A Comprehensive Review of Machine Learning Pipelines for Ship Detection in Satellite Images

Varun Amitabh
Electrical and Computer Engineering
University of Florida
Gainesville, USA
amitabh.varun@ufl.edu

***Abstract*—This paper evaluates various machine learning pipelines for detecting ships in satellite images by employing multiple classifiers(SVM and Random Forest), dimensionality reduction techniques, and a sliding window approach as an application for the best performing model. The work aims at finding the best method with good generalization on unseen data and provides practical insights into its deployment in the real world. Amongst models evaluated, the pipeline using Support Vector Machine(SVM) with PCA performed the best. The paper also discusses dataset characteristics, methodologies for preprocessing, model selection, and ways to resolve misclassifications.**

***Index Terms*—machine learning pipelines, classifiers, generalization, preprocessing**

I. Introduction

Ship detection from satellite images is an important task, with applications in maritime surveillance, environmental monitoring, and border security. It is a challenging problem because of variable image resolutions, occlusions, and background clutter. With the advantage of a binary classification dataset, this work discusses and evaluates multiple machine learning pipelines toward the optimal solution in accomplishing the efficient detection of ships with high accuracy. In this, three pipelines were considered: simple SVM and RF classifiers, Support Vector Machine with PCA, Random Forest with PCA, and the dimensionality-reduction-based pipelines using Isomap. A sliding window approach is then utilized to extend the model to larger scene images based on application purposes.

These results aim at informative guidance toward deployment strategies in real-world applications while also being targeted towards business persons by giving a proper narration of methodology, results, and actionable insights.

II. Dataset Description

The Ship Detection Dataset is used for identifying big ships in satellite images. There is an ever-growing quantity of satellite images every day. A manual review becomes impossible, and this automatically means that detection automation is crucial for various use cases, including port monitoring, supply chain analysis, and maritime surveillance.

This dataset encompasses imagery derived from PlanetScope satellite imagery captured over the San Francisco Bay and San Pedro Bay areas of California. Images are extracted from full-frame visual scenes, then orthorectified to a 3-meter pixel resolution to ensure geometric correction for consistent image scaling and alignment; hence, they are reliable on which an analysis can be based. It consists of 4000 images divided into 3000 being of the 'no-ship' class and 1000 belonging to the 'ships' class. The appearances here are very different due to the changes in size, orientation, and atmospheric conditions. While diverse backgrounds of the class are simulated in real situations in the "no-ship" class. The Scenes directory comprises 8 images that test the performance of classification models that have been trained to spot ships in an actual environment, also demonstrating surrounding features for true positives, false positives, and false negatives.

III. Methodology

The goal is to get a strong classifier of ships, both on the 80×80 images and on the larger scene images. Key steps include the following:

- Preprocessing and dimensionality reduction.
- Classifier optimisation with grid search.
- Sliding window implementation for scene image evaluation.

For the sake of comparison, we first chose two classifiers without dimensionality reduction and performed hyperparamter tuning on the training dataset. For better understanding and interpretability, accuracy and F-1 score have been chosen has the metrics to evaluate on all the trained models, alongwith noting down the time taken to train the best fit model(s) on the training dataset, to form a qualitative comparison.

A. Data Preprocessing

Data preprocessing, an essential part of any training pipeline learning, was carried out on the dataset in different capacities, based on the various configurations to be evaluated. A common data pre processing step used in all of the pipelines was the standard scaler, to standardize or normalize the pixel values to a common scale before applying any other pre processing step. The different or varying values cause problems while training a model and can lead to errors, hence to mitigate this and make the process streamlined and efficient, pre processing is done to prepare the data to ensure the model(s) perform optimally and generalize well to unseen data.

To ensure data compatibility with different algorithms, the following pre processing steps were applied for the various configurations:

B. Classifiers

The two classifiers used here for classification of the dataset are SVM and Random Forest(RF).

RF has an ensemble of trees that can deal with the imbalanced classes either by assigning appropriate weights to classes or by averaging in the inherent mechanism, reducing bias toward the majority class. RF models complex relationships between pixel-level features through several nonlinear decision trees, which can be very useful for identifying varied ship shapes, sizes, and orientations. Besides, because of the average of multiple trees and utilization of bootstrapped samples, RF reduces overfitting even in the existence of noisy or redundant features.

Talking about SVM, the "no-ship" class will include partial ships and bright pixels, which may be challenging to separate from the "ship" class. For instance, in SVM, the kernel trick-the RBF kernel-can map these points that may overlap into higher dimensions to improve separability. SVM also includes a regularization parameter, which will make a balance between low training error and generalization to unseen data. This is important for avoiding overfitting with a noisy dataset.

