Pytorch Project
Plankton Classification

Deep Learning UniPD

Alessio Cocco 2087635
Andrea Valentinuzzi 2090451
Giovanni Brejc 2096046

August 2023

Abstract

The goal of this project was to improve plankton classification through image preprocessing. We computed different sets of images, each filtered by a different transformation to highlight a specific feature. We obtained local features using canny edge, global features using the sobel operator and texture features using the gabor filter. We trained a classifier for every feature and combined the results with an ensemble.

Training

A pretrained Alexnet was used as the base model, from which we replaced the original classifier with a fullyconnected layer with 4096 input and "# of classes" in output. Training was done using batch SGDM as optimization function with the cross entropy loss. The dataset is composed of 3771 plankton images, which have been divided using the cross-validation technique with 2 folds, splitting the dataset in two halves.

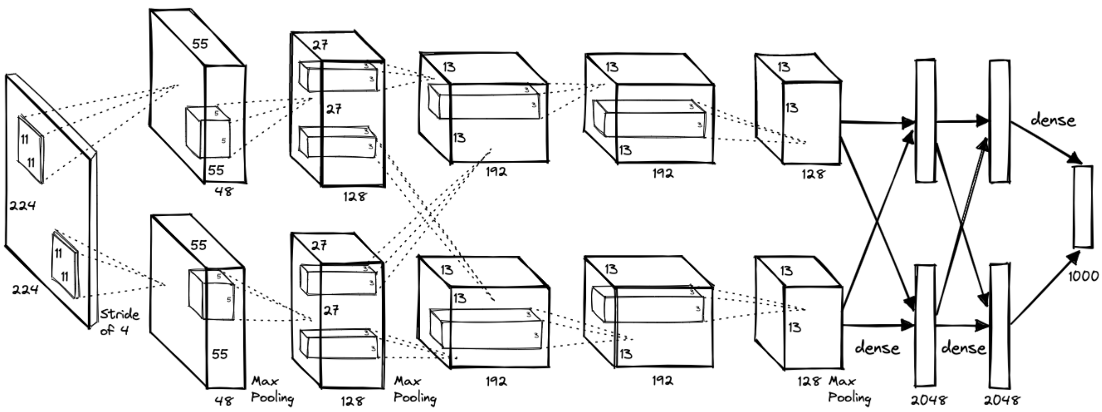


Figure 1: AlexNet architecture

Preprocessing

We implemented in python 4 preprocessing function (each preprocessing include an initial resize):

- original_features
- global_features:
  1. bilateral filter
  2. sobel
  3. equalization
  4. percentile
  5. intensity rescaling
- local_features:
  1. grayscale
  2. otsu thresholding
  3. canny edge
- gabor_features:
  1. grayscale
  2. gabor (real)
  3. normalization

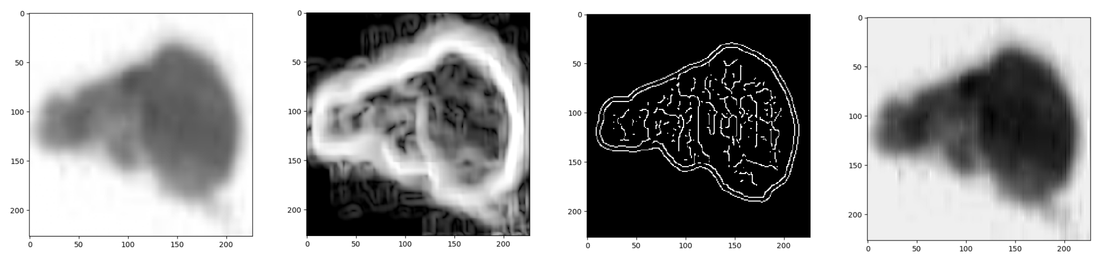


Figure 2: From left: Original, Global, Local, Gabor

Ensemble

We tried 3 types of ensemble, one using the majority voting rule and two with the sum rule.
The first one computes the most voted label and in case of a tie, it chooses the label that obtained the highest score between all classifiers.
The second one is a classical weighted sum rule. By using the classifiers accuracy as weight, the predicted label is the the class with the highest summed score.
Since we noticed that the last classifier is the strongest one, first a score is computed from the mean of the first three classifier and then a label is chosen by summing the mean with the last classifier predictions. The label with higher score is the predicted one.

