

# OpenDogNet

How to detect dog breeds with a smartphone



OPENCLASSROOMS

colab  
kaggle

 Android  
Studio

# Outline

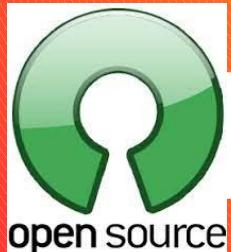
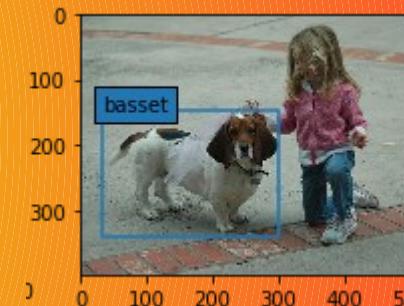
- Introduction
  - Why this app, what do we want to achieve ?
- The theory
  - How does such an app work ?
  - Convolutional neural networks (CNN) : the artificial intelligence (AI) technology powering the breed identification
  - How do I cast a the neural network into a simple intuitive Android app ?
- The practice
  - Let's test it in real life !
- Conclusions
  - does this stuff work or maybe it would be better to get something else even if will cost me money or time watching ads ?

# The goal

We use the technology of supervised deep learning to :



- Identify with good accuracy ( $P > 80\%$ ) the breed of a dog on a photo
- Build a small app (< 20 MB) that will make run the AI program on an Android device



All to be done using Open Souce software only, the app will be free of charge and will not display ads

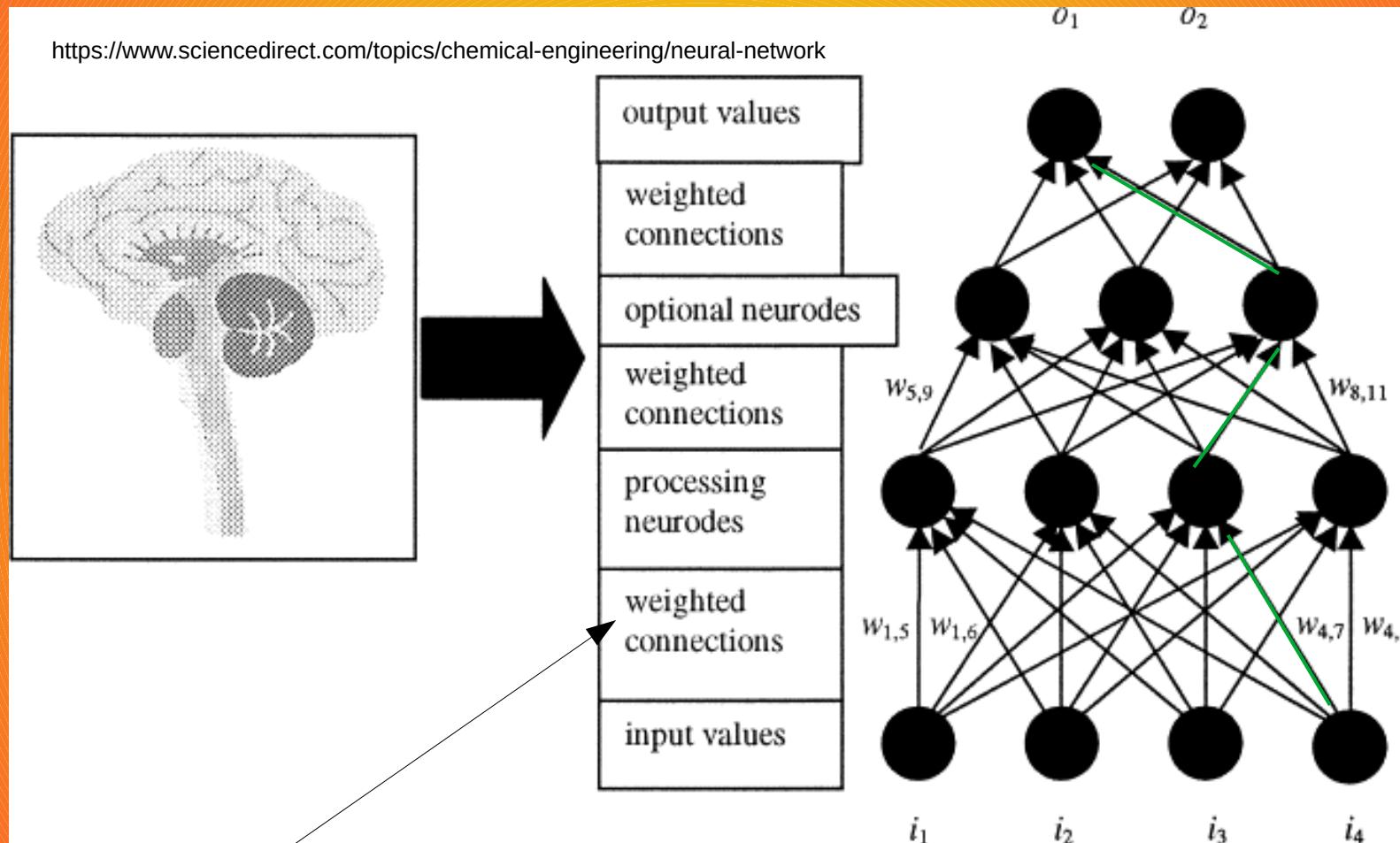
# How does it work ?



