The Road of Pixels: Image Classification with Machine Learning Techniques

Alex Prado
alxprado@ucdavis.edu

Anthony Tran
ahutran@ucdavis.edu

Ryan Tan
rhtan@ucdavis.edu

Lena Ray
lenray@ucdavis.edu

Jen Galicia
jfgalicia@ucdavis.edu

https://github.com/Cauchy-Schwarz/ECS171

Abstract

This study addresses the critical challenge of vehicle image classification, with wide-ranging applications in law enforcement, traffic management, insurance claims processing, and environmental impact assessment. The goal of the project was to develop an accurate image classification model capable of correctly identifying the make, model, and year of a vehicle from a single image. The study leverages the Stanford cars dataset, consisting of 16,185 images spread across 196 distinct classes, each representing a unique combination of make, model, and year. Our approach employs EfficientNetV2-Large, a model with 24 million parameters, trained and optimized under hardware limitations with an Nvidia RTX 3060, contrasting with superior hardware resources available to other groups. The model training incorporates hyperparameters such as an image size of 224x224, a learning rate of 0.001, a batch size of 16, momentum of 0.9, and 30 epochs. The utilization of compound model scaling, fused convolutional layers, and fine-tuning of the classifier layer has aided in optimization. The preprocessing techniques applied to enhance the model's performance include resizing, random transformations, and color normalization. Despite the constraints of limited hardware resources, the model achieves promising results, with a final test accuracy of 88% and training accuracy of 94% within a training time of 2 hours. This performance demonstrates the effectiveness of transfer learning in training image classification models, even with challenging datasets, and serves as a strong foundation for further enhancements.

1 Introduction

Vehicle identification plays a significant role in various domains, including the automobile industry, law enforcement, insurance and claims processing, and even traffic management and surveillance. The ability to accurately assign a car's make, model, and year based on images presents a challenging problem that requires advanced imaging processing techniques. Realistically, many of the uses of vehicle identification listed above would rely on CCTV frames or other low-quality images of cars, varying in angle, number of cars in the frame, and other limitations that would make such a model subject to error. If a reliable vehicle identification model is successfully built, it can be applied to various fields, as aforementioned:

1.1 Law enforcement

Vehicle identification can assist with law enforcement agencies in combating crimes such as vehicle theft, smuggling, and unauthorized vehicle usage. A reliable, accurate vehicle identification model can aid in tracking potential suspects, enhancing surveillance capabilities, and improving overall security models.

1.2 Traffic Management and Urban Planning

Vehicle identification can contribute to traffic management and urban planning by providing data on vehicle types and their traffic flow and traffic patterns. This information could be useful for optimizing traffic signal timings, designing road networks, and allocating resources for infrastructure development. Furthermore, vehicle identification contributes to the development of smart city initiatives.

1.3 Insurance and Claims Processing

Vehicle identification can accurately provide the identification of vehicles in accidents or insurance claims prevents fraudulent activities and expediting claim settlements.

1.4 Environmental Impact Assessment

Vehicle identification can be utilized to assess the impact of certain vehicles on air quality, noise pollution, and carbon emissions. By accurately identifying vehicle types, it helps researchers develop effective mitigation strategies.

These are a few examples that highlight the wide range of domains where vehicle identification can have significant applications. By developing a reliable vehicle identification model, it creates the potential to revolutionize these fields and create opportunities for innovation and advancements.

