Improving Fine-Grained Vehicle Classification via Multitask Learning and Hierarchical Consistency

Gabriel E. Lima*, Eduardo Santos†,*, Eduil Nascimento Jr.†, Rayson Laroca‡,* and David Menotti*
*Department of Informatics, Federal University of Paraná, Curitiba, Brazil
†Department of Technological Development and Quality, Paraná Military Police, Curitiba, Brazil
‡Graduate Program in Informatics, Pontifical Catholic University of Paraná, Curitiba, Brazil
*{gelima,menotti}@inf.ufpr.br †{ed.santos,eduiljunior}@pm.pr.gov.br ‡rayson@ppgia.pucpr.br

***Abstract*—Fine-Grained Vehicle Classification (FGVC) plays a key role in intelligent transportation systems, enabling the recognition of vehicle attributes – such as type, make, and model – from images. Such information supports vehicle identification and can complement automatic license plate recognition by enabling cross-checks and addressing cases with unreadable plates. However, existing approaches often treat these attributes independently, overlooking their hierarchical relationships and differences in task difficulty. This work-in-progress study explores the use of Multitask Learning (MTL) and hierarchical regularization to address these gaps. We evaluate seven deep learning models on a diverse dataset under three training setups: single-task learning, MTL with balanced optimization, and MTL with hierarchical regularization. Results show that MTL consistently improves classification accuracy, while incorporating hierarchical information significantly reduces semantic inconsistencies and enhances confidence calibration. In our best-performing configuration, hierarchy-violating errors dropped from 32.87% (single-task) to 4.10% (MTL with hierarchical regularization). These findings highlight the importance of modeling semantic relationships among attributes in FGVC and suggest promising directions for building more accurate and reliable classifiers. Future work will expand attribute granularity, investigate optimal task combinations, and benchmark against state-of-the-art methods.**

I. Introduction

Fine-Grained Vehicle Classification (FGVC) involves recognizing detailed vehicle attributes from images, such as make, model, and type [1], [2]. This task is central to a wide range of applications, including traffic management, parking systems, and forensic analysis. FGVC can also complement automatic license plate recognition systems, particularly in challenging scenarios where license plates are obscured, illegible, or require cross-verification [3]–[5].

Contemporary FGVC approaches largely rely on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), both of which have achieved high accuracy in distinguishing visually similar vehicles [1], [6]. These approaches typically frame the problem as a single-label classification task, merging multiple vehicle attributes into a single compound label. For instance, class labels may encode the vehicle's make and model [7], or include more granular information such as sub-model and production year [8].

This formulation overlooks the hierarchical and semantic relationships among vehicle attributes, as well as the variation in task difficulty across different levels of granularity. In contrast, prior research in hierarchical fine-grained classification [9]–[11] has shown that Multitask Learning (MTL) approaches that explicitly model these dependencies can improve generalization and lead to more consistent, semantically coherent predictions.

Nevertheless, gaps remain in the current literature. First, existing studies typically explore limited multitask configurations, rarely evaluating combinations involving more than two attributes [10]–[12]. Second, hierarchical modeling is often integrated into specific framework proposals, with little systematic assessment across standard deep learning architectures. Consequently, the impact of jointly learning multiple attributes on model performance remains unclear.

To address these limitations, this work-in-progress study evaluates seven deep learning models in a single-task setting, assessing performance separately for make, model, and type (e.g., bus, car, truck) classification. These models are then adapted to a multitask setting to compare performance when attributes are learned jointly. A hierarchical regularization approach is also applied to enforce consistency between related tasks. The results demonstrate how task relationships influence learning dynamics and affect FGVC results.

This study takes an initial step toward understanding how jointly learning related tasks at different levels of granularity can improve FGVC. It adopts a simplified setup using standard deep learning backbones to evaluate the effects of multi-attribute supervision and hierarchical consistency on representation learning. By avoiding architecture-specific designs, the approach enhances interpretability and reproducibility. The goal is to provide a transparent and practical baseline for studying hierarchical information and to guide future research on more advanced strategies.

The remainder of this paper is structured as follows. Section II reviews related work. Section III outlines the theoretical foundations. Section IV details the experimental setup and Section V discusses the results. Finally, Section VI summarizes the key findings and highlights directions for further research.

II. Related Work

FGVC aims to distinguish between visually similar vehicle categories, such as make and model. Early approaches relied on handcrafted features (i.e., edges, contours, gradients) combined with conventional classifiers [13], [14]. These methods

Accepted at SIBGRAPI 2025 – Workshop of Works in Progress (WiP). DOI: 10.5753/sibgrapi.est.2025.38289.

were limited in scalability and robustness. The introduction of large-scale datasets such as Stanford Cars-196 [8] and CompCars [7] marked a turning point, enabling the adoption of CNNs and shifting the field towards deep learning.

