Deep Learning with Noisy Labels: Exploring Architectures and Techniques

Dodun-des-Perrieres Dan
Faculty of Computer Science
Alexandru Ioan Cuza University)
Iasi, Romania
dandodun@gmail.com

Luca Năstasă Baraș
Faculty of Computer Science
Alexandru Ioan Cuza University
Iasi, Romania
lucanastasa@gmail.com

***Abstract*—This study investigates robust learning techniques for classification tasks involving noisy labels using the CIFAR-100N dataset, where 40% of training labels are corrupted. Several state-of-the-art deep learning architectures were evaluated, including VGG16, ResNet18, ResNet34, Vision Transformers, and Wide ResNet-28, along with noise-robust strategies such as Co-Teaching, label smoothing, MixUp, and CutMix. ResNet34 emerged as the most effective model, achieving a validation accuracy of 74.78% after hyperparameter optimization and extended training. Vision Transformers, while promising for capturing global image relationships, underperformed on this dataset due to their complexity and the limited size of CIFAR-100, achieving only 47.50% accuracy after optimization. Co-Teaching showed potential by filtering noisy samples dynamically, achieving 69.32% accuracy. These results highlight the importance of architectural and training adaptations in handling noisy labels and suggest promising directions for future work, including further exploration of Vision Transformers and hybrid approaches combining Co-Teaching with advanced architectures.**

I. Introduction

A. Context and Background

Mislabeled data presents a significant challenge in the development of robust machine learning models, particularly in tasks that involve datasets with fine-grained class distinctions. This study focuses on the CIFAR-N dataset, introduced in "Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations" [1]. The CIFAR-N dataset provides re-annotated versions of CIFAR-10 and CIFAR-100, incorporating real-world human annotation errors. Unlike synthetic noise typically studied in research, the real-world noise in CIFAR-N introduces complex and practical challenges that more accurately reflect the conditions encountered in real-world applications. For this study, the focus is on this CIFAR-100N dataset with 40% mislabeled data, a particularly demanding scenario for effective classification.

B. Motivation

This study aims to design a model that can achieve high accuracy and maintain consistent predictions, even when a substantial portion of the training data is mislabeled. In real-world applications, mislabeled data is often unavoidable due to human error and the inherent complexity of annotation tasks, thus developing models that can withstand such noise is essential for reliable deployment and usage in practical scenarios. Through systematic exploration of various deep learning architectures and advanced training techniques, this work seeks to identify robust solutions to address the challenges posed by label noise in the CIFAR-100N dataset.

C. Contributions

This report offers the following contributions:

- A comprehensive evaluation of multiple model architectures, including VGG16, ResNet18, Vision Transformers, Wide ResNet-28, and a co-teaching method, applied to the CIFAR-100N dataset.
- Integration and assessment of techniques such as label smoothing, CutMix, and MixUp, to enhance model robustness and generalization.
- Identification of an optimal model configuration using a ResNet34 architecture, fine-tuned with a grid search to determine the most effective hyperparameters.

D. Report Structure

The remainder of this report is structured as follows:

- **Section 2**: Related work, providing an overview of prior studies on learning with noisy labels and robust training methods.
- **Section 3**: Methodology, detailing the dataset and model architectures.
- **Section 4**: Experimental setup, results and ablation study, presenting the performance of various models and techniques, discussion, analyzing the results and their implications for robust learning in noisy conditions.
- **Section 5**: Conclusion, summarizing the key findings and proposing directions for future research.

II. Related Work

A. Learning with Noisy Labels

Training machine learning models with noisy labels is a critical challenge, as mislabeled data can lead to degraded model generalization and overfitting. This is particularly problematic in fine-grained datasets, where subtle distinctions between classes make label noise highly disruptive.

Several strategies have been proposed to address this issue. Robust loss functions, such as the Mean Absolute Error (MAE), reduce sensitivity to outliers by assigning equal weight to all samples, unlike cross-entropy loss which amplifies the

impact of noisy labels [2]. Data augmentation techniques like MixUp [3] and CutMix [4] have also proven effective in mitigating label noise. MixUp generates interpolated samples by combining pairs of images and their labels, encouraging smoother decision boundaries. CutMix, on the other hand, augments training data by cutting and pasting image regions with corresponding label mixing, forcing the model to focus on meaningful image features.

