DEEP TRANSFER LEARNING ACROSS TARGETS AND SENSORS WITH SYNTHETIC APERTURE SONAR DATA

David P. Williams Applied Research Laboratory, Penn State University, State College, USA

1 INTRODUCTION

Convolutional neural networks (CNNs)[1] are a powerful family of deep-learning algorithms within the field of machine learning that now regularly achieve state-of-the-art performance on a wide range of image classification tasks. But to do so, vast amounts of labeled data are required. Unfortunately, in underwater remote-sensing applications, data collections can be time-consuming and prohibitively expensive. Moreover, when a new sensor is introduced, there is a desire to still leverage historical data collected by similar predecessor systems. In general, it is not feasible to wait for the execution of numerous onerous data collections in diverse geographical locations before being able to accurately assess the utility of the new system. However, that is often the practice when a new sensor is developed: automatic target recognition (ATR) algorithm developers start “from scratch” with little to no data. (A similar scenario unfolds if the class of targets of interest evolves to something that has not been encountered previously.) Aside from being untenable over the long term, this approach also squanders previous data collection and curation investments. The effort undertaken here considers the concept of transfer learning[2], with the aim to provide justification for the leveraging of existing sonar databases to ensure continued return on prior investment, even as sensing technology improves and target types evolve.

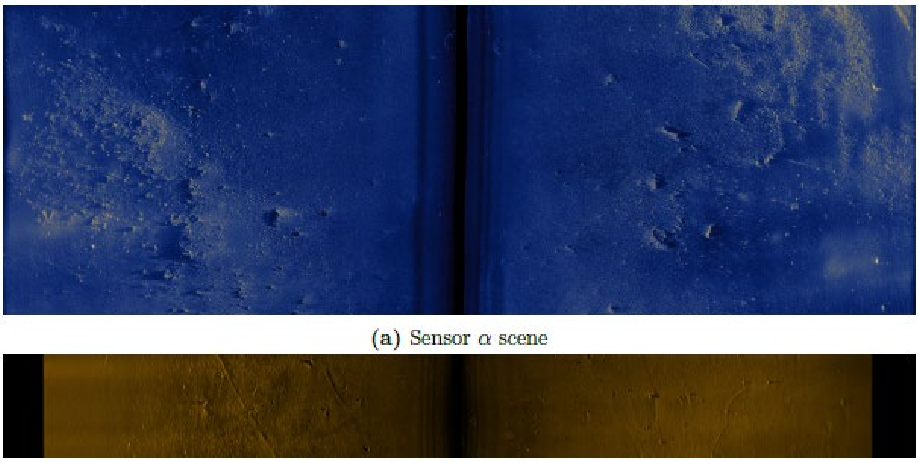
In object recognition with “natural” (i.e., optical) imagery, it has been shown that low-level features learned for one task are useful for related, but distinct tasks[3]. For example, the CNN filters learned to aid in the classification of different bird species can be reused for a novel target classification task in which the goal is to discriminate breeds of dogs. This phenomenon holds because edges, corners, oriented gradients and other fundamental building blocks used in hierarchical representations of images have intrinsic discriminatory value regardless of the final classification task. Although the modality examined in this work is different – acoustics rather than optics – a similar transfer should still be feasible because the underlying concepts remain true. For example, the features learned by a CNN to classify some man-made object like tires should also be transferable to a task that seeks to instead classify cinder blocks.

Other researchers[4-6] have obtained encouraging anecdotal results by employing the transfer-learning paradigm from optical imagery to sonar data, but no extensive *sonar-specific* study has been performed to date. Thus, the objective of the present work is to assess the feasibility of two forms of transfer learning for ATR with sonar data. More specifically, the study seeks to explore and establish the requirements and limitations for successfully leveraging historical data of legacy systems (i) to new systems that share a common sensing modality (e.g., sonars operating at different frequency bands) and (ii) to new but related tasks (e.g., classifying novel targets of interest).

The remainder of this report is organized as follows. Sec. 2 summarizes the data employed in the study, while Sec. 3 outlines the experimental set-up. Results are presented in Sec. 4, and the key findings are summarized in Sec. 5.

2 DATA

This study uses high-frequency (i.e., center frequency greater than 100 kHz) sonar data from two synthetic aperture sonar (SAS) systems, nominally named sensor α and sensor β. A large database of scene-level SAS images from sensor α was curated and divided into two disjoint sets, one for training and one for test purposes. The raw sonar data was reconstructed into imagery with the Advanced Synthetic Aperture Sonar Imagining eNgine (ASASIN) software[7]. Another database of scene-level SAS images from sensor β was also curated and divided into two disjoint sets, one for training and one for test purposes. The raw sonar data was again reconstructed into imagery with the ASASIN software. One example scene from each of the sensors is shown in Fig.1.