Building on that, studies have explored methods to better capture subtle visual differences between vehicles. Strategies included part-based modeling [15] and coarse-to-fine localization [16] to enhance feature discrimination. Other advances include multi-branch architectures [17], attention-based mechanisms [18], feature refinement modules [19], and customized losses designed for fine-grained classification [20].

While make and model recognition have received significant attention, vehicle type recognition has evolved largely in parallel as a distinct task. Early type classification systems were based on geometric templates, edge detection, and dimensionality reduction methods [21], [22]. The availability of type-focused datasets such as BIT-Vehicle [23] and SYSU-Vehicle [24] enabled the adoption of CNN-based models, which quickly outperformed traditional approaches and became the standard in the field.

Despite the maturity of FGVC, relatively few studies have explored multitask frameworks that jointly predict vehicle attributes. This gap has been partially addressed in the broader field of hierarchical fine-grained classification, which models interdependent labels organized across hierarchy levels. Studies typically leverage MTL to exploit shared representations across coarse and fine-grained labels, using hierarchical feature fusion modules and consistency-enforcing loss functions to improve results on finer-grained tasks [10]–[12], [25].

This work adopts a distinct approach from prior studies by focusing on the joint learning of make, model, and type using standard deep learning backbones, without relying on specialized hierarchical modules. This architecture-agnostic setup enables a reproducible and interpretable analysis of how multi-attribute supervision and hierarchical consistency influence FGVC. As an early step in a broader research agenda, it lays the groundwork for exploring more complex granularity-aware modeling and provides a practical baseline for future developments.

III. Background

This section presents the theoretical foundations underlying the experimental research design, focusing on two key components: the multitask setup with task balancing (Section III-A), and the hierarchical regularization method (Section III-B).

A. Multitask Learning and Gradient Balancing

MTL is a paradigm in which a single model is trained to perform multiple related tasks simultaneously [26]. By leveraging shared representations, it can capture common features across tasks, often leading to better generalization and improved performance compared to training each task independently [27]. This work adopts a standard approach consisting of a shared feature extractor followed by task-specific classification heads.

Formally, let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ denote the set of tasks. The model comprises a shared encoder with parameters θ_s, and individual classification heads h_t with parameters θ_t for each $t \in \mathcal{T}$. For an input sample x, the model computes a shared representation $f(x; \theta_s)$, which is then passed to each head to produce task-specific outputs $h_t(f(x))$. The overall training objective is a weighted sum of task-specific classification losses:

$$\mathcal{L}_{\text{MTL}}^{(k)} = \sum_{t \in \mathcal{T}} w_t^{(k)} \mathcal{L}_t^{(k)} \ , \tag{1}$$

where $\mathcal{L}_t^{(k)}$ is the loss for task t at iteration k, and $w_t^{(k)} \in R_+$ is a weight that controls the contribution of each task's loss.

However, tasks in a multitask setting often differ in difficulty and convergence speed, resulting in imbalanced gradient updates. This can lead the model to prioritize easier tasks while underfitting harder ones. To mitigate this, we adopt GradNorm [28] to dynamically adjust the contribution of each task's loss and ensure that all tasks make balanced progress during training.

At each training step k, GradNorm [28] computes the gradient norm of each task's weighted loss with respect to the shared parameters:

$$G_t^{(k)} = \left\| \nabla_{\theta_s} \left(w_t^{(k)} \mathcal{L}_t^{(k)} \right) \right\|_2 \ . \tag{2}$$

The goal is to align each gradient norm $G_t^{(k)}$ with a target value $\hat{G}_t^{(k)}$ that reflects the relative speed at which the task is learning:

$$\hat{G}_t^{(k)} = \bar{G}^{(k)} \left(\frac{r_t^{(k)}}{\bar{r}^{(k)}} \right)^{\alpha} \ . \tag{3}$$

Here, $r_t^{(k)} = \mathcal{L}_t^{(k)} / \mathcal{L}_t^{(0)}$ is the learning ratio of task t, representing how much its loss has decreased relative to its initial value. The terms $\bar{G}^{(k)}$ and $\bar{r}^{(k)}$ denote the average gradient norm and average learning ratio across all tasks, respectively. The hyperparameter $\alpha > 0$ controls the sensitivity to imbalances: higher values emphasize slower-learning tasks more strongly.

