Convolutional Neural Networks for Feature Extraction and Automated Target Recognition in Synthetic Aperture Radar Images

John Geldmacher, Christopher Yerkes, and Ying Zhao, *Memeber, IEEE*

Abstract—Advances in the development of deep neural networks and other machine learning algorithms combined with ever more powerful hardware and the huge amount of data available on the internet has led to a revolution in ML research and applications. These advances present massive potential and opportunity for military applications such as the analysis of Synthetic Aperture Radar (SAR) imagery. SAR imagery is a useful tool capable of capturing high resolution images regardless of cloud coverage and at night. However, there is a limited amount of publicly available SAR data to train a machine learning model. This paper shows how to successfully dissect, modify, and re-architect cross-domain object recognition models such as the VGG-16 model, transfer learning models from the ImageNet, and the k-nearest neighbor (kNN) classifier. The paper demonstrates that the combinations of these factors can significantly and effectively improve the automated object recognition (ATR) of SAR clean and noisy images. The paper shows a potentially inexpensive, accurate, transfer and unsurpervised learning SAR ATR system when data labels are scarce and data are noisy, simplifying the whole recognition for the tactical operation requirements in the area of SAR ATR.

Keywords—k-nearest neighbor, kNN, deep learning, Synthetic Aperture Radar images, SAR images, transfer learning, VGG-16

I. Introduction

The analysis and classification of targets within imagery captured by aerial and space-based systems provides the US intelligence community and military geospatial intelligence (GEOINT) personnel with important insights into adversary force dispositions and intentions. It has also entered the mainstream thanks to openly available tools like Google Earth. The high resolution of space-based sensors and common use of overhead imagery in everyday life means with the exception of decoys and camouflage, an average person is now reasonably capable of identifying objects in electro-optical (EO) imagery. EO images are, however, limited by cloud coverage and daylight. About half of the time when a satellite in low earth orbit could image a target it will be night, necessitating the use of either an infrared (IR) or a synthetic aperture radar (SAR) sensor. Both IR and SAR images require a trained imagery analyst to reliably identify targets. A repetitive and time consuming task that currently requires human expertise, importantly and creatively, is an ideal problem for deep learning. Automated target recognition (ATR) seeks to reduce the total work load of analysts so that their effort can be spent on the more human-centric tasks like presenting and explaining intelligence to a decision maker. ATR is also intended to reduce the time from collection to exploitation by screening images at machine speeds rather than manually. SAR ATR is complicated by the available data to train and assess machine learning models. Unlike other image classification tasks, there is not a large and freely available amount of training data for researchers. Further, the data that is publicly available only covers a small fraction of the types of targets an effective SAR ATR system would be required to identify.

II. Advantages and Challenges of Synthetic Aperture Radar (SAR) Images, Data Description, and Related Work

Synthetic Aperture Radar (SAR) is a radar mounted to a moving platform that uses the platform's motion to approximate the effect of a large antenna. The high resolution that can be achieved by creating a radar with an effective aperture much greater in size than is physically possible allows for radar returns to be processed into images similar to what can be achieved with a camera [19]. SAR imagery provides an important tool for the United States Intelligence Community and military geospatial intelligence (GEOINT) personnel because of its all-weather, day/night collection capability. Additionally, some wavelengths that SAR imaging systems operate in have a degree of foliage and ground penetrating capability allowing for the detection of buried objects or objects under tree cover that would not be observable by other sensors such as EO sensors.

These important advantages of SAR imaging for GEOINT analysts do come with some significant drawbacks inherent to SAR images. Because SAR images are not true optical images, they are susceptible to noise generated by constructive and destructive interference between radar reflections that appear as bright or dark spots called "speckle" in the image [19]. Also various materials and geometries will reflect the radar pulses differently, creating blobs or blurs that can obscure the objects physical dimensions. These issues, as well as problems caused

This will certify that all author(s) of the above article/paper are employees of the U.S. Government and performed this work as part of their employment, and that the article/paper is therefore not subject to U.S. copyright protection. No copyright. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). In: Proceedings of AAAI Symposium on the 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks, November 11-12, 2020, Virtual, published at http://ceur-ws.org/

J. Geldmacher, C. Yerkes, and Y. Zhao are with the Department of Information Sciences, Naval Postgraduate School, Monterey, CA, USA

