2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506246

FEW SHOT LEARNING FOR INFRA-RED OBJECT RECOGNITION USING ANALYTICALLY DESIGNED LOW LEVEL FILTERS FOR DATA REPRESENTATION

Maliha Arif, Abhijit Mahalanobis

Center for Research for Computer Vision (CRCV), University of Central Florida,Orlando,USA

ABSTRACT

It is well known that deep convolutional neural networks (CNNs) generalize well over large number of classes when ample training data is available. However, training with smaller datasets does not always achieve robust performance. In such cases, we show that using analytically derived filters in the lowest layer enables a network to achieve better performance than learning from scratch using a relatively small dataset. These *class-agnostic* filters represent the underlying manifold of the data space, and also generalize to new or unknown classes which may occur on the same manifold. This directly enables new classes to be learned with very few images by simply fine-tuning the final few layers of the network. We illustrate the advantages of our method using the publicly available set of infra-red images of vehicular ground targets. We compare a simple CNN trained using our method with transfer learning performed using the VGG-16 network, and show that when the number of training images is limited, the proposed approach not only achieves better results on the trained classes, but also outperforms a standard network for learning a new object class.

Index Terms— manifold, eigen representation, few shot learning, sparse learning, infra-red datasets

1. INTRODUCTION

While many large datasets exist for object recognition in color imagery, there is a dearth of training data for infra-red machine vision applications. Therefore, the main goal of our paper is to teach a classifier to recognize a new object in infra-red images using very few training examples. The idea behind few-shot learning (FSL)[1, 2], meta-learning [3] and transfer learning [4, 5, 6] is to enable AI systems to learn a new problem quickly using as few training images as possible, while leverage their prior knowledge of old tasks to learn new ones. Vinayls et.al [7] performed one-shot learning by introducing a matching network that maps a small labelled support set and an unlabelled example to its label. However, it assumes there are plenty of training images to train this matching network whereas this isn't true in our case. A meta-learner[8] is fine-tuned to reuse previously learned features to adapt to a new task. This avoids the need to start from a random initialization of parameters as that would most likely overfit and not converge for learning the new task. An optimal starting point for fine-tuning has been proposed by Model-Agnostic Meta-Learning (MAML) [3] algorithm to achieve fast learning on a new task with small number of gradient steps. Inspired by such previous observations, we use analytically derived filters in the first layer to facilitate quick convergence and higher accuracy when very few training images are available. Specifically, these filters are derived by clustering the training data of the existing classes, and then finding the dominant eigenvectors[9] that best represent the salient local features of the manifold[10]. Using these filters in the first layer, the rest of the network builds on the resulting low-level features to learn the classes represented by the training data. Unlike ran-

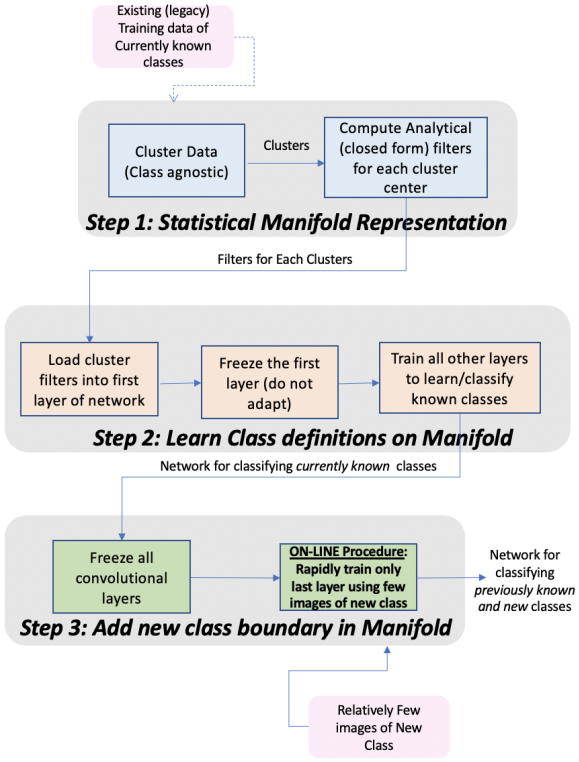

Fig. 1. Proposed step by step workflow.

We gratefully acknowledge the support of Lockheed Martin MFC

dom initialization (which does not impose any restrictions on the search space), analytical representation of the manifold provides strong priors for the learning algorithm, and thereby improves overall performance. To show the efficacy of our method, we apply our approach on a publicly available dataset of medium wave infra-red (MWIR) images of vehicular targets [11]. Using both simple CNNs as well as VGG-16[12, 13], we show that the proposed method not only leads to better performance, but also allows new classes to be accurately learned with relatively fewer images.

2. METHOD

Figure 1 summarizes the key steps of the proposed approach. The first step is the *class-agnostic* clustering of available training data of the known classes. In other words, the data is clustered without regard to the class labels, solely on the basis of local euclidean proximity of the training samples. Therefore, several different classes may occur in any given cluster. The k-means algorithm[14] may be used for forming these clusters. We then find filters that not only best represent each cluster, but also separate it from its neighbors. The number of clusters depends on the overall complexity of the dataset, and may be treated as an experimental design parameter.