The training pipelines for both these models included a pre processing step of normalising the dataset and including various hyperparameters which could be tuned based on the performance and specific classifier. After an exhaustive grid search on the hyperparameters for both the classifiers, the results produced by them were as follows:

For SVM: The Training Accuracy was 0.999375, with an F1 Score of 0.99875. The high performance on the training dataset indicates a high level of overfitting wherein the model most likely is learning the noise present in the dataset as well. This would translate in bad generalisation of the model on new unseen data. As can be seen, when evaluating on the test dataset, the SVM Test Accuracy was 0.96875, with an F1 Score of 0.936386. There is a visible drop in performance for test dataset, but the model is still performing considerably well on the test dataset.

For RF: The Training Accuracy was 0.9996875 with an F1 Score of 0.999375. As before, this model also shows signs of high overfitting on the train set. The test accuracy reported was 0.95375 with an F1 Score of 0.90389. The model performs somewhat lower on the unseen data, though still robust.

C. PCA

The motivation behind PCA is that it acts as a good pre-processing step before applying any learning algorithm on the dataset. PCA essentially reduces the dimensions of the data while preserving the performance of the classifier. The feature size for every 80x80 RGB image is 19,200. Such high dimensions of feature space invite the curse of dimensionality-a problem that usually increases computation costs and makes many models prone to overfitting. PCA reduces the feature space to only the most important components that preserve the essential variance while discarding noise.

In the images, pixel values can be highly correlated. Hence, PCA transforms these correlated features into an orthogonal basis, making the input features uncorrelated to the classifier. This will help improve the performance of a machine learning model such as SVC and Random Forest used here.

Applying PCA retains only components explaining the majority of the variance, filtering out minor variations that are likely due to noise; hence, improving the robustness of a model.

This in turn allows for quicker training and less memory requirement, hence enabling the effective scaling of the models with probably higher or sustained accuracy.

The average RMSE of the reconstruction as a function of number of components preserved follows a sort of decaying exponential function wherein the highest value of RMSE reconstruction is around 0.56 achieved with around 10 PCs and it reduces to around 0.2 as the components exceed 120.

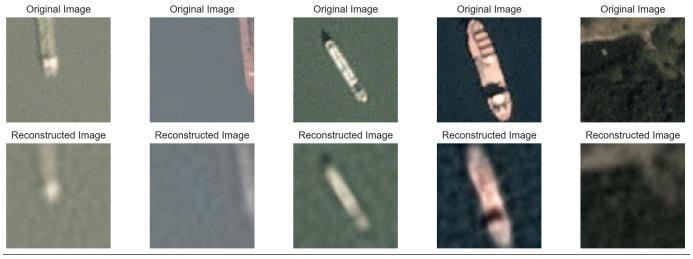

Fig. 1. Reconstruction of Images from 90% Variance

PCA still manages to preserve variance that represents ship and non-ship features in this lower-dimensional space; hence, it maintains the classifier's ability to effectively discern the two classes. This is reinforced by calculating the number of components required to explain 90% variance, which is a good indicator of reducing the dimensions of the data and only keeping the principal components necessary to retain 90% of the data but still successfully reconstruct the data, barring a few details which would be lost because of not using all the components. This is considered a good practice because it ensures an optimal balance between dimensionality reduction and information preservation.

In other words, retaining 90% of the cumulative variance keeps most of the variability of the original data and would allow several of the important pieces of information to remain. High-dimensional data can also result in overfitting, especially with machine learning models such as SVM and Random Forest. In reducing the dimensions, the 90% variance retained limits the degree of overfitting since it reduces the complexity of the feature space without losing critical information.

On calculating, it was found out that 107 components were required to retain 90% variance. This is helpful as the subsequent models would use PCA as a dimensionality reduction pre-processor in their pipelines, with only the first 107 components to select (as they provide the most info required to reconstruct the data reliably).

The pipelines thus leverage PCA for a balance between computational efficiency and classification accuracy since the latter forms a very crucial step in the preprocessing of high-dimensional satellite imagery.

D. Classifiers with PCA as a Pre-Processor

For interpreting information from the dataset and reliably extracting information to be trained on the model, we used PCA as part of the training pipeline which served as a pre processing step. This reduces the computation necessary and still preserves the performance of the models or even outperforms the original models.