- What is deep learning ? What is a CNN ?
- How can I build and train a deep learning alghorithm ?
- How do can I deploy the trained AI to an Android app ?

# Neural networks : principle of DL

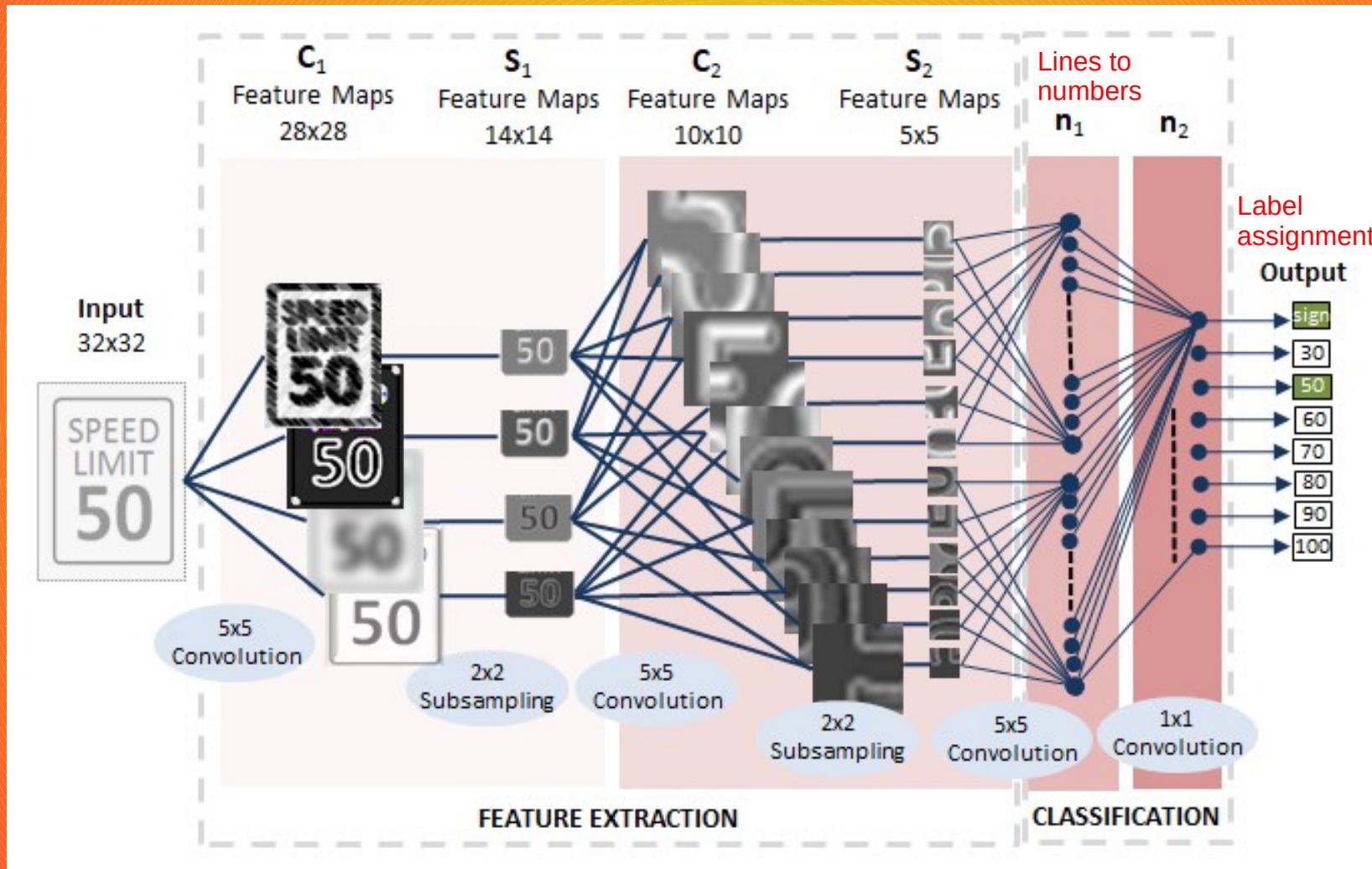
Connected units, named neurons, stacked on layers (that is why is « deep »)



Some connections are more equal than the others, so some input information becomes more important than the rest

