Image Super-Resolution

Digital Signal and Image Management Project
University of Milano-Bicocca

Matteo Breganni 869549
Francesco Cavallini 920835

Dataset Description

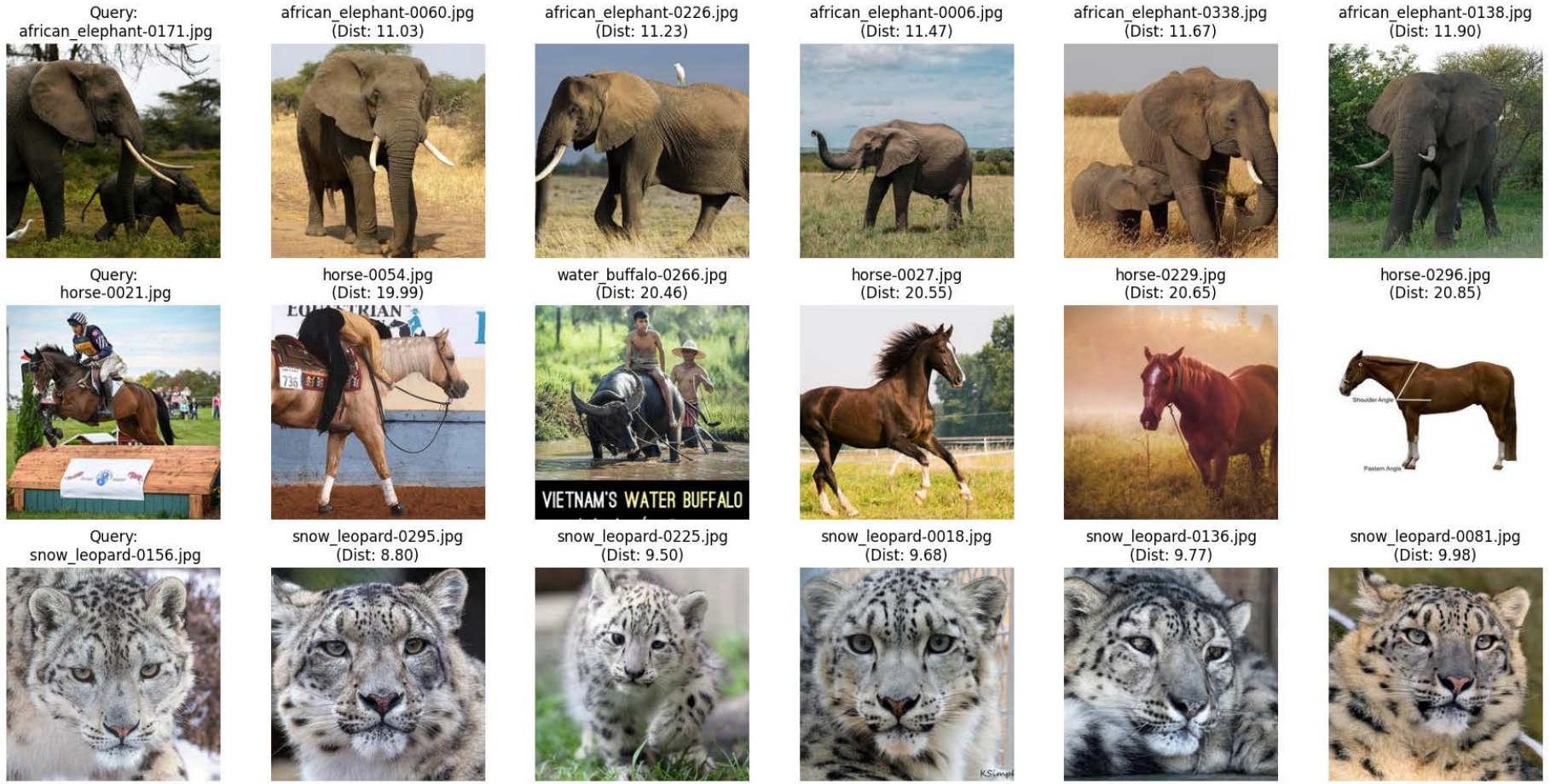
- **Mammals dataset** [1]
- **45 Categories**, 13751 images
- 60-20-20 Train/Val/Test split, category specific
- Index-Category dictionaries saved for later use

