Image Classification using CNNs on the CompCars

Ehsan Eslami Shafigh, Sepideh Ghaemifar, and Abbas Zal

***Abstract*—Fine-grained car classification is a crucial task in computer vision with applications in intelligent transportation, automated surveillance, and automotive inventory management. Despite recent advances, accurately distinguishing car make and model remains challenging due to high intra-class similarity, occlusions, and varying perspectives. This study proposes a robust deep learning-based pipeline for vehicle classification and verification, leveraging ResNet-50 and Inception v3 architectures. Our approach integrates an optimized preprocessing pipeline, CNN-based classification, part-based ensemble learning, and a Siamese network for verification. Experimental results demonstrate that ResNet-50 excels in model classification with 90.15% accuracy, while Inception v3 achieves superior performance in make classification with 93.35% accuracy. A majority voting strategy applied to part-based classification further enhances accuracy, achieving an 81.51% top-1 accuracy with ResNet-50. For verification, our Siamese network outperforms traditional metric-learning approaches, achieving 88.64% accuracy on the easiest dataset and maintaining robust performance across varying difficulty levels. These findings validate the effectiveness of our ensemble-based classification and verification strategies, demonstrating their applicability in real-world automated vehicle recognition systems. Future work will explore attention-based models to enhance fine-grained feature extraction and improve classification robustness.**

***Index Terms*—ResNet-50 ,Inception v3, Ensemble Learning, Car Classification, Compcars, Contrastive learning**

I. Introduction

The rapid advancement of computer vision and deep learning has significantly enhanced automated image classification capabilities, particularly in applications requiring fine-grained recognition, such as vehicle identification. Accurate classification of car make, model, and individual parts is essential for various real-world applications, including intelligent transportation systems, automated surveillance, and inventory management. Despite recent progress, fine-grained vehicle classification remains a challenging task due to high intra-class similarity and variations in lighting, perspective, and occlusions. This study presents a robust deep learning pipeline for car classification using Convolutional Neural Networks (CNNs), specifically leveraging ResNet-50 and Inception v3 architectures.

The challenge of vehicle classification has been extensively studied in the literature, with approaches ranging from handcrafted feature extraction to modern deep learning-based methods. Traditional classification models often struggle to distinguish between visually similar car models, while existing deep learning solutions, though effective, require extensive labeled data and computational resources. Moreover, while previous studies have explored model-based classification, few have investigated a comprehensive multi-component pipeline incorporating preprocessing, part-based classification, ensemble learning, and verification. Our work addresses these limitations by optimizing preprocessing strategies, leveraging transfer learning, and integrating an ensemble voting mechanism for improved performance.

In this paper, we present a multi-faceted deep learning approach for car classification, encompassing preprocessing, CNN-based classification, majority voting for part classification, and a Siamese network for verification. Our contributions can be summarized as follows:

- **Preprocessing Pipeline:** We standardize input images through a structured pipeline involving resizing, augmentation, and normalization to ensure consistency across different deep learning models.
- **CNN-Based Classification:** We employ ResNet-50 and Inception v3 architectures, fine-tuned with transfer learning, to classify car make and model with high accuracy.
- **Part-Based Classification with Ensemble Learning:** To improve robustness, we train independent classifiers for individual car parts and employ a majority voting scheme to aggregate predictions.
- **Verification Using Siamese Networks:** We implement a Siamese Neural Network with contrastive loss to determine similarity between car images, achieving superior verification accuracy across various difficulty levels.
- **Comprehensive Performance Evaluation:** Our models are rigorously evaluated on multiple datasets, demonstrating their effectiveness in real-world classification and verification tasks.

The structure of this paper is as follows. In Section II, we review the related work on car classification using deep learning. Section III details our proposed framework, including model architectures, preprocessing strategies, and training protocols. Section IV presents extensive experimental results and comparative analyses, and Section V concludes the project with discussions on

potential future work.

II. Related Work

The task of vehicle make and model classification has evolved from handcrafted feature-based approaches, such as SIFT and HOG, to deep learning models leveraging Convolutional Neural Networks (CNNs). Early methods struggled with intra-class similarities and varying imaging conditions, but modern architectures like ResNet and Inception have significantly improved classification accuracy [1]. Multi-task learning strategies have also been employed to enhance recognition performance by jointly predicting make and model, reducing computational complexity while improving accuracy [2]. Transfer learning techniques, where pre-trained models are fine-tuned on vehicle datasets, have further demonstrated effectiveness in fine-grained vehicle classification [3].

