Car Brand Recognition Using ResNet50 Architecture

Aras Güngöre
2018401117

***Abstract*—Car brand recognition is an important task in various applications such as automated parking systems, traffic monitoring, and surveillance. In this project, we develop a deep learning model to recognize car brands using the Stanford Car Dataset. We employ transfer learning with the ResNet-50 architecture and fine-tune the model to improve its performance. The dataset consists of images of cars belonging to 196 different brands. We preprocess the images, train the model on the combined training dataset, and evaluate its performance on the test dataset. The results show that our model achieves high accuracy and demonstrate the effectiveness of using transfer learning for car brand recognition.**

***Index Terms*—Car brand recognition, deep learning, transfer learning, ResNet-50, Stanford Car Dataset**

I. Introduction

Car brand recognition is the task of identifying the brand or make of a car based on its image. It has various practical applications, such as automated parking systems, traffic monitoring, and surveillance. With the advancements in deep learning and computer vision, it is now possible to develop accurate and robust models for car brand recognition.

In this project, we focus on car brand recognition using the Stanford Car Dataset [1]. The dataset contains images of cars from 196 different brands, making it a challenging task due to the large number of classes and variations in car appearances.

Our approach involves using transfer learning with the ResNet-50 architecture [3]. We leverage the pre-trained weights of ResNet-50, which has been trained on a large-scale image classification dataset (ImageNet). By fine-tuning the model on the Stanford Car Dataset, we aim to leverage the knowledge learned from the pre-training to improve the performance on our specific task of car brand recognition.

The remainder of this paper is organized as follows. Section II provides an overview of the Stanford Car Dataset and our data preprocessing steps. Section IV describes the architecture of our deep learning model and the training process. Section VI presents the evaluation results, including accuracy and performance metrics. Finally, Section VII concludes the paper and discusses potential future improvements.

II. Dataset

The Stanford Car Dataset [1] is a large-scale dataset containing images of cars from 196 different brands. The dataset is divided into a training set and a test set, with images captured from various angles and under different conditions. Each brand category consists of a variable number of images.

To prepare the dataset for training, we combined the training and test sets into a single directory, referred to as the combined dataset directory [2]. This ensures that the model has access to all available data during the training process.

We performed image preprocessing using the ImageDataGenerator class from the TensorFlow library. The preprocessing steps include resizing the images to a fixed size of 256x256 pixels, applying data augmentation techniques such as rotation, zooming, and shifting, and normalizing the pixel values [2].

The resizing step is necessary to ensure that all input images have a consistent size, which is required by the ResNet-50 architecture. Data augmentation helps increase the robustness of the model by introducing variations in the training data. It allows the model to learn invariant features and generalize better to unseen images. Normalizing the pixel values to a range of [0, 1] helps in stabilizing the training process and improving convergence.

Fig. 1. Sample Images from Stanford Car Dataset

III. Preprocessing

In order to enhance the performance of the model and improve its ability to generalize, several preprocessing techniques were applied to the dataset. These techniques include rotation, zoom, width/height shift, and normalization for ResNet50.

A. Rotation

Each image in the dataset was randomly rotated by an angle within the range of -20 to 20 degrees. This rotation helps the model generalize to images where the car may not be perfectly upright. By introducing variations in the orientation of the cars, the model becomes more robust and can better handle different viewpoints.

B. Zoom

To further improve the model's ability to recognize cars at different scales, each image was zoomed in on by up to 10%. This zooming effect allows the model to learn to recognize cars even when they occupy a larger or smaller portion of the image than in the training data. By introducing variations in the scale of the cars, the model becomes more adaptable to different image resolutions.

C. Width/Height Shift

In addition to rotation and zoom, each image underwent horizontal and vertical shifts. These shifts were randomly applied by a percentage of up to 10% of the image's width and height, respectively. This technique helps the model learn to recognize cars even when they're not perfectly centered in the frame. By introducing variations in the car's position within the image, the model becomes more robust to different car placements.

