Exploring the Role of Optimizers and Architectures in Active Learning

Khadiza Sarwar Moury, Md Hasibul Hasan Shovo, Syed Abdullah Ali, Youssef Midra
Concordia University, Canada
Student IDs: 40343620, 40329539, 40333397, 40341729
(khadizasarwar.moury, mdhasibulhasan.shovo,
syedabdullah.ali, youssef.midra)@mail.concordia.ca

Abstract

Active Learning (AL) is a strategy for reducing the high cost of data annotation in deep learning. It is achieved by selecting the most informative samples for training using query strategies. This is a developing field where the interplay between the underlying model architecture, the optimization algorithm, and the active learning process remains underexplored. This paper presents a comprehensive comparative study of three ResNet architectures (ResNet18, ResNet34, ResNet50) trained using four different optimizers (SGD, Adam, AdamW, RMSProp) within a Loss Prediction based active learning framework. Using the CIFAR-10 dataset and a fixed labeling budget, we demonstrate that the choice of optimizer significantly dictates the learning trajectory. Our results reveal a distinct trade-off: adaptive optimizers like Adam provide superior sample efficiency in the "cold start" phase (low-data period), whereas SGD with momentum consistently achieves higher generalization accuracy once the labeled pool exceeds a critical threshold. Furthermore, we find that deeper architectures like ResNet-50 require a longer "warm-up" period but ultimately provide the highest performance when paired with SGD. These findings suggest that dynamic optimization strategies play an important role in maximizing the efficiency of active learning pipelines. The code is available at Github

Instead of randomly choosing data points, AL prioritizes instances that are expected to improve model performance the most when labeled.

1 INTRODUCTION

Deep learning models have achieved remarkable success in image classification tasks, but this success often comes at the cost of requiring large amounts of labeled data. In many domains, labeling such data can be expensive, time consuming, and some domains; such as medical imaging or satellite imagery; require expert annotators. Active Learning (AL) offers a compelling solution to this problem by selecting the most informative or uncertain samples from a large pool of unlabeled data for annotation. An AL strategy being used is uncertainty sampling [10], where the model queries data points it is least confident about, which helps the model refine its decision boundaries, especially during the early stages of training.

In this research, we explore how different choices of optimizers (SGD, Adam, AdamW, RMSProp) and model architectures (ResNet18, ResNet34, ResNet50) [2] affect the performance and efficiency of an active learning pipeline. Using the CIFAR-10 dataset [3], we simulated a budget constrained labeling scenario by incrementally increasing the labeled dataset over a series of active learning cycles. Through systematic experimentation, we discovered how both optimization strategy and model capacity substantially affect classification accuracy and sample efficiency; factors that are critical for deploying active-learning–based image classifiers in resource-limited environments.

2 RESEARCH OBJECTIVES

- To assess the effectiveness of various optimizers (SGD, Adam, AdamW, RMSProp) in the context of active learning.
- To evaluate how different Residual Network (ResNet) architectures (ResNet18, ResNet34, ResNet50) impact the performance of active learning on the CIFAR-10 dataset.
- To compare the overall performance (accuracy and sample efficiency) of these models and optimizers under a fixed labeling budget.

3 METHODOLOGY

This research adopts an iterative active learning framework to evaluate how differing model architectures and optimizers impact learning efficiency and performance. The study uses the benchmark image classification dataset CIFAR-10 [3].

The active learning process is initialized without labeled data. In the first cycle, a "cold start" strategy is employed where 1,000 samples are selected and labeled randomly. In each subsequent cycle, the model queries an additional 1,000 samples from the unlabeled pool using the Uncertainty Sampling strategy [1], adding them to the labeled training set. This iterative process continues over ten cycles, resulting in a total labeling budget of 10,000 samples per run. To ensure consistent evaluation, the model is further trained using the updated labeled set after each cycle. Performance is assessed via accuracy on a held-out test set, sample efficiency, and computational efficiency. This methodology is applied systematically across all model–optimizer combinations outlined below.

3.1 Model Architectures

We utilized three variations of the ResNet family to analyze the trade-off between model complexity and active learning efficiency. These networks utilize residual learning through identity based skip connections, which mitigate the vanishing gradient problem and enable the stable optimization of deeper models without degradation in accuracy [2]. The architectural differences are visualized in Fig. 1. **ResNet18** is a light, powerful baseline with *11M parameters*. It has a shallow depth and residual block structure. Its ideal for fast training and inference in resource constrained environments. Its reduced parameter count also helps minimize overfitting. **ResNet34** increases the network depth, with *21M parameters*, while maintaining the same basic residual block architecture. This architecture provides a balanced trade-off for more complex datasets requiring detailed feature extraction without a substantial increase in computational cost. **ResNet50** ResNet-50 introduces *bottleneck residual blocks*, which use a 1×1–3×3–1×1 convolutional pattern to greatly increase depth to *25M parameters* while controlling parameter growth. This detailed architecture is ideal for complex tasks requiring high level feature extraction. It also has an optimized design to ensure efficient gradient flow and manageable memory usage.

