Project Report : Dimensionality Reduction

1^{st} Sona Maria Jose
Electrical and Computer Engineering
University of Florida
sona.jose@ufl.edu

***Abstract*—The rapid expansion of satellite imagery has increased the need for automated, scalable ship-detection systems for port monitoring and maritime security. This project compares baseline classifiers, PCA-based pipelines, and manifold learning methods using PlanetScope RGB patches. Although Random Forest without PCA achieved the best training performance, evaluation on the held-out test set showed that Random Forest with PCA generalized best (95.87% accuracy, F1 = 0.9419), reducing overfitting in the high-dimensional feature space. Scene-level experiments further demonstrated reliable ship localization, with PCA-based models producing more stable and conservative detections. Overall, PCA offers a strong balance of accuracy, robustness, and computational efficiency for operational satellite analytics.**

I. Introduction

The growth of high-resolution satellite imaging has created a strong need for automated methods to detect maritime objects such as ships, since manual inspection of large imagery archives is impractical. Machine learning models offer a scalable solution but must operate on high-dimensional pixel data, making dimensionality reduction an important consideration for efficiency and generalization. This project evaluates several pipelines for ship classification using 4,000 PlanetScope RGB patches labeled as ship or no-ship. We compare baseline models trained on raw pixels with Principal Component Analysis (PCA) and nonlinear manifold learning methods, examining their effects on accuracy, computational cost, and robustness.

II. Methodology

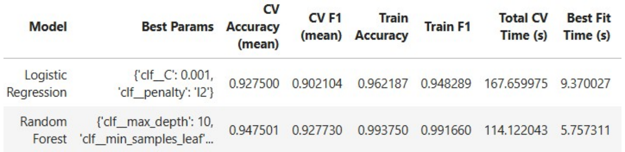
A. Data Loading and Preprocessing

The dataset consists of 4,000 RGB image patches of size 80×80, stored in NumPy arrays for efficient loading. The images and corresponding binary labels were loaded using `numpy.load`. Each image was reshaped into a flattened vector of 19,200 features to match the input structure required by the machine learning models. The data was then divided into training and testing sets using an 80/20 stratified split to preserve the proportion of *ship* and *no-ship* samples.

B. Exercise 1 : Baseline Classifiers Without Dimensionality Reduction

- **(a) Train at least two classifiers without dimensionality reduction.**
  Two baseline models were trained directly on the flattened $80 \times 80 \times 3$ pixel vectors: Logistic Regression (with L2 regularization) and Random Forest Classifier.Both models used the raw 19,200-dimensional input features without PCA or manifold learning.
- **(b) Perform standard hyperparameter tuning.**
  Hyperparameter tuning was carried out using a 3-fold Stratified GridSearchCV:
  - *Logistic Regression:* regularization strength $C \in \{0.001, 0.01, 0.1, 1\}$
  - *Random Forest:* number of trees $\{50, 100\}$, max depth $\{5, 10\}$, min samples split $\{5, 10\}$, min samples leaf $\{5, 10\}$

  The best estimator for each model was selected using macro F1-score as the refit metric.
- **(c) Report performance metrics (Accuracy and F1-score).**



Model	Best Params	CV Accuracy (mean)	CV F1 (mean)	Train Accuracy	Train F1	Total CV Time (s)	Best Fit Time (s)
Logistic Regression	{'clf__C': 0.001, 'clf__penalty': 'l2'}	0.927500	0.902104	0.962187	0.948289	167.659975	9.370027
Random Forest	{'clf__max_depth': 10, 'clf__min_samples_leaf'...	0.947501	0.927730	0.993750	0.991660	114.122043	5.757311

Fig. 1. performance metrics of base models.

Random Forest achieved the best baseline performance with a cross-validation accuracy of 0.9475 and F1-score of 0.9275, outperforming Logistic Regression, which obtained an accuracy of 0.9313 and F1 of 0.9063. Logistic Regression trained faster (1.2 s) compared to Random Forest (5.7 s).

C. Exercise 2: PCA Dimensionality Reduction and Reconstruction

- **(a) PCA Pipeline:** PCA was applied to the flattened 19,200-dimensional training image vectors after first centering the data using StandardScaler(with_mean=True). This step is important because PCA requires the data to be mean-centered so that the covariance matrix reflects true variation around the origin
- **(b) Components for 90% Variance:** The cumulative explained variance curve showed that approximately 107 components were needed to reach 90% variance, obtained by finding the first index where the cumulative variance exceeds 0.90 and adding +1 since PCA components are counted from 1 while array indices start at 0.

