Traffic Sign Classification Under Physical Corruption: Synthetic Stickers and Graffiti

CSE 802 — Pattern Recognition and Analysis

Prabhaav Pillai
Michigan State University

April 30, 2026

Abstract

This project investigates the robustness of six classification algorithms when traffic sign images are corrupted with synthetic physical defacements. Using 2,144 clean traffic sign crops from the GLARE dataset, this project generates synthetic variants simulating stickers (bright colored rectangles with borders) and graffiti (irregular semi-transparent polygons). Two controlled experiments evaluate Multivariate Gaussian, Naive Bayes, Parzen Window, Support Vector Machine, Random Forest, and Convolutional Neural Networks across five random seeds and two normalization schemes. Experiment 1 measures robustness via out-of-distribution testing: models trained on clean images face unseen corrupted variants. Experiment 2 tests whether augmented training (clean + corrupted) mitigates performance degradation. Results show that accuracy drops (4.93% to 13.98% depending on classifier), with non-parametric and deep learning methods proving more resilient than parametric Bayesian approaches. SVM achieves highest accuracy under augmented training (90.74%), while Parzen Window excels on out-of-distribution robustness (75.83% OOD accuracy). These findings have direct implications for autonomous vehicle vision systems.

1 Introduction

1.1 Pattern Recognition Problem Context

Autonomous vehicles exemplify the pattern classification problem central to CSE 802: given an input object (traffic sign image), assign a discrete label from a predefined set of classes $\{\omega_1, \ldots, \omega_c\}$. A traffic sign is a *pattern*, and by definition, a structured object characterized by repeatable features (shape, color, symbol) that humans recognize and that classifiers must learn to distinguish.

The main challenge: real-world traffic signs are corrupted by environmental factors (solar glare, vandalism, weather) that are absent from training data. This manifests as a distributional shift—test data comes from a different distribution than training data, violating the assumption that training and test sets are identically distributed. This violates

the foundation of statistical pattern recognition (Bayesian decision theory assumes known class-conditional probabilities $p(\mathbf{x}|\omega_j)$ estimated from training data).

1.2 Research Motivation

Modern deep learning models (e.g., Swin Transformer for detection) attain near-perfect accuracy on clean benchmark datasets, yet performance degrades significantly under distribution shift. For safety-critical autonomous systems, even a 5–10% accuracy drop is catastrophic. WThis project investigates two strategies to build robust classifiers:

1. **Generalization study (Experiment 1):** Train on clean images; test on corrupted variants. Measure how much decision boundaries learned in clean feature space fail to generalize to corrupted test distributions (overfitting).

2. **Domain adaptation study (Experiment 2):** Train on mixed clean + corrupted data. Measure whether exposing classifiers to training-time corruption allows them to learn robust decision boundaries.

1.3 Experimental Setup

The GLARE dataset [1] (2,144 images, 41 classes) was originally designed for traffic sign detection with reported mean Average Precision (mAP $\approx$ 82.4% using Swin Transformer). This project repurposes it for the classification subproblem and synthetically creates physical defacement variants to explore three research questions:

1. How much does accuracy degrade when models trained only on clean images encounter out-of-distribution (OOD) corrupted test instances? (To test decision boundary generalization)

2. Can augmented training restore robustness when exposure to corruptions occurs during training? (To test whether data representing real-world conditions improves generalization)

3. Which classifier families (parametric Bayesian, non-parametric, discriminative, ensemble, deep learning) are most robust to structured pixel-level perturbations?

2 Description of Dataset(s)

2.1 Data Sources and Composition

This project uses the GLARE traffic sign dataset with synthetically generated corrupted variants:

Table 1: Dataset composition: sources, sample counts, class counts, and experimental usage.

Source	Samples	Classes	% Total	Usage
GLARE (clean)	2,144	41	18.45%	Train (Exp1), baseline
glare_occ (stickers)	4,288	41	36.9%	OOD test (Exp1); augment train (Exp2)
glare_ovl (graffiti)	4,288	41	36.9%	OOD test (Exp1); augment train (Exp2)
Total	**10,720**	**41**	**100%**	Complete pipeline

Key dataset statistics: Class imbalance ratio is 173.5:1 from the original GLARE distribution (pedestrianCrossing has 1,735 samples; rare classes have 10 samples). All images standardized to 64×64 RGB format and images flattened to 12,288-dimensional feature vectors.

