DIGITAL SIGNAL & IMAGE PROCESSING

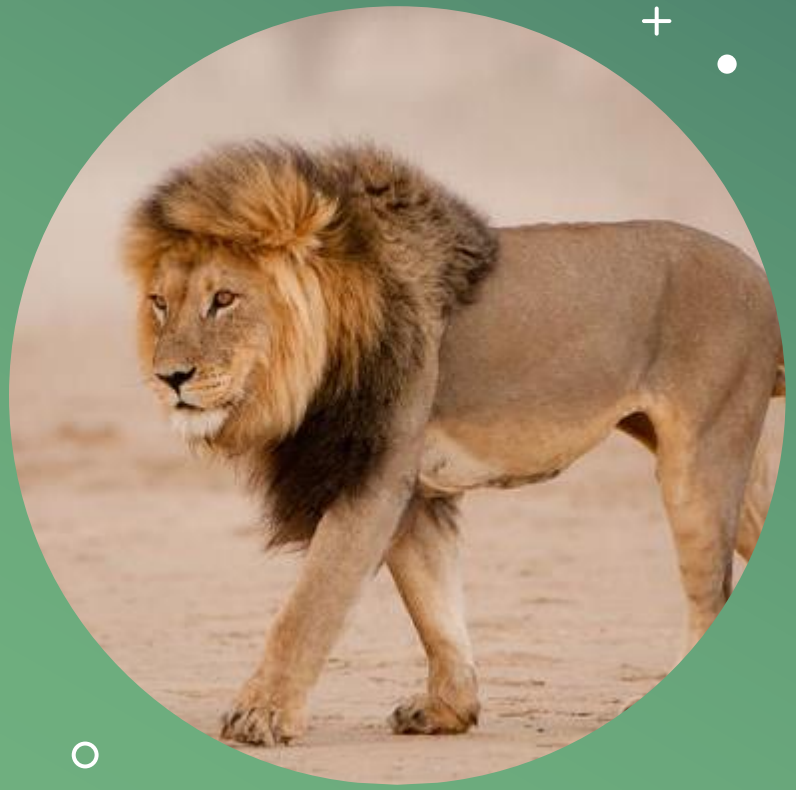
2020-2021

ANIMAL RECOGNITION

IMAGE-AUDIO RECOGNTION & RETRIEVAL FOR PICTURES AND SOUNDS OF ANIMALS

Federico Luzzi - 816753

Federico De Servi - 812166



TOPICS

1) AUDIO RECOGNITION

2) IMAGE RECOGNITION

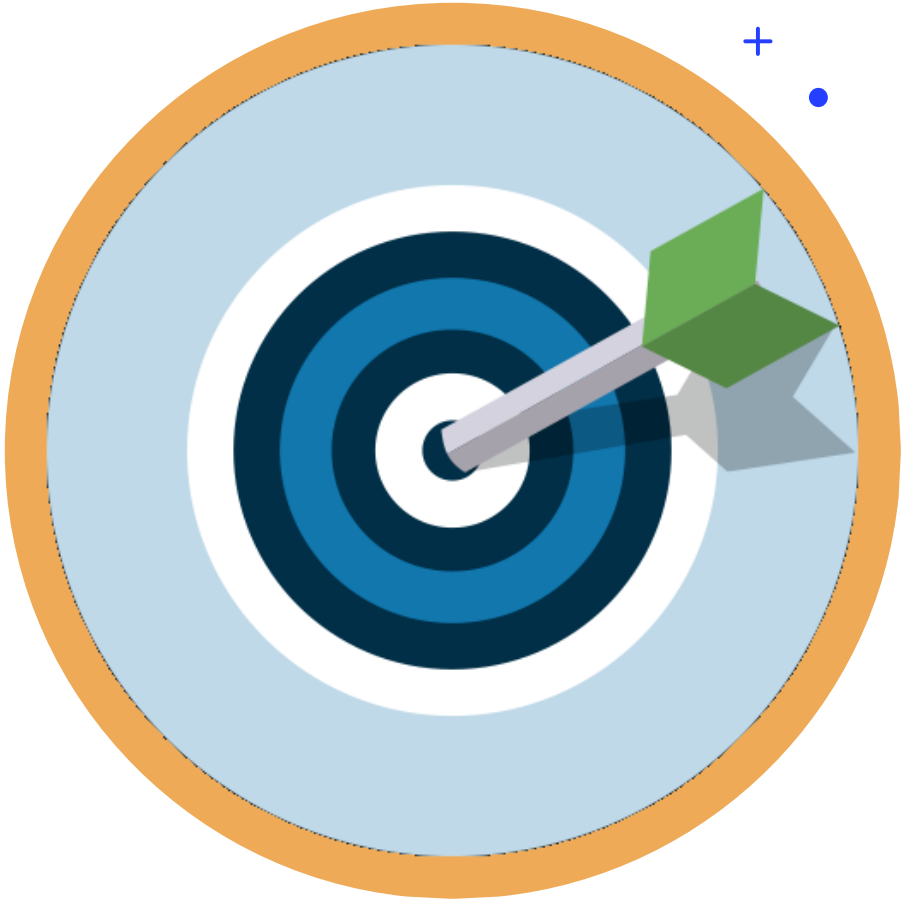
3) IMAGE RETRIEVAL

4) DEMO

AUDIO RECOGNITION

Objective

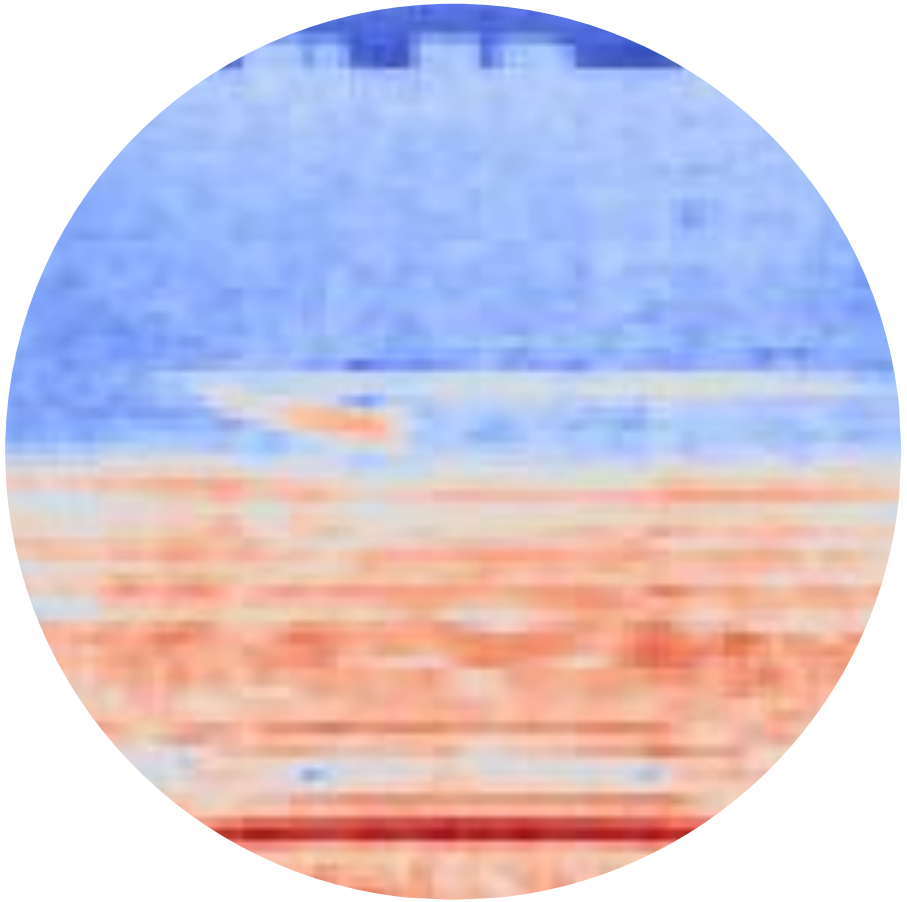
The objective is to train a CNN model in order to classify correctly the biggest number of audio containing sounds of animals. These sounds have a lenght that ranges from 2 to 3.5 seconds and different levels of background noise.