For vehicle verification, Siamese Neural Networks have been widely adopted, leveraging metric learning and contrastive loss to improve similarity measurement [4]. Local feature-aware models using attention mechanisms have further enhanced vehicle re-identification by focusing on distinctive regions [5]. These methods have shown strong performance improvements across various datasets. Our work builds upon these foundations by integrating preprocessing pipelines, part-based classification, ensemble learning, and verification mechanisms to enhance real-world applicability.

CompCars extends traditional car datasets by incorporating **diverse viewpoints**, **detailed part-level images**, and **rich attribute annotations**. It comprises **web-nature data**, sourced online, and **surveillance-nature data**, captured in real-world settings. This dual-modality framework enables robust evaluation of vehicle recognition tasks in controlled and unconstrained environments. Yang et al. [6] assess CompCars for **fine-grained car classification**, **car attribute prediction**, and **car verification**. Using **CNN models**, they examine **explicit** (e.g., door number) and **implicit attributes** (e.g., speed), finding **side-view images** most effective for explicit attributes, while implicit ones remain challenging.

For **car verification**, Yang et al. [6] employ a **Joint Bayesian approach**, outperforming **CNN feature embeddings with SVM classifiers**, especially in distinguishing **subtle intra-make variations**. However, **high intra-class similarity** among car models makes verification more complex than face verification. To improve accuracy, they emphasize **part-based verification**, leveraging distinctive features like taillights as **strong discriminative cues**.

III. Preprocessing and pipeline

Our **preprocessing pipelines** standardize and normalize input images to ensure consistency and robustness across all experiments. For the **Inception v3** model, training images are first resized to $(299, 299)$ before undergoing augmentation through **random horizontal flips**, **15-degree rotations**, and **ColorJitter adjustments** with brightness, contrast, saturation, and hue set to **0.2, 0.2, 0.2, and 0.1** respectively. This is followed by a **random resized crop** to $(299, 299)$ with a scale range of **(0.8, 1.0)**, **random affine transformations** with zero rotation and translation values of $(0.1, 0.1)$, and a conversion to **grayscale** with a probability of **0.1**. Finally, images are transformed into **tensors** and **normalized** using a mean of **[0.485, 0.456, 0.406]** and a standard deviation of **[0.229, 0.224, 0.225]**. The validation and test pipelines for **Inception v3** are simplified to only include **resizing** to $(299, 299)$, **tensor conversion**, and the same **normalization**. Similarly, the **ResNet-50** model employs an analogous augmentation strategy with images resized to $(224, 224)$, incorporating the same sequence of **random horizontal flips**, **15-degree rotations**, **ColorJitter adjustments**, **random resized crops** to $(224, 224)$ within a scale range of **(0.8, 1.0)**, **random affine transformations** with translation values of $(0.1, 0.1)$, and **random grayscale conversion** (with $p = 0.1$), followed by **tensor conversion** and **normalization** using identical parameters.

IV. Model Architecture and Learning Framework

For our tasks, we employ two powerful CNN architectures: ResNet-50 and Inception v3. ResNet-50 is a 50-layer network that addresses the degradation issue in deep models through residual learning and shortcut connections, which facilitate efficient training by allowing direct gradient flow and have proven effective for image classification and object detection. In contrast, Inception v3 refines the original Inception design by incorporating factorized convolutions and asymmetric building blocks, enabling multi-scale feature extraction at reduced computational cost. Its use of parallel convolutional layers with varied filter sizes within Inception modules, complemented by auxiliary classifiers that provide extra gradient signals, ensures a balanced trade-off between model depth, width, and computational efficiency, leading to robust performance on image recognition benchmarks.

