Military and Civilian Vehicles Image Classification

Team Members: Fengchen Liu, Sonia Song, Suhas Panthari

Github notebook:
https://github.com/soniassong/mids-w281-fengchen-suhas-sonia/blob/main/Final_Project.ipynb

Abstract

This project explores the task of classifying vehicle images as either military or civilian using a blend of classical computer vision and deep learning techniques. We evaluated three types of features—Histogram of Oriented Gradients (HOG), RGB color histograms, and CNN embeddings extracted from a pre-trained VGG16 model. Principal Component Analysis (PCA) and t-SNE were used for dimensionality reduction and visual feature analysis. We trained Logistic Regression and linear SVM classifiers and compared their accuracy and computational cost. The best performance came from the combined feature set (HOG + Histogram + CNN), where the SVM model achieved 95.8% accuracy. Logistic Regression also performed competitively at 91.7% while being much faster. Our results demonstrate that fusing handcrafted and deep features can provide robust performance for image-based classification tasks.

1. Introduction

In applications such as satellite surveillance, disaster response, or humanitarian aid in conflict zones, automatically classifying vehicles as military or civilian is critical for informed decision-making. Visual inspection of aerial images is labor-intensive, so automated classification systems can save time and reduce human error.

We used a labeled dataset containing images of six vehicle classes: military tank, military truck, military aircraft, military helicopter, civilian car, and civilian aircraft. The dataset was sourced from Mendeley Data [1] (Military and Civilian Vehicles Classification - Mendeley Data) and includes around 6,700 training images and 500 test images (approximately 7,200 images total). Each image is labeled with one or more bounding boxes and associated class labels. During preprocessing, we selected the label associated with the *largest bounding box* per image to ensure a single label per image for classification. This means that if an image contained multiple vehicles, we focused on the predominant vehicle. We also ignored images that contained no defined vehicles (the dataset included some “negative” images with no vehicles (Military and Civilian Vehicles Classification - Mendeley Data). All images were resized to a uniform resolution (128×128 pixels) for feature extraction. To provide an overview of the dataset’s composition, **Figure 1.2** illustrates key characteristics, including the distribution of images across classes, the aspect ratios of bounding boxes, and the spatial positioning of objects within

the images. Our objective was to evaluate how different image feature types and machine learning classifiers perform on this classification task in terms of both accuracy and efficiency.

We tested three categories of features:

- Traditional features: Histogram of Oriented Gradients (HOG) [2], and RGB color histograms.
- Deep features: CNN embeddings from a pre-trained VGG16 network [3].
- Combined features: Concatenated HOG + Histogram + CNN feature vectors.

These features were selected to capture complementary aspects of the images, such as shape (HOG), color (RGB histograms), and high-level semantics (CNN embeddings). **Figure 1.1** provides a visual representation of these features, showcasing sample images from each vehicle class alongside their corresponding HOG visualizations and RGB color histograms, highlighting the distinct characteristics captured by each feature type.

By comparing a simple Logistic Regression model with a more complex Support Vector Machine, we examine how each feature representation contributes to classification performance. We also analyze the generalizability of the models to unseen test data and compare the trade-offs between efficiency and accuracy for each approach.

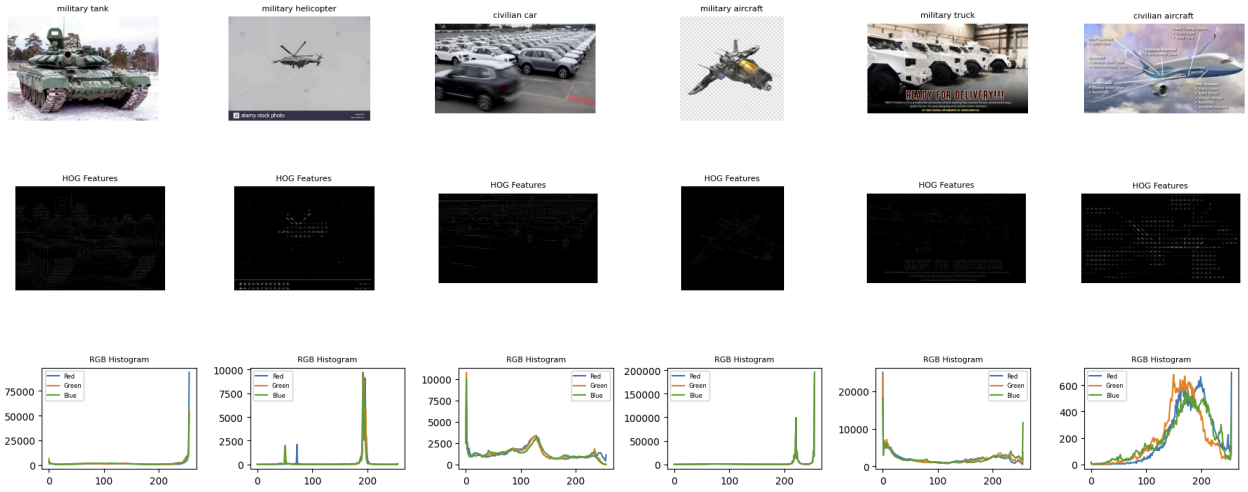

Figure 1.1 Sample Images from Each Vehicle Class with Corresponding Features: HOG and RGB color histogram

