DOG BREED CLASSIFICATION USING NEURAL NETWORKS

AINA BELLONI 5007697

GIULIA BEATRICE CRESPI 5009457

DATASET

Source :
http://vision.stanford.edu/aditya86/ImageNetDogs/

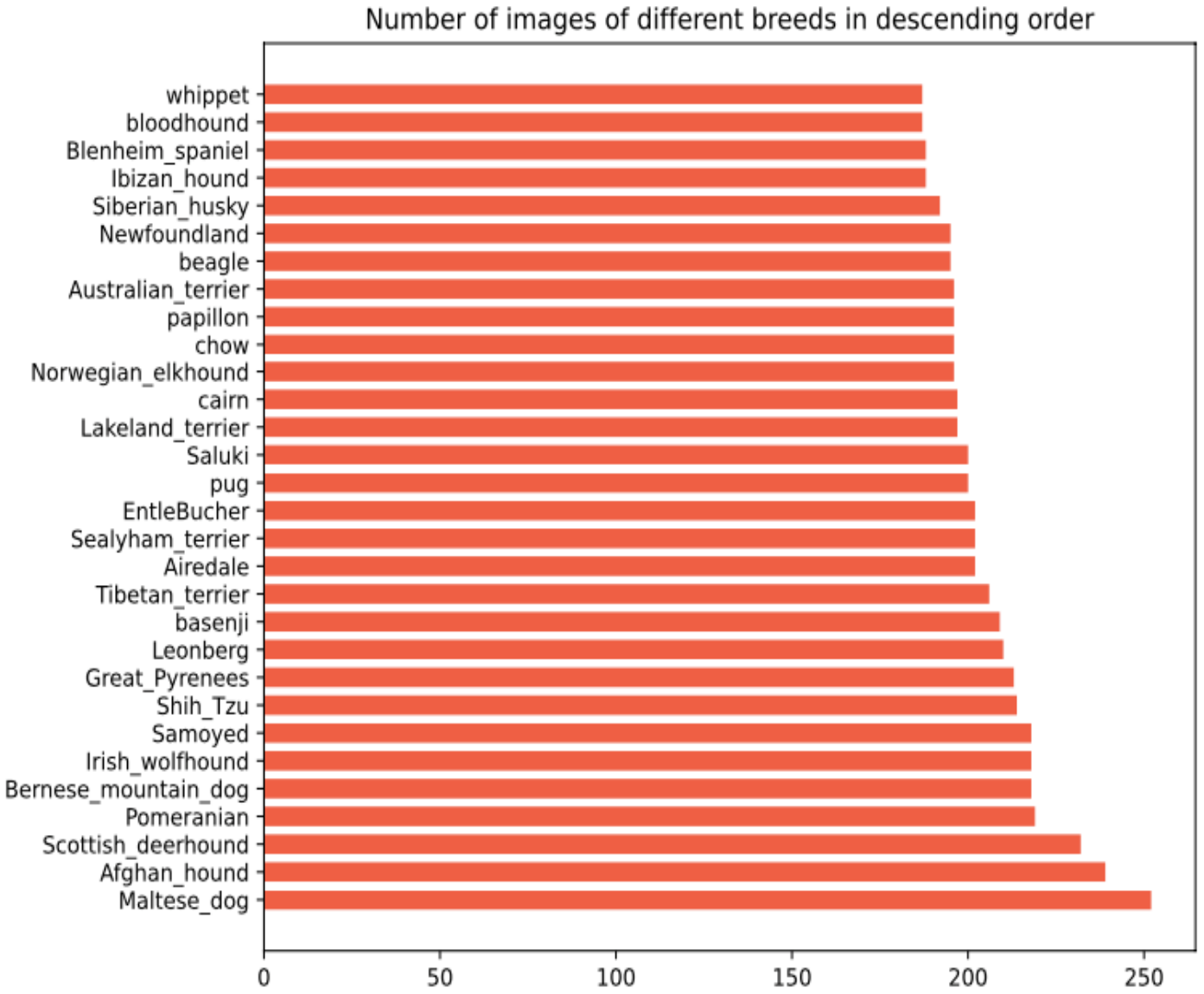
Total number of images : 20580 ~ 170 per breed

Number of breeds : 120

In the folder Images we have 120 folders containing each the images corresponding to one breed, while in the Annotation folder we have the information of the boundary boxes with the dog for each image.