A. Make and Model Classification Task

For the first two tasks, we employ a deep learning-based approach to classify cars by **make and model** using **ResNet-50** and **InceptionV3**, leveraging **transfer learning and fine-tuning**. Training was optimized using the **Adam optimizer**, **cross-entropy loss**, and a **learning rate scheduler** to dynamically adjust learning rates based on validation performance. Both models were

trained for **more than 50 epochs** with **early stopping** implemented to prevent overfitting. The training process followed a structured pipeline: during the **forward pass**, input images were fed into the models to generate predictions; the **cross-entropy loss** function, defined as

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}), \tag{1}$$

computed the discrepancy between the predicted and actual class probabilities; and during the **backward pass**, gradients were calculated and used to update the model weights via the Adam optimizer. The learning rate scheduler monitored validation performance, reducing the learning rate when necessary to ensure stable convergence, while early stopping halted training if validation accuracy did not improve for **10 consecutive epochs**.

B. Parts Classification Task

Inspired by car enthusiasts' ability to identify car models by examining individual parts, our approach trains an independent model for predicting the car model for each car part using either the ResNet-50 or Inception v3 architectures. We adopted the same training procedure outlined in Section IV-A. Also, both models are initialized with pre-trained weights and fine-tuned for our specific car model classification task. After training, predictions from each part are combined using a majority voting scheme, where the final decision is made based on the class receiving the most votes:

$$\hat{y} = \arg\max_{c\in\mathcal{C}} \sum_{p=1}^{P} \mathbf{1}(y_p = c), \tag{2}$$

where:

- $\mathcal{C}$ is the set of car model classes,
- P is the number of predictions (with $P = 8$ in our experiments which is the number of car parts),
- $\mathbf{1}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.
- y_p represents the predicted car model for the p-th car part.
- $\hat{y}$ is the final predicted car model with the maximum votes.

This ensemble strategy minimizes individual model errors and leverages the complementary strengths of both architectures, ultimately enhancing classification accuracy.

You can see an example of majority voting in Figure 1.

C. Verification Task

In this verification task, we implement a **Siamese Neural Network** to determine whether two input images belong to the same class or not. The Siamese network consists of two identical branches that extract feature representations from the input images using a **shared ResNet-50 backbone**. The feature extraction module leverages a **pre-trained ResNet-50**, where all layers except the final classification layer are retained. This ensures that the network benefits from robust feature representations learned from large-scale datasets. After extracting high-level features, a **fully connected (FC) network** with two layers refines the feature embedding, reducing its dimensionality from **2048 to 128** through intermediate layers of **512 neurons** with ReLU activation. The final output is a **128-dimensional embedding vector** for each input image, which serves as a compact and discriminative representation. The network's **forward pass** computes embeddings for two input images separately before comparing their similarity.

To optimize the model for the verification task, a **contrastive loss function** is used. This loss function computes the **Euclidean distance** between the two generated embeddings, measuring their similarity(See Fig[2]). The loss function is formulated as:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left((1-y_i)d_i^2 + y_i\max(0, m-d_i)^2\right) \tag{3}$$

where L is the contrastive loss, N is the batch size, y_i is the label (0 for similar pairs and 1 for dissimilar pairs), d_i is the Euclidean distance between the two feature embeddings, and m is the margin hyperparameter that ensures dissimilar pairs maintain a minimum separation.

The loss function applies two conditions: (1) If the input pair belongs to the **same class** ($y_i = 0$), the loss minimizes the squared Euclidean distance, encouraging the embeddings to be similar. (2) If the pair belongs to **different classes** ($y_i = 1$), the loss maximizes their distance up to a predefined margin, ensuring that different class samples remain sufficiently distinct. The margin parameter (default **1.0**) enforces a separation between dissimilar pairs, effectively improving the discrimination.

V. Results

A. Make and Model Classification

To evaluate the performance of our trained models in make and model classification, we tested them on unseen data using **cross-entropy loss** and **accuracy** as key evaluation metrics.

