Learning More With Less: Reducing Labeling Effort through Active Learning

Carlotta Hölzle **Jannik Hösch** **Marina Wiemers**

Abstract

Modern deep learning models typically require large amounts of labeled data, making manual annotation a key bottleneck in real-world applications. This project explores how active learning (AL) can reduce labeling effort while maintaining high classification performance, using the Oxford-IIIT Pet Dataset. We first implemented a transfer learning pipeline for binary classification (cats vs. dogs) using a ResNet18 pretrained on ImageNet, achieving over 99% accuracy. Building on this, we extended our pipeline to the more complex multi-class classification task of distinguishing 37 cat and dog breeds. Through a series of fine-tuning experiments, including block-wise unfreezing and learning rate decay, our best ResNet101 model reached a test accuracy of 92.43%. In the extension phase, we evaluated four active learning strategies: entropy sampling, least confidence, KMeans clustering, and a hybrid method, against a random sampling baseline on the binary classification task. All strategies were applied in isolation and tested across multiple training setups under label-scarce conditions. Entropy-based sampling emerged as the most stable and effective strategy when combined with random augmentation and robust initialization. Notably, compared to the model trained on the full dataset, entropy-based AL achieved the same accuracy using significantly fewer labeled examples and less training time. This demonstrates that carefully designed AL pipelines can substantially reduce annotation cost without compromising performance. Finally, we adapted the best performing binary classifier model to the multi-class problem and tested the impact of the ration between entropy-selected and random samples on the model performance. Our results highlight that AL effectiveness strongly depends on training calibration and task complexity.

1 Introduction

This report explores the application of active learning strategies to improve training efficiency for binary and multi-class image classification tasks using the Oxford-IIIT Pet Dataset. The underlying challenge is that modern deep learning models typically require large amounts of labeled data to perform well. However, manual labeling is often expensive and time-consuming. Active learning addresses this by selecting which samples should be labeled next, potentially reducing annotation costs while maintaining high performance.

2 Related Work

The Oxford-IIIT Pet Dataset is a frequently used dataset to benchmark transfer learning capacities of classification models. A top performance on this dataset for the multi-class classification task was achieved by Wang et al. [2021], who applied a twist-based architecture using a ResNet50 backbone to this dataset with an accuracy of 94.5% on the multi-class task. In their extensive study on AL, Saifullah et al. [2023] evaluated a wide range of AL strategies across multiple document classification datasets using a ResNet50 architecture. They demonstrated that models trained on 15–40% of the labeled data nearly match the performance of fully supervised models. They recorded that uncertainty-based methods, particularly entropy sampling, consistently outperformed random sampling and other approaches across multiple datasets. While uncertainty-based sampling, especially entropy sampling, is commonly employed in deep AL (Ren et al. [2021], Saifullah et al. [2023]), there is a trend to use hybrid methods, which incorporate both uncertainty and diversity to offer improved performance (He et al. [2024], Chowdhury [2023]).

3 Data

Our project uses the Oxford-IIIT Pet Dataset (oxV), which contains images of 37 cat and dog breeds, with around 200 images per class, totaling approximately 7.400 images. The images vary in pose, lighting, and scale, making the dataset suitable for fine-grained image classification tasks. Each image includes annotations for: breed (species and breed name), a head bounding box, and pixel-level segmentation masks (trimaps). In this project we focus on the breed names as target classes. The dataset and ground truth annotations are roughly 800 MB in size and are publicly available via BitTorrent or direct HTTP download. We applied basic preprocessing, mostly limited to resizing and normalization, as the dataset is well-curated. Given that some parts of both the basic project and the extension rely on binary classificaiton, the different breeds were accordingly grouped into cat and dog breeds for such purposes. For the basic project we used the torch split between train and test and used 20% of the training set for validation and for our AL experiments we used a 70-15-15 split.

4 Methods

4.1 Basic Project

4.1.1 Model Architecture

We experimented with different ResNet architectures pretrained on ImageNet data, replacing the final fully connected layer with either a single neuron output layer with sigmoid activation for the binary classification or a 37-neuron output layer with softmax activations for the multi-class classification task. Deeper architectures, while improving the performance slightly, required significantly longer runtimes, therefore, we decided to use the simplest version, ResNet18, for further experiments. As optimizer we used Adam with mini-batch size of 32 and binary/multiclass cross-entropy loss.

4.1.2 Fine-tuning Strategies

For the more complex multiclass classification task, only fine-tuning the classification layer is not sufficient, instead higher layers of the network were included in the fine-tuning process. The ResNet18 (Figure 1) consists of 4 trainable residual blocks of convolution layers. We decided on block-wise instead of layer-wise fine-tuning since unfreezing only some layers within a block can lead to disruptions of internal feature consistencies. To get a better understanding of the impact of

number of blocks on the network we first trained the network on a fixed number of blocks for 10 epochs. We incrementally increased the number of trainable blocks from the last block backwards (l = 1, 2, 3, 4) and studied the trade-off between computational cost and performance improvement. Next, we implemented gradual unfreezing, where we started by training only the new classification layer and then progressively unfroze earlier blocks after every 5 epochs. To maximize the performance we applied further techniques to this model such as learning rate decay, data augmentation, regularization and batch normalization tuning.

