Ship Detection with Classification Models

Adam Cheng   
*EEL4930: Applied ML Systems*  
*University of Florida*Gainesville, United States  
adam.cheng@ufl.edu

*Abstract* – Satellite imagery holds insight into many industries regarding naval transport, through the quantity and location of ship vessels. Locating these ships manually is a tedious and fruitless task; to speed up this process requires the use of machine learning classification models. The size of satellite image data also requires the use of dimensionality reduction to both capture generalizable features of ships and to increase efficiency. Different classification models will be trained with different reduction techniques with hyperparameter tuning, keeping performance and time in mind. A comprehensive analysis of classification models and dimensionality reduction techniques demonstrates that there are effective models for classifying ships within scenes in an accurate and timely manner.

Keywords—ships, classification, machine learning, dimensionality reduction, hyperparameter

# Introduction

The large amount of satellite data that is received daily holds great insight into numerous industries, ranging from defense to agriculture. This paper will address the detection of ships from satellite imagery. Locating ships from satellite resolution can be difficult, but can lead to easier means to monitor port activity, analyze supply chains, or provide national security. In order to make this task more applicable to real world scenarios, the use of machine learning classification and unsupervised dimensionality reduction algorithms is required. Models will make decisions based off the given image data, perform some from of dimensionality reduction in order to capture the most important features of each image, and then make a classification prediction. Each model will carefully tune hyperparameters to limit overfitting while allowing models to learn generalizable characteristics of the data, with performance and training/prediction time being used to evaluate the best model for classifying ships.

# Classifiers without Reduction

This section will address model creation without the use of dimensionality reduction. Additionally, this will include all of the decisions regarding preprocessing logic as well as initial model selection; since the other models are utilizing the same preprocessing and model selection, this discussion will be largely skipped for future problems, unless information needs to be analyzed for each new dimensionality reduction technique.

The given data is in the form of 4000, 80x80 RGB images. Since each pixel within the image actually has three values assigned to it, my data preprocessing turns each image into a 19200-length 1-D array, as this will maintain all information from the original images in contrast to performing something such as grayscale, which may lose some important RGB information. When considering splitting into training and test sets, it was decided to perform an 80/20 stratified split. The 80/20 split should allow for ample training data while still maintaining enough test data to be representative of the entire dataset. The stratification was performed on the target values; given that there was more representation of ‘no\_ship’ classifications than ‘ship’ classifications, this just ensures that the distribution is consistent across both sets. For all future models, it was also decided to utilize MinMaxScaling, as this preserves the initial distribution of the pixel data while ensuring models are not affected by large feature scales.

In order to obtain a comprehensive idea of which models would have the best performance on ship classification, it was decided to utilize two different models; a soft-margin SVM with a radial basis function (RBF) kernel and a random forest classifier. The SVM was chosen due to its ability to find non-linear classifier surfaces with the RBF kernel, which is likely to exist in image classification. The size of the feature space and sample count should also mean the SVM is relatively efficient enough to complete training and prediction in a reasonable amount of time. Proper hyperparameter tuning combined with the soft-margin should ensure a generalizable decision surface. The random forest was chosen for similar reasons; the decision trees it uses should also be able to find the generalizable, non-linear, decision surface. It should also be robust to overfitting given proper hyperparameter tuning, and is likely to be efficient due to the ability to run multiple decisions trees in parallel.

Hyperparameter tuning used a GridSearch cross validation scheme with 5-folds. GridSearch was chosen as it will find the optimal choices from the given selections; these selections were decided based on information regarding the dataset size and characteristics of each model, which should ensure sufficient hyperparameters. Five folds were used as this would give enough data to both train and validate on, ensuring biased sets do not influence performance too much. Accuracy was chosen as the scoring metric as it should be a good indicator of the model’s ability to correctly label images, which is one of our desired outcomes. The regularization parameter, C, was the only tuned hyperparameter for the SVM, as this should be the most influential in determining the sensitivity of the decision boundary. The tree count and max depth were the hyperparameters tuned for the random forest, as these should give the optimal amount of trees needed for generalizable learning and best limit overfitting through maximum depth. Given the size of the feature space was large, hyperparameters were also made relatively larger to account for a greater number of these features to be useful.