To assess **make classification**, we evaluated **ResNet-50** and **InceptionV3** on the test set. **ResNet-50** achieved a **test loss of 0.2946** and an **accuracy of 92.78**, demonstrating strong feature extraction capabilities. **InceptionV3**, despite having a **higher test loss of 0.4988**, achieved a **slightly higher accuracy of 93.35**. This

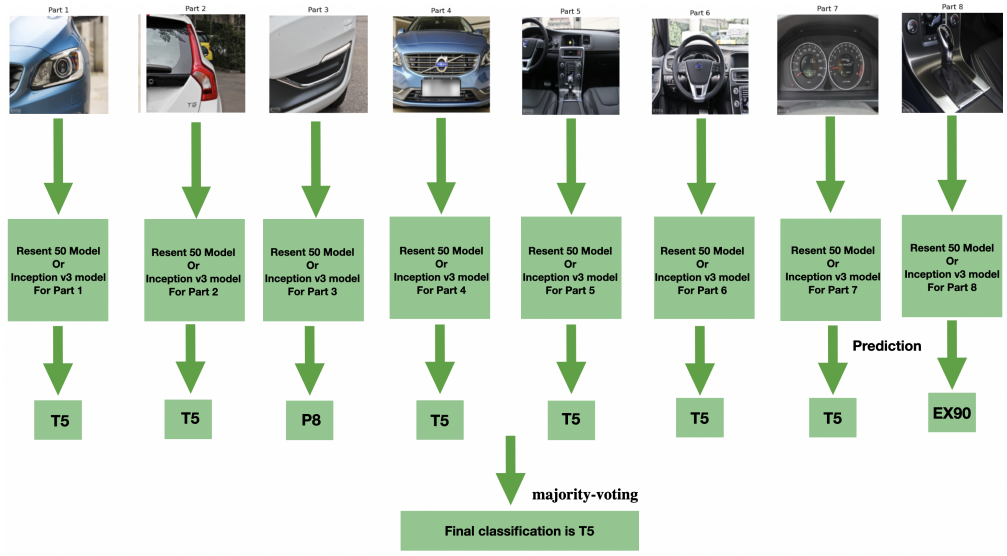

Fig. 1: After training each part with ResNet-50 or Inception v3, we use these models for majority voting. For example, in this figure: Predictions for Parts 1, 2, 4, 5, 6, 7 = T5 ,Prediction for Part 3 = P8 and Prediction for Part 8 = EX90. So, Count: T5 = 6 votes, P8 = 1 vote, EX90 = 1 vote. According to Equation (2), the model with the most votes is T5, so the final classification is **T5**.

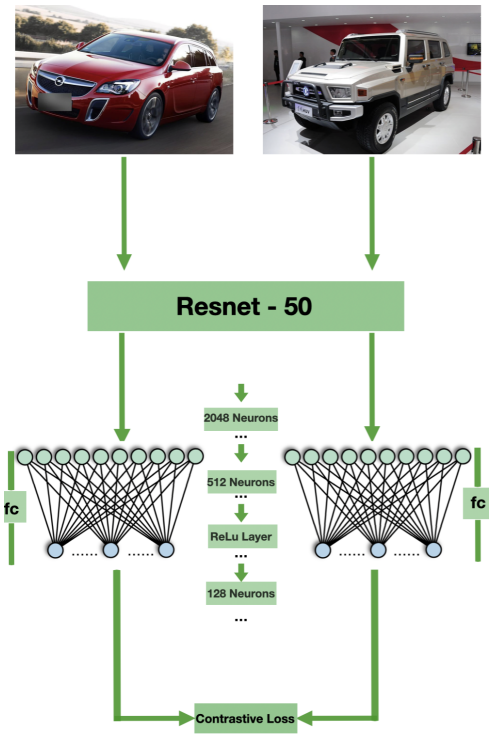

Fig. 2: Architecture of the Siamese Neural Network used for verification. The network consists of two identical branches with a shared ResNet-50 backbone, followed by a fully connected network to reduce the dimensionality of the extracted feature embeddings.

suggests that its **multi-scale feature extraction architecture** was effective in distinguishing between car makes, though with slightly less confident predictions.

For **model classification**, **ResNet-50** achieved better performances, a **test loss of 0.5040** and an **accuracy of 90.15**. **InceptionV3**, with a **higher test loss of 1.0659** and a **slightly lower accuracy of 89.48**, exhibited more uncertainty in predictions[Table 1] .These results highlight **ResNet-50's deep feature extraction advantage** in handling fine-grained differences between car models, whereas **InceptionV3 excelled in make classification**. Both models demonstrated strong potential for **automated vehicle recognition and intelligent transportation systems**.