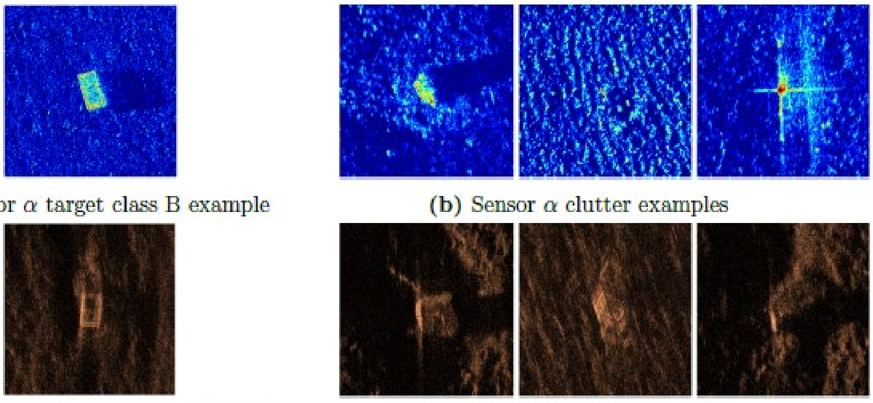


(a) Sensor α scene

(b) Sensor β scene

Figure 1: Example scenes from the two sensors.

The MondrianB detection algorithm[8], which is essentially an energy detector, was then applied to all scene images from both sensors. The output of the detection algorithm applied to a SAS scene is a set of smaller alarm chips. The alarm chips were manually labeled to create “operator truth.” Four types of man-made objects were considered as targets for this manual labeling procedure. Target class A contained three geometric shapes, while target class B consisted solely of a different shape, namely rectangular cuboids. All other alarms were considered clutter (i.e., non-targets). Example target chips and clutter chips from sensor α and sensor β are shown in Fig. 2.



(a) Sensor α target class B example

(b) Sensor α clutter examples

(c) Sensor β target class B example

(d) Sensor β clutter examples

Figure 2: Example target class B and clutter chips from the two sensors.

These resulting *labeled* alarm chips were then used for the classification experiments in this work. A summary of the data sets is shown in Table 1.

Table 1: Number of alarm chips in the data sets

	Training Data			Test Data		
	Clutter	Target Class		Clutter	Target Class	
Sensor	Class	A	B	Class	A	B
α	168454	1540	1213	41060	1462	92
β	10839	0	1200	3679	0	526

3 EXPERIMENTAL SET-UP

A basic CNN with an architecture of alternating convolutional *blocks* – comprising one or more convolutional layers – and pooling layers is used. The input to the CNN is a 300 pixel x 300 pixel SAS magnitude image chip, while the outputs of the CNN's final layer are the probabilities of an image belonging to each class (target or clutter). A high-level schematic of this architecture is shown in Fig. 3.

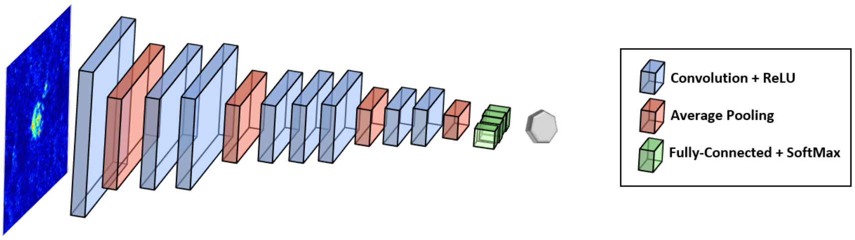


Figure 3: High-level illustration of the 8-convolutional-layer CNN architecture. The input is a SAS magnitude image, and the output is a scalar prediction of class membership.

The CNN contains 4 convolutional *blocks* with a total of 8 convolutional *layers*. Each filter is square, and only 4 filters are used in each convolutional layer. A rectified linear unit (ReLU) activation function is used after each convolutional layer, while a softmax activation is used at the output. All pooling layers use average pooling. The design of the architecture (and specifically the final pooling layer) ensures that the dense layer contains only 4 nodes.