2 Literature Review

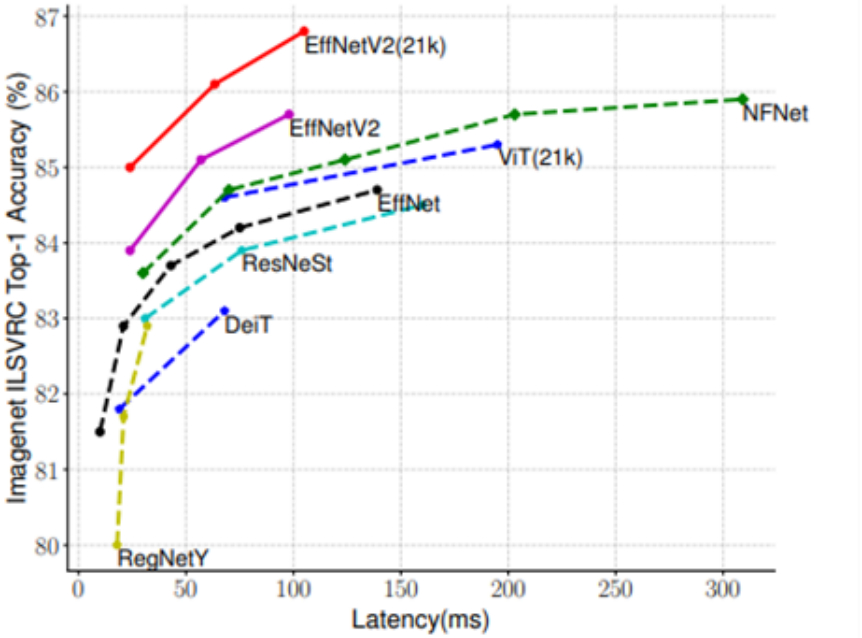
Recent image classification began with a major breakthrough in image classification with the advent of deep learning and Convolutional Neural Networks (CNNs) [3]. The introduction of the LeNet-5 model by LeCun et al. in 1998 heralded the potential of CNNs for digit recognition [4]. But it was the triumph of the AlexNet model by Krizhevsky et al. in the ImageNet competition in 2012 that created a revolution in the field [5]. Post-AlexNet, the image classification field saw novel architectures aiming to enhance performance. Notable ones include VGGNet by Simonyan and Zisserman (2014) [6], GoogLeNet (Inception) by Szegedy et al. (2015) [7], ResNet by He et al. (2016) [8], and DenseNet by Huang et al. (2017) [9]. These architectures introduced game-changing concepts such as deeper networks, skip connections, and inception modules. As the field has matured, transfer learning has emerged as a powerful technique. Leveraging pretrained models on large-scale datasets like ImageNet allowed for the fine-tuning of specific tasks even when the availability of labeled data was limited [10]. An important contribution in this area was the paper "Decaf: A Deep Convolutional Activation Feature for Generic Visual Recognition" by Donahue et al. [11]. Stepping into the present, considerations of efficiency and computational constraints have become critical in image classification. To strike a balance between accuracy and computational efficiency, architectures like MobileNet by Howard et al. (2017) [12] and EfficientNet by Tan and Le (2019) [13] have been developed, making image classification models more deployable on devices with limited resources.

3 Dataset Description

The Stanford cars dataset utilized in this project comprises 16,185 images, which are divided into 8,144 training images and 8,041 testing images, ensuring a roughly equal split for each class. These images are annotated with bounding boxes and labels, representing 196 distinct classes that correspond to specific combinations of make, model, and year (e.g., 2012 Tesla Model S). Spanning the time period up to 2013, the dataset encompasses a range of image qualities, varying from poorly captured images on phone cameras to professionally taken high-resolution photographs. While each image guarantees the presence of at least one vehicle, multiple vehicles in the background are also possible. It is worth noting that some images may only capture partial car views, while others may exhibit noise, such as blurriness from a dirty lens or grain resulting from high ISO settings. Meticulously curated by the Stanford University AI Lab, this dataset

collection, aptly titled "Cars," offers a comprehensive resource for diverse applications in the field. It is important to acknowledge that the released version of the dataset includes 196 categories, which is one less than the initial report due to post-publication cleanup. However, the dataset remains highly reliable, and the results derived from it closely align with those discussed in the original paper [18].