Classification Task	Model	Test Loss	Test Accuracy
Make Classification	ResNet-50	0.2946	92.78
	InceptionV3	0.4988	93.35
Model Classification	ResNet-50	0.5040	90.15
	InceptionV3	1.0659	89.48

TABLE 1: Test Performance of ResNet-50 and InceptionV3 for Make and Model Classification

B. Car Part Classification

The third task's results are reported in Tables 2 and 3, where "taillight – part 2" demonstrates the best accuracy. Figure 3 displays example images with respect to our preprocessing.

Fig. 3: Example images after our proposed preprocessing steps.

"Taillight" wins among the different car parts, mostly likely due to the relatively more distinctive designs and

	Exterior parts				Interior parts				
	Headlight	**Taillight**	**Fog light**	**Air intake**	**Console**	**Steering wheel**	**Dashboard**	**Gear lever**	**Voting**
Top-1	53.28	64.00	29.19	50.67	54.42	54.72	52.86	38.81	81.51
Top-5	71.53	82.37	48.42	73.40	75.88	78.42	76.65	60.95	96.74

TABLE 2: Classification results for the ResNet 50 on car parts. Top-1 and Top-5 denote the top-1 and top-5 accuracy for car model classification, respectively.

	Exterior parts				Interior parts				
	Headlight	**Taillight**	**Fog light**	**Air intake**	**Console**	**Steering wheel**	**Dashboard**	**Gear lever**	**Voting**
Top-1	43.33	57.00	22.28	26.19	38.72	37.30	27.95	11.93	72.53
Top-5	67.86	78.47	41.60	48.49	65.93	65.86	50.12	24.14	96.77

TABLE 3: Classification results for the Inception v3 on car parts. Top-1 and Top-5 denote the top-1 and top-5 accuracy for car model classification, respectively.

the model name printed close to the taillight, which is a very informative feature for our models. As we mentioned, we also combine predictions from the eight car-part models by a voting strategy. This strategy significantly improves performance due to the complementary nature of different car parts.

C. Make and Model Verification

To evaluate the effectiveness of our Siamese network, we applied the model to two verification tasks: **Car Make Verification** and **Car Model Verification**. Each task was tested on three different datasets of varying difficulty: *Easy*, *Medium*, and *Hard*. These test datasets were introduced by [6]. It is important to note that [6] evaluated models exclusively on the **Car Model Verification** task, and their results do not include Car Make Verification. The classification accuracy achieved by our model on each dataset is summarized in Table 4, while Table 5 provides a comparison with prior results from [6] for the Car Model Verification task.

Verification Task	**Easy**	**Medium**	**Hard**
Car Make Verification	0.8831	0.8742	0.5618
Car Model Verification	0.8864	0.8772	0.8261

TABLE 4: Verification accuracy of our model across different difficulty levels.

Model	**Easy**	**Medium**	**Hard**
Siamese Network (Ours)	0.8864	0.8772	0.8261
CNN + Joint Bayesian [6]	0.833	0.824	0.761
CNN + SVM [6]	0.700	0.690	0.659

TABLE 5: Comparison of Car Model Verification accuracy with prior baseline models from [6].

The results indicate that our Siamese network performs well on both the *Easy* and *Medium* datasets, achieving high accuracy in both tasks. Specifically, in the **Car Make Verification** task, the model attains an accuracy of **88.31%** on the *Easy* dataset and **87.42%** on the *Medium* dataset. However, performance significantly drops on the *Hard* dataset, with an accuracy of only **56.18%**. This suggests that distinguishing car makes becomes increasingly challenging as dataset complexity increases, possibly due to more visually similar brands or challenging lighting and viewpoint variations.

For the **Car Model Verification** task, our Siamese network consistently outperforms the previous models reported in [6]. Our model achieves an accuracy of **88.64%** on the *Easy* dataset, which surpasses the highest baseline result of **83.3%** obtained by the CNN + Joint Bayesian approach. Similarly, on the *Medium* dataset, our model attains **87.72%** accuracy, outperforming the CNN + Joint Bayesian model (**82.4%**) by over 5 percentage points. Notably, on the most challenging *Hard* dataset, our model achieves **82.61%**, significantly higher than the best baseline model's accuracy of **76.1%**. These improvements suggest that our Siamese network is more effective in distinguishing between similar car models, even under difficult conditions.