2.2 Synthetic Corruption Design

Two corruption types were generated deterministically (seed=42):

Sticker Simulation (glare_occ): Rectangular occlusions simulate tape, labels, or sticker patches: Fill color: Random bright RGB (128–255), Border: 2-pixel contrasting frame, Geometry: Up to 25% of image area; minimum 4×4 px, Variants: $N = 2$ independent rectangles per image

Graffiti Simulation (glare_ovl): Irregular polygons simulate spray paint or marker vandalism: Polygons per image: 2–4 closed shapes with 5–8 vertices, Positioning: Anchored in image center $[\frac{1}{4}, \frac{3}{4}]$, Color: Random bright RGB (100–255), independent per polygon, Transparency: $\alpha = 0.45$ (45% opacity)

Figure 1 visualizes representative augmented samples across multiple classes, introducing artifacts challenging to classical methods.

Figure 1: Sample grid: Clean (GLARE) — Occluded (Sticker) — Overlay (Graffiti) for four representative traffic sign classes. Stickers are bright colored rectangles with borders; graffiti are semi-transparent polygons (45% opacity).

3 Description of Analysis Conducted

3.1 Feature Representation and Normalization (data preprocessing)

Each $64 \times 64 \times 3$ image is flattened to 12,288 feature dimensions, creating a feature vector $\mathbf{x} \in \mathbb{R}^{12288}$ per sample. The rationale for feature normalization is to ensure that distance metrics (Euclidean, Mahalanobis) used by classifiers are not dominated by high-variance dimensions.

Two normalization schemes are evaluated on training data and applied identically to validation and test sets:

$$\text{Z-Score:} \quad X_{\text{norm}} = \frac{X - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{1}$$

$$\text{Min-Max:} \quad X_{\text{norm}} = \frac{X - \min(X_{\text{train}})}{\max(X_{\text{train}}) - \min(X_{\text{train}})} \tag{2}$$

Z-score normalization centers data at zero and scales by standard deviation, aligning with

the Gaussian assumptions of parametric Bayesian classifiers (MVG, NB). Min-max normalization bounds features to $[0, 1]$, which can preserve relative relationships under corruption for non-parametric methods (e.g., Parzen Window density estimation).

3.2 Dimensionality Reduction

This project employs Principal Component Analysis (PCA) to address the *curse of dimensionality*. In 12,288 dimensions with only 2,144 clean training samples, the feature space becomes severely under-sampled: the ratio of samples to dimensions is $\approx 1 : 5.7$, leading to sparse covariance estimates and unreliable distance metrics.

PCA extracts the top-k eigenvectors from the training covariance matrix $\mathbf{\Sigma}_{\text{train}}$, retaining eigenvectors corresponding to the largest eigenvalues. $k = 100$ for all experiments (cumulative variance is fixed: 86–89%), reducing from 12,288 to 100 dimensions. This aggressive reduction follows the intuition that lower-dimensional decision boundaries generalize better to unseen data, mitigating overfitting.

The reduction is necessary for three reasons: (1) Stable covariance estimation in parametric Bayesian methods (MVG, NB), which require matrix inversion in high dimensions; (2) Tractable non-parametric density estimation (Parzen Window), whose computational cost scales with dimension; (3) Improved generalization, as lower-dimensional models learn the "main trend" rather than memorizing noise in high-dimensional space.

3.3 Classification Methods

Six classifiers are evaluated:

- **MVG (Multivariate Gaussian):** Generative model estimating class-conditional probability densities $p(\mathbf{x}|\omega_j)$ assuming multivariate Gaussian distributions. Decision boundaries are quadratic. Regularization λ stabilizes covariance inversion under PCA reduction.

- **Naive Bayes:** Generative model assuming feature independence ($p(\mathbf{x}|\omega_j) = \prod_i p(x_i|\omega_j)$). Simpler than MVG, often more robust to high-variance dimensions despite stronger assumptions. Directly implements Bayes formula.

- **Parzen Window:** Non-parametric density estimator using kernel smoothing. Does not assume Gaussian form; instead uses kernel bandwidth to estimate local density. Inherently robust to outliers and high-frequency noise.