[1] https://www.kaggle.com/datasets/asaniczka/mammals-image-classification-dataset-45-animals

Pre-Trained CNN Approach

Pre-Trained CNN

- **Features extraction** with MobileNetV2
- Train features to build the **KD-Tree**
- Example query with k=5:



Pre-Trained CNN: Evaluation

- Results evaluated with k=3
- **Accuracy**: 86,20%

- **ANMRR (Average Normalized Modified Retrieval Rank)**: 0.39

- Dataframe created to better nagivate the classification report's results
  - Sorted by F1-Score:

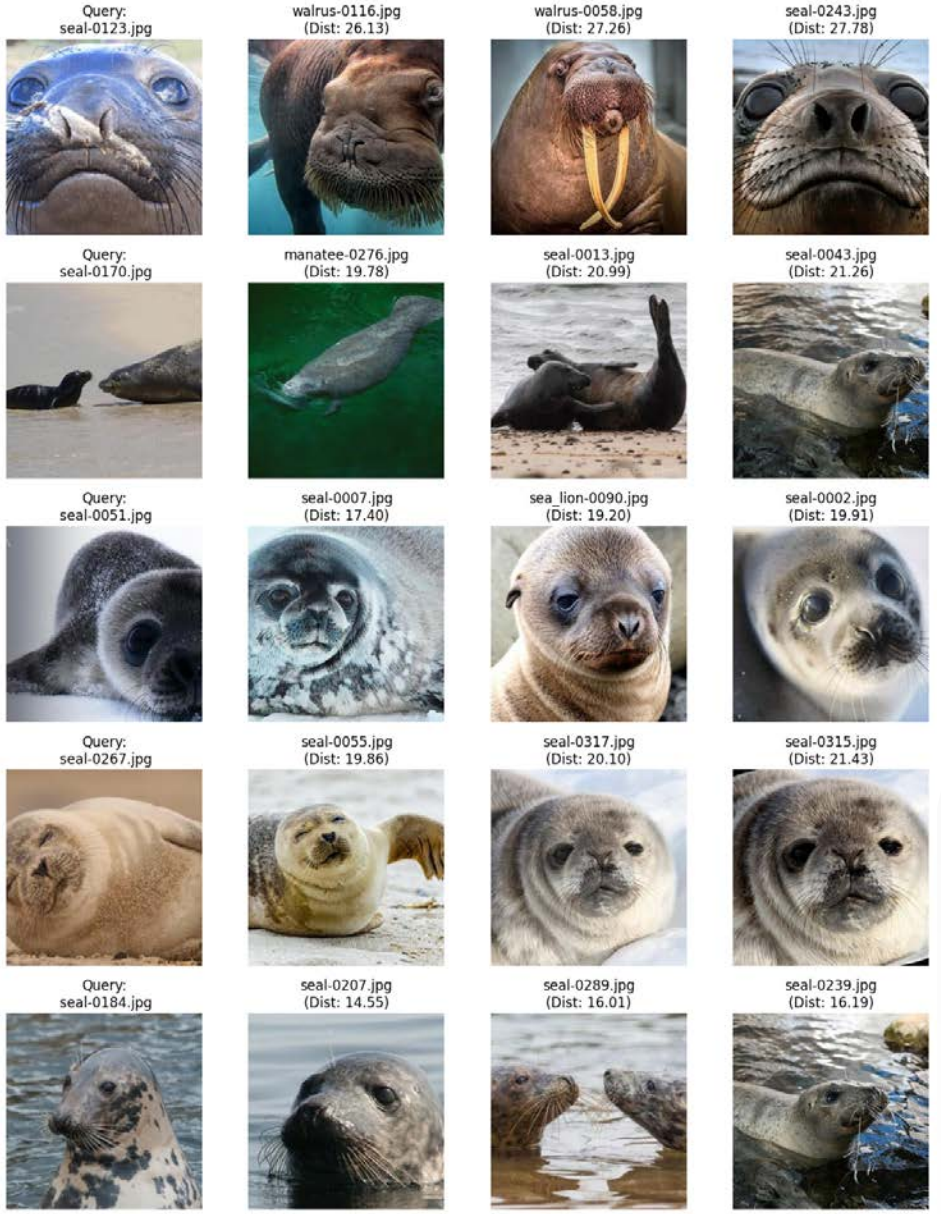
Class	Precision	Recall	F1-Score
orangutan	0.98	0.99	0.99
red_panda	0.99	0.99	0.99
snow_leopard	0.98	0.98	0.98
porcupine	0.97	0.95	0.96
armadillo	0.95	0.96	0.96