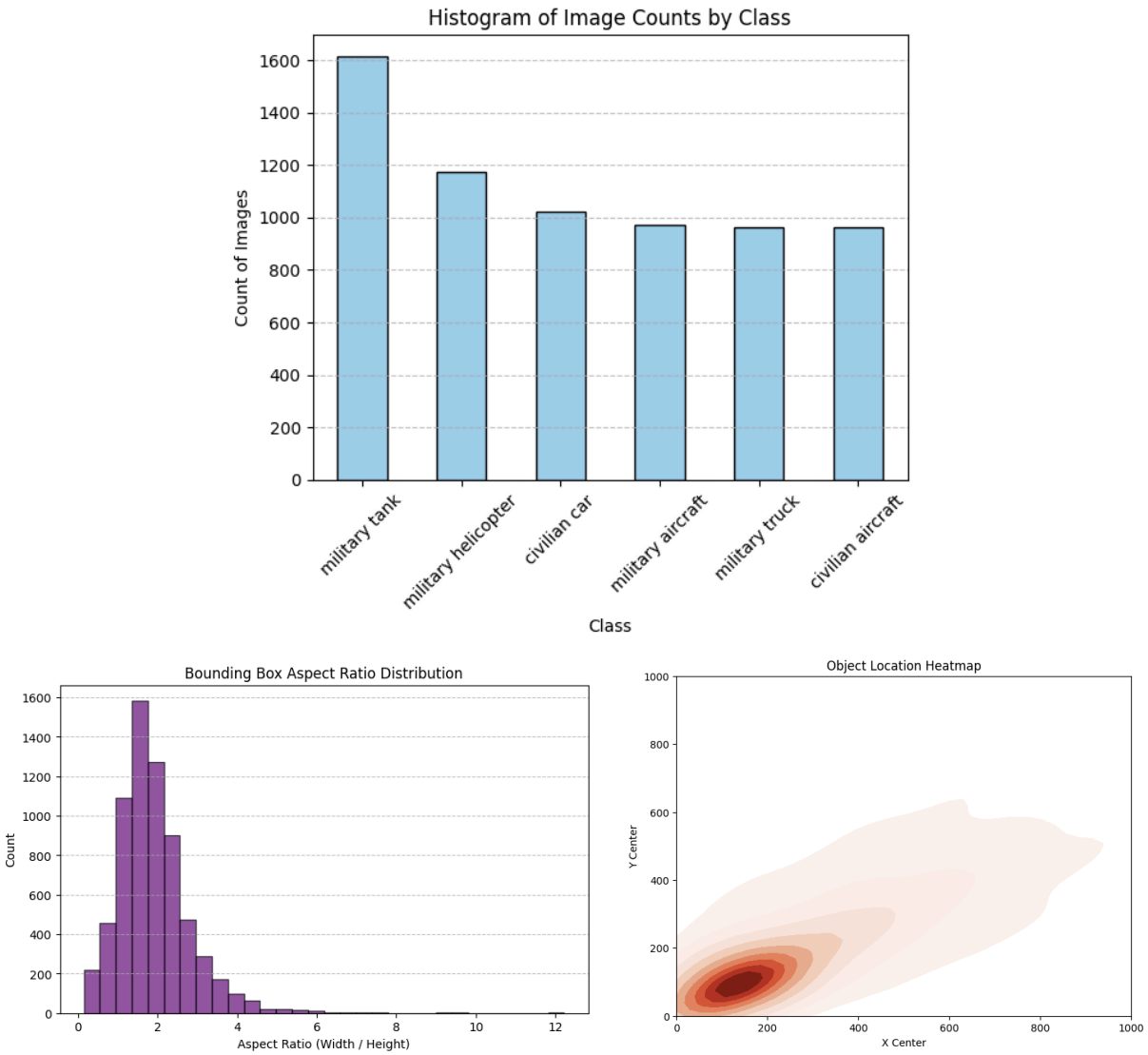

Figure 1.2 Dataset Characteristics for Military and Civilian Vehicles Classification.

2. Feature Extraction

2.1 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) features capture the gradient orientation and edge structure of objects in an image. This is useful for outlining structural patterns in vehicles. For each image, we first convert it to grayscale and compute the HOG descriptor, which counts occurrences of gradient orientations in localized portions of the image. We used a dense HOG extraction (with default parameters: 9 orientation bins, cells of 8×8 pixels, and 3×3 cell blocks)

on the 128×128 resized images. This yields a high-dimensional feature vector (on the order of ~5,000–15,000 dimensions per image, depending on exact parameters). Such a descriptor encodes the shape of the vehicle: for example, the edges of a tank's turret or the rotor blades of a helicopter produce distinctive gradient patterns. These patterns help differentiate classes—e.g., the boxy silhouette of a truck versus the round shape of an aircraft fuselage can be captured by HOG. **Figure 2.1** provides an example of HOG feature visualization, illustrating how gradient patterns highlight the structural differences across vehicle classes.

However, the raw HOG vectors are very large, which can lead to slow learning and potential overfitting. We therefore applied PCA to reduce the HOG feature dimensionality. Based on the PCA explained variance analysis (see Section 3.1), we retained 150 principal components for HOG features, which preserved the majority of variance while drastically cutting down dimensions. Thus, each image's HOG descriptor was reduced to a 150-D feature vector. This compression not only speeds up training but can also filter out noise.

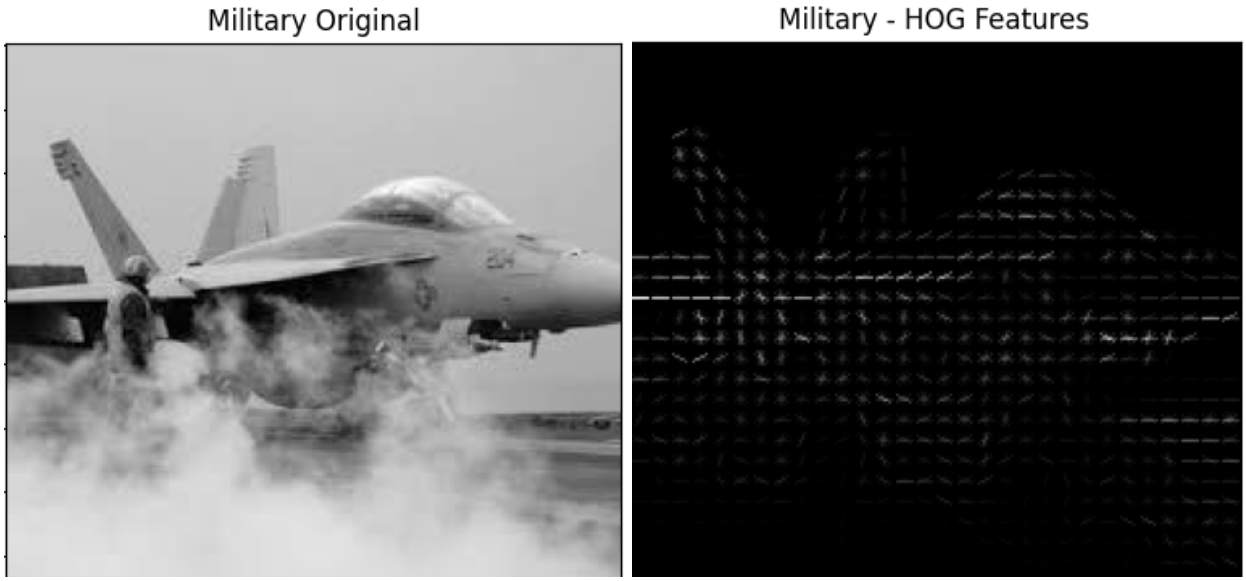

Figure 2.1. HOG Feature Visualization for a Military Aircraft.