# From images to probabilities

Extract edges from the image, identify which edges are most important (neuron weight), pass over to the next layer

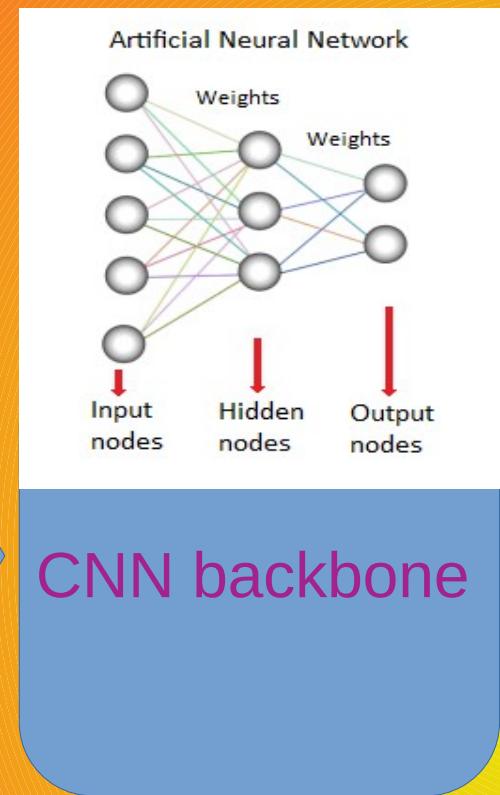


# How an AI learns

Input



299 X  
299 X  
3



flatten

Fully Connected

Class name	Output	Labels
chihuahua	0.75	1
basset	0.11	0
labrador	0.05	0
kelpie	0.02	0

Output (4\_classes)  
Each class has probability  
 $0 < P < 1$

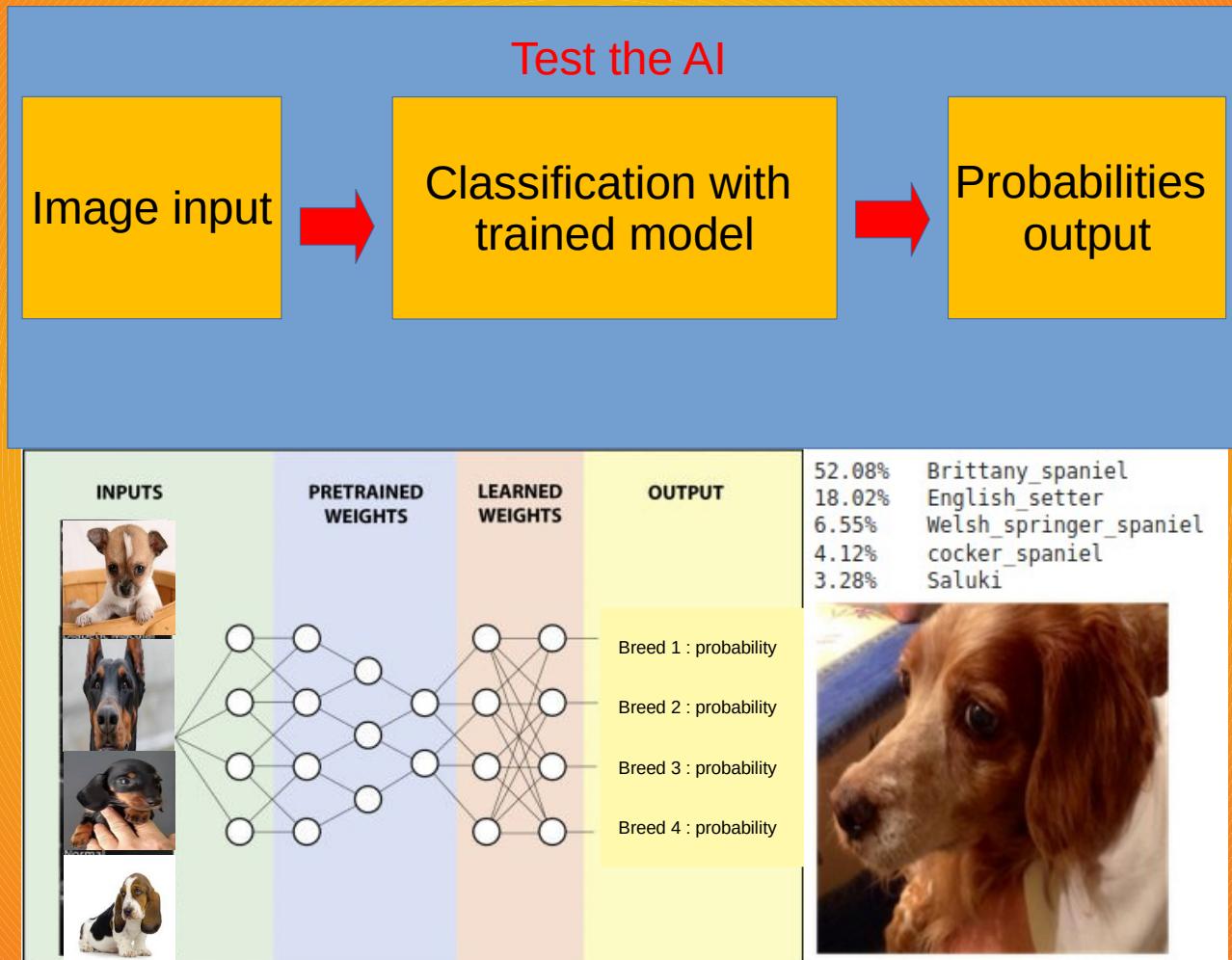
Labels  
3 classes = False (0)  
1 class = True (1)