Dataset

The audio dataset has been downloaded from the Animal Sound Archive made available by the Museum für Naturkunde Berlin through the Gbif website. It is composed of audios whose lenght is variable. The audios has been manually cut in order to isolate the sound (3.5 sec max).

- Moose
- Buffalo
- Chimpanzee
- Deer
- Fox
- Horse
- Lion
- Otter
- Racoon
- Sheep



Preprocessing

First of all, the Resnet preprocessing function is applied. Then, after applying data augmentation to the audios (pitch shift, gaussian noise, normalization), a mel-spectrogram is created from each audio. A Frequency Mask is applied to some of the mel-spectrograms. This audio classification problem became then a classical image classification one, thus giving the opportunity to even use pre-trained models like ResNet.

Modelling

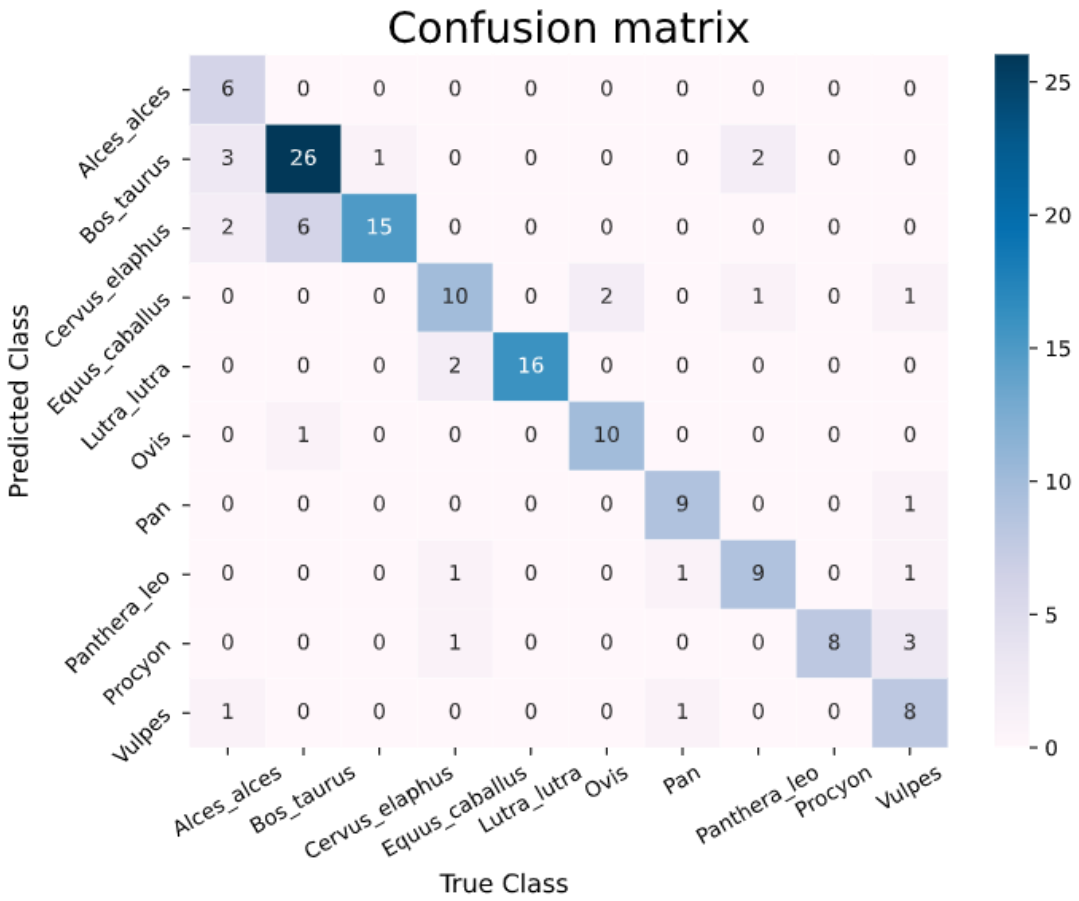
Since the problem is umbalanced, we calculated the first thing to do is to calculate the weights for each of the 10 classes, and to use them during training. Various models has been tried, ranging from simple CNN models to complex image recognition models.

After many trials, this model resulted to be the best. It uses ResNet101 with imagenet weights, followed by a block made of Dropout + ReLu activation function + BatchNormalization layer. Then, a Dense layer with 200 units and l2 weight regularization and another block of Dropout + ReLu activation function + BatchNormalization layer follow. In the end, a Dense layer with 10 units and softmax activation function is added as final layer.

