Utilizing imaging geometry meta-data in classification of synthetic aperture sonar images with deep learning

Narada Warakagoda and Øivind Midtgaard

Citation: Proc. Mtgs. Acoust. 47, 070011 (2022); doi: 10.1121/2.0001607
View online: https://doi.org/10.1121/2.0001607
View Table of Contents: https://asa.scitation.org/toc/pma/47/1
Published by the Acoustical Society of America

ARTICLES YOU MAY BE INTERESTED IN

Improving the realistic rendering of artificial sonar images using Cycle Generative Adversarial Networks
Proceedings of Meetings on Acoustics 47, 070010 (2022); https://doi.org/10.1121/2.0001598

Three inventions to clean ship hulls, decontaminate hospital floors, and treat wounds, using air, sound and water
Proceedings of Meetings on Acoustics 47, 032001 (2022); https://doi.org/10.1121/2.0001606

Optimization of underwater acoustic detection of marine mammals and ships using CNN
Proceedings of Meetings on Acoustics 47, 070012 (2022); https://doi.org/10.1121/2.0001608

Modelled sonar and target depth distributions for active sonar operations in realistic environments
Proceedings of Meetings on Acoustics 47, 070013 (2022); https://doi.org/10.1121/2.0001610

Accuracy of numerically predicted underwater sound of a ship-like structure
Proceedings of Meetings on Acoustics 47, 070001 (2022); https://doi.org/10.1121/2.0001565

Comparison of two different methods to include a beam pattern in parabolic equation models
Proceedings of Meetings on Acoustics 47, 070006 (2022); https://doi.org/10.1121/2.0001593
Utilizing imaging geometry meta-data in classification of synthetic aperture sonar images with deep learning

Narada Warakagoda  
Department of Underwater Robotics, Norwegian Defence Research Establishment: Forsvarets forskningsinstitutt, Kjeller, Akershus, 2017, NORWAY; ndw@ffi.no

Øivind Midtgaard  
Norwegian Defence Research Establishment: Forsvarets forskningsinstitutt, Kjeller, Viken, NORWAY; oivind.midtgaard@ffi.no

Classification of objects in synthetic aperture sonar (SAS) images is a vital task in underwater automatic target recognition (ATR) and deep learning has proven highly successful in this task. Typical deep learning systems used for processing of SAS images are inspired by results in the domain of optical images. However, unlike the common optical images, SAS images can be supplemented with additional meta-information such as the imaging geometry, spatial resolution and signal-to-noise ratio. This paper explores techniques to exploit imaging geometry as an additional source of information for improving the classification performance of SAS images with deep neural networks.

One intuitive way of utilizing the imaging geometry parameters, mainly the ground range and the sensor altitude, is to use them as additional inputs to the system. We have conducted experiments to study this and the paper presents the results of these experiments.

An alternative approach is to consider imaging geometry as a constraint on the space of the input images and hence on the search space of the training problem. We consider different ways to impose this constraint and report the results of the experiments carried out to investigate the merits of the approach.
1. INTRODUCTION

Classification of objects in synthetic aperture sonar (SAS) images is a vital task in underwater automatic target recognition (ATR) and deep learning has proven highly successful in this task.\textsuperscript{5,7,8} Typical deep learning systems used for processing of SAS images are inspired by results in the domain of optical images such as Imagenet.\textsuperscript{1} However, unlike the common optical images, SAS images can be supplemented with additional meta-information such as the imaging geometry, spatial resolution and signal-to-noise ratio. This sort of additional information can provide important clues about the scene which could be useful in SAS image processing tasks such as segmentation, classification and object detection. This paper explores techniques to exploit imaging geometry as an additional source of information for improving the classification performance of SAS images with deep neural networks. We consider this issue in the context of classifying bottom objects into the classes of cylinder, truncated cone, wedge and clutter.

Several parameters including ground/slant range, sensor altitude, seafloor slope and object pose are required to fully define the SAS imaging geometry. Out of these, only the range (ground/slant) and sensor altitude are readily available, and other parameters are needed to be estimated. One intuitive way of utilizing the imaging geometry parameters is to use them as additional inputs to the system. This will increase the dimensionality of the inputs and can help reduce the disentanglement of different classes in the input space. However, at the same time, increased dimensionality leads to increased complexity of the classification problem, which can counter the advantage of easier disentanglement. We have conducted experiments to study these effects and the paper presents the results of these experiments.

