Transfer learning based approach to bounding box leaf-disease detection

C Mahidharah Rajendran
George Washington University
mahidharah@u.nus.edu

Abstract

Cocoa diseases such as Cocoa Swollen Shoot Virus Disease (CSSVD) threaten the livelihoods of millions of smallholder farmers across West Africa. As part of a machine learning contest supported by Makerere AI Lab, Marconi Machine Learning Lab, and the Makerere University Research Innovations Fund (RIF), this work explores mobile-friendly object detection models to identify disease in cocoa leaves. We adopt YOLOv8s, leveraging the proven strengths of convolutional neural networks (CNNs) in spatial pattern recognition, and evaluate three transfer learning strategies: training from scratch, freezing pretrained backbones, and progressive unfreezing. The results show that pre-trained weights, even in unrelated domains, significantly improve accuracy and convergence. Progressive unfreezing offers the best trade-off between performance and efficiency, making it ideal for edge deployment. This research contributes to Amini's broader mission to build scalable, AI-powered solutions that enable data-driven transformations globally.

1 Introduction

Cocoa is a vital cash crop for more than 5 million smallholder farmers worldwide, especially in West Africa where countries like Ghana and Côte d'Ivoire contribute more than 60% to global cocoa production[9, 28]. However, the sector is threatened by diseases such as Cocoa Swollen Shoot Virus Disease (CSSVD), black pod, and leaf spot, which can cause significant yield losses if not detected early. Traditional disease identification methods, typically relying on trained agricultural officers or manual inspection, are slow, costly, and often inaccessible to rural farmers[27, 56]. As climate change exacerbates the spread and emergence of disease, there is an urgent need for scalable automated solutions that empower farmers to diagnose and manage diseases in real-time.

Neural networks and deep learning have emerged as transformative tools in computer vision, capable of learning hierarchical and abstract representations from raw image data [18, 16, 33]. These models outperform classical machine learning approaches in a range of image recognition tasks, including object classification, segmentation, and detection [3, 13]. In the agricultural domain, convolutional neural networks (CNNs) have been particularly effective in recognizing leaf patterns, textures, and disease symptoms, often under varying lighting and environmental conditions [21, 5, 29]. Their inductive biases—such as local connectivity and weight sharing—make them particularly well-suited for extracting spatially structured features from plant imagery [16].

While powerful, many deep learning models are computationally intensive and not directly suited for deployment in resource-constrained environments. For real-world agricultural use cases—especially on mobile phones used by farmers in the Global South—models must be lightweight, efficient, and capable of real-time inference [35, 55]. The YOLO (You Only Look Once) family of object detectors is particularly promising in this regard, offering a single-pass architecture that combines object localization and classification in a single step [38, 24]. Compared to traditional image classification models, YOLO provides an actionable abstraction that allows users to identify multiple regions

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

of interest within an image, making it ideal for point-and-shoot plant diagnostics on edge devices [24, 34].

When labeled data is limited, a variety of techniques have been employed to improve the generalization ability of neural networks. Data augmentation - through geometric transformations, color jittering, or noise injection - artificially increases the diversity of the dataset and is widely used to reduce overfitting [44, 4]. Synthetic data generation and semi-supervised learning further leverage unlabeled data or simulated environments to enhance training [45, 25]. Regularization methods such as dropout, batch normalization, and early stopping also mitigate overfitting in low-data regimes [42, 46]. While each of these techniques has merits, transfer learning has gained widespread adoption as a pragmatic and efficient solution to small datasets, particularly in vision tasks where pretrained convolutional neural networks can be adapted to new tasks with minimal labeled data [23].

Transfer learning is especially effective with CNNs because the lower layers of these networks learn general-purpose visual features such as edges, blobs, and textures, which are transferable across domains [43]. These representations, once learned from large-scale datasets like ImageNet or COCO, can be fine-tuned on task-specific data by adapting higher-level features, leading to faster convergence and improved performance [54, 47]. Studies show that even when the source and target domains are not closely related, CNNs can retain useful representations that accelerate learning and reduce the need for extensive annotation [41]. This has made transfer learning a cornerstone of visual recognition systems in domains like medical imaging [11], remote sensing [37], and increasingly in agriculture, where CNN-based approaches pretrained on large-scale datasets have been successfully applied to plant disease detection, including on cassava, tomatoes, and cocoa leaves [52, 31, 20].

