Car-make Classification Using Deep Learning Approaches

Giuseppe Labate†, Sana Amiri Manesh†

***Abstract*—This paper investigates a deep learning approach for image classification, focusing on the detection and recognition of car makes using the ResNet50 architecture. The study evaluates the performance of both manually trained and pre-trained ResNet50 models on the CompCars dataset, which features a diverse collection of car images. Two loss functions, focal loss and categorical cross-entropy loss, were analyzed for their impact on classification accuracy and their ability to address class imbalance within the dataset. Furthermore, the role of data augmentation techniques, such as image transformations, in enhancing the training process was systematically examined. Experimental results reveal that the pre-trained ResNet50 model, when combined with focal loss and augmented data, achieved superior performance in both accuracy and handling imbalanced classes. This research underscores the significance of combining pre-trained models, advanced loss functions, and data augmentation for optimal performance in car make classification tasks.**

***Index Terms*—Data Augmentation, Neural Networks, Residual Neural Networks, Optimization, Deep Learning.**

I. Introduction

Every day, a very large number of vehicles passes through the streets of our cities and towns. In fact, even though the car sales market volume is far from its 2007 peak , statistics from 2016 state that in Italy alone there are 625 passenger cars for every 1000 inhabitants, without considering trucks or motorbikes. It should not come as a surprise, then, that the idea of detecting and classifying these vehicles can be useful for a wide range of applications, from security to commercial use. [1]. In the rapidly advancing field of machine learning and computer vision, classification tasks, such as car make recognition, have garnered significant attention due to their practical applications in automotive industries and beyond. The availability of large-scale datasets, like the CompCars dataset, has enabled the development of sophisticated models for car make classification. However, challenges arise from the dataset's large size, class imbalances, and the inherent difficulty in distinguishing between similar car makes and models. This paper presents a comprehensive approach to car make classification by leveraging deep learning techniques, particularly using the ResNet-50 architecture, and addressing the aforementioned challenges with innovative solutions such as data augmentation and loss function optimization.

We begin by describing the high-level processing pipeline employed for this task, which includes data collection,data pre-processing, model design, and evaluation strategies. The dataset, CompCars, consists of a diverse range of images captured from various viewpoints, both in surveillance and web-based settings, providing a rich set of features for classification. Data pre-processing, including cropping using bounding boxes, plays a crucial role in managing the large dataset and enhancing model performance by focusing on the car's key attributes. Additionally, the paper compares the impact of different loss functions, cross-entropy and focal loss, on model accuracy and generalization. The results demonstrate that focal loss, in particular, significantly improves convergence and stability, while data augmentation enhances the model's ability to generalize and reduces overfitting.

Through this approach, we aim to present a robust solution for multi-class car make classification, demonstrating the effectiveness of ResNet-50 for such tasks while also contributing valuable insights into how various techniques can improve model performance and efficiency.

II. Related Work

Buzzelli et al. [2] make several notable contributions to car image classification. It highlights biases in existing dataset splits, proposes a more realistic training/testing configuration, and extends dataset annotations with hierarchical labels and car-tight bounding boxes. However, the findings primarily focus on dataset improvements rather than innovative classification methodologies. Skolik et al. [3] explore the optimization of hyperparameters for hybrid quantum-classical neural networks applied to car classification. It introduces a tensor train optimization (TTO) method, demonstrating improved efficiency and reduced computational cost compared to traditional grid search approaches. The hybrid quantum-classical model achieves higher classification accuracy on the Stanford Cars dataset than its classical counterparts. The study emphasizes the potential of quantum-inspired techniques in addressing complex machine learning tasks while suggesting further exploration of sample complexity and generalization bounds for such models. Dwivedi et al. [4] address vehicle damage classification and detection, focusing on automating insurance claim processing using deep convolutional neural networks. Pre-trained CNN models on ImageNet, combined with advanced techniques, achieved a classification accuracy of 96.39%, while the YOLO detector obtained a maximum mAP score of 77.78% for damage region detection. A proposed pipeline integrates classification and detection for enhanced robustness. The study highlights the need for a more diverse dataset to advance automated vehicle damage