Image features selection

Classification

Check output with labels and update weights to have a better match

# Deep learning applied to breeds



# Building our deep learning model

- Where do I get photos etc. ?
- Preprocessing – prepare the data for the AI
- Coding the neural network



# The data to train our model

## Data Explorer

751.55 MB

- annotations
- Annotation
- images
- Images

## Stanford dogs dataset

The screenshot shows a file explorer window titled "Data Explorer". The total size is 751.55 MB. The "Images" folder contains 120 sub-directories, each representing a dog breed. The breeds listed are Chihuahua, Japanese, Shih-Tzu, and Blenheim. Each breed folder contains a specific number of files: Chihuahua has 152 files, Japanese has 185 files, Shih-Tzu has 214 files, and Blenheim has 188 files.

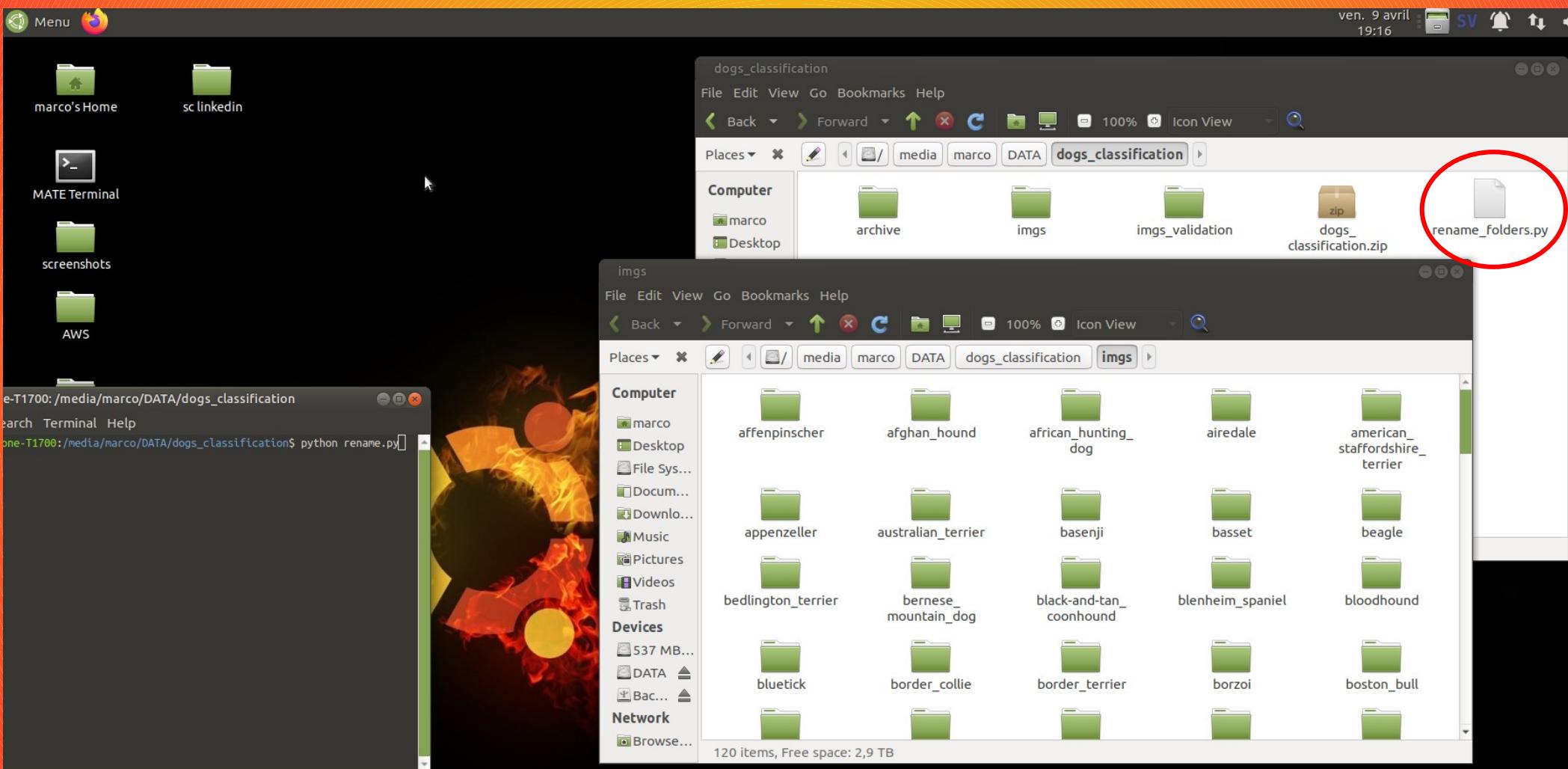
Breed	Number of Files
n02085620-Chihuahua	152 files
n02085782-Japanese...	185 files
n02086240-Shih-Tzu	214 files
n02086646-Blenheim...	188 files