C. Yerkes is also with Oettinger School of Science and Technology Intelligence, National Intelligence University, Bethesda, MD, USA

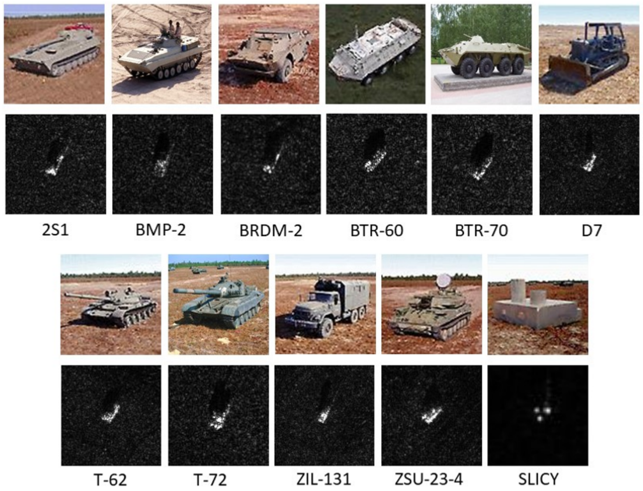


Fig. 1. Example Photographs and MSTAR Images by Class. Photograph of BMP-2 from $https://www.militaryfactory.com/armor/detail.asp?armor\_id = 50$, photograph of BTR-70 from $https://military.wikia.org/wiki/BTR-70$. All other photographs and SAR images adapted from the MSTAR dataset.

by Doppler shift in moving objects and radar shadows, make the identification and classification of objects in SAR images a difficult and tedious task that also requires a well-trained and experienced analyst.

The the Moving and Stationary Target Acquisition and Recognition (MSTAR) data set is a publicly available data set consisting of synthetic aperture radar images of the following 10 classes of military vehicles:

1) 2S1: former Soviet Union (FSU) self-propelled artillery
2) BMP-2: FSU infantry fighting vehicle
3) BRDM-2: FSU armored reconnaissance vehicle
4) BTR-60: FSU armored personnel carrier
5) BTR-70: FSU armored personnel carrier
6) D7: Caterpillar tractor frequently used in combat engineering roles
7) T-62: FSU main battle tank
8) T-72: FSU main battle tank
9) ZIL-131: FSU general purpose 6x6 truck
10) ZSU-23-4: FSU self-propelled anti-aircraft gun
11) SLICY: the Sandia Laboratories implementation of cylinders (SLICY). The SLICY consisting of simple geometric shapes such as cylinders, edge reflectors, and corner reflectors which could be used for calibration of sensors or for modeling the propagation of radar reflections.

Fig. 1 shows the example photographs and MSTAR images by class. It demonstrates the difficulties an imagery analyst would face when identifying targets in SAR imagery. The vehicles that are easily recognizable in photos become blurs in SAR images. Due to its public availability and ease of access for researchers, the data set has become the standard for SAR image Automated Target Recognition (ATR) classification research.

ATR in SAR imagery using "shallow" classification methods, which are traditional classifiers applied directly to SAR images with the breakthrough feature extraction layers as demonstrated in convolutional neural networks, produced generally good results. An SVM method proposed by [27] achieved 91% accuracy in a five-class test [27], while a Bayesian classifier reported a 95.05% accuracy in a 10-classes test [13].

In recent years, the work on classification of SAR imagery has focused on the use of CNNs. In 2015, Morgan showed that a relatively small CNN could achieve 92.1% accuracy across the 10-class of the MSTAR dataset, roughly in line with the shallow methods previously explored. Morgan's method also showed that a network trained on nine of the MSTAR target classes could be retrained to include a tenth class 10-20 times faster than training a 10-class classifier from scratch. The ability to more easily adapt the model to changes in target sets represents an advantage over shallow classification techniques [11]. This is especially valuable in a military ATR context given the fluid nature of military operations, where changes to the order of battle may necessitate updating a deployed ATR system. Malmgren-Hansen et al., explored transfer learning from a CNN pre-trained on simulated SAR images generated by using ray tracing software and detailed computer aided design models of target systems. They showed that model performance was improved, especially in cases where the amount of training data was reduced [10]. The technique of generating simulated SAR images for training could also be valuable in a military SAR ATR context where an insufficient amount of training data for some systems may exist.