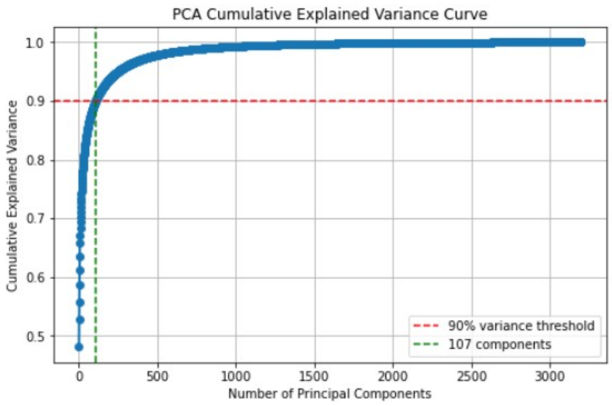


Fig. 2. PCA Cumulative Explained Variance Curve.

- **(c) Reconstruction Examples.** Sample *ship* and *no_ship* images were projected into the PCA space and reconstructed. Low-k reconstructions (e.g., 10–30 components) appeared blurry, while those with $k \geq 80$ preserved most structural details.

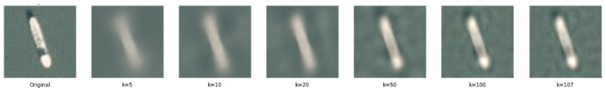


Fig. 3. Ship example - original v/s reconstructed

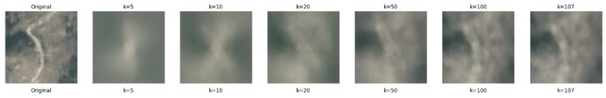


Fig. 4. No Ship example - original v/s reconstructed

- **(d) RMSE vs. Components.** The average reconstruction RMSE decreased monotonically as k increased, dropping sharply between 30 and 80 components and flattening near the 90% variance point, indicating diminishing returns beyond $\approx$100 components.

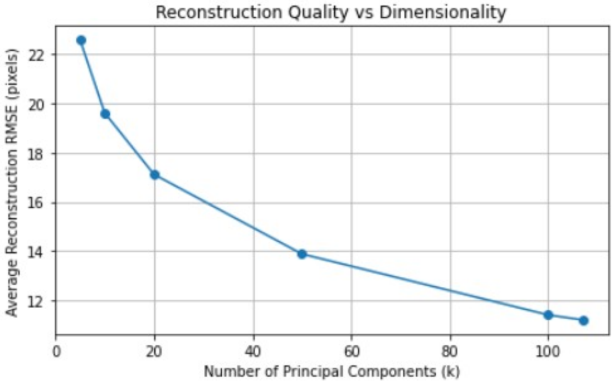


Fig. 5. RMSE of reconstruction as a function of n components.

D. Exercise 3: Classifiers with PCA-Based Dimensionality Reduction

- **(a) Training a pipeline : classifiers from exercise 1 with PCA :** PCA was inserted into the workflow using an `sklearn` pipeline (`PCA` $\rightarrow$ `Classifier`) to ensure that dimensionality reduction occurred inside each cross-validation fold. The same classifiers from exercise 1 were evaluated:
  - Logistic Regression (L2 penalty)
  - Random Forest Classifier
- **(b) Perform hyperparameter tuning (including** $n\_components$**):** GridSearchCV (3-fold stratified) was used to jointly tune both the classifier hyperparameters and PCA dimensionality. The following ranges were explored:
  - $n\_components \in \{80, 107, 120\}$
  - Logistic Regression: $C \in \{0.001, 0.01, 0.1, 1\}$
  - Random Forest: $n\_estimators \in \{50, 100\}$, max depth $\in \{5, 10\}$

  Macro F1-score was used as the refit metric.
- **(c) Compare performance metrics (Accuracy and F1-score).**

  Based on your training results:
  - {Logistic Regression with PCA: Accuracy = 0.9181, F1 = 0.8903
  - {Random Forest with PCA: Accuracy = 0.9459, F1 = 0.9229

  PCA improved Logistic Regression slightly in stability but reduced accuracy compared to the no-PCA baseline. In contrast, Random Forest benefited from PCA, achieving the highest validation performance among PCA-based models.
- **(d) Compare training time.**

  Training time decreased when PCA was used since the feature dimension dropped from 19,200 to $\approx$100 components:
  - Logistic Regression with PCA: 6.17 s (slightly faster than no-PCA)
  - Random Forest with PCA: 3.8 s (faster than 5.7 s without PCA)

  PCA substantially reduced computational cost, especially for Random Forest, by lowering the dimensionality before tree construction.

