HIERARCHICAL VEHICLE CLASSIFICATION WITH MULTI-TASK LEARNING

A PREPRINT

Fannar Freyr Jónuson [*]   Jendrik Engels [*]   András Szabolcs Gyüre [*]

December 5, 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. To assess the necessity of explicit multi-task supervision, we evaluate three distinct output architectures: (1) a **Single-head (flat) classifier** that predicts the 196 combined classes directly, ignoring hierarchical relationships; (2) a **Two-head classifier** that separates the coarse vehicle Make from the fine-grained Model prediction; and (3) a **Three-head classifier** that explicitly disentangles the taxonomy by independently predicting Make, Type, and Model via parallel classification heads.

[*]Department of Computer Science, Aarhus University

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. [2018] 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.

Corrales Sánchez et al. [2020] 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. [2021] discuss the advantages of HLS in improving model generalization by penalizing mistakes on related classes less severely. Similarly, Chen et al. [2023] 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 the shared feature vector, F_{shared}, extracted by the backbone, and splitting it to feed three independent, fully-connected (Dense) classification heads: Make, Type,and Model. This design enables the backbone to learn general, low-level visual characteristics common to all tasks, while each specific head refines the features needed for its respective hierarchical classification level. The entire network is trained end-to-end 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

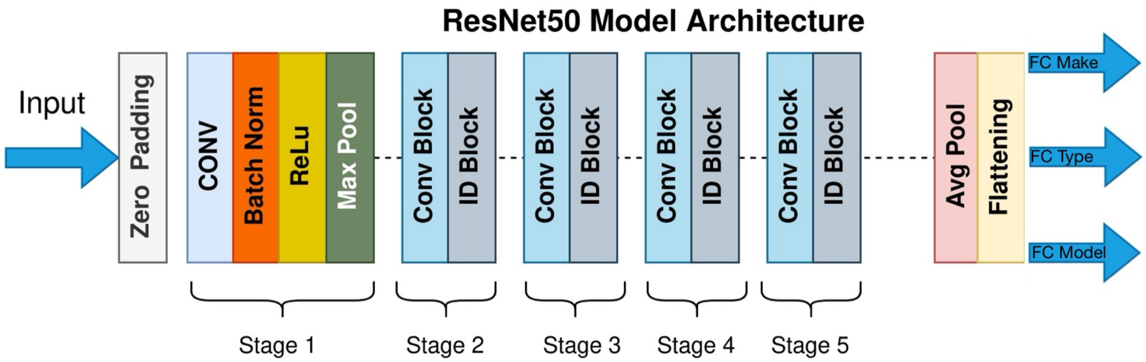

Figure 1: ResNet-50 architecture with 3 output heads

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 $N = 4$ views: the original image (x), a horizontally flipped version (x_{flip}), and two rotated versions ($x_{rot+15°}$ and $x_{rot-15°}$). The final prediction $\hat{y}$ is computed as the average of the probabilities across these augmented views:

$$\hat{y} = \text{argmax}\left(\frac{1}{4}\left(P(x) + P(x_{flip}) + P(x_{rot+15°}) + P(x_{rot-15°})\right)\right)$$

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

4 Results

4.1 Establishing the "Hierarchy Gap" (Baselines)

Our baseline experiments highlight the structural failures of flat classifiers. While fine-tuning improved Top-1 accuracy from 41.72% to 75.92%, it left a 9.56% gap between parent (Make) and child (Model) performance. This suggests the network memorized specific model features without learning the requisite brand hierarchy, essentially guessing the 'Model' while missing the 'Make'.

4.2 The Impact of Regularization & Class Balancing

To address the twin challenges of overfitting and dataset imbalance, we implemented a two-stage regularization strategy. First, addressing overfitting proved to be the single most impactful intervention in our pipeline. The introduction of data augmentation yielded a performance surge from 75.92% to 85.65%. Achieving this 9.73% gain without altering the model architecture or training duration indicates that the baseline models were severely constrained by memorization. By enforcing invariance to geometric transformations, augmentation successfully shifted the learning regime from overfitting to robust generalization.

Subsequently, to address the dataset's long-tail distribution, we implemented class-weighted loss functions. While this raised fine-grained Model accuracy to a new high of 86.02%, it coincided with a slight regression in Make accuracy (92.71% to 91.77%). This trade-off indicates that by reweighting the objective function, the optimizer was forced to reallocate representational capacity from dominant, easy classes to rare, difficult ones. Although this slightly reduced coarse-level scores, it improved the model's ability to generalize to the minority classes that standard training often neglects.

