GhostMixNet: More Features, Less Complexity

Kieran Gallagher, Mathew Martin, Natasha Sebastian

[1]New York University Tandon School of Engineering, New York, USA

Abstract

In this work, we analyze the effectiveness of various preprocessing and attention mechanisms, as well as GhostNet modules, in the training and validation of a ResNet architecture with only 3 million parameters. Trained and evaluated on the CIFAR-10 dataset, the model reaches 95.5% accuracy on the validation data, despite the operations being cheap enough to run on mobile or IoT devices.

Public Repository

The python implementation of GhostMixNet can be found here:
github.com/NYUNeuroNinjas/GhostMixNet

Introduction

ResNet is one of the most influential deep neural network architectures in computer vision, known for its straightforward yet powerful residual learning framework. By stacking a large number of layers using shortcut connections, ResNet achieves remarkable accuracy in tasks such as image classification, object detection, and semantic segmentation. However, while the trend toward deeper and more complex networks has pushed accuracy to new heights, it has not necessarily improved computational or memory efficiency. In many practical scenarios, such as embedded devices, robotics or self-driving cars, models need to operate on resource-constrained hardware and deliver real-time performance. The demand for lighter variants of ResNet has been growing. MobileNet and WideNet have tackled this challenge by introducing architectural modifications to improve speed and reduce complexity. In this paper, we build on these ideas and propose novel optimizations to further enhance Resnet's efficiency while keeping the model parameters under 5M.

Data

The CIFAR-10 Dataset, comprised of 60,000 32x32 color images representing 10 classes, was used for training, validation, and testing of our network architecture. Data augmentations were applied to only the training set in order to improve model generalization.

Methodology

Figure 1: GhostMixNet Architecture

Final Training Setup:

- **Optimizer:** AdamW with Beta1 of 0.9 Beta2 of 0.999 and weight decay of 0.001.
- **Data Augmentation:** RandomCrop, RandomHorizontalFlip, RandomPerspective, RandomPosterize, ColorJitter, AutoAugment, Mixup and CutMix.
- **Regularization:** Dropout with a probability of 0.2 was used to combat overfitting.
- **Learning Rate and Schedule:** Started with a Learning Rate of 0.001 and used a Cosine Annealing Warm Restarts scheduler, with warm restarts after 50 epochs.
- **Batch Size and Epochs:** A batch size of 128 and trained for up to 200 epochs.

GhostNet

In this work, we incorporate the GhostNet concept in [1] to reduce the computational burden in our convolutional backbone. GhostNet generates features in two stages. It first generates a smaller set of intrinsic feature maps with regular convolution. Then, it uses depthwise convolutions known as cheap operations to produce additional ghost feature maps from the intrinsic ones. This strategy reduces both computational cost and parameter count, as the depthwise convolutions are significantly less expensive than full convolutions.

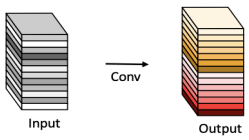

Figure 2: Convolutional layer[1]

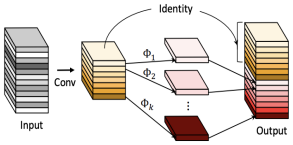

Figure 3: Ghost Module[1]

Squeeze and Excitation Block

Squeeze and Excitation (SE) networks, introduced by Hu et al. in 2020[2], rely on the SE block, a mechanism that models the inter-dependencies between channels in the feature map of a Convolutional Neural Network. The name comes from first the Squeeze layer, in which average pooling is used to produce a single value for every channel of the feature map, before a ReLU or similar activation generates a set of weights emphasizing the importance of each channel. These weights are then applied to the original feature map in the "Excitation" step, which enhances the quality of the features in our ResNet. For the implementation of this block, we used the SqueezeExcitation function provided by the torchvision.ops package, and placed it within our Residual Layer building block.

Residual Layers

Residual Layers form the core of the ResNet architecture, integrating multiple Convolutional Blocks. In this model, each residual layer consists of a series of Ghost Blocks, with the number of blocks determined by the specific architectural configuration. For this project we have utilized three layers of four GhostBlocks each with each GhostBlock comprising two GhostModules and a Squeeze Excitation layer which was found to provide better validation and test algorithms. The stride parameter regulates feature map downsampling, allowing the network to capture features at multiple scales. Skip connections help maintain effective gradient flow, addressing the vanishing gradient problem and improving training stability. This hierarchical structure enables the network to progressively learn more abstract representations of the input data, ultimately enhancing classification performance.

Image Transform Augmentations

PyTorch provides an array of image transforms that were incorporated into the training data augmentation step of GhostMixNet. These seek to generalize the classification capabilities of the model and reduce overfitting, while increasing training loss.