Increasingly in agriculture, CNN-based approaches pretrained on large-scale datasets have been successfully applied to plant disease detection, including on cassava, tomatoes, and cocoa leaves [54, 31, 20]. These successes demonstrate the effectiveness of leveraging generalized visual representations learned from broader image domains and adapting them to specific agricultural tasks with limited data. For instance, Ramcharan et al. showed that a transfer learning-based model achieved high accuracy in identifying cassava leaf diseases, even with relatively few training samples [52]. Similarly, Mohanty et al. used a pretrained AlexNet to classify 14 crop species with over 26 diseases, highlighting the ability of transfer learning to generalize across diverse plant types [31].

This makes transfer learning a go-to technique in agricultural computer vision. Unlike domains such as autonomous driving or facial recognition—where large, annotated datasets like ImageNet or COCO are readily available—agriculture suffers from a scarcity of labeled data. This is due to factors such as the difficulty of capturing large-scale and diverse crop imagery, seasonal constraints, and the need for expert annotation to identify diseases accurately. As noted by Ferentinos, obtaining consistent and annotated agricultural datasets is logistically challenging and expensive, often limiting the use of conventional deep learning approaches [7]. Transfer learning alleviates these issues by allowing researchers to fine-tune models pretrained on general-purpose image datasets, significantly reducing data and computational needs while maintaining strong performance.

Despite recent advancements, existing methods suffer from limitations such as insufficient application of transfer learning across domains with different tasks, and the lack of models specifically designed for cocoa leaf disease detection. Furthermore, there is a need for more lightweight models that can be deployed on resource-constrained devices like smartphones, particularly for agricultural applications where data is scarce. My work addresses these gaps by proposing a transfer learning approach that leverages pretrained YOLOv8 models for cocoa leaf disease detection, specifically for CSSVD, and explores various strategies such as progressive unfreezing to enhance model performance on edge devices.

My contributions are:

- An application of transfer learning for YOLOv8 models in cocoa leaf disease detection, specifically for CSSVD.

- Exploration of various transfer learning strategies including fine-tuning, frozen backbone, and progressive unfreezing to optimize model performance.

- Validation of these strategies on given custom dataset of cocoa leaf diseases, with comparisons against other model configurations to assess the effectiveness of transfer learning.

2 Related Work

Hassan et al. (2021) investigated the use of CNN-based models combined with transfer learning for plant disease identification, focusing on lightweight architectures for mobile deployment. Their study evaluated InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0, confirming that CNNs significantly outperform traditional image processing methods for plant disease classification. MobileNetV2, in particular, achieved a compelling balance between accuracy and model size, making it suitable for edge deployment — a consideration central to this study [12].

Focusing on cocoa diseases, early studies by Kouassi et al. (2024) and Atianashie (2023) applied custom CNNs to identify symptoms of Cocoa Swollen Shoot Virus Disease (CSSVD), establishing deep learning's utility in this domain. However, these efforts typically lacked transfer learning, relying instead on models trained from scratch.

Tovurawa et al. (2025) deviated from this trend by fine-tuning VGG16, ResNet50, and Vision Transformer (ViT) models on the KaraAgroAI Cocoa Dataset. While their results marked a step forward in applying transfer learning to CSSVD detection, their chosen architectures were not optimized for deployment in resource-constrained settings. In contrast, this study emphasizes models like YOLOv8s for their real-time capability and efficiency.

To further improve robustness and generalization, multi-task learning (MTL) has emerged as a complementary strategy. Jiang et al. (2021) showed that multi-task deep transfer learning could improve classification and generalization on rice and wheat disease datasets. Similarly, Lee et al. (2023) introduced Conditional Multi-Task Learning (CMTL), linking host species and disease representations, leading to improved plant disease classification over joint pairwise modeling [19].

MobileNetV3 has also been leveraged in MTL contexts. Zhang et al. (2023) combined a shallow CNN with spatial attention and a custom MobileNetV3Large-Attention model to deliver rapid and accurate leaf disease classification for edge devices [53].

To effectively implement MTL or transfer learning for CSSVD, task similarity is crucial. CSSVD's symptoms — chlorosis, mottling, and leaf deformation — are visually similar to those of Cassava Brown Streak Disease, Grapevine Leafroll Disease, and Tomato Yellow Leaf Curl Virus (TYLCV). Ramcharan et al. (2017), Nagi and Tripathy (2022), and Saeed et al. (2023) successfully applied transfer learning using InceptionV3, VGG16, and Inception-ResNetV2 to these diseases respectively [32, 26, 39].However, these studies are more applicable to transfer learning and CNN use in classification problems.