4.1.3 Imbalanced Classes

As a final exploration of the basic project, we checked how the model performed with imbalanced training data. To do so, the amount of cat breed examples was reduced to 20% of the original size, keeping the size of the dog breeds constant. First, the basic pre-fine-tuning multi-class model was directly applied to this dataset, serving as a baseline. Next, two strategies to tackle the imbalance were employed; weighted cross-entropy loss, which adjusts the loss function to give more importance to underrepresented classes, and a rebalancing of the imbalanced dataset using Pytorch's `Weighted Random Sampler` which draws examples with a probability proportional to their class weight, increasing the likelihood of cat breeds to be trained on.

4.2 Extension: Active Learning

In the extension, we implemented and evaluated four AL strategies spanning three conceptual categories: (i) informativeness-based methods (least confidence, entropy), (ii) representativeness adapted sampling (KMeans), and (iii) a hybrid approach that integrates informativeness with representativeness. The informative-based strategies rely on model output probabilities $\mathbf{p} = [p_1, \ldots, p_C]$ for a C-class classification task. For least confidence, informativeness is measured as:

$$s_{\text{LC}}(x) = 1 - \max_c p_c \tag{1}$$

For maximum entropy, uncertainty is quantified using Shannon entropy:

$$s_{\text{Entropy}}(x) = -\sum_{c=1}^{C} p_c \log(p_c + \varepsilon) \tag{2}$$

where ε is a small constant added for numerical stability. The KMeans sampling method uses intermediate feature representations $\phi(x) \in \mathbb{R}^d$ extracted from the current model. For a pool of N unlabeled samples, we compute features $\{\phi(x_i)\}_{i=1}^N$ and perform k-means clustering in the feature space:

$$\min_{\{\mu_j\}_{j=1}^k} \sum_{i=1}^{N} \min_j \|\phi(x_i) - \mu_j\|^2 \tag{3}$$

One representative sample (closest to each cluster center μ_j) is selected per cluster. The hybrid strategy combines margin-based uncertainty with diversity sampling. For each sample, we compute the margin between the two most probable classes:

$$s_{\text{Margin}}(x) = 1 - (p_{(1)} - p_{(2)}) \tag{4}$$

where $p_{(1)}$ and $p_{(2)}$ are the highest and second-highest predicted class probabilities. Among the top $\beta \cdot k$ most uncertain samples, k final selections are made via k-means clustering in the feature space as in Equation (3), using $\phi(x)$ of the filtered pool. The design of the hybrid strategy was inspired by the approach proposed in Chowdhury [2023], while the implementations of entropy sampling, least confidence, and KMeans selection were guided by the reference implementation provided in CURE Lab. As a baseline, we employed a model that randomly selects the images from the unlabeled pool to be labeled next. First all experiments were conducted on the binary classification problem, the finetuning experiments were extended to the multi-class problem. Initial experiments used ResNet34, starting with zero labeled samples and proceeding through five AL rounds. In each round 10 new samples (5 per strategy) were selected and labeled, and the model fine-tuned for 10 epochs per round using the Adam optimizer and a batch size of 8 (see Figure 2). To assess strategy stability, performance was averaged over 5 independent trials. During early experimentation we unexpectedly observed that random sampling occasionally matched or outperformed structured AL

strategies. This discrepancy prompted an in-depth validation of our pipeline. We identified multiple factors potentially confounding the observed outcomes. First, uncertainty-based methods rely on model confidence scores, which are often unreliable in early training stages, especially with minimal supervision. Second, our design choice to fine-tune the model over successive rounds, without reinitialization, may have led to bias accumulation, as errors from early misclassified samples could propagate. While reinitializing the model at each round could have isolated the marginal effect of each batch, we deemed it computationally intensive and inconsistent with real-world deployment scenarios. Lastly, the combination of small batch size (8 samples) and short training durations (10 epochs) likely restricted the model's ability to extract meaningful signal from the few labeled samples. To address these limitations, we redesigned our training protocol following best practices highlighted in Saifullah et al. [2023]. Specifically, we pre-trained the model on 5% randomly selected labeled data, then increased the labeled set by 2.5% each round, and trained the model for 50 epochs using a larger batch size and the SGD optimizer per round. Early stopping based on validation loss was introduced to prevent overfitting and improve convergence stability. To robustly assess generalization and variance, all strategies were evaluated over multiple seeds. Furthermore, we switched the backbone model to a ResNet18 to counteract increased training times. To assess whether AL yields greater benefits in a more complex setting, we applied the best-performing strategy, entropy sampling, to the multi-class classification task. To gain deeper insight into the influence of earlier design choices, we conducted tests on the ratio between AL selected and randomly selected samples during training. For each configuration, a single training run was performed to evaluate its impact on final test performance. The resulting performance trends and quantitative comparisons are presented in Section 5.2.

