Assignment 4
Advanced Machine Learning
University of Milano-Bicocca
M.Sc. Data Science

Alessandro Riboni
November 25, 2019
ID number: 847160
a.riboni2@campus.unimib.it

Transfer Learning using a CNN pre-trained on IMAGENET

The task of this assignment is to apply Transfer Learning on a pre-trained CNN. This Machine Learning method is very useful because it allows reusing a model developed for a task as a starting point of another one. It is a popular approach in this domain because the best architectures have millions of parameters, and huge databases are necessary to learn them.
This report presents an application of TL on a VGG16, a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition."
The CNN is used as a fixed feature extractor on a new task containing six classes. The neural network has been cut in several points, and a Logistic regression has been used to classify the target according to the features taken from the network.

VGG16

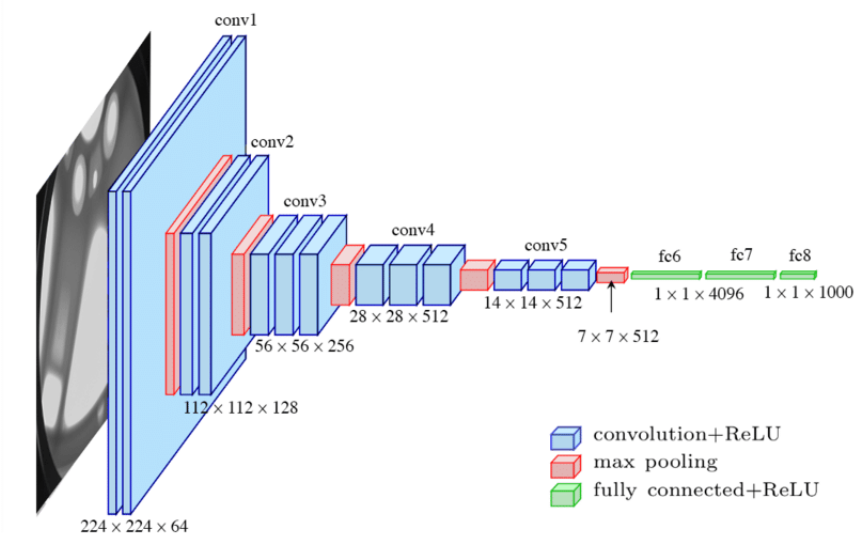


Figure 1: Architecture of VGG16

VGG16 is a convolutional neural network that achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset that contains more than 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It was trained for weeks and was using NVIDIA Titan Black GPUs. The input layer is of a fixed size 224x224 RGB image. The image is passed through a stack of convolutional layers. The filters used have a 3x3 size so as to be able to collect the notions of left/right, up/down, and center. Spatial pooling is carried out by five max-pooling layers, which follow each block of convolutional layers. Finally, there are three Fully-Connected (FC) layers; the first two have 4096 channels each. The third contains 1000 channels (one for each class) and, finally, the last one is the soft-max layer.

New task - Intel Image Classification

The pre-trained CNN was used as a starting point for a new task. The aim is to classify images of size 150x150 in six categories: *building, street, sea, forest, glacier,* and *mountain.*
The dataset Intel image classification contains around 25k, but for this task, only 3000 images have been used for the training set (500 for each class) and 900 for the test set (150 for each class).

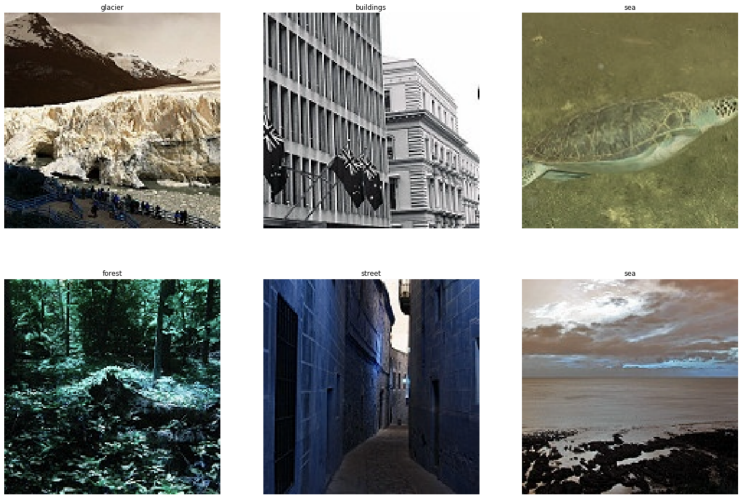


Figure 2: Example of images available in the dataset.

