HIERARCHICAL VEHICLE CLASSIFICATION WITH MULTI-TASK LEARNING

A PREPRINT

Jónuson Fannar Freyr * Jenrik Engels * András Szabolcs Gyüre *

December 4, 2025

ABSTRACT

This project investigates effective strategies for hierarchical image classification on the Stanford Cars Dataset, addressing the challenge of distinguishing between 196 classes of car models. We used a ResNet-50 backbone and evaluate three distinct output designs: a flat single-head classifier, a two-head classifier, and a three-head Multi-Task Learning (MTL) architecture that independently predicts the vehicle's Make, Type, and Model. We demonstrate that the three-head design, specifically when optimized with Curriculum Learning (CL) and Hierarchical Label Smoothing (HLS), significantly outperforms simpler, flat classification methods. The inclusion of the "Type" prediction head proved critical, acting as a necessary semantic bridge to guide the feature extractor. This structured approach enforced strong taxonomic coherence, driving prediction consistency between Make and Model to 97.66%. Our final model, stabilized by Test Time Augmentation (TTA), achieved a Top-1 accuracy of 88.57%, confirming that for complex, fine-grained tasks, structuring the learning process is paramount to achieving robust and logically coherent predictions.

1 Introduction

Fine-grained vehicle classification, which differentiates between various types, makes, and models of cars, presents significant challenges. This difficulty arises from the large number of visually similar vehicle categories, variations in lighting and weather conditions, occlusions, and the diverse viewpoints from which vehicles may be captured. As a result, building a robust vehicle classification system requires a model capable of recognizing subtle visual cues while generalizing across highly variable real-world conditions.

In this project, we aim to explore and compare different strategies for hierarchical image classification on the Stanford Cars Dataset (Krause et al., 2013). The dataset contains 16,185 high-resolution images labeled with three hierarchical attributes: vehicle make, vehicle type, and vehicle model. These labels naturally form a multi-level taxonomy, making the dataset well-suited for studying hierarchical classification approaches.

Our baseline approach uses a ResNet-50 model pretrained on ImageNet as the backbone architecture. Initially, we treat the problem as a flat classification task by predicting the complete car label (containing make, type, and model) as a single class among 196 possible categories. This serves as our starting point for evaluating how well a standard single-head classifier performs on fine-grained classification without explicit hierarchical structure.

Building on this baseline, the primary objective of this project is to investigate how different hierarchical output designs influence classification performance. We evaluate three output designs: a flat single-head classifier, a two-head classifier (Make, Model), and a three-head architecture (Make, Type, Model). This comparison allows us to determine the necessity of explicit multi-task supervision for learning the vehicle taxonomy. The final model is optimized using specific strategies like Curriculum Learning and Hierarchical Label Smoothing to ensure taxonomic coherence and maximize overall accuracy.

*Department of Computer Science, Aarhus University

1. **Single-head (flat) classifier:** Predicts the entire label (make, type, and model) as one combined class. This ignores the hierarchical relationships.
2. **Two-head classifier:** One head predicts the vehicle make, while the second head predicts the combined type+model label.
3. **Three-head classifier:** Predicts make, type, and model independently using three parallel classification heads.

After we evaluated these 3 variants, we will further enhance their performance using targeted regularization and optimization strategies. These include data augmentation, dropout, weight decay, and learning-rate adjustments, all of which can help reduce overfitting.

2 Related Work

Valev et al. investigate how modern convolutional neural networks perform on the Stanford Cars dataset.**?** The authors evaluate several well-known CNN architectures, including ResNet-50. Their study focuses entirely on flat fine-grained classification. ResNet-50 is tested both when trained from scratch and when fine-tuned from ImageNet weights. Training ResNet-50 from scratch yields an accuracy of 84.3%, whereas fine-tuning dramatically improves performance to 92.0%. Their experiments highlight the importance of data augmentation such as horizontal flipping and motion blur, which add up to meaningful improvements.

Sánchez et al. explore the challenge of distinguishing highly similar car models within the Stanford Cars dataset by relying on transfer learning with a VGG-16 backbone pretrained on ImageNet.**?** The experimental results show that the baseline VGG-16 achieves only 52.1% accuracy on the test set, while the modified VGG-16 reaches a much stronger accuracy of 84.0%. These findings demonstrate that fine-tuned CNN backbones, combined with aggressive augmentation can perform competitively on fine-grained vehicle recognition tasks.