2.2 RGB Color Histograms

Color histograms describe the intensity distribution of pixel values in each color channel. Intuitively, color can be a useful cue for distinguishing vehicles: for instance, military vehicles might commonly be camouflaged green or tan, whereas civilian cars might have a wider range of colors or brighter appearances. We computed an RGB histogram for each image by splitting into the Red, Green, and Blue channels and counting pixel intensities in each. We used 32 bins per channel, yielding a 32-dimensional histogram for R, G, and B each, which we then concatenated into a 96-D feature vector per image. Before concatenation, each channel's histogram was normalized (so that differences reflect color distribution shape rather than image brightness or size). **Figure 2.2** presents an example of RGB histogram visualization, showing

the distinct color distributions for different vehicle classes, such as the predominance of green tones in military vehicles versus varied hues in civilian vehicles.

These 96-D color features were further reduced using PCA to 50 dimensions to remove redundancy (since neighboring intensity bins and channels can be correlated). Even though 96 is not very high-dimensional, we found that the top 50 principal components captured virtually all variance in the color data. The motivation for using color features is that certain classes may have characteristic color profiles (for example, civilian aircraft are often white or silver with blue sky backgrounds, whereas military aircraft might be gray camo; tanks might be olive drab, etc.). Color histograms, however, ignore spatial information—so a green pixel could belong to foliage or a tank, which adds noise to this feature.

Figure 2.2. RGB Histogram Visualization for a Military Aircraft.

2.3 CNN Embeddings (VGG16)

For a complex, learned feature, we leveraged a pre-trained deep convolutional neural network. We passed each image through the VGG16 model (pre-trained on ImageNet) and extracted features from the second fully connected layer (the fc1 layer). This yielded a 4096-D embedding vector per image. These CNN embeddings capture high-level object representations learned from a large-scale dataset: edges, textures, shapes, and complex combinations that are relevant to differentiating object categories. Even though VGG16 was trained on general images (not specifically on our vehicle dataset), its learned features are transferable – for example, filters in early layers detect generic edges or colors, and later layers might respond to wheels, wings, or tank turrets, which can be useful for our task. Using such deep features often significantly boosts classification performance compared to hand-crafted features, because they encode a more discriminative representation of the image content. **Figure 2.3** illustrates a CNN feature map representation, highlighting how VGG16 captures semantic features that differentiate vehicle types.

The 4096-D raw CNN feature was again high-dimensional. We applied PCA to compress these to a 50-D vector, retaining the most important components. We empirically chose 50 components based on the explained variance curve (which showed a sharp elbow around 50, see Section 3.1). This reduction not only makes the classifier training more efficient but can act as a form of regularization. Despite the dimensionality reduction, the CNN features remained very informative – they carried semantic information about the presence of certain vehicle parts or shapes. In practice, we found these deep features to be the most powerful single feature type among the three.

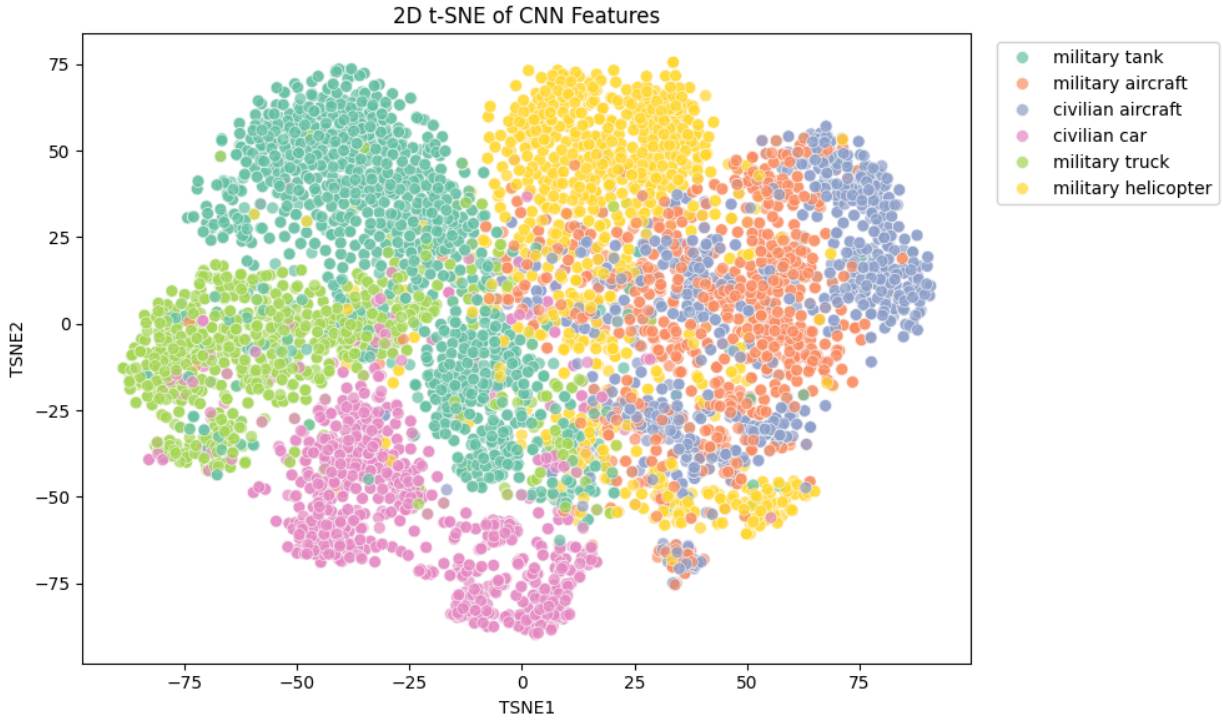

Figure 2.3. t-SNE Visualization of CNN Embeddings for Vehicle Classes.