For object detection, bounding-box and classification problems, advancements in YOLO-based architectures have further solidified the value of lightweight object detectors for plant disease tasks. Yan and Yang [50] introduced FSM-YOLO, which enhances YOLOv8 by incorporating an Adaptive Feature Enhancement Module (AFEM) and Spatial Context-aware Attention (SCAA) for improved feature learning in unstructured environments. Their approach yielded a 2.7% mAP@0.5 gain over the baseline YOLOv8s, demonstrating the importance of channel modulation and spatial reasoning in early-stage disease detection. Complementing this, Gao et al. proposed a YOLOv5-Efficient variant tailored for rice leaf disease detection [10], achieving high precision by integrating attention mechanisms and model compression techniques. Similarly, Wang et al. developed LCGSC-YOLO [48], which combines the lightweight LCNet backbone and GSConv modules to achieve competitive accuracy with low computational cost, making it suitable for real-time mobile deployment. These studies collectively underscore the effectiveness of structurally optimized YOLO variants for plant disease detection and offer complementary perspectives to our methodology, particularly in balancing performance and efficiency for edge applications.Moreover, recent research has shown the effectiveness of transfer learning in plant disease detection using YOLO-based models. For example, Kumar et al. (2024) applied transfer learning to enhance the detection of tomato leaf diseases using YOLOv5 and YOLOv7 models [17]. Their results demonstrated significant improvements in model accuracy, achieving mAP values of 98.8% and 98.7% at detection thresholds of 0.5 and 0.95, respectively. This study emphasizes the potential of YOLO for real-time disease diagnosis in tomato crops through transfer learning.The research to date indicates a strong foundation in applying transfer learning for plant disease detection using CNNs and YOLO models. However, a significant gap remains in the application of YOLO models for CSSVD detection in cocoa leaves, where no studies have employed transfer learning with pretrained YOLO models for this specific disease. Additionally, there is a lack of exploration on transfer learning across tasks that differ in scope, especially in cases where a model trained on one disease or plant species might be repurposed for a different disease on a structurally similar plant species.

This project aims to address these gaps by focusing on transfer learning for YOLO models specifically for cocoa leaf disease detection, particularly CSSVD, and examining how different transfer learning techniques, such as fine-tuning and progressive unfreezing, can enhance model performance on these tasks. The goal is to validate that transfer learning across domain-similar but task-different applications, like cocoa leaf disease detection, can be an effective strategy for improving detection accuracy with limited task-specific data.

3 Methodology

3.1 Evaluation Metric: mAP@0.5

We evaluate detection performance using mean Average Precision at IoU 0.5 (mAP@0.5) – a standard metric in object detection [6]. This metric combines Precision (P) and Recall (R) across detection confidence thresholds to produce a single score per class, then averages over classes for a final mAP [35]. We define:

Precision $P = \frac{TP}{TP+FP}$, the fraction of predicted boxes that are correct.

Recall $R = \frac{TP}{TP+FN}$, the fraction of ground-truth objects correctly detected.

A predicted detection is considered true positive (TP) if its intersection-over-union (IoU) with a ground-truth box exceeds 0.5 (50%) and the class is correct; otherwise it is a false positive (FP). Missed objects count as false negatives (FN). By varying the detection confidence threshold, we obtain a Precision-Recall (PR) curve for each class. The Average Precision (AP) is the area under this PR curve. In practice, AP can be computed by summing precision increments across recall levels. The mean AP (mAP) is then the average of AP values across all object classes [6]. In our case (mAP@0.5), AP is computed at a single IoU threshold of 0.5 (as in the PASCAL VOC challenge) and averaged over the three leaf disease classes.

This mAP metric is especially appropriate for multi-instance, multi-class detection tasks like leaf disease localization, because it balances both the object classification and spatial localization components. Additionally, the metric's established use in COCO and PASCAL VOC [6, 22] makes it the standard for benchmarking object detectors such as YOLO. As shown in the COCO benchmark, models are typically evaluated using multiple IoU thresholds, but mAP@0.5 remains the most interpretable and direct measure for tasks where coarse but correct localization suffices [22].