Details about the specific CNN designed are provided in Table 2; the information shown is sufficient for recreating the network exactly. Here, brackets are used to convey the concept of convolutional *blocks*, in which there are multiple convolutional layers in between pooling layers. (The *i*th set of brackets contains the information about the convolutional layers in the *i*th block.) The convolutional block construct allows deeper networks, and thus greater complexity, without a proportional increase in the number of parameters to learn.

Table 2: Architecture details for the CNN

CNN Label	CNN Depth	Filters Per Conv. Layer	Filter Sizes (Pixels Per Side)	Pooling Factors	Number of Parameters
C-8	8	4	$\begin{bmatrix}9\end{bmatrix}\begin{bmatrix}6\\3\end{bmatrix}\begin{bmatrix}5\\3\\3\end{bmatrix}\begin{bmatrix}4\\3\end{bmatrix}$	4 3 2 2	2169

CNN training is performed in Python with the TensorFlow software library. Training uses an RMSprop optimizer with a binary-cross-entropy loss function. A batch size of 64 is used, with equal numbers selected from each class to combat the severe class-imbalance of the training data. One epoch of training is defined to be a set of 1000 batches. (A fractional epoch of training simply means that fewer than 1000 batches have been used.) Each batch is formed by randomly selecting chips from the full set of training data.

The amount of data needed to train a CNN is directly proportional to the number of parameters to be learned in the model. Therefore, by limiting which (convolutional) layers in a model are trainable, one can reduce training-data requirements. This observation motivates the common use of “freezing” certain layers of a CNN[3] when operating under a transfer-learning paradigm. With each additional convolutional layer that is frozen, the number of parameters that the CNN must learn decreases. The trade-off is that the flexibility of the model is reduced, meaning the capacity of the network to learn complex patterns within the data is limited.

4 RESULTS

Several specific scenarios will be defined to address various questions pertaining to operating under a transfer-learning paradigm. The first set of scenarios focus on the idea of target-concept transfer, while the second set relates to sensor transfer. For each of the scenarios considered, a CNN is also trained from scratch with the specified data, in order to make fair assessments regarding the utility of the transfer-learning paradigm.

4.1 Target Transfer

In this work, a base CNN classifier with the architecture defined in Table 2 was trained to discriminate targets from clutter using data from sensor α. Specifically, 100 epochs of training were executed when the targets were exclusively examples from target class A. No examples from target class B were present during this training. The resulting *baseline* CNN was then used as the initialization for various transfer-learning classification experiments.

Now consider the transfer-learning case, where a set of examples from target class B are made available for training refinement. In scenarios I and II, the target concept is *augmented* to include both target class A and target class B. In scenario III, the target concept is *changed* to consist of only target class B, while target class A is instead considered to be part of the *clutter* class. In scenarios I and III, there are 1213 target class B examples available; in scenario II, the number is limited to only 50.

In scenario I, during the (training) refinement, a specified number of convolutional layers are “frozen,” and thus the parameters therein are not permitted to change. The exercise is repeated for different numbers of frozen convolutional layers, from 0 to (all) 8 of the CNN; the layers nearest the input layer are frozen first. In scenario II, target class A was excluded from the (additional) training.

Regarding scenario I, Fig. 4 (left) shows that when the base CNN is extremely small – as here, with only 2169 parameters – it is preferable to allow maximum flexibility during the refinement process. This is in contrast to huge networks of millions of parameters[9], for which the majority of layers are typically held frozen for transfer-learning purposes.

Fig. 4 (right) shows that refinement that incorporates the new additional target class (B) into the target concept does not have a negative impact on the original target class (A), and that this is also similar to simply training from scratch with both target classes.