Problem: Lot of classes and few data for each class

- Difficult to make accurate predictions
- Risk of overfitting
- Long time to train the model



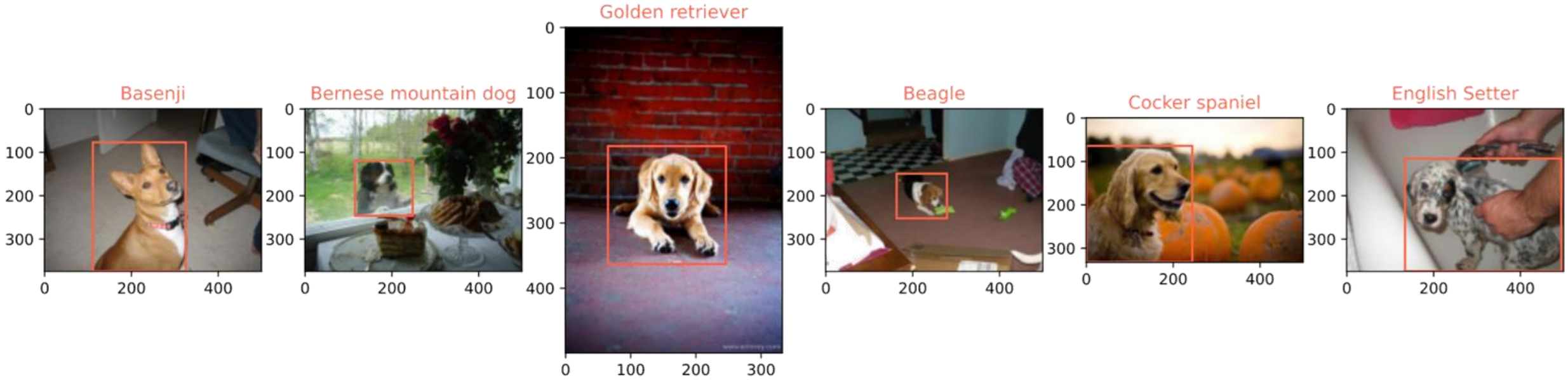
BOUNDARY BOXES

BOXES FROM ANNOTATIONS

Extracting informations contained in the files of the folder Annotations to create the boxes where the dog is contained.

OBJECT DETECTION USING NEURAL NETWORKS

We used Faster R-CNN ResNet50 to detect objects in the images, that creates their own bounding boxes.



PREPARE THE DATA

CROP

Boundary Boxes

RESIZE

224x224x3

DATA AUGMENTATION

Horizontal Flip
Distortion of the image

TENSOR

Torch tensors

NORMALIZE

Mean (0.485, 0.456, 0.406)
Sd (0.229, 0.224, 0.225)

DATASET.IMAGEFOLDER

Import the data and transform

SPLIT

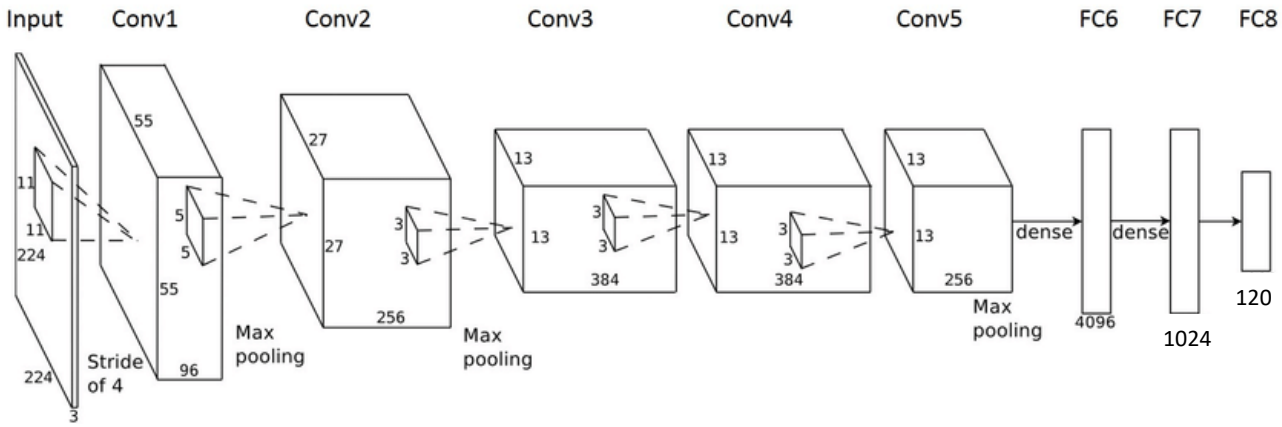
Train 80% - 16464 images
Test 20% - 4116 images