5 Experiments

5.1 Basic Project

Without hyperparameter tuning we achieved test accuracies over 99% on the binary classification problem. As our baseline for further experiments on the multi-class problem we used the validation accuracy of 89.54%, which was achieved by fine-tuning the classification layer for 10 epochs. We first experimented with fine-tuning the last l residual blocks of ResNet. As shown in Figure 3, the runtime increased linearly with larger l. Fine-tuning lower layers degraded the performance, likely due to disruption of low-level feature representations. The best performance of 89.67% validation accuracy was achieved by fine-tuning only the last block together with the classification layer. For gradual unfreezing (Figure 4), we noticed that using too few epochs per unfreezing step resulted in underfitting and poor performance. With 5 epochs per step, performance improved over the first step but declined afterward. Unfreezing only the last block gave the best results with a validation accuracy of 90.02%. A reason for this behaviour could be that lower layers contain very broad features that are sensitive to big changes. Therefore, we implemented a learning rate decay with a starting value of 0.003, which is divided by 0.1 after every step, which solved this problem and allowed for fine-tuning additional lower blocks as shown in Figure 5. Additionally, we introduced L2-regularization of 0.001, decayed in the same way as the learning rate. Together with data augmentation, where we added random horizontal flipping, this introduced a good amount of regularization to counter overfitting. Additionally, we included batch normalization parameters, which improved performance slightly. Our tuned ResNet18 model could reach a top validation accuracy of 91.58% and a test accuracy of 88.44%. Using deeper architectures, improved this performance achieving a test accuracy of 92.43% with ResNet101. This is already very close to state-of the art methods such as the TWIST architecture, underlining the predictive power of ResNets. The summarised results for the test runs on the basic multi-class model, without the aforementioned fine-tuning strategies, trained over 25 epochs, can be found in Table 1. This model performed surprisingly well on the imbalanced dataset, achieving 79.94% in total. Focusing on the species-specific accuracies however, the accuracy on cat breeds dropped from 82.92% to 70.92% when the number of cat examples was reduced. While using a weighted cross-entropy loss achieved a slightly higher total accuracy of 80.21%, this performance is attributed to the model achieving better results on dogs rather than cats, counter to the desired result. The second strategy, involving sampling from all breeds with replacement to achieve a balanced dataset, did not prove successful, leading to a drop in accuracy across all breeds, likely due to overfitting on repeated examples.

5.2 Extension: Active Learning

5.2.1 Binary Classification

To evaluate the effectiveness of our active learning strategies, we designed a series of experiments assessing classification accuracy and learning stability over multiple annotation rounds. First, we analyzed the top-5 samples selected by each strategy from an untrained model to assess early-stage selection behavior. Informativeness based methods (entropy, least confidence) consistently prioritized cat images, indicating a bias in initial confidence estimates. In contrast, KMeans clustering selected a more balanced set dominated by dog images, reflecting its focus on data representativeness, see 6. For our initial experiment we selected the hybrid method as done in Chowdhury [2023], the setup is as described in Section 4.2. Figure 7 showed that random sampling outperformed hybrid selection averaged across five runs. To validate these findings, we increased the number of samples added per round to 10. Despite this adjustment, performance variance remained high and no clear advantage of AL-based selection was observed (see Figure 8). We hypothesized that selecting only AL-prioritized samples may lead to overfitting on rare or unrepresentative examples. To investigate whether this contributed to the limited performance of active learning we compared three model variants: (i) pure random sampling, (ii) pure hybrid sampling, and (iii) a mixed setup augmenting 10 hybrid samples with 5 random ones. As shown in Figure 9, the hybrid+random combination exhibited reduced volatility and improved stability across rounds, whereas the hybrid-only variant showed inconsistent trends and declining accuracy. Based on these insights, we adopted a mixed-sampling strategy for further comparisons: five samples selected per AL method augmented with five random samples per round. Next we investigated which AL method performs best in our setting; entropy, least confidence, KMeans, or hybrid. Results in Figure 10 indicate that entropy sampling, when combined with random selection, consistently achieved the most stable and effective performance. In contrast, the hybrid strategy remained volatile, showing sensitivity to model initialization and unreliable early-stage uncertainty estimates. To benchmark these results, we reintroduced a random-only baseline in Figure 11. Notably, random sampling performed comparably to all AL-enhanced approaches in early rounds, highlighting a key limitation of active learning in low-data regimes: when model confidence estimates are poorly calibrated, random selection can be equally effective. Given that each round had an addition of only 0.14% from the full training set, starting with a model only pretrained on ImageNet the selected subsets may have been too small to offer statistically representative guidance for model updates. In light of these limitations, we adopted a revised training protocol following best practices from Saifullah et al. [2023], as detailed in Section 4.2. This included a warm-start with 5% labeled data, increased training duration from 5 to 7 rounds, and more robust hyperparameters, i.e. increased batch size, changed optimizer to SGD. An additional adjustment made was the switch from ResNet34 to ResNet18 to reduce computational burden under limited GPU availability. These changes improved overall performance drastically and accentuated differences between strategies. Illustrated in Figure 12, a general trend towards rising accuracy as the number of rounds increases, can be observed. While all strategies achieved accuracies above 98.50% after 7 rounds, some showcased more stable performances than others with the purely random selection proving rather volatile. Despite hybrid methods having performed well on similar tasks in the literature, such as in He et al. [2024], Chowdhury [2023], this approach failed to match the performance of the other strategies, outperforming only the random sampling baseline in the last round. The final test accuracies on an independent hold-out set are summarized in Table 2, with entropy + random yielding the best average accuracy of 99.38%.

