Iceberg-Ship Classifier Using SAR Image Maps

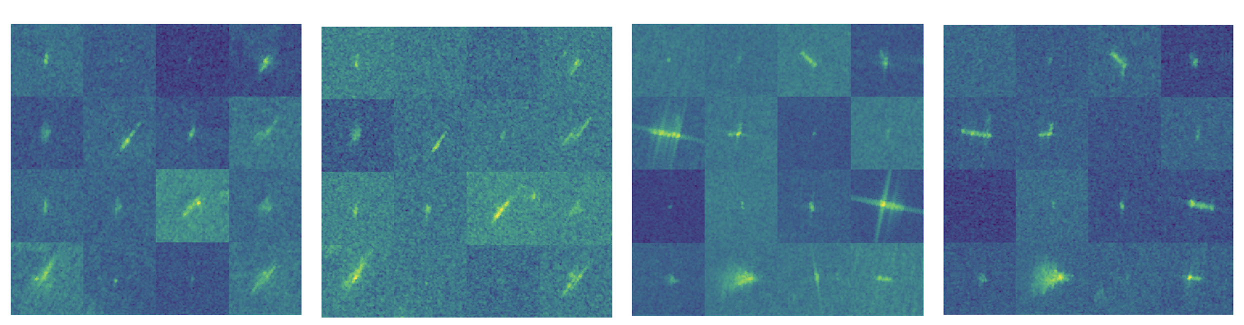
Li Jun, Parulekar Atharva, Samant Dhruv

Abstract

Drifting icebergs present threats to navigation and activities in areas such as offshore of the Coast of Canada. Accurate detection of threatening icebergs is crucial for maintaining safe working conditions. We use SVMs and Convolution Neural Networks to build and test algorithms that automatically identify, using Satellite Aperture Radar images, whether a remotely sensed target is a ship or an iceberg. We tested a deep convoluted neural network and obtained an accuracy of 88% on the val set. Augmenting the images and increasing the data set to 24 fold, we trained the model parameters using the Residual Network framework [2], achieving a test set accuracy of 99%. Further reducing the speckle noise using Lee filter gives us of 97% accuracy on the val set.

Data

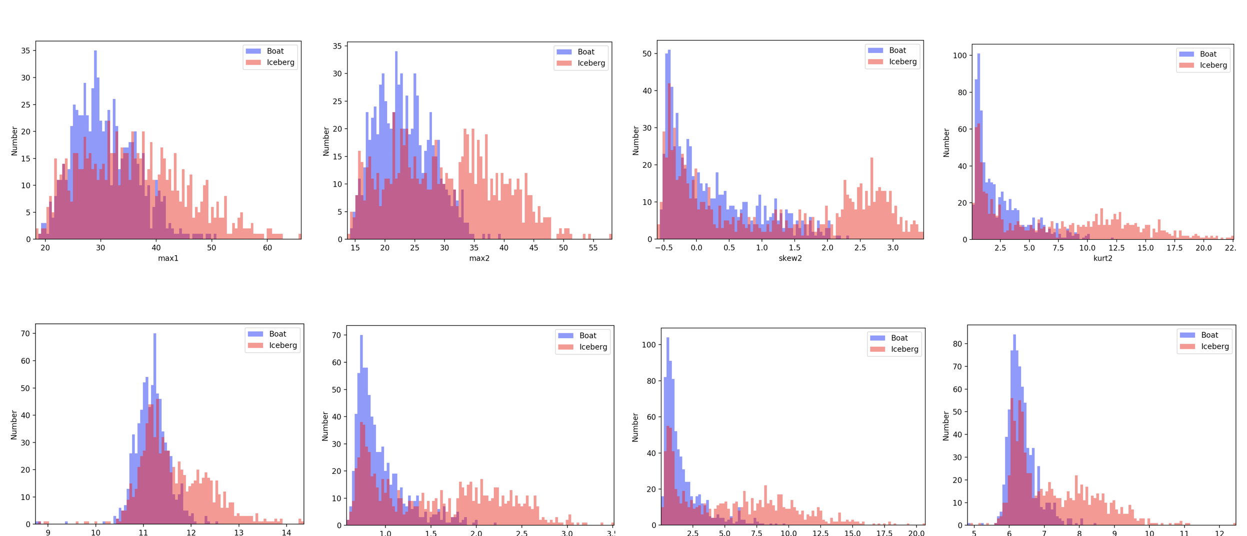
Raw data: Our data comes in a JSON file as 1604 sets of satellite images × 5 fields each: ID, image from HH channel(transmit/receive horizontally), image from HV channel(transmit horizontally and receive vertically), incident angle, and an is iceberg value as a 0 or 1 integer, representing a boat and an iceberg respectively.