120 folders (one for each breed) containing a total of 20000+ photos  
All available for free from Kaggle.com



# Data pre-processing

Our algorithm needs labels to understand what a photo corresponds to



Solution : rename each folder and I state that the folder name is the label for each photo that is inside it

# How do I build an AI



Time to get to my  
programming console

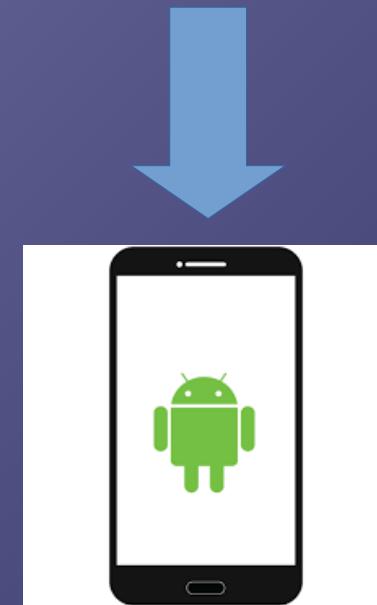
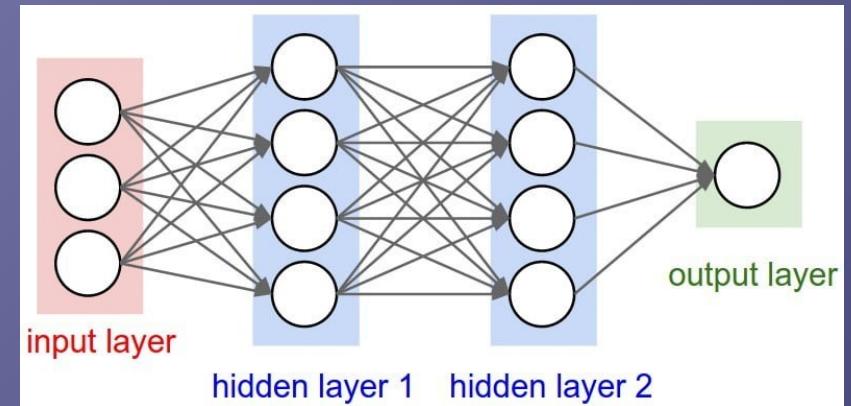
- Program in Python and use already written code (libraries) made for deep learning



- Run the program on a powerful server (cloud computing) so that it processes data much faster than my home PC

# Deploy it to Android

- Fully trained
- Precise (85% accuracy)
- Lightweight (<50 MB)
- Works offline



# From the PC to the phone

Build the CNN



Train it on dogs photos  
(Colabs GPU)