3.2 Motivation and Constraints of the Study

This study is conducted under practical constraints imposed by an ML competition, which requires submitted models to be lightweight and efficient. Specifically, training must be completed within 9 hours on an NVIDIA T4 GPU, and inference must complete within 3 hours. Models must be exportable to formats like ONNX or TensorFlow Lite to support mobile deployment. These constraints necessitate the use of a model with a small memory footprint and fast inference speed, such as YOLOv8s, which is a compact, one-stage object detector [14].

YOLO models have long been valued for their speed and efficiency in edge applications. For instance, YOLOv3 demonstrated strong real-time performance (45 FPS) on standard detection benchmarks [33], while newer variants like YOLOv4 and YOLOv8 extend this legacy with architectural enhancements for both speed and accuracy [49].

In addition, agricultural applications present a common challenge: data scarcity. Collecting annotated plant disease data is time-consuming, requires expert knowledge, and can suffer from inter-annotator variability. As noted by Ferentinos, obtaining consistent and annotated agricultural datasets is logistically challenging and expensive, often limiting the use of conventional deep learning approaches [8]. Transfer learning helps address this by enabling reuse of representations from large pretrained models, improving performance even when task-specific data is limited [30, 40].

3.3 YOLOv8s Architecture and Pretraining

YOLOv8s consists of three main components: a backbone for feature extraction, a neck for multi-scale feature fusion, and a detection head for bounding box regression and classification [**?**]. The backbone is a modified CSPDarknet53, which utilizes Cross Stage Partial (CSP) connections to reduce computational cost while preserving gradient flow. The neck adopts a Feature Pyramid Network (FPN)-like structure to combine low- and high-level features, enabling robust detection of both small and large leaf lesions. The detection head outputs predictions across three scales.

Pretrained weights are used to initialize the network. Two pretrained sources are considered:

A generic YOLOv8s model trained on the MS COCO dataset.

A domain-specific YOLOv8s model trained on a 46-class leaf dataset provided by FODUU AI [**?**].

The leaf-pretrained model achieved 94.6% mAP@0.5 on its own detection task, suggesting its backbone contains filters tuned to leaf shapes and structures. Since early layers in CNNs typically extract general features (edges, textures), they transfer well between tasks [51].

3.4 Dataset Description

The dataset comprises 7,900 annotated images of cocoa leaves, with three classes: Anthracnose, CSSVD, and Healthy. Class counts are imbalanced, with 4,280 healthy, 3,241 CSSVD, and 2,271 anthracnose instances. The dataset is split into training (70%), validation (15%), and test (15%) sets. No data augmentation or resampling is used to preserve fairness across models, and to simulate the real world application of the models. This dataset is provided by Amini as part of the ML contest, Amini Cocoa Contamination Challenge [2].

3.5 Model Comparison and Transfer Learning Strategies

We evaluate four training setups:

- **Model 1:** COCO-pretrained YOLOv8s, all layers fine-tuned.
- **Model 2:** Leaf-pretrained YOLOv8s, all layers fine-tuned.
- **Model 3:** Leaf-pretrained YOLOv8s, with frozen backbone.
- **Model 4:** Leaf-pretrained YOLOv8s, progressively unfreezing (backbone+neck frozen for 5 epochs, then neck unfrozen for next 5 epochs).

As Kornblith et al. show, fine-tuning deeper layers of high-quality ImageNet models improves transfer learning accuracy when applied appropriately [15]. Model 4 uses progressive unfreezing, a method supported in NLP and vision literature for stable optimization [36].

3.6 Training Configuration

All models are trained for 10 epochs using YOLOv8's default training pipeline. The AdamW optimizer is used with automatic learning rate scheduling. Batch size is 8 with image resolution 1024×1024. No model receives special tuning. mAP@0.5 on the validation set is monitored to select the best model.

4 Experiments

4.1 Quantitative Results Against Baselines

4.1.1 Overall Detection Performance

The comparative performance of different training strategies reveals that transfer learning significantly enhances detection outcomes. The baseline YOLOv8s model, pretrained on the generic COCO dataset, achieved a mAP@50 of 0.660 and mAP@50–95 of 0.410. Integrating domain-specific pretrained weights from the FODUU leaf classification model [1] led to a measurable performance improvement across all metrics. Interestingly, freezing the backbone — thus preserving the feature extraction layers

Table 1: Overall detection performance of YOLO models across different training strategies

Model type	Precision	Recall	mAP@50	mAP@50–95
Vanilla YOLO	0.715	0.596	0.660	0.410
Pretrained YOLO	0.771	0.603	0.688	0.431
Frozen backbone	0.751	0.613	0.692	0.437
Progressive unfreezing (10 epochs)	0.723	0.622	0.700	0.449
Progressive unfreezing (15 epochs)	0.760	0.647	0.733	0.499

trained on a different leaf-related task — slightly increased mAP@50–95 to 0.437. This reinforces the notion that lower convolutional layers encode transferable, domain-agnostic visual features, as discussed in Yosinski et al. [51].