CNN as fixed feature extractor

In the following sections are presented the four different cuts made to the network. The basic idea is to remove N layers, then treat the rest of the VGG16 as a fixed feature extractor for the new dataset. Once the features have been extracted for all images, a Logistic Regression classifier is trained for the new task.

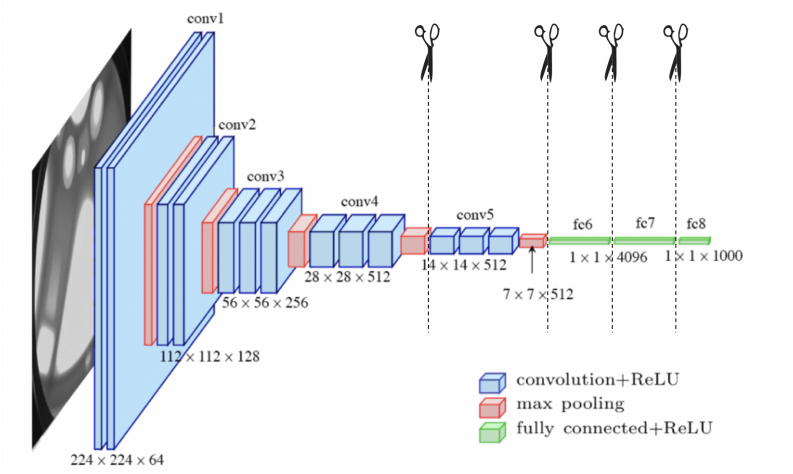
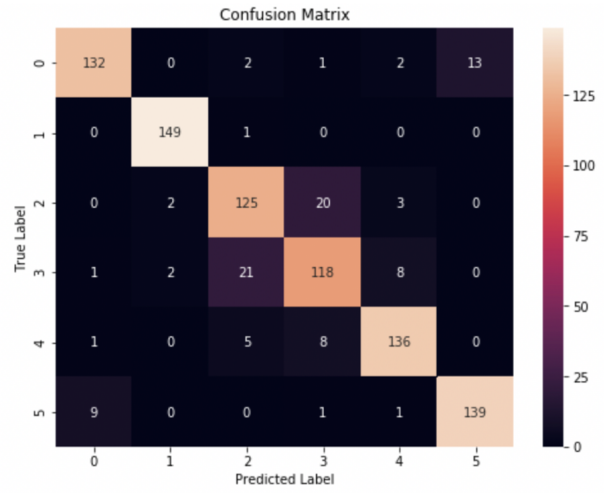


Figure 3: Remove layers of VGG16

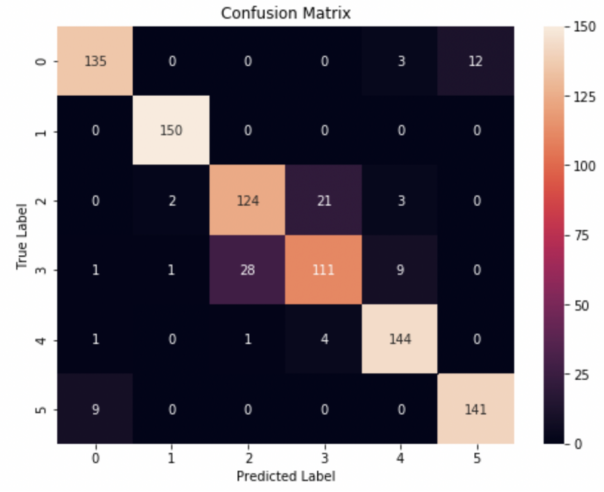
In the first case, the cut was made after the max pooling layer of the fourth convolutional block. The output of the CNN is a feature volume that is collapsed in one dimension by a flatten function.
The extracted features were used to train the logistic regression, and the following figure shows the confusion matrix and accuracy of each class.