5.2.2 Multi-Class Classification

Overall, our findings suggest that entropy-based AL, when combined with random sampling, offers the most robust learning dynamics under constrained labeling conditions. However, the observed performance gains over random sampling were modest. Given that uncertainty-based approaches, such as Entropy, have yet been shown to achieve robust performance and resilience under class imbalance, noise, and data bias on multi-class datasets, such as Tobacco3482 (Saifullah et al. [2023]), it is therefore possible that the classification task was not sufficiently complex to showcase the full benefits of active learning. To test our approach under more challenging conditions, we extended the best-performing model, i.e. entropy-based sampling with an inital amount of randomly selected labeled images, from binary to multi-class classification. To draw deeper insights into the potential of strategically versus randomly selected examples, we fine-tuned the ratio of random samples in each round, varying it from 0% to 100% in 25% increments. All configurations, visualised in Figure 13

over seven rounds, including the random selection baseline, demonstrated adequate performances. As seen in earlier experiments, random sampling performed competitively in early rounds, but, as the number of rounds increases, models with higher entropy-based selections improved over the baseline in terms of stability and accuracy, with the pure entropy strategy ultimately achieving 85.62%. Finally, using the gained insights and understanding from earlier experiments, we replaced the backbone of this best-tuned model with ResNet101 and trained it over four rounds. This configuration achieved a test accuracy of 89.64%, using only 15% of the labeled data.

6 Conclusion

In this project, we explored how active learning can reduce annotation effort while maintaining high classification accuracy. Our best model in the standard supervised setup reached 99.82% test accuracy using 5180 labeled images, requiring 2400 seconds of training. In comparison, our entropy-based AL strategy achieved 99.66% accuracy with only 910 labeled samples (82.4% fewer) and 980 seconds of training (59.2% less). This shows that AL can yield comparable performance at a fraction of the annotation and computational cost. Among the strategies tested, entropy sampling with random augmentation consistently produced the most stable results. Hybrid methods, while conceptually appealing, exhibited high variance and sensitivity to model initialization. Our findings suggest that uncertainty-based AL strategies, when combined with minimal exploration and robust initialization, offer a practical and efficient approach in low-label settings. Future work could explore more expressive architectures (e.g., Vision Transformers) and evaluate AL performance under domain shift or class imbalance.

7 URL to Code Repository

`https://github.com/ge96lip/Explore-Transfer-Learning`

References

Visual Geometry Group - University of Oxford — robots.ox.ac.uk. https://www.robots.ox.ac.uk/~vgg/data/pets/. [Accessed 08-05-2025].

A. Chowdhury. How to train neural networks with fewer data using active learning. https://readmedium.com/how-to-train-neural-networks-with-fewer-data-using-active-learning-445154c30ddf, 2023. Accessed: 2025-05-15.

CURE Lab. deep-active-learning: Query strategies for deep active learning. https://github.com/cure-lab/deep-active-learning. Accessed: 2025-05-15.

Yinan He, Lile Cai, Jingyi Liao, and Chuan-Sheng Foo. Hybrid active learning with uncertainty-weighted embeddings. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=jD761b5OaE. Accepted by TMLR.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021. doi: 10.1145/3472291.

Saifullah, Stefan Agne, Andreas Dengel, and Sheraz Ahmed. Analyzing the potential of active learning for document image classification. *International Journal on Document Analysis and Recognition (IJDAR)*, 26:1–23, 04 2023. doi: 10.1007/s10032-023-00429-8.

Feng Wang, Tao Kong, Rufeng Zhang, Huaping Liu, and Hang Li. Self-supervised learning by estimating twin class distributions, 2021. URL https://arxiv.org/abs/2110.07402.

A Additional Figures

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 1: ResNet-18 Architecture.