†Department of Information Engineering, University of Padova, email: {name.surname}@studenti.unipd.it

Special thanks / Professor Jacopo Pegoraro

identification systems. Yang et al. [1] used CNNs to classify fine-grained car models from images, achieving significant improvements over traditional methods. CompCars dataset has been widely used for car model and part classification due to its extensive variety of car models and part annotations. Ni el al. [5] focuse on vehicle attribute recognition, ranging from coarse-grained classifications like vehicle type to fine-grained distinctions such as make and model. It surveys existing algorithms and compares two methods: direct classification and a flexible metric learning approach. A simulated real-world scenario is designed to evaluate these methods, providing insights into their effectiveness. This study advances the understanding of vehicle attribute recognition by bridging gaps in literature and testing practical applications.

Class imbalance is a common issue in real-world datasets, particularly in tasks like car part classification, where certain parts or models may be overrepresented compared to others. Traditional Loss functions like categorical cross-entropy often perform poorly in such scenarios, leading to biased model prediction.Lin et al. [6] introduce focal loss in the context of dense object detection. Focal loss introduced as a modification of cross-entropy loss, mitigates this issue by focusing on hard-to-classify examples. It down-weights the loss contribution from easy examples allowing the model to focus more on underrepresented or challenging cases. Zhu et al. [7] focus on addressing the class imbalance issue in driving scene recognition, which results in different distribution patterns between majority and minority classes. They propose a novel class focal loss method that integrates category quantity distribution to better balance easy and difficult samples during training, improving recognition accuracy for underrepresented classes.

III. Processing Pipeline

This section outlines the high-level approach and methodology employed for the deep learning-based classification of car makes using the CompCar dataset. The pipeline encompasses several stages, each designed to ensure data integrity, model robustness, and accurate predictions. The key stages in the pipeline are as follows:
Data Collection and Preprocessing : The project begins with the CompCar dataset, a comprehensive dataset containing diverse images of cars with associated metadata. The data preprocessing stage ensures the input data is prepared for the deep learning models by performing the following steps: Data Normalization, Data Augmentation
Data Splitting: The dataset is partitioned into three subsets: Training Set, Validation Set, Test Set
Model Design and Training: The project employs a convolutional neural network (CNN) architecture tailored for image classification tasks. Key aspects of model design and training include:

- Model Architecture: Selection of a pre-trained deep learning model (e.g., ResNet) and fine-tuning its layers to leverage transfer learning.
- Loss Function: Use of a categorical cross-entropy loss function and focal loss suitable for multi-class classification problems.
- Optimization: Employing optimization techniques such as Adam with a learning rate scheduler to enhance training efficiency.
- Regularization: Incorporation of dropout and early stopping to mitigate overfitting.

Evaluation Metrics: During and after training, the model is evaluated using the top-1 and top-5 accuracy metrics. *Model Validation and Early Stopping:* The training process utilized:

- Checkpointing: Automatically saving the model with the best performance on the validation set.
- Early Stopping: Halting the training process when the validation performance ceased to improve, preventing overfitting and saving computational resources.

Final Model Evaluation: For the final evaluation, the model was tested on the held-out test set. The following metrics were computed: Test Loss, top-1 and top-5 accuracy on test dataset.

IV. Dataset

"CompCars" data set is a large-scale dataset that covers not only different car views, but also their different internal and external parts, and rich attributes. Importantly, the dataset is constructed with a *cross-modality* nature, containing a *surveillance-nature* set and a *web-nature* set. In particular, the *web-nature* data contains 161 car makes with 1,687 car models, covering most of the commercial car models in the recent ten years. There are a total of 136, 727 images capturing the entire cars and 27,618 images capturing the car parts, where most of them are labeled with *attributes* and *viewpoints*. The *surveillance-nature* data contains 50,000 car images captured in the front view. Each image in the *surveillance-nature* partition is annotated with *bounding box*, *model*, and *color* of the car. [1]

Data preparation: One of the significant challenges faced while working with the CompCars dataset was its large size, which exceeded the computational capacity of the system. To address this, a *preprocessing step* was implemented using the *bounding boxes* provided in the dataset's label folder to crop the images, focusing exclusively on the cars while removing unnecessary background details. This approach significantly reduced the size of the dataset, making it more manageable for the computer without compromising the essential features required for *classification*. The advantages of cropping extended beyond reducing dataset size. By focusing solely on the car, the input data fed into the model became cleaner, with less noise from background elements such as roads, trees, or other vehicles. This likely improved the *model's performance* by emphasizing the features most relevant to the *classification task*, such as car shape, design, and unique attributes. Additionally, training efficiency was enhanced, as smaller images required less memory and computational resources, allowing the model to process more data in less time.

