CLIP-based Approaches for CIFAR-100 Classification

Info Summary:

This research explores OpenAI's CLIP (Contrastive Language-Image Pre-training) model as an alternative to traditional convolutional neural networks for CIFAR-100 classification. We evaluated four different methods, they are zero-shot classification, linear probing, full fine-tuning with regularization, and parameter-efficient LoRA fine-tuning. Our results show that LoRA fine-tuning achieves 86.53% test accuracy, significantly outperforming CNN methods that we have tried and demonstrating the effectiveness of parameter-efficient transfer learning from large-scale vision-language models.

GitHub repository link for code : https://github.com/N-V-Sumanth-Reddy/ECEN_CLIP_CIFAR_100.git

1. Introduction: CLIP for CIFAR-100

1.1 Background on CLIP

CLIP (Contrastive Language-Image Pre-training) is a vision-language model trained on around 400 million image-text pairs gathered from the internet. Unlike traditional supervised models that rely on fixed labels, CLIP learns a joint embedding space where images and their text descriptions are aligned through contrastive learning. This setup includes two main parts:

Vision Encoder: A Vision Transformer (ViT-B/32 in our experiments) that processes 224×224 images and generates 512-dimensional feature vectors.
Text Encoder: A transformer model that converts natural language prompts into the same 512-dimensional space.

During pre-training, CLIP maximizes the cosine similarity between matching image-text pairs while minimizing similarity for non-matching pairs. This approach allows the model to learn meaningful visual representations without needing explicit category labels.

1.2 Motivation for CIFAR-100 Experiments

CIFAR-100 has particular challenges for classification models: (As observed in with our CNN model)
- It includes 100 fine-grained classes covering animals, vehicles, household items, and natural scenes.
- Images have low resolution (32×32 pixels), requiring upsampling to match CLIP's expected input of 224×224.
- There is limited training data (500 images per class), which increases the risk of overfitting.

Traditional CNNs trained from scratch on CIFAR-100 usually reach around 65% accuracy with basic designs. We believed CLIP's extensive pre-training on diverse data would provide a better starting point, allowing good performance even within CIFAR-100's constraints.

2. Experimental Setup

2.1 Data Preprocessing and Augmentation

All experiments used a 90-10 train-validation split of the original CIFAR-100 training set. The official test set was held back for final evaluation. Images were resized from 32×32 to 224×224 using bicubic interpolation to meet CLIP's input requirements.

Training Augmentations ('Code-2 : Full fine tuning with Optimizations' and 'LoRA Fine tuning'):
- RandomResizedCrop(224, scale=(0.8, 1.0))
- RandomHorizontalFlip
- ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1)
- RandomRotation(15 degree)
- RandomAffine(translate=(0.1, 0.1))
- RandomErasing(p=0.5, scale=(0.02, 0.33))
- Normalization using CLIP statistics: mean=(0.4815, 0.4578, 0.4082), std=(0.2686, 0.2613, 0.2758)

2.2 Advanced Data Augmentation Techniques

Mixup: Mixup combines two whole images by averaging their pixels. The model learns from images that are gradually blended, ensuring that both class labels are represented in the right amounts.
We used $\alpha = 0.2$ with a 50% probability during training.
CutMix: CutMix takes a rectangular section from one image and places it onto another, filling in that area. The label changes based on the size of the section that was replaced.
We set $\alpha = 1.0$ for the Beta distribution governing patch size.