CO

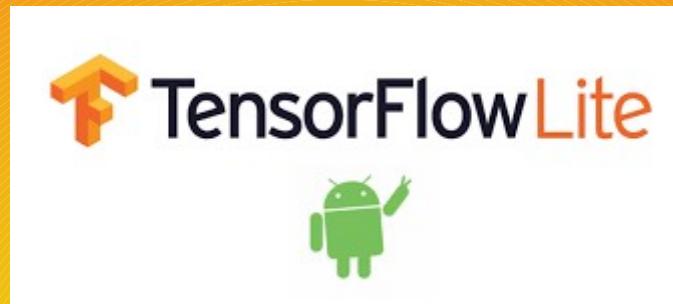


85 %  
accuracy

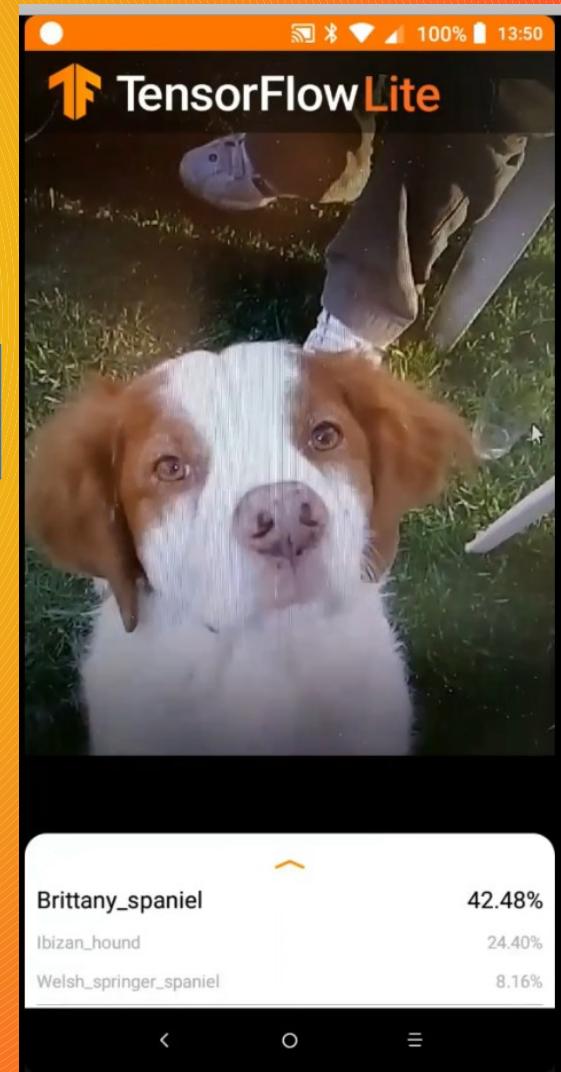
Tensorflow-Lite  
converter



Model file  
.tflite



APK  