B. Co-Teaching for Robust Training

Co-teaching [5] is a robust training framework specifically designed for learning with noisy labels. It leverages the observation that deep networks tend to fit clean samples earlier in training before memorizing noisy labels. In this method, two neural networks are trained simultaneously, each selecting a subset of potentially clean samples to train the other. This cross-updating process helps filter out noisy samples and maintains robust learning under high noise conditions. Co-teaching has demonstrated strong performance in benchmarks such as noisy CIFAR datasets, making it a widely adopted approach in noisy label scenarios.

C. Applications to CIFAR Datasets

The CIFAR-10 and CIFAR-100 datasets are standard benchmarks for evaluating robust learning methods under label noise. While synthetic noise is commonly used for controlled experiments, it often lacks the complexity of real-world errors. To bridge this gap, the CIFAR-N dataset [1] was introduced, containing human annotation errors that more accurately reflect practical challenges. These errors often correlate with visually similar classes, creating a more realistic benchmark for noisy label studies.

In noisy CIFAR datasets, methods such as Co-teaching, MixUp, and CutMix have demonstrated notable improvements in robustness and accuracy. Co-teaching effectively handles extreme noise, while MixUp and CutMix provide strong generalization by regularizing the decision boundaries and enhancing feature learning. Thus, these approaches can form a solid foundation for addressing noisy label challenges in CIFAR and other similar datasets.

III. Methodology

A. Dataset

As stated previously, this study utilizes a dataset based on CIFAR-100, a widely used benchmark for image classification tasks, comprising 60,000 color images of 32x32 pixels. The dataset is divided into 50,000 training images and 10,000 testing images across 100 fine-grained classes, with each class containing 500 training samples and 100 test samples. The fine-grained nature and relatively small sample size per class make CIFAR-100 a challenging dataset for classification tasks.

The CIFAR-100N dataset [1] is a re-annotated version of CIFAR-100 that incorporates real-world human annotation errors collected through Amazon Mechanical Turk. Unlike synthetic noise models, these errors introduce realistic, instance-dependent noise patterns that often align with human tendencies, such as mislabeling "snake" as "worm" due to visual similarities. This makes CIFAR-100N a more practical benchmark for evaluating model robustness. In this study, we specifically use the subset with 40% noisy labels, where a substantial portion of the training labels are incorrect, posing a challenging testbed for robust learning methods.

B. Data Augmentation

To improve robustness to noisy labels and enhance generalization, a variety of data augmentation techniques were employed during training. These augmentations were selected to increase the diversity of the training data and reduce the risk of overfitting. However, not all augmentations were applied uniformly across all models; some augmentations were tailored to specific models based on their characteristics and training needs. Some of the techniques used are detailed below:

1) Standard Augmentations: The following standard augmentations were applied in varying combinations:

- **Random Horizontal Flip**: Images were randomly flipped horizontally with a 50% probability to introduce invariance to orientation.
- **Random Crop**: Random cropping was used to simulate zoom and scale variations by cropping a random portion of the image and resizing it back to the original dimensions.
- **Random Rotation**: Images were rotated by a random angle within a specified range (e.g., $\pm 15^{\circ}$), adding robustness to slight rotations in the dataset.
- **Color Jitter**: Adjustments to brightness, contrast, saturation, and hue were made to simulate variations in lighting conditions.

2) Advanced Augmentations: Advanced data augmentation techniques were employed to enhance model robustness under noisy label conditions. These methods were selectively applied depending on the model's characteristics:

- **MixUp [3] and CutMix [4]**: MixUp generates virtual training samples by linearly blending pairs of images and their labels, smoothing decision boundaries and mitigating overfitting. CutMix augments data by cutting and pasting patches between images while mixing their labels proportionally, encouraging models to focus on multiple image regions. Both methods were used to reduce the impact of label noise by improving generalization.
- **AutoAugment** [6]: AutoAugment, using the CIFAR10 policy, dynamically selects augmentation sequences such as shear, translate, and color transformations to optimize performance. While the policy was originally tailored to CIFAR10, it can provide improvements on CIFAR100 due to the shared structural similarities between the datasets.