4.3 Optimization Dynamics: Interference vs. Curriculum

Curriculum Learning proved essential for resolving the optimization conflicts inherent in Multi-Task Learning. While training all three heads simultaneously caused a performance regression to 85.36% (Experiment 6), implementing a 'coarse-to-fine' freezing schedule restored accuracy to 86.15% (Experiment 7). This improvement validates the hypothesis that the shared backbone requires a stable initialization on high-level concepts (Make/Type) to avoid gradient interference from the fine-grained Model task.

4.4 Architectural Ablation: The Necessity of "Type"

Architectural ablation confirms the necessity of the intermediate 'Type' head. When trained under identical curriculum protocols, the 3-Head architecture outperformed the 2-Head variant by +0.64 percentage points (86.15% vs. 85.51%). This result validates our hypothesis that the 'Type' label serves as a critical semantic bridge, providing the granular supervision required to link coarse Manufacturer features with fine-grained Model details.

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

Optimization via Hierarchical Label Smoothing and Learning Rate Scheduling drove the model to its training peak of 87.95% Top-1 accuracy. Crucially, this phase achieved near-perfect structural coherence, elevating Make-Model consistency from 96.60% to 97.66% and Type-Model consistency to 96.58%. These record metrics confirm that combining soft taxonomic targets with precise learning rate annealing was instrumental in eliminating the final residual inconsistencies in the prediction logic.

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

Deploying Test Time Augmentation (TTA) during inference yielded our highest recorded performance of 88.57%. This final improvement indicates that the trained model had successfully internalized the semantic hierarchy, with remaining errors largely attributable to non-invariant input noise. The TTA ensemble effectively mitigated these fluctuations, extracting the upper bound of the ResNet-50 backbone's capacity on this dataset.

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 Compare our results to other benchmarks:

Table 2: Comparison with state-of-the-art benchmarks.

Paper	Architecture	Acc (%)
Valev et al. [2018]	ResNet-50 (Flat)	92.0
	ResNet-152	92.6
	DenseNet-161	94.6
Corrales Sánchez et al. [2020]	VGG-16	84.0
Ours	**ResNet-50 (Hierarchical)**	**88.6**

4.8 Qualitative Evaluation

Figures 2 and 3 illustrate the distinct behaviors of the multi-head architecture. Sample 1 (Figure 2, left) represents a "hallucination" failure where the model confidently misclassifies all levels, likely due to the unusual top-down perspective occluding standard frontal features.

In contrast, Sample 3 (Figure 2, right) demonstrates the success of the hierarchical constraints. The Make head identifies "Audi" with near-certainty (99.2%), preventing an egregious cross-brand error. The error is isolated to the Model head, which struggles to differentiate the specific "TTS Coupe" trim from similar Audi frames. The heatmaps (Figure 3) further confirm that the heads have learned orthogonal features: the Make head consistently attends to brand identifiers (logos, grilles), while the Model head focuses on fine-grained details (headlights, bumpers), occasionally drifting to background noise in high-uncertainty scenarios.

Figure 4 illustrates additional cases where the model misclassifies both the make and type, but correctly predicts the model. In these samples, the model still assigns a relatively high probability to the correct label, even though it is not considered the most likely. The true type is also ranked second in all cases, but with a very low probability, ranging from 0.74% to a maximum of 28.08%.

