AutoSpecNet
A Multi-Task Learning Car Recognition Method based on MobileNetV2

Luoyan Zhang[1]*
[1]Northeastern University
360 Huntington Ave
Boston, Massachusetts 02115 USA
lyzhang0113@ece.neu.edu

Abstract

In the rapidly evolving field of computer vision, fine-grained image classification remains a critical challenge, particularly in applications such as automotive recognition. This study introduces AutoSpecNet[1], a robust machine learning model leveraging the architecture of MobileNetV2 (Sandler et al. 2018) combined with multi-task learning techniques to classify car images by year, make, and type. The model innovatively extends MobileNetV2 by integrating additional fully connected layers dedicated to each classification task, thereby enabling simultaneous predictions with shared feature extraction. Using a dataset comprised of car images, the model was trained and tested, with transformations applied to enhance generalization capabilities. AutoSpecNet demonstrated superior performance in predicting discrete attributes of cars compared to traditional single-task approaches. The results indicated a notable improvement in classification accuracy for the year, make, and type of cars, showcasing the effectiveness of our approach in handling the intricacies of multi-task classification within the automotive domain. This work not only advances the state of automotive image recognition but also offers a scalable framework for other fine-grained classification tasks in computer vision.

Introduction

The ability to accurately classify images of cars according to specific attributes such as year, make, and type is a significant challenge in the field of computer vision. This capability has extensive applications across various industries, including automotive manufacturing, insurance, and law enforcement. Accurate and automated car recognition systems can assist in tasks ranging from inventory management in sales to vehicle identification in traffic and security surveillance scenarios.

Traditional image classification approaches often rely on single-task learning frameworks that treat each classification task (year, make, and model) independently. While effective to a degree, these methods do not capitalize on the potential synergies between related classification tasks. Multi-task learning (MTL) addresses this limitation by simultaneously learning multiple related tasks, leveraging commonalities and differences across tasks to improve the generalization performance of the model. MTL not only enhances learning efficiency and prediction accuracy but also reduces the computational overhead compared to running multiple independent models.

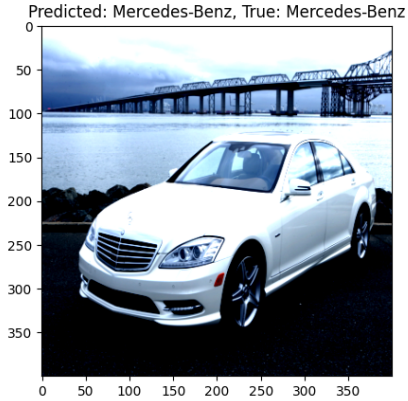

Figure 1: An example of predicting the make of a car successfully

In this study, we introduce AutoSpecNet, a multi-task learning model that integrates the advanced capabilities of MobileNetV2 (Sandler et al. 2018) with specific architectural enhancements to classify cars into their year, make, and type. MobileNetV2 is chosen for its efficiency and effectiveness in feature extraction, which is crucial for handling the high variability and fine-grained nature of automotive images. By adapting MobileNetV2, we enhance its architecture to support multi-task learning, adding separate fully connected layers that target each classification task while sharing lower-level visual representations.

The decision to use MobileNetV2 as a base model stems from its proven performance in mobile and embedded vision applications, where computational efficiency is paramount. This feature is particularly advantageous for real-time applications such as in traffic monitoring systems, where quick and accurate predictions are necessary. Our model is further refined with a set of image transformations to enhance its

*With help from the course instructor Chris Amato and the hardworking TAs of CS 5100