By trying to tailor the augmentation strategies to individual models, we ensured that each model leveraged augmentations best suited to its architecture and training dynamics. Most of these augmentation strategies were selected empirically

through trial and error, allowing for a practical assessment of their impact. This approach facilitated a systematic evaluation of how different augmentations contribute to improving robustness against noisy labels.

C. Models Explored

To evaluate the impact of noisy labels in CIFAR-100N, a diverse set of models was selected, each offering unique strengths for handling noise. These models are as follows:

- **VGG16**: This model served as the baseline for the study. Its straightforward architecture provides a reference point for comparing the performance of more advanced models, making it ideal for assessing the challenges posed by noisy labels in a traditional convolutional network [8].
- **ResNet18**: A compact residual network pretrained on ImageNet, chosen for its efficiency and ability to extract hierarchical features. Its smaller size makes it well-suited for datasets like CIFAR-100 [9].
- **Wide ResNet-28**: This wider variant of ResNet enhances the model's capacity to capture complex patterns, particularly in fine-grained datasets. Its architecture allows it to handle more detailed features, which can be advantageous for mitigating the effects of noisy labels [10].
- **Vision Transformers (ViT)**: Included to explore the potential of attention-based mechanisms in noisy label scenarios, ViT represents a departure from convolutional models by leveraging self-attention for feature extraction [11].
- **ResNet34**: A deeper residual network pretrained on ImageNet, selected for its balance of depth and generalization capabilities. Its enhanced feature extraction makes it a strong candidate for managing label noise effectively [9].
- **Co-Teaching Framework**: A noise-robust training method in which two networks are trained simultaneously, each selecting clean samples for the other to train on. This collaborative filtering strategy is designed to improve resilience to noisy labels by leveraging early learning dynamics [5].

D. Training Pipeline

The training pipeline was designed to maximize robustness under 40% noisy label conditions. Key components of the pipeline were chosen empirically through initial evaluations.

1) Data Augmentation: As mentioned before, the data augmentation techniques were tailored to each model. Standard augmentations like random horizontal flip, random crop, and random rotation were consistently applied. However, for stronger models, a phased approach was used: MixUp was applied during the first half of training to smooth decision boundaries and reduce overfitting to noisy labels, allowing the model to build a noise-tolerant foundation early on. This was followed by CutMix in the second half, which encouraged the model to focus on key discriminative features as training progressed, leveraging its improved ability to identify important image regions.

2) Loss Function and Label Smoothing: Cross-entropy loss with label smoothing was employed to handle noisy labels. A smoothing factor of 0.1 redistributed 10% of the probability mass across incorrect classes, preventing overconfidence and improving generalization. This approach helped mitigate the adverse effects of label noise.

3) Optimizer and Scheduler: The optimizer and learning rate scheduler were empirically chosen for each model based on initial evaluations. One of the most commonly effective combinations was the AdamW optimizer paired with a cosine annealing learning rate scheduler. This setup provided stability in training and allowed for efficient exploration of the parameter space, with the learning rate gradually decreasing to fine-tune the model as training progressed.

4) Batch Size: A batch size of 128 was discovered to be the best chosen as the trade-off between computational efficiency, memory usage, and preserving the dataset's noisy label distribution, thus keeping the training process manageable in terms of resource demands while keeping sufficient variability in each training iteration. The 256 batch size was also used.

5) Evaluation Strategy: During training, models were evaluated periodically on the validation set to monitor their performance and ensure consistent comparisons across experiments. Validation accuracy was used as the primary metric to guide model selection and fine-tuning.

E. Hyperparameter Tuning

Hyperparameter tuning played a critical role in optimizing model performance under noisy label conditions. After evaluating multiple models, ResNet34 emerged as one of the most promising architectures due to its balance of depth and robustness. To further refine its performance, a grid search was conducted to identify the best hyperparameter configuration. The specifics of this process, including the parameter space explored, are detailed in the following sections.