GradNorm [28] minimizes the discrepancy between the actual and target gradient norms by optimizing the objective:

$$\mathcal{L}_{\text{GradNorm}}^{(k)} = \sum_{t \in \mathcal{T}} \left| G_t^{(k)} - \hat{G}_t^{(k)} \right| \ . \tag{4}$$

This loss is minimized only with respect to the task weights $w_t^{(k)}$, which are updated during training.

B. Hierarchical Regularization

Fine-grained classification tasks can involve related attributes organized in a semantic hierarchy, where coarser categories constrain the possible values of finer-grained attributes. To promote consistency across these hierarchically dependent tasks, we adopt a regularization strategy inspired by hierarchical fine-grained classification literature [12], [29]. This regularization encourages predictions at each finer level

to remain consistent with the predictions or ground-truth labels of coarser levels.

Let t_d and t_{d+1} represent two classification tasks in a hierarchical structure, where t_d denotes a coarser level and t_{d+1} a finer level. Each task is formulated as a multi-class classification problem. For each input x, the model outputs predicted probability distributions $\mathbf{p}_x^{(t_d)}$ and $\mathbf{p}_x^{(t_{d+1})}$. Given the ground-truth label $y_x^{(t_d)}$ at level d, a soft target distribution $\mathbf{q}_x^{(t_{d+1}|t_d)}$ is defined by uniformly distributing probability across the set of valid classes in t_{d+1} that are descendants of $y_x^{(t_d)}$. This soft target reflects the permissible fine-level predictions conditioned on the coarse-level label, enforcing semantic consistency across the hierarchy.

For example, in the vehicle classification context, if t_d corresponds to the make (e.g., *Ford*) and t_{d+1} to the model (e.g., *EcoSport*, *Fiesta*), the soft target for model prediction assigns non-zero probability only to models associated with the make "*Ford*", penalizing inconsistent predictions such as "*Corolla*", which belong to a different make.

To implement this regularization, a Kullback–Leibler (KL) divergence penalty between the soft target distribution and the model's prediction at t_{d+1} is computed:

$$\mathcal{L}_{\mathrm{KL}}^{(t_{d+1}|t_d)} = D_{\mathrm{KL}}\left(\mathbf{q}_x^{(t_{d+1}|t_d)} \parallel \mathbf{p}_x^{(t_{d+1})}\right) , \tag{5}$$

where

$$D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p}) = \sum_{j=1}^{C} \mathbf{q}_j \log\left(\frac{\mathbf{q}_j}{\mathbf{p}_j}\right) . \tag{6}$$

Here, $\mathbf{p}_j$ and $\mathbf{q}_j$ denote the predicted and target probabilities for class j at the finer task t_{d+1}, respectively; C is the total number of classes at this level. Eq. (5) loss term can be applied to any pair of hierarchical dependent tasks and seamlessly integrated into the overall training objective as an auxiliary regularization component.

IV. Methodology

This section describes the methodology used to assess the impact of joint learning of vehicle make, model, and type on FGVC. Three setups are evaluated: (*e1*) single-task training; (*e2*) MTL with GradNorm [28]; and (*e3*) MTL with GradNorm and hierarchical regularization. Section IV-A presents the dataset used, and Section IV-B details each experiment.

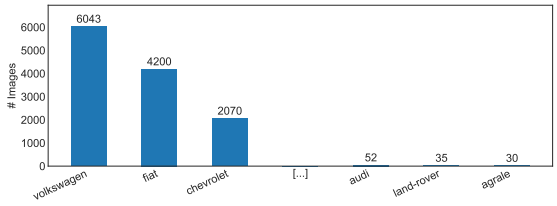
A. Dataset

The dataset used in this study is part of a separate work currently under review and will be released to the public upon publication. It comprises 24,945 images of 16,308 unique vehicles, with annotations for 26 vehicle makes, 136 models, and 14 vehicle types. Images were collected from a real-world surveillance system in a single municipality in Brazil. Images capture diverse operational conditions, including multiple views, partial occlusions, varying lighting environments, and infrared imaging (see Fig. 1).

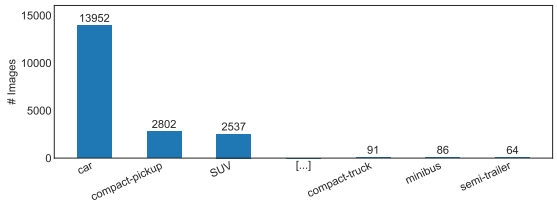
The dataset is highly unbalanced, reflecting real-world vehicle distributions. Fig. 2 illustrates a simplified view of the distribution, highlighting the three most and least frequent

Fig. 1. Images from the explored dataset, showcasing vehicles captured across various viewpoints, environments, lighting conditions, image quality levels, and nighttime infrared scenarios.

vehicle makes and types. To reduce sampling bias, the data were divided into five non-overlapping, stratified folds that maintain class proportions across tasks. To avoid data leakage, all images of a given vehicle (identified by its license plate) were assigned to the same fold [30]. Ten train-validation-test splits were then created using a 3:1:1 ratio, with each fold serving as the test set twice.