1) **Loading Image Paths and Labels:** Image paths and their associated labels are loaded directly from the

dataset directory structure. Each class (car make or model) is represented by its corresponding image paths, which are mapped to unique integer labels derived from the folder hierarchy. The CroppedCarDataset class dynamically iterates through the dataset directories, extracting image paths and associated labels based on their file paths. The correctness of labels is ensured by leveraging the folder structure (e.g., car make and model names) and optionally validating against external label mappings such as .mat files.

2) **Data Augmentation:** To address class imbalance in the dataset, augmentation techniques are applied to generate additional samples for underrepresented classes.Additionally, data augmentation helps prevent overfitting by artificially increasing the diversity of the training data, allowing the model to generalize better to unseen data.
Augmentation Pipeline: The augmentation process uses the following transformations:
   - **Geometric Transformations:** Random affine transformations, rotations, and horizontal flips.
   - **Color Transformations:** Adjustments to brightness, contrast, saturation, and hue.
   - **Resizing and Cropping:** Random resized cropping and center cropping to ensure consistent input dimensions.

   These transformations are implemented using Pytorch 's transforms. Compose functionality and applied dynamically during training.
Training vs. Validation/Test Transforms: Different transformations are applied to the training set compared to the validation and test sets to serve distinct purposes:
   - **Training Set:** Augmentations such as random affine transformations, color jitter, and random cropping introduce variability in the training data. This helps the model generalize better by simulating real-world conditions and reducing overfitting.Transformations like random flips and rotations ensure that the model is not biased toward specific orientations or positions.
   - **Validation and Test Sets:** Only basic transformations (e.g., resizing and normalization) are applied to maintain consistency and evaluate the model's performance on clean, unaltered data.Augmentations are avoided to ensure that the validation and test results reflect the model's true generalization ability.

   Balancing Class Distributions: Underrepresented classes with fewer than 250 samples are identified using a label counting function. For each of these classes, augmented samples are generated to bring their total count to 400. The AugmentedDataset class combines the original and augmented samples into a unified dataset for training.
3) **Data Splitting and Loaders:** The dataset is split into training, validation, and test sets:
   - **Training Set:** Includes augmented samples and original data, shuffled for robust training.
   - **Validation Set:** Comprises 30% of the training data and is used for hyperparameter tuning and early stopping.
   - **Test Set:** Contains unseen data for evaluating the model's performance.

   Data loaders are created using PyTorch's DataLoader class to enable efficient data handling with batch processing and parallelism. Batch size is set to 32, and transformations are applied dynamically during data loading.

V. Learning Framework

1) **Model Construction:** The backbone of the learning framework is the ResNet-50 architecture, modified for multi-class classification with 163 car makes.
Residual Blocks: The architecture includes custom implementations of IdentityBlock and ConvolutionalBlock. The IdentityBlock preserves input dimensions, while the ConvolutionalBlock performs down-sampling with a stride of 2.
Custom Layers:
   - **Dropout Layers:** Applied after residual blocks with a rate of 0.2 to mitigate overfitting.
   - **Batch Normalization:** Normalizes activations after convolutional layers to improve convergence and training stability.
   - **Final Linear Layer:** A fully connected layer maps the extracted features to the 163 output classes, representing car makes. The input size to the linear layer is 8192, computed using the formula:

$$O = \left(\frac{d - F + 2p}{S}\right) + 1 \tag{1}$$

   Here, d is the input dimension, F is the filter size, P is the padding, and S is the stride. This formula is used to compute the size of the output of convolutional layers, ensuring the correct input size for the final layer.
Output Function: The model uses the softmax activation function on the output layer to calculate class probabilities.
Weight Initialization and Pre-trained Models: Xavier initialization is applied to both convolutional and fully connected layers to improve convergence when training from scratch. Additionally, a pre-trained ResNet-50 model can be used, which leverages weights initialized from ImageNet, allowing the network to fine-tune features for car make classification.
2) **Training Process:** The models were trained for 20 epochs with robust optimization and monitoring strategies to ensure effective learning. The training process evaluates the performance of two loss functions.