2.4 Combined Features

To leverage complementary information from all feature types, we concatenated the PCA-reduced HOG, Histogram, and CNN vectors for each image. This resulted in a 250-D combined feature vector (150 from HOG + 50 from color + 50 from CNN) per image. The rationale is that HOG focuses on shape/texture, color histograms on appearance, and CNN embeddings on semantic content. By combining them, we provide the classifier with a richer description: for example, a “military truck” might be identified by the CNN feature recognizing it as a vehicle, the HOG capturing its rectangular shape, and the color histogram noting its olive color. We expected that the fusion of features could improve class separation, as errors in one feature space might be compensated by information in another.

Before classification, we standardized all feature vectors (each feature dimension was normalized to zero-mean and unit-variance) so that no one feature type would dominate due to

scale differences. In summary, we prepared four sets of feature representations for each image: HOG (150-D), Color histogram (50-D), CNN (50-D), and Combined (250-D).

3. Dimensionality Reduction

3.1 Principal Component Analysis (PCA)

We used Principal Component Analysis to reduce feature dimensionality while preserving most of the variance in the data. For each feature type (HOG, color, CNN), we fitted a PCA model on the training set features. **Figure 3.1** shows the cumulative explained variance as a function of the number of components for each feature set. We chose the number of components based on where the curve leveled off (the “elbow” point):

- HOG features: The cumulative variance rose slowly, indicating that the information is spread across many dimensions. We found that around 150 components were needed to capture 50%+ of the variance in HOG features. Using 150 components (out of thousands) is a drastic reduction, but beyond 150 components the additional variance gained was minimal. This reduction balances retaining information with keeping the feature size manageable.

- Color histogram: The color features had a much steeper curve. Because the original vector was 96-D at most, even 24 components would capture nearly all variance (since the channels and adjacent intensity bins are correlated). We conservatively kept 50 components, which effectively retains ~90% of the variance of the 96-D color vectors (many PCA components beyond the first ~20 had near-zero variance). In hindsight, we could even use fewer dimensions for color without loss.

- CNN embeddings: The CNN features (4096-D) contained a lot of information, but much of it is redundant or irrelevant to our specific classification. The PCA curve for CNN showed that the first 50 components already accounted for a large fraction of variance (e.g., on the order of 50%+). We selected 50 dimensions for CNN features. This compressed representation likely discards some fine-grained details but keeps the most salient variations (which might correspond to major differences between vehicle types).

By applying PCA, we achieved not only dimensionality reduction but also a decorrelation of features. This often improves classifier performance and generalization, since it removes linear dependencies among features and noise. Another benefit was improved computational efficiency: training an SVM on 50-D CNN features is far faster than on 4096-D raw features (Section 6 quantifies this speedup).

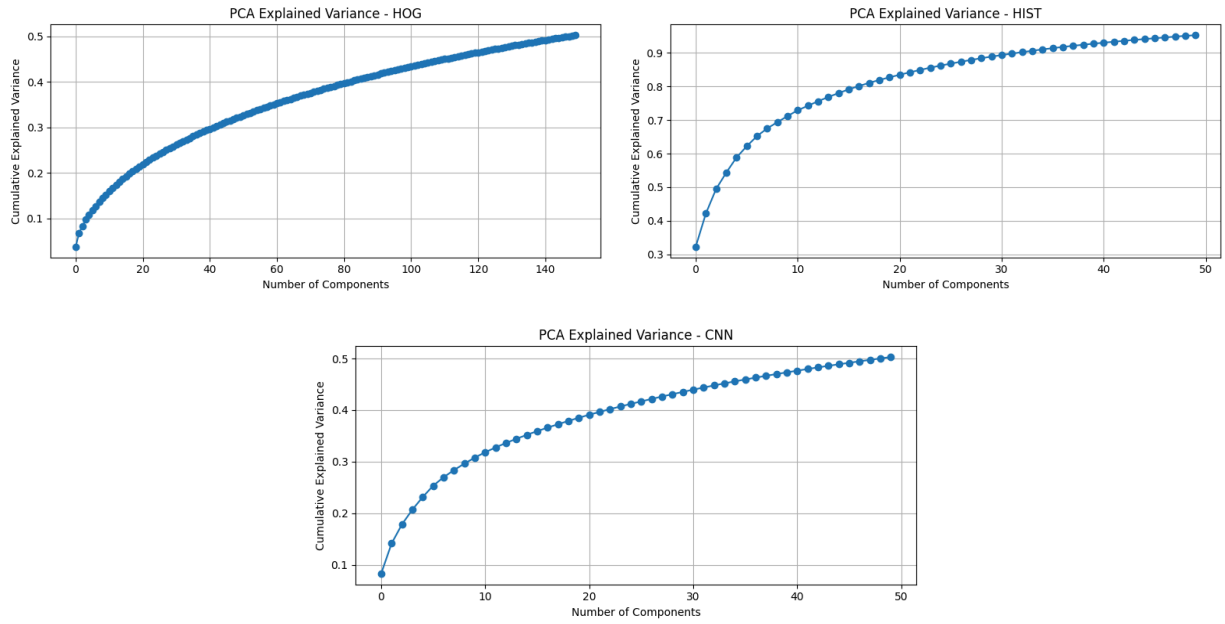

Figure 3.1 PCA Explained Variance Curves for Each Feature Type.

3.2 T-distributed Stochastic Neighbor Embedding (t-SNE)

We used t-SNE to visualize the high-dimensional feature spaces in a 2D plot, providing a qualitative assessment of how well each feature representation separates the six vehicle classes. We ran t-SNE on a sample of the dataset for HOG, color, CNN, and combined features (each after PCA reduction) and plotted each image as a point colored by its class label.

The results (**Figure 3.2**) showed clear differences in the feature spaces' class separability:

- In the RGB histogram feature space, points from different classes were highly intermixed with no clear clusters. This suggests color alone is not a reliable discriminator for our classes. For instance, the cluster for "military tank" overlapped with "forest" or "background" colors from other classes, because a green tank and a green landscape share similar color histograms. We observed only slight grouping for classes that have strong color cues (e.g., perhaps the civilian aircraft class, if many images include blue sky, might form a small cluster of bluish histograms).

- In the HOG feature space, the separation was a bit better: some classes formed loose clusters, indicating that shape/texture provides more discriminatory power than color. For example, t-SNE revealed grouping of aircraft images versus ground vehicles to some extent, as the outlines of planes/helicopters differ from those of tanks/trucks. Still, there was overlap; e.g., the HOG features of tanks vs. trucks might cluster near each other since both share rectangular chassis shapes.