3.2 Optimizers

To evaluate the impact of optimization dynamics on active learning performance, we trained the architectures using four distinct optimizers. Each optimizer employs a different strategy for navigating the loss landscape, influencing convergence speed and generalization capability. **Stochastic Gradient Descent (SGD)** [5] is a classical optimization algorithm that uses momentum and is renowned for finding flatter minima in the loss landscape. This correlates with better generalization on unseen test data; crucial when labeled data is scarce. **RMSProp** [6] is an optimizer that addresses diminishing learning rates by maintaining a moving average of the squared gradients to adapt the learning rate for each parameter individually. This ensures faster convergence and robustness against the noise. Adaptive Moment Estimation (**Adam**) [7] combines the benefits of RMSProp and momentum. It computes adaptive learning rates from estimates of the first and second moments of the gradients. Its popular for it's rapid convergence speed, however it can sometimes lead to overfitting on smaller, initial labeled sets. **AdamW** [8] is a modification of the Adam optimizer that decouples weight decay from the gradient update. By applying weight decay directly to the parameters, it achieves better training stability and improved generalization performance, particularly for deep residual networks trained on image classification tasks.

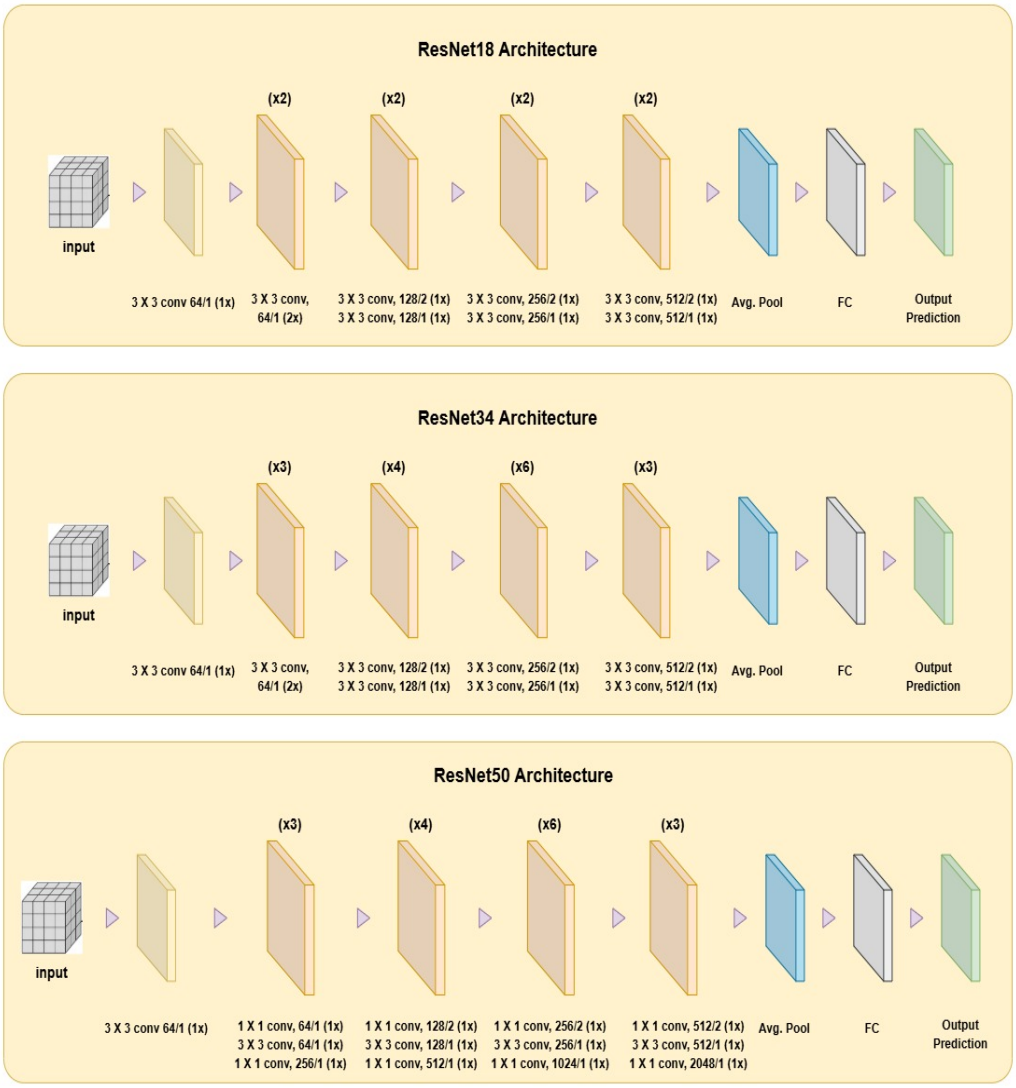


Figure 1: Overview of the ResNet architectures (ResNet18, ResNet34, and ResNet50) utilized in this study, highlighting the difference in depth and convolutional blocks.

4 EXPERIMENT SETUP

4.1 Implementation and Computing Environment

Our experiments were conducted on Concordia University's High-Performance Computing (HPC) facility, *Speed*. We utilized NVIDIA GPUs (V100, 32GB RAM) available within the cluster nodes for computational efficiency. The active learning pipelines were implemented using the PyTorch framework.