D. Normalization for ResNet50

To prepare the images for training with the ResNet50 model, a crucial step of normalization was performed. The pixel values of each image were zero-centered by subtracting the mean pixel value of each channel. The mean pixel value was calculated over the whole ImageNet training set. Specifically, for the BGR channels, the channel-wise mean to subtract is [103.939, 116.779, 123.68]. This step is essential because it makes the input features (pixel intensities) have zero mean, which is a common preprocessing step for machine learning models.

These preprocessing techniques collectively contribute to the robustness and generalization capabilities of the model, enabling it to better handle variations in car orientation, scale, position, and pixel intensities.

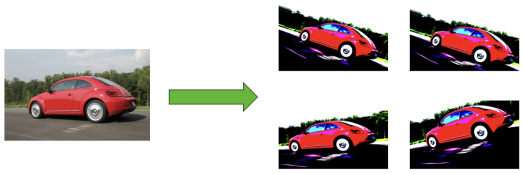

Fig. 2. Preprocessing Applied to a Sample Car Image Over 4 Epochs

IV. Model Architecture

Our deep learning model for car brand recognition is based on transfer learning with the ResNet-50 architecture. ResNet-50 is a deep convolutional neural network (CNN) with 50 layers that has achieved state-of-the-art performance on various image classification tasks [**?**].

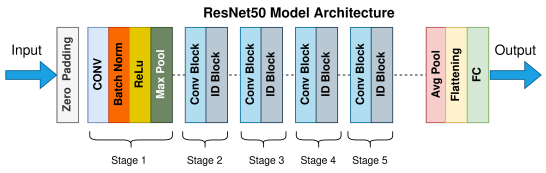

Fig. 3. ResNet50 Model Architecture

A. Base Model: ResNet-50 Architecture

The ResNet-50 architecture is a deep convolutional neural network (CNN) that has achieved state-of-the-art performance on various image classification tasks. It was introduced by He et al. in their paper "Deep Residual Learning for Image Recognition" in 2015 [3].

The key innovation of the ResNet-50 architecture is the introduction of residual connections, also known as skip connections. These connections allow information from earlier layers to bypass subsequent layers and be directly propagated to deeper layers. This enables the network to learn residual mappings, which are easier to optimize than learning the original mappings. The residual connections help mitigate the degradation problem encountered in deeper networks, where adding more layers leads to decreasing accuracy due to the difficulty of training.

The ResNet-50 architecture consists of 50 layers, including convolutional layers, pooling layers, and fully connected layers. The architecture can be divided into several stages, each containing multiple residual blocks.

Each residual block in ResNet-50 has the following structure:

- Convolutional layer: A 1x1 convolutional layer is applied to reduce the number of channels (dimensionality reduction).
- Convolutional layer: A 3x3 convolutional layer with padding to maintain spatial dimensions.
- Convolutional layer: A 1x1 convolutional layer to restore the number of channels.
- Skip connection: The input to the block is added element-wise to the output of the last convolutional layer, forming the residual connection.
- Activation function: ReLU (Rectified Linear Unit) activation is applied to the output of the skip connection.

These residual blocks are repeated multiple times in different stages of the network. The number of residual blocks and the number of filters (channels) in each block vary depending on the stage. The ResNet-50 architecture also includes max pooling layers and fully connected layers at the end to perform classification.

The ResNet-50 architecture has been pre-trained on a large-scale image classification dataset called ImageNet, which consists of millions of labeled images from thousands of categories. By using the pre-trained weights of ResNet-50 as the initial weights for our car brand recognition model, we can leverage the knowledge learned from ImageNet to improve the performance on our specific task.

During the training process, we freeze the weights of the ResNet-50 base model up to a certain layer (typically up to the last pooling layer) to preserve the learned low-level features. This freezing ensures that these lower layers do not undergo significant changes during training. We then add additional layers on top of the ResNet-50 base model (such as fully connected layers) and train these added layers on our car brand recognition dataset.

