Classification of military vehicles in SAR imagery

Christian Keilstrup Ingwersen, Harald Løvenskjold Mortensen and Kasper Pryds Rolsted
Supervised by Jannik Boll Nielsen
DTU Compute, Technical University of Denmark

Abstract

Through the last decades, intelligence has become one of the most potent tools - with both civilian and military applications. Traditionally, this intelligence has been acquired manually which required physical presence on site, but has now in large scale been replaced with remote sensing. As the incoming data increases with dizzying rates, there is a severe need to automate the process of handling and analyzing. When knowing is half the battle, knowing and confirming the position of both allied and hostile military matériel is essential. This intelligence is usually acquired using radar sensing, but the resulting images are not fit for large-scale classification by humans. As such, in this project existing methods for image processing and classification are compared and used to distinguish and classify post-Soviet military vehicles in SAR-images from the MSTAR dataset. Dependent on the methods used to construct the classifiers, the worst-performing methods reach 53%-accuracy, while the best reach 92%. Compared to other studies on the same dataset, this result is decent but not excellent.

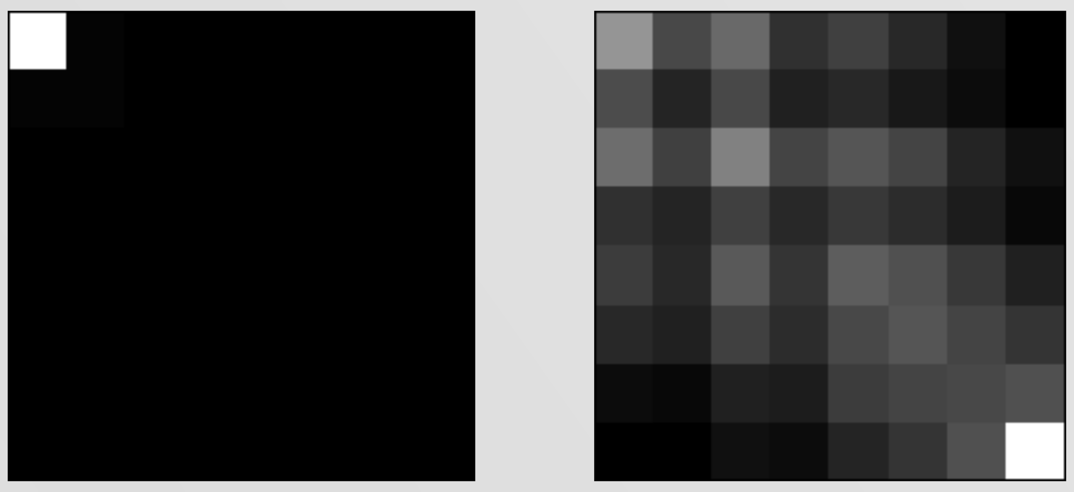
The ten types of vehicles included in the 10-way MSTAR set and examples of their SAR intensity images.