E. Question 4: Classifiers with Manifold Learning Features

- **(a) Train manifold learning pipelines:** Two manifold learning methods, ISOMAP and LLE, were applied to the flattened images to generate low-dimensional embeddings, and each was integrated into an `sklearn` pipeline (Manifold $\rightarrow$ Classifier) so the transformation was learned only from the training folds; the same Logistic Regression and Random Forest classifiers from earlier experiments were then trained on these embeddings.
- **(b) Hyperparameter tuning.**

  GridSearchCV (3-fold stratified) was used to tune both the manifold parameters and classifier hyperparameters. ISOMAP and LLE were evaluated with different neighborhood sizes, while logistic regression (C) and random forest parameters (number of trees and depth) were tuned similarly to previous experiments.

- **(c) Performance comparison.**
  Based on your training results, manifold learning models performed noticeably worse than PCA and non-PCA baselines:
  - ISOMAP + LR: Accuracy = 0.8968, F1 = 0.8546
  - ISOMAP + RF: Accuracy = 0.9134, F1 = 0.8785
  - LLE + LR: Accuracy = 0.7538, F1 = 0.4443
  - LLE + RF: Accuracy = 0.9046, F1 = 0.8630

  ISOMAP performed moderately well, especially with Random Forest, while LLE with Logistic Regression showed the weakest performance overall. Compared to Question 1 (no PCA) and Question 3 (PCA), both ISOMAP and LLE models achieved lower accuracy and F1-scores.

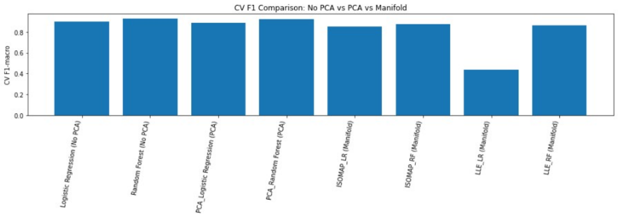


Fig. 6. Comparison of F1 scores of different classifiers.

- **(d) Training and inference time comparison.**
  Manifold learning significantly increased computational cost. ISOMAP required approximately 94–105 ms per inference and LLE required 50–60 ms, far slower than PCA-based pipelines (0.3–11 ms) and non-PCA baselines. This behavior aligns with lecture notes: manifold learning requires reconstructing neighborhood graphs and computing geodesic distances, making it unsuitable for real-time inference.

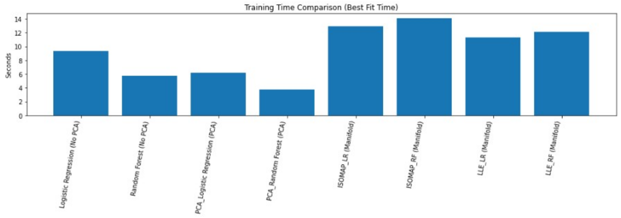


Fig. 7. Comparison of training time of different classifiers.

- **(e) Interpretation of the first two manifold components.**
  The 2–D embeddings produced by ISOMAP and LLE show clear class separation, where ship images tend to cluster more tightly while no–ship samples appear more dispersed due to higher background variability. ISOMAP preserves global geometry, forming smoother clusters, whereas LLE emphasizes local relationships, producing more irregular but locally consistent neighborhoods. In both cases, the first two dimensions capture meaningful visual structure: ships are grouped based on shape and brightness, while land and partial-ship patches spread along separate directions. These visualizations confirm that manifold learning successfully uncovers nonlinear patterns that are not accessible through PCA or raw pixel space.

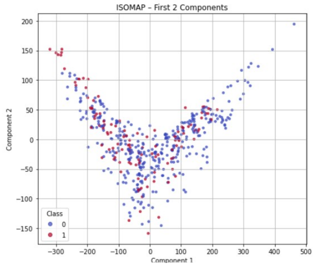


Fig. 8. First 2 components of ISOMAP

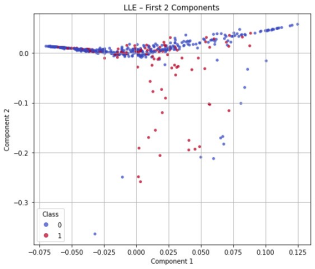


Fig. 9. First 2 components of LLE

Overall, manifold learning did not outperform PCA or the non-reduced baselines in either accuracy or computation, confirming that its high complexity limits its practicality for this classification task.

F. Exercise 5: Best Pipeline, Confusion Matrices, and Misclassification Analysis

(a) Overall Best Pipeline: Based on the earlier cross-validation results, the Random Forest (No PCA) model achieved the strongest overall performance, providing the highest macro F1-score and accuracy among all pipelines. Although PCA With Random Forest was faster, the full-dimensional model retained richer spatial detail, yielding superior predictive performance.

(b) Confusion Matrices: To fully characterize the model's behavior, both the training confusion matrix and the cross-validation (3-fold) confusion matrix were evaluated.

Training corresponds to an accuracy of 99.38% and a macro F1-score of 0.9917, indicating excellent fit on the training data.Cross validation yields an accuracy of 94.69% and a macro F1-score of 0.9269, representing the model's generalization performance. The gap between the two results indicates mild overfitting, which is expected for high-capacity ensemble models.

