Natural Images Classification with Convolutional Neural Networks

Daniela Campisi danielacampisi96@gmail.com

1. Model Description

The model described here **Conv2**, is a deep model consisting of 2 Convolutional Layers, 1 Fully Connected Layer and a final Classifier, used for a multiclass classification problem. Precisely, the task of the model is to recognize the type of image among the 8 classes considered and assign it the correct label.

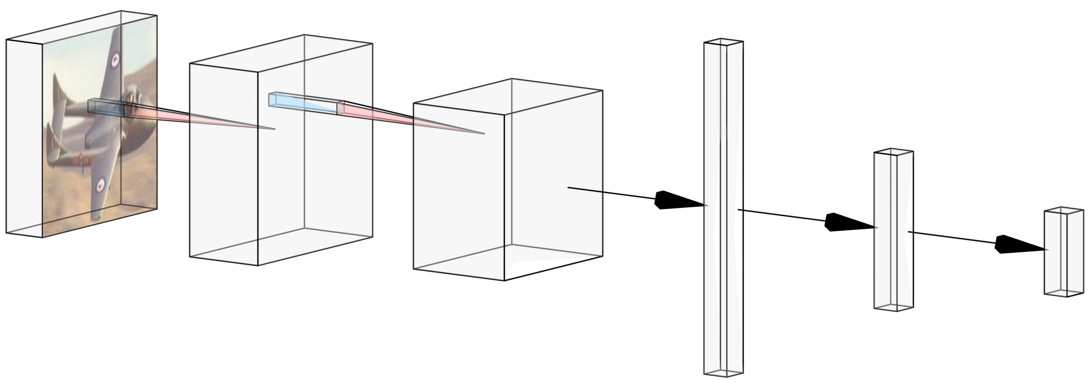

Figure 1: A simple representation of the proposed model.

Initially, the input data flows through the convolutional layers to exploit image information. Then the extracted characteristics are provided to the fully connected layer and to the classifier, which predicts the class result among one of the 8 possible classes. The first convolutional layer consists of 3 input channels (because the model processes RGB images) and 8 output channels, while the second convolutional layers consists of 8 input channels and 16 output channels. Both layers apply kernels of size 3x3 with padding 1x1 and stride 2x2. The first convolutional layer is followed by a ReLu activation function, which introduces non linearity to the model, while the second one is followed also by a Max Pooling layer with kernel size 2x2 and stride 2x2, which reduces the spatial dimension of the input data.

The output of the convolutional part of the model is flattened into 4096 features which represent the input of the fully connected layer, which is followed by a ReLU activation function. Finally, there is the classifier layer which assigns the correct label to the images, namely it classify the image to the correct class, among the 8 present in the dataset.

2. Dataset

This dataset is created as a benchmark dataset for the work on **Effects of Degradations on Deep Neural Network Architectures**, a paper of *Prasun Roy, Subhankar Ghosh et alter*. Convolutional neural networks (CNNs) depend on several sets of filter outputs to perform the final classification task. Thus, it is difficult to predict the outcome of a degraded input in an intuitive way and the classification accuracy largely depends on the model architecture and the nature of the degradation. These image degradations can be obtained due to poor image sensor, lighting conditions, focus, stabilization, exposure time etc.

Natural Images Dataset is about, namely, natural images, meaning images that a human being would observe in the real world, such as landscapes, indoor scenes, roads, mountains, beaches, people, animals, and automobiles, as opposed to synthetic images or images generated by a computer.

The dataset contains 8 different classes -*airplane, car, cat, dog, flower, fruit, motorbike and person* having 727, 968, 885, 702, 843, 1000, 788 and 986 image samples respectively, for a total of 6899 images. The sample images are subjected to data augmentation. First of all they are resized to 128×128 pixels. Then, I applied horizontal and vertical flipping with probability of 0.5 (which is the default value). The transformed data are stored using the ImageFolder class of PyTorch. Finally, data are splitted into training, validation and test sets, composed by 5523, 688 and 688 respectively (just about the 80%, 20%, 20%, of the total respectively).