Cross-Entropy Loss: This loss function was primarily used for standard multi-class classification tasks. It calculates the difference between predicted class probabilities and true class labels.

Focal Loss: A custom implementation of focal loss was also utilized to address class imbalance. This function emphasizes hard-to-classify samples using parameters α (alpha = 1) and γ (gamma = 2).

Optimization: The Adam optimizer with a learning rate of 0.001 was employed for parameter updates. A StepLR scheduler reduced the learning rate by a factor of 0.1 every 7 epochs.

Early Stopping: Training incorporated early stopping with a patience of 5 epochs, halting further training if validation loss stopped improving.

Checkpointing: Checkpoints were saved whenever the validation loss improved, ensuring that the best-performing model weights were preserved.

3) **Evaluating and Testing:** The trained modelâ€™s performance is evaluated on unseen test data using detailed metrics.

   Test Loop: The model is set to evaluation mode, and test data is processed in batches of size 16. Predictions and ground truth labels are stored for comparison.

   Metrics:

   - **Top-1 Accuracy:** Measures the percentage of correct predictions.
   - **Top-5 Accuracy:** Checks if the correct class is among the top 5 predictions, which is particularly useful for multi-class tasks.
   - **Test Loss:** Quantifies the overall model error on test data.

   Inference: A function is provided to predict the class of a single image by pre-processing it, passing it through the model, and decoding the output.

VI. Results

The car make classification task was evaluated using the ResNet50 architecture trained with two loss functions: cross-entropy and focal loss. Both models were trained for 20 epochs, and their performance was evaluated on the training, validation, and test sets. The results highlight the impact of the loss function on model accuracy, loss, and generalization to unseen data. Additionally, the value of data augmentation was evident in improving the model's ability to generalize by increasing the diversity of training samples and reducing overfitting.

1) **Impact of Loss Functions:** The comparison between focal loss and cross-entropy loss revealed key differences in performance. When using the focal loss, the model showed better convergence and stability, as evident from smoother loss curves. Validation accuracy improved steadily, demonstrating the model's ability to handle class imbalances. In contrast, training with cross-entropy loss converged more quickly but exhibited higher fluctuations in validation loss. Additionally, the model struggled with difficult-to-classify classes, leading to lower overall validation accuracy.

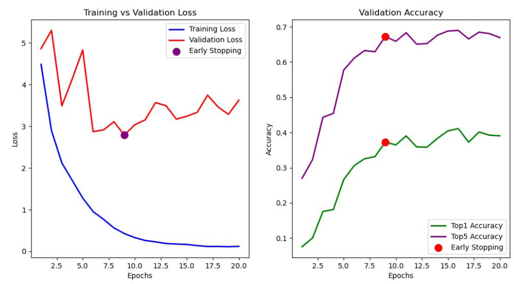

Fig. 1: Training and validation loss curve (left) and accuracy curve (right) for the model using cross-entropy loss.

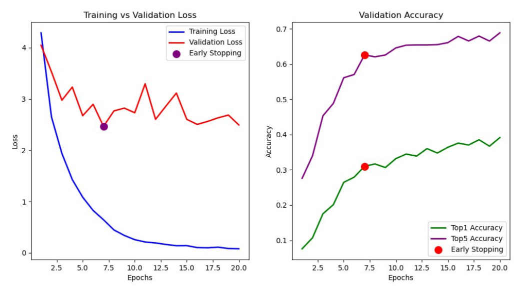

Fig. 2: Training and validation loss curve (left) and accuracy curve (right) for the model using focal loss.