[1]GitHub Repo: https://github.com/lyzhang0113/AutoSpecNet

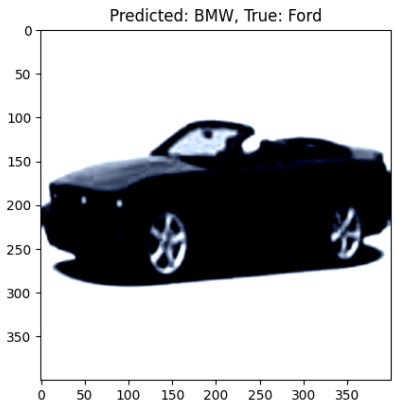

Figure 2: An example of predicting the make of a car unsuccessfully

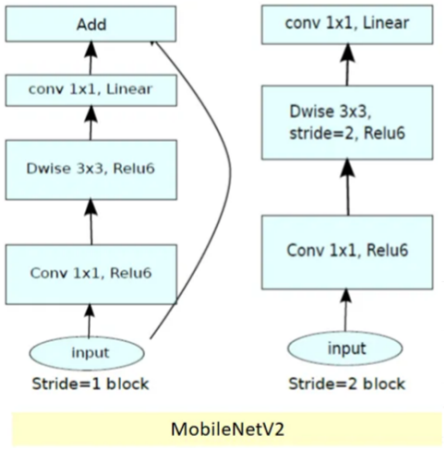

Figure 3: The Architecture of MobileNetV2

robustness and ability to generalize from training data to unseen real-world scenarios.

AutoSpecNet represents a novel approach in the realm of automotive image classification by harnessing the strengths of both convolutional neural networks and multi-task learning. This integration not only addresses the intrinsic challenges of fine-grained classification tasks but also sets a new benchmark for accuracy and efficiency in car recognition technologies.

Background

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) (Sharma, Jain, and Mishra 2018) are a class of deep neural networks highly effective for analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are composed of one or more convolutional layers (often with a subsampling step) followed by one or more fully connected layers. The key feature of CNNs is their ability to develop an internal representation of a two-dimensional image. This capability makes them exceptionally adept at tasks such as image recognition and classification, where they can identify features with spatial hierarchies.

CNNs automatically detect important features without any human supervision. For instance, in a vehicle identification task, a CNN might begin by detecting edges in the first layer, then shapes by combining these edges in the second layer, and finally specific vehicle components in higher layers. This hierarchical feature extraction makes CNNs very efficient for the task of image classification.

MobileNetV2

MobileNetV2 (Sandler et al. 2018) is a significant iteration on the architecture that was specifically designed for mobile and resource-constrained environments. It builds on the ideas from MobileNetV1 (Howard et al. 2017), introducing two key features: linear bottlenecks and inverted residuals. MobileNetV2 uses depthwise separable convolutions which factorize a standard convolution into a depthwise convolution and a 1x1 convolution, reducing computational cost and model size. (Fig. 3) These features make MobileNetV2 not only lightweight but also efficient without compromising much on accuracy, making it ideal for real-time applications and edge computing.

Multi-task Learning (MTL)

Multi-task Learning (MTL) (Crawshaw 2020) is a learning paradigm in machine learning where multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks. This approach to learning leverages shared representations, usually leading to improved learning efficiency and prediction accuracy compared to training separate models for each task. MTL is particularly useful when the tasks are related, as the tasks can provide additional information to each other, which can reduce overfitting and improve model generalization.

In the context of car recognition, multi-task learning allows the model to simultaneously predict multiple attributes (e.g., make, model, and year) of the vehicle. By sharing representations among related tasks, the model can better generalize to new, unseen examples, which is crucial for robust performance across different environments and datasets.

Related Work

The problem of car recognition has been approached through various methodologies in computer vision, each with its strengths and specific applications. This section reviews some of these methods, discusses their relation to the AutoSpecNet project, and explains the choices made in our approach.

Traditional Machine Learning Approaches

Before the widespread adoption of deep learning, traditional machine learning techniques such as Support Vector Machines (SVMs) (Cervantes et al. 2020) and Random Forests (Louppe 2015) were commonly used for image classification

Figure 4: The Stanford Cars Dataset

tasks. These methods often relied on handcrafted features extracted from images, such as color histograms, texture, and edge features. However, they generally lacked the ability to automatically learn feature representations from data, which limited their effectiveness in complex image recognition tasks like distinguishing between different car models or recognizing subtle year-to-year changes.

Single-Task Deep Learning Models

With the advent of deep learning, single-task convolutional neural networks (CNNs) have become the standard for image classification due to their ability to learn powerful hierarchical features directly from image pixels. Architectures like AlexNet (Krizhevsky, Sutskever, and Hinton 2012), VGG (Simonyan and Zisserman 2015), and ResNet (He et al. 2015) have demonstrated significant successes in various image recognition benchmarks. While these models offer robust feature extraction capabilities, they are typically designed to optimize performance on a single output vector, which can be suboptimal for tasks requiring the prediction of multiple interrelated attributes simultaneously.