An alternative approach to exploitation of imaging geometry parameters is to consider imaging geometry as a constraint on the space of the input images or subsequent feature maps. This will in effect restrict the the search space of the training problem. Training with such a constraint can be seen as a manifold regularization problem and the constraint can be included as a term of the loss function. We consider different ways to impose this constraint and report the results of the experiments carried out to investigate the merits of the approach.

2. BACKGROUND

A. DEEP LEARNING

Deep Learning has been immensely successful in many tasks related to artificial intelligence including computer vision. There are a number of well known Deep Convolutional Neural Network (DCNN) architectures that are pre-trained with optical images and usually available in the public domain. Most of these DCNNs have become famous because they have won the ILSVRC image recognition in the past few years. Alexnet\textsuperscript{6} is one such architecture which in fact triggered the current interest in deep convolutional networks after its win in the ILSVRC completion in 2012.\textsuperscript{4} Following this, several high performing DCNNs that have more complicated architectures have been designed by various research groups. The architecture known as Inception-Resnet-v2 is one of them and all the experiments in this work were conducted using this architecture as the backbone.\textsuperscript{6} The reason for this choice is that it is among the best in optical image classification while keeping the number of parameters and computational cost at a reasonable level. It has a complicated architecture which consists of so called inception blocks with different configurations. Inception block itself contains several parallel processing paths consisting of convolution operations as well as direct connections known as residual connections. Interestingly it has only a single fully connected layer at the output.

B. REGULARIZATION

In machine learning, one tries to indirectly maximize the performance of a model on a test set by optimizing the model on a training set. In order to prevent the model is too much optimized for the training set...
and hence losing its ability to perform well on the test set, one can restrict the optimization on the training set. This is done by introducing biases to the training problem which in effect expresses preferences for some solutions over the others. This process is known as regularization. One of the most widely used regularization technique is weight decay and in this case the Euclidean norm of the parameter vector is kept minimum during training, having a loss function of the form

\[ L(w) = L'(w) + \lambda \|w\|^2 \]

where \( L' \) is the loss function before regularization, \( w \) is the model parameter vector and \( \lambda \) is a constant. In this case the regularization term is a simple function of the model parameters. But, in general, it can be a more complex function of the model parameters as well as training data \( x \):

\[ L(w) = L'(w) + \lambda G(w, x) \]

In this work we make use of several such regularization functions.

3. TASK

We consider a classification task where a given sonar image is classified into 4 classes: cylinder, truncated cone, wedge and clutter. The first three classes represent regular geometric shapes whereas the last one is a composite class containing spurious detections and other objects that do not belong to the first three classes.

Figure 1 illustrates the four-class classification task considered in which the input to the classifier is a sonar image and the output is the probabilities of the classes considered.

We implemented a baseline system that realizes the task, by adding a classification head to a base-CNN as shown in Figure 2. In our implementation, we used the Inception-Resnet-V2 architecture as the base-CNN and a single layer fully connected neural network as the classification head.

The base-CNN is initialized with the original parameter values pre-trained on the optical images from the Imagenet dataset. Classification network is initialized to random values before the whole network (base-CNN and the classification network) is trained on the sonar images.
A. DATA SET

The data set used in this work consists of Synthetic Aperture Sonar (SAS) images collected using a HISAS 1030 sensor mounted on a HUGIN underwater vehicle. This is an automatically annotated data set where locations of the objects of regular shapes (cylinder, truncated cone, and wedge) are known. The sonar images were first sent through a blob detector and image snippets of size 299x299 pixels were extracted around each detection. In this way about 72000 snippets were collected, where a vast majority of the images belonged to the clutter class. More specifically, there are 1000 cylinder images, 1200 truncated cone images and 200 wedge images whereas 69600 snippets are clutter images. This is clearly a highly unbalanced data set and therefore class weighting was applied during training to counter this imbalance. About 90% of the total images were used as the training set and the remaining images were set aside as the test set. The training set was augmented through flipping along the across track direction and random translations. This resulted in a final training set size of 140000 images.

B. TRAINING AND EVALUATION

Before the training procedure is started, the base-CNN is initialized with the original, pre-trained parameter values, whereas the classification network is initialized with random values. Then the whole network is trained using the training set described above. We employed the update rule of stochastic gradient descent (SGD) with momentum together with a batch-size of 25 images. In each experiment, the system was trained for 20 epochs.

The loss function used in training of the baseline system is the binary cross entropy (BCE). That is

$$L_{\text{baseline}} = -\frac{1}{N} \sum_{i=1}^{N} \log P(C_{t_i})$$

where $t_i$ is the target class of the $i^{th}$ sample, $P(C_j)$ is the probability of object class $j$ and $N$ is the number of samples in the training set.