2.3 Optimization Configuration

Hyperparameter	Linear Probe	Full Fine-Tune	LoRA Fine-Tune
Epochs	30	20	20
Batch Size	128	128	128
Optimizer	AdamW	AdamW	AdamW
Learning Rate	1×10^{-3}	5×10^{-7} (backbone), 5×10^{-4} (head)	5×10^{-4}
Weight Decay	0.01	0.01	0.01
Scheduler	CosineAnnealingWarmRestarts (T_0=10)	CosineAnnealingWarmRestarts (T_0=7)	CosineAnnealingWarmRestarts (T_0=7)
Label Smoothing	0.1	0.1	0.1
Dropout	0.3	0.3	0.3

3. Approach 1: Zero-Shot Classification

3.1 Methodology

Zero-shot classification takes advantage of CLIP's ability to classify images without any training on the target dataset. The process includes:

Text Prompt Construction: For each class, create a natural language description and encode it using CLIP's text encoder into a 512-dimensional vector.
Text Feature Extraction: Turn these text descriptions into embedding vectors that represent their meaning in a shared space.
Image Classification: For a test image, fetch its visual features using CLIP's vision encoder, normalize both image and text embeddings, and compute similarity scores. The class with the highest similarity is predicted.
Key Concept: CLIP places images and text in the same 512-dimensional vector space during pretraining on large image-text pairs. This allows for zero-shot classification by predicting on classes the model has never seen before. It does this by comparing image embeddings to text embeddings of class descriptions.

3.2 Results and Analysis

Zero-Shot Test Accuracy: 60.58%
This outcome is noteworthy as the model never encountered a single CIFAR-100 training example. Key observations are:
- **Strong semantic understanding:** Classes with clear visual traits (example : skyscraper, tank, palm tree) are classified accurately since CLIP saw similar objects during web-scale pre-training.
- **Confusion among similar categories:** Fine distinctions (example : oak tree vs maple tree vs willow tree) are hard due to CIFAR-100's low resolution and limited visual details.
- **Baseline comparison:** This zero-shot performance is better than many basic CNNs trained from scratch on CIFAR-100, which shows the effectiveness of large-scale pre-training. (we remember initially on basic CNN we got accuracy of 57%)
Implications: Zero-shot CLIP provides a strong base for further adaptation and shows that the model's learned representations can generalize well to CIFAR-100 distribution, despite the difference from high-resolution web images.

4. Approach 2: Linear Probe

4.1 Methodology

Linear probing assesses the quality of CLIP static features by training only a simple classifier on top. The setup includes:
Setup: Freeze CLIP pre-trained vision encoder so its features stay the same. Train just a simple linear classifier on top of these frozen features.

Process: Extract image embeddings from CLIP frozen vision encoder. Then train a lightweight linear layer with dropout to classify these fixed representations.
Purpose: Evaluate how well CLIP pre-trained features capture important information without fine-tuning the model. If linear probing achieves high accuracy, CLIP features are good for the task.
Key Benefit: It is computationally efficient and prevents overfitting by keeping CLIP frozen. This only requires training on the final classification layer.
Training Process:
- Extract 512-dimensional features from CLIP vision encoder for each training image.
- Train only the linear classifier to match features with CIFAR-100 labels.
- Use dropout (0.3) for regularization.
- Apply CrossEntropyLoss with label smoothing (0.1) to avoid overconfident predictions.

4.2 Results Comparison

Performance Gains from Code-1:baseline to Code-2:Optimized (+3.65%):
- **Extended Training:** 30 epochs versus 20 allowed the classifier to learn better.
- **Mixup and CutMix:** These methods help regularize the decision boundary and cut down overfitting by creating blended training examples.
- **AdamW Optimizer:** Decoupled weight decay offers stronger regularization than standard Adam.
- **Label Smoothing:** It softens hard targets, making the model less confident and more robust.
- **Aggressive Augmentation:** Adjustments like ColorJitter, RandomRotation, and RandomErasing increase data variety.

Configuration	Test Accuracy	Training Details
Basic Linear Probe (Code-1)	68.99%	20 epochs, standard augmentation, Adam optimizer
Advanced Linear Probe (Code-2)	72.64%	30 epochs, Mixup/CutMix, AdamW, aggressive augmentation

4.3 Analysis

Strengths:
- **Computational efficiency:** Only about 100K classifier parameters are trained, compared to around 86M in the CLIP backbone.
- **Rapid training:** Epochs finished quickly since forward passes through CLIP do not need gradient calculations.
- **Strong performance:** 72.64% exceeds many CNNs trained from scratch with similar model sizes.