(a) Top-3 most and least frequent vehicle makes, representing ≈ 49% of all dataset samples.



(b) Top-3 most and least frequent vehicle types, representing ≈ 77% of all dataset samples.

Fig. 2. Top-3 most and least frequent classes for the vehicle attributes of make (a) and type (b) in the dataset. These distributions highlight the dataset's class imbalance, with a small number of dominant classes accounting for a large portion of the samples.

This dataset was chosen over existing FGVC alternatives for three main reasons: (i) it captures a more realistic and unconstrained scenario, better aligned with real-world conditions; (ii) to our knowledge, no public FGVC dataset includes annotations for vehicle type, make, and model simultaneously; and (iii) its labels were cross-verified with official vehicle registration data, reducing annotation errors.

B. Experimental Setup

Seven deep learning models are evaluated: EfficientNet-V2 Small [31], MobileNet-V3 Small [32], ResNet-50 [33],

ResNet-101 [33], ViT-B16 [34], and two YOLO classification [35] variants (YOLOv11-nano-cls and YOLOv11-small-cls). These models were selected for their relevance in computer vision, widespread academic and industrial use [36], [37], and open-source availability for reproducibility.

All models are initialized with ImageNet pre-trained weights and fine-tuned with all layers trainable. Training runs for up to $1{,}000$ epochs, with early stopping triggered after 60 epochs without validation improvement. Standard models use the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), a weight decay of 5×10^{-4}, an initial learning rate of 10^{-2}, a minimum learning rate of 10^{-8}, and a batch size of 64. The learning rate decays by a factor of 10 if validation loss plateaus for 25 epochs. YOLOv11 models follow the Ultralytics training procedure, using stochastic gradient descent with three warm-up epochs and a cyclic learning rate between 10^{-2} and 10^{-4}.

In experiment (*e1*), models are trained separately for make, model, and type classification using standard cross-entropy loss. For the multitask setups in (*e2*) and (*e3*), each model is extended with one classification head per task, and GradNorm [28] adjusts task weights by monitoring gradient norms at the last convolutional layer of the backbone. An auxiliary Adam optimizer with a fixed learning rate of 10^{-2} is used, with gradient norms clipped at 1 for numerical stability. We set $\alpha = 1.5$ to control task balancing. In (*e3*), hierarchical regularization is introduced using KL-divergence penalties applied to both (type, make) and (make, model) pairs.

All models are trained using the default YOLO image classification augmentation pipeline[1], which includes random resized cropping to 224×224 pixels, horizontal flipping, and RandAugment [39]. The latter applies two randomly selected transformations drawn from a predefined set of geometric and photometric operations at a fixed magnitude. During inference, images are resized with preserved aspect ratio, then center-cropped to 224×224 pixels and normalized.

Classification performance is assessed using two criteria. First, *per-task metrics* — macro accuracy, micro accuracy, and macro F1-score — assess both overall and class-balanced performance for each attribute. Second, the *hierarchical consistency error* measures the proportion of misclassified attribute tuples that violate known hierarchical relationships. An attribute tuple, defined as the set of predicted values for all attributes, is deemed inconsistent if any value conflicts with the hierarchy. All results are reported as averages across the dataset splits.

V. Results and Discussion

Table I reports the classification results for vehicle make, model, and type across the three experimental setups (*e1*, *e2*, and *e3*). Higher values indicate better performance for Micro-Accuracy (Mi-Acc), Macro-Accuracy (Ma-Acc), and macro F1-Score (F1), while lower values are preferred for the Hierarchical Consistency Error (HC-Err). As performance differences may be small, all comparisons are based on pairwise Wilcoxon signed-rank tests [40], [41] ($p < 0.05$) to assess statistical significance.

The analysis begins with a separate evaluation of each experiment. In (*e1*), EfficientNet-V2 Small consistently delivered the highest performance across all tasks, establishing itself as the best single-task model. This trend continued in (*e2*), where it again led in most metrics, although its advantage in vehicle type recognition was not always statistically significant. The only exception occurred in (*e3*), where it was outperformed in type classification by both ResNet-50 and ResNet-101 in terms of macro accuracy and F1-score.