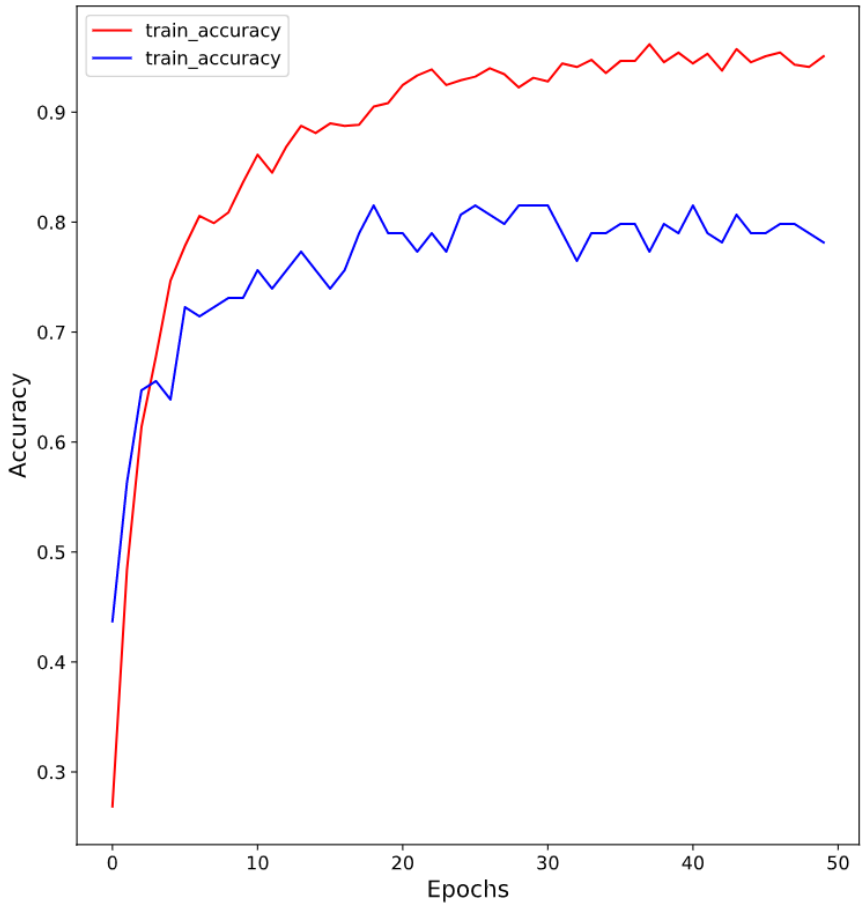
ResNet101

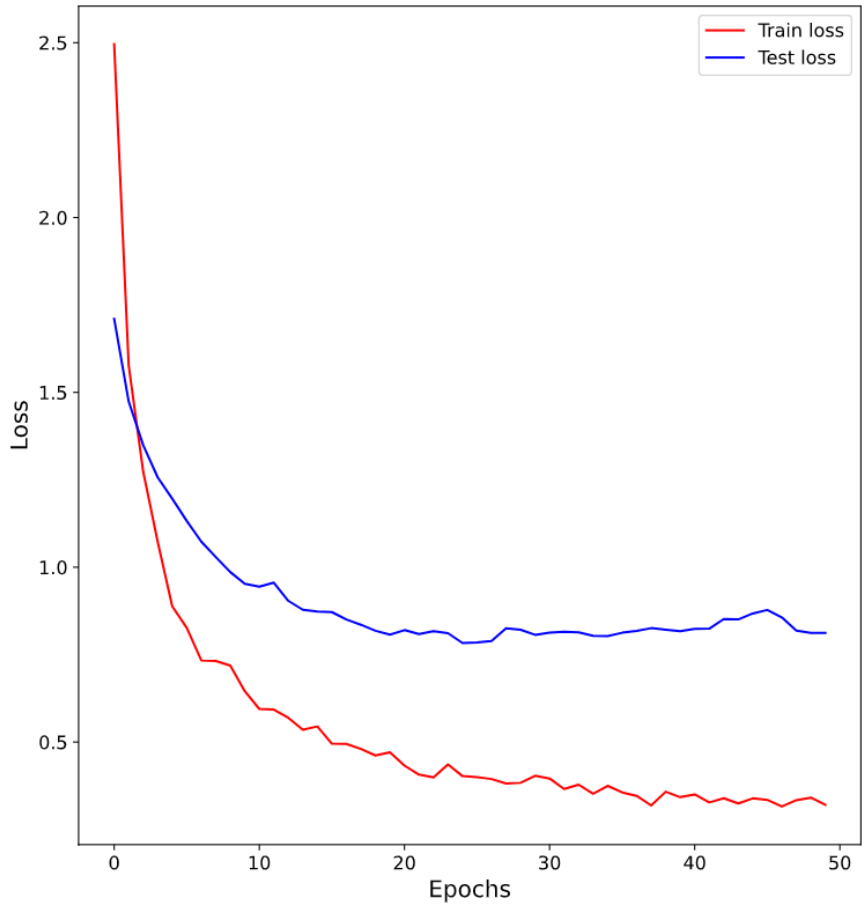
```
+
Dropout(0.5)
+
ReLU()
+
BatchNormalization()
+
Dense(200, activation='relu', kernel_regulari
zer=regularizers.l2(l2=0.0005))
+
Dropout(0.5)
+
ReLU()
+
BatchNormalization()
+
Dense(10, activation='softmax')
```

Results


Confusion matrix
Predicted Class
True Class
Alces_alces
Bos_taurus
Cervus_elaphus
Equus_caballus
Lutra_lutra
Ovis
Pan
Panthera_leo
Procyon
Vulpes
6 0 0 0 0 0 0 0 0 0
3 26 1 0 0 0 0 2 0 0
2 6 15 0 0 0 0 0 0 0
0 0 0 10 0 2 0 1 0 1
0 0 0 2 16 0 0 0 0 0
0 1 0 0 0 10 0 0 0 0
0 0 0 0 0 0 9 0 0 1
0 0 0 1 0 0 1 9 0 1
0 0 0 1 0 0 0 0 8 3
1 0 0 0 0 0 1 0 0 8
25
20
15
10
5
0

Results


train_accuracy
train_accuracy
Accuracy
Epochs
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
10
20
30
40
50


Train loss
Test loss
Loss
Epochs
0.5
1.0
1.5
2.0
2.5
0
10
20
30
40
50

Classifiers

ResNet101

Cut at «conv3_block1_2_relu»