Iceberg HH Iceberg HV Ships HH Ships HV

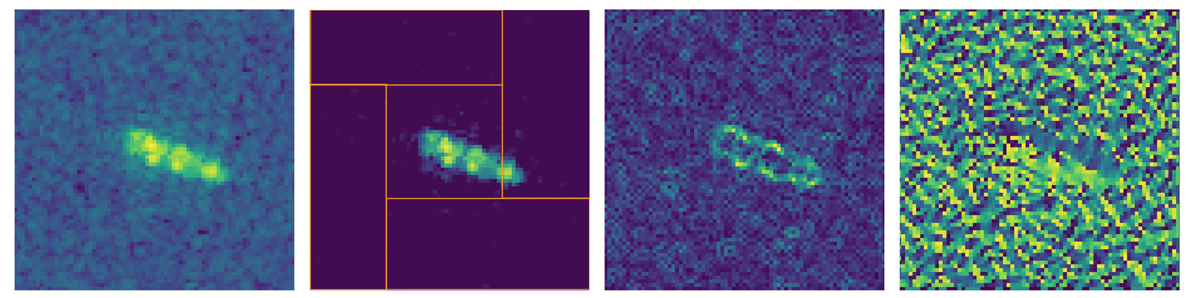
Data augmentation: We augmented the data 24 fold, resulting in 38496 training images. Specifically, we rotated each image by multiples of 30 degrees, and then we flipped each image horizontally.

Visualization: We plot histograms of max, standard deviation, skewedness, and kurtosis of pixel activations as well as mean standard deviation and kurtosis of histogram of gradients.



Models

Feature Extraction



Original Processed g - Magnitude θ - Angle

We used weighted histogram of scaled gradient angles with gradient magnitudes as weights to count occurrences of gradient orientation in center and edge areas.

$$g = \sqrt{g_x^2 + g_y^2} \quad g_x = xcomponent(g)$$
$$\theta = \arctan(g_y/g_x) \quad g_y = ycomponent(g)$$

Baseline

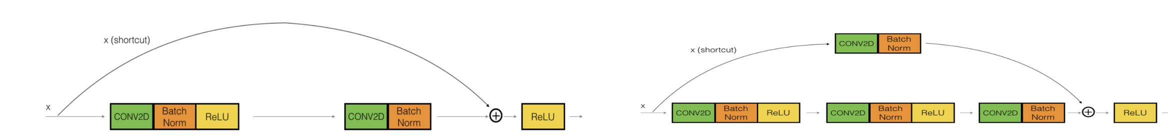
We perform the following procedure on the data:

- Perform standardization of image pixels.
- Use Lee filter to denoise image.
- Extract features like HOG and HOG of certain transforms and filters.
- Perform PCA on features.
- Use an SVM with regularization.
- Evaluate results using 10 fold cross-validation.

$$f = \arg\min_{f\in\mathcal{H}} \left\{ \frac{1}{\lambda n}\sum_{i=1}^{n}(1 - yf(x))_+ + \frac{1}{2}||f||^2_{\mathcal{H}} \right\},$$

ResNet and CNN

- For automatic feature extraction and parameter sharing for reduction of overfitting, we have experimented with some CNN models.
- We standardize pixels and create RGB channels.
- We implement Lee filter to denoise.
- We implement a 560,193 parameter CNN with dropouts and sigmoid output.
- We train ResNet 50 as our final model [2].