	precision	recall	f1-score	support
Building	0.92	0.88	0.90	150
Forest	0.97	0.99	0.98	150
Glacier	0.81	0.83	0.82	150
Mountain	0.80	0.79	0.79	150
Sea	0.91	0.91	0.91	150
Street	0.91	0.93	0.92	150
accuracy			0.89	900
macro avg	0.89	0.89	0.89	900
weighted avg	0.89	0.89	0.89	900

Figure 4: Results of first case

In the second case, the cut was made after the max pooling layer of the fifth convolutional block. In this situation, it was removed only the three Fully-connected layers. The output of the CNN is a feature volume that is collapsed in one dimension by a flatten function.
The extracted features were used to train the logistic regression, and the following figure shows the confusion matrix and accuracy of each class.

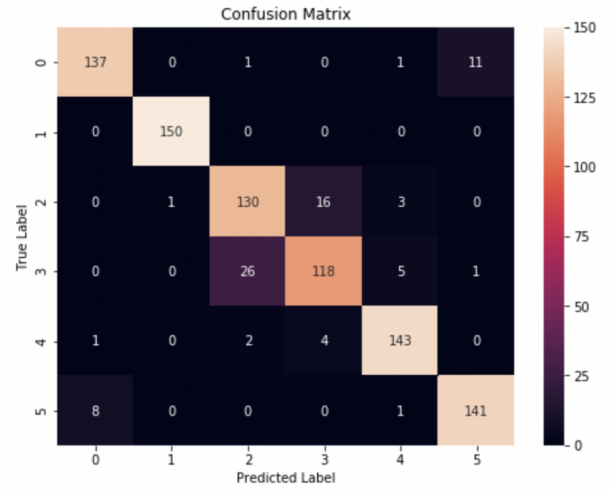


	precision	recall	f1-score	support
Building	0.92	0.90	0.91	150
Forest	0.98	1.00	0.99	150
Glacier	0.81	0.83	0.82	150
Mountain	0.82	0.74	0.78	150
Sea	0.91	0.96	0.93	150
Street	0.92	0.94	0.93	150
accuracy			0.89	900
macro avg	0.89	0.89	0.89	900
weighted avg	0.89	0.89	0.89	900

Figure 5: Results of second case

In the third case, the cut was made after the first Fully-connected layer. In this situation, it was removed two Fully-connected layers, and for every image, the network predicts a 4096-D vector.
The extracted features were used to train the logistic regression, and the following figure shows

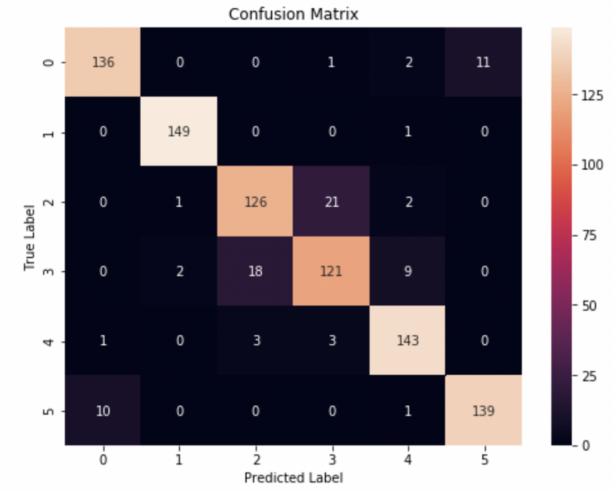
the confusion matrix and accuracy of each class.



	precision	recall	f1-score	support
Building	0.94	0.91	0.93	150
Forest	0.99	1.00	1.00	150
Glacier	0.82	0.87	0.84	150
Mountain	0.86	0.79	0.82	150
Sea	0.93	0.95	0.94	150
Street	0.92	0.94	0.93	150
accuracy			0.91	900
macro avg	0.91	0.91	0.91	900
weighted avg	0.91	0.91	0.91	900