Other Multi-Task Learning Frameworks

Multi-task learning (MTL) frameworks have been developed to improve learning efficiency and prediction accuracy across related tasks. For instance, some studies have utilized an MTL approach to simultaneously detect the make, model, and type of vehicles but often with separate dedicated branches for each task without the integration of shared feature layers that inform each task directly. While these models leverage some benefits of MTL, they may not fully capitalize on the potential synergies between closely related tasks as effectively as an integrated model like AutoSpecNet.

Why AutoSpecNet?

The decision to use AutoSpecNet, building on MobileNetV2 within a multi-task learning framework, was driven by the need for a lightweight model that can operate efficiently in real-time applications while still providing the robustness required for accurate multi-attribute classification. MobileNetV2's architecture offers an excellent balance of computational efficiency and feature extraction capability, which is crucial for deployment in resource-constrained environments. Furthermore, by customizing this base model to support multi-task learning specifically for car attributes, AutoSpecNet allows for shared learning across tasks, thereby enhancing the model's generalization capabilities across diverse datasets.

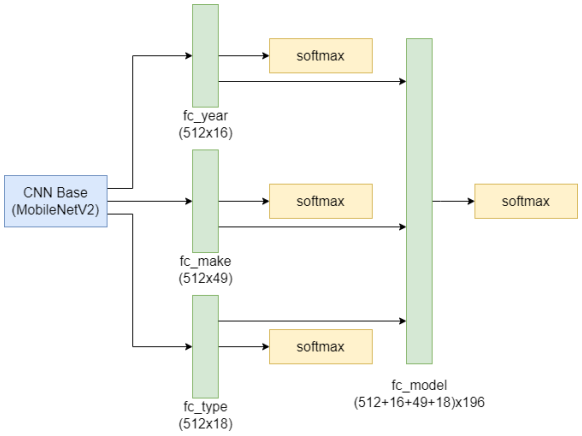

Figure 5: The Architecture of AutoSpecNet

Project Description

This section provides a detailed account of the methodologies and technical innovations employed in the AutoSpecNet project, including the dataset used, the model architecture, and the training procedures.

Dataset

AutoSpecNet utilizes The Stanford Cars Dataset (Li 2018) for training and validation. This Dataset has a collection of 16,185 images with labels such as "2012 BMW M3 Coupe", which combines year, make, model, and type in a single string. A key preprocessing step involves parsing the labels to separate the year, make, and type into distinct categories, enabling the system to perform multi-task learning effectively.

Model

Architecture: AutoSpecNet is based on the MobileNetV2 architecture, which is modified to support multi-task learning. The network is adapted to predict multiple attributes (year, make, and model) from a single forward pass. (Fig. 5) After

the feature extraction layers of MobileNetV2, three separate fully connected layers branch out to predict each attribute:

1. Year Head: A fully connected layer that predicts the manufacture year of the car.
2. Make Head: A fully connected layer that predicts the make of the car.
3. Type Head: A fully connected layer that predicts the model of the car.

These heads allow for task-specific processing, while the shared MobileNetV2 backbone ensures that the general features extracted are effectively utilized across all tasks.

Training

Optimization The model is trained using Stochastic Gradient Descent (SGD) (Ruder 2016) with a learning rate of 0.01, momentum of 0.9, and weight decay of 0.0001. This setup helps in mitigating the risk of overfitting and ensures smooth convergence.

Loss Function

$$\begin{aligned} TotalLoss = Loss_{main} &+ \lambda_{year} * Loss_{year} \\ &+ \lambda_{make} * Loss_{make} \\ &+ \lambda_{type} * Loss_{type} \end{aligned}$$

A composite loss function is used to train AutoSpecNet, which is a weighted sum of the cross-entropy losses for each task. The losses for the year, make, and model predictions are weighted equally ($\lambda_{year} = \lambda_{make} = \lambda_{type} = 0.1$) in the initial configurations to balance their contributions to the overall model performance.

Scheduling and Metrics A learning rate scheduler reduces the learning rate by a factor of 10 at predetermined epochs to fine-tune the training process. Accuracy, precision, and recall are computed for each attribute to monitor the training progress and evaluate the model's performance.

Validation The model is periodically evaluated on a held-out validation set to monitor its generalization capabilities. The best-performing model on the validation set is saved for final testing and real-world application.

