

---

# A TEMPLATE FOR THE *arxiv* STYLE

---

A PREPRINT

**David S. Hippocampus\***

Department of Computer Science  
Cranberry-Lemon University  
Pittsburgh, PA 15213  
hippo@cs.cranberry-lemon.edu

**Elias D. Striatum**

Department of Electrical Engineering  
Mount-Sheikh University  
Santa Narimana, Levand  
stariate@ee.mount-sheikh.edu

November 30, 2025

## ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Keywords** First keyword · Second keyword · More

## 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. Specifically, we focus on three classification strategies:

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.

---

\*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

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

## 3 Methods

### 3.1 Dataset and Pre-Processing??

- The dataset contains 16185 images of 196 classes of cars. The dataset is split into 8144 training images and 8041 test images. Each class has been split around a 50-50 split.
- Hierarchical Label Definition:

| Hierarchy Level | Description  | Example             | Classification Difficulty                |
|-----------------|--|---------------------|--|
| Head 1: Make    | The manufacturer or brand of the vehicle.<br>This is the coarse-grained class. | BMW                 | Easy<br>(Highly distinct features/logos) |
| Head 2: Type    | The general body style of the vehicle,<br>inferred from the model name.        | Sedan               | Medium<br>(Defined by shape/proportions) |
| Head 3: Model   | The specific model and year of<br>the vehicle (the original 196 classes).      | 3-Series Sedan 2012 | Hard<br>(Subtle visual differences)      |

### 3.2 Model Architecture

- **Backbone:** ResNet-50 (pretrained on ImageNet) without any modifications.
- **Multi-Task Heads:** The Data Architecture is based on the Shared-Bottom Multi-Task Learning paradigm. This approach uses a single Convolutional Neural Network (CNN) backbone to extract a universal, high-dimensional feature representation from the input image, which is then fed into multiple, distinct "heads," each responsible for a single classification task (e.g., Make, Model, or Type).
- **Multi-Head Architecture:** The architecture utilizes a multi-head design, based on the Shared-Bottom Multi-Task Learning paradigm, where the feature vector,  $F_{shared}$ , extracted by the backbone, is split and fed simultaneously into three independent, fully-connected (Dense) classification heads: Make, Type, and Model. This split allows the backbone to learn general, low-level visual characteristics shared by all tasks, while each head refines the features necessary for its specific, hierarchical classification level. The entire network is trained end-to-end by optimizing a single Combined Training Objective,  $L_{Total}$ , which is the weighted sum of the Categorical Cross-Entropy losses ( $L_{CCE}$ ) from each head:

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

### 3.3 Training Strategies

- **Baseline:** First we set up a baseline for training which contained batch size = 32 with 15 epochs and a 0.0001 learning rate, where the ImageNet layers were frozen and only the fully connected layer was unfrozen. Then we unfroze the rest of the model parameters, to help the model recognize the shapes of the elements of the car models. Then we added the Make classification head that only predicts the make of the car (fx. BMW) while the first head was predicting the full label to make sure that the model is learning the hierarchy. Then later we added a third, Type classification head that only predicts the type of the car (fx. Sedan)
- **Data Augmentation, Normalization:** We added 4 different types of data augmentation and normalization
- Random Resized: and Cropped Images were first resized to a larger dimension (256x256) and then a  $224 \times 224$  area was randomly cropped. This combination forces the model to learn features that are robust to variations in position, scale, and perspective (i.e., the car is not always perfectly centered).
- Random Horizontal Flip: Flips the image along the vertical axis with a 50
- Random Rotation: Images were randomly rotated up to  $\pm 15$  degrees. This provides rotational invariance, which is particularly useful for handling slight tilts or oblique angles in real-world images.

- Color Jitter: Randomly alters the brightness, contrast, and saturation of the images to make the model invariant to lighting conditions.
- Normalization: Input images were normalized using the mean and standard deviation of the ImageNet dataset, matching the pre-training conditions of the ResNet backbone.
- **Curriculum Learning:** We addressed curriculum learning with the 3 model head to make the model first learn easy (Make) and medium (Type) tasks before focusing on the hard (Model) task. To achieve it, we increased the number of epochs to 20 and in the 5 epochs, we froze the Model head, so it only trains on the Make and Type heads.
- **Hierarchical Label Smoothing (HLS):** While we were trying to squeeze out the last percentages of accuracy in training, we used HLS, to make the model penalize more the easy misses than the hard misses. Which means that for example mistaking a BMW M3 for a BMW 328i (Sibling error) shouldn't give the same loss as mistaking a BMW M3 for a Ford F-150 (Distant error). The technique we used for it is instead of a hard [0, 1, 0, 0] target, use soft targets that give partial credit to siblings. While the hard target gives 100% weight loss for the correct car, while the soft targets give 90% to the correct car and 10% distributed among other cars in the same Make

### 3.4 Inference Strategies:

To evaluate the robustness of our model, we employ Test Time Augmentation (TTA) during the inference phase. Standard inference utilizes a single center crop of the input image. In contrast, 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:

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

This approach exploits the model's learned invariance to geometric transformations (due to the data augmentation in Section 5.4) to stabilize and often improve final predictions. We assumed that it is going to slightly increase our accuracy during testing.