IV. Experimental setup and Results

A. Baseline Model: VGG16

To establish a baseline for comparison, we trained a VGG16 model on the CIFAR-100N dataset with 40% noisy labels. The VGG16 architecture, despite being an older model, provides a useful reference point due to its relatively simple structure and historical significance in image classification tasks.

1) Experimental Setup: The VGG16 model was trained using the SGD optimizer with Nesterov momentum, a learning rate of 0.01, and a CosineAnnealingLR scheduler. A batch size of 256 was used to balance computational efficiency and memory usage. Data augmentation techniques, including RandomCrop, RandomHorizontalFlip, MixUp, and CutMix, were applied to improve robustness and generalization. These augmentations, previously described, were selected based on their effectiveness in addressing noisy labels.

2) Results and Observations: The VGG16 model achieved a validation accuracy of **59.75%**, serving as a baseline for

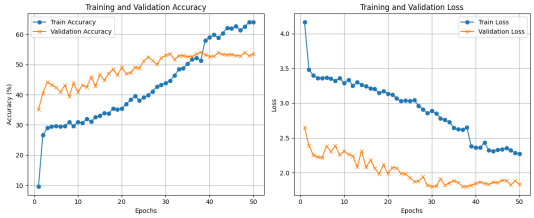

Fig. 1. Training and Validation Accuracy/Loss for VGG16

comparison with more advanced architectures. The total training time was **705.77 seconds**, while peak memory usage during training reached **2632.97 MB**.

These results highlight the limitations of the VGG16 model, which struggles to achieve high accuracy due to its shallow architecture. While augmentations like CutMix and MixUp offered some improvement, the baseline performance emphasizes the need for more robust and modern architectures.

B. ResNet18

Following the baseline experiments with VGG16, we explored ResNet18, a deeper convolutional network that introduces residual connections. These connections help mitigate the vanishing gradient problem, enabling effective training of deeper architectures. ResNet18 was chosen due to its balance of computational efficiency and representational power, making it a natural progression from the shallower VGG16 model. Additionally, its ability to model hierarchical features aligns well with the complex patterns present in CIFAR-100N.

1) Experimental Setup: Two configurations of ResNet18 were evaluated to assess the impact of pretraining:

- **Non-Pretrained ResNet18:** The model was trained from scratch using the SGD optimizer with Nesterov momentum, a learning rate of 0.01, and the CosineAnnealingLR scheduler. Data augmentations include RandomCrop with padding, RandomHorizontalFlip, MixUp, and CutMix. A batch size of 256 was used for training.
- **Pretrained ResNet18:** The model, initialized with weights pretrained on ImageNet1k, was fine-tuned using the AdamW optimizer with a learning rate of 0.001 and the CosineAnnealingLR scheduler. To align with the pretrained weights, images were resized to 224x224. The same augmentations and batch size as the non-pretrained configuration were applied.

2) Results and Observations: The non-pretrained ResNet18 achieved a validation accuracy of **54.47%**, a modest improvement over the VGG16 baseline, showcasing the benefits of deeper architecture and residual connections. In contrast, the pretrained ResNet18 significantly outperformed it with a validation accuracy of **71.46%**, leveraging features learned from large-scale data and benefiting from the resized 224x224 input.

The pretrained model required **244 seconds** of training time and **7679.56 MB** of peak memory, reflecting the additional computational demands of fine-tuning a larger model. This increase in resource usage can also be attributed to the higher input resolution of 224x224, which aligns with the pretrained ImageNet1k weights but significantly exceeds the original 32x32 resolution of CIFAR images. Figures 2 and 3 illustrate the superior convergence and performance of the pretrained model compared to the slower progress of the non-pretrained version.

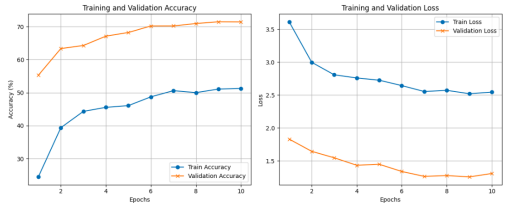

Fig. 2. Training and Validation Accuracy/Loss for Pretrained ResNet18.

