Vehicle Type Classification — Project Report

GROUP MEMBER:

BILAL KHAN

IZHAR ULLAH

Abstract

This project develops an image classifier to recognize four vehicle types: **Bus, Car, Truck, Motorcycle**. Using the Kaggle dataset, a convolutional neural network (CNN) model was trained and evaluated. The final test confusion matrix indicates 79 correct predictions out of 80 test images (one Car misclassified as Truck), yielding a test set accuracy of **98.75%**. Training and validation accuracy curves over 22 epochs show stable performance with validation accuracy around 95–97%. The report summarizes dataset preparation, model training, evaluation, discussion, and suggested future work.

1. Problem statement

Classify images of vehicles into four categories — Bus, Car, Truck, Motorcycle — to support traffic monitoring, automated tolling, and intelligent transportation systems.

2. Dataset

- Source: Kaggle — *Vehicle Type Recognition* (contains images labeled into 4 classes: Car, Truck, Bus, Motorcycle). (Kaggle)
- The Kaggle page and associated notebook show the dataset is curated for teaching/benchmarking and contains images for the four classes; the dataset piece

used for testing appears to include 20 test images per class (total 80), as inferred from the confusion matrix counts in your figure. (Kaggle)

- Typical preprocessing steps (applied in most notebooks of this type): resizing images to a fixed size (e.g., 224×224), normalization (scaling pixel values to [0,1] or mean-std normalization), shuffling, and splitting into train/validation/test sets.

3. Method

1. **Data loading & splitting**
   a. Split dataset into training, validation and test sets. (From the plots we observed training and validation curves and a separate confusion matrix for test.)
2. **Data augmentation** (commonly used)
   a. Random flips/rotations, brightness/zoom jittering, to increase robustness.
3. **Model architecture**
   a. Typical choices: a custom CNN or fine-tuned pre-trained model (MobileNetV2 / ResNet50 / VGG). The training/val learning curves are consistent with transfer-learning setups (fast convergence). If you used transfer learning, common components are: pre-trained backbone, global pooling, a small dense head, softmax for 4 classes.
4. **Training setup**
   a. Loss function: categorical cross-entropy.
   b. Optimizer: commonly Adam or SGD with momentum.
   c. Number of epochs: **22 epochs** (the training curve shows epochs 0–21).
   d. Metrics tracked: training accuracy and validation accuracy (plotted).
5. **Evaluation**
   a. Final evaluation on held-out test set — confusion matrix plotted (counts provided).

4. Results

Confusion Matrix

- Counts read directly from the provided confusion matrix figure:

- **Bus** — predicted Bus: 20 / 20
  - **Car** — predicted Car: 19; misclassified as Truck: 1
  - **Truck** — predicted Truck: 20 / 20
  - **Motorcycle** — predicted Motorcycle: 20 / 20
- Total test samples = 80 (20 × 4).
- **Test accuracy** = (20 + 19 + 20 + 20) / 80 = **79 / 80 = 0.9875 = 98.75%**.

Training / Validation Accuracy Curve

- Epochs: 0 → 21 (22 epochs total).
- Training accuracy starts low (~0.61 at epoch 0 in the plot), quickly rises and stabilizes near ~0.95–0.97.
- Validation accuracy is steady around 0.94–0.97, with no dramatic overfitting visible; there is a single dip in training accuracy around epoch 11–12 (train briefly drops to ~0.83) but recovers. Final train and val both ~0.95–0.96.

Key metrics summary

- Test accuracy (from confusion matrix): **98.75%**.
- Validation accuracy (final epoch, from plot): ~**95–97%**.
- Training: converged, with one transient drop but stable overall.

5. Interpretation & Discussion

- The model generalizes well on the held-out test set (98.75% accuracy). The confusion matrix shows only one misclassification (a Car labeled as Truck). That may be due to:
  - a visually ambiguous image (e.g., partial occlusion, perspective making a car look truck-like)
  - dataset bias or label noise
  - insufficient intra-class diversity during training
- The training and validation curves indicate successful learning; validation accuracy tracking training accuracy closely suggests minimal overfitting. The momentary drop in training accuracy (epoch ~11–12) could be due to a learning-rate change, augmentation randomness, or a temporary weight update instability — but since it recovers quickly, it is not concerning.

- Because this is a small dataset (the Kaggle dataset listing indicates a relatively small curated set), high test accuracy can sometimes indicate that the test set is similar to training images (limited domain shift). For deployment, evaluate on more diverse images (different weather, camera angles, occlusions) or extend the dataset.

6. Limitations

- The dataset is relatively small and curated; models trained on it may not generalize to traffic-camera imagery with different viewpoints and lighting.
- Without the full notebook contents I could not verify exact preprocessing, augmentation, or architecture details; the report used the plots and dataset page to infer outcomes.
- Class imbalance (not visible from the figures) could affect real-world performance if class distributions differ.

7. Recommendations / Future work

1. **Expand dataset**: include more images per class from different sources (surveillance cameras, phone photos, different geographies).
2. **Domain adaptation**: if deploying to traffic-cameras, fine-tune on a sample of target-camera images.
3. **Use object detection + classification**: detect vehicles in the image first (YOLO/SSD/Detectron), then classify cropped vehicles — improves robustness in complex scenes.
4. **Ensemble or compare models**: try ResNet50, MobileNetV2, EfficientNet-lite and compare. Transfer learning tends to be effective when data is limited.

8. literature review

Below is a concise literature summary and references relevant to vehicle classification and the dataset used.

Background & common approaches

- Classical vehicle classification moved from handcrafted features (HOG, SIFT) + SVM to deep learning approaches using CNNs, achieving substantially better accuracy and robustness. Transfer learning (fine-tuning pre-trained CNNs like ResNet, VGG, MobileNet) is widely used when labeled data is limited. (statworx.com)

Datasets

- **CompCars**: a large-scale car dataset for fine-grained classification with web and surveillance images; used widely in research for vehicle recognition tasks and benchmarking. The CompCars dataset demonstrates the benefits of large, diverse datasets for robust classification. (mmlab.ie.cuhk.edu.hk)
- **Kaggle Vehicle Type Recognition**: the dataset you used contains 4 coarse vehicle categories (Car, Truck, Bus, Motorcycle) and is suitable for learning image-level classifiers. The dataset is often used for learning/teaching projects. (Kaggle)

Surveys and recent work

- Recent surveys cover both detection and classification of vehicles using deep convolutional neural networks, summarizing methods, datasets, and performance trade-offs; they emphasize transfer learning, object detection pipelines, and domain adaptation. Examples: Berwo et al. (survey on deep learning techniques for vehicle detection/classification) and Tan et al. (review on AI systems for vehicle classification). These provide a useful overview of state-of-the-art practices and open challenges. (MDPI)

Transfer learning & fine-grained classification

- Transfer learning from ImageNet-trained backbones (ResNet, EfficientNet, MobileNet) is common and effective for vehicle recognition, especially when the dataset size is modest. Studies and demo notebooks show good practical performance when fine-tuning top layers or the entire backbone at a small learning rate. (statworx.com)

9. References

- Kaggle — Vehicle Type Recognition dataset (ASHWIN.S). (Kaggle)
- Berwo, M. A. et al., *Deep Learning Techniques for Vehicle Detection and Classification* (survey). (MDPI)
- CUHK Multimedia Lab, *CompCars Dataset*. (mmlab.ie.cuhk.edu.hk)
- Tan, S.H. et al., *Artificial intelligent systems for vehicle classification: A survey*. (ScienceDirect)
- Example transfer-learning guide: *Car Model Classification — Transfer Learning with ResNet* (blog). (statworx.com)