When modifications are introduced to the baseline architecture, the loss function is changed accordingly, essentially by adding a regularization term. Details of these modifications are described in the section on experiments.

Once the system, either the baseline or a modified version of the baseline is trained, its performance is evaluated on the test set.

We calculated several evaluation metrics on the test set after each training epoch.

- **Accuracy**: This is the ratio between the number of correctly classified images and the total number of images in the test set. This is a metric not suitable for a highly imbalanced data set like ours.

- **Average Recall**: We calculated Recall averaged over all classes. i.e.

$$R_{\text{ave}} = \frac{1}{C} \sum_{i=1}^{C} \frac{n_{ii}}{\sum_{j=1}^{C} n_{ij}},$$

where $n_{i,j}$ is the number of class $i$ images classified into class $j$ and $C$ is the number of classes.

- **Average Area Under the Curve**: Unlike the previous two, this metric does not depend on a particular threshold in classification. Therefore, this is a highly suitable metric for our problem. We first create receiver operating characteristics (ROC) curves, that is the graph of the true positive rate against the false positive rate, for each of the classes. Then the area under the ROC curve (AUC) is calculated for each class and the final metric is obtained by averaging AUC values for all classes.
4. EXPERIMENTS

We conducted two main lines of experiments, one with geometry information as direct inputs and the other with geometry information as a means of regularization.

A. GEOMETRY PARAMETERS AS AN INPUT

In this line of experiments, we slightly modified the baseline system so that geometry information becomes an additional input. Figure 3 shows the basic idea of this approach. As seen from the figure, modifications to the baseline architecture is minimum and the loss function is the same as in the baseline.

![Figure 3: Additional input configuration](image)

In this approach, we used only readily available imaging geometry information, i.e. ground range and sensor altitude. This two dimensional geometry information vector can be injected to the system at many different locations of the architecture. Out of these, we considered only two different locations as shown in Figure 4 and Figure 5. In both of these architectures, we advocate injecting geometry information at an early stage of processing, i.e. early fusion.

![Figure 4: Additional geometric information input concatenated with the input image](image)

Both in Figure 4 and Figure 5, we first create a matrix by replicating the 2D imaging geometry vector 299 times. Then the resulting matrix is passed through a convolution neural network layer that gives out a 3D tensor, which is shown in color violet. This whole operation is represented by the box names Information Transform in Figure 4 and Figure 5.

Once the 3D tensor representation is obtained, this is concatenated along the channel dimension with the input image (Figure 4) or with an intermediate feature map (Figure 5). In the base-CNN architecture, Inception-Resnet-V2, there is a set of layers called stem, and the output of this is a natural choice for extracting the feature map for concatenation in Figure 5.

Once the geometry parameters are injected in this manner, the whole network including the Information Transform network is trained on SAS images as described in Section 3.2. Note that in this case the loss function optimized is the same as that of baseline system.
B. GEOMETRY INFORMATION AS REGULARIZATION

In this line of experiments we used the imaging geometry information to derive a regularization term. The main idea of this approach is illustrated in Figure 6. As shown in this figure, the baseline architecture (the upper part of the network) is supplemented with a couple of operations (the lower part of the network) which are used to compute a regularization term that is added to the baseline loss function. These operations take the imaging geometry parameters as inputs and compute an intermediate representation $Z^B$ which is compared with the equivalent representation $Z^A$ derived from the input image. The regularization term is a measure of the distance between $Z^B$ and $Z^A$. By adding this loss to the optimization procedure in training, we try to keep $Z^A$ as close as possible to $Z^B$. In this way, information contained in imaging geometry parameters are transferred to the system parameters.

![Figure 5: Additional geometric information input concatenated with an intermediate feature map](image)

**Figure 5: Additional geometric information input concatenated with an intermediate feature map**

**Figure 6: Regularization configuration where geometry information is used to derive a regularization term**

Figure 7 shows more details of calculating $Z^A$ and $Z^B$. As this figure shows, in computing $Z^B$ imaging geometry is first used as an input to a SAS simulator. In order to simulate a SAS image of an object with a known geometric shape, we need ground range and sensor altitude as well as the object orientation and seafloor slope. Since only the ground range and the sensor altitude are readily available from sonar data records, the other parameters, object orientation and seafloor slope are estimated using the given SAS image. For this purpose, a home grown image processing algorithms are used. Once all the parameters are known, the SAS simulator can generate a simulated image. Then the stem of the baseCNN is used to transform the simulated image to a feature map $Z^B$.