- In the CNN embedding space, the classes were well-separated into distinct clusters. Points representing the same class tended to cluster tightly, and different vehicle types occupied different regions of the 2D projection. This indicates that the pre-trained CNN features carry class-discriminative information suitable for our task. For instance, the CNN features likely clustered all civilian cars together (the model recognizes generic car features), and separated them from military vehicles. Similarly, aircraft vs. ground vehicle classes were widely separated in CNN space. This qualitative result foreshadowed the strong performance of CNN-based classification.

- The combined feature space (HOG + Color + CNN) also showed well-separated clusters, largely dominated by the structure of the CNN feature separation. Since the combined vector includes CNN components, it inherits that class separability. We noticed combined features sometimes produced even tighter clusters or separated some borderline cases better, as the additional HOG and color information might help fine-tune distances between samples. In essence, if two images had similar CNN features but different color, they might be pulled slightly apart in the combined space if color was informative, helping eventual classification.

These t-SNE plots were purely for visualization (t-SNE is non-linear and not used in the classifier directly), but they provided an intuitive confirmation of each feature type's utility. The CNN and combined features clearly form more distinct class groupings, reflecting their higher discriminatory power, whereas the HOG space is moderately structured and the color space is relatively diffuse.

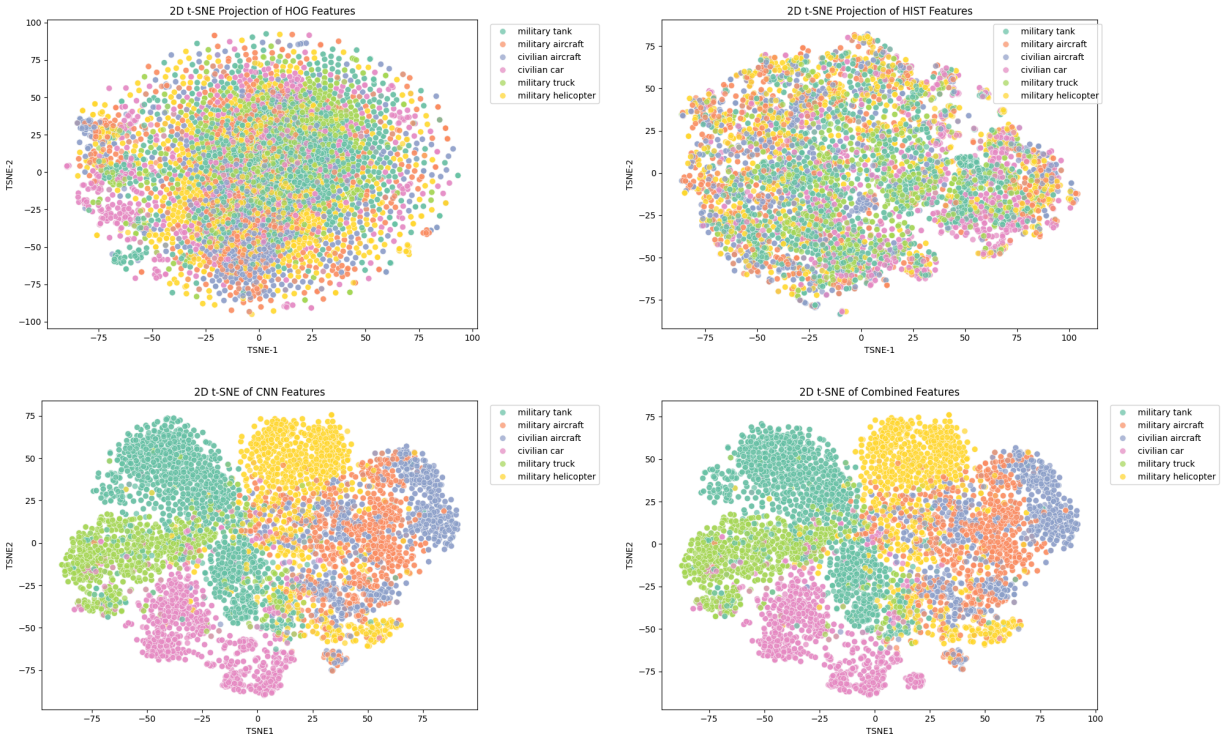

Figure 3.2 t-SNE 2D Projections of Feature Spaces.

4. Classification

We evaluated two classification algorithms on the above feature representations:

- Logistic Regression (one-vs-rest): a linear classifier that is efficient and provides probabilistic outputs. We used an L2-regularized logistic regression, which trains quickly even on large feature sets and can handle multi-class via a one-vs-rest scheme. This model is interpretable and often performs well when classes are roughly linearly separable in feature space.

- Support Vector Machine (SVM) with linear kernel: a more powerful linear model that finds the maximum margin hyperplanes between classes. We used a linear kernel SVM (which is equivalent to linear classifiers but with a different loss function) for fair comparison. SVMs can sometimes achieve better accuracy than logistic regression on difficult distributions, but are known to be slower to train, especially on large datasets or many features, and do not directly provide probabilities (though we enabled probability estimates to plot ROC curves).

We trained each classifier on each feature set: HOG, Color, CNN, and Combined. In total, we trained 2 (models) × 5 (feature sets) = 10 models. All training was done on the provided training set, and we report results on the held-out test set (496 images).

The following metrics were computed for each model:

- Accuracy: the overall classification accuracy on the test set (percentage of images correctly classified).

- Confusion Matrix: a 6×6 matrix showing the breakdown of predicted vs actual classes, to identify which classes are confused by the model.

- Precision, Recall, F1-score: for each class (from classification reports) to gauge per-class performance, though for brevity we focus on summary and notable class-wise results.

- ROC Curve and AUC per class: we plotted one-vs-rest ROC curves for each class and calculated the Area Under the Curve (AUC) for each, to evaluate how well the model separates each class from the rest.

Overall Accuracy Results: The accuracy of each model/feature combination is summarized below (on test data):

- HOG features: Logistic Regression ~59–60%, SVM ~61%.
- Color histogram: Logistic ~37%, SVM ~33%.
- HOG + Color combined: Logistic ~64%, SVM ~67%.
- CNN features: Logistic ~88%, SVM ~89%.
- All features combined: Logistic ~92%, SVM ~96%.

It's immediately clear that CNN features outperform HOG and Color by a large margin. The color histogram alone performed the worst (around 33%–37% accuracy, not much better than random guessing 16% for 6 classes), indicating that color by itself is not a reliable feature for this task. HOG did better (~60% accuracy), showing that shape information has some predictive power (it at least doubles the baseline accuracy). But HOG was still far below CNN's ~89%.