In contrast, the attention-based ViT model performed significantly worse, especially on make and model recognition. Furthermore, its large gap between micro and macro accuracy indicates a strong bias toward majority classes. This lower performance is likely due to the higher data volume requirements of attention-based architectures. Consequently, this model was excluded from experiments (*e2*) and (*e3*) result analysis.

Task-wise comparison revealed that type recognition achieved the highest scores, followed by make and then model classification. This reflects the increasing difficulty as the number of classes grows (14 vehicle types, 26 makes, and 136 models), making finer distinctions more challenging. In single-task settings, however, make and model recognition often showed no statistically significant difference in macro accuracy or F1-score for the same classifier. This may be due to feature overlap between visually similar makes and models [3], which poses similar challenges when tasks are learned independently.

When comparing results across experiments, models trained under the multitask setup (*e2*) consistently outperformed their single-task counterparts from (*e1*). This suggests that jointly learning vehicle attributes can improve generalization in FGVC. Notably, the gains were more pronounced in macro metrics, indicating better handling of underrepresented classes.

Unlike the gains observed in (*e2*), most models in experiment (*e3*) showed lower performance after hierarchical regularization was applied. Nonetheless, all models saw reduced hierarchical consistency error (HC-Err) in (*e3*). For instance, EfficientNet-V2's HC-Err dropped from 32.87% in the single-task setup to 14.97% in the multitask setup, and further to just 4.10% when trained with hierarchical regularization. These results show that enforcing attribute hierarchy leads to more coherent predictions of related attributes.

Hierarchical regularization also influenced prediction confidence. In the model recognition task using EfficientNet-V2 Small, single-task and multitask models exhibited overconfident predictions, while the hierarchical variant produced a clearer distinction between correct and incorrect outputs (see Fig. 3). A similar effect was observed in make recognition. In contrast, type recognition showed minimal changes, likely because it was used only as a reference attribute in the regularization process.

Fig. 3 also reveals a shift in the overlap of confidence distributions. While the hierarchical setup showed greater overall overlap between correct and incorrect predictions ($\approx 43\%$)

[1]Augmentation parameters are omitted for brevity. Full details are available in the official Ultralytics documentation [38].

TABLE I

CLASSIFICATION RESULTS (%) FOR ALL MODELS ACROSS THREE EXPERIMENTAL SETUPS: (*e1*) SINGLE-TASK, (*e2*) MULTITASK, AND (*e3*) MULTITASK WITH HIERARCHICAL REGULARIZATION. HIERARCHICAL CONSISTENCY ERROR (HC-ERR) IS ALSO REPORTED, WITH (*e3*) ACHIEVING THE LARGEST REDUCTION. BEST RESULTS ARE IN BOLD; MULTIPLE BOLD ENTRIES WITHIN A COLUMN DENOTE NO STATISTICALLY SIGNIFICANT DIFFERENCE. RESULTS WERE AVERAGED OVER 10 RUNS, WITH STANDARD DEVIATIONS IN PARENTHESES.

(e1) single-task learning — separate models are trained independently for each attribute.

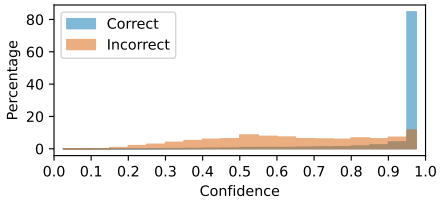
Classification Model	Make			Model			Type			HC-Err ↓
	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	
EfficientNet-V2 Small [31]	**94.43 (0.57)**	**84.99 (1.63)**	**86.36 (1.44)**	**90.91 (0.63)**	**86.16 (1.08)**	**87.25 (0.76)**	**96.12 (0.66)**	**88.97 (1.97)**	**90.23 (1.58)**	32.87 (1.74)
MobileNet-V3 Small [32]	91.27 (0.74)	77.99 (1.68)	80.19 (1.44)	86.52 (0.75)	79.16 (1.18)	80.90 (1.14)	95.15 (0.68)	85.61 (1.93)	87.41 (1.64)	36.84 (1.99)
ResNet-50 [33]	93.62 (0.51)	83.53 (1.73)	84.96 (1.08)	89.89 (0.87)	84.58 (1.21)	85.73 (0.79)	95.45 (0.61)	86.75 (2.34)	88.39 (1.90)	33.74 (1.80)
ResNet-101 [33]	93.80 (0.72)	83.49 (1.53)	85.01 (1.16)	90.20 (0.59)	84.87 (0.78)	86.09 (0.56)	94.69 (1.54)	84.64 (4.47)	85.92 (4.51)	**32.09 (1.25)**
ViT-B16 [34]	31.67 (1.68)	09.09 (3.43)	07.89 (3.04)	29.40 (1.71)	14.29 (3.58)	15.44 (3.51)	71.59 (1.52)	32.52 (5.17)	34.70 (5.09)	63.07 (2.33)
YOLOv11-nano-cls [35]	91.85 (0.99)	80.33 (2.81)	82.22 (2.17)	86.23 (1.38)	79.60 (1.94)	80.41 (1.73)	94.31 (0.80)	85.51 (2.54)	86.50 (2.16)	40.15 (3.99)
YOLOv11-small-cls [35]	93.07 (0.62)	82.68 (1.99)	84.36 (1.55)	87.53 (0.99)	81.33 (1.80)	82.47 (1.38)	95.20 (0.53)	87.00 (1.85)	88.18 (1.35)	38.38 (2.51)