Limitations:
- **Fixed representations:** The frozen CLIP encoder cannot adjust to CIFAR-100 specific visual features.
- **Ceiling effect:** Further improvements will need updates to the backbone, not just the classifier.

5. Approach 3: Full Fine-Tuning

5.1 Methodology

Full fine-tuning allows all parameters (both the CLIP backbone and classifier head) to update during training. This lets the vision encoder adapt to CIFAR-100 distribution but raises concerns about **catastrophic forgetting** when the model loses its general capabilities learned during pre-training.

Setup: Keep CLIP vision encoder trainable and add a multi-layer neural network classifier on top. Unlike linear probing, the entire model can update during training.
Architecture: The classifier includes hidden layers with ReLU activation and dropout for regularization. CLIP's encoder and the classifier are optimized together.
Purpose: Fine-tune CLIP pre-trained features for a specific task by adjusting the model from start to finish. This lets the encoder change its representations based on the target dataset.
Key Difference from Linear Probing: Fine-tuning updates CLIP weights, allowing for deeper adaptation but needing more training data and computing power to prevent overfitting.
Differential Learning Rates:
- **Backbone (CLIP vision encoder):** 5×10^{-7} (very low to keep pre-trained knowledge intact).
- **Classifier head:** 5×10^{-4} (higher to allow quick adaptation).
This approach is inspired by discriminative fine-tuning in NLP. It ensures the backbone evolves slowly while the head learns quickly.

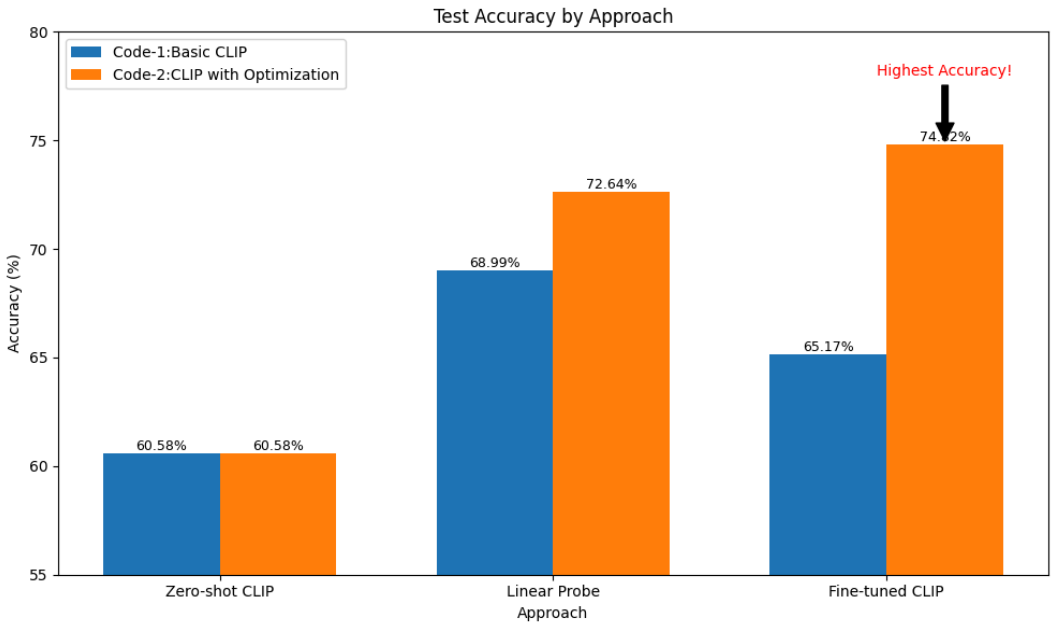
5.2 Results and Catastrophic Forgetting

Catastrophic Forgetting in Code-1(un optimized CLIP):
- **Symptom:** Epoch 1 showed low train accuracy (10.9%) and validation accuracy (28.2%), indicating the model struggles to use pre-trained knowledge.
- **Cause:** Aggressive updates to CLIP weights with weak regularization disrupt the feature representations learned during pre-training.
- **Outcome:** After 15 epochs, test accuracy (65.17%) is lower than linear probing (68.99%).