The highest performance was attained through progressive unfreezing, where the model's neck and detection head were fine-tuned first, followed by the backbone. After 15 epochs, this model reached a mAP@50 of 0.733 and mAP@50–95 of 0.499. These results affirm prior evidence that gradual fine-tuning facilitates smoother adaptation of deeper network layers [36], enabling superior generalization in the target task.. Incorporating pretrained weights from a domain-specific leaf model modestly improves performance across all metrics. Interestingly, freezing the backbone yields slightly better mAP@50–95 (0.437), demonstrating that the early convolutional layers retain rich, transferable feature representations learned from a related but entirely different leaf classification task. The most substantial gains come from the progressive unfreezing strategy, particularly with 15 epochs, which reaches a peak mAP@50 of 0.733 and mAP@50–95 of 0.499. These results affirm the value of controlled adaptation when using pretrained models for domain transfer, and that transfer learning when used correctly, can show great performace.

4.1.2 Class-Wise Detection Performance

Table 2: Class-wise detection performance (mAP@50–95) of YOLO models

Model type	Healthy	CSSVD	Anthracnose
Vanilla YOLO	0.421	0.415	0.393
Pretrained YOLO	0.452	0.429	0.412
Frozen backbone	0.445	0.435	0.430
Progressive unfreezing (10 epochs)	0.468	0.444	0.434
Progressive unfreezing (15 epochs)	0.492	0.510	0.493

Per-class analysis further illustrates the efficacy of transfer learning. The model pretrained on FODUU weights with progressive unfreezing over 15 epochs achieved the best class-wise mAP@50–95 across all three classes: Healthy (0.492), CSSVD (0.510), and Anthracnose (0.493). The Anthracnose class, in particular, saw the most significant relative improvement, rising from 0.393 in the vanilla YOLO model. These results suggest that fine-tuning enables better discrimination of complex or underrepresented disease patterns, consistent with findings from Shin et al. [40] in limited medical datasets. across all three disease categories with the use of transfer learning. The progressive unfreezing model (15 epochs) achieves the highest detection accuracy across all classes: Healthy (0.492), CSSVD (0.510), and Anthracnose (0.493). Notably, the Anthracnose class, which initially underperforms in the vanilla model (0.393), sees the greatest relative improvement, validating that deeper fine-tuning helps recover performance in underrepresented categories. Table with comparison scores. Highlight improvement over baselines.

4.2 Insights

4.2.1 Transfer Learning Across Related Domains

This study further validates the premise that transfer learning can be effectively applied across domains with shared visual structure, even when tasks differ. The FODUU model, originally trained to classify 46 types of plant leaves, enabled improved detection in our cocoa leaf disease task. The notable performance of the frozen-backbone model (mAP@50 = 0.692) supports the argument that early layers of CNNs, which capture low-level features such as texture and edges, remain valuable when transferred to structurally similar but task-divergent applications [51]. These findings extend the insights of Pan and Yang [30] and Kornblith et al. [15], who emphasized the generality of learned features across domains with related characteristics. of transfer learning between related domains, even when the downstream task differs. While the FODUU model was trained to classify 46 plant species by leaf type (a classification task), its low-level filters generalize well to our object detection task on cocoa leaf diseases. This supports the hypothesis that features extracted in the backbone layers are largely domain-agnostic and can be reused effectively across tasks that share structural visual characteristics [51]. The strong performance of the frozen-backbone model (0.692 mAP@50) demonstrates that even without retraining early layers, these features offer significant utility. As a result, practitioners should consider not only transferring between identical tasks, but also between structurally similar domains with different task objectives.