4 Proposed Methodologies



The task of creating a proficient image classification model is complicated, requiring not only an extensive comprehension of Convolutional Neural Networks (CNNs) but also a substantial understanding of their theoretical foundations and practical applications on top of properly selecting optimal hyperparameters, making sound architectural decisions, and implementing effective regularization techniques that often requires extensive experimentation. Another important obstacle to the development of these complex models is the issue of hardware limitations. Our team tried using Google Colab, an online platform offering GPU access, which operates on a credit system that was quickly depleted when training machine learning models. The best hardware at our disposal, an Nvidia RTX 3060, has 3.5k CUDA cores and 12GB VRAM, falls short of the computational demands of machine learning models on expansive datasets. To overcome those limitations, the idea of employing pre-trained models became a compelling and a necessary proposition. The use of such pre-existing models allowed us, a group of students of machine learning, to achieve great results and eliminate the need to grasp and know every detail of the model's architecture via transfer learning, and obtain high-end computational resources. Various models including EfficientNet, VGG16, ResNet50, and ResNet34 were evaluated. Ultimately, EfficientNet emerged as the best choice due to its ability of maintaining an equilibrium between accuracy and computational consumption. It also demonstrates superior efficiency by utilizing fewer parameters than the aforementioned models (the exception being ResNet34), which often results in faster training times and less memory usage. VGG16, despite being recommended for its power and accuracy, is a relatively large model encompassing 138 million parameters, making it impractical for our limited computational resources. ResNet50, a variation of the Residual Network architecture, enjoys popularity in classification tasks owing to its structural depth and incorporation of skip connections that correct vanishing gradient problems. However, it stills require more computational resources than EfficientNet. This particularly model was chosen at first, however, during training, the model was still running for

5 hours and just achieving a 33% accuracy. ResNet34, a more streamlined version of ResNet50, provides greater efficiency in terms of resource computational usage but it falls short in accomplishing complex tasks compared to its more extensive counterparts. The picture above compares all the models' accuracy.

4.1 EfficientNet:

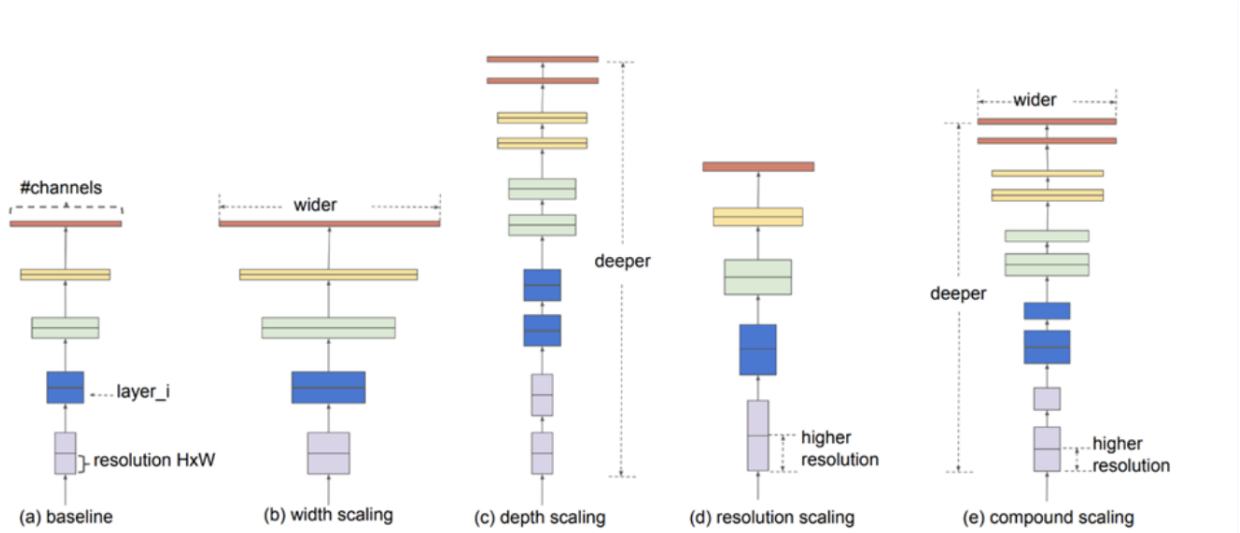
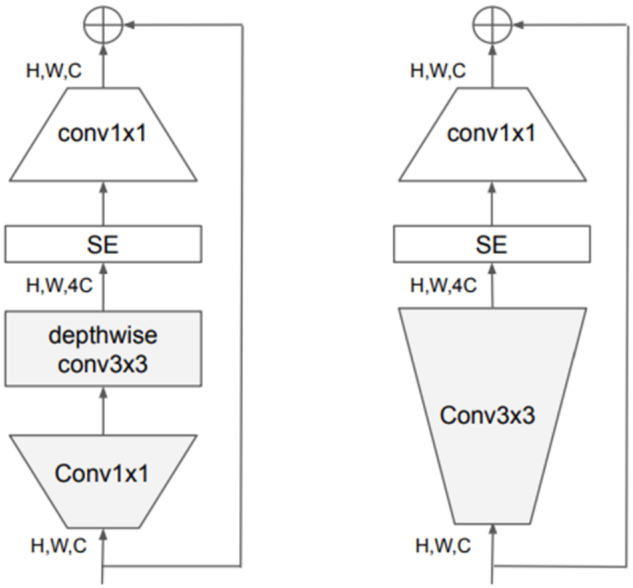