Extracting morphological and textural features from intensity images

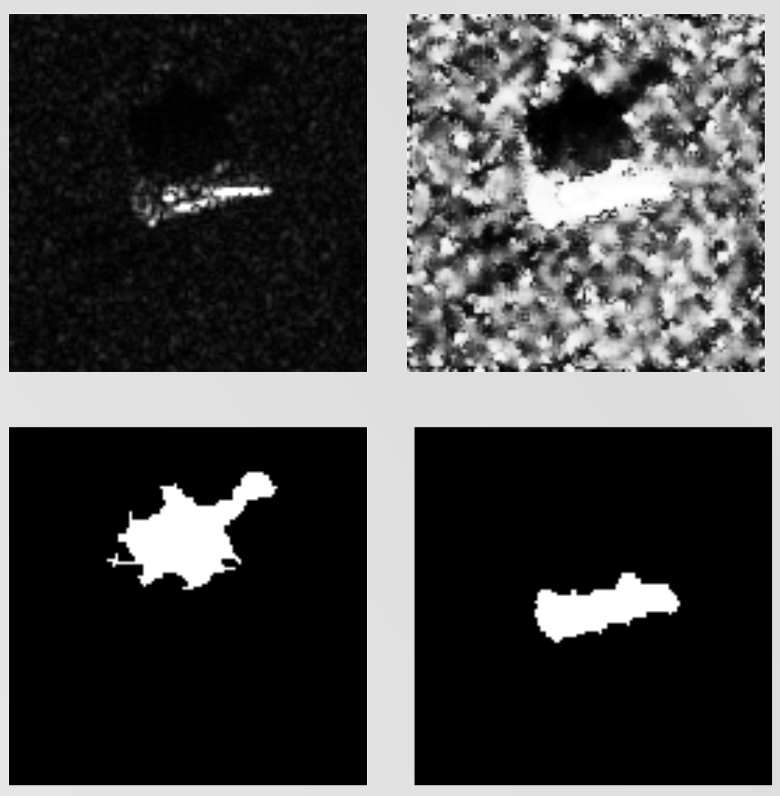
The images are first filtered with a 5-by-5 windowed Lee-filter to reduce speckle, then equalized with a 100-bin Histogram-equalization. This allows the image to easily be thresholded and the largest elements found, whereafter the shape of the vehicle and the shadow are extracted. For each of these shapes, 4 morphological features are computed, Major- and Minor-axis-length, object area and a measure of circularity. The textural features are computed from the raw image after a histogram-equalization, so as to have a proper Gray Level Co-occurrence Probabilities to compute from.

Histogram equalization

As visible from the GLCP of the raw image (left), the distribution of data is extremely skewed to the upper left corner. This makes it very difficult to compute any gray-level features. Thus, we transform the image by a histogram-equalization to yield a more balanced GLCP (right).



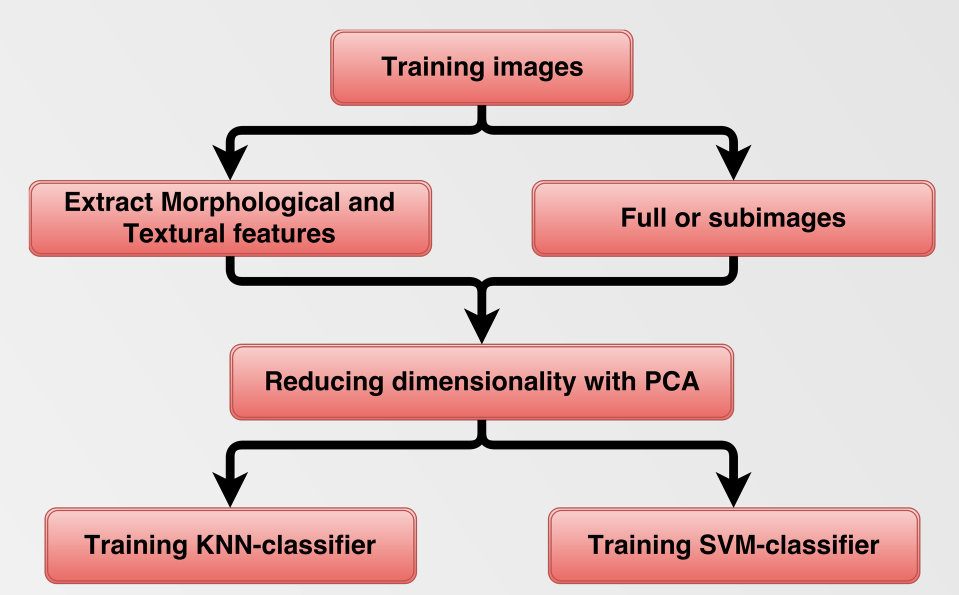
Prior to extracting textural features, the images are not filtered with the Lee-filter, as the speckle is not just a noise but can also be considered a feature of the sensed object. The computed features are: Contrast, Homogenity, Entropy and the variance to mean ratio.