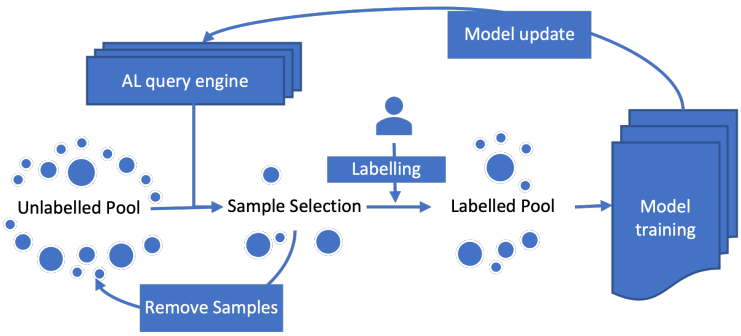


Figure 2: Simplified Model Training pipeline

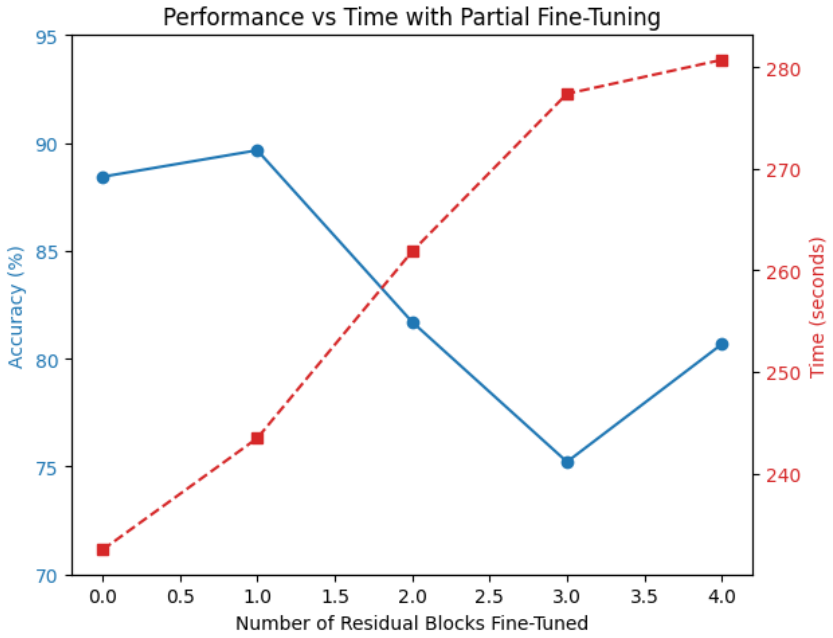


Figure 3: Accuracy and runtime for strategy 1 (partial fine-tuning). For each number of blocks to be fine-tuned the model was trained for 10 epochs.

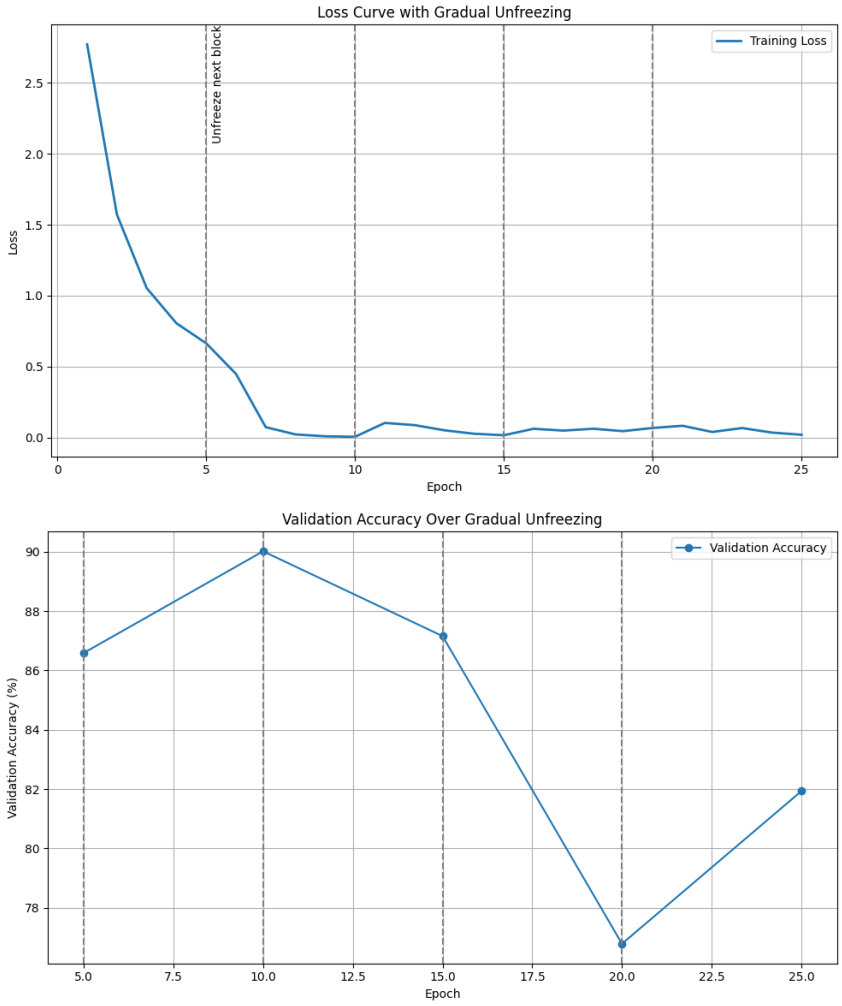


Figure 4: Training loss and validation accuracy curves for strategy 2 (gradual unfreezing). The model was trained on 25 epochs and after every 5 epochs another residual block was unfrozen.