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

The key principle of EfficientNet lies in its methodology of uniformly scaling all dimensions of depth, width, and resolution of the network using a compound coefficient as pictured above [13]. EfficientNet's architecture embodies a list of critical optimizations: Compound Model Scaling and Fused Convolutional Layers[13]. Compound Model Scaling allows the network to balance the scaling of depth (the number of layers), width (the number of neurons in a layer), and resolution (the input image size). This helps improve the network's performance while avoiding the suboptimal outcomes associated with scaling only one of these dimensions. Fused Convolutional Layers combine the convolution operation and the activation function into a single step, reducing the memory usage and potentially speeding up the computations [13]. EfficientNet also utilizes Depthwise convolutions and MBConv to enhance the efficiency of CNNs by reducing computational complexity while maintaining performance [13]. MBConv, or Mobile Inverted Bottleneck Convolution, is an inverted residual block with a bottleneck. The whole point of this block is to expand the number of channels, apply depthwise convolution, and then compresses the channels again, allowing the model to learn more complex representations while reducing the number of parameters and computational cost.

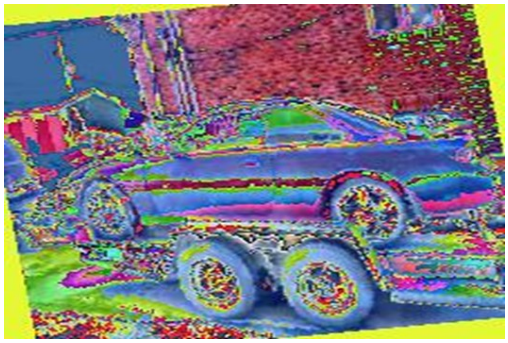


The Neural Architecture Search (NAS) framework used by EfficientNet optimizes accuracy, parameter efficiency, and training[13]. Its design choices involve MBConv and Fused-MBConv operation types as previously mentioned, and presented in the image above. With a smaller search space, larger network searches can be executed, and up to 1000 models can be sampled, each trained for about 10 epochs with reduced image sizes[13] [17]. The search reward integrates model accuracy (A), normalized training step time (S), and parameter size (P), using a weighted product $A * S^w * P^v$, where w =-0.07 and v=-0.05, were empirically determined weights to balance trade-offs[17]. The specific version of EfficientNet used for the image classification task was the EfficientNetV2-Large model with 24 Million parameters[17]. The model was compiled using the ADAM optimizer for its capacity to compute adaptive learning rates for different parameters, promoting more efficient training. This method, known as Adaptive Moment Estimation (ADAM), uses an exponentially decaying average of past squared gradients and an exponentially decaying average of past gradients to compute the adaptive learning rate for each weight in the network, mitigating issues with learning rates [17]. The final model was equipped with a softmax activation to output probability predictions among classes, which is preferable for multi-class classification problems [17]. The softmax function outputs a vector that represents the probability distribution of a list of potential outcomes, making it a good fit for the classification tasks.

