Image Classification on CIFAR10 & CIFAR100 With CutMix, CutOut and MixUp Data Augmentations

Dzodzoenyenye Adjowa Senanou, Prince Mensah, Najlaa Mohamed, Leonard Sanya
African Masters for Machine Intelligence (AMMI)
dsenanou@aimsammi.org, pmensah@aimsammi.org
nmohamed@aimsammi.org, lsanya@aimsammi.org

1 Introduction

Image classification is crucial in computer vision applications like autonomous driving and medical diagnosis. As convolutional neural networks (CNNs) grow more powerful, they also become more prone to overfitting and poor generalization on unseen data. Regularization strategies such as dropout and data augmentation combat these issues. Data augmentation improves CNN robustness by introducing variability and noise into the training process, making models less dependent on specific visual features and better at leveraging the full context of input data.

In this project, we adapted the AlexNet model, a pioneering deep learning architecture, for the CIFAR datasets. AlexNet's convolutional layers extract spatial features, followed by fully connected layers for classification. Despite its success on large-scale datasets like ImageNet, adapting AlexNet to the smaller CIFAR datasets presents challenges due to the smaller image size and potential overfitting.

This work evaluates CutMix, CutOut, and MixUp on CIFAR-10 and CIFAR-100 to assess their effectiveness in improving image classification. Each technique is applied to augment training samples, and the augmented images are used to train models, followed by performance evaluation.

2 AlexNet Architecture

AlexNet, introduced in [1], was the pioneering CNN architecture trained on GPUs, significantly enhancing training performance. It features 5 convolutions and 3 fully connected layers. The architecture includes overlapping max pooling layers and ends with a softmax classifier for 1000 classes. AlexNet architecture (figure 1) was trained on ImageNet, with 60M parameters and 650,000 neurons, revolutionizing image classification with deep learning.

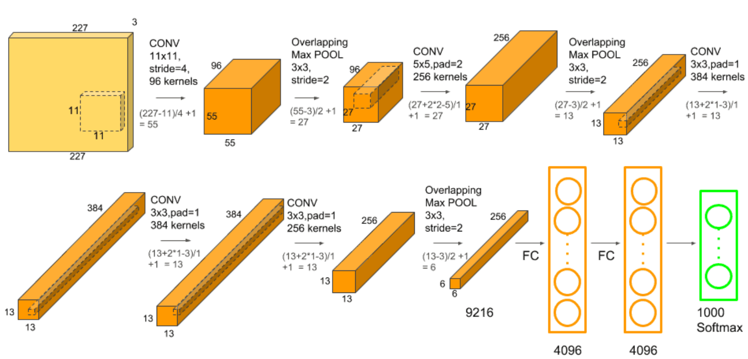


Figure 1: AlexNet model Architecture (Credits: Sunita Nayak)

2.1 ReLU Activation Function, Local Response Normalization (LRN), Max-Pooling

The authors illustrated the superiority of the Rectified Linear Unit (ReLU) over Tanh and sigmoid activations in training deeper networks. Experimentation on the CIFAR10 dataset (Figure 2) reveals that with ReLU (solid line) achieves a 25% error rate 6 times faster than Tanh (dotted lines). For consistency with AlexNet, our experiment utilizes it in Section 3 .

The local response normalization (LRN) technique was applied in the first and second convolutional layers to enhance model generalization during training. The formulation of LRN is represented by Equation (1):

$$b^i_{x,y} = a^i_{x,y} \Big/ \left(k + \alpha \sum_{j=\max(0,i-\frac{n}{2})}^{\min(N-1,i+\frac{n}{2})} (a^j_{x,y})^2 \right)^\beta \tag{1}$$

Here, $b^i_{x,y}$ denotes the LRN output of a neuron at position (x, y) in kernel i, while $a^i_{x,y}$ represents the output from the activation function of the same neuron. Parameters k, α, and β adjust the normalization, and N denotes the kernel depth.

Pooling layers in convolutional neural networks summarize outputs within each kernel map. Throughout the network, overlapping pooling (refer to Figure **??**) was utilized by setting a stride smaller than the kernel size. Our task, implemented entirely from scratch, adheres to these techniques.