These four images show the process of extracting morphological features, going clockwise from top left: Raw image, after Lee-filter and histogram equalization, the extracted shape of the vehicle, the extracted shape of the shadow.

Training classifiers

Several classifiers are constructed using different methods and input-spaces. Input consists either of the raw full intensity image, a subwindow or a feature-vector of textural and morphological features. To increase classification performance, the feature-space is variance-normalized using PCA. Lastly, a KNN- or SVM-classifier is trained.



Evaluation of classifiers

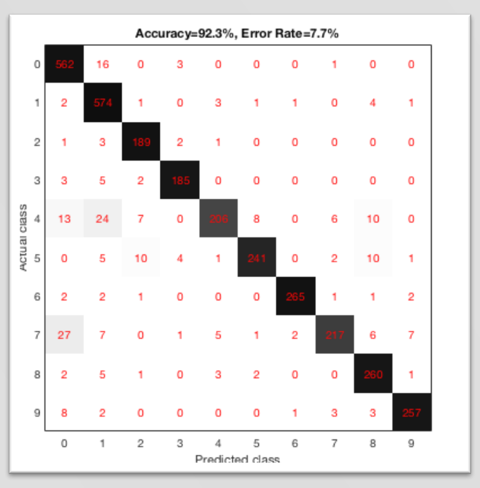
To estimate the performance of a classification-model, before testing it on a test set, one can compute the Generalization Error. This is an essential tool that allows us to evaluate models without involving the actual test-set and thereby eliminate any cases of subconscious or unintended training of the model on the test data. The generalization error is computed by K-fold cross-validation

$$\hat{E}^{gen}_{\mathcal{M}} = \sum_{k=1}^{K} \frac{N_k^{test}}{N} \hat{E}^{test}_{\mathcal{M},k}$$

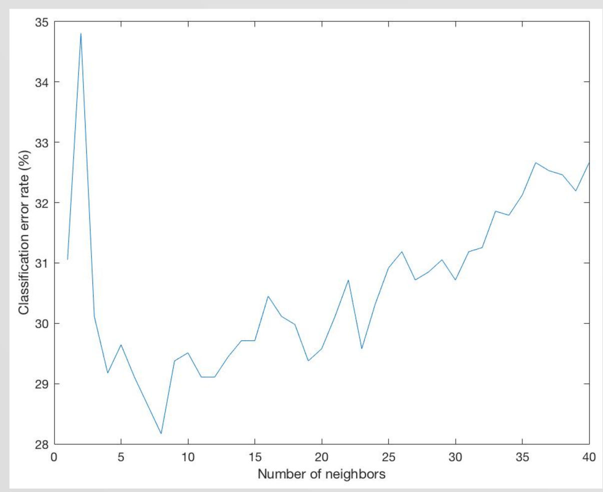
In each fold the test error is computed as a percentage of misclassified objects. When this is computed for all folds, the generalization error is computed with the described formula.

Results

Using PCA-based variance-normalization on the raw data set proved to have the best performance, as when reduced to a 50-dimensional problem, KNN peaked at 92.3% and SVM at 89.6%. These results are significantly higher than the results attained using the raw feature-vectors, respectively 43% and 45%. This probably indicates that both some decisive features are not encompassed by our vectors and that some clutter-features are present in the background of several images, and so unjustly contribute to classification.



Confusion Matrix for 100-dim. PCA, KNN on full images



Finding the optimal amount of neighbors for KNN

Future work

The natural next step from KNN- and SVM-classification is to apply deep learning to the MSTAR data set. We are looking to apply a convolutional neural network to the data set, where other studies have achieved up to 99% classification accuracy.

Acknowledgement

Thanks to Postdoc Jannik Boll Nielsen for his weekly meetings and supervision during the project.

Christian Keilstrup Ingwersen: s154264@student.dtu.dk
Harald Løvenskjold Mortensen: s154436@student.dtu.dk
Kasper Pryds Rolsted: s152987@student.dtu.dk