5 Model Implementation

In our pursuit to build a model that effectively learns and generalizes, the dataset was divided into training, validation, and test subsets in a 70:15:15 ratio, ensuring a diverse and representative sample for each phase of our methodology. The preprocessing stage of the data involved leveraging techniques like rotation, zooming, and horizontal flipping to augment the images. These augmentations enhance the model's capacity to generalize by exposing it to varied imaging conditions akin to real-world scenarios. As an example, the two pictures below offer a contrasting experience of how we humans visually recognize objects. To standardize the dataset and to facilitate efficient learning, normalization of pixel intensities was conducted based on the mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] statistics derived from the ImageNet dataset, which is a common way of working around computing the actual mean of a data set with limited computational resources. Normalization mitigates potential issues associated with gradient explosion or vanishing gradient during model training. With respect to the model architecture, as previously said, we

((a)) Human eye



((b)) Computer Visualization

Figure 1

adapted EfficientNet while retaining all the weights except for the 'classifier' (or 'fc') layer, which was unfrozen and reconfigured to output 196 classes, aligning with the number of car classes in our dataset. The model was trained for 30 epochs, using a batch size of 16, a learning rate of 0.001, and a momentum of 0.9, with the training process taking approximately 2 hours. Our input images were resized to 224x224 to match the default input size of EfficientNet. The model's effectiveness was assessed using the test set, with the evaluation metrics comprising accuracy, precision, recall, and F1-score for each car class. The model achieved an overall accuracy of 0.88, signifying it correctly classified 88% of the images in the test set. Moreover, the weighted averages of precision, recall, and F1-score were also commendable, at 0.89, 0.88, and 0.88 respectively.

5.1 Experimental Results

Table 1: Classification Report

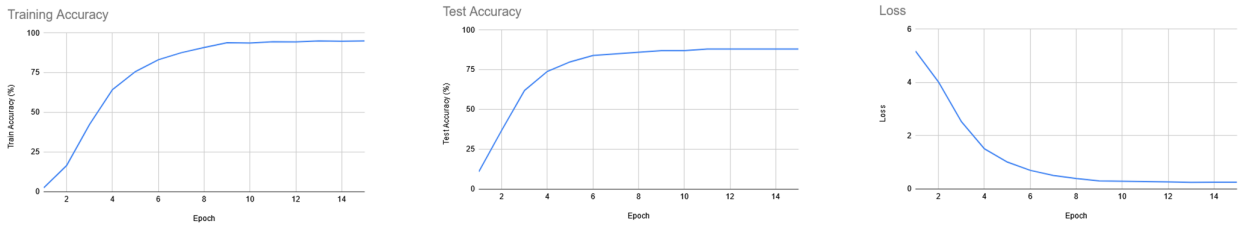
((a))