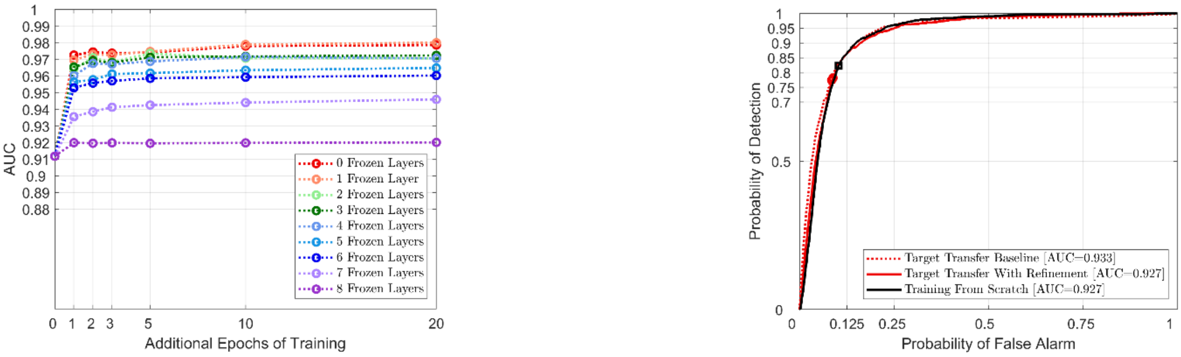


Figure 4: Scenario I: (left) Performance on target class B of sensor α when different numbers of convolutional layers are frozen in the CNN during re-training from target class A to target classes A and B. (right) Performance on target class A for the numbers of training epochs that achieved the best AUC. The baseline corresponds to the CNN trained only on target class A, with no refinement. The target-transfer case corresponds to 20 epochs of refinement for the CNN with no layers frozen; the training-from-scratch case corresponds to 100 epochs of training. These latter two cases use examples from both target classes A and B.

Regarding scenario I, Fig. 5 (left) shows that for target class B, refinement leveraging examples from the new target class is successful, and that this enables a comparable level of performance to be achieved in less time than training from scratch. Regarding scenario II, Fig. 5 (right) shows that transfer learning can still be successful when only a limited number of target class B examples are available to learn from. In fact, when this is the case, very minimal refinement of the extant CNN is required to tailor the classifier to the new target class. Moreover, employing transfer learning enables superior performance to training from scratch. This scenario highlights when transfer learning is most necessary: When faced with limited training data from the new target class of interest, leveraging the data relationships learned by the initial baseline classifier of a related task is invaluable for achieving good performance.

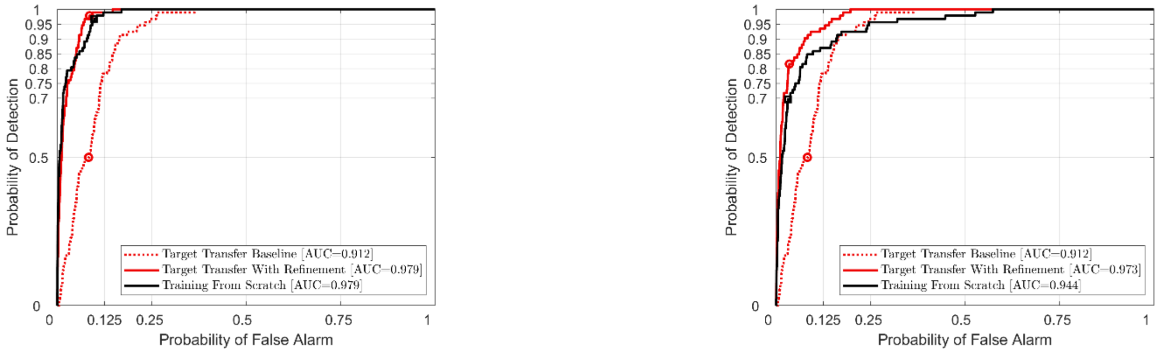


Figure 5: Scenarios I and II: Performance on target class B of sensor α when the number of target class B sensor α examples available to train on is (left) 1213 and (right) only 50. The curves correspond to the numbers of training epochs that achieved the best AUC. The baseline corresponds to the CNN trained only on target class A, with no refinement. For the {left, right} cases, the target-transfer case corresponds to {20, 0.3} epochs of refinement for the CNN with no layers frozen; the training-from-scratch case corresponds to {100, 1} epochs of training.

Regarding scenario III, Fig. 6 shows that transfer learning can successfully “unlearn” a target concept in order to be tailored to a completely new target class.

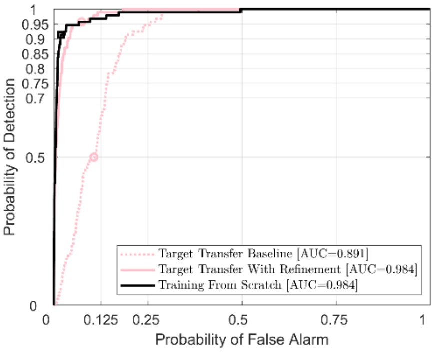


Figure 6: Scenario III: Performance on target class B of sensor α for the numbers of training epochs that achieved the best AUC when target class A is treated as (additional) *clutter*. The baseline corresponds to the CNN trained only on target class A as the target concept, with no refinement. The target-transfer case corresponds to 1 epoch of refinement, and the training-from-scratch case corresponds to 20 epochs of training.

