Vehicle Counting and Classification Using Computer Vision:

A Deep Learning Approach to Intelligent Transportation Systems

Birendra Khimding, Matt Hashemi, and Victor Hugo Germano

Shiley-Marcos School of Engineering

University of San Diego

AAI-521: Computer Vision

Final Project - Group 3

December 7, 2025

Table of Contents

Abstract

This project presents a comprehensive computer vision system for automated vehicle detection, classification, and traffic analysis using the UA-DETRAC dataset. We developed a custom convolutional neural network (CNN) achieving 99.7% validation accuracy across four vehicle classes (car, van, bus, others). The system processes 598,281 cropped vehicle images and demonstrates real-time processing capabilities (203.8 FPS). Traffic analytics reveal an 81:1 light-to-heavy motor vehicle ratio with short-term forecasting achieving R^2=0.947. Additional image restoration capabilities using pretrained Hugging Face models (Stable Diffusion, GAN colorization) demonstrate applicability to degraded surveillance footage. This work provides a scalable foundation for intelligent transportation systems, offering automated traffic monitoring, classification-based analytics, and predictive modeling for urban infrastructure planning.

Introduction

Traditional traffic monitoring relies on manual counting, pneumatic tubes, and inductive loop detectors—methods that are labor-intensive, expensive, and spatially limited. Computer vision systems transform existing surveillance cameras into intelligent sensors providing continuous, automated monitoring with richer data collection capabilities. A single camera with machine learning can replace multiple physical sensors while simultaneously collecting vehicle classifications, trajectories, speeds, and behavioral patterns (Wen et al., 2020).

This project addresses the challenge of automated vehicle classification and traffic analysis through deep learning. We developed an end-to-end pipeline encompassing dataset preprocessing, CNN model training, traffic analytics, and real-time video processing. Our system distinguishes between Light Motor Vehicles (LMV) and Heavy Motor Vehicles (HMV),

enabling infrastructure planning decisions such as differential toll structures, maintenance scheduling, and road reinforcement allocation based on traffic composition.

The modular architecture demonstrates software-defined adaptability: the same video stream currently classifying vehicle types can be extended to detect pedestrians, cyclists, traffic violations, and environmental conditions through software updates alone. This flexibility is essential for autonomous vehicle integration, micro-mobility tracking, and multi-modal transit coordination in evolving urban environments.

Dataset and Methodology

Dataset Selection

We utilized the UA-DETRAC (University at Albany DETection and tRACking) dataset, a large-scale traffic surveillance benchmark containing over 140,000 frames across 100 video sequences. The dataset provides XML annotations with bounding boxes and vehicle classifications for 8,250 vehicles, capturing diverse traffic conditions including weather variations, illumination changes, and multiple camera angles. Sequence MVI_20011 served as our primary analysis target, containing 664 frames at 540×960 resolution with 7,655 vehicle instances across four classes: cars (92.1%), vans (3.9%), others (2.8%), and buses (1.2%). This class imbalance presents realistic challenges for robust classifier development (see Appendix B, Figures 1-4 for dataset visualizations).

Preprocessing Pipeline

The preprocessing pipeline extracts individual vehicle crops from annotated frames for CNN training. Each annotated bounding box is cropped, resized to 64×64 pixels, and normalized to

[0,1] range. The 64×64 resolution balances computational efficiency with feature preservation, reducing memory footprint from ~230KB to ~4.8KB per crop while maintaining aspect ratio through center-crop strategy. RGB normalization improves gradient stability during training, following PyTorch best practices. String labels are converted to integer indices for efficient cross-entropy loss computation (see Appendix A, Code Sample 1 for implementation details). This process generated 598,281 total vehicle crops stored as compressed NumPy archives (.npz format). Quality assurance included random sample visualization to confirm proper cropping, label distribution verification matching original dataset statistics, and validation of no corrupted or zero-dimension crops. The dataset was split 80/20 for training (478,624 samples) and validation (119,657 samples) with fixed random seed for reproducibility.