2.2 Strategies for Preventing Overfitting

Training a model with 60 million parameters provides significant representation power, but this can lead to overfitting and poor generalization [2]. To mitigate this, two primary strategies are commonly employed:

2.2.1 Data Augmentation

Increasing the dataset size helps reduce overfitting. Various augmentation techniques are utilized, such as image translation, horizontal reflections, and adjustments to RGB channel intensities.

2.2.2 Dropout

During training, Dropout randomly sets the output of hidden neurons to zero with a certain probability (see Figure 4), preventing over-reliance on specific neurons. All neurons contribute during testing.

3 Enhancing Model Performance: MixUp, CutOut, CutMix

We implemented three additional data augmentation techniques—CutOut [2], MixUp [3], and CutMix [4]—to assess their impact on model classification accuracy.

CutOut: This CNN regularization method involves randomly masking out regions of input images (refer to Figure 5), promoting robust feature learning by the network.

MixUp: Generates virtual training examples by linearly interpolating between pairs of examples and their labels (refer to Figure 5). Mathematically, this can be represented as:

$$\begin{aligned} \hat{x} &= \lambda x_i + (1-\lambda) x_j \\ \hat{y} &= \lambda y_i + (1-\lambda) y_j \end{aligned} \tag{2}$$

where x_i, x_j are raw inputs vectors y_i, y_j are one-hot label encodings and $(x_i, y_i), (x_j, y_j)$ are two randomly sampled examples from the training set. λ is a hyperameter between 0 and 1.

CutMix: Combines pairs of training samples by cutting and pasting patches, with ground truth labels mixed proportionally to the area of patches. Given samples (x_a, y_a) and (x_b, y_b), CutMix generates a new training sample $(\hat{x}, \hat{y})$, which is utilized for model training. The CutMix operation is defined as

$$\begin{aligned} \hat{x} &= M \odot x_a + (1-M) \odot x_b \\ \hat{y} &= \lambda y_a + (1-\lambda) y_b \end{aligned} \tag{3}$$

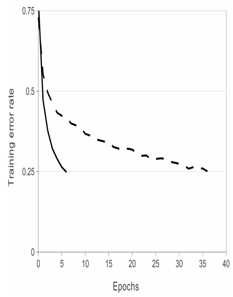


Figure 2: AlexNet with ReLU

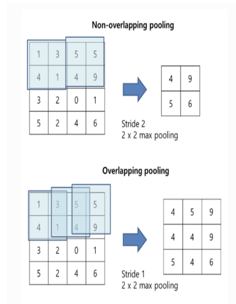


Figure 3: Overlapping Pooling

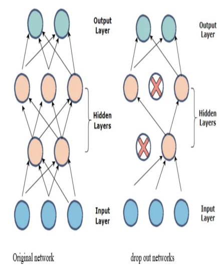


Figure 4: Dropouts

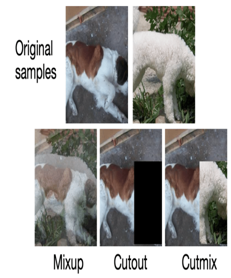


Figure 5: Data augmentation [5]

4 Experiments and Results

This section summarizes our implementation strategies and findings based on the AlexNet architecture. We trained and evaluated our model extensively on both CIFAR10 and CIFAR100 datasets, focusing on the effects of various augmentation techniques.

4.1 AlexNet Trained on CIFAR10 & CIFAR100

Following the paper's framework, we constructed the AlexNet architecture from scratch and trained it on CIFAR10 and CIFAR100 datasets. Despite challenges with the initialization scheme suggested by the authors, which initially led to poor performance and generalization, we experimented with two approaches: disabling the scheme and using the default initialization, and implementing the Kaiming He initialization. Our comparison showed no significant difference, so we adopted the default scheme.

CIFAR10 consists of 10 classes with 600 images each, while CIFAR100 has 100 classes grouped into 20 superclasses, also with 600 images per class. Both datasets contain 32x32 color images split into 50000 training and 10000 test samples. After training our model for 20 epochs with default augmentations, we evaluated performance using Top-1 and Top-5 accuracy metrics. Detailed results are presented in Table 1.