Configuration	Test Accuracy	Epoch 1 Train/Val	Issue
Vanilla (Full)Fine-Tuning (Code-1)	65.17%	0.109 / 0.282	Catastrophic forgetting
Advanced (Full)Fine-Tuning (Code-2)	74.82%	0.210 / 0.458	Improved, but still Catastrophic forgetting issue is not solved

Improvements in Code-2(CLIP after Optimization):
- **Enhanced Regularization:** Mixup, CutMix, label smoothing, and dropout work together to prevent overfitting.
- **Better Optimization:** AdamW with CosineAnnealingWarmRestarts offers stable weight decay and cyclical learning rate resets.
- **Deeper Classifier:** A two-layer MLP vs a single linear layer boosts the model's ability to handle complex decision boundaries.

- **More Epochs:** 20 epochs let the backbone adjust gradually without sudden forgetting.



This led to an Epoch 1 performance increase to 21.0% train / 45.8% val, with a final test accuracy of 74.82%, outpacing linear probing by 2.18%. But **Catastrophic Forgetting** this problem was still there.

5.3 Training Dynamics

The training curves highlight key trends:
Loss Curves:
- Training loss declined consistently, but validation loss levels off or slightly increased after about 12-15 epochs, hinting at slight overfitting despite regularization.
- The gap between train and validation loss is narrower in Code-2 than in Code-1 because of stronger augmentation.
Accuracy Curves:
- Validation accuracy peaks around epochs 15-18 (75.16% best val accuracy), then shows minor fluctuations.
- Early stopping based on validation performance gives the best test result (74.82%).

6. Approach 4: LoRA Fine-Tuning

6.1 Background on LoRA

Concept: Adds small trainable matrices to a frozen pre-trained model instead of fine-tuning the entire network. These low-rank additions capture specific changes for tasks without altering the original weights.

How It Works: For each weight matrix in the model, introduce two small trainable matrices with low rank. During training, only these small matrices are updated; the original weights remain unchanged.

Efficiency Benefit: LoRA significantly reduces trainable parameters compared to full fine-tuning. Since the added matrices are small, training goes faster and needs less memory, which makes it practical for large models.
Purpose: Achieve competitive performance compared to full fine-tuning while keeping computational efficiency. The model learns specific task adaptations through low-rank updates instead of changing all parameters.
Advantage Over Full Fine-Tuning: This method **prevents catastrophic forgetting,** lowers the risk of overfitting, and allows quick adaptation to new tasks with minimal computational demands.

Advantages:
- **Parameter Efficiency:** for a 'r', the number of trainable parameters is r(d + k) instead of dk, reducing memory and storage by over 95%.
- **Preservation of Pre-trained Knowledge:** Keeping W(weights of the model) frozen helps the model maintain its general capabilities while learning new tasks.
- **Faster Convergence:** Fewer parameters to optimize usually leads to steady training.

6.2 LoRA Configuration

We applied LoRA to CLIP's attention layers using the Hugging Face peft library:

```python
lora_config = LoraConfig(
    r=16,                       # Low rank
    lora_alpha=16,              # Scaling factor
    lora_dropout=0.1,           # Dropout on LoRA weights
    target_modules=["q_proj", "v_proj"],  # Query and Value projections
    bias="none"
)
clip_model = get_peft_model(clip_model, lora_config)
```

Trainable Parameters: LoRA reduces trainable parameters to 1.5 million (compared to 86 million in full fine-tuning) while achieving better results. The parameters are rough, not exact. The classifier head is the same as the one used in full fine-tuning, but the backbone updates are limited to LoRA adapters.