(e2) Multitask learning — a single model jointly predicts all attributes.

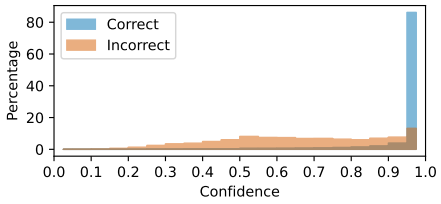
Classification Model	Make			Model			Type			HC-Err ↓
	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	
EfficientNet-V2 Small [31]	**95.85 (0.54)**	**87.61 (1.16)**	**89.30 (1.20)**	**91.35 (0.82)**	**86.89 (1.50)**	**87.83 (1.32)**	**97.01 (0.55)**	**89.87 (2.06)**	**91.45 (1.80)**	**14.97 (1.73)**
MobileNet-V3 Small [32]	92.50 (0.64)	81.03 (1.83)	83.00 (1.53)	87.16 (0.83)	80.10 (1.29)	81.71 (1.04)	95.69 (0.68)	86.61 (1.90)	88.57 (1.69)	17.44 (1.68)
ResNet-50 [33]	95.14 (0.55)	86.49 (1.21)	87.82 (1.35)	90.40 (0.73)	84.72 (1.05)	86.26 (0.71)	96.62 (0.57)	88.61 (2.60)	90.46 (2.28)	16.75 (2.17)
ResNet-101 [33]	95.12 (0.51)	86.71 (1.01)	87.95 (1.10)	90.65 (0.66)	85.43 (0.95)	86.92 (0.79)	96.73 (0.63)	**90.16 (2.63)**	**91.61 (2.20)**	17.21 (1.39)
YOLOv11-nano-cls [35]	92.93 (0.59)	82.00 (1.73)	83.45 (1.44)	87.53 (0.84)	81.53 (1.15)	82.48 (0.75)	95.53 (0.76)	86.56 (2.51)	87.92 (2.22)	20.39 (1.75)
YOLOv11-small-cls [35]	93.72 (0.60)	84.08 (2.15)	85.52 (1.83)	88.47 (0.72)	83.07 (1.24)	83.71 (1.15)	95.86 (0.61)	86.55 (2.09)	88.40 (1.94)	19.18 (2.30)

(e3) Multitask with hierarchical regularization — KL-based penalties are applied to enforce consistency between related attribute predictions.

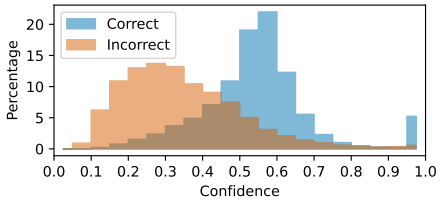
Classification Model	Make			Model			Type			HC-Err ↓
	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	Mi-acc ↑	Ma-acc ↑	F1 ↑	
EfficientNet-V2 Small [31]	**95.87 (0.55)**	**86.70 (2.18)**	**88.47 (1.70)**	**89.96 (1.36)**	**82.50 (1.36)**	**84.84 (1.29)**	**96.19 (0.69)**	84.62 (2.71)	86.95 (2.17)	**4.10 (0.89)**
MobileNet-V3 Small [32]	91.61 (0.84)	76.62 (2.87)	80.13 (2.33)	83.34 (0.71)	70.80 (1.32)	75.60 (1.15)	94.26 (0.72)	82.65 (1.68)	84.96 (1.18)	5.54 (0.68)
ResNet-50 [33]	95.16 (0.64)	84.92 (1.87)	87.30 (1.61)	88.82 (0.70)	79.67 (1.26)	83.60 (0.93)	**96.04 (0.67)**	85.21 (2.41)	**87.73 (2.12)**	**4.30 (0.86)**
ResNet-101 [33]	95.29 (0.70)	84.99 (1.77)	87.64 (1.48)	89.09 (0.80)	80.30 (1.27)	83.85 (0.94)	**96.22 (0.68)**	**85.90 (3.38)**	**88.36 (2.84)**	**4.41 (0.87)**
YOLOv11-nano-cls [35]	93.31 (0.72)	80.38 (2.22)	83.19 (1.92)	85.64 (0.86)	74.42 (1.40)	77.99 (1.21)	94.70 (0.75)	79.42 (2.10)	81.36 (2.00)	6.43 (0.88)
YOLOv11-small-cls [35]	94.52 (0.57)	83.76 (1.95)	86.16 (1.57)	87.95 (0.89)	78.92 (1.43)	81.84 (1.03)	95.35 (0.71)	79.93 (2.68)	81.72 (2.33)	5.71 (1.00)



