Nadav Toledo – 209496009
Eyal Shubeli - 314967308

Assignment 1 Report – Practical Deep Learning Workshop

1) A.
   - Total number of training images: 8144
   - Total number of test images: 8040
   - Number of unique classes: 196

   B. <u>Sample Structure</u>

   Each sample in the dataset contains:

   Image file: High-resolution RGB image.
   Bounding box: (x_min, y_min, x_max, y_max) coordinates for cropping the car.
   Class ID: Label for the car model (1-196).

   <u>Preprocessing and Augmentation</u>

   Preprocessing: Cropping the images into the bounding box and resizing to a uniform size (e.g., 224x224).

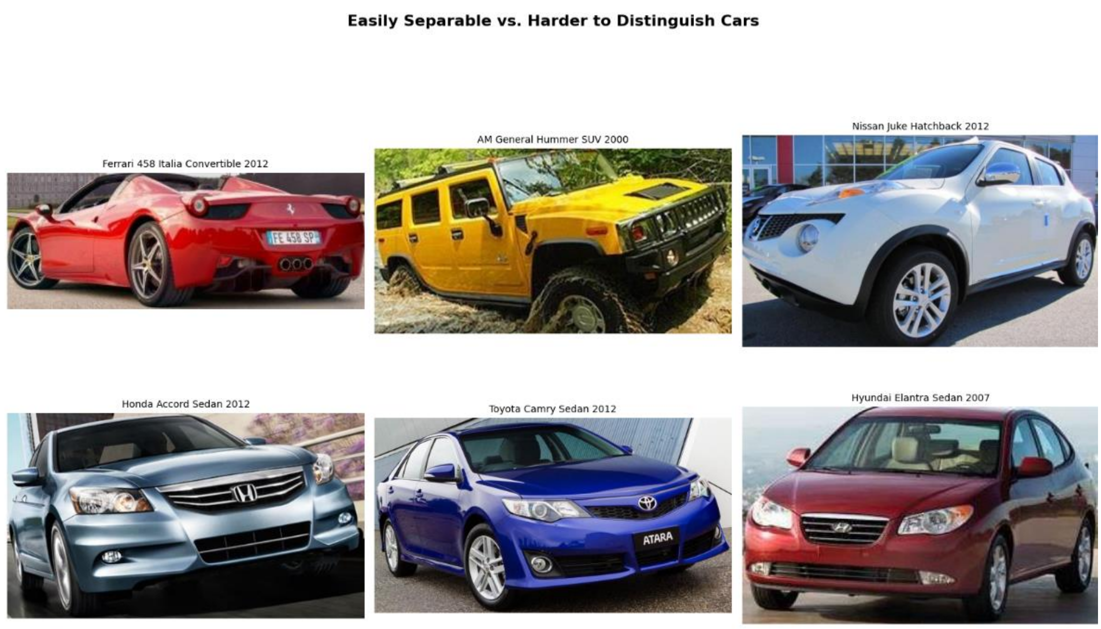
   Augmentation :
   Random rotations and flips.
   Color adjustments (brightness, contrast, hue).

   C. Min samples per class: 24, Max samples per class: 68, Average samples per class: 41.5, Median samples per class: 42.0

   D. The Stanford Cars dataset has been used in various methods for image classification. Benchmarks include:

   Convolutional Neural Networks (CNNs):
   ResNet-50: Accuracy ≈ 92.5%
   VGG-16: Accuracy ≈ 89.7%
   Vision Transformers (ViT):
   ViT-B/16: Accuracy ≈ 94.6%
   - Fine-tuning pretrained models have shown to significantly improve performance.

   E.



Nadav Toledo – 209496009
Eyal Shubeli - 314967308

2)

a.

Model	Time	Test Loss	Test Accuracy
Initial model	54 minutes	0.128	12.76%

b. The model's misclassifications likely stem from several factors. First, the dataset is large and diverse, making it challenging to capture all subtle distinctions between 196 classes effectively. Second, limited computational resources restrict the ability to train deeper architectures or fine-tune hyperparameters extensively, which could improve performance. Lastly, while the chosen architectures are robust, they may not be complex enough to fully model the intricate patterns in the data, especially when coupled with RGB inputs and fine-grained classification requirements. These constraints collectively impact the model's capacity to generalize effectively. Our suggestion for improvement:

1. A learning rate scheduler dynamically adjusts the learning rate during training, which can improve the ability of the model to generalize between identifying broader car categories and differentiating subtle details specific to vehicle types or models.