The real SAS image is input to the baseCNN and the output of the stem $Z^A$ is the corresponding feature map.

$Z^A$ together with the previously calculated feature map $Z^B$ is used to calculate the regularization loss, as shown in Figure 8.

We considered two variants of regularization loss functions: Binary Cross Entropy (BCE) loss and Barlow Twins (BT) loss.  

---

N. Warakagoda and Ø. Midtgaard  
Utilizing imaging geometry meta-data for classification of SAS images  
Proceedings of Meetings on Acoustics, Vol. 47, 070011 (2022)  
Page 6
BCE loss $\mathcal{L}_{\text{BCE}}$ is defined using the following expression:

$$
\mathcal{L}_{\text{BCE}} = \frac{1}{N} \sum_{i}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)
$$

$$
p_i = \sigma(\text{MLP}(Z^A \| Z^B))
$$

$$
y_i = \begin{cases} 
1 & \text{if } Z^A \text{ and } Z^B \text{ from the same object} \\
0 & \text{if } Z^A \text{ and } Z^B \text{ from different objects} 
\end{cases}
$$

where

- $\sigma(\cdot)$: sigmoid function
- MLP: a multi-layer perceptron
- $Z^A \| Z^B$: concatenation of $Z^A$ and $Z^B$
- $N$: number of image pairs

One disadvantage of BCE loss is that one needs both similar and different simulated-real image pairs to calculate this loss. There are too many ways to draw different image pairs and in practice it is difficult to pick the different image pairs that have the highest impact on training.

BT-loss $\mathcal{L}_{\text{BT}}$ is defined using the following expressions:

$$
\mathcal{L}_{\text{BT}} = \sum_{i} (1 - C_{ij})^2 + \lambda \sum_{i} \sum_{j} C_{ij}^2
$$

where

$$
C_{ij} = \frac{\sum_{b} z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_{b} z_{b,i}^A z_{b,i}^A} \sqrt{\sum_{b} z_{b,j}^B z_{b,j}^B}}
$$

with $z_{b,i}^A$ being the $i$th element of the $b$th feature vector $Z^A$ and $z_{b,j}^B$ being the corresponding value for $Z^B$. This loss has the advantage that it can be defined using only the similar simulated-real image pairs.\(^9\)
5. RESULTS

We trained the baseline system, the two variants of the additional input configuration and the BCE and BT loss variants of the regularization configuration. The variation of the evaluation metrics, accuracy, average recall and average AUC for the baseline system are shown in Figure 9. Typically, the evaluation metrics improve initially and after a certain point they tend to decrease. This behaviour can be observed for recall and AUC, but the accuracy metric appears to rise during whole training.

![Figure 9: Development of evaluation metrics during training for the baseline system. Solid line shows the 25% smoothened value while the faded line shows the original value](image)

Table 1 shows the maximum value of each evaluation metric in the different experiments conducted. The winner with respect to each metric is shown in bold. We can see that all the winners except for AUC come from the regularization experiments. Baseline wins if the evaluation is performed with respect to AUC and no smoothing is done. Even in this case the regularization experiments have very close performance.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Variant</th>
<th>Accuracy SP=0</th>
<th>Recall SP=0</th>
<th>AUC SP=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Baseline</td>
<td>67.69</td>
<td>49.29</td>
<td>90.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>65.76</td>
<td>48.06</td>
<td>86.98</td>
</tr>
<tr>
<td>Additional Input</td>
<td>With image</td>
<td>89.65</td>
<td>50.82</td>
<td>88.43</td>
</tr>
<tr>
<td></td>
<td>With stem features</td>
<td>87.57</td>
<td>47.19</td>
<td>87.20</td>
</tr>
<tr>
<td>Regularization</td>
<td>BCE loss</td>
<td>91.99</td>
<td>59.85</td>
<td>86.23</td>
</tr>
<tr>
<td></td>
<td>BT loss</td>
<td>91.65</td>
<td>55.80</td>
<td>89.49</td>
</tr>
</tbody>
</table>

6. CONCLUSION

Imaging geometry parameters (sensor altitude and range) can be used to improve the classification performance. Inclusion of raw geometry parameter inputs lead to improved performance over the baseline. However, regularization with simulated data based on geometry parameters gives better performance than direct geometry inputs.
ACKNOWLEDGMENTS

We thank the Royal Norwegian Navy and Kongsberg Maritime for providing HISAS data used in the work. We also thank the NATO STO Centre for Maritime Research and Experimentation for providing the MUSCLE SAS data, which was collected with funding from the NATO ACT.

REFERENCES


