Image Classification using CIFAR100

Sai Akhilesh Ande
Dept. of Mechanical and Industrial Engineering
College of Engineering, Northeastern University
Boston, MA
ande.s@northeastern.edu

***Abstract*—Image Classification refers to the task of categorizing images by assigning pre-defined labels. Today, Convolutional Neural Networks(CNNs) are the state-of-the-art methods for Image Classification. The goal of this project is to explore various CNN architectures and build an effective CNN-based model to classify images from the CIFAR-100 [1] dataset. This work involves Convolutional Neural Networks like ResNet, VGG16, and Transfer Learning using EfficientNet.**

***Index Terms*—ResNet9, VGG16, EfficientNet**

I. Introduction

Image Classification is one of the fundamental problems in the domain of Computer Vision. It has wide applications like Object Recognition, Facial Recognition, Self-driving cars, etc. It involves assigning pre-defined labels to a group of pixels or vectors within an image dependent on particular rules. It is important because, in this era of data, with the Internet of Things(IoT) and Artificial Intelligence (AI) becoming ubiquitous technologies, we now have huge volumes of data being generated. In the form of photos or videos, images make up for a significant share of global data creation. Some common applications are Automated inspection and quality control, Object recognition in driverless cars, Detection of cancer cells in pathology, Face Recognition in Security, Traffic monitoring, and congestion detection.

In this project, I have built and trained CNN models from scratch based on traditional architectures like VGG16, and ResNet9. I trained them with different batch sizes, optimizers, and learning rate schedulers with the objective of attaining the best possible accuracy on the CIFAR100 dataset [1]. The VGG16 model is a very deep network with 16 layers. I achieved an accuracy of only 55% with the model and this might be due to the problem of vanishing or exploding gradient problem. The ResNet9 model is a not-so-deep model with only 9 layers. I achieved the best accuracy of 74%. I found that applying Transfer Learning using pre-trained models gave much better performance. Transfer learning with the EfficientNet model and fine-tuning gave the best accuracy of 81.32%.

II. Background

Yann LeCun proposed the LeNet-5 [2] in 1998. It showed the preliminary success of multi-layered CNNs for image classification. Since then, multiple variants of this architecture came into existence for various image classification tasks and achieved notable results on MNIST, CIFAR and ImageNet datasets. The recent trend with large datasets such as ImageNet has been to scale up the models even further and mitigate the problem of overfitting through intermittent dropout layers. VGG [3] is one such architecture that has been successful at the ImageNet 2014 competition. VGG architecture has 16-19 Convolutional layers followed by three dense layers whose final layer emits softmax normalized probabilities over 1000 different classes in the ImageNet challenge. The input images are of size 224x224 RGB images.

Post its success in the challenge, VGG [4] has been used as a pre-trained model by removing the final dense layers and training additional custom layers over the frozen base model for many other downstream tasks. When even deeper architectures were experimented with, they showed a rapid deterioration in performance. In order to mitigate this, residual connections were introduced in very deep neural networks. Experiments have shown this architecture to have a robust performance even for 1000-layer deep models. As the model gets deeper it has been able to extract different high/medium/low-level features and through the residual connections fuse them together thereby learning richer representations that are useful for the end tasks.

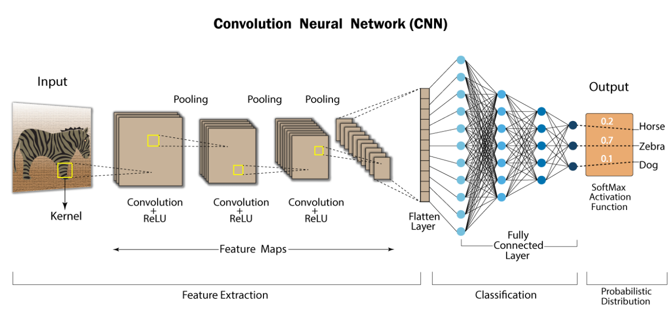

Fig. 1. Convolutional Neural Network

Another such deep network is EfficientNet [5] which was proposed in 2019 with a novel model scaling method that uses a simple yet highly effective compound coefficient to scale up CNNs in a more structured manner. The **efficient adaptive ensembling** [6] [7] based on the EfficientNet is so far the best performing CNN model on the CIFAR100 dataset with a testing accuracy of 96.80%. This model is published in 2022.