Experiments

The experimental design of the AutoSpecNet project was structured to evaluate the model's performance across various operational settings, including changes in image size, batch size, and learning rates. These variables were systematically varied to observe their impact on the model's accuracy in classifying cars by year, make, and type. Below, we detail the setup for each experiment, the rationale behind the choices, and the observed outcomes.

Image Size Variation

Setup and Rationale Two different image sizes were tested: 224x224 pixels and 400x400 pixels. The choice of these sizes was driven by the need to understand the trade-off between computational efficiency and classification accuracy. Smaller images are faster to process but might lose critical details necessary for fine-grained classification tasks.

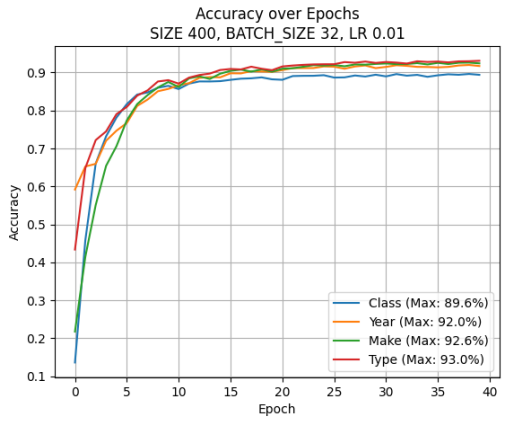

Figure 6: BEST RESULT: Accuracy over Epochs (Image Size 400, Batch Size 32, Learning Rate 0.01)

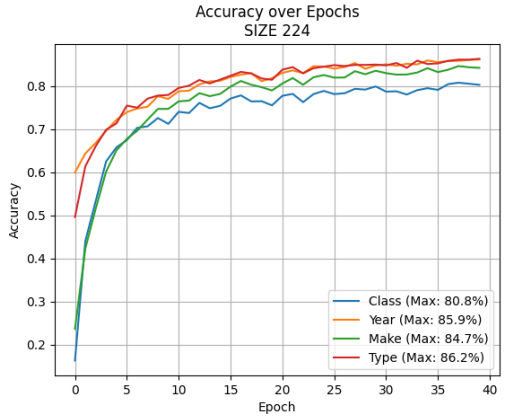

Figure 7: Accuracy over Epochs (Image Size 224, Batch Size 32, Learning Rate 0.01)

Results Graphs (Fig. 6 and Fig. 7) showing the accuracy of the model with different image sizes indicate that the larger image size (400x400) yielded better performance ($> 92\%$ Accuracy). This improvement is likely due to the preservation of more detailed features in larger images, which are crucial for distinguishing between closely similar car models.

Batch Size Variation

Setup and Rationale Experiments were conducted with batch sizes of 16, 32, and 64, all using an image size of 224 pixels. These sizes were selected to investigate how the batch size affects the model's learning dynamics and its stability during training.

Results Accuracy graphs (Fig. 8, Fig. 7, and Fig. 9) for different batch sizes show that a medium batch size of 32 offers the best compromise between training stability and model performance. Smaller batch sizes tended to exhibit higher variance in training accuracy, whereas larger batches compromised the model's ability to generalize effectively

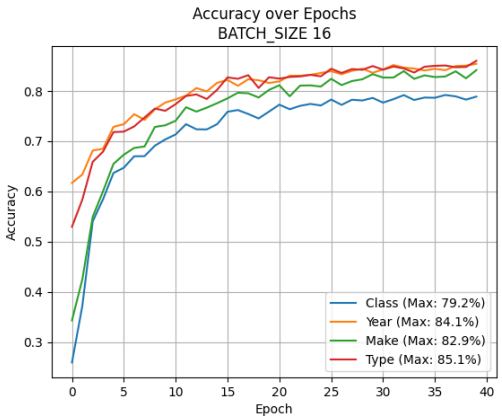

Figure 8: Accuracy over Epochs (Image Size 224, Batch Size 16, Learning Rate 0.01)