file

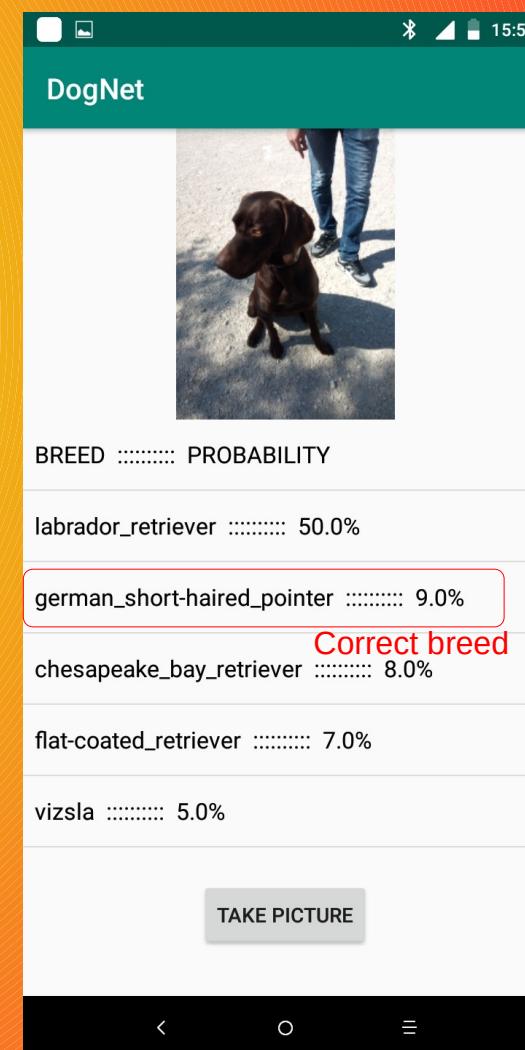
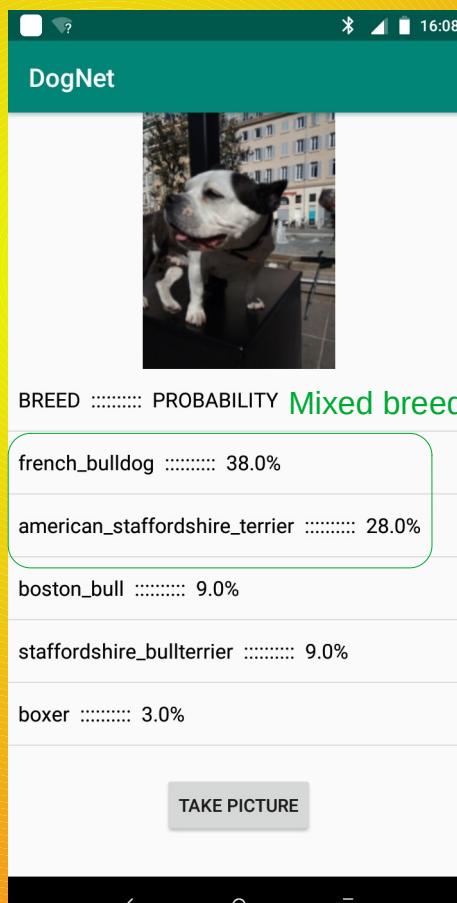
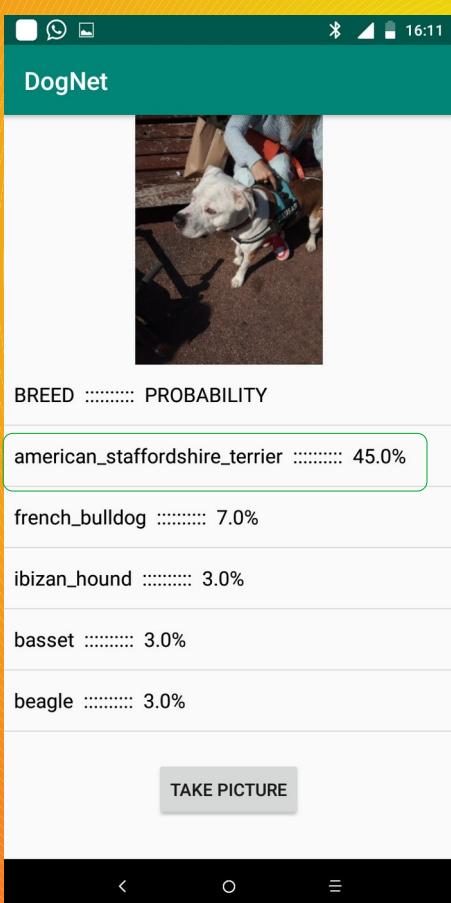
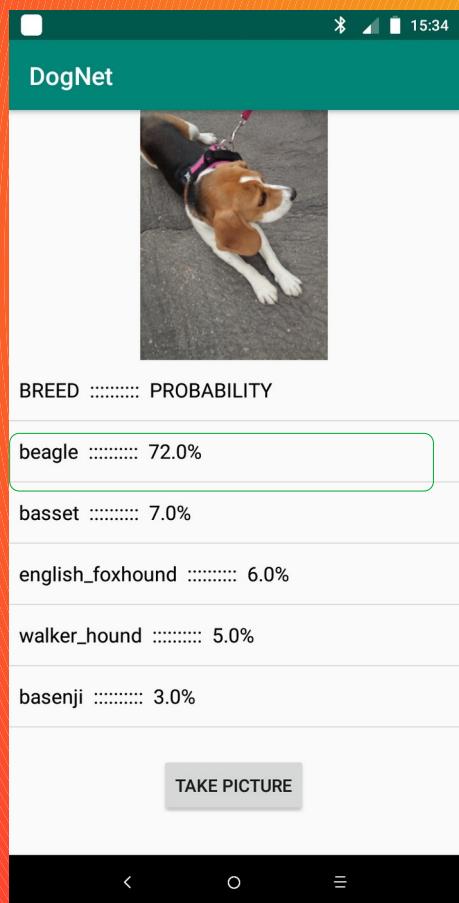