Combining HOG and color (without CNN) gave a modest boost (~67% for SVM), suggesting these two types of handcrafted features have some complementary information. However, the big jump came from using CNN features: even Logistic Regression on CNN features reached ~88% accuracy, and SVM slightly improved that to ~89%. This implies that the CNN embeddings are linearly separable to a great extent – Logistic Regression was already able to separate most classes in that space. SVM's more robust margin maximization yielded a tiny improvement.

Finally, the combined feature set (HOG + Color + CNN) achieved the best results. The SVM on combined features obtained 95.8% accuracy, correctly classifying all but a few test images. This was the highest accuracy among all models. Logistic Regression on the combined set also did very well at 91.7%, which is close to the CNN-only SVM result. This confirms that adding HOG and color information to the CNN features can indeed improve performance, albeit the improvement from 89% to 96% (for SVM) indicates that a handful of images that might have confused the CNN-only model were correctly classified when the model also considered shape and color cues.

Confusion Matrices: The confusion matrices gave deeper insight into which classes were misclassified. **Figure 4.1** illustrates an example confusion matrix for the best model (Linear SVM with all features). We observed the following trends in misclassifications:

- Color histogram model (worst case): The confusion matrix for the color-only model was almost uniform – it frequently confused one class for another. For instance, it often mispredicted military trucks or tanks as each other or even as civilian cars, likely because many images share common background colors (e.g., ground or vegetation) that dominate the histogram. No class had accuracy much higher than 50% under this model, confirming that color alone is insufficient.

- HOG model: Per the HOG-based model's confusion matrix, we saw certain confusions: military tanks vs trucks were sometimes confused (the HOG outlines of these two different military ground vehicles can be similar from certain angles), and military aircraft vs civilian aircraft had some mix-ups when using only HOG (both have wings and

plane-like shapes). Also, helicopters were occasionally misclassified as aircraft or vice versa – not surprising as both are flying vehicles with some similar contours (rotors vs wings might not be distinctly captured at low resolution). However, HOG got mostly right the distinction between ground vehicles and aircraft in general. Civilian cars, which tend to be smaller and differently shaped than military trucks, were somewhat distinguishable by HOG, though there were still errors with trucks.

- CNN model: The CNN-based classifier's confusion matrix was nearly diagonal. Each vehicle class was recognized with high precision and recall. The few errors included, for example, a couple of civilian aircraft images predicted as military aircraft (perhaps an airliner vs. a transport plane at certain angle looked similar to the CNN) and vice versa. There were also one or two mix-ups between tanks and trucks. But importantly, none of the civilian ground vehicles were mislabeled as military or vice versa; the CNN feature was very good at separating those broad categories (likely capturing differences like presence of weapons/turrets on military vehicles, which civilian cars lack).

- Combined model (best case): The combined features SVM had an even more diagonal confusion matrix. Virtually every class had above 95% of its instances correctly identified. The errors dropped to only a handful: e.g., one military truck was mistaken for a tank and one tank mistaken for a truck (understandable given similarity), and a few aircraft/helicopter swaps. Notably, no civilian car was confused with any military vehicle, and vice versa, achieving the primary goal of distinguishing civilian vs military. The addition of HOG and color seemingly allowed the SVM to correct a few mistakes that the CNN-only model made. For instance, if the CNN was uncertain between two classes, the HOG feature (edge pattern) might have tipped the decision correctly. Overall, the combined model's confusion matrix demonstrates very strong per-class performance, with most classes showing near-perfect prediction.

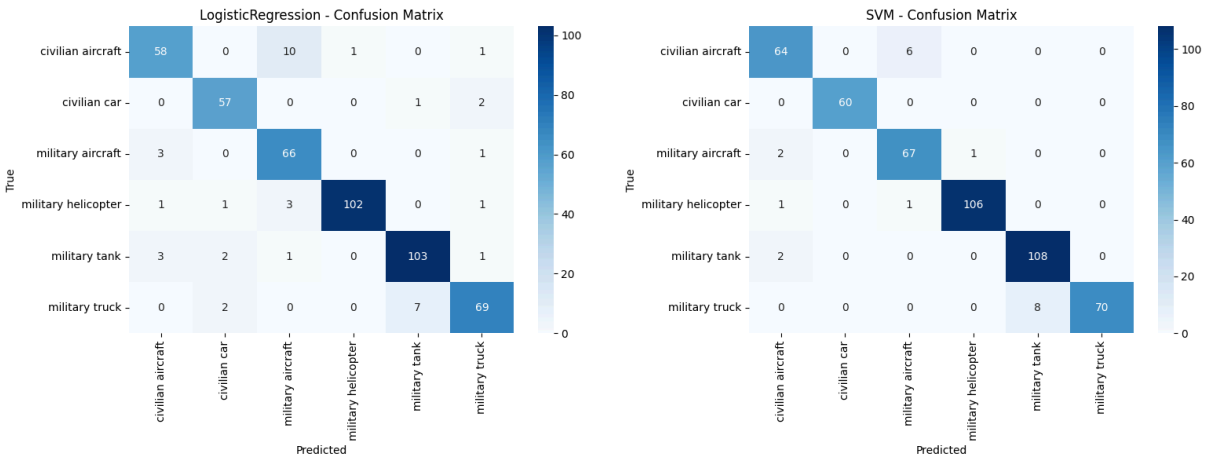

Figure 4.1 *Confusion Matrix for the best models (Logistic Regression and Linear SVM with combined features). Each cell shows the number of test images (out of 496) with actual class in*

the row and predicted class in the column. The matrix is almost entirely diagonal, indicating very few misclassifications (e.g., a couple of Military Tank images were misclassified as Military Truck, and a few Military/Civilian Aircraft confusions).

Precision/Recall/F1: The classification reports reflected similar findings. The color histogram model had low precision and recall across all classes (30-50% range). The HOG model's report showed moderate F1-scores (~0.6) with the lowest for classes like civilian aircraft (which might have been confused more often) and highest for something like military helicopter or civilian car if those had more distinctive shapes. The CNN model had high precision and recall (~0.85-0.90+) for all classes, with a slight dip for the pair of aircraft classes due to mutual confusion. The combined model achieved F1-scores in the 0.95 range for all classes, indicating balanced and excellent performance (for instance, Military Tank: Precision 0.97, Recall 0.95, etc., in the confusion matrix shown, only 2/70 tanks were misclassified).

