Assignment 4

Daniele Gilio

June 1, 2020

0.1 Introduction

This assignment is about image classification. We are going to build a classifier that is able to distinguish between 12 flower species. The dataset we will be working on contains 80 images per species in the training set and 20 per species in the test set.

0.2 Data Augmentation

Before passing to the actual classifiers description let us take a look at a very useful possibility in image recognition tasks. Working with images has a very positive element: *Data Augmentation*. Since most features in images are independent from a lot of transformations (rotations, color saturation, image exposure, translation, etc.) we can expand a dataset by applying these transformations to it. Most of the times the bigger the dataset the better the performance. That said we are going to present our results with both the basic dataset and the augmented one.

0.3 Low Level Classifier

The first approach we tested was using a classifier that used *Low Level Features*, which are specifically "crafted" functions that extract features from images. We tried using only one low level feature and then we tried using them all. With regards to the classifier we tested both a Single Layer Perceptron and a Multi Layer Perceptron. The first one had only the input layer, which dimensions matched the ones of the low level features, and the output layer made of 12 neurons. The MLP had the same input layer than the sequence $[512, 128, 32, 12]$, representing the width of each subsequent layer.

0.3.1 Single Feature

The Single Feature strategy obtained a very low accuracy score. We choose to use the *RGB Co-occurence Matrix* amongst all the possible ones because it was the best performing one. As we can see in Table 1 the MLP got a much higher score as expected. Note that the augmented dataset increases considerably the performance of the networks. Both networks were trained for 500 epochs with a batch size of 16. The learning rate was set at 10^{-3} and was not modified during training.

	SLP	MLP
Test Accuracy (Base) [%]	45.83	60.83
Test Accuracy (Augmented) [%]	60.83	66.66

Table 1: Single Low Feature Results

0.3.2 Multiple Features

After the poor results obtained we decided to use all the available Low Level Features which are:

- Color Histogram
- Edge Direction Histogram
- Co-occurence Matrix
- RGB Co-occurence Matrix

In order to use them together we just concatenated them in a big feature vector. The hyperparameters stayed the same for the MLP but we increased the epochs of training to 800 for the SLP. This time the result were a bit better but we still consider them insufficient, as we can see in Table 2. Anyway the significance of using the augmented dataset is once again proven.

	SLP	MLP
Test Accuracy (Base) [%]	59.583	69.583
Test Accuracy (Augmented) [%]	71.25	72.91

Table 2: Multiple Low Features Results

0.4 Neural Features

0.4.1 PVMLNet as Feature Extractor

A leap forward performance wise is to use a CNN (Convolutional Neural Network) as a feature extractor. This way the valuable features are autonomously learnt by the network. In order to do this we used a pre-trained network called "PVMLNet", which structure we can see in Figure 1. This network was trained on a subset

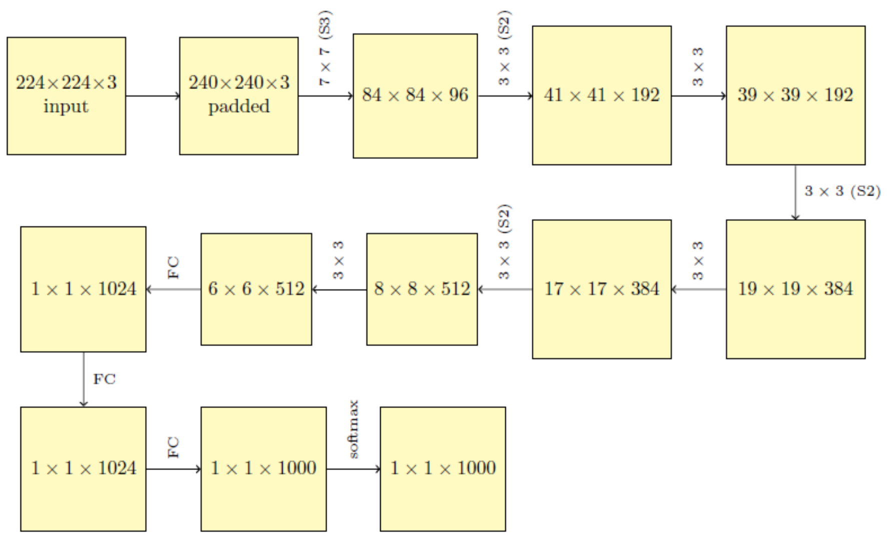

Figure 1: PVMLNet Structure

of the ImageNet Dataset. This means that we are using the features that the network learnt on that dataset hoping that they are general enough to work on ours. To use PVMLNet as a feature extractor we need to remove at least the last two layers, and use the activations of the last remaining layer as the feature vector. We basically pass the each item in the dataset through the PVMLNet and record the activations of the last active layer as its features. Once we processed the whole dataset we train a SLP for the final classification. The first test was performed using the layer before the last two as the feature vector constructor (we will indicate this layer as -3, the last layer with this notation would than be -1 and the one before -3 would be -4). We then used layers $-4, -5$ and -6[1]. We can see all this results in Table 3. Note that we did not keep using the base dataset for all the tests we conducted, since we can be confident that using the augmented one is almost always the better choice. The SLP was trained for 500 epochs with a batch size of 50. The learning rate was set at

	Layer -3	Layer -4	Layer -5	Layer -6
Test Accuracy (Base) [%]	88.75	//	//	//
Test Accuracy (Augmented) [%]	91.25	92.083	93.75	94.167

Table 3: PVMLNet Feature Extractor + SLP Results

10^{-3} and updated at each epoch[2]. Table 3 seems to indicate that the further back we go into PVMLNet the better the result. We were not able to prove this result due to hardware limitations (we ran out of memory during layer -7 testing). Looking at Figure 2 we can see that we probably would have encountered the law of diminishing returns.