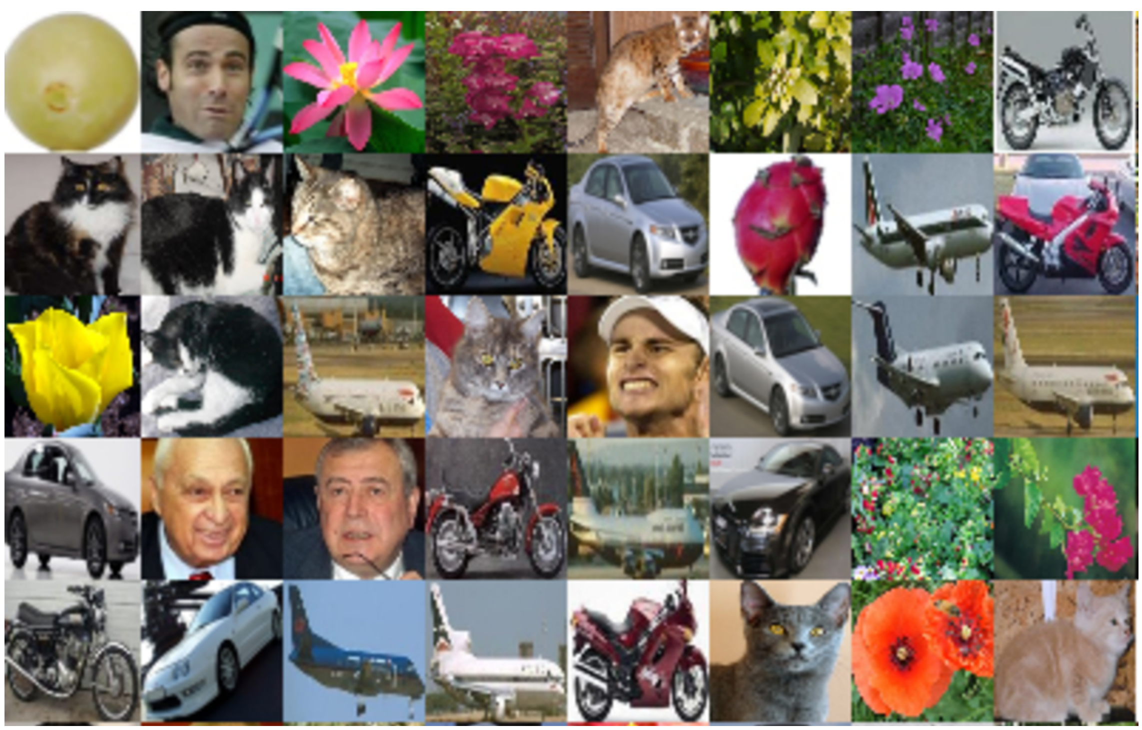

Figure 2: Example images from the dataset.

3. Training procedure

The training procedure starts with the Conv4 model, which has 4 convolutional layers, 1 fully connected layer and a final classifier, and continues by training 3 other models, deleting each time a convolutional layer (*ablation study*), in order to verify at the end which of the 4 models has a higher validation accuracy value.

To optimize the model during training the Cross-Entropy Loss has been employed as loss function, which aims to make the model output be as close as possible to the desired output (truth values). The chosen optimizer is Adam with the learning rate of 0.01. The batch size has been set to 32. The training set has been shuffled by the data loader. Finally, all the models are trained for 10 epochs on Google Colab.

4. Experimental Results

Models were trained as described in the previous section. Table 1 shows test results for the proposed architectures as well as of ablation studies (i.e., different variants of the final architecture when adding or removing layers).

Model	Validation Accuracy	Test Accuracy
Conv4	75.43%	74.01%
– 1 Conv Layer (Conv3)	80.97%	80.11%
– 1 Conv Layer (Conv2)	84.66%	83.10%
– 1 Conv Layer (Conv1)	73.86%	74.86%
Best model (Conv2)	84.66%	83.10%

Table 1: Performance of the models.

The ablation study showed that by gradually removing a level from the starting model -Conv4, network with 4 convolutional layers- there is an improvement in the value of validation accuracy, up to a model with two convolutional levels. If, however, we continue to delete a layer from Conv2, the validation accuracy value decreases, reason why the best model is Conv2.