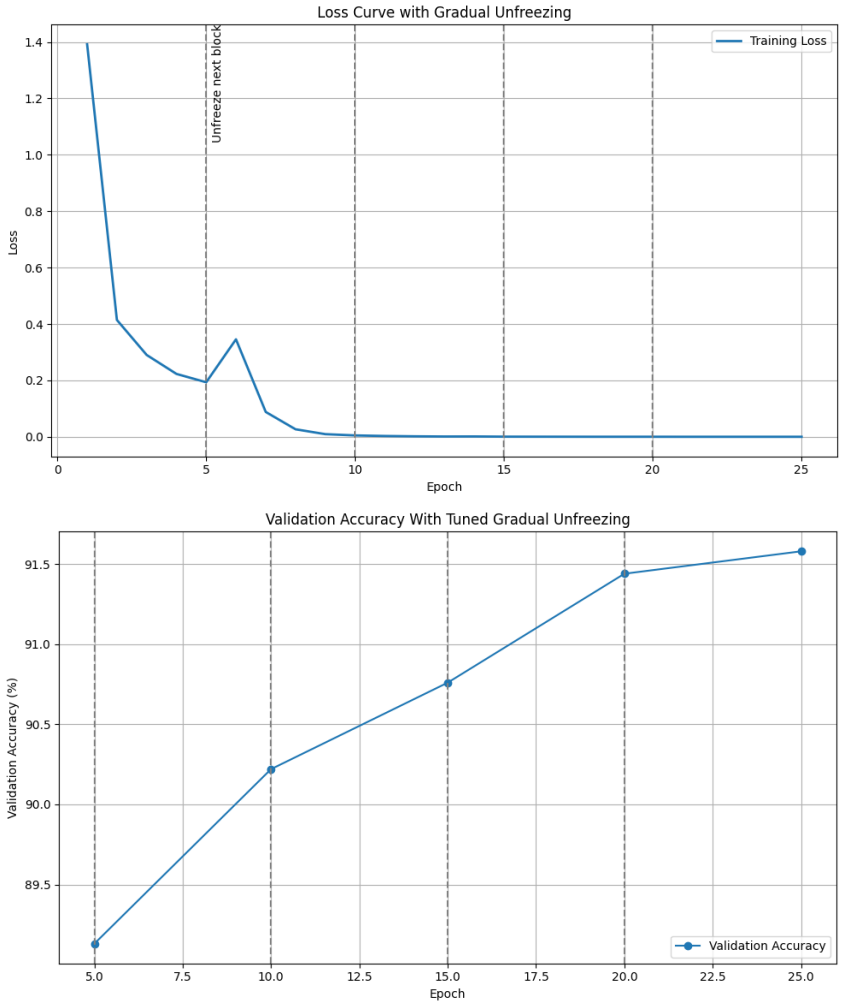


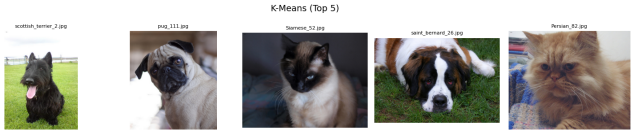
Figure 5: Training loss and validation accuracy curves for best performing model including optimization described in section 5.1. The model was trained on 25 epochs and after every 5 epochs another residual block was unfrozen.

Scenario	Total Acc.	Cat Acc.	Dog Acc.
Balanced (Baseline)	83.51%	82.92%	83.79%
Imbalanced + CE Loss	79.94%	70.92%	84.23%
Imbalanced + Weighted CE Loss	80.21%	68.13%	85.96%
"Rebalanced" + CE Loss	69.91%	53.34%	77.80%

Table 1: Final test accuracies, aggregated over cat and dog breeds, of a baseline and four networks handling class-imbalance. All models were trained on 25 epochs.

(a) Entropy sampling

(b) Least Confidence



(c) KMeans clustering

(d) Hybrid strategy

(e) Random baseline

Figure 6: Top-5 samples selected in the first round by each active learning strategy from an untrained model. Differences across strategies highlight variations in selection behavior.

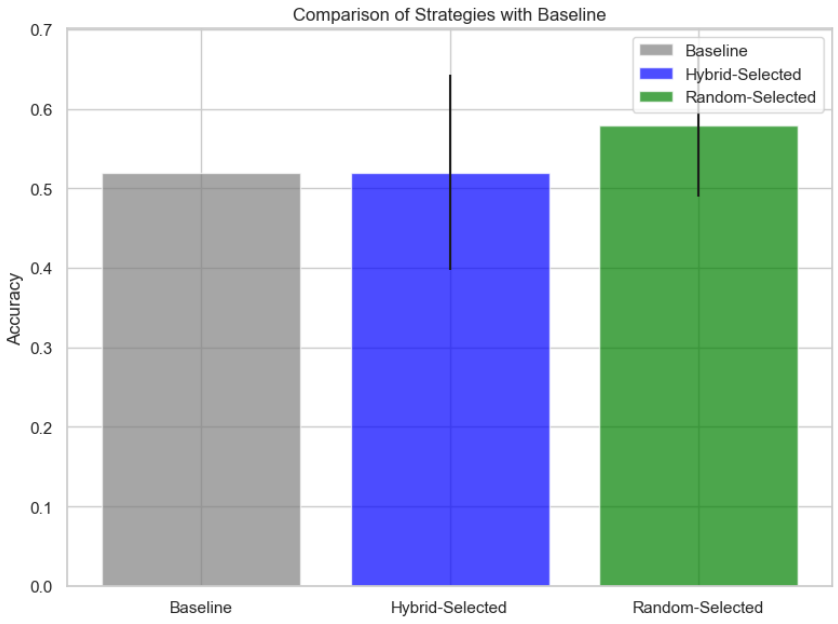


Figure 7: Accuracy comparison of hybrid strategy vs. random selection after fine-tuning on 20 samples.