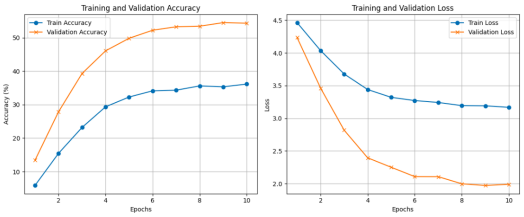

Fig. 3. Training and Validation Accuracy/Loss for Non-Pretrained ResNet18.

These results highlight the advantages of ResNet18 over VGG16, particularly when leveraging pretrained weights. The pretrained ResNet18's ability to generalize better under noisy label conditions underscores the effectiveness of transfer learning and the importance of aligning input data with the pretrained model's training configuration.

C. Vision Transformer

Vision Transformers (ViTs) are significantly more complex models compared to traditional architectures like VGG16 and ResNet. Their reliance on self-attention mechanisms enables them to model long-range dependencies and global relationships across different regions of an image, making them appealing for tasks involving noisy labels where such context is critical. Additionally, ViTs often benefit from pretraining on large-scale datasets like ImageNet, providing them with a strong foundation for feature extraction.

In this experiment, we fine-tuned a pretrained Vision Transformer using the SGD optimizer with a learning rate of 0.001. Standard augmentations, including `RandomCrop(224, padding=4)` and `RandomHorizontalFlip`, were applied to enhance generalization. A batch size of 256 was used to maintain training efficiency. Initially, this setup achieved a validation accuracy of **29.84%**.

To address the poor initial performance, we introduced a warmup phase for the learning rate. During this phase, the learning rate begins at a low value and gradually increases over the first few epochs before stabilizing. This prevents

large updates to model parameters early in training, reducing instability and helping the optimizer converge effectively. With this modification, the validation accuracy improved to **47.50%**.

Despite this improvement, the performance of ViTs remained lower than that of simpler architectures like ResNet34. This is likely due to the size and complexity of ViTs, which require significantly larger datasets to fully utilize their capacity. The relatively small size of CIFAR-100, combined with the added challenge of noisy labels, may have hindered the model's ability to generalize effectively. These results suggest that while ViTs offer advanced capabilities, their application in noisy label scenarios with limited data requires additional regularization, architectural modifications, or pretraining on task-specific datasets to unlock their full potential.

D. Wide ResNet-28

Wide ResNet-28x2 was explored as a natural extension to ResNet18, given its increased width, which offers a larger capacity for feature learning. The model was trained using the AdamW optimizer with a learning rate of 0.001, the CosineAnnealingLR scheduler, and augmentations including RandomCrop, RandomHorizontalFlip, CutMix, and MixUp, with a batch size of 256.

Despite these efforts, the model achieved a validation accuracy of only **43.02%**, falling short of the pretrained ResNet18. The lower performance may be attributed to its reduced depth and limited tuning of hyperparameters. This outcome suggests that Wide ResNets require further optimization to fully leverage their potential in noisy label scenarios.

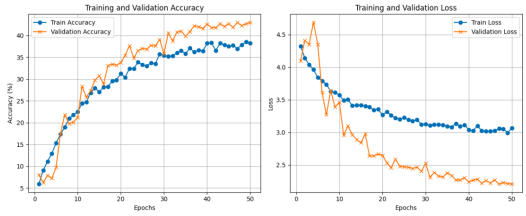

Fig. 4. Training and Validation Accuracy/Loss for Wide ResNet-28x2.

E. Co-Teaching

In addition to experimenting with single-model architectures, we implemented the **Co-Teaching** framework, a method designed to address noisy labels by training two neural networks simultaneously. This approach leverages the observation that neural networks tend to learn easier (likely clean) examples earlier in training, before gradually overfitting to noise. By exchanging information about potentially clean samples, Co-Teaching helps filter out noisy labels, thereby improving robustness.