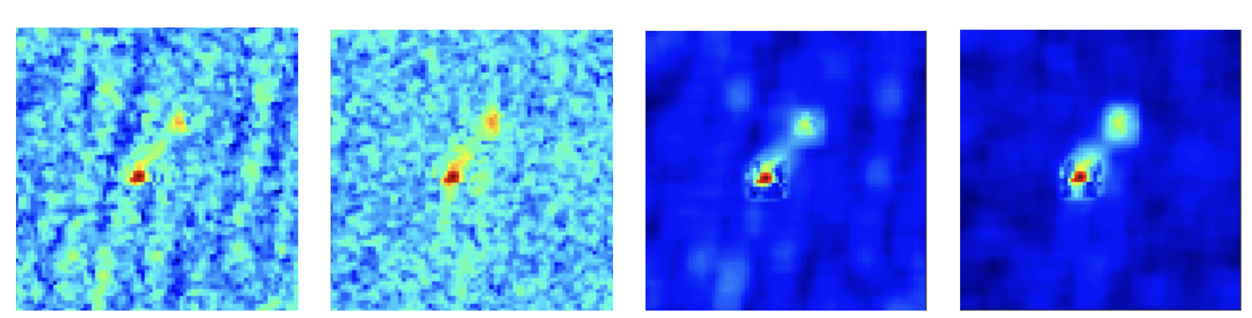
Identity Block Conv Block

Image Processing

Lee Filter: We implement a Lee filter[1] to reduce the speckle noise associated with SAR images. We apply a spatial filter to pixels, which replaces the center pixel value with the value calculated using local statistics of neighboring pixels in a square window. The despeckled value is given by:

$$P_C = L_M + K(\dot{P_C} - L_M)$$
$$K = L_V/(L_V + AV)$$

L_M = local mean, L_V = local var
AV = additive noise var, and P_C = central pixel



Original HH Original HV Lee HH Lee HV

Future Work

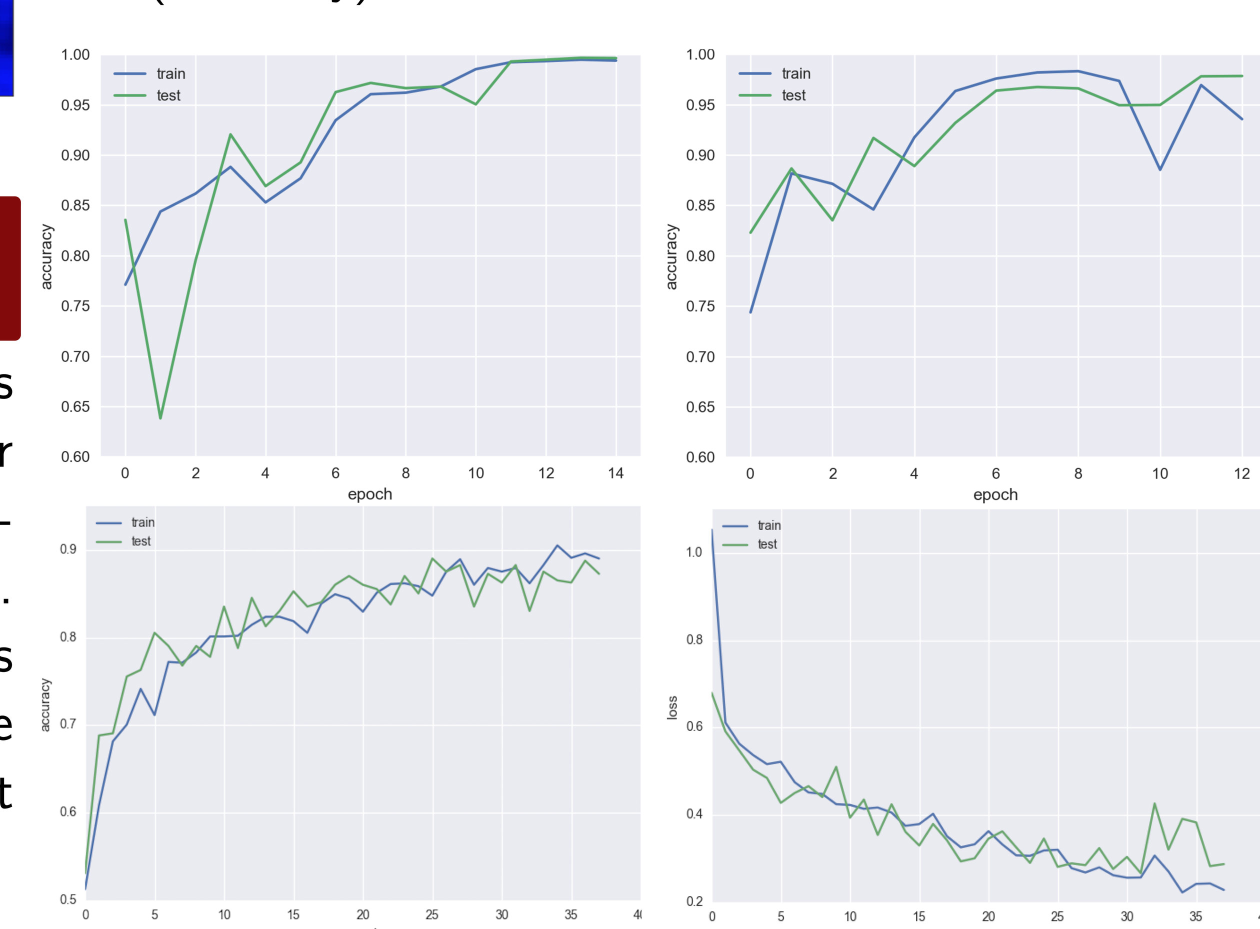
We would like to train our model for more than 50 epochs using the inception network, with skip connections. Other despeckling techniques like enhanced Lee filter, Frost Filter, and Kuan Filter could give better and faster results. Our final model will be an ensemble of deep neural nets which work in tandem and are trained separately. The model could be tested on SAR images from different sources to test whether the parameters are transferable.