## 4 Results

Organize results by "Research Question" rather than date.

### 4.1 Establishing the "Hierarchy Gap" (Baselines)

- **Objective:** Show that flat classifiers fail to capture relationships
- **Data:** Compare the Frozen Baseline (42% Acc) vs. Unfrozen (76% Acc).
- **Key Finding:** Even with decent accuracy, the "Gap" between Make Accuracy (85.5%) and Model Accuracy (75.9%) was nearly 10%, indicating the model was guessing Models without knowing the Brand.

### 4.2 The Impact of Regularization & Class Balancing

- **Objective:** Solving the overfitting problem.
- **Data:** Show the jump from 77% (Phase 3) to 86.6% (Phase 4) just by adding Data Augmentation.
- **Observation:** Note that Class Balancing (Phase 5) helped rare classes but slightly hurt overall consistency (The "Robin Hood" effect).

### 4.3 Architectural Ablation: The Necessity of "Type"

- **Objective:** Solving the overfitting problem.
- **Data:** Compare Experiment 8 (2-Head Curriculum: 86.49%) against Experiment 7 (3-Head Curriculum: 86.15% - initial) and Experiment 9/10 (Final 3-Head: 89%).
- **Key Finding:** While 3-Head initially struggled due to interference, once optimized (see next section), it outperformed the 2-Head approach, proving that the "Type" layer acts as a necessary semantic bridge.

#### 4.4 Optimization Dynamics: Interference vs. Curriculum

- **Objective:** Solving the "Task Interference" problem in Multi-Task Learning.
- **Data:** Contrast Experiment 6 (3-Head No Curriculum: 85.36%) vs. Experiment 7 (3-Head With Curriculum: 86.15%).
- **Key Finding:** Without curriculum, gradients conflicted. Freezing the hard head allowed the backbone to learn stable features first.

#### 4.5 SOTA Performance: Label Smoothing & LR Scheduler

- **Objective:** Pushing the limit.
- **Data:** Present the final model (Experiment 10) achieving 89.07% Top-1 Accuracy.
- **Key Finding:** The Scheduler + HLS closed the consistency gap significantly.

#### 4.6 The Impact of Inference Strategies

either talk about the results of tta here or above in SOTA, report the final number (88.57%)

### 5 Discussion

This is the most critical section for grading. Interpret the patterns.

#### 5.1 The "Frankenstein Car" Problem

- Discuss Hierarchical Consistency.
- In early experiments, consistency was low (90%). The model would predict "Toyota" (Make) and "Honda Civic" (Model).
- By Experiment 9 (HLS), consistency reached 96.6%, and finally 97.66% with the scheduler. This proves the model learned the taxonomy, not just pixel patterns.

#### 5.2 The Role of "Type" as Scaffolding

- Analyze why 3\_head\_curriculum eventually beat 2\_head\_curriculum.
- The "Type" head (Sedan, SUV, Coupe) provides Intermediate Scaffolding. It is easier to learn than "Model" but provides more structural information than "Make." It bridges the semantic gap.

#### 5.3 The "Free Lunch" of Training Schedules

- Discuss the final experiment (LR Scheduler).
- ou gained 1.5% accuracy (87.9% → 89.07%) just by changing the learning rate schedule. This indicates the architecture was sound, but the optimizer needed "fine-grained" control to settle into the sharp minima of the loss landscape.

#### 5.4 Robustness via Inference Ensembling

Explain how TTA serves as a proxy for measuring model robustness, argue that this proves our model is robust and "production-ready"

### 6 Conclusion

Conclude that while ResNet50 is powerful, structuring the learning process (Curriculum) and enforcing taxonomy (Multi-head + HLS) creates a model that is not only more accurate but logically robust.

## 7 Headings: first level

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. See Section 7.

### 7.1 Headings: second level

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j | y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \quad (1)$$

#### 7.1.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

**Paragraph** Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

## 8 Examples of citations, figures, tables, references

### 8.1 Citations

Citations use `natbib`. The documentation may be found at

<http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf>

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing [Kour and Saabne, 2014a, Hadash et al., 2018] but other people thought something else [Kour and Saabne, 2014b]. Many people have speculated that if we knew exactly why Kour and Saabne [2014b] thought this...

### 8.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi. See Figure 1. Here is how you add footnotes.<sup>2</sup> Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

---

<sup>2</sup>Sample of the first footnote.



Figure 1: Sample figure caption.

Table 1: Sample table title

| Part     |                 |                        |
|----------|-----------------|------------------------|
| Name     | Description     | Size ( $\mu\text{m}$ ) |
| Dendrite | Input terminal  | $\sim 100$             |
| Axon     | Output terminal | $\sim 10$              |
| Soma     | Cell body       | up to $10^6$           |

### 8.3 Tables

See awesome Table 1.

The documentation for `booktabs` ('Publication quality tables in LaTeX') is available from:

<https://www.ctan.org/pkg/booktabs>

### 8.4 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

## References

- George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014a.
- Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.
- George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014b. doi:10.1109/SOCPAR.2014.7008025.