4.2 Dataset and Preprocessing

We utilize the CIFAR-10 dataset, consisting of 60,000 32×32 color images across 10 classes. The dataset is split into 50,000 training images and 10,000 test images. To prevent overfitting, especially in the early active learning cycles where labeled data is minimal, we apply **standard augmentations** during training. Input images are **normalized** using the channel-wise means and standard deviations calculated from the training set.

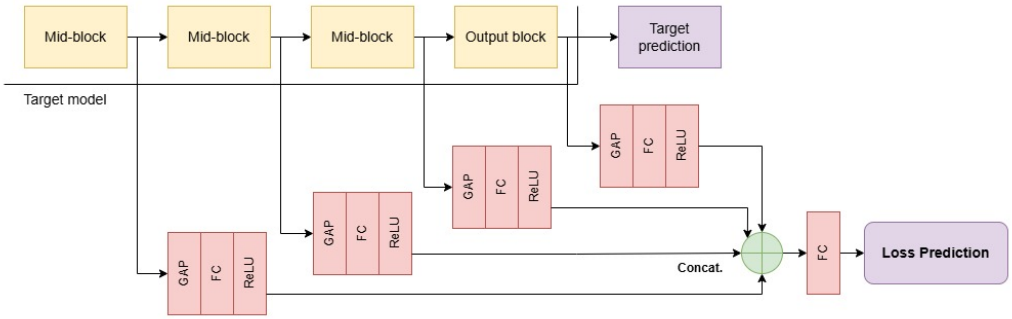


Figure 2: The architectural flow of the Active Learning module. The module extracts features from mid level blocks of the target ResNet, aggregates them, and predicts the loss value to guide sample selection [4].

4.3 Model Architectures

We evaluate three variations of the ResNet family to analyze the trade-off between model capacity and active learning efficiency. All models are initialized with random weights at the start of the experiment to simulate a scenario where no pre-trained weights are available for the specific domain.

4.4 Optimization Strategies

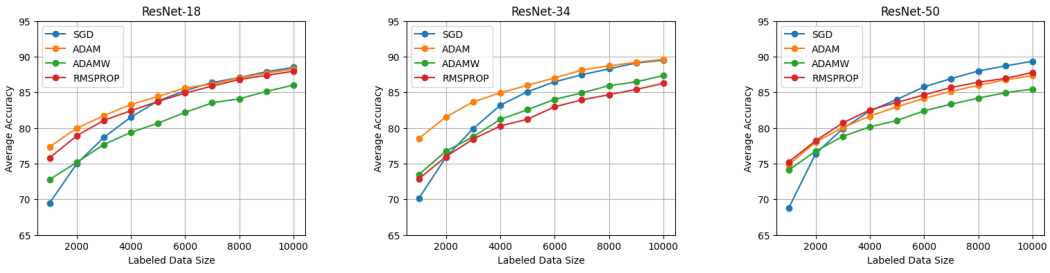
We compare four separate optimization algorithms to determine their stability and convergence speed within the AL loop. The hyperparameters for each were selected based on standard literature benchmarks for CIFAR-10:

1. **Stochastic Gradient Descent (SGD):** Used with a Nesterov momentum of 0.9 and a weight decay of $5e^{-4}$. This serves as the standard baseline for computer vision tasks [5].
2. **Adam:** An adaptive moment estimation method. We utilize default parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) [7].
3. **AdamW:** A modification of Adam that decouples weight decay from the gradient update. This is crucial for regularization in adaptive methods [8].
4. **RMSProp:** Utilizes a moving average of squared gradients to normalize the gradient. We set the smoothing constant $\alpha = 0.99$ [6].