Furthermore, when compared to the CNN + SVM baseline, which achieved 70.0%, 69.0%, and 65.9% accuracy on the *Easy*, *Medium*, and *Hard* datasets respectively, our model demonstrates substantially better performance across all levels of difficulty. The improvements over both traditional classifiers (CNN + SVM) and metric-learning-based approaches (CNN + Joint Bayesian) highlight the benefits of using a contrastive learning paradigm with a Siamese architecture for fine-grained car model verification.

VI. Concluding Remarks

A. Summary

In this study, we developed a deep learning-based pipeline for fine-grained vehicle classification, leveraging ResNet-50 and Inception v3 architectures, an ensemble-based part classification strategy, and a Siamese network for verification. Our approach demonstrated strong performance in make and model classification, with ResNet-50 excelling in distinguishing car models and Inception v3 performing well in make classification. Additionally, our ensemble voting mechanism significantly improved part-based classification accuracy by leveraging multiple perspectives of a vehicle. Finally, the Siamese network exhibited superior verification capabilities, outperforming traditional metric-learning approaches across different difficulty levels.

B. Key Findings

The key findings of this study are as follows:

- Both ResNet-50 and Inception v3 achieved high accuracy in car make and model classification. ResNet-50 demonstrated superior performance in model classification, while Inception v3 performed slightly better in make classification.
- Car model prediction based on taillight images yielded the highest accuracy among all individual car parts. Overall, ResNet-50 outperformed Inception v3 in model classification in terms of top-1 accuracy. Moreover, the voting strategy significantly improved model prediction performance compared to classification based on individual car parts.
- Car verification using a Siamese network with a ResNet-50 backbone, trained with contrastive loss, achieved higher accuracy than methods based on Joint Bayesian and Support Vector Machines (SVM).

C. Relevance and Applicability

Our findings highlight the practical applicability of deep learning in automated vehicle recognition. This work has potential applications in intelligent transportation systems, automated surveillance, and automotive inventory management. The integration of ensemble learning and verification networks enhances classification robustness, mitigating errors introduced by individual model uncertainties and improving overall system reliability.

D. Limitations

Despite its effectiveness, our proposed model has certain limitations:

- Like all deep learning models, our approach requires large amounts of labeled data to achieve optimal performance.
- The computational cost of training and fine-tuning ResNet-50 and Inception v3 is relatively high, which may limit deployment in resource-constrained environments.

E. Lessons Learned and Challenges

One of the main challenges encountered in this study was preventing model overfitting during training. To mitigate this issue, we applied extensive data augmentation techniques to help the model learn the most relevant features while avoiding overfitting to noise. Balancing model complexity, regularization techniques, and data preprocessing played a crucial role in improving the model's generalization capability.

F. Future Work

As a potential improvement, future research could explore attention-based models to enhance feature extraction by capturing both local (e.g., car logos on wheel rims and car hoods) and global features in car images. This approach could further improve classification and verification performance, making the system more robust in real-world applications.

References

[1] B. Satar and A. E. Dirik, *Deep Learning Based Vehicle Make-Model Classification*, p. 544â€“553. Springer International Publishing, 2018.

[2] D. Avianto, A. Harjoko, and Afiahayati, "Cnn-based classification for highly similar vehicle model using multi-task learning," *Journal of Imaging*, vol. 8, no. 11, p. 293, 2022. PMID: 36354866; PMCID: PMC9697843.

[3] K. Ka, "Cars classification using deep learning," 2017.

[4] Q. Zhang, M. Pei, M. Chen, and Y. Jia, "Vehicle verification based on deep siamese network with similarity metric," in *Advances in Multimedia Information Processing – PCM 2017* (B. Zeng, Q. Huang, A. El Saddik, H. Li, S. Jiang, and X. Fan, eds.), (Cham), pp. 773–782, Springer International Publishing, 2018.

[5] H. Wang, S. Sun, L. Zhou, L. Guo, X. Min, and C. Li, "Local feature-aware siamese matching model for vehicle re-identification," *Applied Sciences*, vol. 10, no. 7, 2020.

[6] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," 2015.