2) **Impact of Data Augmentation:** In this section, we present the results of applying data augmentation to address class imbalance in the dataset. The Fig.3 shows the number of images per car make before and after the augmentation process.Before data augmentation, the number of images per car make varied significantly, with some classes having very few images while others had a large number of samples. This disparity created an imbalance, which could potentially hinder the model's ability to learn effectively from underrepresented classes. After applying data augmentation, the number of images per car make became more balanced, with all classes reaching a more consistent count. The augmented dataset ensures that the model has a more equal distribution of data across all classes, improving the training process and helping to prevent overfitting.
   The impact of data augmentation on the model's performance was significant. Without data augmentation, the model showed clear signs of overfitting, as evidenced by the larger gap between training and validation loss curves. Validation accuracy also peaked at a lower level compared to experiments with augmented data. Conversely, the use of data augmentation greatly improved generalization by diversifying the training examples.

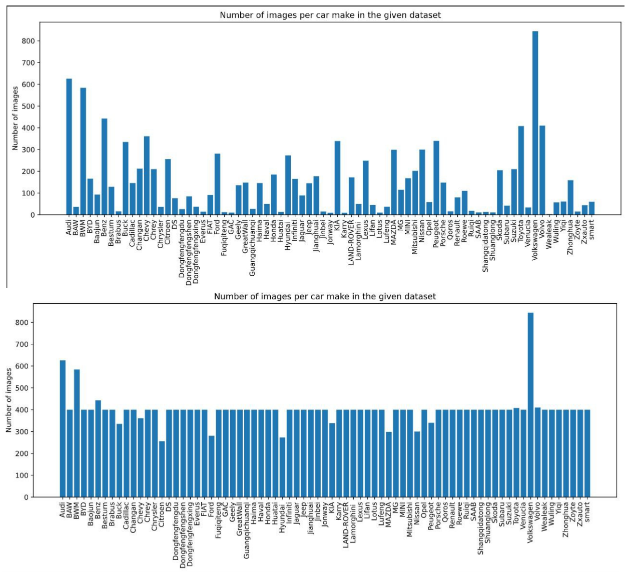

Fig. 3: Number of images for each car˙make before applying the data augmentation (top) and Number of images for each car˙make after applying the data augmentation (buttom)

This stabilization in validation loss and consistent improvement in accuracy demonstrated the critical role of augmentation in enhancing the robustness of the model.

VII. Concluding Remarks

In conclusion, this project successfully demonstrated the application of deep learning techniques for car make classification using the CompCars dataset. The results underscore the effectiveness of pre-trained ResNet50, data augmentation, and focal loss in addressing class imbalance and achieving robust classification performance.

VIII. Exam rules

References

[1] C. C. L. Linjie Yang, Ping Luo and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3973–3981, 2015.

[2] M. Buzzelli and L. Segantin, "Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments and results," *Sensors*, vol. 21, no. 2, p. 596, 2021.

[3] A. M. D. K. M. P. A. M. A. S. Asel Sagingalieva, Andrii Kurkin and D. V. Dollen, "Hyperparameter optimization of hybrid quantum neural networks for car classification," *arXiv*, vol. 2205, no. 04878, 2022.

[4] S. N. O. E. B. M. B. K. S. R. S. A. T. Mahavir Dwivedi, Hashmat Shadab Malik and A. Rathi, "Deep learning-based car damage classification and detection," in *Proceedings of the International Conference on Artificial Intelligence and Data Engineering*, vol. 1133 of *Advances in Intelligent Systems and Computing*, pp. 207–221, Springer, 2020. First Online: 14 August 2020.

[5] X. Ni and H. Huttunen, "Vehicle attribute recognition by appearance: Computer vision methods for vehicle type, make and model classification," *Pattern Recognition Letters*, vol. 93, pp. 357–368, 2021. Open access, Published: 26 June 2020.

[6] R. G. K. H. P. D. Tsung-Yi Lin, Priya Goyal, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2017 of *IEEE*, pp. 2980–2988, IEEE, 2017. First Online: 2 October 2017.

[7] L. Y. K. L. Xianglei Zhu, Jianfeng Men, "Imbalanced driving scene recognition with class focal loss and data augmentation," *Neural Processing Letters*, vol. 13, pp. 2957–2975, 2022. Published: 07 June 2022.