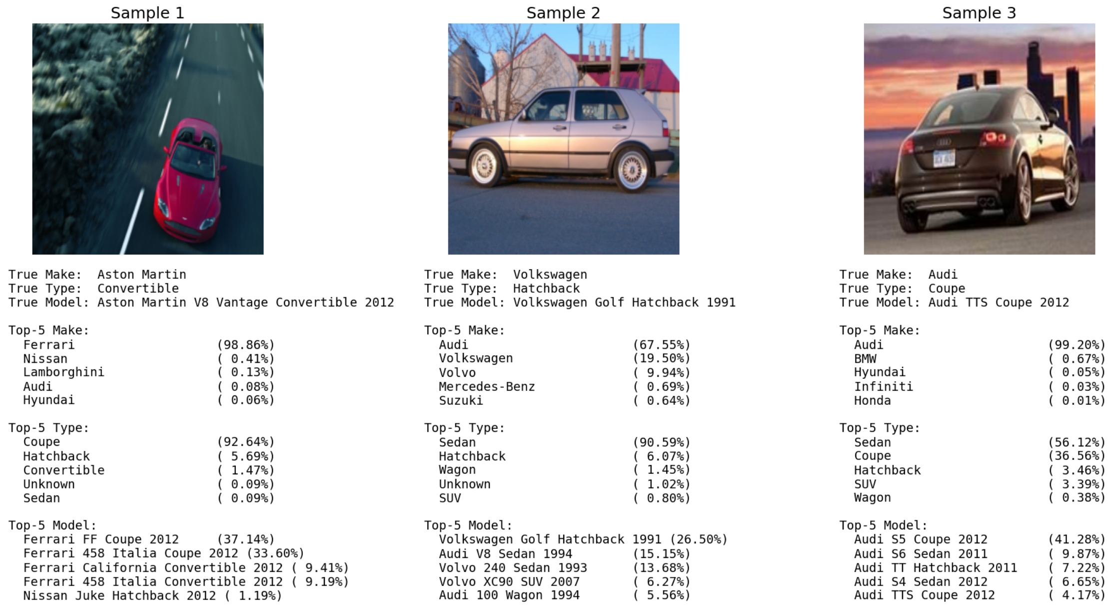

Figure 2: Analysis of error modes. Sample 1 represents a total classification failure due to viewpoint. Sample 3 demonstrates a "near-miss" where the hierarchy is correct (Audi), but the specific trim is ambiguous.

5 Discussion

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

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

Figure 3: Grad-CAM attention maps. Note the distinct focus of the heads: The Make head (top row) attends to global shapes and grilles, while the Model head (bottom row) focuses on fine-grained trim details like lights and wheels.

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

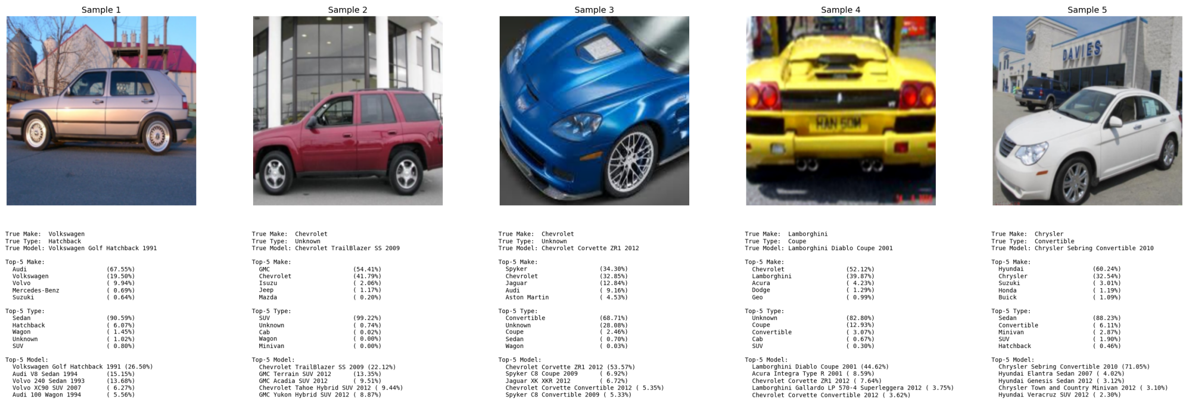

Figure 4: Analysis of error modes. All samples represent the case of predicting the make and type wrong.

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.

References

Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. URL `https://api.semanticscholar.org/CorpusID:16632981`.

Krassimir Valev, Arne Schumann, Lars Sommer, and Jürgen Beyerer. A systematic evaluation of recent deep learning architectures for fine-grained vehicle classification, 2018. URL `https://arxiv.org/abs/1806.02987`.

Héctor Corrales Sánchez, David Fernández-Llorca, S. Vigre, Álvaro Quintanar Pascual, Javier Lorenzo Díaz, and Noelia Hernández. *CNNs for Fine-Grained Car Model Classification*, pages 104–112. 04 2020. ISBN 978-3-030-45095-3. doi:10.1007/978-3-030-45096-0_13.

Hieu H. Pham, Tung T. Le, Dat Q. Tran, Dat T. Ngo, and Ha Q. Nguyen. Interpreting chest x-rays via cnns that exploit hierarchical disease dependencies and uncertainty labels. *Neurocomputing*, 437:186–194, 2021. doi:10.1016/j.neucom.2020.03.127.

Yuzhao Chen, Zonghuan Li, Zhiyuan Hu, and Nuno Vasconcelos. Taxonomic class incremental learning. *arXiv preprint arXiv:2304.05547*, 2023.