The use of a linear SVM as a replacement for the softmax activation that is typically used for multiclass classifiers in neural networks has been shown to be potentially more effective for some classification tasks [22]. Transfer learning from ImageNet to MSTAR with an SVM classifier was explored by [1] in 2018. Their methodology compared the performance of an SVM classifier trained on mid-level feature data extracted from multiple layers from AlexNet, GoogLeNet, and VGG-16 neural networks without retraining the feature extracting network. Although they reported over 99% accuracy when classifying features extracted from mid-level convolutional layers from AlexNet, performance of the SVM on features from fully-connected layers did not achieve 80% accuracy. The best performance reported on from the VGG-16 architecture was 92.3% from a mid-level convolutional layer, but only 49.2% and 46.3% from features extracted in the last two fully-connected layers [1].

III. Transfer Learning and Feature Extraction

CNNs require a very large amount of data to train an accurate model and it is not uncommon for data sets with tens or even hundreds of thousands of images to be used when training a model. Transfer learning presents one possible solution when training a CNN on a limited data set by leveraging knowledge from a previously learned source task to aid in learning a new target task [14]. In an image classification problem, transfer learning works by training a CNN on a data set that has a very large number of images and freezing the parameters for a certain number of layers and extracting mid-level feature representations before training further layers and the final classification layer [7].

ImageNet is an open source labeled image database organized in a branching hierarchical method of “synonym sets” or “synsets”. For example, the “tank” synset is found in a tree going from vehicle to wheeled vehicle to self-propelled vehicle to armored vehicle to tank. The ImageNet database consists of over 14 million labeled images organized into over 21,000 synsets. Pre-trained ImageNets are often used in transfer learning.

Transfer learning is typically used when source and target tasks are not too dissimilar in order to avoid negative transfer. Negative transfer occurs when the features learned in the transfer learning method actually handicap the model performance [14]. However, transfer learning becomes more useful when a curious phenomenon that many deep neural networks trained on natural images learn similar features across images from different domains.

Evidence shows that low and mid-level features could represent basic ATR features in images such as texture, corners, edges, and color blobs [9], and the low and mid-level neural network feature extraction function resembles the actual biological and human neurons’ function. Low and mid-level of features extracted from CNNs are likely common across even dissimilar data sets. A transfer learning approach between different domains is feasible and ATR tasks are evidently successful in cross-domain applications [2], [8]. For example, the application of transfer learning to remote sensing target detection and classification was studied [16], which showed that a CNN classifier trained on a photographic data set could be retrained to perform remote sensing classification of ships at sea with a good performance.

IV. SAR ATR: Feature Extraction Combined with Shallow Classifiers

A. *Multistep Classifier*

In practice and in cross-domain applications, very few people train an entire CNN from scratch because it is relatively rare to have a data set of sufficient size. For this reason, transfer learning with feature extraction combined with shallow classifiers are suitable choices for SAR images.

The network architecture employed in this paper was a modified VGG-16 architecture [18]. The original VGG-16 architecture is shown in Fig. 2. It consists of two or three linked convolutional/pooling blocks, three fully-connected layers, and a softmax activation in the end to determine the class label. The network employs a 3x3 kernel and a stride of one so that each pixel is the center of a convolutional step. The architecture has been modified to freeze model weights for the first two convolutional/pooling blocks (e.g., the first six layers). The model top has also been replaced with a fully-connected layer, a dropout layer to mitigate overfitting, and two final fully-connected layers with a softmax activation for classification [16]. This is also referred as a modified VGG-16 architecture or a VGG-16 architecture in this paper. It was initialized with the ImageNet weights and had the first two convolutional/pooling blocks frozen for training to take advantage of the broad feature detection of the pre-trained network as shown in Fig. 3. Our method, as in Fig. 4, shows that the dense layer of 1024 features extracted are saved and used as the input to a shallow classifier.

