Object Recognition and Computer Vision
Sketch Recognition

Maxence Gollier
Institut Polytechnique de Paris
19 Pl. Marguerite Perey, 91120 Palaiseau, France
maxence.gollier@ip-paris.fr

Abstract

We perform sketches image classification based on the dataset from [1]. We use transfer learning to obtain human-level accuracy.

1. Introduction

The provided data is the TU-Berlin sketch dataset which has 250 classes. There are $12,000$ training images and $2,250$ validation images. The final test set contains $5,750$ images.

2. Preprocessing

Since we only have 48 images per class, we use data augmentation in order to obtain better results. We used pytorch *RandomAffine*, *RandomCrop* and *RandomAffine* to get more robust models. These transformations are made at training and also allowed us to reduce models overfitting.

3. Transfer learning

The training being performed locally on a rather small GPU, we choose to use transfer learning to train our model. All architectures presented use ImageNet1k pre-trained weights which allowed us to use bigger architectures despite our small computational power. For the optimization part, we tried different settings but found that SGD with a momentum fixed at 0.9 and an exponential learning rate schedule gives the best trade-off between validation set accuracy and overfitting. We now turn our attention to present the different models that were trained

3.1. ResNet50

We used the architecture from [2] which is implemented in torchvision.For this architecture, we fine-tuned the 6 last layers of the network with various data augmentation parameters and learning rate initialization. Overall, we obtained an accuracy of 69.5% with this architecture.

3.2. EffNet

We then used [3]. We trained this architecture from end-to-end again with various data augmentation parameters and learning rates but did not find any good result with the most promising results hardly getting above 50% accuracy.

3.3. GoogleNet

Finally, we turned our attention to a simpler architecture, namely GoogleNet which is a CNN implemented in torchvision. Tweaking the hyper-parameters allowed us to get an accuracy of 68%.

Finally, we mixed the ResNet50 and GoogleNet architectures by summing the outputs from each to get our current accuracy on Kaggle which is about 71.5%.

4. Conclusion

We reached almost 73% on Kaggle which is the human-level accuracy according to [1] with an accuracy of 73%. However, we did not score as much as other participants this means that there should be hyperparameters that haven't been well adjusted. With such a long time per epoch, we didn't take the time to estimate these but are convinced that better performance could be attained this way with the same architecture (namely, ResNet and ConvNet).

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

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[3] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 1