After hyperparameter tuning, the SVM parameter C was found to be 10, and the tree count and max depth were found to be 200 and 20, respectively. These models were then evaluated; results are in Table 1 and II and will be discussed in section V and VI.

# Principal Component Analysis

For this section, principal component analysis (PCA) was utilized to perform dimensionality reduction on the images. This will include an explanation as to how PCA was conducted, a discussion on the influence of features based on root mean-squared error (RMSE) and variances, and image reconstructions.

In accordance with section II, MinMaxScaling was used before PCA. By using the PCA, it was found that a total of 102 principal components were needed to explain 90% of the variance. Given that the original number of components was equivalent to the size of the training dataset (3200), the reduction in feature space was successful, and demonstrates that PCA was able to learn a good amount from just a few components, while still maintaining the general characteristics of the initial dataset. Through PCA reduction, model selection and training should be much faster given the shrinkage by such a large factor; for ship classification, this means being able to determine much more quickly the amount of ships or the location of ships, allowing for faster decision making from satellite images.

A collage of images of a comet

Description automatically generated

Fig. 1. Original images versus reconstructions using 102 principal components (top = original, bottom = reconstructions, right = no\_ship, left = ship).

Reconstructing the images using the 90% variance threshold appears to yield results that capture the majority of the qualities of the original image. In Figure 1, it is possible to see that the reconstructions maintain essentially the correct locations of items within the image, at only the cost of some fuzziness and loss of very minute details; color, shape, and positioning remains relatively the same.

A graph of a person

Description automatically generated with medium confidence

Fig. 2. Plotting the Average RMSE of each image as a function of the Principal Component count

Figure 2 demonstrates an exponential-decay type graph of the RMSE compared to the amount of principal components. This implies that each new principal component adds a decreasing marginal benefit; the most prominent principal components hold the most information regarding the original images with each new component providing smaller amounts of information. Combined with the fact that more principal components also means a larger amount of computation time, it appears that the selected value of 102 on Figure 2 provides a relatively low RMSE while still not exceeding an unnecessary amount of principal components.

# Classifiers with PCA

Having completed PCA for our dataset, this section will focus on training the same models from section II with PCA. These PCA models will utilize the exact same preprocessing strategy (training/test split, pixel adjustment, MinMaxScaling, etc.) as the models in section II. The models will also be the same; soft-margin SVM with RBF kernel and random forest. This section will however include the hyperparameter tuning specific to the PCA models, and a discussion into how PCA may be useful for the model to capture more generalizable information from the images.

In terms of image classification, PCA will in general improve efficiency, training, and prediction time. This is due to the lower amount of components required, allowing for streamlined computations from the two models. In addition, PCA may actually help the model to fixate less on specifics of each pixel, and instead capture more generalizable information due to the limiting of the amount of unique components from the dataset that it is capturing. Additionally, given that PCA is linear, there may still be some interpretability within components if desired.

Hyperparameter tuning for PCA again utilized GridSearch with 5-folds. Having to now tune hyperparameters for PCA, GridSearch should again be able to find optimal choices from our given selection. The five folds were again chosen to ensure that training and validation sets were indicative of the entire dataset, and to prevent possible bias from validation sets that may be randomly distinct from the dataset. Accuracy was again used as the scoring metric as it should capture the model’s ability to correctly label ship images. As with section II, the value of C for the SVM and the values of tree count and maximum depth for the random forest were tuned. However, to account for the fact that there were now less components through dimensionality reduction, the grid search was made to search through smaller values of maximum depth, to ensure that the random forest would not overfit. Furthermore, the hyperparameter for component count in the PCA was also tested. These values were around the range of 100, due to the finding in section III that 90% of the variance was captured in 102 features, implying that any more or less may either make the model overfit or underfit, respectively.