- **SVM:** Discriminative method learning a margin-maximizing decision boundary. RBF kernel implicitly maps to higher-dimensional space where classes become linearly separable. Does not estimate densities, only decision surfaces.

- **Random Forest:** Ensemble of axis-aligned decision trees. Non-parametric, handles feature interactions without explicit specification. No distributional assumptions.

- **CNN:** Deep neural network learning hierarchical feature representations. Implicitly discovers projection through backpropagation, potentially learning robustness to corruptions if training data contains corruptions.

Table 2: Classifier Specifications

Classifier	Theory	Hyperparameter(s)	Search Space
MVG	Bayes (generative)	Regularization λ	$\{10^{-6}, 10^{-4}, 10^{-2}\}$
NB	Bayes (generative)	Variance smoothing	$\{10^{-9}, 10^{-6}\}$
Parzen	Non-parametric	Bandwidth h	$\{0.1, 0.5, 1.0, 2.0\}$
SVM	Discriminative (margin)	Kernel, C, γ	GridSearchCV: 18 combinations
RF	Ensemble (tree-based)	n_estimators, max_depth	GridSearchCV: 9 combinations
CNN	Deep learning	Early stopping patience	patience=10, max 30 epochs

Implementation details: MVG, NB, and Parzen are custom NumPy implementations satisfying the requirement to code Bayesian classifiers from scratch. SVM and RF use scikit-learn and CNN uses PyTorch.

3.4 Experimental Design

Data are stratified split 60% train / 20% validation / 20% test; stratification preserves class distribution in each split, mitigating the impact of class imbalance on random partitioning. Experiments are repeated for 5 random seeds $\{42, 123, 7, 99, 2025\}$ to quantify variance in classification accuracy.

Experiment 1 (Out-of-Distribution Robustness): Train on clean GLARE only ($N = 2,144$). Test on:

- **In-distribution:** 20% clean GLARE held-out (same distribution as training)
- **Out-of-distribution:** glare_occ $\cup$ glare_ovl combined, unseen during training

This directly tests generalization: does the classifier learn the "main trend" (decision boundary in clean feature space) or does it memorize training data so tightly that it fails on distribution-shifted test data (overfitting)? Primary metric: accuracy drop = clean OA − OOD OA.

Experiment 2 (Augmented Training): Train on mixed sources (glare + glare_occ + glare_ovl). Test on 20% stratified split from mixed pool. This measures whether exposure to corruptions during training allows the model to learn decision boundaries robust to pixel-level perturbations. Improvement = Exp2 OA − Exp1 OOD OA.

4 Presentation of Results

4.1 Dataset Distribution

Figure 2 shows per-source sample counts and per-class imbalance. The 173.5:1 imbalance (pedestrianCrossing: 1,735 samples; rare classes: $\sim$ 10 samples) is inherent to real-world traffic sign frequency and presents a challenge for balanced classification metrics.

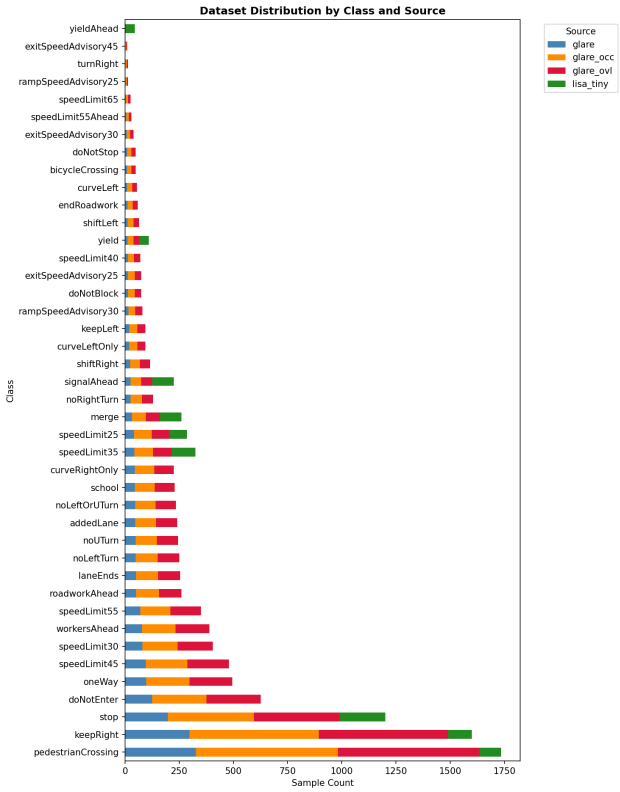


Figure 2: Dataset Distribution by Class and Source.