Class	Precision	Recall	F1-Score
seal	0.58	0.52	0.55
sea_lion	0.66	0.61	0.64
yak	0.69	0.70	0.70
vicuna	0.68	0.73	0.70
walrus	0.79	0.66	0.72

Pre-Trained CNN: Evaluation

- Testing the **worst class** (seal)
  - **F1-Score**: 0.55 (overall accuracy was 86,20%)
  - **Class-specific ANMRR** (5 test images, 3 retrieved images each): 0.47 (was 0.39)

- Often confused with similar animals like:
  - Walrus
  - Mantee
  - Sea lion

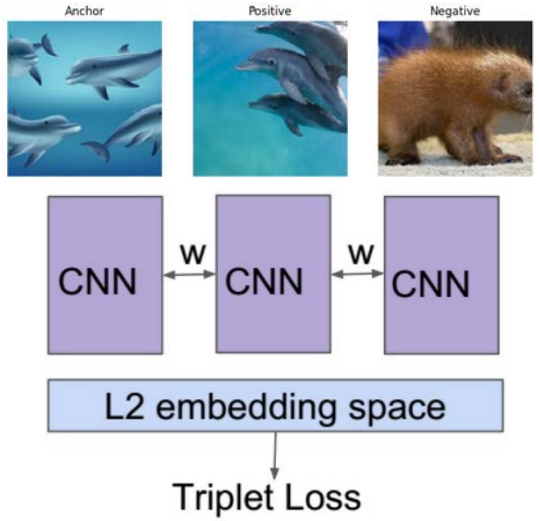


Siamese Network Approach

Siamese Network

- The Siamese Network requires:
  - **Encoder model** (CNN shared between the images that extracts their embedding)

Layer Type	Output Shape	Param #
Input	(244, 244, 3)	0
Conv, 64, 3x3	(244, 244, 64)	1792
Batch normalization + max pooling 2x2	(112, 112, 64)	256
Conv, 128, 3x3 + max pooling 2x2	(56, 56, 128)	73,856
Conv, 256, 3x3 + max pooling 2x2	(28, 28, 256)	295,168
Global Average Pooling	(256)	0
Dense, 128	(128)	32,896

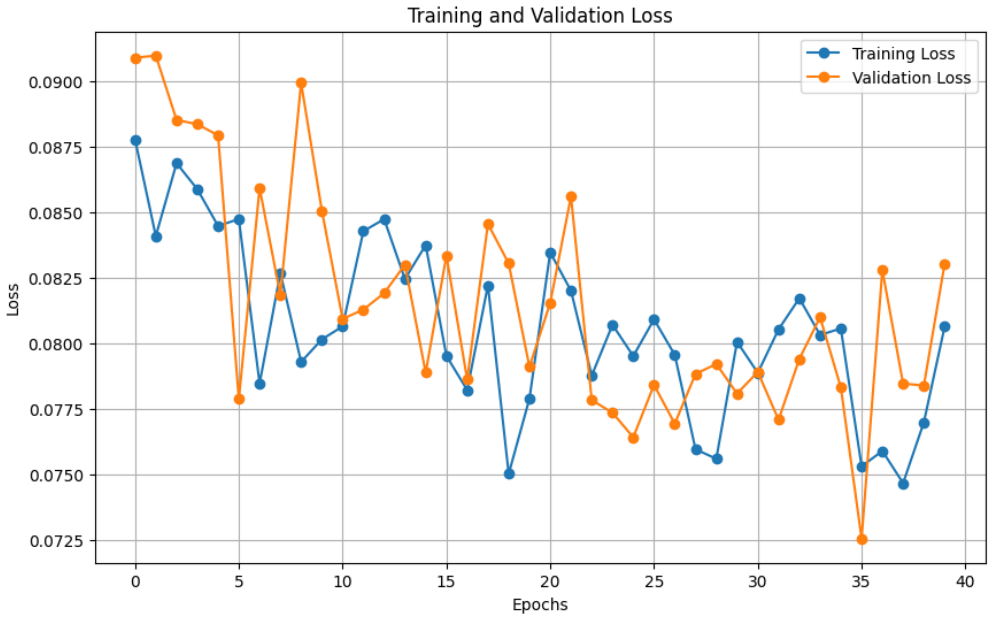


- **Triplet loss** that evaluates the results on:
  - **Anchor** (reference image)
  - **Positive** (image from the same class as the anchor)
  - **Negative** (image from a different class)