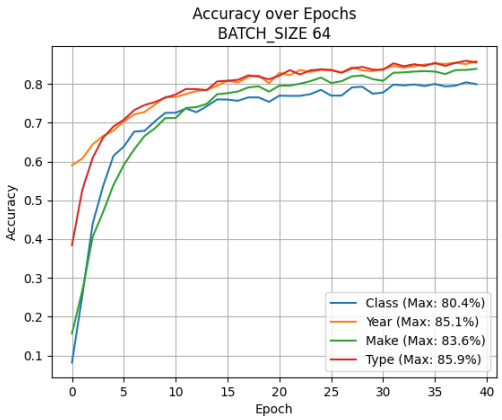

Figure 9: Accuracy over Epochs (Image Size 224, Batch Size 64, Learning Rate 0.01)

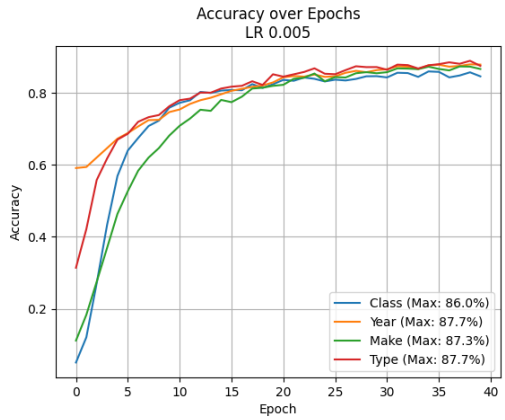

Figure 10: Accuracy over Epochs (Image Size 224, Batch Size 32, Learning Rate 0.005)

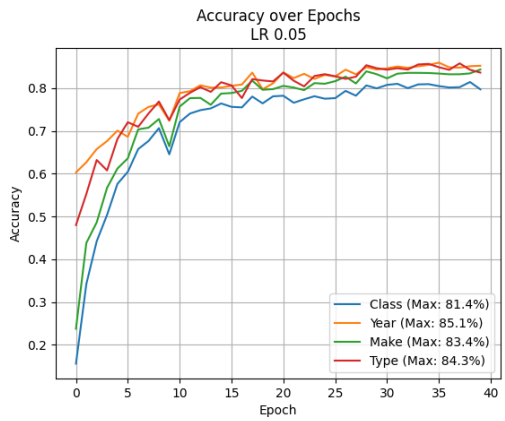

Figure 11: Accuracy over Epochs (Image Size 224, Batch Size 32, Learning Rate 0.05)

from the training data.

Learning Rate Variation

Setup and Rationale Learning rates of 0.005, 0.01, and 0.05 were tested, all on an image size of 400 pixels. The objective was to determine the optimal learning rate that provides the fastest convergence without overshooting the minimal loss.

Results The results (Fig. 10, Fig. 6, and Fig. 11) from varying the learning rates demonstrate that a learning rate of 0.01 struck the best balance between rapid convergence and training stability. Higher rates led to potential overshooting of minimal loss points, while lower rates slowed down the training process unnecessarily.

General Observations

Performance Analysis The experiments collectively highlight the sensitivity of the AutoSpecNet model to training hyperparameters and input specifications. The model performs best with larger image sizes, which facilitate the capture of detailed features essential for accurate classification across the nuanced categories of cars.

Limitations and Failures The model shows limitations under conditions of smaller image sizes and inappropriate batch or learning rate settings, where it fails to either capture sufficient detail or adjust adequately to the nuances in the data. These failures underline the importance of carefully chosen hyperparameters in training effective deep learning models for specific tasks like fine-grained classification.

Conclusion

The AutoSpecNet project aimed to develop a robust multi-task learning model capable of accurately classifying cars by year, make, and type from images, utilizing the efficient architecture of MobileNetV2. This project not only demonstrated the feasibility of using a modified MobileNetV2 for multi-task learning but also provided insights into how various training parameters influence model performance.