III. APPROACH

A. Dataset

The CIFAR100 dataset [1] is a subset of the ”80 million tiny images” dataset [8]. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. It has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a ”fine” label (the class to which it belongs) and a ”coarse” label (the superclass to which it belongs).

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

Fig. 2. CIFAR100 classes and superclasses

As each class contains only 500 training samples and 100 testing samples, I used the test set itself as the validation set without any further splitting.

B. Visualizing the data

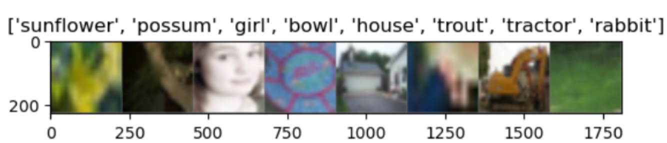

Fig. 3. CIFAR100 sample images

Fig. 4. CIFAR100 sample images

Fig. 5. CIFAR100 sample images

C. Transformations

Image transformations like RandomResizedCrop(), RandomHorizontalFlip() and Normalization with the mean and standard deviation of Training Images were applied for the Training and Testing image datasets.

D. Loss Function

Cross Entropy Loss was used for multi-class classification.

$$L_{crossentropy}(y,p) = -(y\log(p) + (1-y)\log(1-p))$$

E. Evaluation Metrics

I choose to use Accuracy as a performance measure because the classes are balanced in Train and Test sets.

$$Accuracy(y,\hat{y}) = \frac{1}{n_{\text{samples}}}\sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