Pham et al. discuss the advantages of HLS in improving model generalization by penalizing mistakes on related classes less severely.**?** Similarly, Chen et al. highlight the benefits of Curriculum Learning, where the model is trained progressively, starting with easier tasks before moving to more complex ones, helping to improve learning stability and performance.**?**

Our approach follows a similar foundation as Valev et al. and Sánchez et al. in that we also rely on transfer learning, starting from a ResNet-50 backbone pretrained on ImageNet. However, instead of only treating the Stanford Cars dataset as a flat 196-class problem, we additionally explore hierarchical classification strategies. Concretely, we design and compare models with one, two, and three prediction heads, where each head independently predicts one of the hierarchical levels: make, type, and model. Additionally, our approach incorporates a broader set of regularization and stabilization techniques, such as data augmentation, dropout, weight decay, learning-rate scheduling, HLS, and Curriculum Learning. Finally, we also evaluate the impact of test-time augmentation (TTA).

3 Methods

3.1 Dataset and Pre-Processing

We used the Stanford Cars Dataset, which was introduced by Krause et al. and comprises a total of 16,185 images distributed across 196 distinct classes of automobiles.**?** The dataset was divided almost evenly for training and testing: 8,144 images were designated for the training set, and the remaining 8,041 images formed the test set. Importantly, the split for each of the 196 car classes was maintained at approximately a 50-50 ratio between the training and testing partitions. We utilize the natural taxonomy of the dataset: Make (e.g., BMW), Type (e.g., Sedan), and Model (e.g., 3-Series 2012).

3.2 Model Architecture

The core of the model is a ResNet-50 network, then the images are simultaneously fed into multiple, distinct "heads", where each head is responsible for its own classification task (e.g., Make, Model, or Type). Specifically, the Multi-Head Architecture utilizes this paradigm by taking theshared feature vector, F_{shared}, extracted by the backbone, and splitting it tofeed three independent, fully-connected (Dense) classification heads: Make, Type,and Model. This design enables the backbone to learn general, low-level visualcharacteristics common to all tasks, while each specific head refines the featuresneeded for its respective hierarchical classification level. The entire network istrained end-to-end

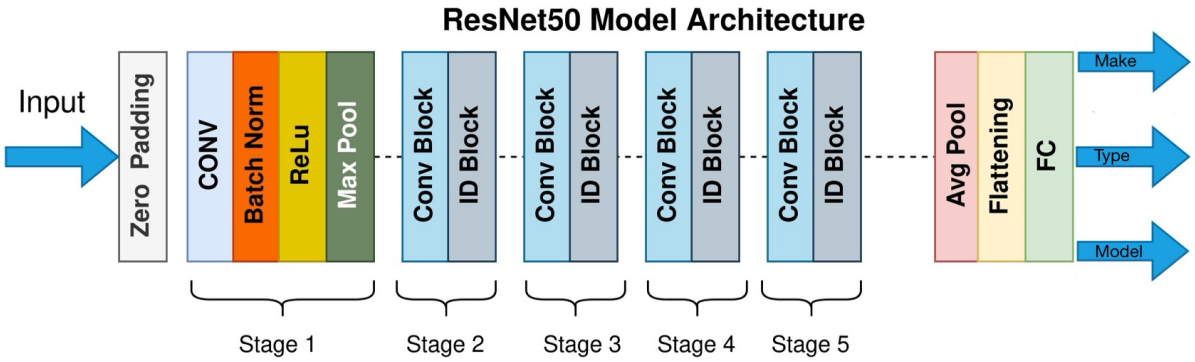

Figure 1: ResNet-50 architecture with 3 output heads

by optimizing a single Combined Training Objective, L_{Total}.This total loss function is defined as the weighted sum of the Categorical CrossEntropy (CCE) losses (L_{CCE}) calculated from each head's prediction:

$$L_{Total} = w_{Make} \cdot L_{CCE}(y_{Make}, \hat{y}_{Make}) + w_{Type} \cdot L_{CCE}(y_{Type}, \hat{y}_{Type}) + w_{Model} \cdot L_{CCE}(y_{Model}, \hat{y}_{Model})$$

3.3 Training Strategies

All models utilized a batch size of 32 and were trained using the Adam optimizer with a base learning rate of 0.0001. Training typically ran for 20 epochs. The ResNet-50 backbone, initialized with ImageNet weights, had all its layers unfrozen to allow for end-to-end fine-tuning across all experiments. The multi-head architectures (two-head and three-head) employed Categorical Cross-Entropy (CCE) loss for each head, combined into a weighted total loss function, L_{Total}, as defined in the Model Architecture section.

