Fall 2024: CSI5341
Assignment 2

Due: 23:00 on October 21st, 2024

University of Ottawa - Université d'Ottawa

Jochen Lang

1 Texture Image Comparison

This assignment compares a multilayer perceptron with a simple LeNet style CNN architecture on a binary classification task. You are given a maximum budget of 250,000 parameters. A goal assignment is to achieve reasonable results with both architectues. The supplied images show two types of stoneflies and are available at `https://web.engr.oregonstate.edu/~tgd/bugid/stonefly9/` [1, 2]. For this assignment, we are only using the types **Yor**aperla sp. with 483 images and **Cal**ineuria californica with 459 images which are also directly available on Brightspace. Note that the color images are fairly large at 1280×960 for feeding into a neural network and you will have to decide on a smaller resolution that you want to work with. You won't have to use the same resolution for both architectures. Image size is one of your optimization choices.

1.1 Getting Started

You will need to download the two subsets from BrightSpace or from Oregon State University. Unpack the images in a directory relative to your jupyter notebook called `stoneflies`. We will be marking your notebook with the data installed in `stoneflies/yor` and `stoneflies/cal`, and your notebook will have to work with the images at these locations. Use the **first 50 images** for each category as your test set. Do not rename images, directories or reorganize the data. You will loose marks if your notebook does not work with images at the expected locations.

1.2 Image Preprocessing [2]

You need to write a python function that loads images and preprocesses them as you desire for your network. Your notebook must load the original images and modify them (you may want to save them for efficiency) but it is not acceptable to process manually or with another tool. You must use only scikit-image functions (or functions that you implement from scratch).

1.3 Multi-Layer Perceptron [3]

Build a multilayer perceptron model to classify an image by using the pre-processed images. For this part of the assignment, you must build and train the Multi-layer perceptron model with the Keras API of Tensorflow. Make sure to print a listing of your model in the Jupyter notebook. Do not use more than a maximum of 250,000 parameters in total in your MLP. Try to obtain good classifier performance. Print your learning curves for training (and validation if you have used a validation set). Give the confusion matrix of your network on the training and testing data sets.

1.4 CNN [3]

Build a CNN model to classify an image by using the pre-processed images (your input images can be different than in Question 1.3). For this part of the assignment, you must build and train the CNN model with the Keras API of Tensorflow. Make sure to print a listing of your model in the Jupyter notebook. Do not use more than a maximum of 250,000 parameters in total in your CNN. Try to obtain good classifier performance. Print your learning curves for training (and validation if you have used a validation set). Give the confusion matrix of your network on the training and testing data sets.

1.5 Comparison MLP and CNN [2]

Evaluate both of your methods from Sections 1.3 and 1.4 on training and test set by analyzing the accuracy, recall and precision. Visualize some of miss-classified samples. Compare the results and discuss the differences and their causes. Consider classifier performance but also other criteria, e.g., training effort and prediction speed. Your brief discussion based on quantifiable criteria need to be contained in your Jupyter notebook.

2 Submission

You will need to submit your solution in a Jupyter file, do *not* submit the image data. Make sure you have run all the cells. All text must be embedded in the Jupyter file, I will not look at separately submitted text files. If your Jupyter file needs a local python file to run, please submit it as well. Assignment submission is only though Virtual Campus by the deadline. No late submissions are allowed, you can submit multiple times but only your last submission is kept and marked.

References

[1] D. A. Lytle, G. Martínez-Muñoz, W. Zhang, N. Larios, L. Shapiro, R. Paasch, A. Moldenke, E. N. Mortensen, S. Todorovic, and T. G. Dietterich, "Automated processing and identification of benthic invertebrate samples," *Journal of the North American Benthological Society*, vol. 29, no. 3, pp. 867–874, 2010.

[2] G. Martinez-Munoz, N. Larios, E. Mortensen, W. Zhang, A. Yamamuro, R. Paasch, N. Payet, D. Lytle, L. Shapiro, S. Todorovic, *et al.*, "Dictionary-free categorization of very similar objects via stacked evidence trees," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 549–556, IEEE, 2009.