In our case, SVM paired with PCA turned out to be the overall best performing model with a training accuracy of 0.9959 and an F1 Score of 0.9918. Even with such high numbers on the training set, the testing accuracy turned out to be 0.9725 with an F1 Score of 0.9442. This goes on to show that the model performs really well on the dataset and on unseen data, alike. We also get to see how PCA improves the generalising ability of SVM for the unseen data. But this does come at a cost, as it took the model very long to train and converge, taking approximately 86 minutes to train on the parameter grid with an exhaustive grid search. This time was reduced to 14 minutes when we fixed the number of components for PCA to 107 (required to explain 90% variance in data), and let the grid search for rest of the parameters and report on the best fit model which turned out be: ('pca_n_components': 107, 'svc_C': 10, 'svc_kernel': 'rbf')

For RF, the Training Time was about 5.7 minutes, searching for the best fit model with paramters 'pca_n_components': 110, 'rf_max_depth': 10, 'rf_n_estimators': 50, giving a training accuracy of 0.9981 and F1 Score of 0.9962. This showed strong training performance but looked prone to overfitting. Although the testing accuracy was 0.9550 with F1 Score of 0.9011. However, it was outperformed by SVM with PCA in test accuracy and F1 score, suggesting PCA is beneficial but perhaps more for SVM than Random Forest.

E. Manifold Learning Algorithm

For the final comparison, we used a manifold learning algorithm, namely Isomap, after having tested out various other techniques like MDS, t-SNE and LLE, all of which perofrmed poorly in comparison to Isomap for the dataset. The classifiers paired with Isomap work well in the sense for this dataset offers unique advantages, particularly for high-dimensional image data like satellite imagery, where underlying relationships may be nonlinear.

Unlike PCA, which works under a linear relationship present in the data, the algorithm of Isomap does much better in a nonlinear manifold. This could be beneficial for satellite images, for example, where all the relationships between pixels do not lie on a simple linear plane.

Isomap calculates the geodesic distances-other than straight-line Euclidean distances-which are the shortest paths on the manifold. In that way, one ensures that the reduced representation truly captures the global structure of the data-something quite essential in images with complicated spatial relationships.

Normally, satellite images have complex patterns at high dimensions, for example, orientation and size of ships among other factors involving the atmosphere. Dimensionality reduction using Isomap may retain proper patterns which might get lost when using PCA.

a) SVM: The components of Isomap were fixed to 3 for ease of convergence of the model and so that the grid search did not spend too much time going though different values for it. As such the Best Params were 'svc_C': 10, 'svc_kernel': 'rbf', reporting training Accuracy of 0.8994 and F1 Score of 0.7745, with the model taking 1.7 seconds to fit on the training data, which was compartively very fast compared to the other models trained uptil now.

The testing Accuracy was 0.8337 with an F1 Score of 0.6434. SVM had lower performance on the test set, reflecting the challenge of effectively capturing data structure with Isomap in this context.

b) RF: The Training Time to fit on the dataset was 252.56 seconds, while fixing the components to 3 for the manifold algorithm, getting the parameters of the best fit model as 'rf_max_depth': None, 'rf_n_estimators': 200 The training set accuracy and F1 score reported was 1.0000, which achieved perfect training performance but showed a significant performance drop in testing accuracy and F1 score (0.8825 and 0.7403, respectively), indicating some overfitting or difficulty in generalizing with Isomap as the preprocessor.

On comparing with the previous models, manifold learning seems to be underperforming, most likely due to limited capacity to capture high-dimensional relationships in the dataset. On visualising the first 2 components of the data, we can say that the model is trying to select the background ie the ocean in this case and the shape of the ships, which are sort of horizontal lines all throughout the dataset.

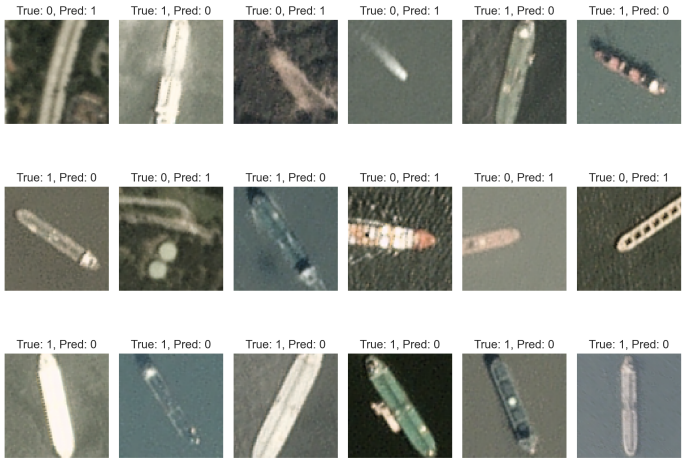

Fig. 2. Misclassification Samples for Best Performing Model

IV. Best Overall Pipeline

The results therefore imply that the application of the SVM with PCA pipeline is an excellent choice because it strikes a balance with high accuracy, robust F1 scores, and better generalization. The PCA pre-processing step helps the SVM to capture the important features without overfitting, as evident from the strong test performance metrics.