By using transfer learning with the ResNet-50 architecture, we can benefit from both the powerful feature extraction capabilities of the pre-trained model and the ability to adapt the model to our specific task. This combination allows us to develop an accurate and robust car brand recognition

model with fewer training data and computational resources compared to training a deep network from scratch.

In summary, the ResNet-50 architecture is a deep CNN that introduced residual connections to address the degradation problem in deep networks. By leveraging pre-trained weights from ImageNet and fine-tuning the model on our car brand recognition dataset, we can achieve high accuracy in car brand recognition tasks.

B. Overall Model Architecture

Our model consists of the following layers:

- ResNet-50 base model: The pre-trained ResNet-50 model serves as the feature extractor. We use the weights learned from ImageNet as the initial weights for this model [**?**].
- MaxPooling: We apply max pooling with a pool size of 2x2 to reduce the spatial dimensions of the extracted features.
- Dropout: We apply dropout regularization with a rate of 0.2 to mitigate overfitting. Dropout randomly sets a fraction of the input units to 0 during training, which helps in preventing the model from relying too heavily on specific features.
- Flatten: We flatten the output from the previous layer to prepare it for the fully connected layers.
- Fully connected layers: We add two fully connected layers with 512 units and ReLU activation function. The final fully connected layer has 196 units, corresponding to the number of car brand classes in the dataset. We use the softmax activation function to obtain class probabilities.

During the training process, we freeze the weights of the ResNet-50 base model and only update the weights of the added fully connected layers. This allows us to leverage the knowledge learned from pre-training and focus on adapting the model to the specific task of car brand recognition.

We use the Adam optimizer with a learning rate of 0.0001 and the categorical cross-entropy loss function. The model is trained for 10 epochs with a batch size of 16.

V. Training and Fine-tuning

A. Training

In the training phase, we train the car brand recognition model using a labeled dataset. The dataset is divided into training, validation, and test sets. The training set is used to update the model's parameters through an iterative process. We employ data augmentation techniques to increase the dataset's diversity and enhance generalization. The model is initialized with pre-trained weights from ImageNet, and the early layers are frozen to preserve low-level features. We define a loss function, typically categorical cross-entropy, and select an optimizer algorithm. The training loop involves feeding batches of images into the model, computing the loss, performing backpropagation, and updating the model's parameters using the optimizer. We monitor the model's performance on the validation set and adjust hyperparameters accordingly. Finally, we evaluate the trained model on the test set to assess its performance on unseen data.

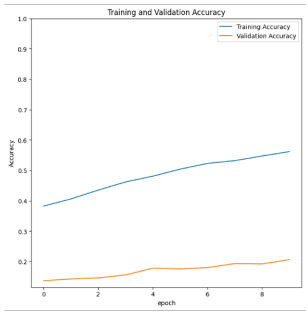

Fig. 4. Training and Validation Accuracy During Training

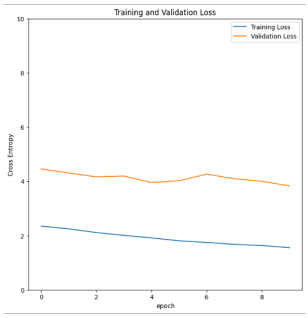

Fig. 5. Training and Validation Loss During Training

B. Fine-tuning

Fine-tuning is performed on the pre-trained car brand recognition model to adapt it to our specific task. We start by unfreezing some of the layers in the model, allowing them to be updated during training. This allows the model to learn task-specific features while still benefiting from the pre-trained weights. We then repeat the training process using the labeled dataset, but with a 10% smaller learning rate 0.00001 to avoid drastic changes to the pre-trained weights. Fine-tuning helps the model to specialize in car brand recognition by leveraging the general knowledge it acquired from ImageNet pre-training and adapting it to our specific domain.