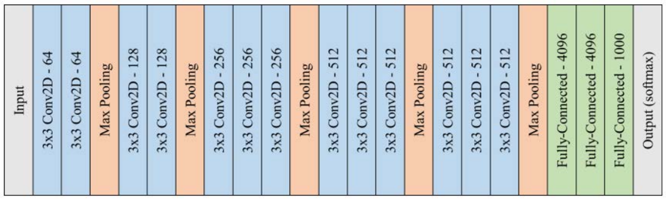


Fig. 2. The Original VGG-16 Architecture

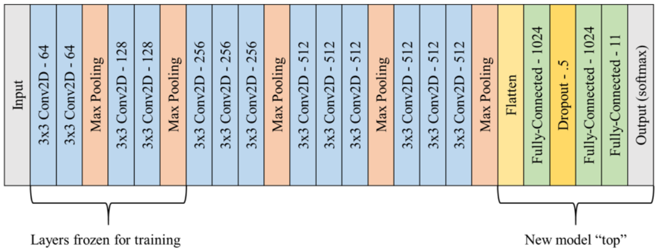


Fig. 3. VGG-16 with Transfer learning by freezing the first six layers where the weights taken directly from the ImageNet

In our experiment, the standard VGG-16 model is implemented in the Keras application program interface (API) with TensorFlow as the backend. The ImageNet weights available in the Keras are ported from the Visual Geometry Group at Oxford University that developed the VGG architecture for ILSVRC-2014 localization and classification tasks [18]. We also use, Orange, which is an open source data science and machine learning toolkit that allows users to easily manipulate data through a graphical user interface. Orange has several built-in machine learning algorithms and simplifies the data management and pre-processing requirements to allow users to experiment with approaches to machine learning and data science [3].

A CNN is trained on the 2200 training images with a 20% validation split. The training and test data were both then run through the retrained neural network. The last fully connected layer before the neural network’s output was saved as a 1024-dimensional vector for each image as shown in Fig. 3.

The extracted features run through the Orange workflow are pictured in Fig. 5. The precision and recall are used to compare the base CNN performance, kNN, SVM, and random forest

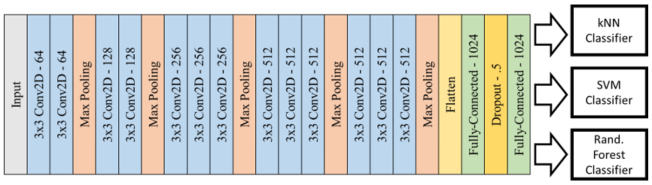


Fig. 4. The multistep classifier: Extract features from the VGG-16 and then apply a shallow classifier

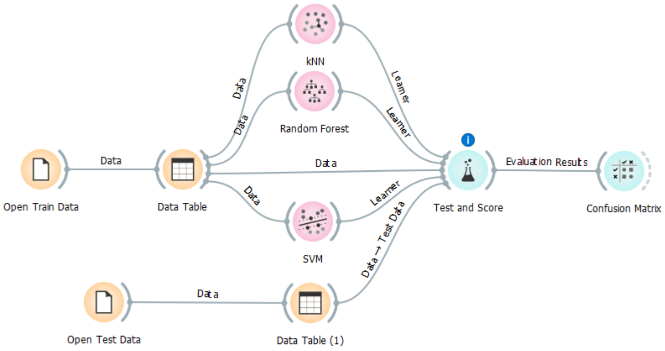
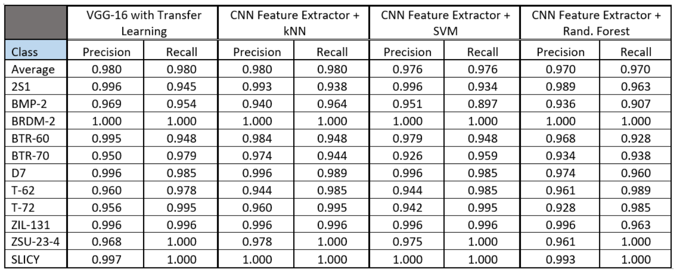


Fig. 5. The Orange Workflow



	VGG-16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + Rand. Forest	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.980	0.980	0.980	0.980	0.976	0.976	0.970	0.970
2S1	0.996	0.945	0.993	0.938	0.996	0.934	0.989	0.963
BMP-2	0.969	0.954	0.940	0.964	0.951	0.897	0.936	0.907
BRDM-2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
BTR-60	0.995	0.948	0.984	0.948	0.979	0.948	0.968	0.928
BTR-70	0.950	0.979	0.974	0.944	0.926	0.959	0.934	0.938
D7	0.996	0.985	0.996	0.989	0.996	0.985	0.974	0.960
T-62	0.960	0.978	0.944	0.985	0.944	0.985	0.961	0.989
T-72	0.956	0.995	0.960	0.995	0.942	0.995	0.928	0.985
ZIL-131	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.963
ZSU-23-4	0.968	1.000	0.978	1.000	0.975	1.000	0.961	1.000
SLICY	0.997	1.000	1.000	1.000	1.000	1.000	0.993	1.000