DATALOADER

Batch_size : 16
Train shuffle = True
Test shuffle = False

AlexNet

- Model imported from **PyTorch**
- Changed the last 2 layers
- **Loss** : CrossEntropyLoss
- **Optimizer** : Stochastic Gradient Descent with learning rate 0.001
- Two implemetations:
  - **Not pretrained** : updated all the parameters
  - **Pretrained** : used as a fixed feature extractor, updated only the paramters of the last layer

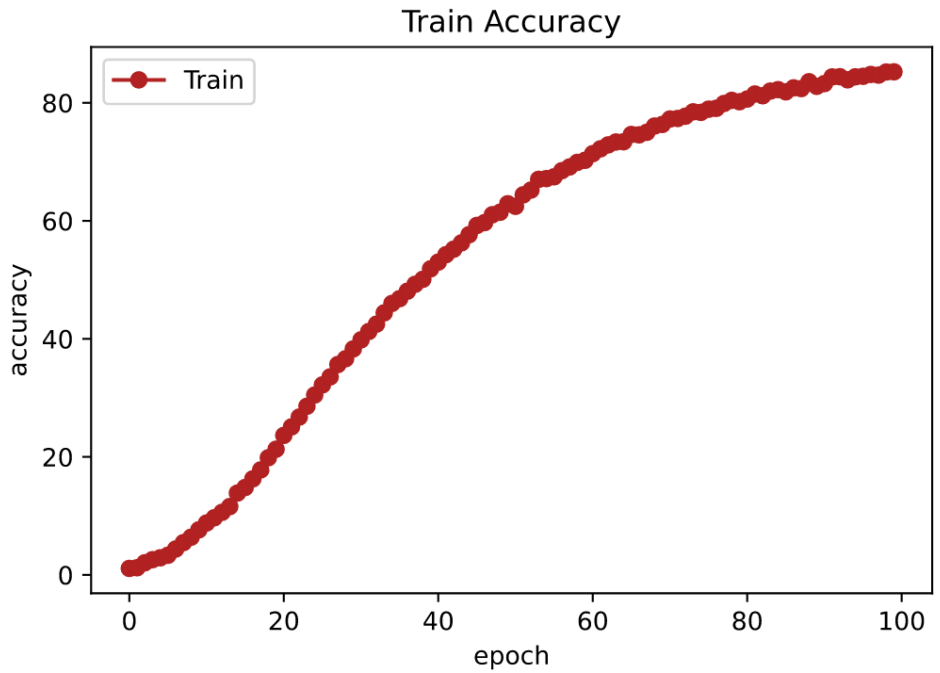



```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=1024, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=1024, out_features=120, bias=True)
  )
)
```