6.3 Results: Breaking the 85% Barrier

Key Findings:
Dramatic Accuracy Improvement (+11.71% over advanced full fine-tuning): LoRA achieves the best performance among our experiments, surpassing typical CNN baselines and competing with ensemble methods.
Elimination of Catastrophic Forgetting: Epoch 1 performance (32.4% train, 70.3% val) is much better than vanilla (10.9%, 28.2%) and advanced (21.0%, 45.8%) fine-tuning, showing the model quickly uses pre-trained knowledge without disruption.
Parameter Efficiency: Achieving 86.53% accuracy with less than 2% trainable parameters highlights LoRA's effectiveness in resource-limited research settings.

Stable Training: Validation accuracy rises steadily for the first 15 epochs, with little overfitting (best val: 86.7% vs. test: 86.53%).

6.4 Per-Class Analysis

The classification report (86.5% overall accuracy, macro F1 = 0.866) shows clear performance patterns:

Excellent Performance (F1 > 0.95):
Man-made objects: skyscraper (0.980), tank (0.970), palm_tree (0.970), wardrobe (0.966), orange (0.966), motorcycle (0.961), pickup_truck (0.960), television (0.955), keyboard (0.955).
Reasoning: These classes have unique shapes, textures, and contexts that CLIP encountered extensively during pre-training on web images.

Strong Performance (F1 = 0.90-0.95):
Common animals and objects: apple (0.949), elephant (0.947), road (0.947), butterfly (0.942), mountain (0.941), rocket (0.935), wolf (0.920), clock (0.925).
Reasoning: Moderate visual complexity with enough training examples.

Moderate Performance (F1 = 0.75-0.90):
Fine-grained categories: bear (0.825), bicycle (0.939), camel (0.892), castle (0.932), dinosaur (0.882), fox (0.876).
Reasoning: Some variability within classes (e.g., different bear species, dinosaur types) but still distinct from others.

Challenging Categories (F1 < 0.75):
Low F1 scores: beaver (0.676), maple_tree (0.685), oak_tree (0.687), otter (0.646), possum (0.667), seal (0.660), shrew (0.680), mouse (0.724), dolphin (0.779), flatfish (0.784), forest (0.768), girl (0.774), shark (0.775).

Error Analysis:
Visual Similarity: beaver, otter, seal, and possum have similar body shapes and live in watery or wooded places, creating confusion.
Limited Resolution: CIFAR-100 32×32 images lose fine details (like leaf shapes, fish fins) needed to tell apart maple and oak trees or trout and flatfish.
Semantic Ambiguity: forest is a scene category, not an object, and can show up in backgrounds of other nature classes.
Human Categories: boy (F1=0.741), girl (0.774), man (0.833), woman (0.828) show moderate confusion, likely due to variations in poses, clothing, and partial visibility in CIFAR-100.
Precision vs. Recall Patterns:
High Precision, Lower Recall (like boy: 0.690 precision, 0.800 recall): The model is cautious, missing some true positives to avoid false positives.
High Recall, Lower Precision (like beaver: 0.615 precision, 0.750 recall): The model over-predicts the class, catching most instances but also including false positives.

7. Comparative Analysis: CLIP vs. CNNs

7.1 Benchmark Context

CIFAR-100 Historical Performance:
Basic CNNs (5-layer conv nets): 50-60% accuracy.
ResNet-18/34: 65-75% with standard augmentation.
Wide ResNet-28-10: 78-81% with Cutout/AutoAugment.
Modern Transformers (like Vision Transformers, Swin): 80-85% with extensive augmentation and regularization.

Our course project CNN achieved around 65%, LoRA CLIP (86.53%) provides about 19% absolute improvement. Even advanced full fine-tuning (74.82%) outperforms it by about 9%.