Fig. 6. Comparison of VGG-16 with the multistep classifiers

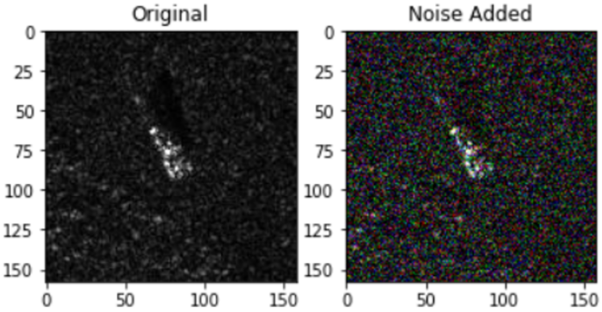
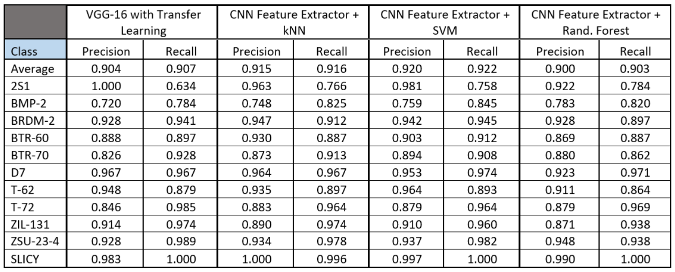


Fig. 7. Examples of Noisy SAR images



	VGG-16 with Transfer Learning		CNN Feature Extractor + kNN		CNN Feature Extractor + SVM		CNN Feature Extractor + Rand. Forest	
Class	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Average	0.904	0.907	0.915	0.916	0.920	0.922	0.900	0.903
2S1	1.000	0.634	0.963	0.766	0.981	0.758	0.922	0.784
BMP-2	0.720	0.784	0.748	0.825	0.759	0.845	0.783	0.820
BRDM-2	0.928	0.941	0.947	0.912	0.942	0.945	0.928	0.897
BTR-60	0.888	0.897	0.930	0.887	0.903	0.912	0.869	0.887
BTR-70	0.826	0.928	0.873	0.913	0.894	0.908	0.880	0.862
D7	0.967	0.967	0.964	0.967	0.953	0.974	0.923	0.971
T-62	0.948	0.879	0.935	0.897	0.964	0.893	0.911	0.864
T-72	0.846	0.985	0.883	0.964	0.879	0.964	0.879	0.969
ZIL-131	0.914	0.974	0.890	0.974	0.910	0.960	0.871	0.938
ZSU-23-4	0.928	0.989	0.934	0.978	0.937	0.982	0.948	0.938
SLICY	0.983	1.000	1.000	0.996	0.997	1.000	0.990	1.000

Fig. 8. Comparison of VGG-16 with the multistep classifiers for added noise

classifiers. For the kNN classifier, k was set to 11 and weighted Euclidian distance was used to determine which class label to assign to test images. A sigmoid kernel was used in the SVM classifier, and the random forest consisted of 10 decision trees. These settings were unchanged in experiments two and three.

B. Results

The baseline model, which is the modified VGG-16, is shown in Fig. 3. The modified VGG-16 without transfer learning and trained exclusively on MSTAR, resulted in an average precision and recall of 0.96. The same modified VGG-16 with full transfer learning of the convolutional layers with weights from ImageNet resulted in an average precision and recall of 0.88. Although, the transfer learning approach has the advantage of converging much more quickly than the CNN initialized with random weights, the full transfer learning of all convolutional weights and only retraining the CNN top did not match the performance of the non-transfer learning approach, suggesting some negative transfer occurs in the later convolutional layers. As shown in Fig. 6, the best performed is the modified VGG-16 with partial transfer learning in Fig. 3 and resulted an average precision and recall of 0.98. The multistep classifier using a kNN classifier in Fig. 4 was able to match the best baseline performance with an average precision and call of 0.98, while the SVM and random forest classifiers fell short of the baseline model's performance.