Class Name	Precision	Recall	F1-Score
AM General Hummer SUV 2000	0.95	0.95	0.95
Acura Integra Type R 2001	0.89	0.93	0.91
Acura RL Sedan 2012	0.62	0.75	0.68
Acura TL Sedan 2012	0.83	0.93	0.88
Acura TL Type-S 2008	0.91	0.95	0.93
Acura TSX Sedan 2012	0.97	0.78	0.86
Acura ZDX Hatchback 2012	0.90	0.92	0.91
Aston Martin V8 Vantage Convertible 2012	0.78	0.56	0.65
Aston Martin V8 Vantage Coupe 2012	0.78	0.76	0.77
Aston Martin Virage Convertible 2012	0.85	0.85	0.85
Aston Martin Virage Coupe 2012	0.87	0.87	0.87
Audi 100 Sedan 1994	0.69	0.85	0.76
Audi 100 Wagon 1994	0.88	0.88	0.88
Audi A5 Coupe 2012	0.67	0.83	0.74
Audi R8 Coupe 2012	0.95	0.88	0.92
Audi RS 4 Convertible 2008	0.83	0.83	0.83
Audi S4 Sedan 2007	0.87	0.91	0.89
Audi S4 Sedan 2012	0.79	0.69	0.74
Audi S5 Convertible 2012	0.84	0.86	0.85
Audi S5 Coupe 2012	0.61	0.55	0.57
Audi S6 Sedan 2011	0.86	0.91	0.88
Audi TT Hatchback 2011	0.54	0.80	0.65
Audi TT RS Coupe 2012	0.86	0.64	0.74
Audi TTS Coupe 2012	0.80	0.67	0.73
Audi V8 Sedan 1994	0.81	0.67	0.73
BMW 1 Series Convertible 2012	1.00	0.83	0.91
BMW 1 Series Coupe 2012	0.83	0.95	0.89
BMW 3 Series Sedan 2012	0.89	0.79	0.84
BMW 3 Series Wagon 2012	0.80	0.88	0.84
BMW 6 Series Convertible 2007	0.92	0.75	0.83
BMW ActiveHybrid 5 Sedan 2012	0.81	1.00	0.89
BMW M3 Coupe 2012	0.86	0.86	0.86
BMW M5 Sedan 2010	0.84	0.88	0.86
BMW M6 Convertible 2010	0.69	0.83	0.76
BMW X3 SUV 2012	0.90	0.97	0.94
BMW X5 SUV 2007	0.91	0.95	0.93

((b))

Class Name	Precision	Recall	F1-Score
BMW X6 SUV 2012	0.97	0.86	0.91
BMW Z4 Convertible 2012	0.97	0.93	0.95
Bentley Arnage Sedan 2009	0.85	1.00	0.92
Bent. Cont. Flying Spur Sedan 2007	0.74	0.80	0.77
Bent. Cont. GT Coupe 2007	0.77	0.52	0.62
Bent. Cont. GT Coupe 2012	0.70	0.82	0.76
Bent. Cont. Supersports Conv. Conver.2012	0.91	0.86 0.89	
Bent. Mulsanne Sedan 2011	0.97	0.89	0.93
Bugatti Veyron 16.4 Convertible 2009	0.83	0.91	0.87
Bugatti Veyron 16.4 Coupe 2009	0.76	0.81	0.79
Buick Enclave SUV 2012	0.89	0.98	0.93
Buick Rainier SUV 2007	0.86	0.88	0.87
Buick Regal GS 2012	0.85	0.94	0.89
Buick Verano Sedan 2012	0.90	0.95	0.92
Cadillac CTS-V Sedan 2012	0.96	1.00	0.98
Cadillac Escalade EXT Crew Cab 2007	0.86	0.95	0.90
Cadillac SRX SUV 2012	1.00	1.00	1.00
Chevrolet Avalanche Crew Cab 2012	0.85	0.87	0.86
Chev. Camaro Convertible 2012	0.95	0.82	0.88
Chev. Cobalt SS 2010	0.97	0.83	0.89
Chev. Corvette Convertible 2012	0.89	0.87	0.88
Chev. Corvette Ron Fellows Edition Z06 2007	0.85	0.92	0.88
Chev. Corvette ZR1 2012	0.86	0.93	0.90
Chev. Express Cargo Van 2007	0.53	0.62	0.57
Chev. Express Van 2007	0.62	0.46	0.52
Chev. HHR SS 2010	1.00	0.97	0.99
Chev. Impala Sedan 2007	0.95	0.88	0.92
Chev. Malibu Hybrid Sedan 2010	0.89	0.89	0.89
Chev. Malibu Sedan 2007	1.00	0.86	0.93
Chev. Monte Carlo Coupe 2007	0.89	0.91	0.90
Chev. Silverado 1500 Classic Ext. Cab 2007	0.89	0.95	0.92
Chev. Silverado 1500 Extended Cab 2012	0.65	0.70	0.67
Chev. Silverado 1500 Hybrid Crew Cab 2012	0.68	0.57	0.62
Chev. Silverado 1500 Regular Cab 2012	0.78	0.80	0.79
Chev. Silverado 2500HD Regular Cab 2012	0.77	0.86	0.81
Chev. Sonic Sedan 2012	0.81	0.81	0.81