4.2.2 Epoch Sensitivity and Training Duration

The experiment extending progressive unfreezing to 15 epochs, although beyond the competition constraints, was designed to explore the upper bound of potential gains from deeper adaptation. The results indicate that increasing training duration leads to consistent improvements in detection performance—mAP@50 rising from 0.700 (10 epochs) to 0.733 (15 epochs), and mAP@50–95 from 0.449 to 0.499. This corroborates existing literature that highlights the importance of sufficient training epochs for gradual adjustment of pretrained layers [36], particularly in tasks requiring fine-grained localization. just to explore what more gains transfer learning can show, beyond the scope of the contest. Comparison between the 10 and 15 epoch versions of the progressively unfrozen model highlights the importance of sufficient adaptation time. The jump in mAP@50 from 0.700 to 0.733, and in mAP@50–95 from 0.449 to 0.499, indicates that longer progressive training enables deeper layers to recalibrate gradually, which aligns with literature supporting layer-wise fine-tuning [36].

4.3 Summary of Findings

The key findings from this evaluation are:

- Transfer learning from a domain-specific plant classification model significantly improves leaf disease detection.
- Freezing the backbone retains generic visual features that remain relevant to new tasks, minimizing the need for full retraining.
- Progressive unfreezing yields the highest performance, allowing for gradual and stable adaptation of deep layers.
- Transfer learning should be considered not only between identical tasks, but also across related domains with shared structural characteristics, particularly in resource-limited agricultural applications.

5 Conclusion

In this study, we identified a gap in the application of transfer learning across tasks in similar domains. Through our experimentation, we demonstrated that transfer learning provides considerable gains in model performance and should be considered whenever curating a model for a new task. This approach not only optimizes performance but also enables learning from previously trained models, allowing for the reuse of learned representations that can be valuable for new applications. Coupled with other techniques such as data generation, this approach can lead to the creation of more effective models.

Future Work: Moving forward, there is potential to extend transfer learning across other related domains, where sufficient literature supports task similarity. Additionally, more sophisticated transfer learning techniques can be explored, particularly those that go beyond simple fine-tuning to enhance model performance. Given the constraints of this project, we focused on efficiency, but future work can explore more advanced transfer learning techniques. Another exciting direction is the exploration of hybrid models, where transfer learning might not only be applied within similar architectures but also across different model types. For example, transferring knowledge from classification models to object detection models, such as YOLO, remains largely unexplored but holds significant promise for leveraging existing classification research in the domain of detection.

References

[1] FODUU AI. Yolov8 leaf detection and classification, 2023. https://huggingface.co/foduucom/plant-leaf-detection-and-classification.

[2] Amini.ai. Amini cocoa contamination challenge. `https://zindi.africa/competitions/amini-cocoa-contamination-challenge/data`, Feb 2025. Accessed: 2025-02-14.

[3] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. arXiv preprint arXiv:2004.10934.

[4] M. Brahimi, K. Boukhalfa, and A. Moussaoui. Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4):299–315, 2017.

[5] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[7] K. P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[8] K. P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[9] World Cocoa Foundation. Cocoa market update. *World Cocoa Foundation Report*, 2020.

[10] W. Gao, C. Zong, M. Wang, H. Zhang, and Y. Fang. Intelligent identification of rice leaf disease based on yolo v5-efficient. *Crop Protection*, 183:106758, 2024.

[11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[12] S. Hassan, S. Tariq, and S. M. Daudpota. Comparative analysis of lightweight cnn architectures for mobile-based plant disease classification. *Computers and Electronics in Agriculture*, 191:106510, 2021.

[13] G. Jocher, A. Chaurasia, J. Qiu, and A. Stoken. Yolo by ultralytics, 2023. https://github.com/ultralytics/yolov8.

[14] G. Jocher and et al. Yolo by ultralytics, 2023. https://github.com/ultralytics/ultralytics.

[15] S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 2661–2671, 2019.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, page 1097–1105, 2012.

[17] N. S. Kumar, J. Sony, A. Premkumar, M. R, and J. J. Nair. Transfer learning-based object detection models for improved diagnosis of tomato leaf disease. *Procedia Computer Science*, 235:3025–3034, 2024.

[18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] H. Lee and et al. Conditional multi-task learning for plant species and disease identification. *Computers and Electronics in Agriculture*, 205:107627, 2023.

[20] M. Lin, Q. Chen, and S. Yan. Network in network, 2013. arXiv preprint arXiv:1312.4400.

[21] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, page 2980–2988, 2017.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, ..., and P. Dollár. Microsoft coco: Common objects in context. In *European conference on computer vision*, page 740–755, 2014.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, page 740–755, 2014.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, page 21–37, 2016.