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

Siamese Network: Training Results

- **Terrible training results**
  - Barely any learning



- **Network too small**
  - Bigger network would be harder and more expensive to train
- **Output features** of the encoder are **too few** (128) for the 45 categories

Siamese Network: Evaluation

- Terrible peformance on the KD-Tree:
  - **Accuracy**: 12,14%
  - ANMRR: 0.88

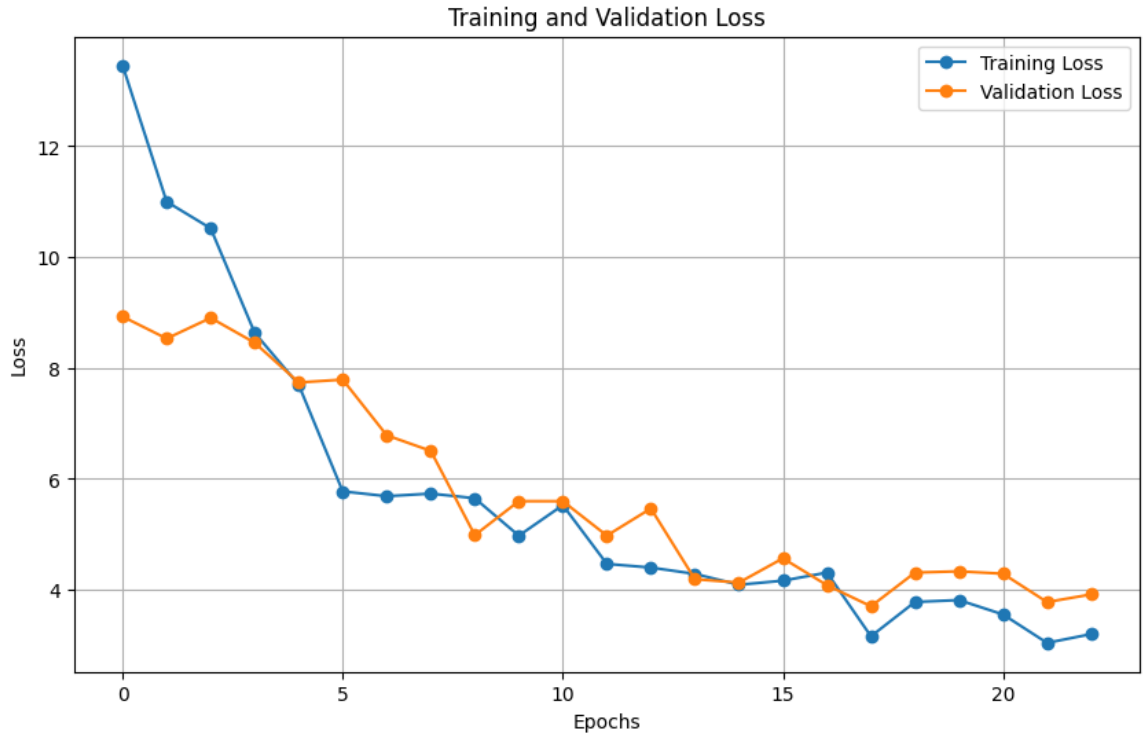
Class	Precision	Recall	F1-Score
mantee	0.44	0.45	0.44
blue_whale	0.43	0.38	0.40
zebra	0.50	0.30	0.37
polar_bear	0.34	0.30	0.32
dolphin	0.28	0.33	0.30

Class	Precision	Recall	F1-Score
Mongoose	0.03	0.03	0.03
Rhinoceros	0.03	0.04	0.03
Opossum	0.04	0.05	0.04
Yak	0.05	0.05	0.05
Squirrel	0.05	0.05	0.05