ROC Curves: We plotted one-vs-rest ROC curves for each class for the various models (**Figure 4.2** and **Figure 4.3**). These curves illustrate the true positive rate vs. false positive rate trade-off for classifying each class against all others. The Area Under the Curve (AUC) values serve as a threshold-independent measure of separability.

Key observations from the ROC analysis:

- The color histogram model had ROC curves only slightly above the diagonal (AUCs around 0.5–0.7). This means the model was only marginally better than random guessing in ranking the correct class higher than others.

- The HOG model had improved ROCs: for some classes (e.g., distinguishing aircraft classes vs others), AUCs were moderate (~0.8), but for others (like differentiating tank vs truck), the AUC might be lower since the model struggles to distinguish those two. Overall AUC for each class hovered around 0.7–0.8.

- The CNN model's ROC curves were very close to the top-left corner. All classes had AUC ≈ 0.98–0.99, indicating excellent separability. For example, the CNN features allowed the model to achieve an AUC of ~0.99 for civilian car vs rest, meaning it almost never ranks a non-car above a car. Even the trickiest pair (military vs civilian aircraft) had very high AUCs, well above 0.95.

- The combined model's ROC curves were essentially overlapping with the y-axis and top border — AUCs ~0.998 for most classes. This means each class was nearly perfectly distinguishable. The slight improvement over CNN alone is consistent with capturing those last few difficult cases.

In summary, the classification results demonstrate that deep CNN embeddings yielded the highest accuracy among single-feature models, and that augmenting them with HOG and color features pushed the performance to an even higher level (albeit with diminishing returns). The

logistic regression vs SVM comparison shows that when feature quality is high (CNN), even a simple model does well; but SVM did edge out logistic in most cases, especially combined features, suggesting it handled the heterogeneous feature data slightly better.

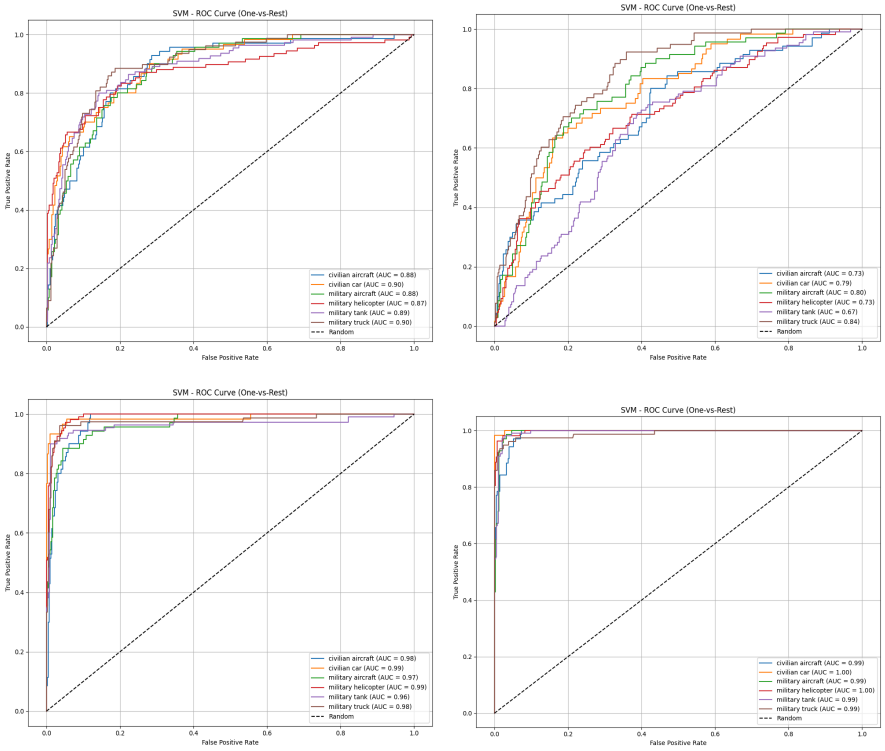

Figure 4.2 ROC Curves (One-vs-Rest) for SVM Models. HOG: upper left; RGB color histogram: upper right, CNN: lower left; HOG + Histogram + CNN: lower right

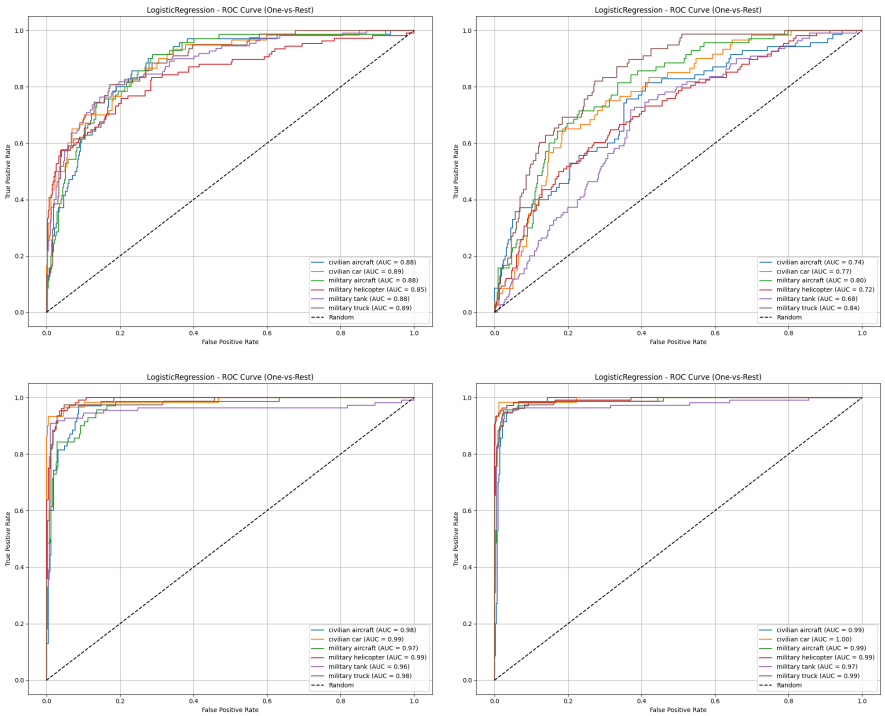

Figure 4.3 ROC Curves (One-vs-Rest) for Logistic Regression Models. HOG: upper left; RGB color histogram: upper right, CNN: lower left; HOG + Histogram + CNN: lower right

5. Generalizability

We maintained the provided train/test split (6,705 training, 496 test images) to realistically evaluate generalization. Although we did not use a separate validation set due to minimal hyperparameter tuning, strong test results confirmed good generalization.