Hyperparameter tuning found values of 10 for the value of C and 120 for the component count of the PCA-SVM model. It also found values of 20 for max depth, 200 for tree count, and 100 for the component count of the PCA-Random Forest model. The hyperparameters appear to be similar to the models without PCA, and the values for component count are as expected, similar to the range of 100 from the 90% variance of the PCA from section III. Model results are in Table I and II, and will be discussed in sections V and VI.

# Classifiers with Manifold Dimensionality Reduction

In addition to PCA, the use of manifold dimensionality reduction techniques was another point of focus for model training. This section will focus on the choice of Locally Linear Embedding (LLE) as the manifold learning technique for ship classification and the updated hyperparameter tuning utilizing LLE in combination with the two model types. This section will also visualize the dimensions of the unfolded LLE manifold. Performance will be compared between all six trained models, comparing performance metrics, overfitting, and training/prediction time. As with section II and IV, preprocessing and base models will remain the same.

LLE was the manifold learning algorithm of choice for our given data. One main reason LLE was chosen over other manifold learning techniques was due to the scalability and speed of LLE compared to other manifold learning techniques; due to only requiring neighborhood graphs rather than computing distances across the entire manifold, it should scale better for the size and count of images within the dataset. LLE should also be able to capture local trends that will help in the classification of ships; this could include things such as rotations or coloring of ships. This will allow an LLE to best capture the underlying manifold structure, and should help preserve these local relationships.

Hyperparameter tuning was a 5-fold GridSearch using accuracy as a scoring metric. As with section II and IV, the five folds should ensure bias within validation sets is limited. Accuracy should allow the model to best classify ships as well. GridSearch should also allow for the optimal hyperparameters to be found given that the selections have been analyzed to best match the size of the sample. In addition to the tuning of C for the SVM and tree count and maximum depth for the random forest, this hyperparameter tuning will tune the component count and neighbor count for the LLE. Tuning the neighbor count implies a K-Nearest Neighbors approach; this was chosen as it should be sufficient to capture local relationships between related images. Tuning neighbor count should also ensure the data maintains its local relationships while excluding images that are not related, with isolated islands most likely being minimal. The component count should also find the amount of dimensions that best captures the inherent manifold structure. Given the expected lower amount of components, maximum depth for the random forest was also placed in a lowered range, in a similar approach to section IV.

With this tuning, the component count and neighbor count for the LLE in the SVM model were chosen to be 15 and 20, respectively, and the value of C for the SVM was 10. The component count and neighbor count were the same for the Random Forest, with values of maximum depth and tree count being 20 and 200, respectively.

A close-up of a graph

Description automatically generated

Fig. 3. Visualization of the first two dimensions of LLE dimensionality reduction

The first two dimensions displayed in Figure 3 show some fairly clear trends. It appears that on the horizontal axis, images with darker overall pixels tend to be on the positive side and images with lighter overall pixels appear on the negative side. This also appears to work for the image average, rather than any specific spot on the image; images with boats are slightly more towards the center than pure ocean images, probably due to ship pixels giving a brighter average. A solid trend for the vertical axis is slightly harder to find; the two separate prongs could demonstrate that darker/lighter trend among the right/left prongs. It may also have to do with noisy data, as it seems that images with more noisy images are towards the bottom of the graph.