(a) Single-task setup.



(b) Multitask setup.



(c) Multitask + hierarchical regularization setup.

Fig. 3. Confidence distributions for correct and incorrect predictions in vehicle model recognition using EfficientNet-V2 Small (the best-performing model). Confidence values are averaged over 10 runs, with distributions normalized within each group (correct/incorrect).

than the single-task and multitask-only setups ($\approx$ 27%), its overlap occurred in the lower confidence range (0.4 to 0.5) rather than the high-confidence range (0.9 to 1.0). This suggests that hierarchical regularization promotes more cautious predictions and can help improve classifier calibration.

VI. CONCLUSIONS

This work-in-progress study explored how Multitask Learning (MTL) and hierarchical regularization can improve Fine-Grained Vehicle Classification (FGVC) across three tasks: make, model and type recognition. We proposed and evaluated three experimental setups to isolate the effects of task interaction and label hierarchy. The findings provide a foundation for future research that leverages label hierarchies and explores more advanced learning strategies in hierarchical fine-grained visual classification.

Results indicated that multitask learning generally improved classification performance across tasks and models. In contrast, hierarchical regularization did not always increase accuracy but consistently enhanced semantic consistency. This was evident in a substantial reduction in hierarchical consistency errors, dropping to 4% with the best-performing model. It also led to more cautious and calibrated confidence distributions. These findings highlight a trade-off between accuracy and structured consistency, with important implications for the reliability and interpretability of FGVC methods.

Future research should focus on: (i) expanding the dataset to include finer-grained attributes such as subtypes, sub-models, and production years; (ii) evaluating the impact of different attribute pairings on classification performance to identify

optimal combinations; (iii) improving the balance between accuracy and hierarchical consistency; and (iv) comparing results with existing hierarchical fine-grained classification methods. Moreover, leveraging hierarchical information to develop general confidence calibration approaches represents a promising avenue for exploration.

ACKNOWLEDGMENTS

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)*, through the *Programa de Excelência Acadêmica (PROEX)* - Finance Code 001, and in part by the *Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)* (# 315409/2023-1). We thank the support of NVIDIA Corporation with the donation of the Quadro RTX 8000 GPU used for this research.

REFERENCES

[1] S. H. Tan, J. H. Chuah, C.-O. Chow, and J. Kanesan, "Cross-granularity network for vehicle make and model recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, pp. 5782–5791, 2025.

[2] S. Wolf, D. Loran, and J. Beyerer, "Knowledge-distillation-based label smoothing for fine-grained open-set vehicle recognition," in *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, 2024, pp. 330–340.

[3] I. O. Oliveira *et al.*, "Vehicle-Rear: A new dataset to explore feature fusion for vehicle identification using convolutional neural networks," *IEEE Access*, vol. 9, pp. 101 065–101 077, 2021.

[4] V. Nascimento *et al.*, "Toward advancing license plate super-resolution in real-world scenarios: A dataset and benchmark," *Journal of the Brazilian Computer Society*, vol. 1, no. 31, pp. 435–449, 2025.

[5] L. Wojcik, G. E. Lima, V. Nascimento, E. Nascimento Jr., R. Laroca, and D. Menotti, "LPLC: A dataset for license plate legibility classification," *Conference on Graphics, Patterns and Images)*, pp. 1–6, 2025.

[6] D. Liu, "Progressive multi-task anti-noise learning and distilling frameworks for fine-grained vehicle recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, pp. 10 667–10 678, 2024.

[7] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3973–3981.

[8] J. Krause, J. Deng, M. Stark, and L. Fei-Fei, "Collecting a large-scale dataset of fine-grained cars," https://ai.stanford.edu/ jkrause/papers/fgvc13.pdf, 2013.