- Even the **best classes** don’t have good performance
- The **worst classes** are barely ever retrieved

Fine-Tuned Siamese Network

- Same structure of the Siamese Network (**triplet loss**)
- **Encoder network: MobileNetV2** with no freeze or added layer
  - **Fine-tuning** instead of training a bit model from scratch to have a «hot start»



- Much better learning process

FT Siamese Network: Evaluation

- Performance slightly lower than the first method:
  - **Accuracy**: 83,80% (was 86,20%)
  - **ANMRR**: 0.40 (was 0.39)

Class	Precision	Recall	F1-Score
red_panda	0.99	1	0.99
snow_leopard	0.98	0.97	0.98
porcupine	0.98	0.94	0.96
orangutan	0.96	0.96	0.96
zebra	0.95	0.96	0.95

Class	Precision	Recall	F1-Score
seal	0.52	0.57	0.54
sea_lion	0.63	0.51	0.56
yak	0.70	0.69	0.70
blue_whale	0.66	0.77	0.71
alpaca	0.76	0.67	0.71

- Very similar on the top classes
- Slightly different on the worst classes

Comparison and Combination

Comparing the two best methods

- Comparison dataframe created
  - F1-Score delta
    - Sorted by **delta**

Class	F1-Score1	F1-Score2	Delta
alpaca	0.84	0.71	0.13
camel	0.88	0.76	0.12
tapir	0.86	0.77	0.09
horse	0.84	0.77	0.07
sea_lion	0.64	0.57	0.07

Class	F1-Score1	F1-Score2	Delta
highland_cattle	0.82	0.87	-0.05
walrus	0.72	0.74	-0.02
vampire_bat	0.80	0.82	-0.02
vicuna	0.70	0.72	-0.02
african_elephant	0.93	0.94	-0.02

- The delta almost always **favors the first method**
- **Few classes favor the second method** slightly

Combining the two best methods

- Attempt to improve the performance by **combining the two best methods**
  - **Normalize** each array of feature arrays between 0 and 1
  - **Concatenate** each corresponding feature array (now twice the size)
  - Define a new KD-Tree with the new feature arrays

- Test the KD-Tree with k=3
  - **Accuracy**: 87.09% <-- was 86,20% and 83,80%
  - **ANMRR**: 0.39 <-- was 0.39 and 0.40
    - Slight performance increase

Class	Precision	Recall	F1-Score
red_panda	0.99	0.99	0.99
orangutan	0.98	0.99	0.99
snow_leopard	0.99	0.98	0.98
porcupine	0.97	0.96	0.97
koala	0.95	0.97	0.96

Class	Precision	Recall	F1-Score
seal	0.58	0.58	0.58
sea_lion	0.69	0.60	0.64
yak	0.73	0.71	0.72
walrus	0.80	0.70	0.75
water_buffalo	0.77	0.73	0.76

First method vs Combined

- **First method** results:

Class	Precision	Recall	F1-Score
orangutan	0.98	0.99	0.99
red_panda	0.99	0.99	0.99
snow_leopard	0.98	0.98	0.98
porcupine	0.97	0.95	0.96
armadillo	0.95	0.96	0.96

Class	Precision	Recall	F1-Score
seal	0.58	0.52	**0.55**
sea_lion	0.66	0.61	**0.64**
yak	0.69	0.70	**0.70**
vicuna	0.68	0.73	**0.70**
walrus	0.79	0.66	**0.72**

- **Combined** method results:

Class	Precision	Recall	F1-Score
red_panda	0.99	0.99	0.99
orangutan	0.98	0.99	0.99
snow_leopard	0.99	0.98	0.98
porcupine	0.97	0.96	0.97
koala	0.95	0.97	0.96