Classical Classifiers

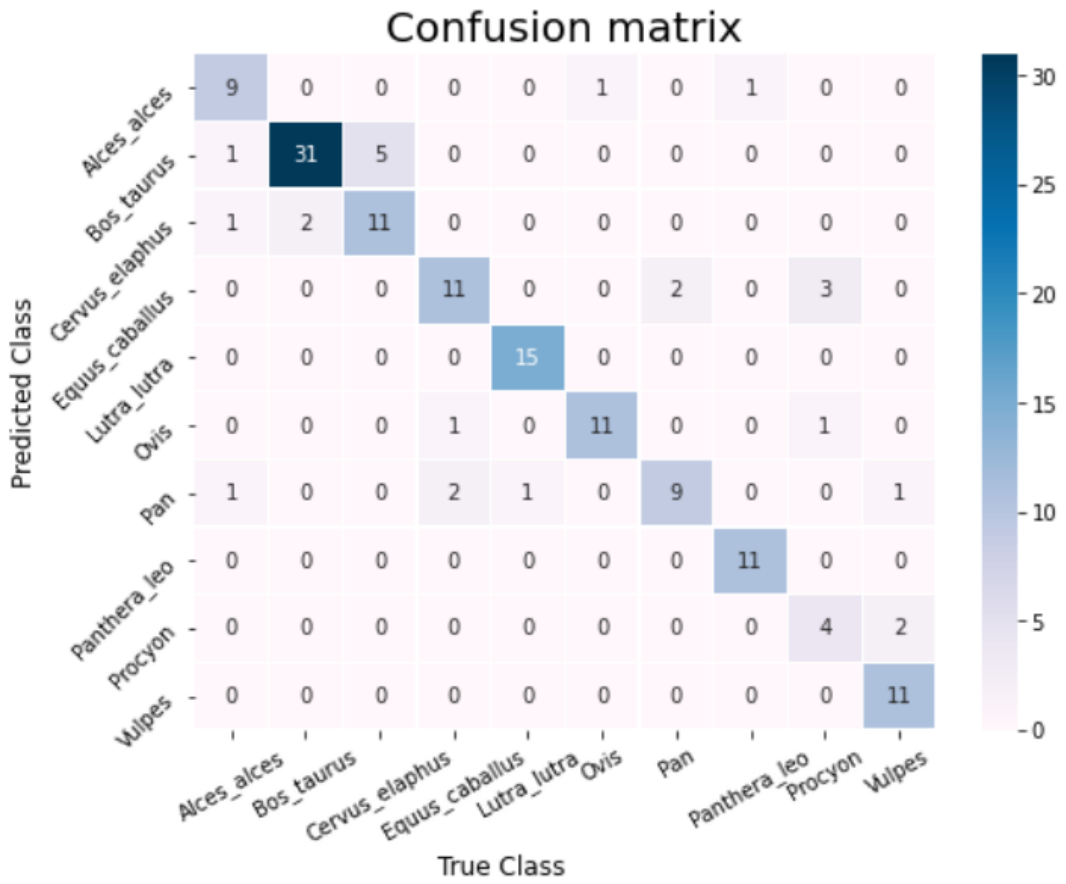
- SVM
- Logistic Regression

Since the audio dataset is small and very different from the ImageNet dataset, another approach used was to use the ResNet as a feature extractor and then train some classical classifiers (SVM, Logistic Regression) on those features.

This improved the performances obtained on the test set, at the expense of efficiency and execution speed of the predictions.

This also explains why the Demo contains the fine-tuned ResNet and not this approach.

Results (SVM)


Confusion matrix
Predicted Class
True Class
Alces_alces
Bos_taurus
Cervus_elaphus
Equus_caballus
Lutra_lutra
Ovis
Pan
Panthera_leo
Procyon
Vulpes
9 0 0 0 0 1 0 1 0 0
1 31 5 0 0 0 0 0 0 0
1 2 11 0 0 0 0 0 0 0
0 0 0 11 0 0 2 0 3 0
0 0 0 0 15 0 0 0 0 0
0 0 0 1 0 11 0 0 1 0
1 0 0 2 1 0 9 0 0 1
0 0 0 0 0 0 0 11 0 0
0 0 0 0 0 0 0 0 4 2
0 0 0 0 0 0 0 0 0 11
30
25
20
15
10
5
0

Results (L. Regression)

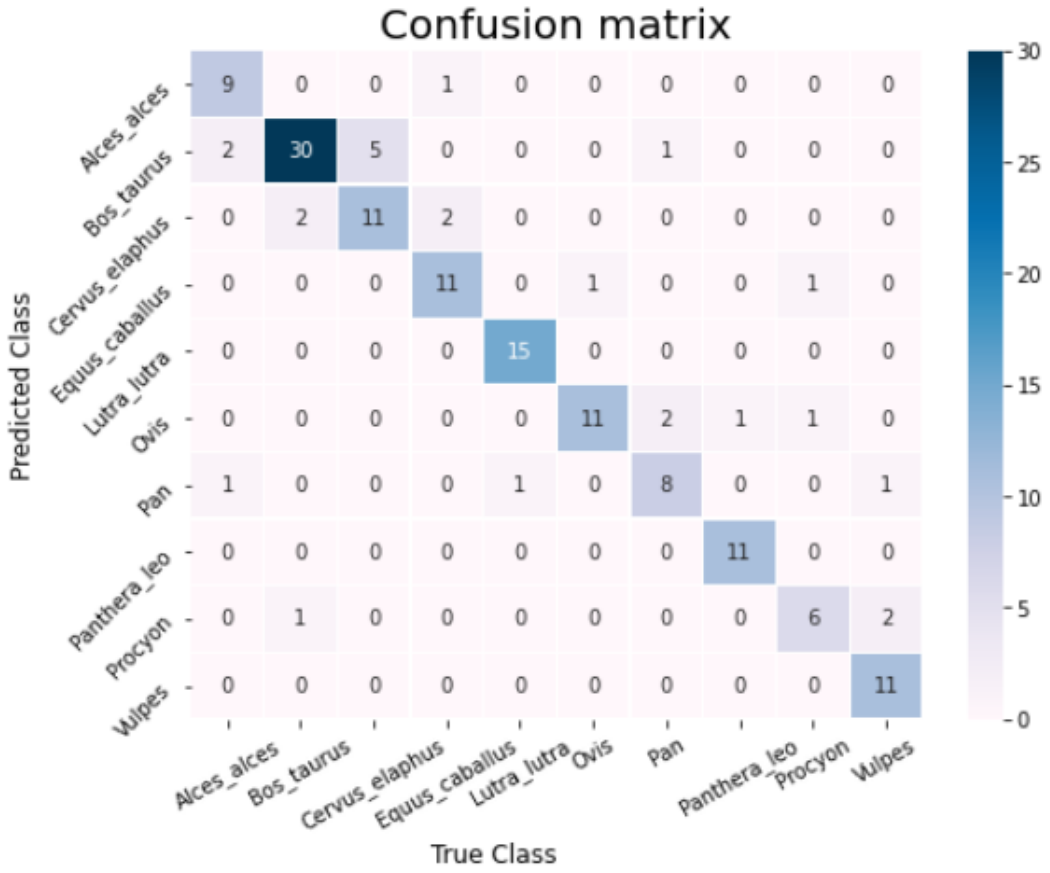
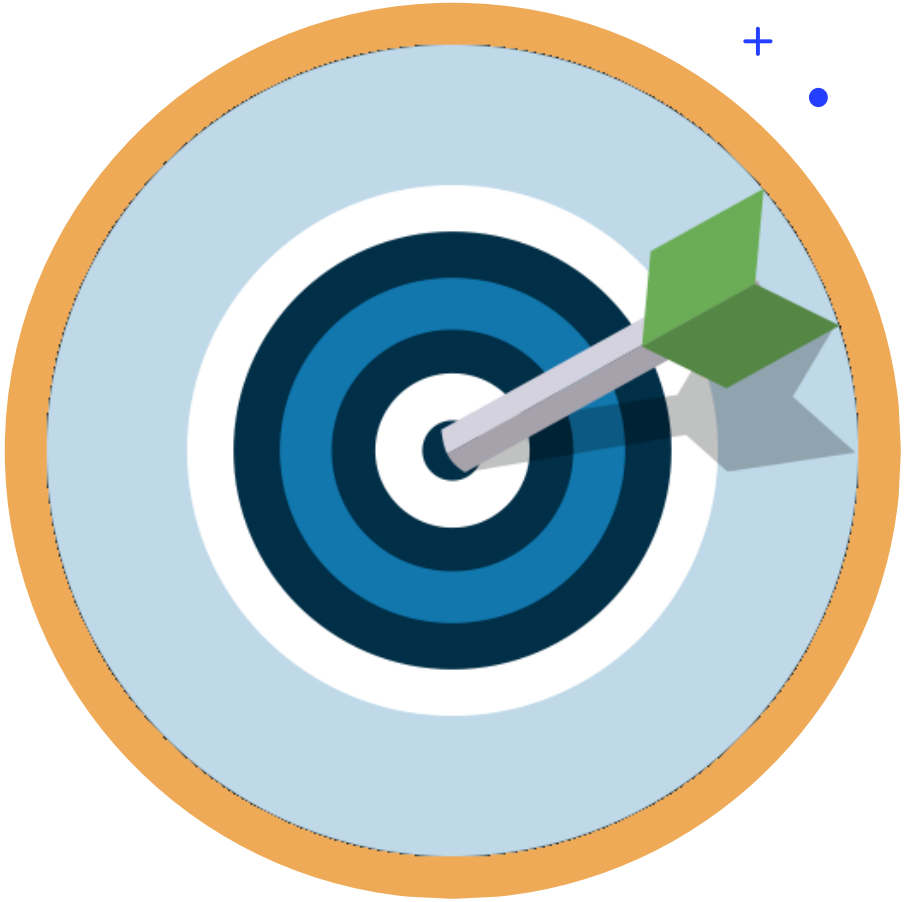


