Fall 2023: ELG7186
Assignment 3

Due: 23:00 on October 26th, 2023 in Virtual Campus

University of Ottawa - Université d'Ottawa

Jochen Lang

1 CNNs, Keras API and Regression

This assignment will give you a chance to familiarize yourself with CNNs and with the different techniques for monitoring and controlling the training process in tensorflow. In this assignment, we will look at weed species, and the weed bounding box.

You must use Keras with the tensorflow backend, i.e., the package `tensorflow.keras`. For this assignment, you may use other tensorflow packages and scikit-learn, scikit-image or pandas but *not* other deep learning frameworks, e.g., pytorch, mxnet etc.

2 Data Preparation

The data for this assignment are plant images at different resolution captured with a variety of cameras. There are images (Figure 1) showing plants with different species. The images are part of a Plant Seedlings dataset by Giselsson et al. [1] which can be downloaded from the Aarhus University, Denmark

`https://vision.eng.au.dk/plant-seedlings-dataset/`

However, you must work with the subset of images posted on BrightSpace as `a3_dataset.zip`. There are 250 images for each of the 4 classes. As Figure 1 shows, there is a great variety of plants and image conditions. Organize your data into training, validation and test set appropriately.

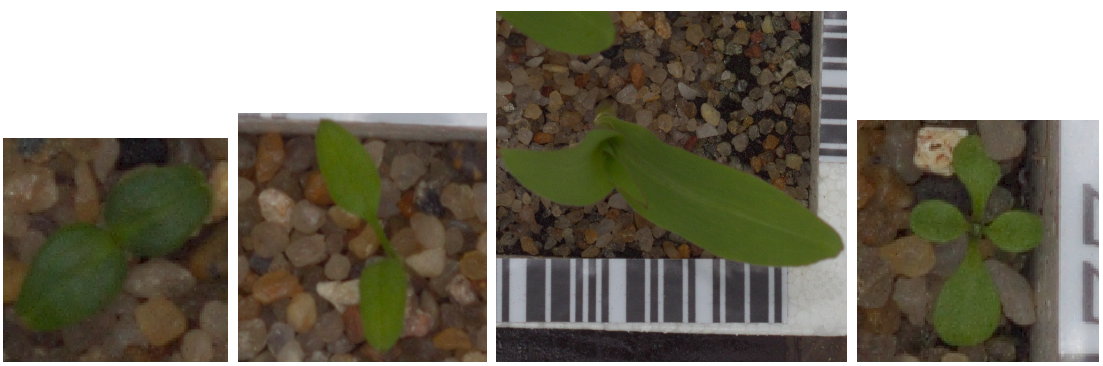

Figure 1: Examples of plant images for species *Cleavers*, *Common Chickweed*, *Maize*, *Shepherd's Purse*, respectively.

3 Basic Transfer Learning

For this assignment, you are asked to use the Keras implementation of VGG-16 as a starting point. Have a look at the transfer learning example jupyter notebook `mnistVGG.jypnb` to get started.

3.1 Classification Network [2]

Using the first 2 blocks of VGG-16 add extra Keras layers to create your own version of a CNN network for the classification of the images according to the number of leaves in the plant images. Note that there will be 4 classes. The last layer from VGG-16 will be `block2_pool` and you are allowed to add no more than *five* fully connected or convolutional layers to the network including the final output layer. You can use as many pooling, flattening, 1×1 convolution layers, etc. as you wish but do *not* use any regularization. Train this simple network on the training set while monitoring convergence on the validation set. As input to the model use images of size no larger than 128×128. Note, it is highly recommended to use even smaller input images to try things out. You are not expected to fine-tune the initial VGG layers, set the VGG layers untrainable. Print your learning curves for training and validation. Give the confusion matrix of your network on the training including validation and testing data sets.

3.2 Regression Network [2]

Figure 2: Examples of the bounding box for the weed. The red box represents the bounding box, the red numbers indicate the width and the height of the bounding box.

Using the same setup as in Question 3.1 but with a different head for a regression network to predict the height and width of the bounding box for the weed in the image. Figure 2 shows an example of the bounding box for the weed. The ground truth of the bounding box height and width is stored in the `bbox.json` file under the `a3_dataset.zip`. The `bbox.json` file contains a series of dictionary following the structure:

```
{
    class_name:{
```

```
        image_name:{
            height: int(h),
            width: int(w)
            }
        }
}
```

Normalize the height and width corresponding to the image size. Your network needs to output two single float value ranging between 0 to 1 for the height and width separately. Again, please print your learning curves for training and validation. You are not expected to fine-tune the initial VGG layers. Give your mean squared error on training including validation and testing data sets.

4 Improving the Model [1]

Incorporate two regularization methods (e.g., Batch Normalization, Dropout, Weight Normalization etc.) into your layers of the network. Please pick the model from Question 3.1 and Question 3.2 that performs better. You are not expected to fine-tune the initial VGG layers. Again, please print your learning curves for training and validation and print the corresponding metrics for your model.

5 Discussion [1]

The size of the training data is quite small. Discuss based on your learning curves if overfitting is occurring with your networks in Question 3.1 and Question 3.2. Discuss based on your learning curves and final metrics in Question 4, how large a improvement can be observed from regularization.

6 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] T. M. Giselsson, M. Dyrmann, R. N. Jørgensen, P. K. Jensen, and H. S. Midtiby, "A Public Image Database for Benchmark of Plant Seedling Classification Algorithms," *arXiv preprint*, 2017.