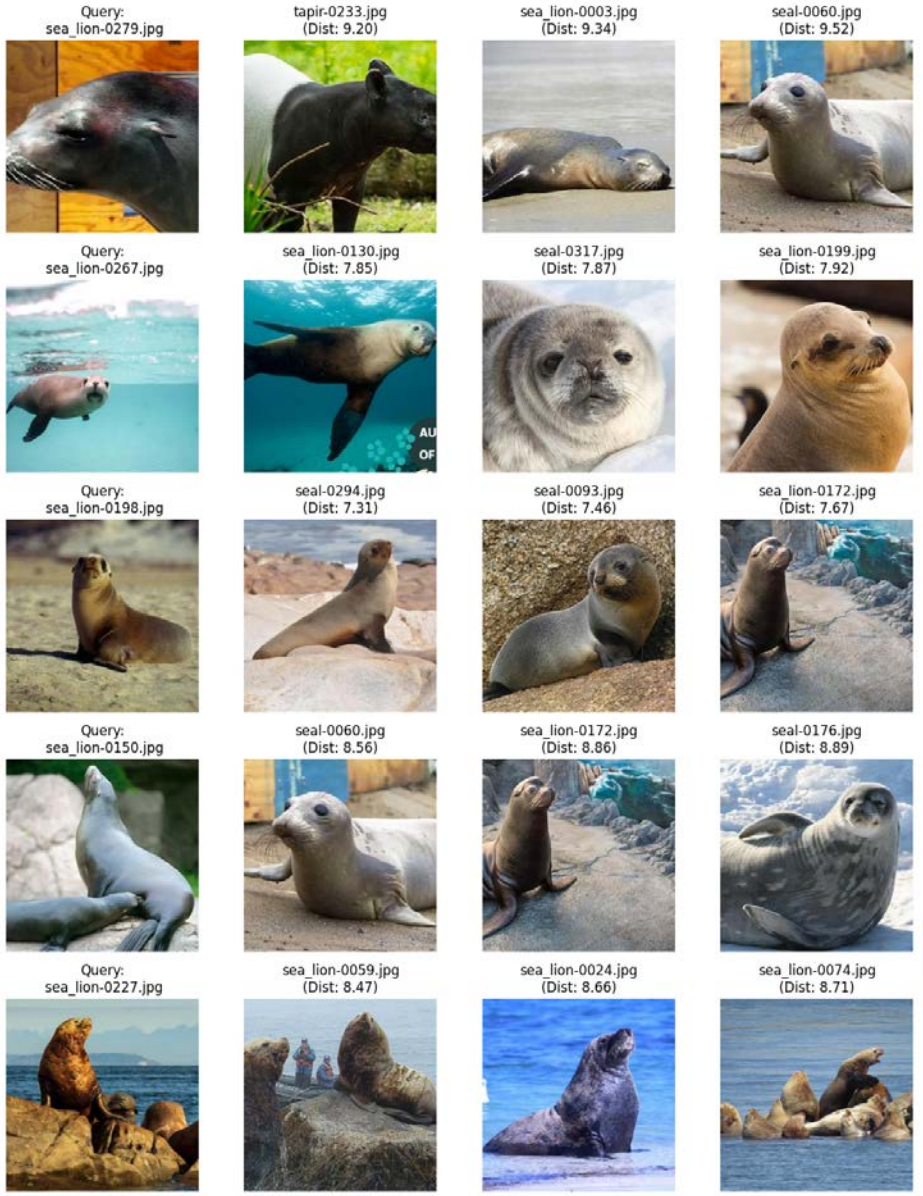
Class	Precision	Recall	F1-Score
seal	0.58	0.58	**0.58**
sea_lion	0.69	0.60	**0.64**
yak	0.73	0.71	**0.72**
walrus	0.80	0.70	**0.75**
water_buffalo	0.77	0.73	**0.76**

- The main difference is the slight increase in performance for the **worst classes**

Evaluating the Combination

- Testing the **worst class** (seal)
  - **F1-Score**: 0.58 (worst class was 0.55)
  - **Class-specific ANMRR** (5 test images, 3 retrieved images each): 0.44 (was 0.47)

- Still **struggling** because of the high similarity with other classes like *sea_lion*
  - But **better performance** than before



Relevance Feedback

Relevance Feedback

- **Relevance feedback** to improve the performance of the worst classes
- Returning 10 images from the class «*sea_lion*»:

- Only 4 of them are from the correct class

- Manually flag each **relevant** (0, 1, 4, 9) and **irrelavant** (2, 3, 5, 6, 7, 8) image
- Use the **Rocchio Algorithm** to calculate the new query:
  - Query = alpha * original_query + beta * relevant_mean – gamma * irrelevant_mean
  - with alpha=1, beta=0.75, gamma=0.15

- The new query returns all correct images

Thanks for your attention