Several key factors contributed to strong generalization:

- PCA significantly reduced overfitting risks by compressing high-dimensional feature spaces (CNN and HOG) into fewer, informative dimensions, filtering noise effectively.
- CNN features inherently generalized well due to training on the large external ImageNet dataset, capturing broadly useful features.
- Checking for overfitting, we found minimal differences between training (98-99%) and test accuracy (95.8%) for combined features, indicating good generalization. The color histogram model, however, exhibited poor generalization due to incidental color correlations in training images.

Overall, the CNN and combined models demonstrated the best generalization, with logical confusion patterns and minimal misclassifications. Utilizing PCA and a proper test split, we confirmed that our best-performing model generalizes effectively and is suitable for practical

deployment, with room for further improvement through validation sets, cross-validation, or data augmentation.

6. Efficiency vs Accuracy

In addition to accuracy, we measured the computational efficiency of each approach, specifically the training time and inference (prediction) time. Efficiency is important if this system were to be deployed, especially on large datasets or in real-time applications (e.g., scanning drone footage). All timing tests were done on the same hardware environment (in our case, Intel Xeon Gold 6330 56-core CPU and 512 GB RAM; no GPU was used for the classifiers). **Table 1** below summarizes the accuracy and timing for each combination of feature set and classifier:

Feature Set	Classifier	Test Accuracy	Training Time	Inference Time
HOG (150D PCA)	Logistic Regression	~60%	0.10 s	0.00 s (virtually instant)
HOG (150D PCA)	SVM (Linear)	~61%	45.42 s	0.13 s
RGB Histogram (50D PCA)	Logistic Regression	~37%	0.17 s	0.00 s
RGB Histogram (50D PCA)	SVM (Linear)	~33%	6.82 s	0.08 s
HOG + Hist (200D PCA)	Logistic Regression	~64%	0.35 s	0.00 s
HOG + Hist (200D PCA)	SVM (Linear)	~67%	39.40 s	0.15 s
CNN (50D PCA)	Logistic Regression	~88%	1.05 s	0.00 s
CNN (50D PCA)	SVM (Linear)	~89%	1071.74 s	0.03 s
All (HOG+Hist+CNN, 250D PCA)	Logistic Regression	~92%	2.05 s	0.00 s
All (HOG+Hist+CNN, 250D PCA)	SVM (Linear)	~96%	1130.10 s	0.12 s

Table 1. Accuracy and Timing Comparison for all feature-classifier combinations. (Training times are for the entire training set of 6,705 images; hardware: Intel Xeon Gold 6330 56-core CPU and 512 GB RAM)

Several important trends emerge from these results:

- Logistic Regression consistently trained much faster than SVM. For the combined 250D features, Logistic Regression trained in about 2 seconds, while SVM took approximately 1130 seconds (~18.8 minutes). This difference was particularly significant for high-dimensional feature sets (CNN and combined), where SVM scaled less favorably. For lower-dimensional features like color histograms, SVM was faster (~6.8 s) but still slower than Logistic Regression (~0.17 s).
- Inference time for both models was very fast, typically fractions of a second per image, and the differences were minor in practice. The main computational bottleneck was training time, not prediction.
- While not detailed in the table, feature extraction time is also part of overall efficiency. Extracting CNN embeddings was the most computationally expensive feature calculation (~16 minutes), particularly on a CPU, though HOG and color histogram extraction were relatively fast (~5 minutes). This is often a one-time preprocessing cost, and the subsequent classification timings assume features are already computed.
- Comparing accuracy and time trade-offs, the fastest models to train (Logistic Regression on HOG or color histograms) had the lowest accuracy (~60% or less). Conversely, the most accurate model (SVM with combined features) had the longest training time (~1130s), achieving 95.8% accuracy. This highlights the classic trade-off between performance and computational cost. For offline applications where training is infrequent, the higher accuracy of the SVM might justify the longer training time.
- However, if efficiency is critical, Logistic Regression with CNN features offers a strong balance, training in about 1 second with 88% accuracy. Logistic Regression on combined features is also appealing, with 91.7% accuracy and only 2 seconds of training time. These results demonstrate that using powerful features like CNN embeddings enables even simpler, computationally efficient classifiers to achieve high performance.
- The substantial difference in training time between SVM and Logistic Regression on high-dimensional features (CNN and combined) was somewhat unexpected and suggests that implementation details of the solvers play a significant role.
- The cost of PCA was relatively small and offset by the significant reduction in training time it enabled, making it a beneficial step for efficiency.

In summary, optimizing for accuracy points towards the SVM with combined features, while optimizing for efficiency favors Logistic Regression, particularly with combined or CNN features, offering a good balance between speed and competitive accuracy (91-92%).

7. Conclusion

In this project, we developed an image classification system to distinguish military and civilian vehicles using both classical and deep learning features. CNN embeddings significantly outperformed traditional features, achieving nearly 90% accuracy alone. Combining CNN, HOG, and RGB histograms boosted accuracy further, with the best-performing linear SVM model achieving 95.8% accuracy, and logistic regression closely following at 91.7%.

PCA effectively reduced dimensionality, improved computational efficiency, and maintained accuracy, while t-SNE visualizations confirmed the superior separability of CNN-based features compared to traditional features. The combined model made only minimal, understandable classification errors, highlighting its reliability.

We observed a clear trade-off between the high accuracy of SVM and the computational speed of logistic regression, making logistic regression suitable for frequent retraining or deployment on resource-limited systems.

Future improvements could involve ensemble methods, fine-tuning CNN architectures specifically for vehicle classification, incorporating spatial information through techniques like bag-of-visual-words, or exploring non-linear classifiers. Our approach successfully demonstrated the power of integrating traditional and deep learning methods for practical, accurate vehicle classification.

References

1. Gupta, P., Pareek, B., Singal, G., & Rao, D. V. (2021). Military and civilian vehicles classification. Mendeley Data, 1. – Dataset (Military and Civilian Vehicles Classification - Mendeley Data).

2. 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.

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