# Test it on the field !!!

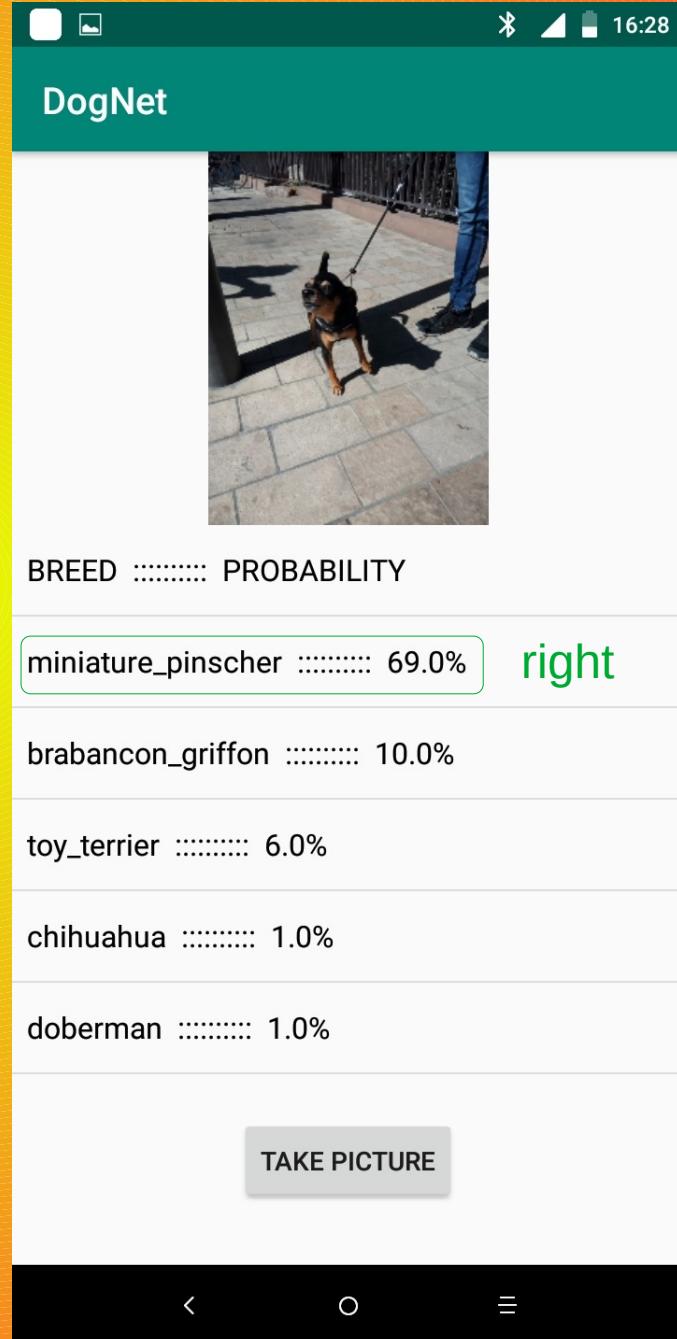
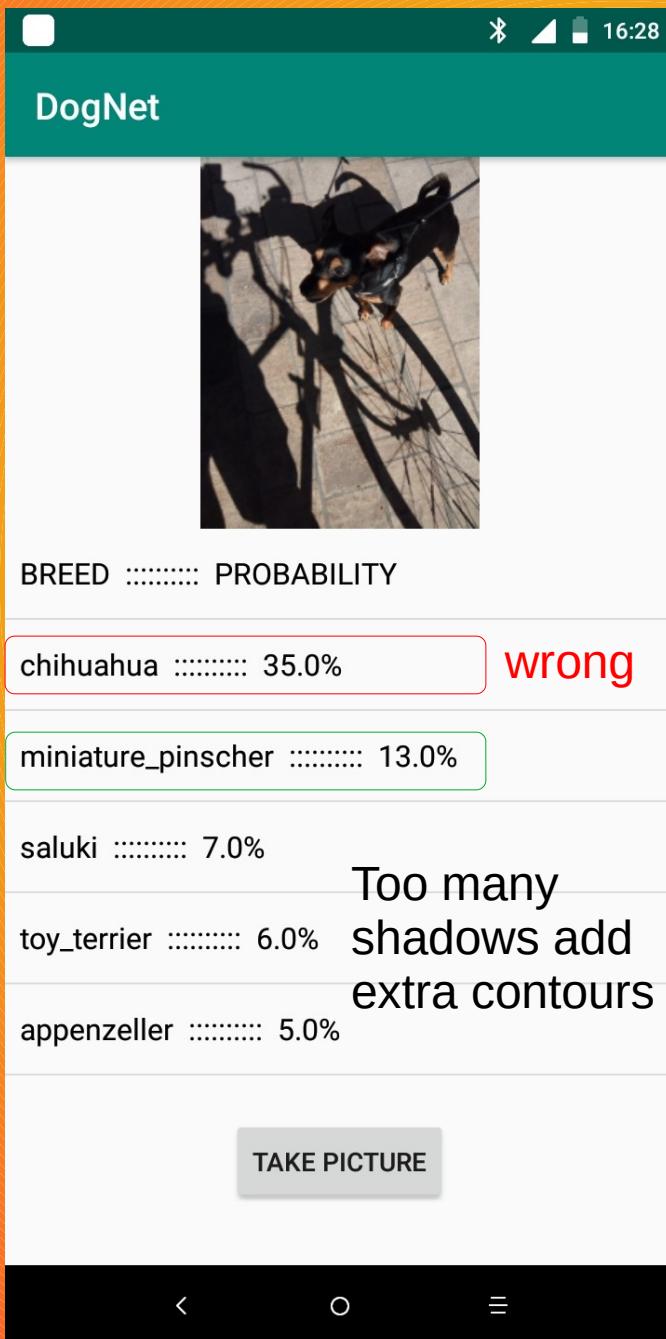


Walk around Marseille old harbour (Vieux Port) and kindly ask dog owners

# The aftermath



# Shadows can confuse an AI



# Conclusions

- Deep learning model developed with Tensorflow light works like a charm (85 % accuracy on test set)
- When deployed to Android, we obtain a lightweight app (14 MB) that recognises correctly most breeds
- Dog owners get quite interested and ask questions when they see the app in action

Links to Python codes, references etc. can be found below,  
in the video description

# Credits

Android developer : Giulio Giandonato



Main model for test photos: Ala (Giulio's dog)

Video editing : Giovanni Brajato



## Translations

Leszek Spalek : Polish  
Kani Pae : Thai  
Roberto di Pace : Swedish  
Amine Hocine : Arabic  
Belinda Verhagen : Dutch  
Susan Barnes : German  
Oleg Rofman : Russian  
Joana Sadio : Portuguese



.... and more friends who translated the app description into even more languages