	Top-1 Errors		Top-5 Errors	
Datasets	Our Approach	Pre-trained	Our Approach	Pre-trained
CIFAR-10	14.65%	10.93%	0.72%	0.21%
CIFAR-100	44.14%	32.58%	16.03%	8.96%

Table 1: Comparisons of Top 1 and Top 5 Errors

The summary in Table 1 compares our scratch-built model with a pre-trained AlexNet on ImageNet, despite the low resolution of CIFAR datasets. Our models achieved 14.65% and 44.14% Top-1 error rates on CIFAR10 and CIFAR100, and 0.72% and 16.03% Top-5 error rates respectively.
The pre-trained model consistently outperformed our model, benefiting from learning features on higher-resolution data.

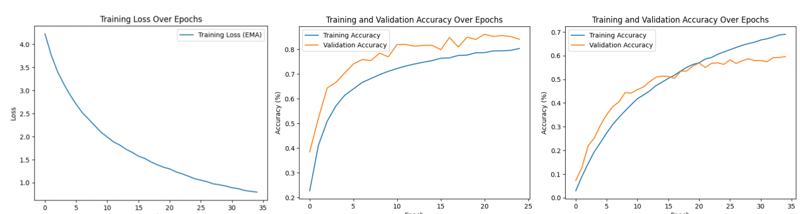


Figure 6: Training Curves

After implementing the proposed data augumentation techniques [2, 3, 4], we incorporate them in the training and evaluate the model performance using Top-1 and Top-5 accuracy metrics. Detailed results are presented in Table 2.

	Top-1 Errors			Top-5 Errors			Accuracy		
Datasets	CutOut	MixUp	CutMix	CutOut	MixUp	CutMix	CutOut	MixUp	CutMix
CIFAR10	14.78%	13.12%	14.90%	0.90%	1.00%	0.81%	85.39%	87.27%	85.10%
CIFAR100	44.15%	41.04%	53.43%,	15.86%	15.78%	23.28%	56.37%	58.98%	46.57%

Table 2: Comparisons of MixUp, CutOut and CutMix error rates

Based on the summary in Table 2 on the model evaluation when trained using the Cutout, Cutmix and Mixup, we notice a significance change in the Top-1 error rate when using the Mixup data augmentation. Our models achieved 13.12% and 41.04% Top-1 error rates on CIFAR10 and CIFAR100,respectively, as opposed to 14.65% and 44.14% Top-1 error rates on CIFAR10 and CIFAR100, respectively, when trained without the Mixup. Cutmix data augmentation seems not to perform well on CIFAR100 as we notice an increase in the error rates 53.43% and 23.28% for Top-1 and Top-5. In general, MixUp, CutOut and CutMix techniques seems to perform better in terms of accuracy and error rates on CIFAR10 compared to CIFAR100 dataset.

4.2 Investigating the Effects of Augmentation Techniques on CIFAR-10/100 Images

4.2.1 Experiment 1: Adapting AlexNet for CIFAR-10 and CIFAR-100 with Original ImageNet Augmentations

In this experiment, we modified AlexNet for CIFAR-10 and CIFAR-100 by adjusting the architecture with 3x3 kernel sizes and adding adaptive average pooling for 32x32 resolution images. Original AlexNet data augmentation techniques such as random cropping, horizontal flipping, and PCA color augmentation were applied. However, we encountered challenges where augmented images (see Figure 7 and Figure 8) often contained unlearnable features and appeared predominantly black. This issue stemmed from:

- **Strong PCA Augmentation:** The aggressive PCA color augmentation on low-resolution CIFAR images caused significant distortions, making the images difficult to interpret.
- **Padding and Cropping:** Random cropping with padding introduced large black areas, impeding effective feature learning.

Training was conducted for 200 epochs using SGD optimizer (initial learning rate: 0.1, momentum: 0.9, weight decay: 5e-4) and a ReduceLROnPlateau scheduler. Despite these augmentations, challenges with image quality notably impacted the training process.

4.2.2 Experiment 2: Refining PCA Color Augmentation for CIFAR-10 and CIFAR-100

Following issues observed in Experiment 1, where PCA color augmentation resulted in unrecognizable features and predominantly black images, further refinement was necessary to control color intensity.

Modifications made to PCA color augmentation included:

- **Clipping**: Ensuring pixel values stayed within the valid range (0-255).
- **Scaling Down Perturbation**: Reducing the perturbation by lowering the standard deviation (alpha_std=0.1) of random values.