1. Performance

| Models | Training | | Testing | |
| --- | --- | --- | --- | --- |
| F1-Score | Accuracy | F1-Score | Accuracy |
| SVM  w/o Reduction | 0.99937 | 0.99969 | 0.93299 | 0.9675 |
| Random Forest  w/o Reduction | 0.99938 | 0.99969 | 0.89710 | 0.95125 |
| SVM  w/ PCA | 0.99308 | 0.99656 | 0.93333 | 0.9675 |
| Random Forest  w/ PCA | 1.0 | 1.0 | 0.89617 | 0.9525 |
| SVM  w/ LLE | 0.91742 | 0.96 | 0.86514 | 0.93375 |
| Random Forest  w/ LLE | 1.0 | 1.0 | 0.85347 | 0.92875 |

1. Time

| Models | Time (seconds) | |
| --- | --- | --- |
| *Training* | Prediction |
| SVM w/o Reduction | 11 | 7.1 |
| Random Forest  w/o Reduction | 41 | 0.076 |
| SVM w/ PCA | 5.7 | 0.14 |
| Random Forest w/ PCA | 9.7 | 0.11 |
| SVM w/ LLE | 14 | 4.4 |
| Random Forest w/ LLE | 16 | 4.4 |

Among all the performance metrics in Table I, it appears that basically all the models performed strongly, with abnormally high performance in training, and good performance in testing. The high training performance (accuracy and F1-score around 1) does indicate that the models tend to be overfitting, which may necessitate further hyperparameter tuning. It appears that the best testing scores come from the SVM w/ PCA, but the model that appears to have the best performance while overfitting the least is the SVM w/ LLE. Looking at the time in Table II. The quickest model is the SVM w/ PCA; in general however, random forests perform better than SVMs, and PCA is the fastest, followed by LLE and then no reduction.

# Model Performance and Solutions

Utilizing the metrics from section V, this section will cover the confusion matrices, misclassified samples, and the addressing of the models’ issues. These will be synthesized into a discussion on the best model to be used for ship classification to both be accurate but also realistic in the efficiency to be used in real-world applications.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Fig. 4. Visualized confusion matrices (top to bottom = no reduction, PCA, LLE, left = SVM, right = Random Forest)

A close up of a knife

Description automatically generated

A collage of different types of ships

Description automatically generated

Fig. 5. Example misclassified samples of false positives and false negatives from the SVM model with no reduction

A collage of images of water and buildings

Description automatically generated

Fig. 6. Example ship classifications from the SVM model with no reduction on full scenes with ships

From the confusion matrices in Figure 4, it does appear that the majority of misclassifications occur from false negatives for all models. This implies that the models may not be fully capturing the generalizable qualities of images of ships. From Figure 5 and 6, one can see that these false negatives appear to be for ships that are either running the entire length of the image or have some sort of lighting or color that appears to obfuscate against the background. One reason for this may be the human made boundary for whether part of a ship is counted as a ship or not; tweaking this may be able to aide in these classifications. The false positives arise from either half images from ships, noisy images, or images of objects that are long and thin, like a ship. Ultimately, addressing misclassifications may require more hyperparameter tuning; with more time, fine tuning these may be possible, and would also help handle overfitting, due to a better capability to capture generalizable features while limiting specifics. Manually engineering features such as a ship’s tapered edge may be useful in helping to solve these misclassifications; this however, would probably require more complex models.

When considering each model’s performance and time, the overall best pipeline is likely the SVM with PCA. While the overfitting is still high, it is relatively lower in magnitude compared to the other models that are highly overfitting; this may also be able to be solved with more hyperparameter tuning, including things such as testing other kernels and regularization parameters. Additionally, having the highest performance in testing indicates a strong ability to learn the features of the ship classifications. Finally, the training and prediction time are relatively very quick, which would be able to keep up with the sheer amount of image data from satellite imagery. PCA will also be slightly more easy to interpret, as it may be possible visualize each component as an image. However, if additional hyperparameter tuning was not an option, it may be better to opt for the SVM with LLE. While the performance is worse than the SVM with PCA, the scores are less indicative of overfitting, with F1-score and accuracy score in training in the low and mid 0.90s, respectively. The prediction time is also relatively good compared to other models, which should still allow for people to find ships across scenes relatively quickly. Ultimately, when attempting to locate ships both effectively and efficiently in business contexts, proceeding with the SVM with PCA or SVM with LLE is a good option.