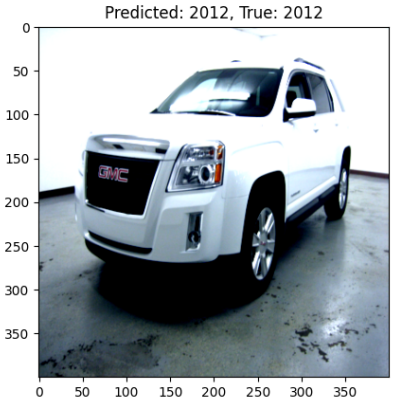

Figure 12: An example of predicting the year of a car successfully

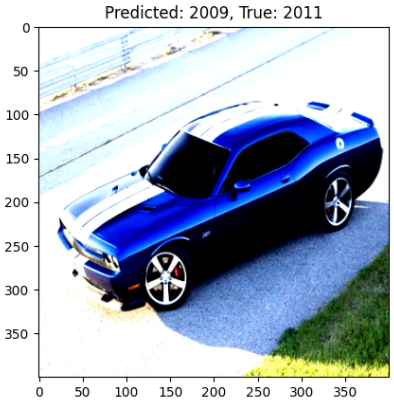

Figure 13: An example of predicting the year of a car unsuccessfully

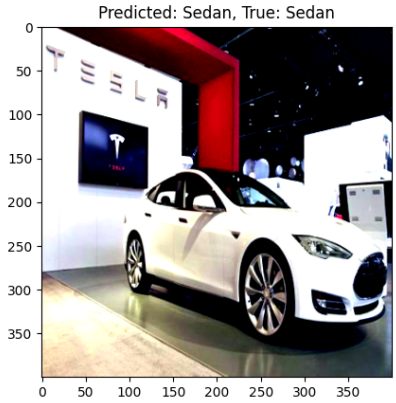

Figure 14: An example of predicting the type of a car successfully

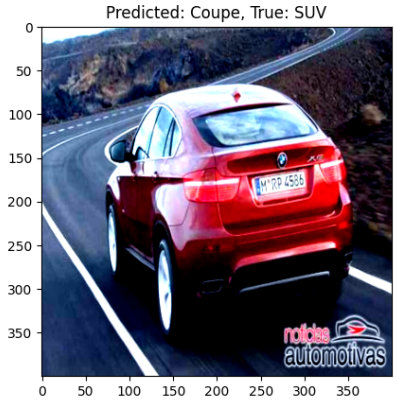

Figure 15: An example of predicting the type of a car unsuccessfully

Key Findings

1. **Image Size Impact**: The experiments confirmed that larger image sizes (400x400 pixels) enhance the model's ability to capture detailed features necessary for fine-grained classification tasks, leading to higher accuracy.
2. **Optimal Batch Size**: A batch size of 32 was found to be optimal, balancing the need for computational efficiency while maintaining the stability and generalization capability of the model.
3. **Learning Rate Optimization**: A learning rate of 0.01 was most effective, offering a good compromise between fast convergence and avoiding the overshooting of the loss minimum.

Insights and Lessons Learned

- **Multi-Task Learning Efficiency**: Integrating tasks within a single model structure (AutoSpecNet) using shared features but distinct classification heads proved to be an efficient strategy for handling multiple related classification tasks.
- **Parameter Sensitivity**: The project highlighted the sensitivity of deep learning models to hyperparameter settings, emphasizing the need for careful tuning, especially in applications involving fine-grained image classifications.
- **Practical Implications**: The results underscore the potential for deploying lightweight, efficient models like AutoSpecNet in real-world applications where resources are constrained but high accuracy is required.

Future Directions

Continuing from this project, future work could explore the integration of additional contextual data (e.g., environmental conditions, vehicle settings) to further enhance the accuracy and robustness of the classification. Moreover, expanding the dataset to include a wider variety of vehicle types and

conditions could help in testing the scalability and adaptability of the model.

In conclusion, the AutoSpecNet project not only achieved its goal of developing an effective multi-task learning model for car classification but also provided valuable insights into the dynamics of model training and performance optimization. These learnings pave the way for further innovations in the field of machine learning and computer vision.

References

Cervantes, J.; Garcia-Lamont, F.; Rodríguez-Mazahua, L.; and Lopez, A. 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408: 189–215.

Crawshaw, M. 2020. Multi-Task Learning with Deep Neural Networks: A Survey. arXiv:2009.09796.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C.; Bottou, L.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Li, J. 2018. Stanford Cars Dataset. Accessed: 22 Feb. 2024.

Louppe, G. 2015. Understanding Random Forests: From Theory to Practice. arXiv:1407.7502.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.

Sandler, M.; Howard, A. G.; Zhu, M.; Zhmoginov, A.; and Chen, L. 2018. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. *CoRR*, abs/1801.04381.

Sharma, N.; Jain, V.; and Mishra, A. 2018. An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*, 132: 377–384. International Conference on Computational Intelligence and Data Science.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.