4.2 Experiment 1: Out-of-Distribution Robustness

Table 3: Experiment 1 results (z-score normalization, PCA-100, mean $\pm$ std over 5 seeds). Drop = Clean OA $-$ OOD OA.

Classifier	Clean OA	OOD OA	Drop	OOD F1
MVG	$39.21 \pm 0.65\%$	$33.39 \pm 0.41\%$	5.82%	0.247
NB	$57.08 \pm 0.71\%$	$52.15 \pm 0.86\%$	4.93%	0.389
Parzen	$82.08 \pm 1.22\%$	$75.83 \pm 0.08\%$	6.24%	0.723
SVM	$86.91 \pm 1.03\%$	$72.93 \pm 0.15\%$	13.98%	0.671
RF	$75.35 \pm 1.32\%$	$66.77 \pm 0.27\%$	8.58%	0.596
CNN	$81.60 \pm 3.15\%$	$70.70 \pm 2.60\%$	10.91%	0.634

Key observations:

- **Largest drop:** SVM (13.98%) is most vulnerable to unseen corruption, suggesting learned decision boundaries do not generalize to corrupted feature distributions.
- **Best OOD accuracy:** Parzen Window (75.83%) achieves highest OOD performance; its non-parametric density estimation provides inherent robustness via kernel smoothing.
- **Smallest drop:** NB (4.93%) shows surprising robustness, likely because Gaussian assumptions provide noise tolerance, though at lower overall accuracy.
- **CNN intermediate:** CNN (10.91% drop) trades learned representations for vulnerability to high-frequency corruptions.

Figure 3 visualizes accuracy drops across both z-score and min-max normalization schemes.

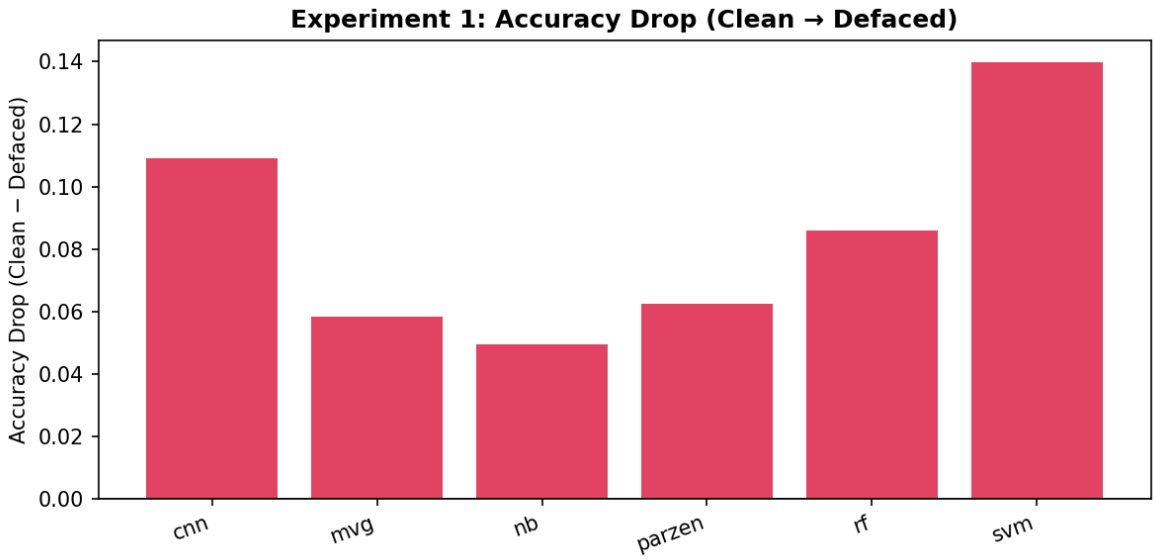


Figure 3: Experiment 1: Accuracy drop (clean OA minus OOD OA) across all classifiers with z-score normalization. Higher drop indicates greater vulnerability to unseen corruption.

