Assignment: Transfer Learning on Intel Image Classification

David Bertoldi – 735213
email: d.bertoldi@campus.unimib.it

Department of Informatics, Systems and Communication

University of Milano-Bicocca

1 Dataset

The chosen dataset is called Intel® Image Classification and it was initially published on Analytics Vidhya by Intel® to host an image classification challenge to promote OpenVINO™, a toolkit or optimizing and deploying AI inference [1][2].

The dataset contains images of natural scenes around the world and they belong to 6 classes: buildings, forests, glaciers, mountains, sea and streets. The images are of size 150 × 150px and can be colored (3 channels, RGB) or rarely in grayscale (still with 3 channels). Figure 1 shows 16 entries of the training dataset.

There is a total of ∼ 24 000 images, divided into Train (∼ 14 000), Test (∼ 3 000) and Prediction (∼ 7 000) folders. The last one does not contain labels and it is intended for unsupervised learning and it will be ignored in this work.

Figure 1: 16 random entries of the train dataset

The distribution of the images across the classes follows a uniform distribution $U(\mu, \sigma)$: in the train set each class has an average $\mu = 2\,339$ images with $\sigma = 105.45$ and in the test set $\mu = 500$ and $\sigma = 36.92$. We didn't find any bias inside the dataset since all the classes are equally populated and so we didn't applied any kind of data augmentation on particular classes for rebalacing.

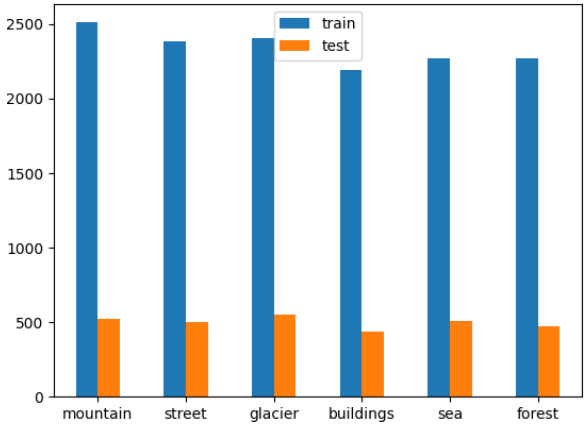

Figure 2: 16 entries of the train dataset

The 6 classes are encoded with numbers 0 to 5 and Table 1 shows the mapping between the numerical and nominative form.

Number	Class
0	Building
1	Forest
2	Glacier
3	Mountain
4	Sea
5	Street

Table 1: Mapping between numbers and names

2 The model

The chosen dataset presented similarieties with ImageNet: the 6 classes of Intel® Image Classification are scattered and distributed in the 1 000 classes of ImageNet. For this reason a pretrained model on ImageNet speeded up the learning process. The chosen model is VGG16, a 16−layers deep CNN proposed by Karen Simonyan and Andrew Zisserman at the

Cut	Trainable parameters	Dimension
`fc1`	117 479 232	$1 \times 1 \times 4096 = \mathbf{4\,096}$
`block4_pool`	7 635 264	$14 \times 14 \times 512 = \mathbf{100\,352}$
`block3_pool`	1 735 488	$28 \times 28 \times 256 = \mathbf{200\,704}$

Table 2: Number of VGG16's trainable parameters and dimensions of the extracted features at each cutting point

University of Oxford[3]. Figure 3 shows an overview of its architecture.

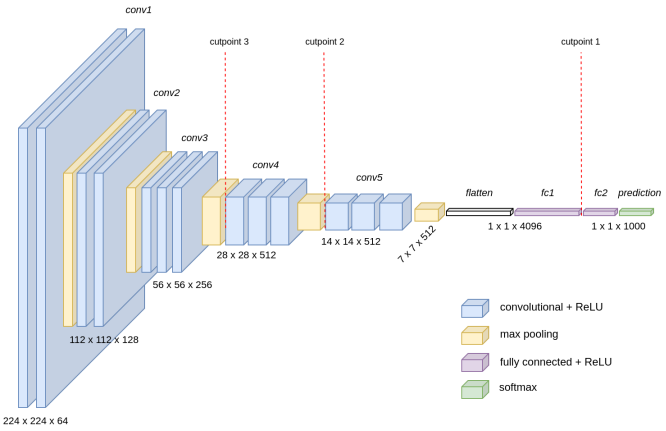

Figure 3: Architecture of VGG16 with the cuts applied in this work

In this work we proposed different cuts to the network and feeded its outputs to a "classic" machine learning model, a SVM, and benchmark the performance of the hybrid architecture.

The chosen cuts were after the first dense layer (`fc1`), after the fourth pooling layer (`block4_pool`) and after the third pooling layer (`block3_pool`). The different cuts led to different challenges, such as the high dimensionality of the features.

As the cuts approached the input, the number of trainable parameters decreased exponentially; however the representation of the features will have an increasingly higher dimension. The higher dimensionality affects the training performance of the SVM and a fine tuning on the management of the memory. For this reason we used less samples during the training phase as the dimensionality increased, increasing the risk of underfitting. Table 2 describe the details of the problem.

References

[1] Practice Problem: Intel Scene Classification Challenge
`https://datahack.analyticsvidhya.com/contest/practice-problem-intel-scene-classification-challe`
[2] OpenVINO™ documentation
`https://docs.openvino.ai/latest/index.html`
[3] *Very Deep Convolutional Networks for Large-Scale Image Recognition*
Karen Simonyan, Andrew Zisserman
`https://doi.org/10.48550/arXiv.1409.1556`