C. Adding Noise

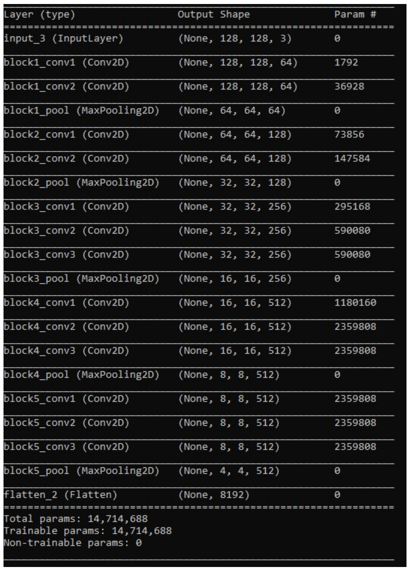
As described before, ATR of SAR images are typically sensitive to the noise in the images. A CNN is known to be vulnerable to the noisy models both from environmental perturbation or adversarials' deliberated manipulations [6], [21]. To study the effect, random Gaussian noise with a noise factor of 0.2 was added to the images from the data set. Fig. 7 shows an example of an original SAR image with one added noise. The feature extraction from CNN and follow on shallow classification process was then repeated without retraining the base model in order to test the robustness of the model. The baseline model (i.e., modified VGG-16 model) was then retrained on the noisy images for 30 epochs and accuracy was compared.

Neither the neural network nor any of the multistep classifiers proved robust enough to handle the addition of random noise to the images. However, after retraining the kNN and SVM multistep classifiers perform better than the modified VGG-16 with partial transfer learning.

V. Discussion

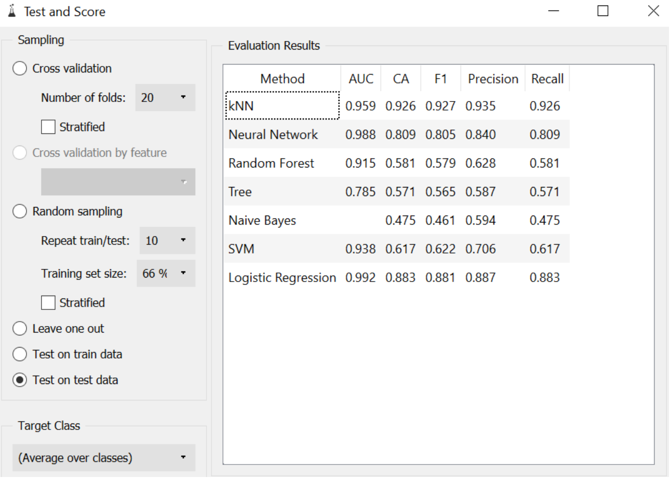
Performance on the SLICY class is of interest because it demonstrates the model's ability to discriminate an invalid target from a valid target. All other classes, with the exception of the D7, are former Soviet Union military equipment. The D7 is a Caterpillar bulldozer. Up-armored versions of the D7 and related equipment are often used in combat engineering roles. In a military context this means they are likely to be a valid target. As demonstrated by the high precision and recall in this class across all models, valid targets are very rarely classified as a SLICY (high precision) and the random objects are not being accepted as valid targets (high recall).

The performance of the kNN classifier is also notable since the use of an SVM for classification after feature extraction is previously studied; however, little research has been done on



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 128, 128, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 128, 128, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 128, 128, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 64, 64, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 64, 64, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 64, 64, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 32, 32, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 32, 32, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 16, 16, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 16, 16, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 8, 8, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 8192)              0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

Fig. 9. VGG-16 TensorFlow architecture layout



Sampling

○ Cross validation

Number of folds: 20

□ Stratified

○ Cross validation by feature

○ Random sampling

Repeat train/test: 10

Training set size: 66 %

□ Stratified

○ Leave one out

○ Test on train data

● Test on test data

Target Class

(Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
kNN	0.959	0.926	0.927	0.935	0.926
Neural Network	0.988	0.809	0.805	0.840	0.809
Random Forest	0.915	0.581	0.579	0.628	0.581
Tree	0.785	0.571	0.565	0.587	0.571
Naive Bayes		0.475	0.461	0.594	0.475
SVM	0.938	0.617	0.622	0.706	0.617
Logistic Regression	0.992	0.883	0.881	0.887	0.883

Fig. 10. Multistep classifiers results: VGG-6 transfer learning features + k-means (k=2048) + kNN & other methods in Orange

the performance of kNN rather than a softmax activation for neural network output.