IMAGE RECOGNITION

Objective

This time the objective is to perform a classic image classification problem. After applying some data augmentation techniques to the images (*ImagesWithAttributes2*), transfer learning with different pretrained models has been used.



Dataset

The image dataset has been downloaded from the Animal with Attributes 2 website. It is composed of thousands of pictures of animals of various sizes. The selected classes are:

- Moose
- Buffalo
- Chimpanzee
- Deer
- Fox
- Horse
- Lion
- Otter
- Racoon
- Sheep

Preprocessing

First of all, the Resnet preprocessing function is applied. After doing that, data augmentation techniques, like zooming, flipping horizontally, rotating, shifting and changing the brightness are applied to the images. Then, the images are imported as 224x224x3 rgb images and fed them to the model.

Modelling

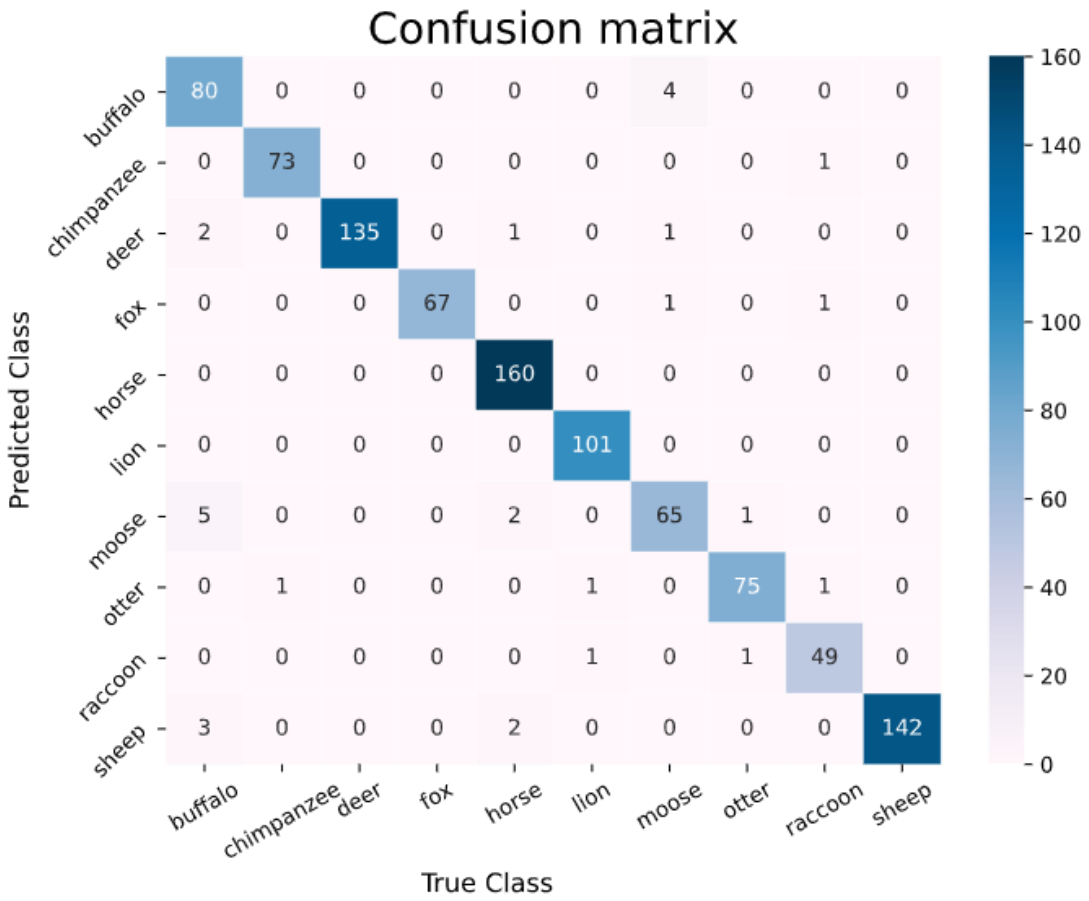
Since it is still an umbalanced problem, the weights for each of the 10 classes are calculated, and used during training. The obvious choice here was to use transfer learning from previously trained models on imagenet.

After many trials, this model resulted to be the best. It uses ResNet101 with imagenet weights, followed by a BatchNormalization layer. Then, a Dense layer with 50 units and l2 weight regularization and another BatchNormalization layer follow. In the end, a Dense layer with 10 units and softmax activation function is added as final layer.