IV. MODELS

A. VGG16

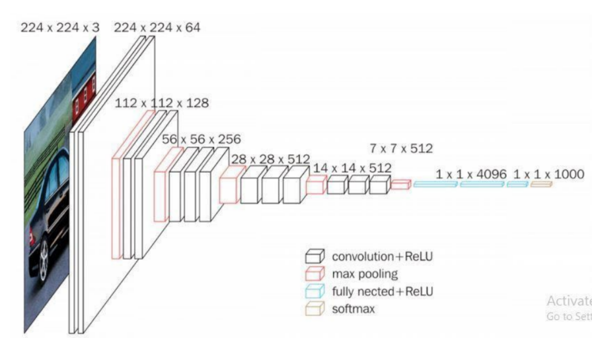

Fig. 6. VGG16

B. ResNet9

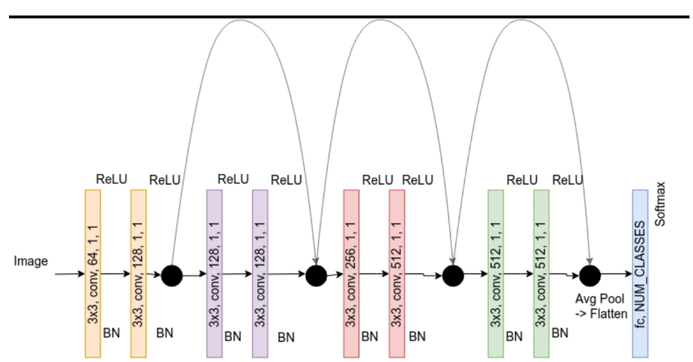

Fig. 7. ResNet9

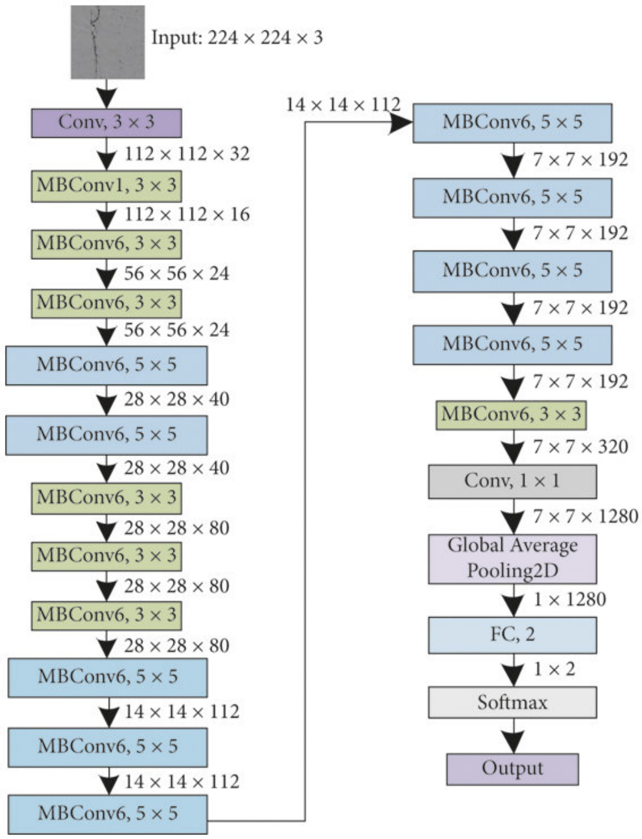

Fig. 8. EfficientNet_b0

Optimizer	Batch size	Test Acc
SGD (LR = 0.01)	512	71.88
	400	65.97
	64	72.51
Adam + Nestrov Momentum	512	69.84
	400	74.32
	64	72.78
RMSProp	512	69.2
	400	70.96
	64	71.43

Fig. 9. ResNet9 results

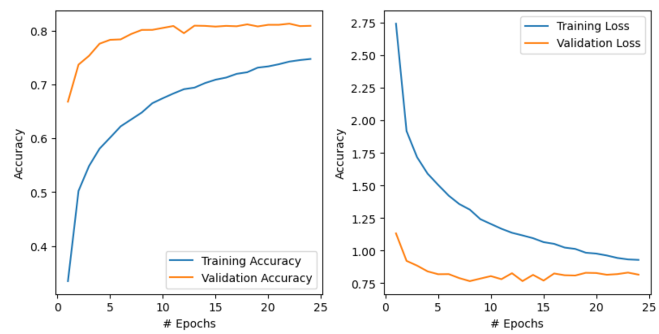

Fig. 10. Transfer Learning with EfficientNet_b0

C. Transfer Learning with EfficientNet

V. Results

A. VGG16

Adam with Nesterov momentum and RMSProp optimizers were used with ReduceLROnPlateau Learning-rate scheduler. Although VGG16 performed the best on the ImageNet dataset, this architecture didn't yield good results when I trained it from scratch on the CIFAR100 dataset. The best-achieved performance was 55.17% Average Testing Accuracy with SGD optimizer. It might be because of the vanishing and exploding gradients problem given its deep nature.

B. ResNet9

SGD, Adam with Nesterov momentum and RMSProp optimizers were used with ReduceLROnPlateau Learning-rate scheduler. The best performance of 74.1% Test Accuracy is obtained with Adam and Nesterov momentum with a batch size of 400. ResNet9 architecture gave much better results.

C. Transfer Learning with EfficientNet

Fine-tuning the EfficientNet model initialized with pre-trained weights with a learning rate of 0.0001, batch-size of 8, Adam optimizer with weight decay of 1e-4, ReduceLROnPlateau learning rate scheduler gave the best performance of 81.31% Test Accuracy.

VI. Conclusion

From the series of experiments, I conclude that the EfficientNet model initialized with pre-trained weights and fined-tuned on the CIFAR100 dataset performed the best when compared to the rest of the models. For Image Classification, pre-trained models trained on other large-scale datasets such as "ImageNet" could be used on the CIFAR100 dataset by optionally adding additional learnable layers and fine-tuning. This works well because the pre-trained networks already have a rich feature representation that can be tuned to learn the given task. Adopting transfer learning techniques significantly saves a large amount of training time and also helps in achieving much better performances when compared to training from scratch. Deeper architectures can better extract complex patterns and rich features from the data which is evident from the performance of pre-trained EfficientNet.

The next steps could be

- To try a different combination of Image Transformations
- To implement data augmentation so that we can have more Training data.

- To apply Transfer Learning with the latest models like Coca[9]

REFERENCES

[1] https://www.cs.toronto.edu/ kriz/cifar.html
[2] Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541–551, December 1989.
[4] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
[5] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks Mingxing Tan, Quoc V. Le
[6] Efficient Adaptive Ensembling for Image Classification. Antonio Bruno, Davide Moroni, Massimo Martinelli 2022
[7] https://paperswithcode.com/sota/image-classification-on-cifar-100
[8] http://groups.csail.mit.edu/vision/TinyImages/
[9] CoCa: Contrastive Captioners are Image-Text Foundation Models Jiahui Yu, ZiRui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, Yonghui Wu · 2022