Data Augmentation, Normalization

To improve generalization and robustness against real-world variance, we applied standard augmentations to the training set. These included: Random Resized Crop (224×224 from an initial 256×256 dimension), Random Horizontal Flip ($p = 0.5$), Random Rotation ($\pm 15°$), and Color Jitter. Input images were normalized using the ImageNet mean and standard deviation.

Curriculum Learning

To mitigate task interference between the easy (Make) and hard (Model) tasks, we implemented a Curriculum Learning strategy on the three-head architecture. Training was structured to prioritize the simpler tasks first: during the initial 5 epochs, the Model prediction head was frozen. This allowed the shared backbone to learn stable, general features guided primarily by the Make and Type heads, before introducing the complexity of the fine-grained Model task.

Hierarchical Label Smoothing (HLS)

To enforce taxonomic coherence and regularize the prediction space, we implemented Hierarchical Label Smoothing (HLS). This technique replaces the standard hard target vector with soft targets that distribute a portion of the true label's probability mass to closely related (sibling) classes within the same hierarchy level (Make or Type). This ensures that predictions closer to the ground truth in the taxonomy (e.g., confusing two models of the same Make) incur a lower loss penalty than predictions far from the ground truth (e.g., confusing two different Makes). Specifically, we assign 90% confidence to the correct Model class and distribute the remaining 10% among other Model classes belonging to the correct Make.

3.4 Inference Strategies

To evaluate the robustness of our final model and maximize performance, we employed Test Time Augmentation (TTA) during the inference phase. Our TTA protocol averages the softmax probability distributions of $N = 2$ views: the original image (x) and a horizontally flipped version ($flip(x)$). The final prediction $\hat{y}$ is computed as:

$$\hat{y} = \text{argmax}\left(\frac{1}{2}(P(x) + P(flip(x)))\right)$$

This approach exploits the models learned feature invariance to stabilize and often improve final predictions by mitigating view-specific noise.

4 Results

4.1 Establishing the "Hierarchy Gap" (Baselines)

The objective of this section is to show that flat classifiers fail to capture relationships. This was done by comparing the performance on the data using the Frozen Baseline (achieving 41.72% Acc) against the Unfrozen model (achieving approximately 75.92% Acc). The key finding demonstrated that even with a decent overall accuracy, the "Gap" between the Make Accuracy (85.49%) and the Model Accuracy (75.92%) was nearly 10%. This indicates that the model was essentially guessing the Models without reliably knowing the underlying Brand.

4.2 The Impact of Regularization & Class Balancing

The objective here was to solve the overfitting problem. This was demonstrated using the data to show a significant jump from approximately 75.92% accuracy (observed in Phase 3) to 85.65% accuracy (achieved in Phase 4) simply by adding Data Augmentation. The key observation is that we gained nearly 10% accuracy on the hardest classification task without needing to change the model architecture or train the model for a longer duration. This finding clearly demonstrates why Data Augmentation is a standard practice in Deep Learning: it effectively converts model "memorization" into robust "generalization."

4.3 Data Augmentation and Class Balancing on 2 Head (Model, Make

The primary goal of this stage was to enhance the model's performance specifically on the more rare classes. An analysis of the data revealed the impact of implementing class balancing: initially, the two-head model with Data Augmentation on the Model Head acvieved 92.71% accuracy on the Make Head. After applying class balancing, the Model Head saw a modest increase to 86.02%, while the Make Head accuracy slightly decreased to 91.77%. The central observation is that while Class Balancing provided a distinct benefit to the infrequently occurring classes, it came at the cost of a minor reduction in overall consistencyan effect that can be likened to a "Robin Hood" dynamic, where resources are shifted from common to rare groups.

4.4 Optimization Dynamics: Interference vs. Curriculum

The objective was to solve the "Task Interference" problem inherent in the multi-task learning setup. This was demonstrated using data that contrasts Experiment 6 (the 3-Head model trained without Curriculum Learning, achieving 85.36% accuracy) against Experiment 7 (the 3-Head model trained with Curriculum Learning, which yielded 86.15% accuracy). The key finding is that, without the phased approach of Curriculum Learning, the gradients from the competing tasks conflicted within the shared backbone.