1) Implementation Details: The Co-Teaching framework was implemented using two instances of **ResNet18** pretrained on ImageNet. During each training step, both models predicted outputs for the same batch of inputs, and a custom loss function was used to select the cleanest samples based on their loss values. The key steps in the implementation were:

- **Loss Calculation:** Individual cross-entropy losses were computed for both models, and samples were ranked by their loss values.
- **Forget Rate:** A dynamically decreasing forget rate was applied, starting at **40%** and linearly reduced over epochs, retaining more samples as training progressed.
- **Exchange of Clean Samples:** Each model was trained using the cleanest samples identified by the other model, ensuring robustness against noisy labels.
- **Optimizers and Schedulers:** Both models were optimized using **AdamW** with a learning rate of 0.001 and a weight decay of 1×10^{-4}. Learning rate scheduling was managed using **CosineAnnealingLR**.

2) Experimental Setup: The models were trained for **20 epochs** on inputs resized to 224x224, with standard augmentations such as `AutoAugment`, `RandomHorizontalFlip`, and `RandomRotation`. Due to the two-model structure, the total training time for all epochs was approximately **1480 seconds**.

3) Results and Observations: The Co-Teaching framework produced the following results:

- **Model 1:**
  - Validation Accuracy: **68.45%**
  - Validation Loss: **1.32**
- **Model 2:**
  - Validation Accuracy: **69.32%**
  - Validation Loss: **1.29**

While these results fell below the validation accuracy achieved by ResNet34, Co-Teaching demonstrated strong robustness to noisy labels by dynamically filtering out mislabeled samples.

4) Discussion: Due to hardware and time limitations, the implementation could not be further fine-tuned or extended. With additional resources, such as more epochs, larger batch sizes, or experimentation with more advanced architectures (e.g., deeper ResNets or Vision Transformers), the Co-Teaching framework has the potential to surpass the performance of ResNet34 in noisy label scenarios. Despite these constraints, the observed results highlight the promise of collaborative learning frameworks, especially in cases where label noise significantly impacts performance.

F. ResNet34

Building on the promising results of ResNet18, we extended our experiments to **ResNet34**, a deeper variant with additional residual layers. ResNet34's increased depth offers enhanced feature extraction capabilities, making it a strong candidate for further improving robustness under noisy label conditions.

1) Training Configuration: The ResNet34 training pipeline incorporated several key strategies to handle noisy labels and enhance generalization:

- **Label Smoothing Loss:** A custom loss function with a smoothing factor of 0.1 was used. This function redistributes some probability mass from the ground truth class

to other classes, mitigating overconfidence in predictions and improving robustness to noisy labels [12].

- **Advanced Data Augmentations:** Techniques such as CutMix and MixUp were alternated during training. MixUp was applied in earlier epochs to promote smoother decision boundaries, while CutMix was used in later epochs to encourage the model to focus on discriminative image regions.
- **Hyperparameter Optimization:** A grid search was conducted to identify the optimal combination of optimizer, learning rate, weight decay, and scheduler, ensuring the best possible performance.

2) Grid Search for Hyperparameter Optimization: To identify the optimal training configuration for ResNet34, we performed a **grid search** over the following hyperparameters:

- **Optimizer:** SGD with Nesterov momentum and AdamW.
- **Learning Rate:** $\{0.01, 0.001, 0.0001\}$.
- **Weight Decay:** $\{0.001, 0.0001\}$.
- **Scheduler:** CosineAnnealing, StepLR, ReduceLROnPlateau.

Each combination of these hyperparameters was evaluated over **10 epochs**, resulting in a total of **36 experiments** ($2 \times 3 \times 2 \times 3$). The grid search was conducted using **Weights and Biases (W&B)**, enabling efficient tracking and comparison of all configurations. This systematic approach ensured that the most effective setup for ResNet34 was identified while maintaining reproducibility.

3) Grid Search Visualization: To better understand the performance of different hyperparameter configurations during the grid search, Figure 5 presents the validation accuracy trends for selected runs. Each line corresponds to a unique combination of optimizer, learning rate, weight decay, and scheduler, showcasing the variability in model performance across configurations.

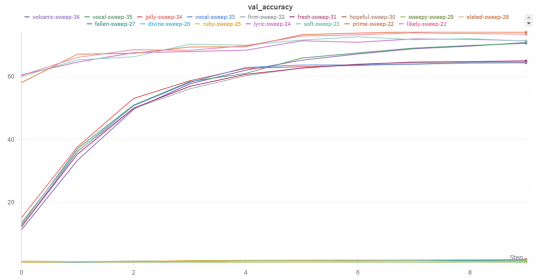

Fig. 5. Validation Accuracy Trends for Selected Grid Search Runs. Each line represents a unique hyperparameter combination.