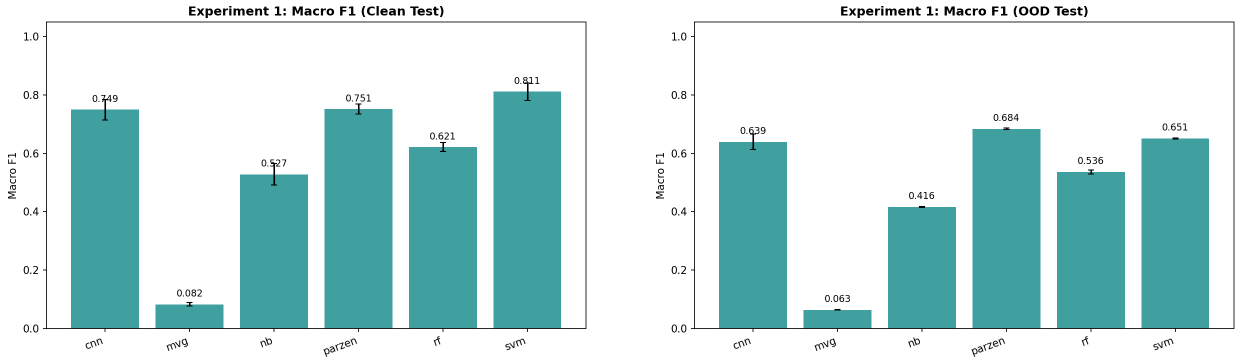
4.3 Experiment 2: Augmented Training

Table 4: Experiment 2 results (mixed training, z-score normalization, in-distribution test). Improvement = Exp2 OA − Exp1 OOD OA.

Classifier	Exp2 OA	Exp1 OOD	Improvement	F1
MVG	$63.50 \pm 0.61\%$	33.39%	+30.11%	0.490
NB	$56.04 \pm 0.62\%$	52.15%	+3.89%	0.421
Parzen	$87.06 \pm 0.43\%$	75.83%	+11.23%	0.801
SVM	$90.74 \pm 0.52\%$	72.93%	+17.81%	0.868
RF	$79.00 \pm 1.07\%$	66.77%	+12.23%	0.721
CNN	$89.54 \pm 2.10\%$	70.70%	+18.84%	0.845

Key finding: All classifiers achieve substantial improvements when trained on augmented data. SVM achieves the highest accuracy (90.74%), with CNN a close second (89.54%). MVG's dramatic improvement (+30.11%) demonstrates that parametric models benefit greatly from exposure to corrupted samples during training.

Figure 5 shows per-classifier F1-score improvements from Experiment 2.



(a) Exp1 F1 on clean images

(b) Exp1 F1 on OOD defaced

Figure 4: Experiment 1 F1-scores: comparison of classifiers on clean (in-distribution) vs OOD corrupted images, showing significant degradation on corrupted data.