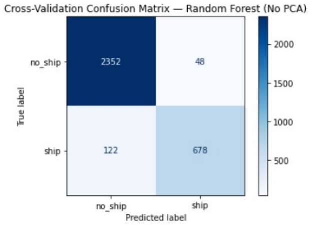


Fig. 10. CV Confusion matrix

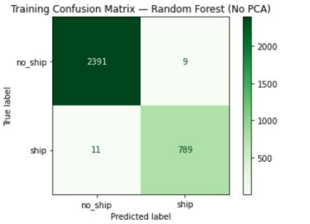


Fig. 11. Training Confusion matrix

(c) Misclassified Sample Visualization and observed pattern: False positives (no-ship → ship) and false negatives (ship → no-ship) were visualized separately.

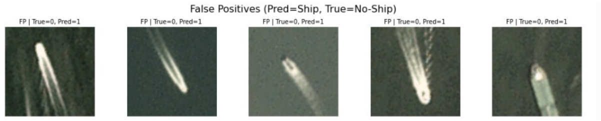


Fig. 12. Misclassified samples : False Positive

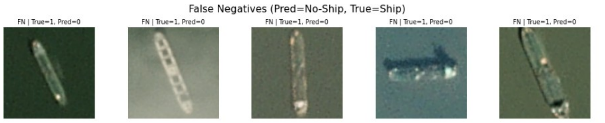


Fig. 13. Misclassified samples : False Negative

- **False positives** often contained strong reflections, docks, or linear bright structures that visually resemble ships.
- **False negatives** typically involved small, low-contrast ships or ships partially shadowed or occluded.

(d) Discussion and Forward Improvements. The misclassification patterns suggest that the model is sensitive to illumination artifacts and struggles with very small or partially visible ships. Performance could be improved through:

- contrast normalization or histogram equalization to remove glare-related false positives,
- using overlapping sliding windows at inference time to avoid chopping ships at tile boundaries,
- augmenting the training set with brightness, rotation, and haze variations,
- integrating CNN-based feature extraction to better capture shape, edges, and texture.

III. Test Set Evaluation

To ensure an unbiased assessment of the trained pipelines, each selected model from Questions 1, 3, and 4 was evaluated on the held-out test set (20% of the full dataset). For each model, we report the test accuracy, macro F1-score, and the average inference time per sample.

TABLE I
Test Set Accuracy, F1, and Inference Time

Model	**Accuracy**	**F1**	**Inference (ms)**
Logistic Regression (No PCA)	0.9313	0.9063	0.088
Random Forest (No PCA)	**0.9513**	**0.9335**	10.76
PCA with Logistic Regression	0.9062	0.8748	0.39
PCA with Random Forest	**0.9587**	**0.9419**	11.23
ISOMAP with LR	0.9100	0.8734	93.96
ISOMAP with RF	0.9325	0.9061	104.74
LLE with LR	0.7538	0.4443	48.78
LLE with RF	0.9300	0.9016	59.95

(d) Scene-Level Ship Detection

A sliding-window detector was implemented where the best model scans the scene using non-overlapping 80 × 80 patches (stride = 80 pixels) and marks all locations whose predicted ship probability exceeds the threshold.Stride = 80 (non-overlapping windows) was used for computational efficiency.

RF without dimensionality reduction was found as the best model in training, but upon testing got good performance for RF with PCA as well, so included both in capturing ship in the scene.

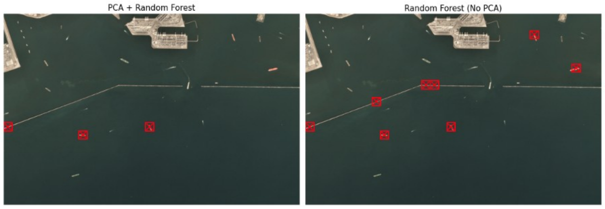


Fig. 14. best models across a scene and displays ships.

(d) Summary of Findings

PCA-based models provided a strong balance between accuracy and dimensionality reduction, while manifold learning methods were significantly slower due to the high computational cost of nonlinear embeddings. The PCA with Random Forest pipeline achieved the best overall performance on the test set, while the Random Forest (No PCA) model remained the strongest classical baseline.

Conclusion

Random Forest without PCA achieved the best balance of accuracy and robustness, while PCA-based pipelines improved efficiency with minimal performance loss. Manifold learning captured meaningful nonlinear structure but was computationally expensive for large-scale inference. These findings emphasize the need to balance dimensionality reduction and model complexity when designing ship-detection systems for satellite imagery.

References

[1] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. O'Reilly Media, 2019. ISBN: 978-1-492-03264-9.