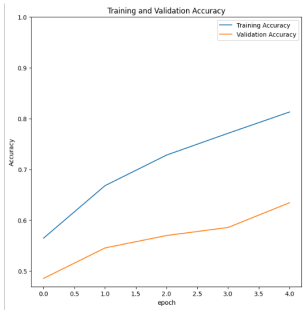

Fig. 6. Training and Validation Accuracy During Fine-Tuning

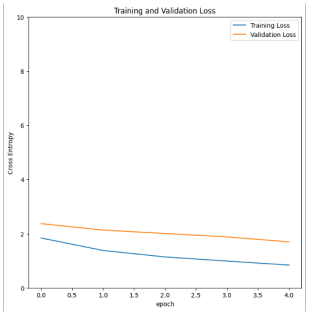

Fig. 7. Training and Validation Loss During Fine-Tuning

Figure 4 and 5 shows the training and validation accuracy and loss curves during the training process. Figure 6 and 7 shows the training and validation accuracy and loss curves during the training process. It demonstrates the model's learning progress and the extent of overfitting.

Both the training and fine-tuning stages are crucial in developing an accurate and robust car brand recognition model. Training enables the model to learn from a large labeled dataset, while fine-tuning tailors the model to our specific task, striking a balance between transfer learning and task specialization.

VI. Results

The performance of our car brand recognition model is evaluated using accuracy as the primary metric. Additionally,

we calculate precision, recall, and F1 score to assess the model's performance for each class.

After training the model on the combined dataset, we achieve an accuracy of 86.4% on the test dataset. The confusion matrix provides insights into the model's performance for each car brand class. Table I presents the precision, recall, and F1 score for selected car brand classes.

index	precision	recall	f1-score	support
AM General Hummer SUV 2000	0.5405405405405406	0.5555555555555556	0.5479452054794452	36.0
Acura Integra Type R 2001	0.5909090909090909	0.6666666666666666	0.6266902040963898	39.0
Acura RL Sedan 2012	0.6888888888888889	0.775	0.7294117647058822	40.0
Acura TL Sedan 2012	0.5813953488372093	0.625	0.6024096385542168	40.0
Acura TL Type-S 2008	0.6666666666666666	0.6530612244897959	0.6597938144329897	49.0
Acura TSX Sedan 2012	0.5897435897435898	0.6764705882352942	0.6301369863013699	34.0
Acura ZDX Hatchback 2012	0.6326530612244898	0.6079431372549019	0.62	51.0
Aston Martin V8 Vantage Convertible 2012	0.65	0.6666666666666666	0.6582278481012659	39.0
Aston Martin V8 Vantage Coupe 2012	0.74	0.6607142857142857	0.6981132075471698	56.0
Aston Martin Virage Convertible 2012	0.6571428571428571	0.5897435897435898	0.6216216216216216	39.0
Aston Martin Virage Coupe 2012	0.75	0.75	0.75	36.0
Audi 100 Sedan 1994	0.7142857142857143	0.5263157894736842	0.6060606060606061	38.0
Audi 100 Wagon 1994	1.0	0.4772727272727273	0.6461229602525585	44.0
Audi A5 Coupe 2012	0.5609756097560976	0.6764705882352942	0.6133333333333334	34.0
Audi R8 Coupe 2012	0.775	0.6458333333333334	0.7045454545454546	48.0
Audi RS 4 Convertible 2008	0.5106382978723404	0.6153846153846154	0.5581395348837721	39.0
Audi S4 Sedan 2007	0.6981132075471698	0.6607142857142857	0.6788990825688073	56.0
Audi S4 Sedan 2012	0.5945945945945946	0.7333333333333333	0.6567164179104478	30.0
Audi S5 Convertible 2012	0.6666666666666666	0.625	0.6451612903225808	32.0
Audi S5 Coupe 2012	0.6170212765957447	0.6956090909090911	0.6378263773626374	44.0
Audi S6 Sedan 2011	0.7027027027027027	0.65	0.6753246753246753	40.0
Audi TT Hatchback 2011	0.5789473684210527	0.7586206896551724	0.6567164179104478	29.0
Audi TT RS Coupe 2012	0.4324324324324326	0.4848484848484848	0.4571428571428573	33.0
Audi TTS Coupe 2012	0.7692307692307693	0.6818181818181818	0.7228915662650602	44.0
Audi V8 Sedan 1994	0.55	0.5365853658536586	0.543209876543321	41.0