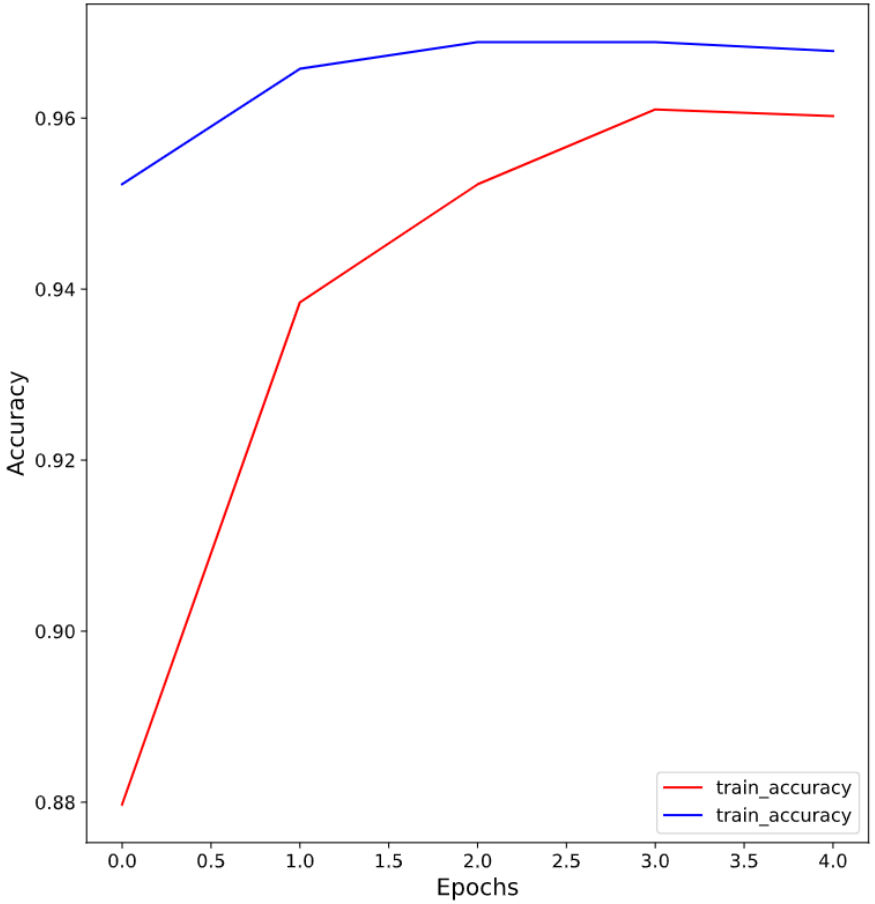
ResNet101

```
+
BatchNormalization()
+
Dense(50, activation='relu', kernel_regulariz
er=regularizers.l2(l2=0.0005))
+
BatchNormalization()
+
Dense(10, activation='softmax')
```

Results


Confusion matrix
Predicted Class
True Class
buffalo
chimpanzee
deer
fox
horse
lion
moose
otter
raccoon
sheep
80 0 0 0 0 0 4 0 0 0
0 73 0 0 0 0 0 0 1 0
2 0 135 0 1 0 1 0 0 0
0 0 0 67 0 0 1 0 1 0
0 0 0 0 160 0 0 0 0 0
0 0 0 0 0 101 0 0 0 0
5 0 0 0 2 0 65 1 0 0
0 1 0 0 0 1 0 75 1 0
0 0 0 0 0 1 0 1 49 0
3 0 0 0 2 0 0 0 0 142
160
140
120
100
80
60
40
20
0

Results


0.96
0.94
0.92
0.90
0.88
Accuracy
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs
train_accuracy
train_accuracy

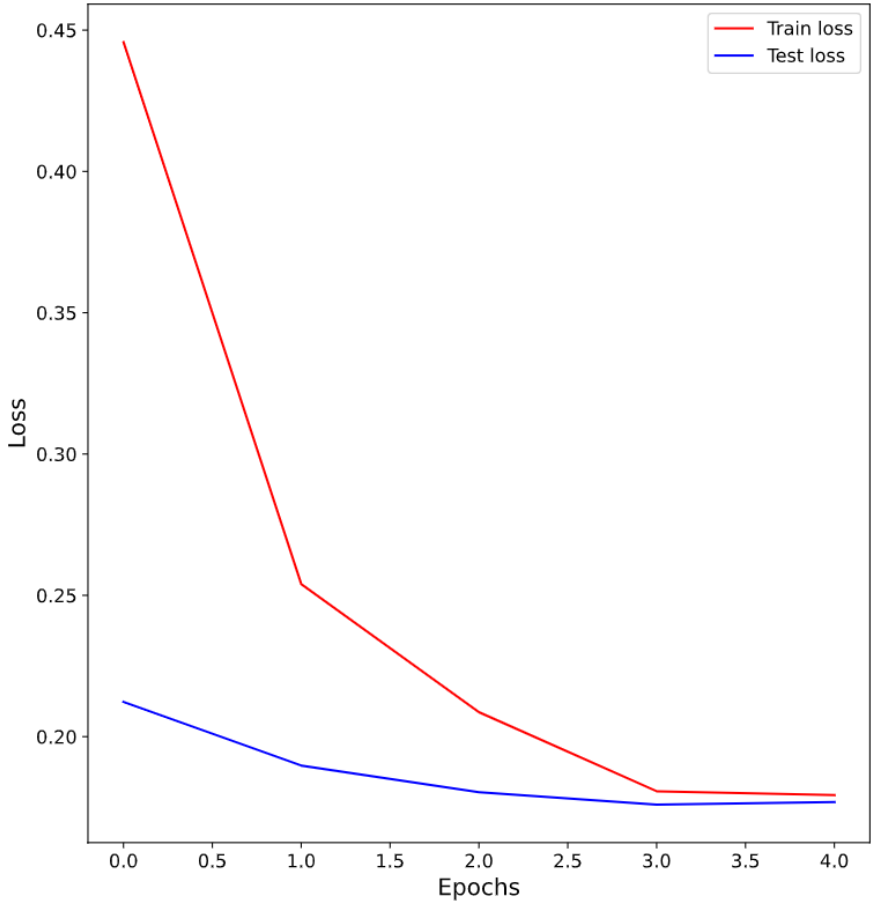
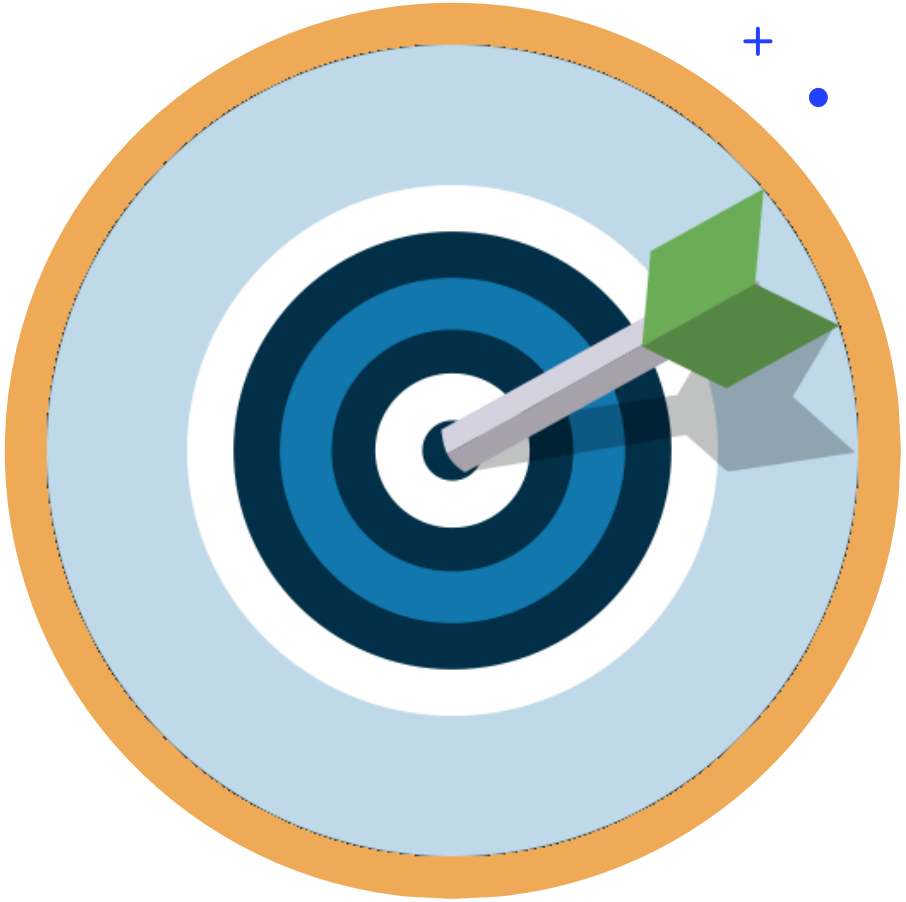