Figure 6: Results of third case

In the fourth and last case, the cut was made before the output layer. In this situation, it was removed only one Fully-connected layers and for every image the network predicts a 4096-D vector.
The extracted features were used to train the logistic regression, and the following figure shows the confusion matrix and accuracy of each class.



	precision	recall	f1-score	support
Building	0.93	0.91	0.92	150
Forest	0.98	0.99	0.99	150
Glacier	0.86	0.84	0.85	150
Mountain	0.83	0.81	0.82	150
Sea	0.91	0.95	0.93	150
Street	0.93	0.93	0.93	150
accuracy			0.90	900
macro avg	0.90	0.90	0.90	900
weighted avg	0.90	0.90	0.90	900

Figure 7: Results of fourth case

In all cases presented, a Logistical Regression was used as a classifier. Due to the excessive dimensionality of the features, a grid search was not applied for each case.
The solver *lbfgs* was used, which is suitable for handle multinomial loss. This optimization algorithm is in the family of quasi-Newton methods and compared to the solver *newton-cg* uses a limited amount of computer memory. Working on balanced data, it was not necessary to assign weights to the classes. In addition, the default values for the other parameters of the classifier were used.

Consideration

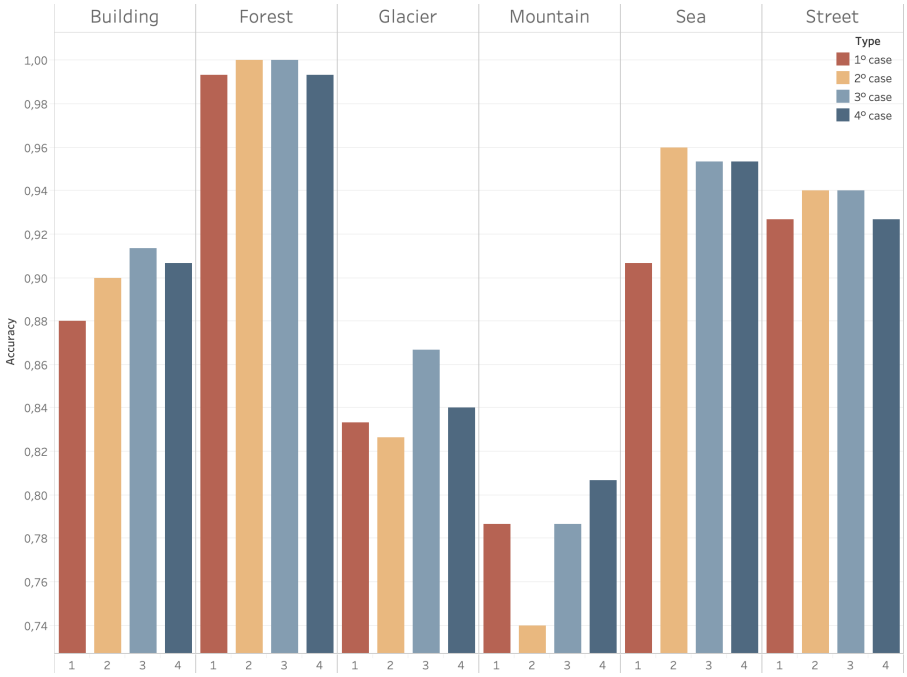


Figure 8: Side-by-side bar chart of accuracy

From the classification reports and the side-by-side bar chart previously presented, it can be seen that the best case is the third, i.e. removing the last two Fully-connected layers.
The classes that have a worse classification are *forest* and *mountain*.
The huge amount of data used to train the network and its complexity justify overfitting. In fact, in all cases presented the accuracy on the training set is 100% for each class.
In conclusion, it can be observed that the performance of the model is very good for all the cases presented.
The differences are minimal, and the choice of the best method must also be influenced by the computation time needed to extract the features and train the classification model.
In particular, by cutting the network before Fully-connected layers, the computation time is longer due to the larger feature size extracted from the pre-trained VGG16.