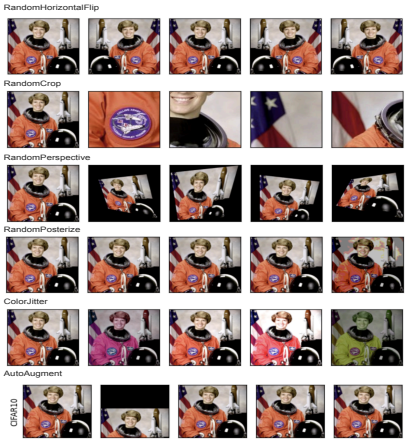

Figure 4: Image Augmentations used in GhostMixNet

Mixup Training Augmentation

Introduced by Zhang et al.[5], Mixup is a technique used for data augmentation that generates training data based on a weighted combination of random pairs of images from the training set. Data is generated in a sample $(\tilde{x}, \tilde{y})$ where:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j$$

Here, (x_i, y_i) and (x_j, y_j) are the samples and their labels from the training set. The benefit of this augmentation is that it encourages the model to learn smoother decision boundaries by training on linear interpolations of input samples and their corresponding labels. This reduces overfitting and helps the model generalize better to unseen data.

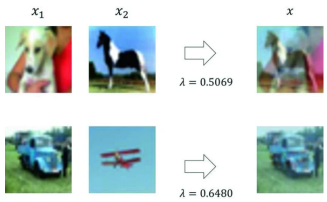

Figure 5: Mixup CIFAR-10 Training[3]

CutMix Training Augmentation

CutMix, introduced by Yun et al.[4], is an augmentation strategy used improve the performance of Convolutional Neural Networks (CNNs) by overcoming the limitations of traditional regional dropout methods. Unlike techniques such as Cutout and Mixup, which either remove portions of an image or blend entire images together, CutMix replaces randomly selected regions of an image with patches from another image. This approach retains the advantages of regional dropout, such as encouraging a model to focus on less discriminative features, while also making efficient use of all pixels in the dataset. The ground truth labels are also mixed proportionally to the number of pixels of combined images.

This process encourages the model to develop stronger generalization capabilities, as it must learn to recognize objects from partial views and varying contexts.

Mathematically, we can represent the combining operation as:

$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$
$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B$$

where $\mathbf{M} \in \{0, 1\}^{W \times H}$ denotes a binary mask indicating where to drop out and fill in from two images, $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is element-wise multiplication.

($\tilde{x}$ and $\tilde{y}$) is the training sample generated by combining the training samples (x_A, y_A) and (x_B, y_B)

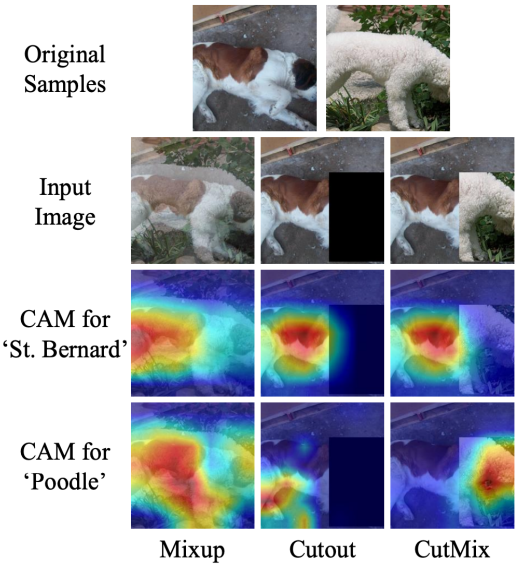

Figure 6: Class activation mapping (CAM) visualizations on 'Saint Bernard' and 'Miniature Poodle' samples using various augmentation techniques. [4]

Results and Analysis

The capabilities of the GhostNet CutMix/Mixup Residual Network (GhostMixNet) were evaluated on the aforementioned CIFAR-10 dataset, with consideration given to various configurations of methodologies, optimizations, and schedulers.

Model Desc.	Parameters	Val Acc.	Test Acc.
ResNet	4.6M	91.4%	78.3%
ResNet+SE+ Image Transforms	4.9M	94.3%	83.5%
GhostNet + SE+ Image Transforms	3.0M	95.1%	84%
GhostNet + SE+ Image Transforms+ CutMix/Mixup	3.0M	95.5%	85.4%

The initial ResNet architecture, with 4.6M parameters (a reduction from ResNet18, which has 11M), achieved a validation accuracy of 88.2% over 50 epochs, but a Test Accuracy of only 78.3%. This model was improved upon by adding SE blocks within each residual block, and the basic image transforms as discussed above. This increased the val accuracy considerably to 94.3%, but the test accuracy failed to see the same increases, likely as a result of overfitting to the training set. Further research into the causes of overfitting led to the reduction of parameters through the implementation of GhostNet, as well as later additions of the CutMix and Mixup augmentations.

These results demonstrate that the proposed architecture is suitable for real-world applications where computational power is limited. The methodologies used within GhostMixNet each benefit the model in a distinct way, and give it the ability to generalize to a wide range of test data.

References

[1] Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; and Xu, C. 2020. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1580–1589.

[2] Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-Excitation Networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[3] Oki, H.; and Kurita, T. 2019. Mixup of Feature Maps in a Hidden Layer for Training of Convolutional Neural Network. In *Lecture Notes in Computer Science()*, 635–644. Cham: Springer.

[4] Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *arXiv preprint arXiv:1905.04899*.

[5] Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.