The full grid search results, including all configurations and metrics, are available on Weights and Biases (W&B) and can be accessed at [7].

4) Best Hyperparameter Configuration: Deft-Sweep-19: The best-performing run from the grid search, labeled **deft-sweep-19**, achieved a validation accuracy of **74.37%**. This run used the following hyperparameters:

- **Optimizer:** AdamW
- **Learning Rate:** 0.001
- **Weight Decay:** 0.001
- **Scheduler:** CosineAnnealingLR

The model was trained for **10 epochs**, with key metrics including:

- **Training Accuracy:** 51.09%
- **Validation Accuracy:** 74.37%
- **Training Loss:** 2.38
- **Validation Loss:** 1.15

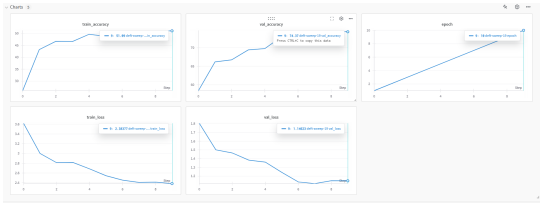

Fig. 6. Performance Metrics for Best Run (Deft-Sweep-19).

5) Final Run with Extended Epochs: Based on the success of the **deft-sweep-19** configuration, the same hyperparameters were used for a final run, with the only modification being an increase in the number of training epochs to **20**. This adjustment was made in the hope of achieving a slight improvement in validation accuracy by allowing the model more time to converge.

Extending the training to 20 epochs resulted in a slight improvement, with the validation accuracy increasing to **74.78%**. This marginal gain suggests that the model's performance was nearing its ceiling, with diminishing returns from additional training. While the improvement is modest, it reinforces the importance of well-optimized configurations to balance training time and accuracy.

G. Discussion of Results

The experiments conducted on various architectures highlight the progression in performance as we explored deeper and more robust models. Each model's results are summarized in Table I, followed by a detailed comparison with the baseline (VGG16).

1) Summary of Results: Table I provides an overview of the validation accuracy, training time, and memory usage for all models and configurations tested in this study.

TABLE I
SUMMARY OF MODEL PERFORMANCE

Model	Configuration	Val Accuracy (%)
VGG16	Baseline	59.75
ResNet18	Non-Pretrained	54.47
ResNet18	Pretrained	71.46
Vision Transformer	Config 1	29.84
Vision Transformer	Config 2	47.50
Wide ResNet-28	—	43.02
Co-Teaching	Model 1	68.45
Co-Teaching	Model 2	69.32
ResNet34	Best Run (10 epochs)	74.37
ResNet34	Extended Run (20 epochs)	74.78

2) Comparison with Baseline (VGG16): The VGG16 baseline served as a starting point for evaluating performance under noisy label conditions. While it achieved a validation accuracy of 59.75%, its shallow architecture and lack of modern design features limited its ability to handle the complexities of noisy CIFAR-100N.

- **ResNet18 (Pretrained)** demonstrated the most significant improvement over the baseline, achieving a validation accuracy of 71.46%. The use of pretrained weights and residual connections proved highly effective in addressing noisy labels, highlighting the benefits of transfer learning. However, this improvement came with increased computational demands in terms of training time and memory usage due to the higher input resolution and model complexity.
- **Wide ResNet-28x2**, despite its increased width, underperformed compared to VGG16 and ResNet18, achieving only 43.02%. While the model's width increased its capacity for feature extraction, its shallower depth and lack of fine-tuning may have limited its robustness in handling noisy labels. Its computational demands were comparable to ResNet18 but without the corresponding improvement in accuracy.
- **Co-Teaching** surpassed the baseline with both models achieving validation accuracies of 68.45% and 69.32%, respectively. This result highlights the effectiveness of collaborative filtering in mitigating noisy label issues. However, the approach required double the resources compared to single-model experiments, as it involves training two networks simultaneously.
- **ResNet34 (Extended Run)** achieved the highest validation accuracy of **74.78%**, marking a **15.03%** improvement over VGG16. Despite its deeper architecture and additional training time, it balanced performance and computational demands effectively. Its consistent results across grid search and extended training emphasize the value of well-optimized configurations for handling noisy labels.