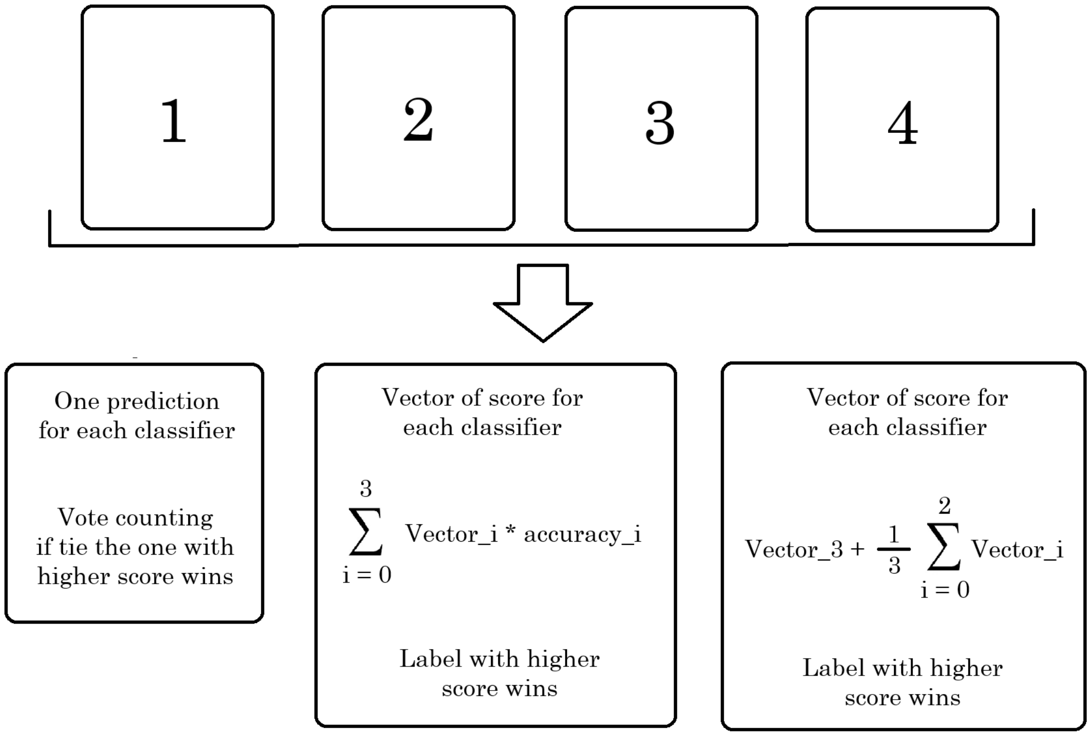
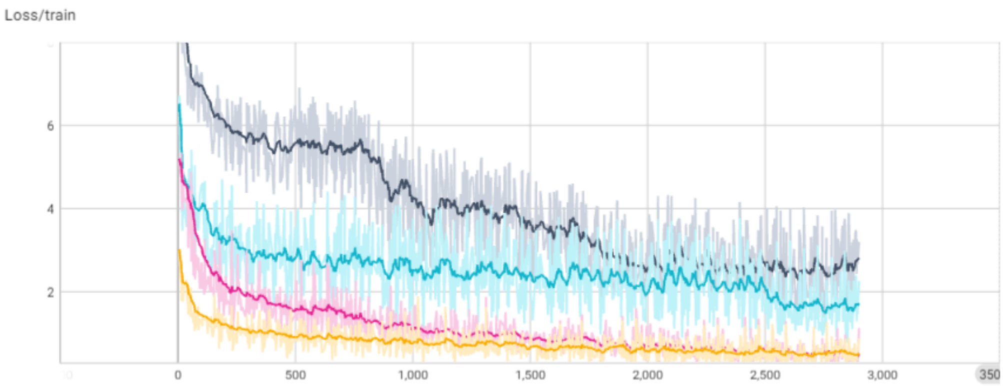


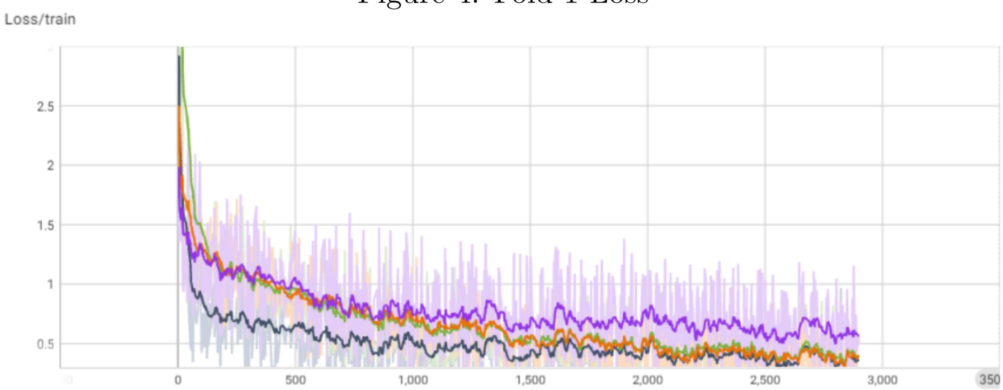
Figure 3: Ensemble

Results



	Run	Smoothed	Value	Step	Relative
●	runs\Fold1_Gabor	0.5143	0.8697	2,900	55.32 sec
●	runs\Fold1_Global	1.7219	2.2587	2,900	1.077 min
●	runs\Fold1_Local	0.505	1.127	2,900	1.068 min
●	runs\Fold1_Original	2.8085	3.1967	2,900	1.087 min

Figure 4: Fold 1 Loss



	Run	Smoothed	Value	Step	Relative
●	runs\Fold2_Gabor	0.3602	0.3757	2,900	56.61 sec
●	runs\Fold2_Global	0.3856	0.4107	2,900	1.072 min
●	runs\Fold2_Local	0.3979	0.5043	2,900	1.071 min
●	runs\Fold2_Original	0.5667	0.5607	2,900	1.09 min

Figure 5: Fold 2 Loss

Single Classifier Accuracy

Accuracy	C_1	C_2	C_3	C_4
Fold 1	62.46	65.59	63.94	75.50
Fold 2	80.65	75.66	77.09	81.12

Ensemble Accuracy

Accuracy	E_1	E_2	E_3
Fold 1	58.54	53.5	62.09
Fold 2	77.44	71.85	77.52

Conclusion

Ensemble results were not able to improve the overall accuracy obtained from the different single feature classificators. However training using different features has shown to be able to improve the models accuracy and could be a better solution rather than using the original images. Especially the gabor filter has given the best results overall. For future improvements, a better usage of the gabor filter could be implemented. For now we have only used the real component of the filter, while there still is the imaginary part to take into account that could improve the model even further.