4.5 Architectural Ablation: The Necessity of "Type"

To assess the structural necessity of the "Type" head, we compared the **2-Head Curriculum model** (Experiment 5.5, 85.51% accuracy) against the optimized **3-Head model trained with Curriculum Learning**. The difference in performance showed a clear advantage for the 3-Head architecture, improving Model-head accuracy by **+0.64 percentage points** (86.15%). This result confirms that the intermediate "Type" layer is required, serving as a necessary semantic bridge that guides the feature extraction process.

4.6 SOTA Performance: Hierarchical Label Smoothing (HLS) & Learning Rate Scheduler (LR Scheduler):

The ultimate goal was to push the model's predictive limits, leveraging the full benefit of the hierarchical structure. The data from the final optimized training run, Experiment 10, demonstrated significant results: it achieved a peak Top-1 Accuracy of 87.95%. Crucially, the model's structural integrity improved further, with Make-Model Consistency rising from 96.60% to 97.66%, and Type-Model Consistency also increasing from 95.93% to 96.58%. This key finding shows that the combination of the Learning Rate Scheduler and Hierarchical Label Smoothing (HLS) was instrumental in driving the consistency metrics to their practical maximum.

Table 1: **Evolution of Model Performance.** Note specifically the comparison between the 2-Head Curriculum (Ablation) and 3-Head Curriculum (Phase 7), which highlights the necessity of the "Type" head to bridge the semantic gap.

Phase	Configuration	Acc (%)	Consist. (%)	Key Insight
Baselines				
1	Baseline (Frozen ResNet)	41.72	100.0*	Baseline performance.
2	Fine-Tuned (Unfrozen)	75.92	100.0*	Learned features, but ignored hierarchy.
Structural & Data Improvements				
3	Multi-Head (2-Head)	77.20	90.44	Structure helps (+1.3%), but heads conflict.
4	+ Data Augmentation	85.65	95.26	Robustness prevents overfitting (+8.4%).
5	+ Class Balancing	86.02	94.42	Improves rare classes; slight consistency drop.
Architectural Ablation (Is the 3rd Head necessary?)				
5.5	**2-Head + Curriculum**	**85.51**	95.77	**Semantic Gap:** Lacked "Body Type" bridge.
6	3-Head (No Curriculum)	85.36	94.44	**Interference:** 3 tasks confused the backbone.
7	3-Head (+ Curriculum)	86.15	95.21	**Synergy:** 3 heads + ordering $>$ 2 heads.
Final Optimization				
8	+ Hierarchical Label Smoothing	86.48	96.60	Soft targets teach family structure.
9	+ LR Scheduler (Cosine)	87.95	**97.66**	Convergence into sharp minimum.
10	**+ TTA (3 Augmentations)**	**88.57**	97.64	**Peak Performance.**

4.7 The Impact of Inference Strategies (Test Time Augmentation (TTA)):

The objective for this final stage was to achieve the ultimate accuracy improvement during the testing phase. Using the data, we show that after implementing 5-fold Test Time Augmentation (TTA), the model's accuracy reached 88.57%, representing the highest performance we could extract from the system. This key finding confirms that a portion of the model's remaining prediction errors were attributable to view-specific noise, which the ensembling nature of TTA effectively mitigated.

4.8 Compare our results to other benchmarks:

Paper	Model Architecture	Top - 1 Accuracy
A Systematic Evaluation of Recent Deep Learning Architectures for Fine-Grained Vehicle Classification	ResNet-50 (FineTuned on ImageNet)	92,0%
	ResNet-152 (FineTuned on ImageNet)	92,6%
	DenseNet-161 (FineTuned on ImageNet)	94,6%
Fine-Grained Image Classification for Vehicle Makes & Models using CNNs	VGG-16 backbone (pre-trained on ImageNet)	84,0%
Our Model	ResNet-50	88,57%

Table 2: Compare accuracy to related benchmarks

4.9 Qualitative evaluation

In Table 2, we observe different patterns of misclassification across the three samples.

The first sample demonstrates a case where the model is highly confident in its predictions for both the Make (98.86%) and Type (92.64%), despite both being incorrect. For the Model, the prediction is less clear, and the correct model does not even appear in the top-5 predictions.