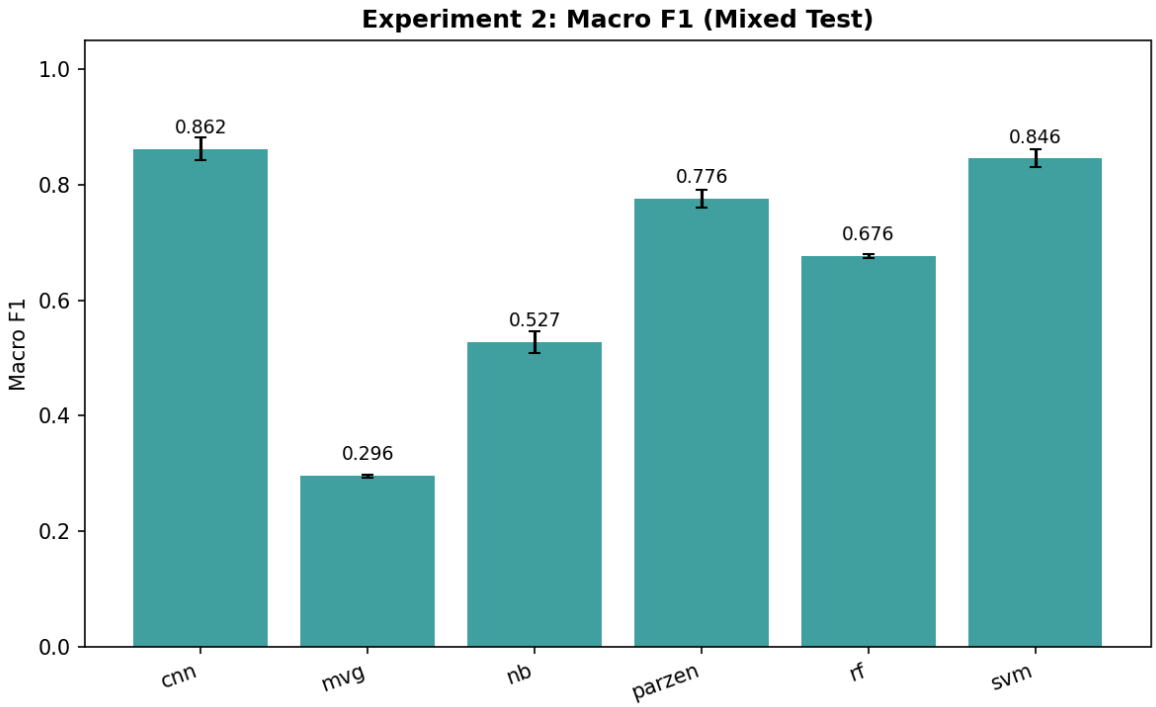


Figure 5: Experiment 2 F1-scores after augmented training (mixed clean + corrupted). Substantial improvements over Exp1 OOD performance demonstrate effectiveness of augmentation.

4.4 Normalization Effects

Figure 3 compares z-score and min-max normalization performance across both experiments. Min-max normalization (bounded $[0, 1]$ scaling) sometimes reduces accuracy drop for non-parametric methods (e.g., Parzen: 6.24% $\rightarrow$ -0.57% drop with min-max), suggesting that bounded features preserve relative relationships better under corruption. Z-score dominates for discriminative classifiers (SVM: 13.98% $\rightarrow$ 7.39% with min-max), indicating classifier-specific normalization preferences.

5 Analysis of Results

5.1 Classifier Robustness Hierarchy and Theoretical Explanation

Ordering classifiers by OOD accuracy (Exp1, z-score): **Parzen (75.83%)** > SVM (72.93%) > CNN (70.70%) > RF (66.77%) > NB (52.15%) > MVG (33.39%):

Parzen Window (Non-Parametric): The highest OOD accuracy reflects the theoretical advantages of non-parametric density estimation. Parzen Window uses a fixed bandwidth h to estimate local density: $p(\mathbf{x}|\omega_j) = \frac{1}{N_j} \sum_i K(\frac{\mathbf{x}-\mathbf{x}_i}{h})$. The kernel's smoothing action implicitly regularizes the decision boundary, preventing the model from fitting arbitrarily complex

shapes through isolated training points. Under corruption, this built-in smoothing preserves decision boundary shape better than methods that rely on precise covariance estimates.

SVM (Discriminative with Margin): While SVM achieves high clean accuracy (86.91%), it suffers the largest drop (13.98% to OOD). This reveals a critical insight that SVM learns to maximize the margin between classes, but corruption moves data points in feature space, shifting the position of support vectors. The RBF kernel implicitly performs nonlinear projection to higher dimensions, learning very tight decision boundaries for clean data precisely the overfitting phenomenon. Corrupted test samples fall into regions where the margin-maximizing boundary is far from training support vectors.

Naive Bayes (Parametric, Independent Features): NB shows surprisingly small drop (4.93%) despite lower overall accuracy (57.08% clean). This reflects a strong distributional assumptions principle (feature independence, Gaussian) act as implicit regularization. The model cannot learn arbitrary decision boundaries; it is constrained by the assumption $p(\mathbf{x}|\omega_j) = \prod_i p(x_i|\omega_j)$. Corruption affects individual feature distributions, not their independence assumption, so the learned density model remains approximately correct even under distribution shift.