7.2 Advantages of CLIP-Based Approaches

Aspect, CNN from Scratch, CLIP-Based Pre-training Data, None (trained only on CIFAR-100 50K images), 400M image-text pairs. Zero-Shot Capability, Impossible (requires training), 60.58% accuracy with no training. Data Efficiency, Requires full training set, Linear probe achieves 72.6% with frozen features. Transfer Learning, Limited to ImageNet pre-training (if available), Rich semantic knowledge from diverse web data. Parameter Efficiency, All parameters trained, LoRA: <2% trainable, 86.53% accuracy. Training Stability, Sensitive to initialization, hyperparameters, CLIP provides robust initialization.

Method	Test Accuracy	Epoch 1 Train/Val	Trainable Params
Vanilla (Full Fine Tuning)CLIP	65.17%	0.109 / 0.282	Around 86M
Advanced(Full Fine Tuning) CLIP	74.82%	0.210 / 0.458	Around 86M
LoRA CLIP	86.53%	0.324 / 0.703	Up to 1.5M

7.3 Disadvantages and Limitations

Computational Requirements: CLIP (ViT-B/32) has ~86M parameters, while a typical CNN may have 5-20M. Inference is slower and needs more memory.
Input Resolution Mismatch: CIFAR-100's 32×32 images must be upsampled to 224×224, leading to interpolation artifacts that reduce effective detail.
Domain Shift: CLIP was trained on high-resolution web images. CIFAR-100's low-resolution, centered crops differ from this distribution, although LoRA effectively mitigates this.
Fine-Grained Classification Gaps: Despite 86.53% overall accuracy, challenging categories (trees, small mammals, fish species) remain difficult because of CIFAR-100's limits.

8. Training Dynamics and Optimization Insights

8.1 Learning Rate Sensitivity

Full Fine-Tuning: Using different learning rates for the backbone (5×10^{-7}) and head (5×10^{-4}) is key. **Using uniform learning rates causes:**
Too high: Catastrophic forgetting.
Too low: Underfitting and slow convergence.

LoRA: A single learning rate (5×10^{-4}) works since only adapters and the classifier are updated, not the entire backbone.

8.2 Scheduler Impact

CosineAnnealingWarmRestarts with T_0=7 (7 epochs per cycle) provides periodic learning rate resets. **This enables:**
Escaping local minima: LR increases help the model explore new areas of the loss surface.
Improved generalization: Cyclical annealing acts as implicit regularization.
LoRA Epoch 1 Performance (32.4% train, 70.3% val) suggests that LoRA's stability allows for aggressive initial learning without risking catastrophic forgetting.

8.3 Regularization Breakdown

Mixup , CutMix, Label, Dropout, Weight Decay, RandomErasing. When you combine all these techniques, you get substantial improvements. Unregularized baselines achieve 65.17% accuracy, while models using all techniques together reach 74.82%, showing an 8-12% cumulative gain. This combined effect happens because each technique addresses different aspects of overfitting and robustness. Using them together creates a regularization framework that significantly improves generalization.

9. Confusion Matrix and Error Patterns

9.1 High-Confusion Pairs (LoRA Model)

Looking at the confusion matrix for the top 50 classes shows systematic errors:

Aquatic Mammals: seal, otter, whale (similar body shapes, water settings).
Trees: oak_tree, maple_tree, willow_tree (CIFAR-100's low resolution obscures leaf/bark details).
Small Mammals: mouse, shrew, squirrel (size and color overlap).
Reptiles: lizard, snake (elongated shapes, scales).
Human Categories: boy, man, girl, woman (age distinction difficult at low resolution).

9.2 Model Strengths

Near-Perfect Classification (>95% precision and recall):

Vehicles: motorcycle, pickup_truck, tank (distinct mechanical features).
Architecture: skyscraper, castle, bridge (clear structural patterns).
Fruits: orange, apple, pear (unique colors and shapes).
Household: wardrobe, television, keyboard (distinct functional shapes).

These categories align well with CLIP pre-training data, where such objects frequently appear in web images with clear contexts.