4.5 Active Learning Configuration

The experiment simulates a pool-based active learning scenario. The interaction between the target model and the loss prediction mechanism is detailed in Fig. 2.

- **Loss Prediction Module:** Instead of relying on fixed heuristic metrics like entropy, we employ a learnable *Loss Prediction Module* [4]. As illustrated in Fig. 2, this module is attached to the intermediate layers of the target ResNet model. It aggregates feature maps from multiple depths (mid blocks) using Global Average Pooling (GAP) and Fully Connected (FC) layers. These features are concatenated and passed through a final FC layer to predict a scalar loss value for a given input. During the active learning acquisition step, the model queries samples from the unlabeled pool that yield the highest predicted loss, under the assumption that high loss samples are the most informative for the current model state.
- **Budget Protocol:** The total budget is 10,000 samples.
  - *Cold Start:* As the model has no initial knowledge, the first cycle ($n = 1$) selects 1,000 samples randomly to establish an initial decision boundary.
  - *Active Cycles:* For cycles $n = 2$ to 10, the model queries the top-1,000 most informative samples (highest predicted loss) from the remaining unlabeled pool.



(a) (b) (c)

Figure 3: Performance comaprison of 3 architectures (ResNet-18, ResNet-34, ResNet-50) on 4 different optimizers (SGD, ADAM, ADAMW, RMSProp).

- **Training Routine:** For training, we apply a standard augmentation scheme including 32×32 size random crop from 36×36 zero-padded images and random horizontal flip, and normalize images using the channel mean and standard deviation vectors estimated over the training set. For each of active learning cycle, we learn the model set $\{\Theta_{\text{target}}, \Theta_{\text{loss}}\}$ for 200 epochs with the mini-batch size of 128 and the initial learning rate of 0.1. After 160 epochs, we decrease the learning rate to 0.01. The momentum and the weight decay are 0.9 and 0.0005 respectively. After 120 epochs, we stop the gradient from the loss prediction module propagated to the target model. After each acquisition step, the model is retrained from scratch (re-initialized) on the accumulated labeled set to prevent catastrophic forgetting and to isolate the effect of the added data.

5 PERFORMANCE EVALUATION

The goal of this study is to analyze different model-optimizer pairings. To achieve this we compare the performance of each model–optimizer combination on the CIFAR-10 dataset. The evaluation in our project naturally extends standard classification evaluation with budget aware metrics that are common in active learning. The following metrics were evaluated after every cycle:

1. **Accuracy (per active learning cycle)**: Tracks model's ability to classify correctly across cycles. After each cycle, we evaluate accuracy on the CIFAR-10 test set:

$$\text{Accuracy}^{(t)} = \frac{1}{n_{\text{test}}} \sum_{j=1}^{n_{\text{test}}} \mathbf{1}\left[f_\theta^{(t)}\left(x_j^{(\text{test})}\right) = y_j^{(\text{test})} \right].$$

2. **Sample Efficiency**: Measures how quickly a model reaches a given accuracy relative to the labeled data. Conceptually, we visualize:

$$\text{Accuracy}^{(t)} \text{ vs. } N_{\text{labels}}^{(t)}.$$

3. **Accuracy vs. Budget**: Treating the labeling budget B as the x-axis, we report:

$$\text{Accuracy}(B), \qquad B \in \{B_1, \ldots, B_{10}\}.$$

6 RESULTS

This section presents the comparative analysis of ResNet architectures and optimizers across the ten active learning cycles. We examine the impact of optimization strategies on model convergence as the labeled dataset size grows from 1,000 to 10,000.

The experimental results across the three ResNet (18, 34, and 50) architectures reveal a trade-off between the sample efficiency of optimizers (e.g., Adam, AdamW) and the superior generalization of Stochastic Gradient Descent (SGD) with momentum as the labeled data budget expands.

In the early "cold start" phases, especially with limited labeled data ($\lesssim 4000$ samples), adaptive optimizers exhibit quicker convergence and higher initial accuracy across all models. For instance, with ResNet-18, Adam achieved $\approx 77\%$ accuracy, outperforming SGD's $\approx 69\%$ in the initial cycles.