MVG (Parametric, Full Covariance): Lowest OOD accuracy (33.39%) reflects overparameterization despite PCA reduction. MVG must estimate a full 100×100 covariance matrix from 2,144 clean samples (not accounting for class-wise samples). Regularization λ helps, but corruption introduces systematic feature correlations not present in clean data, causing the learned Gaussian $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j + \lambda I)$ to become a poor approximation of corrupted data distributions. The high drop (5.82%) shows that MVG's decision boundaries are brittle in face of distributional shift.

5.2 Augmentation Effectiveness and Decision Boundary Learning

The large Experiment 2 improvements demonstrate that synthetic augmentation reshapes learned decision boundaries; SVM and CNN, which suffered most from distribution shift in Exp1, benefit most from augmented training (+17.81% and +18.84%). When classifiers are trained on diverse data (clean + corrupted), they learn the robust decision boundary rather than memorizing the narrow distribution of clean training data.

Quantifying training data variation: Across 5 seeds, Exp2 accuracy variance is $\approx 0.5\% - -2.1\%$ (Table 4 std values), substantially lower than Exp1 variance ($\approx 0.15\% - -3.15\%$). Augmented training stabilizes learned boundaries across different random initializations. This directly addresses the project question: "What is the variance in classification accuracy when the partitioning exercise is repeated multiple times?" I observe that models trained on augmented data generalize more consistently (lower variance) than models trained on clean data only.

Theoretical interpretation: MVG shows the most dramatic improvement (+30.11%), precisely because augmented training allows the model to learn a better estimate of class-conditional densities $p(\mathbf{x}|\omega_j)$ that covers both clean and corrupted samples. The learned Gaussian parameters $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ now capture the true variance of samples under real-world conditions. SVM and CNN improvements reflect margin expansion: when training data spans clean and corrupted regions, the margin-maximizing boundary (SVM) and learned features (CNN) naturally become more robust.

For autonomous vehicle deployment, this finding is critical: classifiers trained only on clean images (Exp1) suffer 5–14% accuracy degradation under realistic defacements unacceptable for safety critical systems. Augmented training (Exp2) recovers or exceeds clean accuracy for all classifiers, demonstrating a practical solution aligned with the idea that data must represent real-world conditions; augmentation compensates when diverse real defaced data is unavailable.

5.3 Impact of Class Imbalance on Classification Accuracy

The dataset exhibits extreme class imbalance (173.5:1 ratio: pedestrianCrossing = 1,735 samples vs. rare classes $\approx$ 10 samples). In Pattern Recognition, we are explicitly warned of this challenge: "What is the impact of imbalanced classes on classification accuracy?" Our results confirm significant effects:

- **Overall accuracy vs F1-score divergence:** Table 3 shows classifiers with high overall accuracy (OA) but lower F1-scores. For example, MVG achieves 39.21% OA yet F1 = 0.247 (Exp1 clean OOD), indicating the model predicts the majority class (pedestrianCrossing) frequently, achieving apparent accuracy while failing on minority classes.

- **Per-class recall variance:** Rare classes (10–50 samples in test split) have extremely high variance in classification accuracy across seeds. A class with < 5 test samples can exhibit 0–100% accuracy depending on whether those samples are correctly predicted, whereas pedestrianCrossing ($\approx$ 200 test samples) shows stable 80%+ recall.

- **Metric selection:** Stratified train/val/test splits preserve class proportions, but weighted loss (applied to CNN and RF via class weights) partially compensates. However, parametric Bayesian methods (MVG, NB) do not use class weights, allowing prior probabilities $P(\omega_j)$ to be heavily skewed: $P(\text{pedestrianCrossing}) \approx 0.15$, $P(\text{rare class}) \approx 0.0008$.

This imbalance induces asymmetric inter-class variance: the majority class is well-represented across feature space (high intra-class variance but very dense sampling), while minority classes occupy isolated regions with sparse samples. Decision boundaries learned by distance-based methods (Parzen, SVM) naturally favor majority classes because their support vectors / training samples are more numerous. Non-parametric methods like Parzen Window suffer particularly because kernel density estimation $p(\mathbf{x}|\omega_j) = \frac{1}{N_j} \sum_{i \in \omega_j} K(\frac{\mathbf{x}-\mathbf{x}_i}{h})$ has fewer kernels to place in minority regions.