0.4.2 Transfer Learning

Another way to use PVMLNet as a feature extractor is via *Transfer Learning*. This method consists in directly replacing the trained SLP to the removed layers in the CNN. This may seem unnecessary since this method yields the same results as the one described above. The major advantage of Transfer Learning is the possibility of *Fine Tuning* the resulting network. This process is nothing but an additional training performed with a very low Learning Rate. The choice of this hyperparamter is crucial since in this case is very easy to fall into overfitting. We chose to fine tune our network for 25 epochs, with a starting learning rate of 10^{-4}. We needed to divide the training set in two, each epoch the network is trained on half the dataset in order to avoid memory

[1]Note that in order to use those layers we need to delete all the ones after them from PVMLNet and adjust the SLP input layer accordingly.

[2]As always with update it with $\mathrm{LR} = \mathrm{LR}_0 \,/\, \sqrt{t}$, where t is the current epoch number.

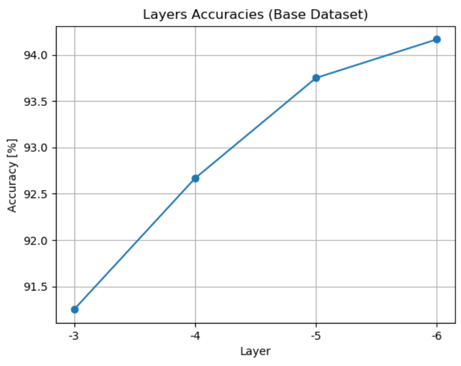

Figure 2: Layers Accuracy

limitations. Our results are presented in Table 4. We can clearly see that we had gained a little accuracy in the Layer -3, we slightly overfitted in Layer -4 case and we did not gain anything in Layer -5 and -6 cases. If we consider that the mean improvement was 0.079%, fine tuning does not seem a very good idea for this problem with this dataset, or at least with the hyperparameters we used.

	Layer -3	Layer -4	Layer -5	Layer -6
Test Accuracy (Base) [%]	89.167	//	//	//
Test Accuracy (Augmented) [%]	91.666	91.666	93.75	94.167

Table 4: Fine Tuned PVMLNet+SLP Results

0.5 Results Analysis

In order to better evaluate our network we computed the Confusion Matrices and annotated the image that was incorrectly classified with the most confidence. Since we did test multiple configuration we are going to discuss only the best classifier for each class, meaning SLP, MLP and Layer -6 networks trained on the augmented dataset. We can notice right away the performance gradient from left to right in Figure 3. The SLP has a very

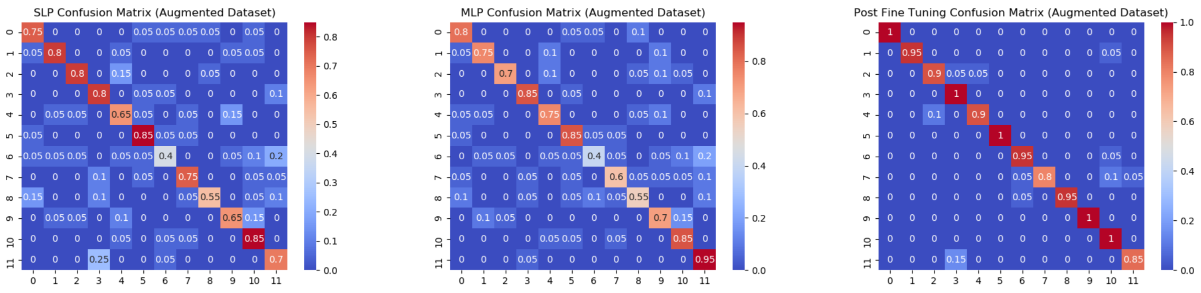

Figure 3: Confusion Matrices: SLD (Left), MLP (Center) and Layer -6 (Right)

noisy Confusion Matrix, Class 6 (Iris) is the most misclassified one with an accuracy of only 40%. The "most wrong" classification[3] is in fact an Iris (Class 6) which is classified as a Windflower (Class 11) with a confidence of 88.82%. The image is easily distinguishable by a human. The MLP Matrix is less noisy but Class 6 is still the less precise. That said though, the "wrongest" image[4] is a Pansy (Class 8) classified as a Lily-Valley (Class 7) with a confidence of 99.9963%. Again, the image is completely distinguishable. As for the Level -6, we have most of the classes correctly classified. The less precise class is Lily-Valley (Class 7), but the most misclassified image[5] is a Colts-Foot (Class 2) classified as a Dandelion (Class 4) with a confidence of 99.9999%. Honestly speaking if we did not know that the image was a Colts-Foot we probably would have struggled to classify it.

[3]image_0425.jpg
[4]image_1313.jpg
[5]image_0884.jpg

0.6 Conclusions

Based on the results we obtained, we can conclude that the Neural Features are the most effective. Letting a Network learn significant features by itself is the cleverest choice in this case. The SLP and MLP Networks greatly fall behind, achieving an accuracy of 30% less than the Layer 6 case. The results obtained with Layer 6 are not perfect, but still, we would not be able to outperform it without increasing our flower species knowledge.

I affirm that this report is the result of my own work and that I did not share any part of it with anyone else except the teacher.