[25] S. P. Mohanty, D. P. Hughes, and M. Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016.

[26] M. Nagi and B. K. Tripathy. Transfer learning-based identification of grapevine diseases using vggnet variants. *Computers and Electronics in Agriculture*, 199:107150, 2022.

[27] A. Ofori, F. K. Padi, G. A. Ameyaw, A. M. Dadzie, M. Opoku-Agyeman, O. Domfeh, and F. O. Ansah. Field evaluation of the impact of cocoa swollen shoot virus disease infection on yield traits of different cocoa (theobroma cacao l.) clones in ghana. *PLOS ONE*, 17(1):e0262461, 2022.

[28] International Cocoa Organization. Annual report. *International Cocoa Organization Report*, 2021.

[29] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. B. da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279, 2020.

[30] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[31] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning, 2017. arXiv preprint arXiv:1712.04621.

[32] A. Ramcharan and et al. A mobile-based deep learning model for cassava disease diagnosis. *Frontiers in Plant Science*, 8:1852, 2017.

[33] J. Redmon and A. Farhadi. Yolov3: An incremental improvement, 2018. arXiv preprint arXiv:1804.02767.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[35] H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 658–666, 2019.

[36] S. Ruder. Fine-tuning pretrained language models: Weight freezing and layer-wise learning rates. `https://sebastianruder.com/`, 2018.

[37] O. Russakovsky, J. Deng, H. Su, and et al. Imagenet challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[38] O. Russakovsky, J. Deng, H. Su, and et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[39] S. Saeed and et al. Deep learning-based classification of tomato leaf curl virus using inception-resnet architectures. *Journal of Plant Pathology*, 105(3):765–774, 2023.

[40] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, ..., and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298, 2016.

[41] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:60, 2019.

[42] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016:3289801, 2016.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 1–9, 2015.

[44] Y. Tang and Y. He. Deep learning for plant disease detection. *Trends in Plant Science*, 27(7):728–745, 2022.

[45] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161:272–279, 2019.

[46] Tzutalin. Labelimg, 2015. GitHub repository. `https://github.com/tzutalin/labelImg`.

[47] Ultralytics. Yolov8 documentation, 2023. https://docs.ultralytics.com/.

[48] J. Wang, C. Qin, B. Hou, Y. Yuan, Y. Zhang, and W. Feng. Lcgsc-yolo: a lightweight apple leaf diseases detection method based on lcnet and gsconv module under yolo framework. *Frontiers in Plant Science*, 15:1398277, 2024.

[49] P. Wang and et al. An improved yolov4 method for real-time vehicle detection. *Computers, Materials & Continua*, 68(3):4223–4236, 2021.

[50] C. Yan and K. Yang. Fsm-yolo: Apple leaf disease detection network based on adaptive feature capture and spatial context awareness. *Digital Signal Processing*, 155:104770, 2024.

[51] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27, 2014.

[52] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *arXiv preprint arXiv:1710.04026*, 2017.

[53] Y. Zhang, X. Liu, and H. Chen. Dual-task lightweight cnn framework for real-time plant disease diagnosis. *Plant Methods*, 19(1):1–15, 2023.

[54] Y. Zhang and B. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *IJCNLP*, 2017.

[55] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[56] H. Zhu, C. Lin, G. Liu, D. Wang, S. Qin, A. Li, J.-L. Xu, and Y. He. Intelligent agriculture: deep learning in uav-based remote sensing imagery for crop diseases and pests detection. *Frontiers in Plant Science*, 15:1435016, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 claims made in the abstract and introduction accurately reflect the paper's contributions and scope. The paper highlights the positive impact of pre-trained weights, the benefits of progressive unfreezing, and its relevance for edge deployment. The results presented align with the existing body of work, and the claim fits within the broader mission of building scalable, AI-powered solutions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

Answer: [Yes]

Justification: I discuss existing literature in my introduction and related work sections where I highlight limitations of work by other authors.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: I do not include theoretical results in my findings.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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: Code and datasets will be provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: Code and datasets will be submitted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: It is discussed under my methodology section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

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: I do not have such reports unfortunately.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: Compute requirements are stated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [NA]

   Justification: I have not reviewed the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: I do explain the postive impacts in my abstract, introduction and conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **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: No such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: I will cite them clearly under methodology.

    Guidelines: They are cited.

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Code is well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **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? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: They were used for in text citations, generating abstract and parts of the introduction and to summarise results.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.