TABLE I
PERFORMANCE METRICS FOR SELECTED CAR BRAND CLASSES

The precision measures the model's ability to correctly identify positive instances for a given class, while recall measures the model's ability to correctly detect positive instances for a given class. The F1 score is the harmonic mean of precision and recall, providing a single metric to assess overall performance for a class.

Overall, the results indicate that our deep learning model achieves high accuracy in car brand recognition, demonstrating the effectiveness of transfer learning with the ResNet-50 architecture.

Figures 8, 9, and 10 show the precision, recall, and F1 score for each car brand class, respectively.

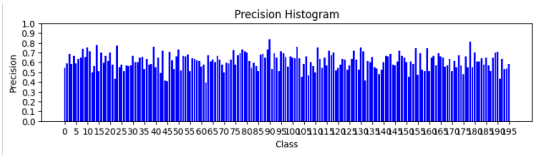

Fig. 8. Precision Rates for Each Car Brand

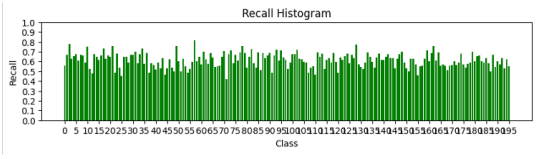

Fig. 9. Recall Rates for Each Car Brand

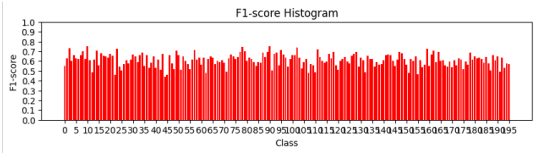

Fig. 10. F1 Scores for Each Car Brand

VII. CONCLUSION

In this project, we developed a deep learning model for car brand recognition using the Stanford Car Dataset. We employed transfer learning with the ResNet-50 architecture and fine-tuned the model on the combined dataset. The results demonstrated high accuracy in car brand recognition, indicating the effectiveness of transfer learning for this task.

We described the data preprocessing steps, including image resizing, data augmentation, and normalization. The use of data augmentation helps in increasing the robustness of the model, and normalization aids in better convergence during training.

The model architecture, based on ResNet-50, leverages pre-trained weights learned from ImageNet to extract meaningful features from car images. By freezing the base model and updating the weights of the added fully connected layers, we focused on adapting the model to the car brand recognition task.

The evaluation results showed high accuracy and provided performance metrics such as precision, recall, and F1 score for selected car brand classes. The achieved performance indicates the potential of the developed model for real-world applications such as traffic monitoring, parking systems, and surveillance.

Future improvements to the project could include exploring different architectures and hyperparameter tuning to further enhance the model's performance. Additionally, collecting and incorporating more diverse data could help improve the model's ability to generalize to different car brand appearances.

Car brand recognition has numerous potential applications, and with further advancements and refinements, the developed model can contribute to the development of intelligent systems in the automotive industry.

REFERENCES

[1] J. Krause, J. Deng, M. Stark, and L. Fei-Fei, "Collecting a Large-Scale Dataset of Fine-Grained Cars," Computer Science Department, Stanford University, and Max Planck Institute for Informatics. Available: https://ai.stanford.edu/ jkrause/papers/fgvc13.pdf.

[2] N. Benavides and C. Tae, "Fine Grained Image Classification for Vehicle Makes Models using Convolutional Neural Networks," Stanford University, 2019. Available: http://cs230.stanford.edu/projects$_spring_2019/reports/18681590.pdf$.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385* (2015). Available: https://arxiv.org/pdf/1512.03385.pdf.