0.45
0.40
0.35
0.30
0.25
0.20
Loss
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epochs
Train loss
Test loss

IMAGE RETRIEVAL

Objective

The goal of the **Image Retrival** task is to select the 9 most similar animal images based on a query one. In particular, this tool would perform on real animals images and picture ones.



Dataset

The image dataset is the same as before. It is composed of thousands of pictures of animals of various sizes. The same classes as before are selected:

- Moose
- Buffalo
- Chimpanzee
- Deer
- Fox
- Horse
- Lion
- Otter
- Racoon
- Sheep

Some paintings belonging to those 10 classes also selected and downloaded from google image search for testing purposes.

Preprocessing

The neural network wants an input image of 224x224 pixels with 3 color channels (RGB). So the *query image* would be resized to fit in.

Also resnet preprocess function is applied.

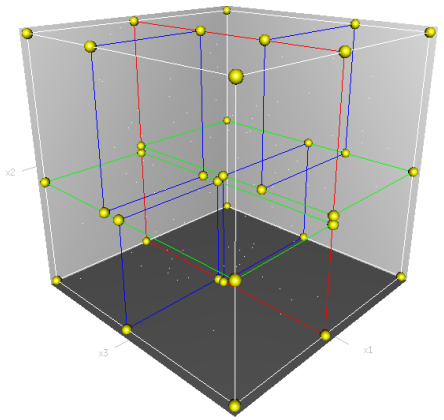
Modelling

Different popular neural networks have been tried to generate the feature space: mobilenet_v2, inceptionV3, ...

The best model tried is **ResNet101**, the whole model is used except the feedforward part.

Each instance have 2048 features.

In order to get a fast response on the features space a **KDTree** structure is implemented.



ResNet101

↓

Feature Extraction

↓

KDTree

Results

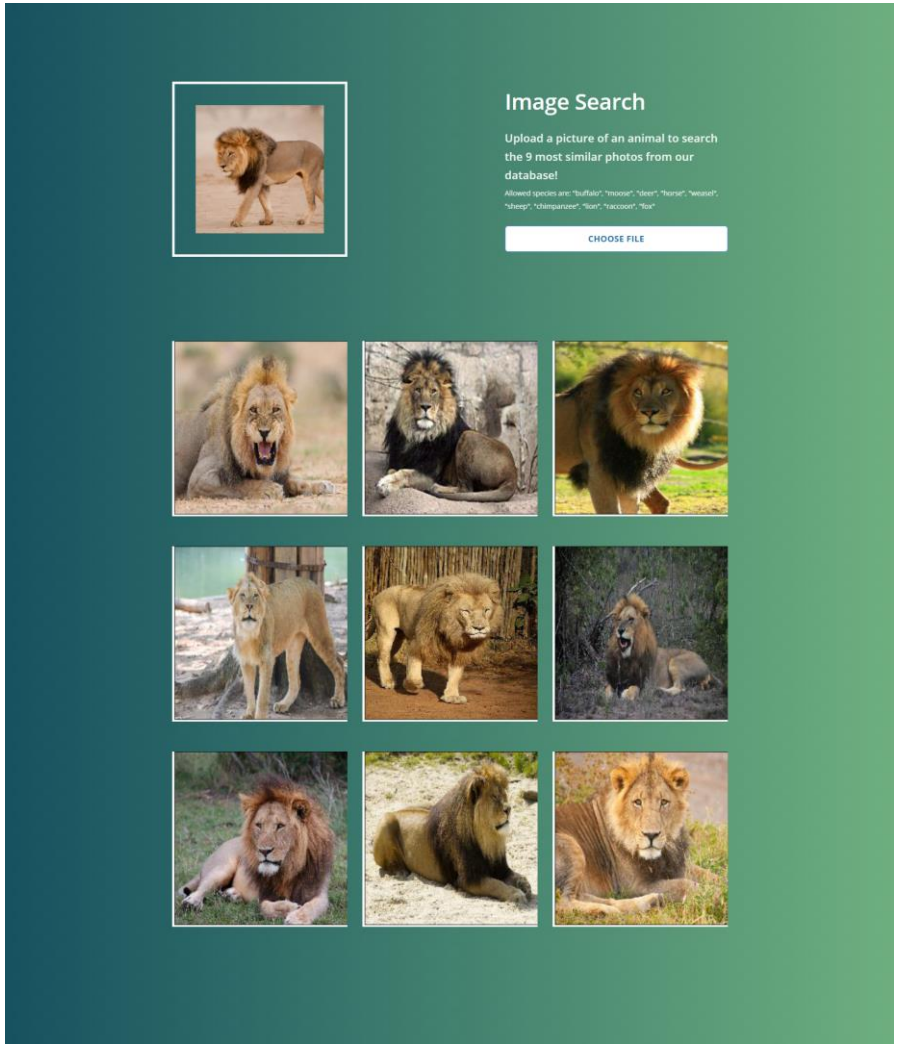


Image Search
Upload a picture of an animal to search the 9 most similar photos from our database!
Allowed species are: "buffalo", "moose", "deer", "horse", "weasel", "sheep", "chimpanzee", "lion", "raccoon", "fox"
CHOOSE FILE