Despite these adjustments and maintaining experiment 1 settings, the outcomes remained unchanged, with images still exhibiting similar issues as observed in Experiment 1.

4.2.3 Experiment 3: Baseline Training Without Augmentation

This experiment aimed to establish a baseline performance of an AlexNet model on CIFAR-10/CIFAR-100 datasets without any augmentation, following challenges observed in previous experiments with PCA color augmentation.

The model achieved a high training accuracy of 99.47% but exhibited significant overfitting, reflected in a validation accuracy of 62.96% on CIFAR-100 (Figure 9) and 82.09% Training Accuracy, while Validation was at 83.53% (Figure 10).

This disparity in accuracies highlights the critical role of augmentation techniques in improving generalization and reducing overfitting, aiming to enhance overall model performance and robustness.

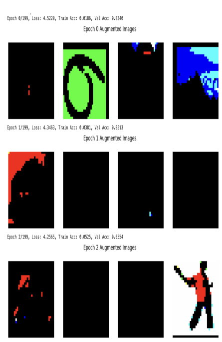


Figure 7: Adapted AlexNet With Original Augmentation on Cifar100

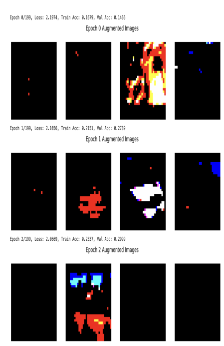


Figure 8: Adapted AlexNet With Original Augmentation on Cifar10

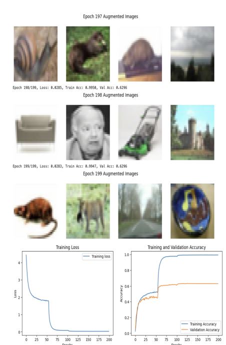


Figure 9: Baseline Training Without Augmentation on Cifar100

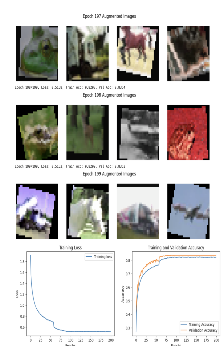


Figure 10: Baseline Training Without Augmentation on Cifar10

4.2.4 Experiment 4: Regularization using Basic Augmentation Techniques on Baseline Model

To increase training data variability while preserving essential features, to address overfitting observed in Experiment 3, we introduced basic augmentation techniques including random horizontal flips, rotations, cropping, and color jitter (see Fig 11 and 12). The model showed training accuracies of 52.85% and 99.99%, and validation accuracies of 51.56% and 87.01% over 150 epochs for CIFAR-100 and CIFAR-10 respectively. This suggests that while overfitting was mitigated, these augmentations alone may not sufficiently enhance generalization, especially on the more challenging CIFAR-100 dataset.

Additionally, a variant using only horizontal flipping and rotation (see Fig 13 and 14) resulted in training accuracies of 70.69% and 90.41%, and validation accuracies of 51.92% and 83.42% over 150 epochs for CIFAR-100 and CIFAR-10 respectively.

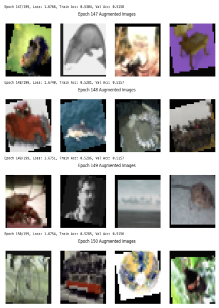


Figure 11: Regularizing Baseline Model on Cifar100

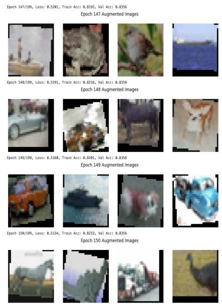


Figure 12: Regularizing Baseline Model on Cifar10

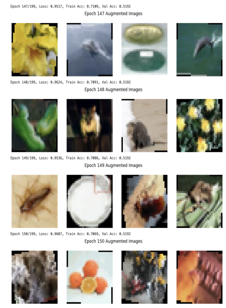


Figure 13: Regularizing Baseline Model on Cifar100

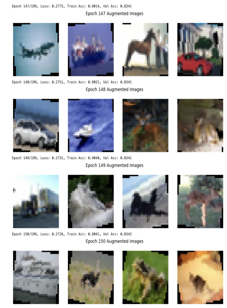


Figure 14: Regularizing Baseline Model on Cifar10

Datasets	Exp 3 (No Augmentation)	Exp 4 (RHF, R, C, CJ)	Exp 4 (HFR)
CIFAR-100	(*): 99.47% , (**): 62.96%	(*): 52.85% , (**): 51.56%	(*): 70.69% , (**): 51.92%
CIFAR-10	(*): 82.09%, (**): 83.53%	(*): 99.99%, (**): 87.01%	(*): 90.41%, (**): 83.42%