2. Gradient clipping stabilizes the training process by ensuring that gradients remain within a manageable range, which is particularly helpful when learning fine-grained distinctions between similar vehicle types or models, reducing noise and divergence.

3. A hierarchical CNN structure captures features in stages, starting with broad patterns (e.g., car shapes) in early layers and progressing to specific details (e.g., unique features of different models) in deeper layers, thereby improving the network's capacity to classify cars into types or models effectively.

c. We prioritized and decided to forgo the hierarchical CNN structure mainly because computing capabilities did not support it.

Model	Time	Test Loss	Test Accuracy
With learning rate scheduler	53 minutes	0.115	22.97%
With Gradient clipping	54 minutes	0.121	21.82%

Nadav Toledo – 209496009
Eyal Shubeli - 314967308

d.

Model	Time	Test Loss	Test Accuracy
With inference time Augmentation	59 minutes	0.1046	23.7%

e. During the training process, we deleted one category of vehicles and added it at this stage so that it appears for the first time now in both training and testing.

Model	Time	Test Loss	Test Accuracy
With extra category	55 minutes	0.115	23.33%

3)

5 Epoches

	Model Name	# Parameters	Validation Loss	Validation Accuracy (%)	Test Loss	Test Accuracy (%)	# Unique Correct Samples	# Unique Errors
0	ResNet-50	23909636	0.9400	75.45	0.8789	77.02	190	297
1	VGG-16	135063556	1.7591	52.73	1.7204	53.41	192	578
2	DenseNet-121	7154756	1.3922	73.48	1.3844	74.27	194	332
3	EfficientNet-B0	4258624	1.5058	64.33	1.4672	67.14	192	461

10 Epoches

	Model Name	# Parameters	Validation Loss	Validation Accuracy (%)	Test Loss	Test Accuracy (%)	# Unique Correct Samples	# Unique Errors
0	ResNet-50	23909636	0.747988	80.724371	0.716123	81.718692	195	254
1	VGG-16	135063556	1.944726	53.038674	1.878858	55.030469	188	598
2	DenseNet-121	7154756	0.949230	77.470841	0.906739	79.082204	196	301
3	EfficientNet-B0	4258624	0.874384	76.058932	0.857383	77.104838	195	304

Nadav Toledo – 209496009
Eyal Shubeli - 314967308

d.

The accuracy for the combination of Feature Extractor + Random Forest is reported as 21.18% - 23.39% which appears to be significantly lower than the results achieved with fine-tuned deep learning models.

e.

Model	Runtime (minutes)	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy	# parameters	Main Changes
ResNet-50	23	0.75	80.72	0.71	81.71	$24 \cdot 10^6$	
VGG-16	35	1.94	53.04	1.87	55.03	$135 \cdot 10^6$	
DenseNet-121	26	0.95	77.47	0.91	79.08	$7 \cdot 10^6$	
EfficientNet	22	0.87	76.06	0.86	77.1	$4 \cdot 10^6$	
Feature Extractor+ RF	3	7.6355	21.18	7	23.39	$25 \cdot 10^6$	

```
Feature Exractor with Random Forest
Runtime (s): 169.04
Validation Loss: 7.6355
Validation Accuracy (%): 21.18
Test Loss: 7.0027
Test Accuracy (%): 23.39
# Parameters in Feature Extractor: 25557032
```

4) This project provided valuable insights into the process of building and fine-tuning neural networks. By implementing various architectures and adapting them for car classification tasks, we deepened our understanding of model design, training, and optimization. Through hands-on coding, experimentation, and iterative improvement, we enhanced our technical skills and problem-solving abilities. Additionally, extensive reading and research on state-of-the-art methods enriched our knowledge and experience, enabling us to approach challenges more effectively and critically evaluate the performance of different approaches. This comprehensive learning process has been instrumental in advancing our expertise in deep learning and practical application.

This project also presented numerous challenges that pushed us to develop creative solutions. Classifying images into 196 different categories using RGB inputs required a deep understanding of the dataset and careful model design. We experimented with improvements like L2 regularization and gradient clipping to stabilize training and enhance performance. Additionally, we considered creating a hierarchical network structure for stage-wise classification, which added complexity but showed promise. Throughout the process, we built and compared three different networks, meticulously analyzing their performance at each stage to understand their behavior and identify areas for improvement. These efforts deepened our understanding of model development and evaluation, even as we navigated computational limitations. We remain confident that with better resources, our results could be significantly refined.