Image Classification using Deep Learning and Neural Network

SOURAJIT GHOSH

Dept: CSE(AIML), 3RD YEAR, ROLL:59

Under the supervision of our teacher:

DR. SWARNENDU GHOSH

ABSTRACT

In the ever-evolving landscape of computer vision, the progress made in deep learning and neural networks has positioned image classification as a formidable tool for identifying and categorizing objects within images. This technological advancement has brought about a significant transformation across diverse sectors, spanning from the realm of medical imaging to the development of autonomous vehicles. The impact of this innovation has proven profound, and its prospects for further applications are on a continuous ascent.

KEYWORDS- Image classification, Machine learning, Convolutional neural networks (CNNs), Tiny ImageNet dataset, Object recognition, Visual categorization

INTRODUCTION To TINY IMAGENET:

Tiny ImageNet is a compact yet powerful dataset designed to facilitate research and development in the field of computer vision and image recognition. Serving as a subset of the larger ImageNet dataset, Tiny ImageNet retains the diversity and complexity of its parent dataset while significantly reducing its scale. This downsizing makes it an ideal resource for researchers, developers, and enthusiasts who seek to explore and experiment with image classification models in a more computationally manageable environment. Containing 200 diverse classes, each with 500 training images, 50 validation images, and 50 test images, Tiny ImageNet offers a balanced and representative sample of the broader ImageNet categories. The dataset covers a wide range of objects, animals, and scenes, challenging machine learning models to generalize effectively across various visual concepts.

The compact size of Tiny ImageNet facilitates faster training and experimentation with deep learning models, making it an attractive choice for educators, students, and researchers who may not have the computational resources required for larger datasets. Researchers often leverage Tiny ImageNet as a stepping stone before transitioning to more extensive datasets, allowing for rapid prototyping and iterative model development.

Whether for educational purposes, benchmarking, or as a starting point for novel research endeavors, Tiny ImageNet provides a rich and diverse collection of images that enables the exploration of cutting-edge techniques in image recognition, classification, and object detection within a resource-efficient framework. Its balance between complexity and manageability makes it an invaluable asset in advancing the capabilities of computer vision systems.

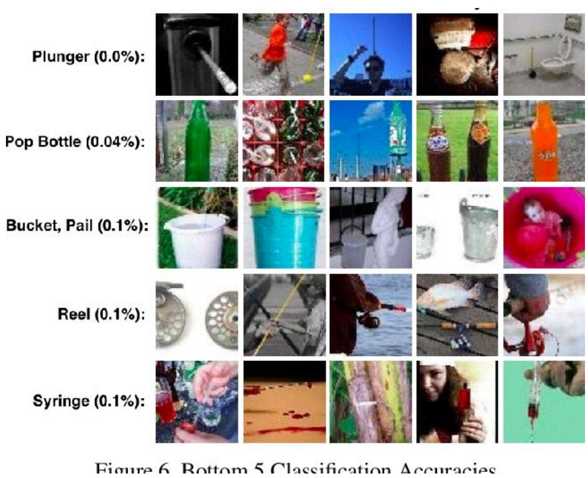


Fig1.Tiny ImageNet Classification

Dataset:

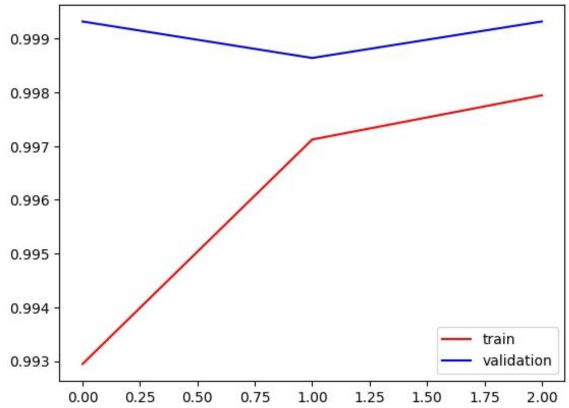
The Tiny ImageNet dataset [4] is a modified subset of the original ImageNet dataset [1]. Here, there are 200 different classes instead of 1000 classes of the ImageNet dataset, with 100,000 training examples and 10,000 validation examples. The resolution of the images is just 64x64 pixels, which makes it more challenging to extract information from it. A glance at the images shows that it is difficult for the human eye to detect objects in some images.

Model Design

In the beginning, without much effort, we implemented the vanilla ResNet-34 and ResNet50 models as our Network 1 and 2 respectively.
For ResNet-34, we achieved an accuracy of 99.8% on the training set and 33.5% on the validation set with a batch size of 500 images running for 100 epochs. For ResNet-50 we achieved an accuracy of 49% on the training set and 26.2% on the validation set with a batch size of 512 images running for 50 epochs.

Results :

For Network 1 we trained our model with 17.9 Million parameters for 235 epochs with a batch size of 256 for 32x32 and 16x16 resolution images and a batch size of 64 for 64x64 resolution images. At first, we ran 32x32 resolution training images for 15 epochs and reached a validation accuracy of 25%. At this point, the model had already started overfitting, so we switched to 64x64 resolution images and Figure 6. Loss curve of Network 1 ran it for 30 epochs which increased the validation accuracy to 48%. In order to train the model on low-resolution images and simulate scaled out augmentation, we trained it on 16x16 resolution images for just 10 epochs. We kept epochs low here because although we wanted the model to learn from low information images, we also wanted to avoid overfitting. We changed back to 64x64 resolution images and stuck to it for the rest of the training. After running it for 30 more epochs, our validation accuracy saturated around 48-49%, and now we implemented image augmentation. We performed some of the augmentations on every image and ran our model for another 150 epochs. Due to several augmentations, now the model trained much slower but had yet to begin overfitting. The gap between training and validation accuracy remained small, and we trained it until the validation accuracy saturated to around 59%. Here, the training accuracy reached 67%, so we did not train it further to avoid overfitting. We selected the best model from our training using model checkpoint which gave an accuracy of 59.5% on the validation dataset. For Network 2 we trained our model with 11.8 Million parameters for 108 epochs with a batch size of 128 image



As we can see from Figure 7, Cyclical Learning Rate helped us in early attainment of higher validation accuracy steeply bringing down the validation loss. It is also distinctly visible that after 70 epochs our validation loss reached a plateau for almost 10 epochs. At the end of 108 epochs, we achieve a validation accuracy of 62.73% whereas our training accuracy remained at 68.11%, indicating a narrow gap again. Both the networks achieved validation accuracy higher than Shallow ConvNets and ResNet18 (with no Dropout) models and are very close to state-of-the-art ResNet-18 (with various Dropout techniques) models

Future Work and Conclusion:

We tailored the state-of-art deep CNN models to the 200-class image classification problem presented in Tiny ImageNet Challenge. We implemented and modified VGGNet, ResNet, and Inception-ResNet. We trained all our models from scratch and the top-performing model achieves a top-1 test error of 43.1%. We also conducted thorough analysis on the results. We found our best model is able to recognize images with clear features, while failing to recognize others with multiple objects or ambiguous features. We also investigated different initialization strategies, and we found variance scaling initialization works better than xavier initialization for our ResNet-based models. Furthermore, we discovered that a proper weight decay strength can give a small boost to the performance.