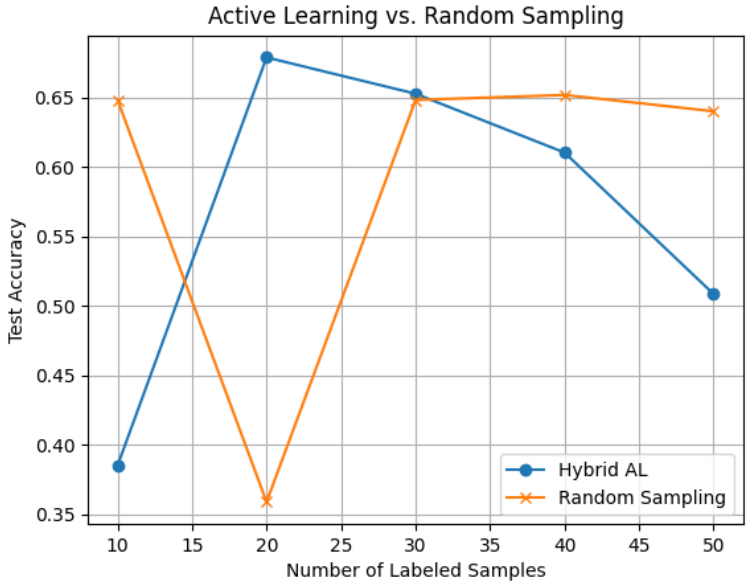


Figure 8: Hybrid vs. random performance with 10 new samples per round.

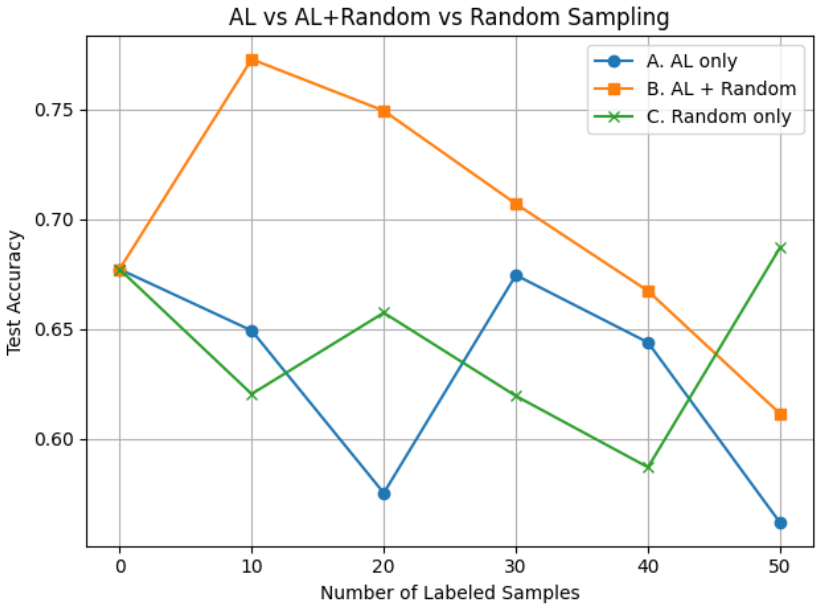


Figure 9: Comparison of pure hybrid, hybrid + random, and random-only strategies.

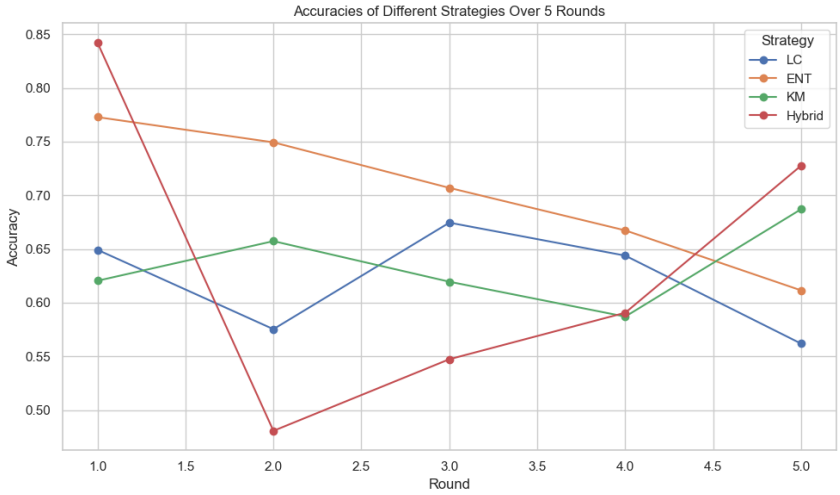


Figure 10: Performance comparison across entropy, hybrid, least confidence, and KMeans strategies.

