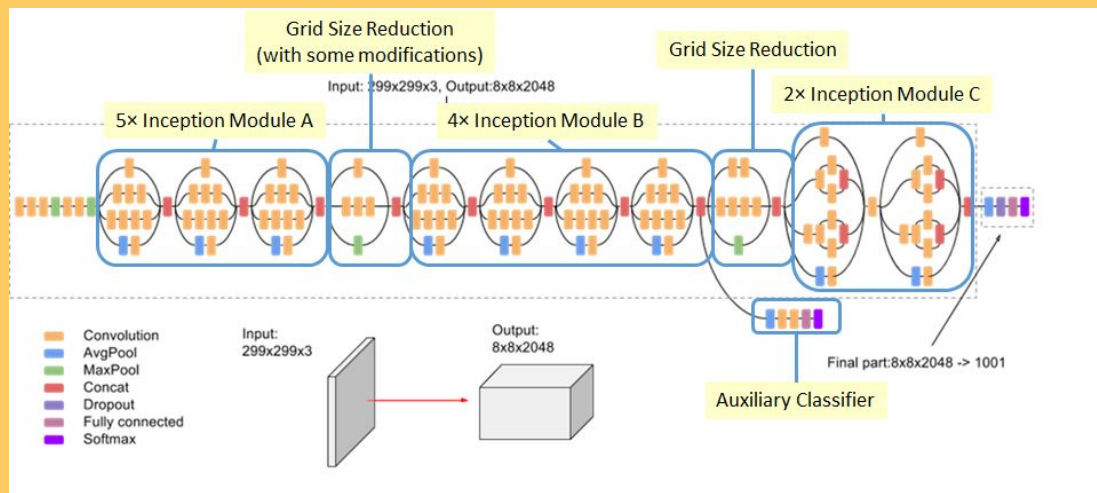


# Généralisation

Inception V3 sur 120 races

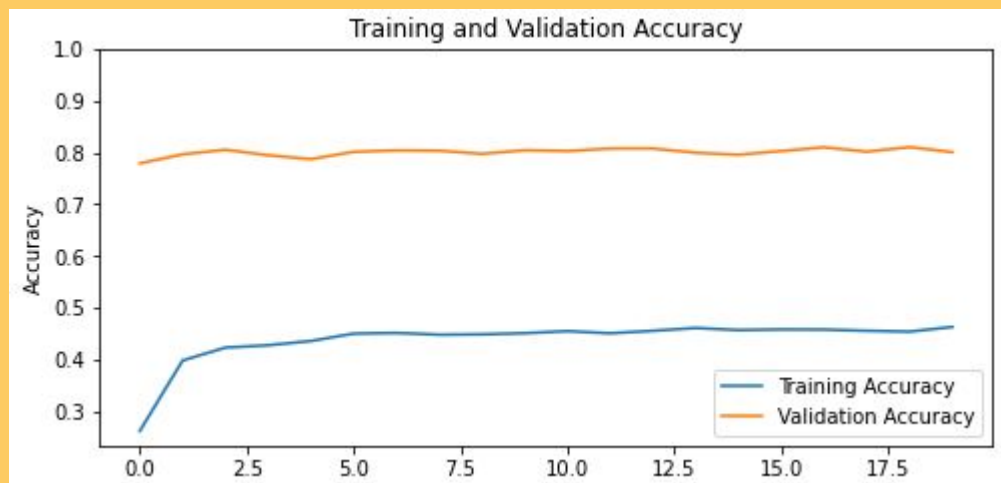
# INCEPTION V3 : ARCHITECTURE

- Développé en 2015.
- Environ 300 couches
- Meilleure rapport temps/performance
- Plus léger pour le déploiement (même si la version actuelle est de 86 Mo)



# RÉSULTATS après entraînement

- 720 sec. (soit environ 12 min.) + 753 sec. (soit environ 12 min.)
- Précision :
  - 80% sur le jeu de validation
  - 77% sur le jeu de test



