Image Processing for Remote Sensing
Homework III

1. Introduction

The goal of this homework is to use a CNN architecture for supervised scene classification on multi-label remote sensing images. During experiments different data augmentation techniques have to be investigated.

2. Dataset

For this homework, you will be working with the multi-label version of the UCMerced dataset introduced in the practical lecture 7. In order to be able to train on a multi-label version of it, you need to implement the instance method *read_multilabels(self)* of the UCMerced Dataset class defined in the notebook from Lab 7. Following the introduced procedure of downloading and extracting the ZIP file for the dataset, two identical multi-label files in .xlsx and .txt format can be found under "./data/UCMerced_LandUse/multilabel". For convenience we recommend to parse the .xlsx file by using the function *read_excel()* from the pandas package. The ordering of the data samples equals the internal ordering of the dataset class stored under the attribute *self.img_paths.* In the report, provide a statistical analysis of the class distribution of the multi-label version.

3. Training Design

Before training, split the dataset into three non-overlapping sets: training (70%), validation (10%) and test (20%). Due to multiple labels per sample, a class balancing split is not possible, a random selection of data points is sufficient. Apply Transfer Learning as learned in the practical lecture 7 and choose a pre-trained ResNet18 as the model architecture. Be aware that also the model has to be adapted to the new label set. In order to be able to train the model on multiple labels, the loss function has to replaced by a multi-label loss function (e.g. binary cross entropy, see https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html).

4. Experiments

The empirical objective of this homework is to analyze the effect of different data augmentation strategies. As explained during the practical lecture, you can add them to the transformations applied to the training data. Compare the results for:

- A baseline that does not contain any data augmentation
- A method that includes random affine transformation (e.g. degrees=10) and random auto-contrast
- A method that consists of two or three augmentation methods that you select on your own
- A method that includes the composed data augmentation technique RandAugment (https://arxiv.org/abs/1909.13719)

For more details on data augmentation methods in PyTorch see https://pytorch.org/vision/main/transforms.html

5. Metric and Evaluation

Since in multi-label classification the prediction of a class is non-exclusive, different metrics are applied to evaluate these tasks. The most descriptive choices are f1-score micro, f1-macro, mAP micro and mAP-macro. The f1-scores will be part of the multi-label classification report, the mAP scores have to be added manually (see https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html#sklearn.metrics.average_precision_score). Be aware that the prediction logic changes when predicting on multiple labels. Instead of transforming the class outputs into probabilities that sum to one by using a softmax function, each class output has to be mapped to a probability individually and thresholded by 0.5 to eventually decide on the final prediction of each class. After observing the final results of different data augmentation techniques, add a small note to the report, why do you think some of the data augmentation techniques worked better or worse. Support the presentation of your results by plotting meaningful graphs, confusions matrix as well as sample images for the augmented data.

Hints

- [1] When retraining a model, the optimizer has to be re-initialized, too, as it depends on the model parameter

- [2] The batch evaluation in *train_epoch()* and *val_epoch()* written to the tqdm progress bar can be omitted for multi-label classification since its computation is more complex than a simple accuracy score.

Submission (Deadline: 18/07/2022, 23:59)

It is essential that each homework should be done individually. Each student should submit the following materials to the ISIS system:

(1) One PDF report (around 3-4 pages) in which you should explain all steps to obtain final results, and explain your observations. Basically, the report should include a description of all the steps and parameters that you select for training (e.g., learning rate, number of epochs, the size of mini-batch), the results including example images with its prediction and tabular metric comparison obtained on the test and validation sets. This file should be named as IP4RS2022-HW3_*{your-name}*_report.pdf

(2) One zip file containing the corresponding codes, which is named with IP4RS2022-HW3_*{your-name}*_code.zip.