However, as the labeled data budget increases up to the final 10,000 samples, SGD with momentum demonstrates a steeper learning curve and eventually surpasses the adaptive methods. For both ResNet-18 and ResNet-34, all optimizers converged to a high performance range ($\approx 86\% - 89.5\%$). The trade-off is most pronounced in the deepest model, ResNet-50, where SGD initially lags significantly ($\ll 70\%$ at 1,000 samples) but exhibits the steepest improvement, eventually overtaking all adaptive optimizers to achieve the highest overall accuracy ($\approx 90\%$). This reinforces the hypothesis: adaptive methods are superior for maximizing performance with sparse data, while SGD, despite requiring a more data "warm-up," returns better final decision boundaries and generalization when more labeled data is available.

7 CONCLUSION

In this paper, we conducted a systematic evaluation of ResNet architectures and optimization algorithms within a Loss Prediction-based active learning framework. Our experiments on the CIFAR-10 dataset demonstrate that there is no single "best" optimizer for all stages of active learning. Instead, we observed a distinct cross-over effect: adaptive optimizers (Adam, AdamW) are significantly more sample-efficient in the early, low-data regimes, while SGD with momentum provides superior generalization as the labeled dataset grows. Specifically, for the deepest architecture (ResNet-50), SGD started with the lowest accuracy but concluded with the highest, suggesting that deeper models benefit most from the regularization effects of SGD once sufficient data is available.

Limitations and Future Work. A limitation of this study is the focus on CIFAR-10 as the only dataset for performance benchmarking. While CIFAR-10 is a solid foundation for benchmarking, in more complex and demanding tasks, results may vary; for e.g., on higher resolution datasets like ImageNet, MPII (Human Pose Estimation Dataset), and VOC2007/VOC2012 (Object Detection datasets). These datasets can be explored in future work using the same active learning approach.

Additionally, retraining from scratch at each cycle is computationally expensive. Future work can explore hybrid optimization strategies, starting with Adam and switching to SGD, and investigate fine-tuning techniques to reduce the computational overhead of the active learning pipeline. We can apply our findings in various settings, such as medical or industrial imaging, where labeling costs are high and data is scarce.

Furthermore, exploring semi-supervised learning approaches, where the model learns from an initial labeled set and then iteratively uses the unlabeled set, could enhance overall efficiency. Lastly, as Large Language Models (LLMs) continue to develop, a possible avenue for future research is to check the performance of LLM models in labeling unknown datasets to completely automate or significantly accelerate the data annotation process within the active learning loop.

References

[1] B. Settles, *Active Learning*. in Synthesis Lectures on Artificial Intelligence and Machine Learning. Cham: Springer International Publishing, 2012. doi: 10.1007/978-3-031-01560-1.

[2] 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, pp. 770–778.

[3] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[4] D. Yoo and I. S. Kweon, "Learning Loss for Active Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 93-102.

[5] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning (ICML)*, 2013, pp. 1139-1147.

[6] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26-31, 2012.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[8] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.

[10] T. N. Vo, "Active Learning for Multi-class Image Classification," in *arXiv preprint arXiv:2505.06825*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The comparative nature of the study is stated in the abstract and introduction The findings regarding the trade-off between sample efficiency and final accuracy is discussed, and are supported by the Results section.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a dedicated paragraph "Limitations and Future Work," in the Conclusion, which discusses the restriction to the CIFAR-10 dataset and the computational cost of retraining.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper presents an empirical comparative study and does not propose new theoretical theorems or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The Experiment Setup section details the architectures, specific hyperparameters (learning rates, momentum, weight decay), the query strategy (Loss Prediction), and the budget protocol used.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The CIFAR-10 dataset is publicly available. A link to the code repository is provided with the author information.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The Experiment Setup section explicitly lists the batch size, epochs, learning rate schedulers, optimizer parameters, and data augmentation techniques.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We conducted multiple trials, for which we report the general trends in accuracy and sample efficiency in the main text and figures. This is done without explicitly plotting error bars or calculating statistical significance values in this report.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Implementation and Computing Environment section specifies the use of Concordia University's HPC facility and NVIDIA V100 (32GB) GPUs.

9. **Broader impacts**

   Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

   Answer: [NA]

   Justification: Our project explores optimization strategies for active learning on a standard benchmark (CIFAR-10) and does not have immediate or specific societal impacts requiring discussion beyond general machine learning efficiency.

10. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The study utilizes standard image classification models and public datasets which do not pose any risk for misuse.

11. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new datasets or assets were created; this study relies on preexisting, publicly available benchmarks and architectures.

12. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [NA]

    Justification: Large Language Models were not used as a component of the core research methodology.