To further explore the relations of feature extraction, transfer learning, and kNN, we ran an additional experiment where we first extracted the transfer weights of the first six layers of the VGG-16 architecture from ImageNet. Since the flatten dimension is 32x32x128=131,072, as shown in Fig. 9, we applied the unsupervised learning k-means algorithm to group the 131,072 dimension into 2048 clusters. The reasoning here is that the first six layers probably embed the best features (texture, corners, edges, and color blobs) that can be used for classification. Finally, We performed kNN and other supervised learning methods in Orange based on the 2048 dimensional train and test data. Fig. 10 shows the test data results from Orange for the VGG6-transfer-kmeans-kNN method with an average precision and recall of 0.93. The six layers of transfer learning together with k-means and kNN provide an inexpensive (without GPU or AWS, for example) and no supervised learning or no class labels required approach for SAR ATR. Recently, various learning-to-hash algorithms [24] are used to approximate the exact nearest neighbors, which translates a supervised learning problem and kNN into an index/search problem [15], and simplifies the whole recognition for the tactical operation requirements in the area of SAR ATR. If there are no class labels of SAR available, our multistep classifiers with transfer learning and kNN can provide an unsupervised classification with a high accuracy and confidence to match an object which looks like another object seen before.

Fig. 10 also shows the comparison of kNN with other supervised learning methods. The kNN method is the best among all the methods for the average precision and recall, where classification of the SLICY has a precision of 0.98 and a recall of 1. The future work is to test on more and different data sets (e.g., EO and IR data) to validate if the multistep methods can apply to cross-domain ATR problems.

VI. Conclusion

Cross-domain transfer learning from photographs to SAR imagery is effective for training a neural network both for feature extraction and classification. A retrained neural network can function as an efficient feature extractor for training a shallow classifier. kNN and SVM classifiers are potentially useful replacement for softmax activation in a neural network. Multistep classification methods using a shallow classifier trained on features extracted from a neural network, outperformed the base neural network when tested on noisy data and as the amount of training data decreases. This is valuable to improve CNN in a broader machine vision community by applying feature extraction followed by shallow classifiers for clean and noisy images. Transfer learning and kNN multistep classification methods could be significant in terms of setting up a robust image indexing system with minimum supervised training and learning required.

Currently, the analysis community does not have an established standard for the percent of correctly identified targets by an imagery analyst. Instead the analysis relies on the user's experience and confidence in their own work, providing responses such as "possible main battle tank" or "likely BMP-2", and thus a direct comparison to expert-level performance is difficult to establish. Both the baseline model employing transfer learning and the shallow classifiers using a neural network as a feature extractor performed with a high degree of accuracy and would be valuable in an operational context as an aid to GEOINT analysts.

REFERENCES

[1] Al Mufti, M., Al Hadhrami, E., Taha, B., & Werghi, N. (2018). Automatic target recognition in SAR images: Comparison between pre-trained CNNs in a tranfer learning based approach. 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). https://doi.org/10.1109/ICAIBD.2018.8396186

[2] Chen, D., Liu, S., Kingsbury, P., Sohn, S., Storlie, C. B., Habermann, E. B., Naessens,J. M., Larson, D. W. & Liu, H. (2019). Deep learning and alternative learning strategies for retrospective real-world clinical data. Digital Medicine (2019)2:43 ; https://doi.org/10.1038/s41746-019-0122-0.