Model Architecture and Training

CNN Architecture

The VehicleClassifier implements a custom convolutional neural network with four convolutional blocks and two fully connected layers. The architecture employs progressive channel expansion (3→32→64→128→256) to capture increasingly complex features, with 3×3 kernels providing efficient receptive field growth. Each convolutional block includes batch normalization for training acceleration, ReLU activation, and 2×2 max pooling for spatial dimension reduction (64×64→4×4, representing 16× reduction). The classifier head flattens 256×4×4 features to 4,096 dimensions, passes through a 256-unit hidden layer with dropout (p=0.5) for regularization, and outputs class logits. Total parameters: ~2.5 million (see Appendix A, Code Sample 2 for complete architecture).

Training Configuration

Training employed Adam optimizer (lr=0.001) with cross-entropy loss, batch size 128, and 15 epochs. The 80/20 train-validation split maintained temporal independence with different video sequences in each set. Best model checkpointing preserved the configuration achieving peak validation accuracy. Training converged rapidly: epoch 1 achieved 91.95% training and 99.53% validation accuracy, with final epoch 15 reaching 99.80% training and 99.70% validation accuracy (train loss 0.0070, validation loss 0.0099). The small train-validation gap (<1%) indicates excellent generalization without overfitting despite the 2.5M parameter model.

Performance Metrics

Validation employed hold-out strategy with per-class precision, recall, and F1-scores. The model achieved 99.7% overall accuracy with exceptional per-class performance: cars (1.00/1.00/1.00 precision/recall/F1), vans (0.98/0.98/0.98), others (0.99/0.97/0.98), and buses (1.00/1.00/1.00). Confusion matrix analysis revealed minimal misclassification: only 1.5% of vans mistaken for cars and 1.7% of others misclassified as cars, aligning with class imbalance and perspective effects. The weighted average F1-score of 1.00 demonstrates robust handling of minority classes despite severe imbalance (92% cars vs. 1% buses).

Results and Analysis

Classification Performance

The CNN achieved near-perfect classification on dominant classes (cars, buses) and strong performance on minority classes (vans, others). Perfect scores for cars (F1=1.00) reflect well-represented training data and clear visual distinctiveness. Buses also achieved perfect scores

(F1=1.00) due to distinctive large size despite representing only 1.2% of training data. Vans and others achieved F1=0.98, demonstrating robust generalization to underrepresented classes. Size-based confusion patterns (vans→cars 1.5%, others→cars 1.7%) indicate perspective effects and form factor similarity as primary classification challenges rather than fundamental model limitations (see Appendix B, Table 1 for complete classification report and Figure 5 for confusion matrix visualization).

Traffic Analytics

Vehicle counts were aggregated into Light Motor Vehicle (LMV: car/van/others) and Heavy Motor Vehicle (HMV: bus) categories for traffic policy analysis. Sequence MVI_20011 exhibited mean 11.53 total vehicles per frame (11.39 LMV, 0.14 HMV), yielding an 81:1 LMV:HMV ratio (98.79% vs. 1.21%). This ratio informs infrastructure planning: HMV proportions directly correlate with road wear patterns, emission profiles, and freight corridor optimization (see Appendix B, Figures 6-7 for temporal visualizations).

Short-term traffic forecasting employed Random Forest regression with K=5 frame history to predict next-frame vehicle counts. The model achieved R^2=0.947 and MAE=0.202 vehicles, demonstrating excellent predictive capability for adaptive traffic signal control and congestion warning systems. High concordance between actual and predicted values validates the approach for operational deployment (see Appendix B, Figure 8 for forecast visualization).