AlexNet results

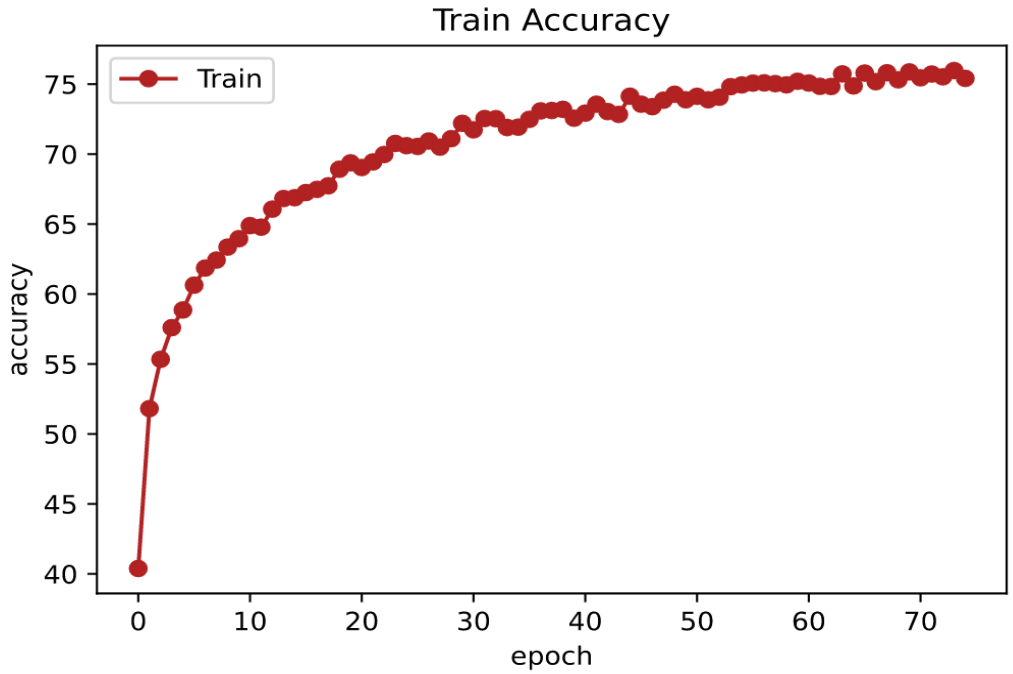
AlexNet not pretrained in 100 epochs reached an accuracy of:

- 85,25% on train set
- 85,59% on test set



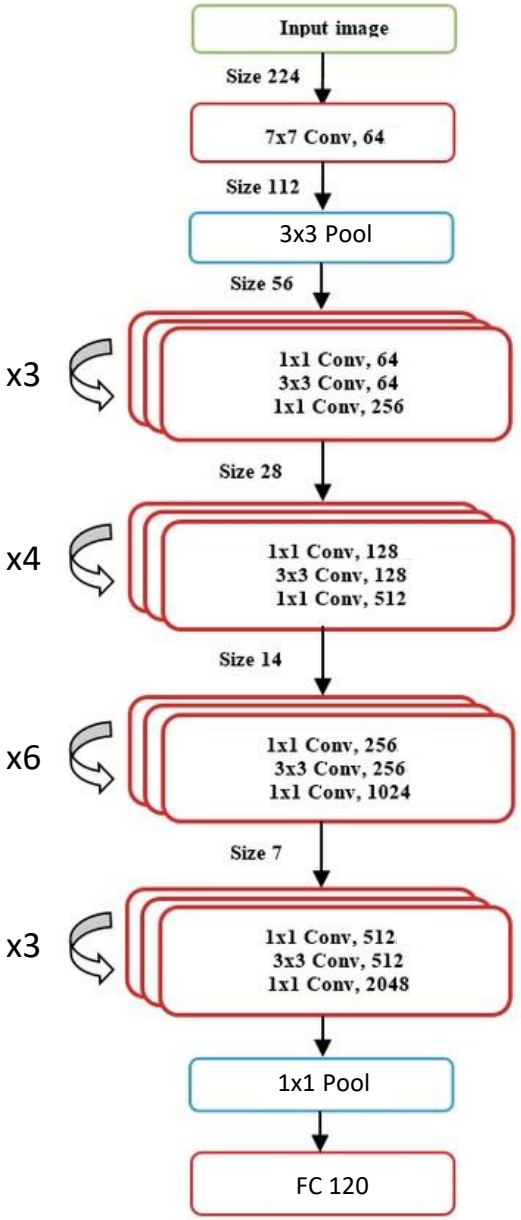
AlexNet pretrained as fixed feature extractor in 75 epochs reached an accuracy of:

- 75,41% on train set
- 82,09% on test set



ResNet50

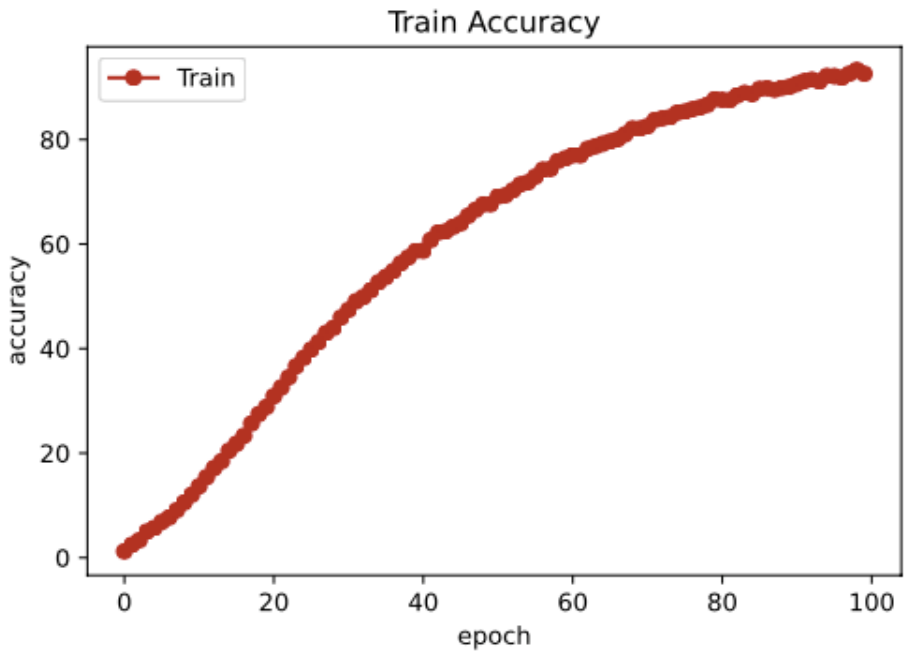
- Model imported from **PyTorch**
- Changed the last layer
- **Loss** : CrossEntropyLoss
- **Optimizer** : Stochastic Gradient Descent with learning rate 0.001
- Two implemetations:
  - **Not pretrained** : updated all the parameters
  - **Pretrained** : used as a fixed feature extractor, updated only the paramters of the last layer



ResNet50 results

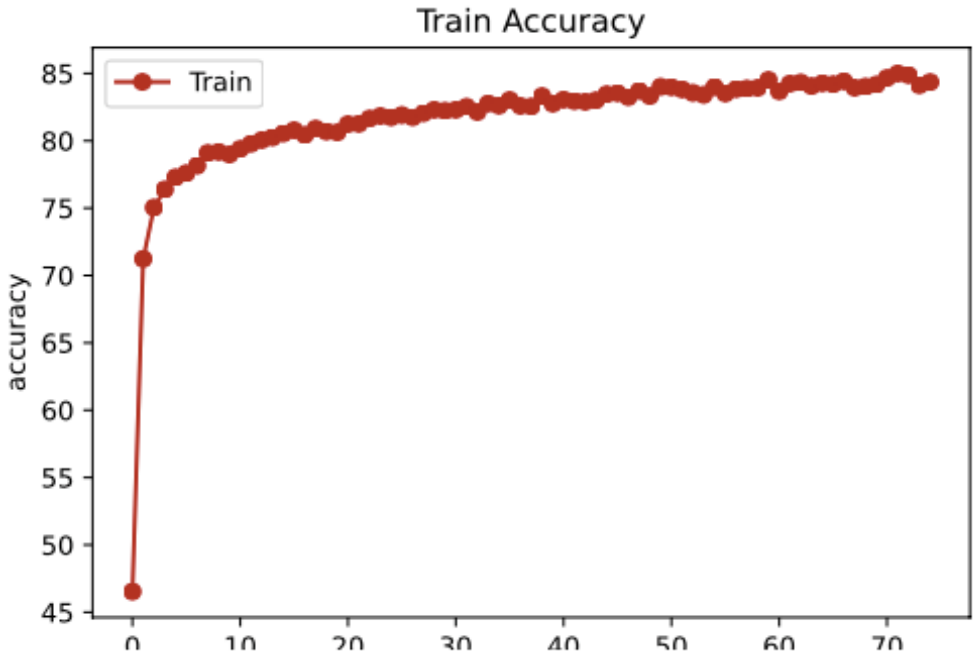
ResNet50 not pretrained in 100 epochs reached an accuracy of:

- 92.58% on train set
- 87,83% on test set



ResNet50 pretrained as fixed feature extractor in 75 epochs reached an accuracy of:

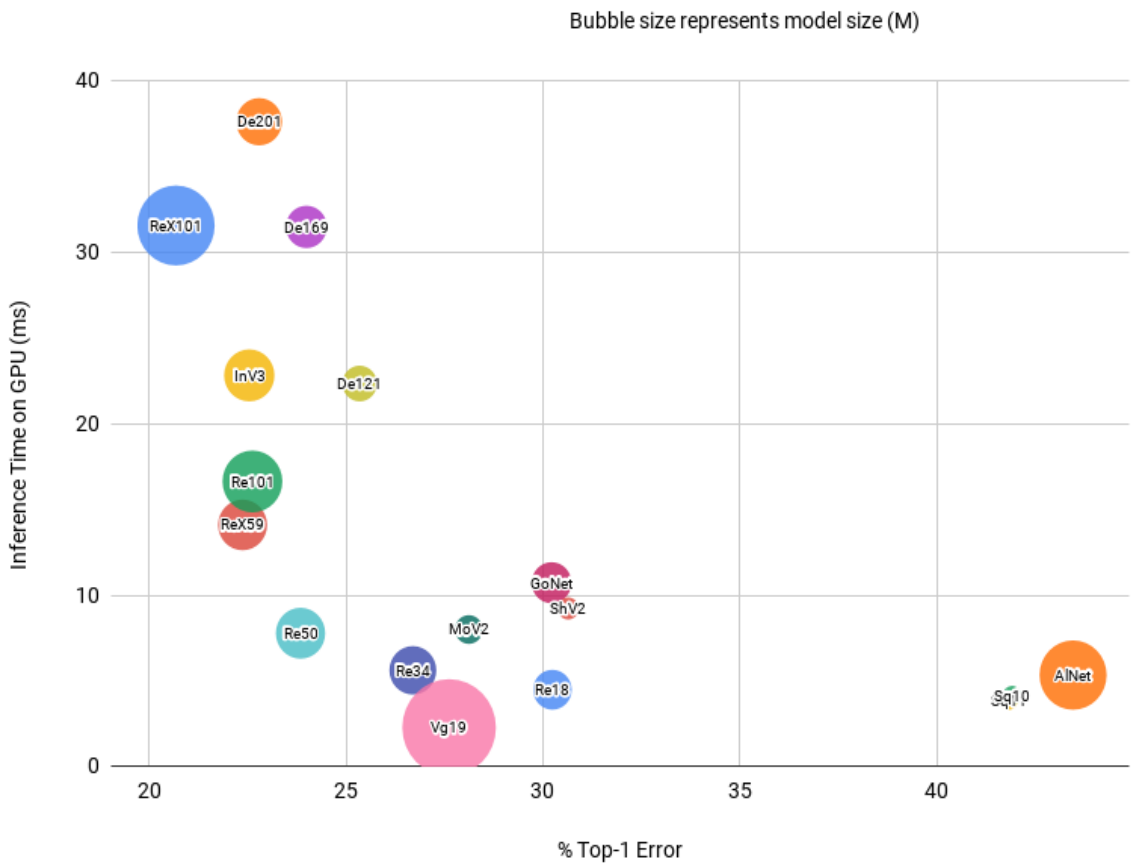
- 84,35% on train set
- 87,27% on test set



CONCLUSIONS

- **Differences between AlexNet and ResNet50**: ResNet is a deeper model so the results are better than AlexNet.
- **Differences between pretrained and not pretrained**: thanks to the work done previously on ImageNet we can have a faster training of the model. ResNet50 pretrained obtains a train accuracy of 75% in only 3 epochs.
- **Possible ideas to improve the performances**:
  - Greater number of epochs for not pretrained models
  - Fine tuning for the pretrained models
  - Decrease batch size
  - Decrese learning rate

PRE-TRAINED MODEL COMPARISON



PREDICTIONS

	MARGOT		KOBE	
REAL BREED	BERNESE MOUNTAIN DOG		UNKNOWN	
ALEXNET NOT PRETRAINED	AFGHAN HOUND		DHOLE	
ALEXNET PRETRAINED	GORDON SETTER		DHOLE	
RESNET50 NOT PRETRAINED	BORDER COLLIE		DINGO	
RESNET50 PRETRAINED	BERNESE MOUNTAIN DOG ✔		KELPIE