The model exhibits varying performance across different car classes. Some classes, such as "FIAT 500 Abarth 2012," "Dodge Dakota Crew Cab 2010," "Dodge Magnum Wagon 2008," "Ford Fiesta Sedan 2012," "Hyundai Santa Fe SUV 2012," "Infiniti G Coupe IPL 2012," "Isuzu Ascender SUV 2008," "Mercedes-Benz 300-Class Convertible 1993," "Volkswagen Beetle Hatchback 2012," and "Volkswagen Golf Hatchback 2012," demonstrate high precision, recall, and F1-scores, all above 0.95. Conversely, certain models, such as "Dodge Caliber Wagon 2012," "Ferrari 458 Italia Coupe 2012," "Suzuki SX4 Sedan 2012," and "Toyota Camry Sedan 2012," have F1-scores below 0.8, indicating lower performance. This discrepancy suggests that the model has learned to accurately identify some car classes but encounters difficulties with others. The presence of class imbalance is evident as the number of samples (support) varies significantly across classes, ranging from 27 samples for "FIAT 500 Abarth 2012" to 68 samples for "GMC Savana Van 2012." This potential class imbalance might affect the model's performance, leading it to favor classes with more samples. Moreover, there appears to be a trade-off between precision and recall for certain classes. For instance, the "Dodge Caliber Wagon 2007" demonstrates high recall but lower precision, indicating the model tends to over-predict this class. Conversely, the "Ferrari FF Coupe 2012" exhibits high precision but lower recall, suggesting the model may under-predict this class. In addition, inconsistencies can be observed within the same class, where precision, recall, and F1-scores differ noticeably. For example, the "Dodge Charger SRT-8 2009" showcases a precision of 0.92, recall of 0.86, and an F1-score of 0.89. This variance implies that the model's performance varies across different evaluation metrics.



((a)) Training Accuracy ((b)) Test Accuracy ((c)) Loss

Figure 2: Performance Evaluation

6 Conclusion

In the future, we may want to acquire better hardware that would enable us to perform Grid Search effectively and to ensure proper testing of multiple models using different hyperparameters. Additionally, updating the dataset would be crucial in helping us achieve our motivation; that is, deploying a reliable vehicle identification model that could be used today. As mentioned before, our current dataset only includes cars from 2013 and before, which undermines the purpose and the relevance of our motivation. We would also like to try deploying our model on video hardware to test the model's ability to perform inference in real-time, as well as drawing bounding boxes around classified images. We may want to consider switching to Yolov5 or Yolov7 for this as they are known for their effectiveness in object detection tasks. For the optimization of our current model, to reduce model complexity, we propose performing manual or ML-assisted pruning of weights. Moreover, utilizing TensorRT can maximize the utilization of CUDA technology, further optimizing the model's performance. To conclude, it is safe to say that our project has successfully implemented transfer learning and achieved an acceptable accuracy on a challenging dataset given the materials taught throughout this course. Our final test accuracy stands at 88%, while the training accuracy is at 94%.

References

1. Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111-122.

2. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.

3. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

4. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 25, 1097-1105.

6. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

9. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).

10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 211-252.

11. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647-655).

12. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

13. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105-6114).

14. Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2 (ICCV '99)*. IEEE Computer Society, 1150â1157.

15. Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.

16. Turk, M., & Pentland, A. (1991). Face recognition using eigenfaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 586-591).

17. Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. In *International Conference on Machine Learning 2021*.

18. Krause, Jonathan, et al. "3D Object Representations for Fine-Grained Categorization." 4th IEEE Workshop on 3D Representation and Recognition, 2013, Sydney, Australia. Web. Retrieved from *https://www.kaggle.com/jessicali9530/stanford-cars-dataset*