# 4



## DÉPLOIEMENT

Mettre notre modèle dans  
un programme facile  
d'utilisation



# OUTILS UTILISÉS



TensorFlow

Modélisation



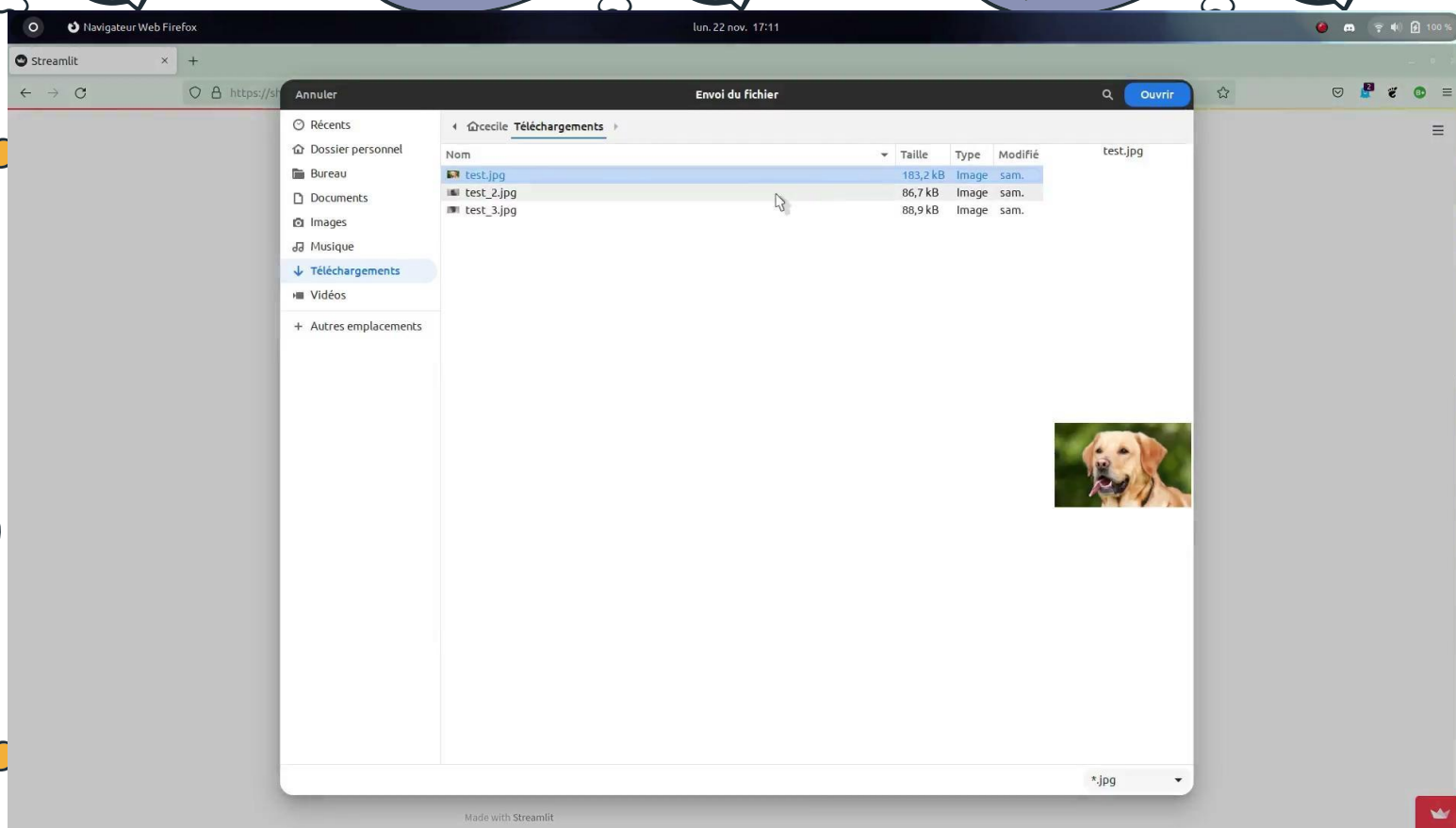
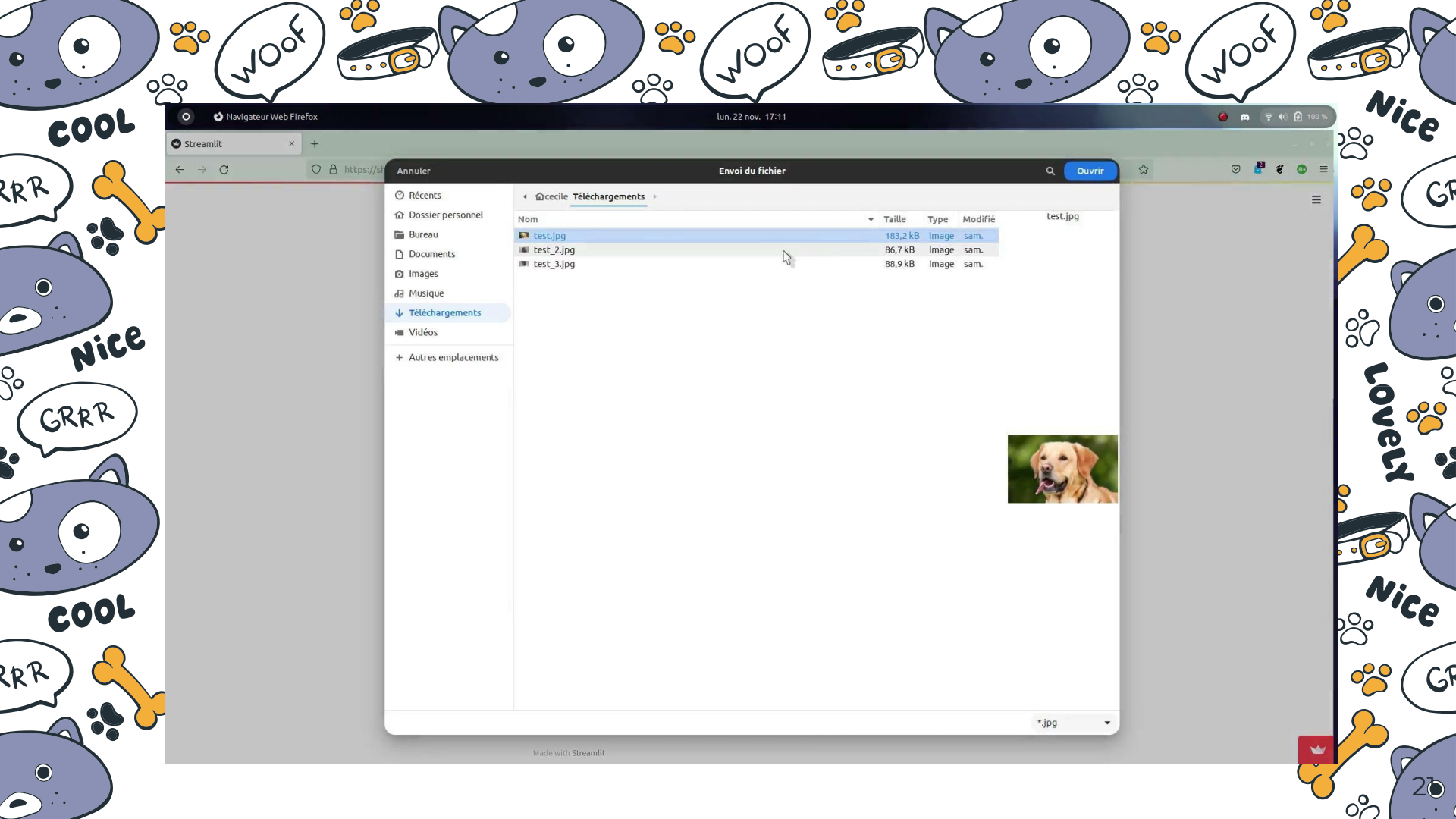
GitHub