4.2 Sensor Transfer

The base CNN was trained to discriminate targets from clutter using data from sensor α. Specifically, 100 epochs of training were executed when the targets were exclusively examples from target class A. This resulting CNN was used as the baseline classifier for scenario VI. In scenarios IV and V, a set of examples from target class B of sensor α were made available for training refinement; the aforementioned classifier was refined, with no layers frozen, for 20 epochs with both target classes A and B constituting the target concept. This resulting CNN was used as the *baseline* classifier initialization for scenarios IV and V.

We wish to investigate the feasibility of transferring a CNN trained on sensor α data so that it is appropriate for data collected by sensor β. For this transfer-learning refinement, we use data from sensor β exclusively. Specifically, we use clutter class examples and target class B examples, and the target concept is defined to be target class B. (There are no target class A examples from sensor β.) Using this data, the extant (baseline) CNN is refined, with no layers frozen. In scenario IV, there are 1200 target class B examples from sensor β available; in scenarios V and VI, the number is limited to only 50.

Regarding scenarios IV and V, in Fig. 7 the red dashed curves correspond to the CNN trained only on sensor α data. With additional training using sensor β data, performance rises to the solid red curves. These should be directly compared to training from scratch using only sensor β data, which is shown by the solid black curves. When there is a sufficient number of target class B examples from sensor β, transfer learning and training from scratch are comparable. However, when only a limited number of examples are available, transfer learning significantly outperforms training from scratch. The negative impact of having only a limited number of target class B examples from sensor β to train on can be observed by directly comparing the solid red curves.

Scenario VI examines the impact of needing to transfer across both target and sensor *jointly*, rather than only across sensor. In this scenario, the object of interest is target class B, but it is assumed that no examples of it were available from sensor α. Therefore, transfer to the sensor of interest, sensor β, must also include transfer across target type, from target class A to target class B. To complicate matters, only a limited number of target class B examples from sensor β is available.

Results from scenarios V and VI are collected and presented together in Fig. 8. Here, the blue dashed curve is the result of training with only sensor α data, and more specifically only target class A data. By refining with target class B examples from sensor β, performance improves to the blue solid curve. The red dashed curve is the result of training with only sensor α data, but target class B examples were available. By refining with target class B examples from sensor β, performance improves to the red solid curve.

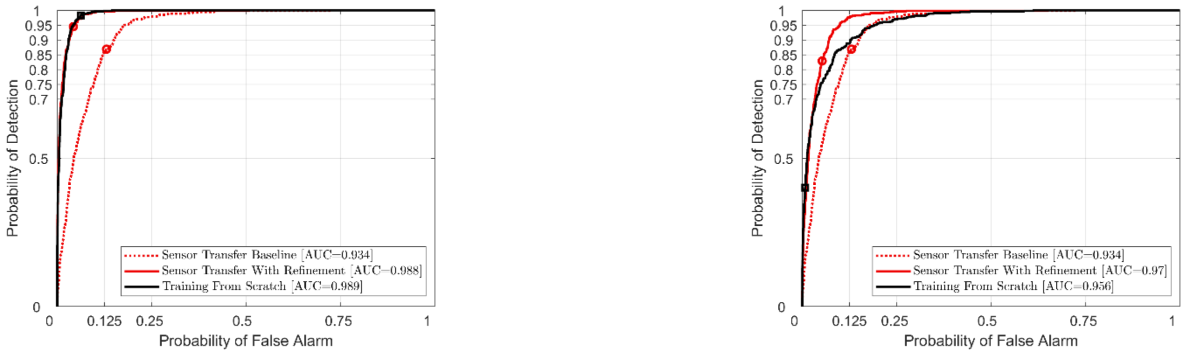


Figure 7: Scenarios IV and V: Performance on target class B of sensor β when the number of target class B sensor β examples available to train on is (left) 1200 and (right) only 50. The curves correspond to the numbers of training epochs that achieved the best AUC. The baseline corresponds to the CNN trained on target classes A and B using only sensor α data, with no refinement. For the {left, right} cases, the sensor-transfer case corresponds to {50, 0.4} epochs of refinement for the CNN with no layers frozen; the training-from-scratch case corresponds to {50, 20} epochs of training.