Results

We tabulate the results of the models we have tried :

Model	Data	Noise	Train	Val
SVM	1604	Yes	98.35	67.19
SVM	1604	No	77.20	70.51
CNN	1604	Yes	89.45	88.33
CNN	1604	No	84.62	88.28
ResNet	34646	Yes	99.44	99.63
ResNet	34646	No	97.42	97.84

The larger dataset used is the augmented data set while the smaller dataset used is the original dataset. Our train test set split is 8:2 and 9:1 for CNN and ResNet respectively. For the SVM we use 10 fold cross validation as an evaluation metric.Below clockwise from top left Resnet50(noise),Resnet50(denoised), CNN(loss) and CNN(accuracy)

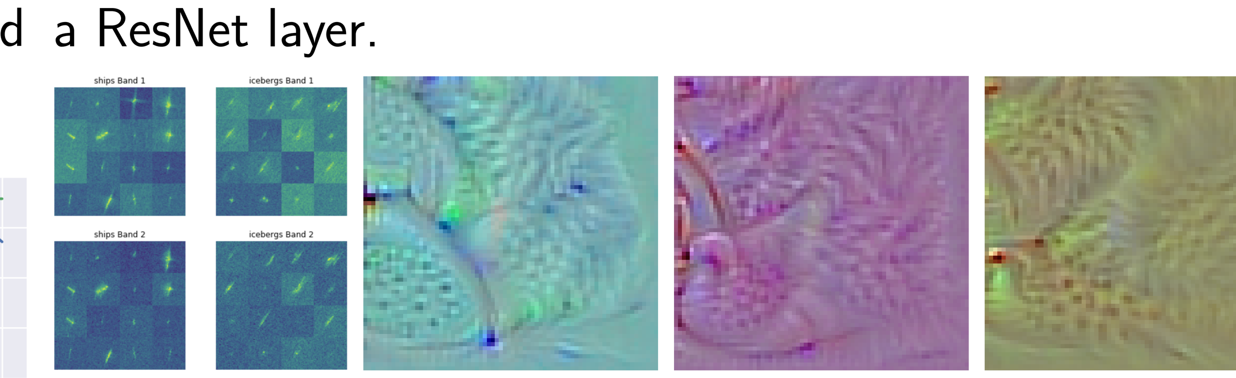


Discussion

We infer the following from this exercise:

- Denoising significantly improves the SVM performance from 67% to 70%.
- ResNet performs better than baseline and CNN.
- The performance of CNNs drop as we denoise the data. Some crucial image features are possibly lost.
- Models on denoised data generalize well and rarely overfit the data.
- The model trains faster after denoising (using enhanced Lee filter) and achieves a 95% accuracy in 6 epochs.

The figures below show the 3 band inputs that maximizes the activation of a filter in the first convolution of a ResNet layer.



Montage Band 1 Band 2 Band 3

References

[1] Yafeng Zhang Junling Zhu Jianguo Wen. "A new algorithm for SAR image despeckling using an enhanced Lee filter and median filter". In: *IEEE* (2013).

[2] Shaoqing Jian Kaiming Xiangyu. "Deep Residual Learning for Image Recognition". In: *IEEE* (2015).