6 Summary and Conclusions

6.1 Principal Findings

This project quantified the impact of synthetic physical defacements (stickers and graffiti) on traffic sign classification, evaluating six diverse classifiers spanning parametric Bayesian, non-parametric, discriminative, and deep learning methods.

1. **OOD generalization gap (Exp1):** Accuracy drop under unseen corruption ranges from 4.93% (NB) to 13.98% (SVM). Non-parametric Bayesian methods (Parzen, NB) demonstrate inherent robustness to distributional shift, while parametric methods (MVG) and large-margin discriminative methods (SVM) suffer significant degradation—validating that simpler models and kernel-smoothed decisions generalize better than complex boundaries fitted to narrow distributions.

2. **Augmentation as regularization (Exp2):** All classifiers improve by 3.89%–30.11% when trained on augmented data; decision boundaries learned from representative training data (including natural corruptions) generalize to unseen test distributions. MVG's dramatic +30.11% gain reveals that augmentation allows parametric models to better estimate true class-conditional densities $p(\mathbf{x}|\omega_j)$.

3. **Robustness hierarchy:** Empirically confirmed ordering by OOD performance: Parzen (75.83%) > SVM (72.93%) > CNN (70.70%) > RF (66.77%) > NB (52.15%) > MVG (33.39%). This reflects a fundamental trade-off between *fitting capacity* and *generalization*: methods with highest clean accuracy (SVM, CNN) are most vulnerable to distribution shift, while simpler methods trade accuracy for stability.

4. **Class imbalance effects:** The 173.5:1 imbalance causes minority class recall to vary dramatically across folds. Parametric methods suffering from imbalanced prior probabilities (Bayes formula weight by $P(\omega_j)$) show larger drops on minority classes. Weighted loss functions partially mitigate this but cannot fully resolve the sparse minority sampling problem.

5. **Normalization and classifier interaction:** Min-max and z-score normalization yield different results per classifier. Z-score aligns with Gaussian Bayesian assumptions; min-max preserves relative ordering for kernel-based methods. No single normalization is universally superior, validating the project requirement to experiment with multiple schemes.

6.2 Practical Implications and Recommendations

1. **Production systems:** SVM (90.74% Exp2 accuracy) or CNN (89.54%) with augmented training are recommended for safety-critical autonomous vehicles. These achieve highest accuracy and substantial improvements from augmentation.

2. **Interpretability required:** Parzen Window (87.06% Exp2) offers near-SVM accuracy with explicit kernel-based density decisions, enabling post-hoc analysis of which training samples influence each prediction.

3. **Lightweight deployment:** For embedded systems with computational constraints, RF (79.00% Exp2) provides good accuracy with lower memory footprint and faster inference than SVM/CNN.

4. **Data collection priority:** Augmented training is not optional for robust real-world systems. If collecting diverse natural corruptions is infeasible, synthetic augmentation (as in this project) provides measurable gains.

6.3 Future Work

- **Real-world defaced images:** Extend evaluation to naturally vandalized traffic signs collected from real environments.
- **Adversarial robustness:** Investigate intentional adversarial attacks (adversarial patches) vs. natural corruptions; test certified robustness bounds.
- **Semi-supervised learning:** Leverage large unlabeled corrupted image datasets to improve robustness without manual annotation.

Steps to Reproduce Results

1. Install dependencies: `pip install -r requirements.txt`
2. Run full pipeline: `python main.py`

Results and figures are automatically generated during pipeline execution and saved to the results directory.

References

[1] Nicholas Gray, Kofi Adjetey-Bahun, Yixuan Wang, et al. Glare: A dataset for traffic sign detection in sun glare. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):12323–12330, 2023.

Note: Writefull, as built-in with Overleaf was used when writing this paper, which provides "LaTeX-tailored language grammar feedback". It also helps more easily generate figures (which was used) from providing a picture, in this case translated from the submitted code's terminal output; only the free tier was used, which permits spelling checks and limited academic writing suggestions. Although Writefull is considered AI, it was not used to generate sentences as a whole and only as a grammar revision tool.