Déploiement  
et interface



Streamlit







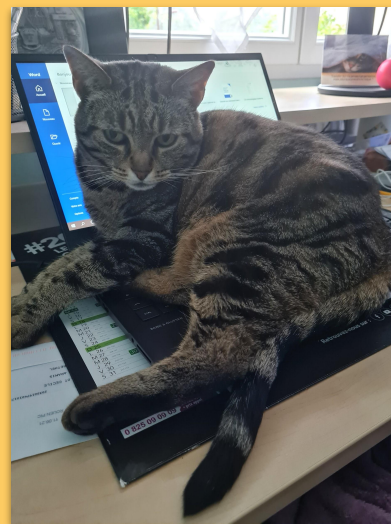
5



conclusion

# ET LA SUITE... ?

- Acquisition de données représentatives de la réalité du refuge
- Vérification des races les plus présentes en refuge
- Traitement des races croisées ? (Souvent présentes dans les refuges)
- Généralisation à d'autres espèces ? (Chats, lapins, etc.)



# Merci

Avez-vous des questions ?  
[cecile.gltslmcs@protonmail.com](mailto:cecile.gltslmcs@protonmail.com)  
[cecileguillot.com](http://cecileguillot.com)

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** and illustrations by **Stories**  
Please keep this slide for attribution