[3] Demšar, J., Zupan, B., Leban, G., & Curk, T. (2004). Orange: From Experimental Machine Learning to Interactive Data Mining. In Knowledge Discovery in Databases: PKDD 2004 (Vol. 3202). Springer. https://doi.org/10.1007/978-3-540-30116-5_58

[4] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recogition. https://doi.org/10.1109/CVPR.2009.5206848

[5] Gandhi, R. (2018, June 7). Support Vector Machine—Introduction to Machine Learning Algorithms. Towards Data Science. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. The MIT Press. ISBN: 0262035618

[7] Kang, C., & He, C. (2016). SAR image classification based on the multi-layer network and transfer learning of mid-level representations. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS. https://doi.org/10.1109/IGARSS.2016.7729290

[8] Khosravi, P., Kazemi, E., Zhan, Q., Malmsten, J. E., Toschi, M., Zisimopoulos,P., Sigaras, A., Lavery, S., Cooper, L., Hickman, C., Meseguer, M., Rosenwaks, Z., Elemento, O., Zaninovic, N. & Hajirasouliha, I. (2019). Deep learning enables robust assessment and selection of human blastocysts after in vitro fertilization,Digital Medicine (2019)2:21; https://doi.org/10.1038/s41746-019-0096-y

[9] Liu,L., Chen,J., Fieguth,P., Zhao, G., Chellappa, R. & Pietikäinen, M. (2019). From BoW to CNN: Two Decades of Texture Representation for Texture Classification. International Journal of Computer Vision (2019)127:74–109

[10] Malmgren-Hansen, D., Kusk, A., Dall, J., Nielsen, A., Enghold, R., & Skriver, H. (2017). Improving SAR Automatic Target Recognition Models With Transfer Learning From Simulated Data. IEEE Geoscience and Remote Sensing Letters, 14(9), 1484–1488.

[11] Morgan, D. (2015). Deep convolutional neural networks for ATR from SAR imagery. Algorithms for Synthetic Aperture Radar Imagery XXII, 9475. https://doi.org/10.1117/12.2176558

[12] Nielsen, M. (2015). Neural Networks and Deep Learning. Determination Press. http://neuralnetworksanddeeplearning.com/

[13] O'Sullivan, J. A., DeVore, M. D., Kedia, V., & Millier, M. I. (2001). SAR ATR performance using a conditionally Gaussian model. IEEE Transactions on Aerospace and Electronic Systems, 37(1), 91–108. https://doi.org/10.1109/7.913670

[14] Pan, S., & Yang, Q. (2010). A Survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10). https://doi.org/10.1109/TKDE.2009.191

[15] Peng, T., Boxberg, M., Weichert, W., Navab, N. Marr, C. (2019). Multi-task learning of a deep k-nearest neighbour network for histopathological image classification and retrieval. bioRxiv doi: https://doi.org/10.1101/661454

[16] Rice, K. (2018). Convolutional Neural Networks For Detection And Classification Of Maritime Vessels In Electro-Optical Satellite Imagery [Master's thesis, Naval Postgraduate School]. http://hdl.handle.net/10945/61255

[17] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Kholsa, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115, 211–252. https://doi.org/10.1007/s11263-015-0816-y

[18] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks For Large-Scale Image Recognition. International Conference on Learning Representations 2015.

[19] Skolnik, M. (1981). Introduction to Radar Systems (Second). McGraw-Hill, Inc.

[20] Stewart, M. (2019, February 26). Simple Introduction to Convolutional Neural Networks. Towards Data Science. https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

[21] Sitawarin,C. & Wagner, D. (2019). On the Robustness of Deep K-Nearest Neighbors. arXiv:1903.08333v1

[22] Tang, Y. (2013). Deep Learning using Linear Support Vector Machines. 2013 ICML Challenges in Representation Learning. https://arxiv.org/abs/1306.0239

[23] Timothy D. Ross, Jeff J. Bradley, Lannie J. Hudson, & Michael P. O'Connor. (1999). SAR ATR: so what's the problem? An MSTAR perspective. 3721. https://doi.org/10.1117/12.357681

[24] Wang, J., Zhang, T., Song, J., Sebe, N., & Shen, H. T. (2017). A Survey on Learning to Hash. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.(2017)13:9.

[25] Glorot,X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In Geoffrey Gordon, David Dunson, Miroslav Dudík (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (pp. 315–323). PMLR. http://proceedings.mlr.press/v15/glorot11a.html

[26] Yiu, T. (2019, June 12). Understanding Random Forest: How the Algorithm Works and Why it Is So Effective. Towards Data Science. https://towardsdatascience.com/understanding-random-forest-58381e0602d2f3c8

[27] Zhao, Q., & Principe, J. (2001). Support vector machines for SAR automatic target recognition. IEEE Transactions on Aerospace and Electronic Systems, 37(2), 643–654.