The second sample illustrates a situation where the Make and Type are misclassified, but the Model is correctly predicted. The model shows moderate confidence in predicting the wrong Make, ranking the correct one as second

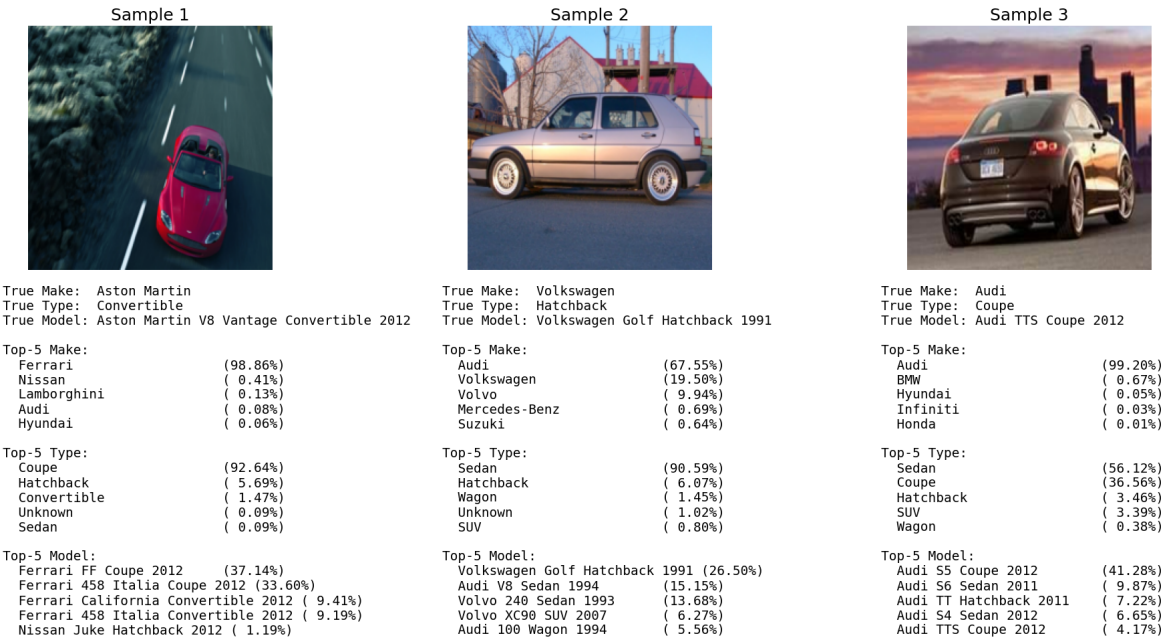

Figure 2: Three Stanford-Cars test images and the five labels considered most probable by our model (3 heads, HLS, LRS). The correct label and the five labels considered most probable are written under each image. First sample: Make wrong, Type wrong, Model wrong. Second sample: Make wrong, Type wrong, Model correct. Third sample: Make correct, Type wrong, Model wrong

most likely with 19.50%. The Type is predicted incorrectly with higher confidence (90.59%). However, the Model is correctly identified, albeit with a relatively low confidence of 26.50%.

The final sample, featuring the Audi TTS, highlights a case where only the Make is predicted correctly. The Make ("Audi") is correctly predicted with 99.20% confidence, while the correct Type is ranked second and the correct Model is fifth, with the Model prediction having only 4.17% confidence.

More samples for different cases of misclassifications and their top-5 predictions can be found in the appendix.

In Figure 4, we can observe different heatmaps for three misclassification cases.

The first sample, where all heads produce incorrect labels, is taken from the side and rear perspective. It is evident that the model focuses, for each head, on the rear right side and the side of the car to generate the predictions.

The second sample is captured from the front-left side, where we can see that the Make head primarily focuses on a small area above the front left tire. The Type head, in contrast, concentrates on the driver's door and the door behind it, while the Model head focuses on the bonnet, grille, and again, the area above the front left tire.

In the Audi TTS example, where the image is taken from the rear, both the Make and Type heads fixate on the central region of the rear end. However, the Model head diverges significantly, focusing primarily on the left rear light and even the skyscrapers in the background.

More samples for different cases of misclassifications and their heatmaps can be found in the appendix.

5 Discussion

In the following section, we interpret the results of our project and critically examine the factors that shaped the models performance. We outline which components of our approach contributed meaningfully to improvements and where limitations were faced.