Real-World Application

Real-world validation processed traffic_example.mov (640×372, 60 FPS, 1,977 frames, 33 seconds) using background subtraction (MOG2) for detection combined with CNN classification. Processing achieved 203.8 FPS (9.7 seconds total), demonstrating 3.4× real-time

capability on CPU without GPU acceleration. Results: mean 1.2 vehicles/frame, maximum 4, traffic density 88.8% low (0-2 vehicles), 11.2% medium (3-5 vehicles), 0% high (6+ vehicles). Class distribution: 82.8% cars, 5.6% vans, 5.9% buses, 5.6% others across 2,361 total detections. Successful generalization to different resolution (640×372 vs. 540×960 training) and frame rate (60 vs. 25 FPS) validates model robustness (see Appendix B, Figures 9-10).

Image Restoration Capabilities

Additional capabilities were developed using pretrained Hugging Face models for degraded footage enhancement. Stable Diffusion x4 Upscaler provides super-resolution (+5 dB PSNR improvement, SSIM=0.88) and denoising (-20 dB noise reduction) at 15-30 seconds per frame on GPU. GAN-based colorization (ResNet-34 + U-Net) converts grayscale to color (PSNR=28.7 dB, SSIM=0.84) at 2-5 seconds per frame. These modules operate independently in the current implementation but demonstrate feasibility for preprocessing degraded surveillance footage in production scenarios where computational cost justifies accuracy gains. The diffusion upscaler combines learned generative priors with input conditioning to synthesize plausible high-frequency details, occasionally hallucinating textures that are perceptually plausible but not ground-truth accurate. The GAN colorizer generates realistic color distributions (gray roads, blue skies, multicolor vehicles) suitable for modernizing archival black-and-white traffic footage (see Appendix B, Figures 11-14 for restoration examples).

Discussion and Future Work

This project successfully demonstrates end-to-end computer vision methodology for intelligent transportation systems. Key achievements include 99.7% classification accuracy, real-time

processing capability, and accurate short-term forecasting. The modular architecture enables iterative improvement without system redesign.

Several enhancements would transition the system from research prototype to production deployment. First, integrating robust object detection (YOLOv8, Detectron2) would enable end-to-end processing without reliance on ground-truth annotations. Second, multi-object tracking (DeepSORT, ByteTrack) would assign persistent vehicle IDs across frames, preventing double-counting and enabling trajectory-based analyses (speed estimation, lane changes, dwell times). Third, addressing class imbalance through targeted data augmentation (random erasing, photometric jitter, geometric transforms), oversampling underrepresented categories, and loss-level adjustments (class-weighted cross-entropy, focal loss) would further improve minority class performance. Fourth, temporal modeling incorporating tracker-derived features (counts per track, speeds, entry/exit events) would increase forecasting power and enable anomaly detection. Finally, formalizing selective application of image restoration based on quality metrics (PSNR/SSIM thresholds) would optimize the computational cost-benefit trade-off for degraded inputs.

Conclusion

This work demonstrates a complete computer vision pipeline for vehicle classification and traffic analytics, achieving 99.7% accuracy on 598,281 vehicle samples with real-time processing capability (203.8 FPS). The system successfully analyzes traffic composition (81:1 LMV:HMV ratio), performs short-term forecasting (R^2=0.947), and demonstrates generalization to real-world video with different specifications. Additional image restoration capabilities showcase potential for enhancing degraded surveillance footage. The modular, software-defined architecture

provides a foundation for intelligent transportation systems applicable to smart city infrastructure, environmental monitoring, and autonomous vehicle perception. With proposed enhancements (YOLO integration, multi-object tracking, temporal modeling), the system can transition to production-ready deployment for automated traffic management and urban planning applications.

References

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).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (pp. 448-456). PMLR.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 [Computer software]. GitHub. https://github.com/ultralytics/ultralytics

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations (ICLR).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems (Vol. 32).

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10684-10695).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M. C., Qi, H., ... & Siebel, N. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding, 193, 102907. https://doi.org/10.1016/j.cviu.2020.102907

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (pp. 38-45).