[9] C. Jin, L. Luo, H. Lin, J. Hou, and H. Chen, "Hmil: Hierarchical multi-instance learning for fine-grained whole slide image classification," *IEEE Transactions on Medical Imaging*, vol. 44, no. 4, pp. 1796–1808, 2025.

[10] R. Wang, C. Zou, W. Zhang, Z. Zhu, and L. Jing, "Consistency-aware feature learning for hierarchical fine-grained visual classification," in *ACM International Conference on Multimedia*, 2023, pp. 2326–2334.

[11] J. Zhao, Y. Peng, and X. He, "Attribute hierarchy based multi-task learning for fine-grained image classification," *Neurocomputing*, vol. 395, pp. 150–159, 2020.

[12] T. Chen, W. Wu, Y. Gao, L. Dong, X. Luo, and L. Lin, "Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding," in *ACM International Conference on Multimedia*, 2018, pp. 2023–2031.

[13] X. Clady, P. Negri, M. Milgram, and R. Poulenard, "Multi-class vehicle type recognition system," in *Artificial Neural Networks in Pattern Recognition*, L. Prevost, S. Marinai, and F. Schwenker, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 228–239.

[14] V. Petrovic and T. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *British Machine Vision Conference (BMVC)*, vol. 2, 2004.

[15] M. Biglari, A. Soleimani, and H. Hassanpour, "Part-based recognition of vehicle make and model," *IET Image Processing*, vol. 11, pp. 483–491, 2017.

[16] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1782–1792, 2017.

[17] H. Wang, J. Peng, Y. Zhao, and X. Fu, "Multi-path deep cnns for fine-grained car recognition," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10 484–10 493, 2020.

[18] Y. Yu *et al.*, "Cam: A fine-grained vehicle model recognition method based on visual attention model," *Image and Vision Computing*, vol. 104, p. 104027, 2020.

[19] L. Lu, Y. Cai, H. Huang, and P. Wang, "An efficient fine-grained vehicle recognition method based on part-level feature optimization," *Neurocomputing*, vol. 536, pp. 40–49, 2023.

[20] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4204–4212, 2019.

[21] M.-P. Jolly, S. Lakshmanan, and A. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 293–308, 1996.

[22] X. Ma and W. Grimson, "Edge-based rich representation for vehicle classification," in *IEEE International Conference on Computer Vision (ICCV)*, 2005, pp. 1185–1192.

[23] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[24] B. Hu, J.-H. Lai, and C.-C. Guo, "Location-aware fine-grained vehicle type recognition using multi-task deep networks," *Neurocomputing*, vol. 243, pp. 60–68, 2017.

[25] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 574–589.

[26] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] G. R. Gonçalves *et al.*, "Multi-task learning for low-resolution license plate recognition," in *Iberoamerican Congress on Pattern Recognition (CIARP)*, Oct 2019, pp. 251–261.

[28] Z. Chen *et al.*, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International Conference on Machine Learning (ICML)*, 2018, pp. 794–803.

[29] L. Zhang, S. Huang, W. Liu, and D. Tao, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8330–8339.

[30] R. Laroca *et al.*, "Do we train on test data? The impact of near-duplicates on license plate recognition," in *International Joint Conference on Neural Networks (IJCNN)*, June 2023, pp. 1–8.

[31] M. Tan and Q. Le, "EfficientNetV2: Smaller models and faster training," in *International Conf. on Machine Learning*, 2021, pp. 10 096–10 106.

[32] A. Howard *et al.*, "Searching for MobileNetV3," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[34] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021, pp. 1–22.

[35] Ultralytics, "YOLOv11 Image Classification," https://docs.ultralytics.com/tasks/classify/, 2025, accessed: 2025-08-07.

[36] R. Laroca, M. dos Santos, and D. Menotti, "Improving small drone detection through multi-scale processing and data augmentation," in *International Joint Conference on Neural Networks (IJCNN)*, 2025.

[37] G. E. Lima, R. Laroca, E. Santos, E. Nascimento Jr., and D. Menotti, "Toward enhancing vehicle color recognition in adverse conditions: A dataset and benchmark," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, Sept 2024, pp. 1–6.

[38] Ultralytics, "Data augmentation using ultralytics yolo," https://docs.ultralytics.com/pt/guides/yolo-data-augmentation/, 2025, accessed: 2025-08-07.

[39] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 3008–3017.

[40] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, Dec. 2006.

[41] F. Wilcoxon, *Individual Comparisons by Ranking Methods*. New York, NY: Springer New York, 1992, pp. 196–202. [Online]. Available: https://doi.org/10.1007/978-1-4612-4380-9_16