RHF:Random Horizontal Flips, HFR:Horizontal Flipping and Rotation, R: Rotations, C: Cropping, CJ:ColorJitter.(*):Train Accuracies, (**):Validation Accuracies.

Table 3: Results of Experiments 3 and 4 on CIFAR-100 and CIFAR-10.

In Table 3, Experiment 3 demonstrated significant overfitting on CIFAR-100, with high training accuracy but low validation accuracy. Experiment 4, first part, introduced basic augmentation techniques which mitigated overfitting and improved results for CIFAR-10, but did not significantly enhance CIFAR-100's generalization. In the second part of Experiment 4, simpler augmentations slightly improved CIFAR-100's training accuracy, yet validation accuracy remained low. Thus, the effectiveness of augmentation techniques varies with dataset complexity.

5 Conlusion

In this project, we provided a comprehensive review of the AlexNet architecture proposed by Alex et al. (2012) [1]. We constructed the AlexNet architecture from scratch, incorporating the data augmentation techniques suggested in their paper. We conducted an evaluation of the model's performance using the CIFAR10 and CIFAR100 datasets. Additionally, we investigated the effects of cutout, cutmix, and mixup data augmentation techniques on the model's generalization abilities.

Our findings indicate that the model demonstrates strong generalization capabilities on CIFAR10 in comparison to CIFAR100. Furthermore, we observed a significant improvement in the Top-1 error rate when employing the mixup data augmentation technique on both CIFAR10 and CIFAR100 datasets.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

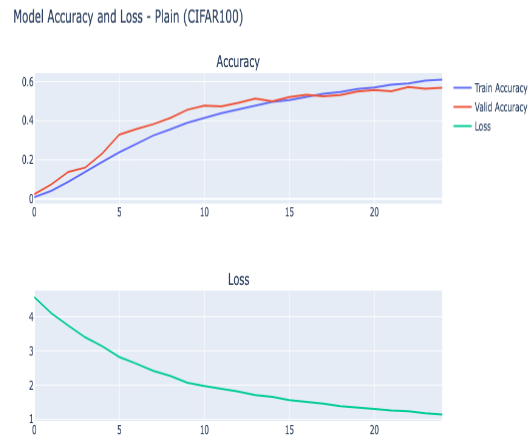
[3] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

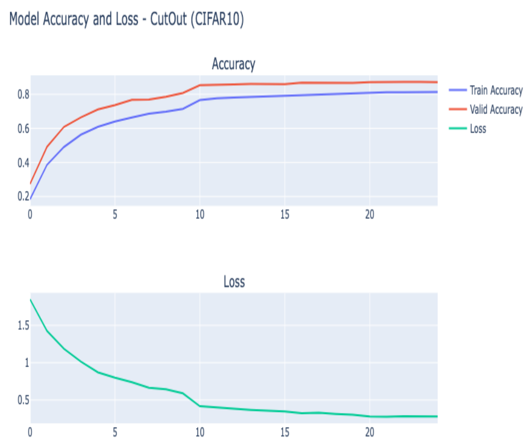
[4] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[5] Zihan Yang, Richard Sinnott, Qiuhong Ke, and James Bailey. Individual feral cat identification through deep learning. pages 101–110, 12 2021.

Appendix


Model Accuracy and Loss - Plain (CIFAR10)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss


Model Accuracy and Loss - Plain (CIFAR100)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss


Model Accuracy and Loss - CutOut (CIFAR10)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss


Model Accuracy and Loss - CutOut (CIFAR100)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss


Model Accuracy and Loss - MixUp (CIFAR10)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss


Model Accuracy and Loss - MixUp (CIFAR100)
Accuracy
Train Accuracy
Valid Accuracy
Loss
Loss