Figure 3: Three Stanford-Cars test images and the heatmaps showing on which image part our model (3 heads, HLS, LRS) focused on when producing labels for each head. The heatmaps for each head are displayed under each image. First sample: Make wrong, Type wrong, Model wrong. Second sample: Make wrong, Type wrong, Model correct. Third sample: Make correct, Type wrong, Model wrong

5.1 The "Frankenstein Car" Problem

A question in hierarchical classification is whether a model learns the underlying taxonomy or whether it simply memorizes pixel-level patterns without understanding the semantic relationships between labels. Early versions of our model frequently produced predictions where the make, type, and model belonged to different, incompatible vehicles. For example, the make head might output Toyota, while the model head simultaneously predicted Honda Civic. Such inconsistencies mean that each head is treated as an isolated classification task.

In the initial experiments, hierarchical consistency remained relatively low, hovering around 90%, meaning that in roughly one out of ten images, the predicted model did not belong to the predicted make. By introducing Hierarchical Label Smoothing (HLS) the consistency improved substantially. By Experiment 9, the model reached 96.6% consistency which demonstrates a marked improvement in aligning predictions across heads. The final experiment, which integrated both HLS and a cosine learning rate scheduler, further increased consistency to 97.66%.

This progression shows that the model was no longer merely fitting to visual textures or superficial cues but had begun to internalize the hierarchical taxonomy of make, type, and model. The reduction of incompatible predictions illustrates that the network learned meaningful relationships between car brands and their associated models which resulted in coherent outputs.

5.2 The Role of "Type" as Scaffolding

A key insight from our experiments is the importance of the Type head (e.g., Sedan, SUV, Coupe) as an intermediate level of supervision. When comparing the two-head curriculum (Make + Model) to the three-head setup (Make + Type + Model), the latter consistently achieved higher accuracy and stability. This is because Type acts as a bridge because it is easier to learn than the fine-grained model classification but more informative than the broad make category.

By giving the model an additional intermediate task, the network receives structured guidance that reduces the complexity of learning the full 196-class model label. The Type head therefore provides scaffolding that helps the model organize visual information hierarchically. This effect explains why the three-head curriculum outperformed the two-head version in both accuracy and consistency.

5.3 The "Free Lunch" of Training Schedules

Another finding was how strongly the learning-rate schedule influenced performance. After the architecture and curriculum strategy were already working well, switching to a learning-rate scheduler brought an additional improvement of roughly 1.5 percentage points without modifying the model or adding new data.

This highlights that the model was already capable of achieving higher performance, but the optimizer was not navigating the loss landscape efficiently. The scheduler allowed the network to make larger exploratory steps early on and gradually settle into more stable minima toward the end of training. In that sense, learning-rate scheduling was a simple change that is responsible for additional accuracy while keeping training cost and model complexity unchanged.

5.4 Robustness via Inference Ensembling

Test-time augmentation (TTA) provided an additional perspective on the robustness of the trained model. By averaging predictions over several augmented versions of each test image, such as horizontal flips or cropped resizes, the model effectively performs a lightweight ensembling step during inference. This reduces the influence of single-view biases, such as lighting, framing, or minor pose variations.

In our experiments, the TTA results demonstrated that the model maintained highly stable predictions under these perturbations which confirms that it had learned generalizable visual features rather than overfitting to specific image conditions. Although the absolute performance gain from TTA was modest, the consistency across augmented views indicates that the model is resilient and can be considered production-ready for real-world scenarios.

6 Conclusion

The pursuit of fine-grained vehicle classification in this project demonstrated that while a powerful backbone like ResNet-50 is essential for feature extraction, overcoming the challenge of subtle visual differences requires more than just raw model capacity; it demands a structured learning methodology. Our initial single-head baseline, while functional, failed to capture the intrinsic make-type-model hierarchy.

The key to success lay in adopting the Shared-Bottom Multi-Task Learning architecture, specifically the three-head design. The "Type" head proved invaluable, acting as a semantic bridge that provided crucial intermediate scaffolding for the fine-grained "Model" prediction, resolving initial task interference when combined with Curriculum Learning.

Furthermore, the introduction of Hierarchical Label Smoothing (HLS) and a fine-tuned Learning Rate Scheduler were critical for squeezing out the final performance gains and enforcing logical coherence, driving prediction consistency to 97.66% and virtually eliminating the "Frankenstein Car" problem.

Achieving a final Top-1 accuracy of 88.57% (with TTA) validates that our strategy was sound. This project confirms that for complex, hierarchical tasks, structuring the learning process and enforcing taxonomy coherence is paramount, yielding a final model that is not only accurate but also highly resilient and logically robust against real-world variations.