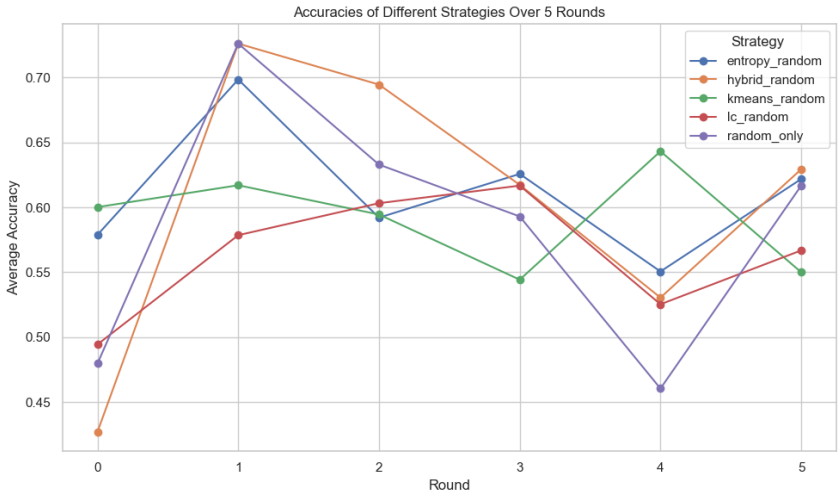


Figure 11: Random sampling compared to active learning strategies with added random augmentation.

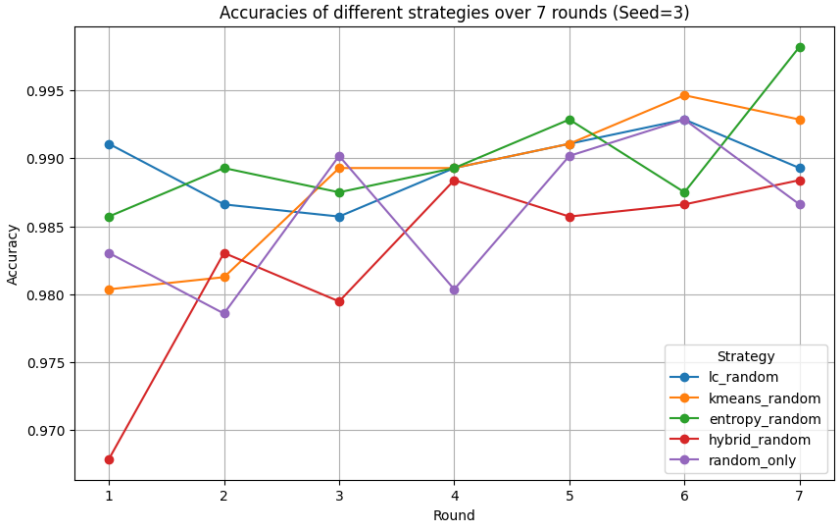


Figure 12: Accuracies across rounds for different strategies. The initialisation round 0 was omitted for clarity of the plot.

Strategy	Seed	Accuracy (%)	Avg. Accuracy (%)
lc_random	3	98.93	98.75
	12	98.57	
kmeans_random	3	99.29	98.89
	12	98.48	
entropy_random	3	99.82	**99.38**
	12	98.93	
hybrid_random	3	98.84	98.53
	12	98.21	
random_only	3	98.66	98.66
	12	98.66	

Table 2: Test accuracies in final round per strategy, for different seeds.

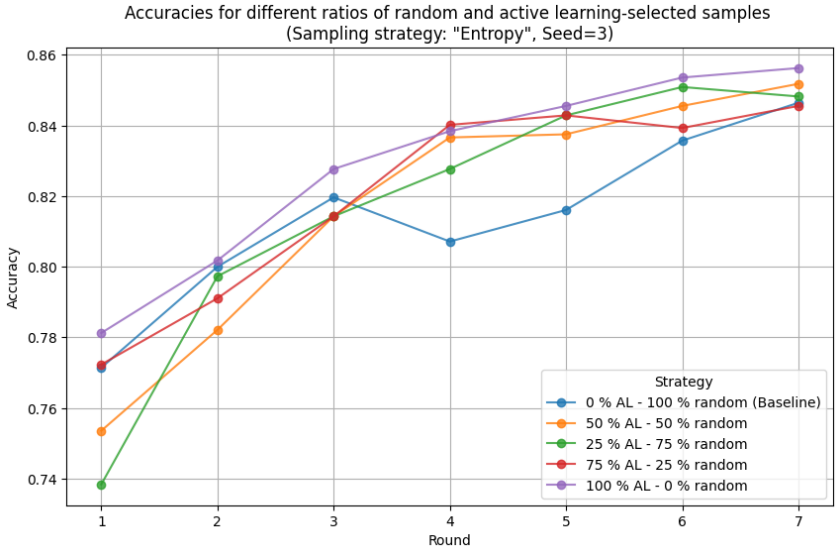


Figure 13: Accuracies on the multi-class task, across rounds for different ratios of sampling between random selection and active learning (entropy-based) selection. The initialisation round 0 was omitted for clarity of the plot.