Even with great performance metrics on the test set, there still are a few misclassifications which can be visualised for the model, to better help us understand the working of it and why it may be misclassifying such samples. On taking a closer look, we get to see that 22 samples have been misclassified in total, which is a small number considering the size of the dataset. Here, out of the 18 samples shown, we can make out that some of the ships in the images were blue in color which makes them camouflage with the ocean color, confusing the model and hence making a wrong judgement on the samples. Similarly, the model confuses between objects which resemble the shape of a ship but actually are not, again making the model classify incorrectly.

Based on this, we can conclude that due to the inherent nature of the dataset, the model makes a few bad calls on the classification, attributing to the shape and color of the objects in the samples present. Talking about some potential steps for improvement:

- Analyzing which features or principal components contribute to misclassifications could help refine the feature set. It may be useful to explore non-linear feature interactions or domain-specific features that PCA may have overlooked.
- Adjust the class weights or apply more targeted sampling techniques, to balance the dataset and reduce the bias of more dominant features.

V. Comparisons

From the various results obtained, we can deduce that SVM with Isomap as a dimensionality reduction step had the lowest prediction time of just 0.03s, followed by that of the Random Forest Classifier with 0.17s. A general trend could be noticed where SVM took more time to train, but did outperform RF in the overall context with the test dataset metrics.

VI. Application

The sliding window method was applied to larger scene images as an application of the best trained model based on the reporting metrics from the train and test data.

The Window Size chosen was 80×80, matching the training data resolution., with a Stride of 10x10 for Overlapping windows for exhaustive coverage. The same Preprocessing steps of Scaling and PCA transformation were integrated within the pipeline so that the same feature vector of the image would be fed to the best trained model.

Here, as discussed above, the model makes incorrect predictions on grey lines which resemble the shape of a ship and draws a bounding box around it when travesring through the image. This can be noticed for quite a few areas where misclassifications have occured.

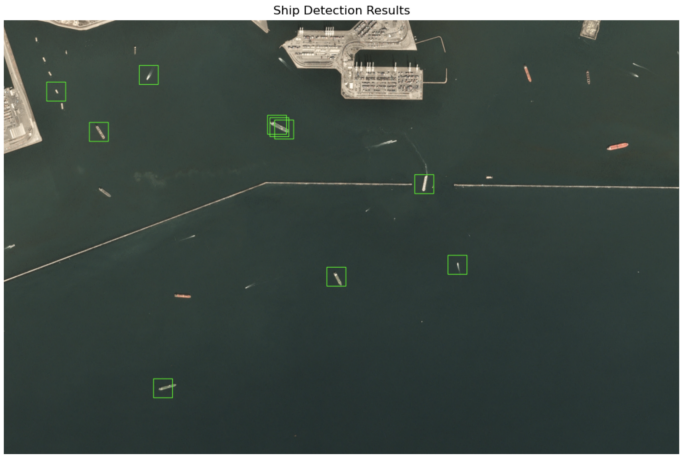

Fig. 3. SLiding Window on Sample Image

VII. Results

Based on the confusion matrix for the best performing model, we get to know that the False Negatives had Small or occluded ships missed by the model. False Positives had High-contrast or ship-like structures misclassified as ships. While the misclassified samples reveal that Occluded ships lead to false negatives and Non-ship regions (e.g., docks) contribute to false positives.

Isomap Pipelines underperformed due to limited capacity to capture high-dimensional relationships in the dataset whereas SVM with PCA outperformed other methods in accuracy, generalization, and F1 scores.

VIII. Conclusion

This study demonstrated that SVM with PCA is the most effective pipeline for ship detection in satellite imagery, offering a robust solution for both test data and large scene images. Despite misclassifications, targeted improvements such as augmentation, hybrid features, and post-processing can significantly enhance performance.

A. Recommendations

Based on the results from the various models trained, there can be a few key takeaways:

- Ensemble Models: Combine SVM and RF to leverage their complementary strengths.
- Feature Integration: Combine PCA-transformed features with other helpful features like texture descriptors for better performance.
- Data Augmentation: Include rotations, occlusions, and scaling of ship images for better generalisation and robustness of the model(s).
- Remove overlapping bounding boxes to reduce false positives.
- Contextual Filtering: Introduce constraints based on location or surroundings of ships.