Object recognition and computer vision 2023/2024
Assignment 3 : Sketch Image classification

Mathis Le Bail
mathis.le-bail@entsa-paris.fr

Abstract

The goal of this project is to design and train a model to classify a dataset of sketches which contains 250 different classes. This report outlines the approach taken to establish a tailored model, starting with a brief introduction of the dataset followed by the preprocessing steps, the model architecture, and the selection of hyperparameters. Finally, the report concludes by presenting the final performance of the obtained model.

1. Dataset presentation

The dataset is adapted from [1] which contains 250 different classes of sketches. The dataset we are working with has been reduced in the number of images. It consists of 12,000 training images, 2250 validation images and 5750 test images. They are all gray scale sketches of dimension 1111×1111 pixels.

2. Data Preparation

We resize the images to match the input size expected in the model architecture used. The pre-trained ResNet34 model will be used as explained later. So, we apply the same transformation to our images as the one used on ImageNet data for training the ResNet34 weights. This transform is composed of the following steps : the sketches are resized to a 256×256 dimension then cropped around their center point to a square of size 224×224 pixels. Finally, they are normalized. The mean and the standard deviation are chosen to be the same as the ones of the dataset on which the pre-trained model was originally trained i.e m_{train} : 0.485 and std_{train} : 0.229.

In addition, we apply a data augmentation technique to "artificially" increase the diversity of sketches. To achieve this, we add the same dataset to which random flips and 30-degree rotations have been applied to the images. It is a common technique that usually helps the model to generalize better to variations in the input data, and this is what we noticed in practice in our results.

3. Model architecture

By comparing the results obtained between some 'basic' models and models pre-trained on ImageNet databases, we observe a significant performance gap in favor of pre-trained models like ResNet for instance. This is not surprising given that the training dataset remains relatively small and composed of simple shapes. The pre-trained model brings superior prior knowledge that cannot be directly obtained from the data. We choose to use the ResNet34 architecure from *torchvision* with the weights reproducing closely the results of the model's paper. To fine-tune the pre-trained ResNet34 model to our need, we replace the last fully connected layer by a classification layer specific to our case i.e. which has 250 nodes.

4. Training strategy

There are two training phases. First, all the layers in the pre-trained ResNet34 model except the last classification layer that we added are frozen. This means these layers will not be updated during the first epochs of training so we will only learn the parameters of the new final layer. It allows the model to adapt to the new dataset without losing the knowledge acquired during pre-training. After a certain number of training epochs, which we empirically set to 8 (the number of epochs beyond which we no longer observed improvements in the metrics), we start the second phase and all the parameters of the layers are unfrozen which means all the layers of the model are fine-tuned.

5. Choice of hyper parameters

Several combinations of hyperparameters were first tested using a random search to narrow down the possibilities, and then with a grid search to refine those that seemed promising. We thus empirically chose the best combination : a learning rate of 0.01, a momentum of 0.9 and a weight decay of 0.001 for the SGD optimizer. In the second phase, the learning rate is divided by 10 to prevent drastic changes to the pre-trained weights. To avoid too much over-fitting, a dropout layer was also added just before the final classification layer, a dropout probability of 0.7 is chosen.

6. Results

The model is trained with a first phase of 8 epochs and a second phase of 8 epochs (due to GPU disconnections for longer durations). We obtain a final validation accuracy of 78.53% for the best epoch.

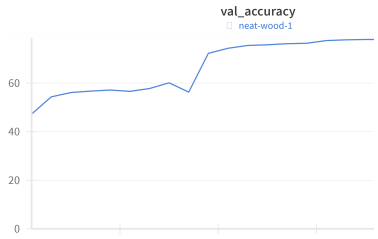

Figure 1. Evolution of the validation accuracy of the model. The significant jump in accuracy corresponds to the start of the second phase when all model weights are unfrozen.

References

[1] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31 (4):44:1–44:10, 2012. 1