Deep Learning
(MO434A/MC934B)

Prof. Alexandre Xavier Falcão

Project of the Course in May 7, 2023

1 Scope of the project

This project involves the following tasks using the available fish dataset in `https://www.kaggle.com/datasets/crowww/a-large-scale-fish-dataset`, which contains images of nine classes of fishes (BlackSeaSprat: 000, GiltHeadBream: 001, HourseMackerel: 002, RedMullet: 003, RedSeaBream: 004, SeaBass: 005, Shrimp: 006, StripedRedMullet: 007, and Trout: 008). The project consists of preparing notebooks to execute and assess the results of the tasks below. The description of the work and the discussion of its results, with illustrative figures and tables, should be presented in a pdf file. Provide the notebooks and pdf file of the report, which can also include notebooks and discussion about other tasks presented as exercises in the notebooks of the lectures. The notebooks of the project can be prepared based on combinations and adaptations of routines from pytorch-convnet.ipynb, visualize-pytorch-model-outputs.ipynb, and few-shotlearning-siameseNN.ipynb available for the lectures.

1. Preparation of two versions of the fish dataset for the experiments.

2. Construction and evaluation of a Convolutional Neural Network (CNN) that classifies the fish images into one of the nine classes. It will use the first version of the dataset.

3. Substitution of the above CNN's backbone by the backbone of one Deep Neural Network (DNN), pretrained on the ImageNet dataset, and comparison with the performance of your CNN on the same test sets. For comparison, evaluate fine-tuning the DNN and training it from scratch. It will use the first version of the dataset.

4. For each class, use Grad-CAM (https://arxiv.org/pdf/1610.02391.pdf) to evaluate which parts of the fish are the most relevant for correct classification. It will use the first version of the dataset.

5. Pretrain the backbone and the hidden linear layer(s) of your CNN using **supervised contrastive learning** for weight initialization before training your CNN for nine classes, with and without freezing the pretrained weights, and compare the results with the ones in tasks 2 and 3. Verify if the strategy allows a reduction in the number of training images. It will use the second version of the dataset.

6. Evaluate the impact of the above strategy in visual class separation when projecting features of the last hidden layer of the CNN after tasks 2 and 5.

2 Preparation of two versions of the fish dataset.

Rename the original image files to 00X_YYYYY.png, where X is the class in $0, 1, \ldots, 8$ and YYYYY is the number of the image in the class. By reading the file names in the notebook, split the dataset into training, validation, and test sets for the experiments, which will constitute the first version of the dataset. For the second version of the dataset, create pairs of images from the same class and image pairs from distinct classes.

This task can also be created from the first version's training, validation, and test sets, generating a second version of training, validation, and test sets for pairwise comparison. While the target values in the first version are the class numbers, X, the target values in the second version are either 0 (same class) or 1 (distinct classes).

3 Construction and evaluation of a CNN.

To classify the nine classes, build a CNN with a few convolutional blocks and linear layers. Describe all evaluated architectures and their results for several splits of training, validation, and test sets, reporting mean and standard deviation of Cohen's Kappa and accuracy per class.

4 Substitution of the CNN's backbone by one of a DNN.

Replace the backbone of the above CNN with one of a DNN. Select the best one among at least three popular DNNs for the fish classification problem. Report mean and standard deviation of Cohen's Kappa and accuracy per class for the compared ones. Compare the best DNN with the best CNN model above.

5 Which parts of the fishes are the most discriminative?

This dataset contains the fish masks, which can be explored to eliminate activations outside the object. However, only do that if it is necessary. The idea is to verify which image regions are the most important for class separation at the output of the last convolutional block (feature extractor). The attention regions are expected to be on the discriminative parts. It can be verified by visualizing those regions in several images of the same class, to which classification is correct. Please do it for all classes and add the analysis to the final report. After that, add some marker (e.g., a disk) in the background of most images from a given class and verify if the retrained network changes attention to that marker when clasifying images of that class.

6 Can the training set be reduced via contrastive learning?

Siamese networks trained by supervised contrastive learning (http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf) can learn when pairs of images are from the same or distinct classes. The model (Figure 1) can be trained from a reasonable number of image pairs, created with a few images from all classes (as shown in few-shotlearning-siameseNN.ipynb). Use this idea with convolutional blocks and hidden layers of the CNN to pre-train the model with fewer training images. Try different contrastive loss functions (e.g., triple loss as in https://arxiv.org/abs/1503.03832) that you will find in the literature and report the best model. After that, use the pretrained layers in the best CNN model and retrain it for fish classification. Evaluate the pretrained layers with frozen and unfrozen weights during the training process. Compare the results of training the CNN from scratch with the one pretrained with contrastive learning for smaller training sets.

7 Can such contrastive learning improve visual class separation?

Since linear layers reduce dimensionality, favoring projection methods, evaluate the role of supervised contrastive learning in this process by comparing the feature projection at the last hidden layer of the CNN models in Section 6. Assuming that contrastive learning can reduce the training set size, it should be possible to project labeled and unlabeled samples of a problem, visualize the data distribution with partial information of class separation, and easily annotate the unsupervised samples in the projection. In this recent work, for instance, https://www.sciencedirect.com/science/article/abs/pii/S0031320323003503, even without contrastive learning, we show that a similar idea is feasible to create pseudo annotations for training DNNs automatically.

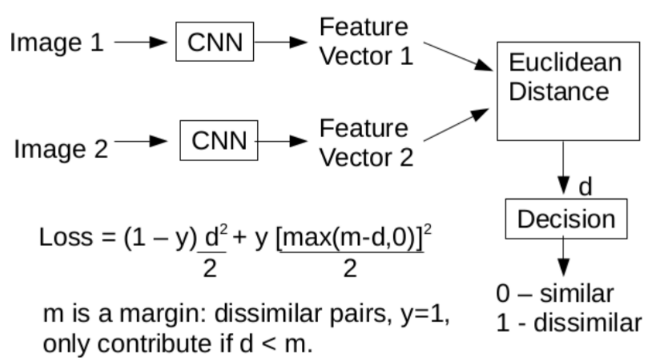

Figure 1: A Siamese network for image comparison. CNN consists of a sequence of convolutional blocks followed by flattening and linear layers for dimensionality reduction. The figure also presents the contrastive loss used in few-shotlearning-siameseNN.ipynb.