3) Insights and Observations:

- **Data Augmentations:** The use of advanced augmentations such as MixUp and CutMix was instrumental in enhancing the performance of deeper models like ResNet18 and ResNet34, particularly under noisy label conditions.
- **Pretraining:** Models pretrained on large-scale datasets like ImageNet demonstrated significant advantages, as seen with ResNet18 and ResNet34.
- **Collaborative Learning:** Co-Teaching showed promise, particularly for dynamically filtering noisy labels. While it did not surpass ResNet34, its potential for further optimization remains significant.
- **Limitations of Wide ResNet:** Despite its increased width, Wide ResNet-28x2 underperformed, suggesting that width alone is insufficient for handling noisy labels without complementary architectural or optimization improvements.
- **ResNet34's Stability:** The ResNet34 configuration emerged as the most robust solution, with its performance only slightly improving when training epochs were increased. This underscores the importance of well-tuned hyperparameters and architectural depth.

V. Conclusion

This study explored various deep learning architectures and training techniques to tackle the challenge of noisy label classification in the CIFAR-100N dataset. By comparing baseline models, pretrained networks, and collaborative frameworks, we identified key strategies for improving robustness under noisy label conditions.

The experiments demonstrated that deeper architectures like ResNet34, when paired with advanced augmentations (e.g., MixUp, CutMix) and fine-tuned hyperparameters, provide the most effective approach, achieving a validation accuracy of 74.78%. Pretraining on large-scale datasets like ImageNet further amplified performance, highlighting the value of transfer learning in noisy label scenarios.

However, collaborative methods such as Co-Teaching also showed promise by dynamically filtering noisy labels, indicating potential for future exploration. While Wide ResNet-28x2 fell short, its capacity for learning complex features warrants further optimization to fully leverage its architecture.

A. Future Directions

Based on the findings of this study, the following directions are suggested for future work:

- **Enhancing Co-Teaching Methods:** Further exploration of Co-Teaching frameworks to improve their scalability and effectiveness under diverse noise conditions, including real-world datasets with complex noise patterns.
- **Optimizing Vision Transformers:** Investigating tailored augmentations, loss functions, and training strategies to enhance the performance of Vision Transformers in noisy label scenarios.
- **Hybrid Approaches:** Exploring combinations of Co-Teaching with advanced architectures, such as Vision Transformers or deeper ResNets, creating ensemble models to leverage the strengths of both methodologies for improved robustness.

B. Final Remarks

This work underscores the importance of systematic experimentation in noisy label classification, balancing architectural complexity, pretraining, and augmentation strategies. The insights gained provide a foundation for advancing robust learning methods and their applications in real-world noisy environments.

References

[1] T. Nishi, I. Yamada, M. Matsui, and M. Sugiyama, "Learning with noisy labels revisited: A study using real-world human annotations," in *International Conference on Learning Representations (ICLR)*, 2022. Available: http://www.noisylabels.com/

[2] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[3] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018. Available: https://arxiv.org/abs/1710.09412

[4] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Available: https://arxiv.org/abs/1804.06872

[6] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. Available: https://arxiv.org/abs/1805.09501

[7] D, "Grid search experiments for ResNet34 on noisy CIFAR100," Weights and Biases (W&B), 2024. Available: https://wandb.ai/dandodun-universitatea-alexandru-ioan-cuza-din-ia-i/noisy_cifar100?nw=nwuserdandodun

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. Available: https://arxiv.org/abs/1409.1556

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Available: https://arxiv.org/abs/1512.03385

[10] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2016. Available: https://arxiv.org/abs/1605.07146

[11] A. Dosovitskiy, J. T. Springenberg, M. R. A. P. P. L. et al., "Discriminative untrained neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. Available: https://arxiv.org/abs/1412.6806

[12] C. Szegedy, V. Vanhoucke, Z. Wu, and Y. Zhang, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Available: https://arxiv.org/abs/1512.00567