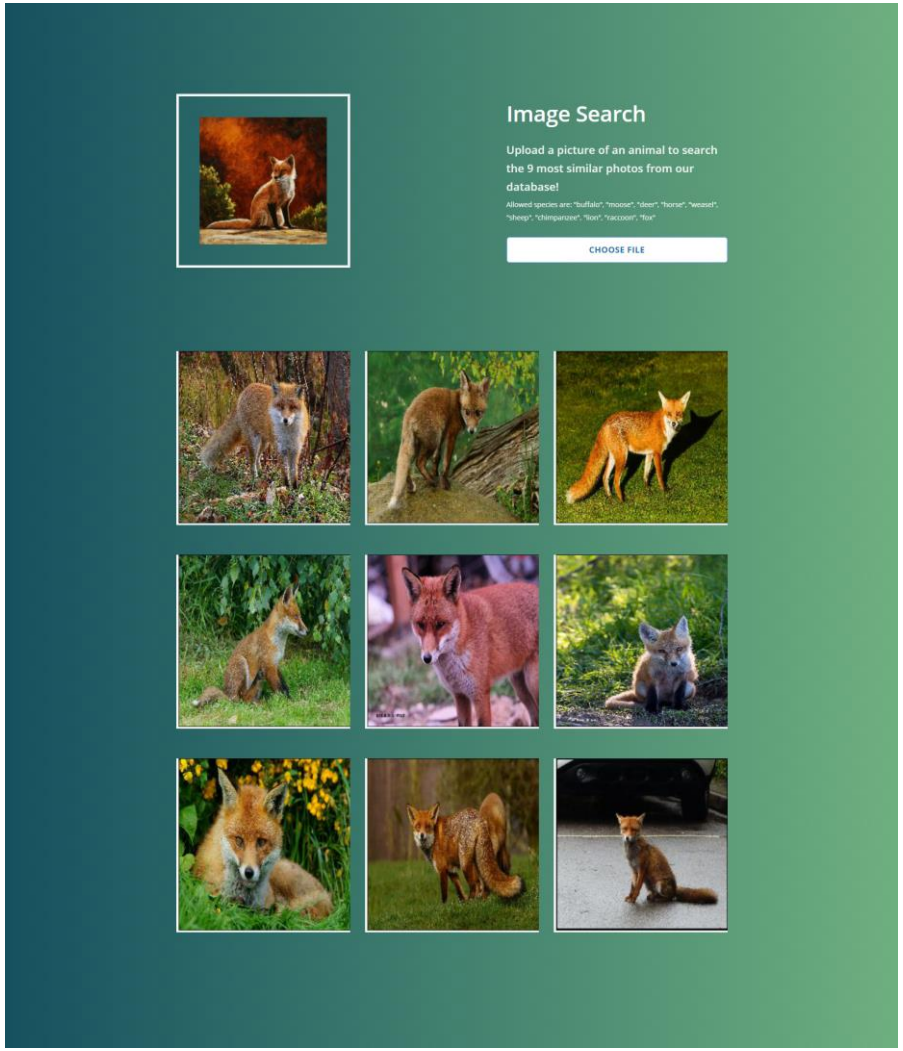
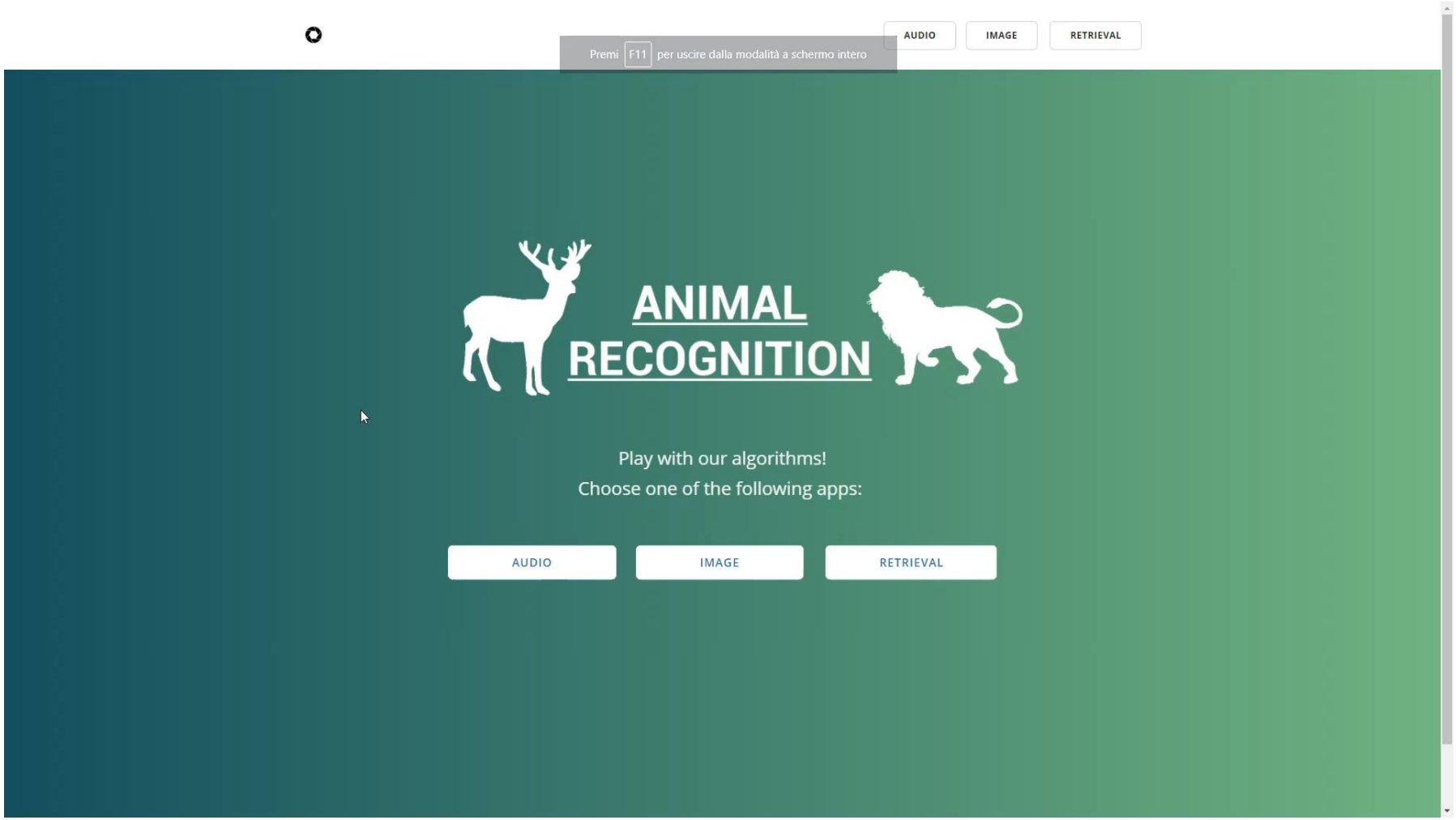


Image Search
Upload a picture of an animal to search the 9 most similar photos from our database!
Allowed species are: "buffalo", "moose", "deer", "horse", "weasel", "sheep", "chimpanzee", "lion", "raccoon", "fox"
CHOOSE FILE

DEMO

Demo


AUDIO
IMAGE
RETRIEVAL
Premi F11 per uscire dalla modalità a schermo intero
ANIMAL RECOGNITION
Play with our algorithms!
Choose one of the following apps:
AUDIO
IMAGE
RETRIEVAL

GRAZIE

Federico Luzzi - 816753

Federico De Servi - 812166