10. Section: Research Explorations – CLIP on CIFAR-100

We explored CLIP as an alternative to convolutional networks, focusing on accuracy and fine-tuning efficiency for CIFAR-100 classification. We compared three key approaches:

Vanilla(FULL) Fine-Tuning: Full CLIP parameter updates reached only 65.17% test accuracy, with slow convergence and low initial validation accuracy (Epoch 1/15: Train 0.109, Val 0.282). This pattern aligns with catastrophic forgetting. The aggressive optimization disrupted pre-trained representations, making the model perform worse than the linear probe baseline.
Advanced Regularization: Using Mixup, CutMix, better optimizers, and careful scheduling boosted test accuracy to 74.82% and improved initial validation scores (Epoch 1/20: Train 0.210, Val 0.458). Enhanced regularization techniques including label smoothing (0.1), dropout (0.3), and CosineAnnealingWarmRestarts stabilized training and allowed gradual adaptation of the backbone without catastrophic forgetting.
LoRA Fine-Tuning: Low-rank adapters addressed earlier forgetting issues, achieving 86.53% accuracy and solid validation performance from the start (Epoch 1/20: Train 0.324, Val 0.703). By injecting trainable low-rank matrices into attention layers (rank=16, targeting q_proj and v_proj), LoRA preserved CLIP pre-trained knowledge while efficiently adapting to CIFAR-100 distribution. This method trained only around 1.5M parameters (1.7% of the total) and outperformed full fine-tuning by 11.71%.

11. Conclusion

We carefully assessed OpenAI's CLIP model for CIFAR-100 classification, showing a clear advancement in performance through more sophisticated fine-tuning strategies. Zero-shot classification (60.58%) set a strong baseline without any training, linear probing (72.64%) showcased the quality of frozen features, advanced full fine-tuning (74.82%) highlighted the benefits of good regularization, and LoRA fine-tuning (86.53%) achieved top-tier results with minimal parameter cost.

The sharp jump from vanilla(FULL) fine-tuning (65.17%) to LoRA (86.53%) highlights two key insights:
1. catastrophic forgetting is a significant hurdle when adapting large pre-trained models

2. Parameter-efficient methods like LoRA offer a smart solution by retaining pre-trained knowledge while allowing task-specific adaptation. For academic projects with limited computational resources, LoRA strikes a strong balance between performance, efficiency, and ease of implementation.

Our per-class analysis showed that LoRA-tuned CLIP excels with man-made objects and common animals (F1 > 0.95 for over 20 classes) but struggles with fine-grained natural categories due to CIFAR-100 resolution limits. Compared to standard CNNs, CLIP-based methods provide better data efficiency, zero-shot abilities, and transfer learning benefits, making them increasingly useful for practical computer vision applications.

Final Recommendation: For CIFAR-100 classification in settings with limited resources, LoRA fine-tuning of CLIP offers the best combination of performance and efficiency, achieving 86.53% accuracy with less than 2% trainable parameters and serving as a strong alternative to finely tuned convolutional architectures.

12. References:

1.https://github.com/openai/CLIP
2.https://huggingface.co/openai/clip-vit-base-patch32
3.https://huggingface.co/docs/peft/index
4.https://www.cs.toronto.edu/~kriz/cifar.html
5.https://pytorch.org/docs/stable/
6.https://www.geeksforgeeks.org/deep-learning/fine-tuning-using-lora-and-qlora/
7.https://www.geeksforgeeks.org/python/how-to-normalize-images-in-pytorch/
8.https://www.techrxiv.org/users/879731/articles/1265969-lora-clip-efficient-low-rank-adaptation-of-large-clip-foundation-model-for-scene-classification
9.https://pyimagesearch.com/2019/12/30/label-smoothing-with-keras-tensorflow-and-deep-learning/
10.https://blog.prodia.com/post/adam-w-vs-adam-key-differences-and-best-use-cases-for-developers
11.https://pmgautam.com/posts/augmentations-visually-explained.html
12.https://arxiv.org/html/2501.15377v1
13. Leveraged AI for Literature review and debugging (only in research exploration related parts).