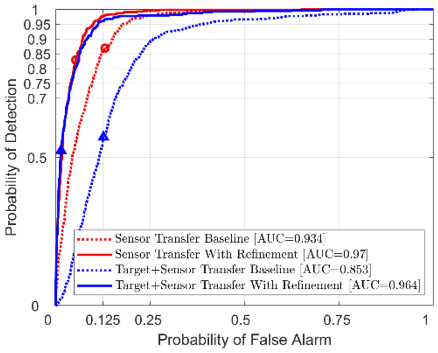


Figure 8: Scenario VI: Performance on target class B of sensor β when only 50 examples of target class B of sensor β are available for training. The curves correspond to the numbers of training epochs that achieved the best AUC. The sensor-transfer baseline corresponds to the CNN trained on target classes A and B using only sensor α data. The target-and-sensor transfer baseline corresponds to the CNN trained on only target class A using only sensor α data. The sensor-transfer case corresponds to 0.4 epochs of refinement for the CNN with no layers frozen; the target-and-sensor transfer case corresponds to 3 epochs of refinement for the CNN with no layers frozen.

5 CONCLUSION

An extensive set of experiments was undertaken to explore the feasibility of transfer learning with CNNs and SAS imagery. The main findings are as follows.

When using a *tiny* CNN in a transfer-learning context, there is not a benefit to freezing certain convolutional layers. Because of the small size of the CNN, the reduction in training time from freezing certain layers is not significant. Instead, the greater network flexibility afforded by not freezing any layers is more valuable.

When the target concept is *augmented* to include an additional target class, transfer learning can be successful, and without degrading performance on the original target class.

When a sufficient number of examples from the new target class is available for training, transfer learning can more quickly attain a given level of performance compared to training the CNN from scratch, but the final performance will be similar. However, when the number of examples from the new target class is limited, there is a distinct advantage to employing a transfer-learning paradigm. Moreover, in this case, very little refinement is needed to effect the successful target-concept transfer. However, it remains an open question as to how many examples from the new target class are required, and also how many examples are needed before training from scratch could perform comparably. Ongoing and future experiments are investigating these topics.

When the target concept is *changed* to a different target class, transfer learning can again be successful. That is, the process can completely “unlearn” a target concept in order to be tailored to a new target class. But again, training from scratch would perform comparably provided sufficient training data were available.

When classification is to be made using data from a new *sensor*, transfer learning is indeed viable. In this study, the two data sources were both high-resolution SAS sensors that produce imagery of roughly similar resolution. It remains an open question regarding how similar the data products must be for sensor transfer to be successful. For instance, would transfer from a side-scan sonar to a SAS sensor, or vice versa, be feasible?

Finally, it was also observed that joint transfer across both target concept and sensor was possible. But it was also discovered that it is preferable to first transfer across target concept if there is data available from the first sensor, before attempting to transfer across sensors.

The results in this study should be useful for informing work in which data of a new target concept and/or data from a new sensor must be considered. For example, suppose a CNN had been trained for a given set of targets of interest. It is now clear that that extant CNN can be quickly refined via transfer learning in order to tailor the classifier to a new target of interest. Similarly, if a CNN had been trained with data collected from a certain sensor, it is now understood that the extant CNN should be adaptable via transfer learning to function with data collected by a new, yet related sensor. These findings should reduce the need for extensive new data collections – either of a novel target type or from a new sensor – and thereby save resources.

6 REFERENCES

1. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436-444, 2015.
2. R. Caruana, “Learning many related tasks at the same time with backpropagation,” Advances in Neural Information Processing Systems, vol. 7, 1994.
3. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?”, Advances in Neural Information Processing Systems, vol. 27, 2014.
4. J. McKay, I. Gerg, V. Monga, and R. Raj, “What's mine is yours: Pretrained CNNs for limited training sonar ATR,” Proc. IEEE OCEANS, 2017, pp. 1-7.
5. M. Emigh, B. Marchand, M. Cook, and J. Prater, “Supervised deep learning classification for multi-band synthetic aperture sonar,” Proceedings of the 4th International Conference on Synthetic Aperture Sonar and Synthetic Aperture Radar, vol. 40, 2018, pp. 140-147.
6. N. Warakagoda and O. Midtgaard, “Transfer-learning with deep neural networks for mine recognition in sonar images,” Proceedings of the 4th International Conference on Synthetic Aperture Sonar and Synthetic Aperture Radar, vol. 40, 2018, pp. 115-122.
7. I. Gerg, D. Brown, S. Wagner, D. Cook, B. O'Donnell, T. Benson, and T. Montgomery, “GPU acceleration for synthetic aperture sonar image reconstruction,” Proc. IEEE OCEANS, 2020.
8. D. Williams, “On the utility of multiple sonar imaging bands for underwater object recognition,” Proc. IEEE OCEANS, 2022.
9. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.