2.1. Filter Derivation and the HybridNet

Step 1: Consider a large dataset of images which includes several different classes. Assume that these are grouped into C clusters using k-means (or any other suitable clustering method). It should be noted that a cluster may contain training images of several different classes. Let $\mathbf{x}_{ij}$ represent a windowed section of the *i-th* training image vector in cluster j which is the same size as filter $\mathbf{q}_j$ defined ahead. The following statistic is estimated over all possible positions of the window across the full image. For each cluster, we define the mean vector $\mathbf{m}_j = 1/N_j \sum_i \mathbf{x}_{ij}$ and the correlation matrix $\mathbf{R}_j = 1/N_j \sum_i \mathbf{x}_{ij}\mathbf{x}_{ij}^T$, where N_j represents the number of training samples in cluster j. We then compute the Euclidean distances[15] between the centers of all pairs of clusters j and k given by $D_{jk} = (\mathbf{m}_j - \mathbf{m}_k)^T(\mathbf{m}_j - \mathbf{m}_k)$. For any given cluster j, we sort the corresponding values of D_{jk} in ascending order, and select a subset L of the C neighboring clusters whose centers are closest to its own. The matrix $\mathbf{R}_L = 1/L \sum_{k \in L} \mathbf{R}_k$ is the average of the correlation matrices of these neighboring clusters. We are interested in finding a filter $\mathbf{q}_j$ which is highly correlated to the training vectors $\mathbf{x}_{ij}$ that belong to cluster *j*. At the same time, we wish to minimize the correlation between $\mathbf{q}_j$ and the samples which belong to the neighboring cluster now denoted by $\mathbf{y}_{ik}$, $k \in L$. This can be achieved by maximizing the ratio φ, given by:

$$\varphi = \frac{E\{|\mathbf{x}_{ij}^T \mathbf{q}_j|^2\}}{E\{|\mathbf{y}_{ik}^T \mathbf{q}_j|^2\}} = \frac{\mathbf{q}_j{}^T \mathbf{R}_j \mathbf{q}_j}{\mathbf{q}_j{}^T \mathbf{R}_L \mathbf{q}_j} \quad (1)$$

Fig. 2. Images of the targets in the DSAIC-ATR dataset. This montage represents targets at 1.5km test range and appear in order as described in the text.

where the operator $E\{.\}$ represents the expectation over all i. We see that the optimum choice for $\mathbf{q}_j$ is the dominant eigen-vector which satisfies $\mathbf{R}_L^{-1}\mathbf{R}_j\mathbf{q} = \lambda\mathbf{q}$, and where λ is the dominant eigen-value. It is also possible to use more than one eigen-vector (in addition to the dominant one) which correspond to other large eigen-values. The number of eigen-vectors to use for each cluster is a design parameter which can be experimentally selected. *Step 2:* The second step is to load the filters into the first layer of the network and hold them fixed while training the remaining layers. We use a relatively simple CNN for illustrative purposes, although the proposed method generalized to any type of network. The spatial dimension for all the filters is 5 x 5 (i.e. $\mathbf{q}_j$ is 25 x 1 dimensional vector). For the infra-red dataset, the first layer has as many filters as the number of clusters formed in step 1. The subsequent four layers have 32,64,32,32 filters respectively. These convolutional blocks are followed by two fully connected layers, one sized 64 and another of size 9. The network is then evaluated on the infra-red (IR) dataset using the training and testing protocols described in Section 3. We refer to the resulting base network as the *HybridNet* because of the hybrid analytical and iterative learning technique. *Step 3:* The final step , *'Add new class boundary in Manifold'* as shown in Figure 1 allows us to teach the network a new class which was excluded entirely from the first two steps (i.e. clustering and training the base network). With all the convolutional layers of the network frozen, the new class is added by simply fine-tuning the fully connected layers with just a few training samples. To maintain class balance, an equivalent number of training images of the previously trained classes are also used for this third and final step.

3. EXPERIMENTS

In this section, we discuss the performance of a baseline network trained conventionally from scratch, and compare it to the performance of the HybridNet on the infra-red dataset. We also discuss the results of adding a new object class with

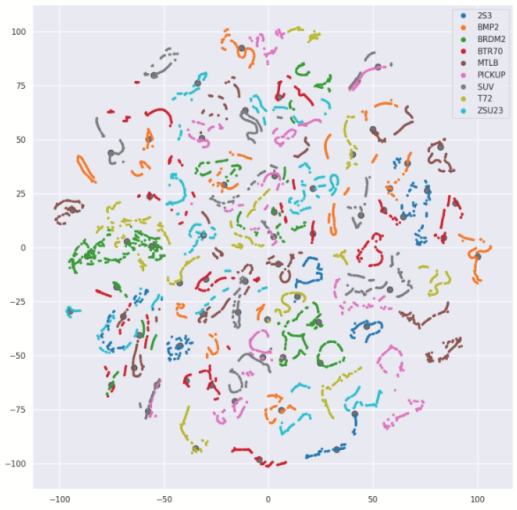

Fig. 3. t-SNE visualization for DSIAC training set. The Gray dots represent the centers of the 65 clusters formed by the k-means algorithm.

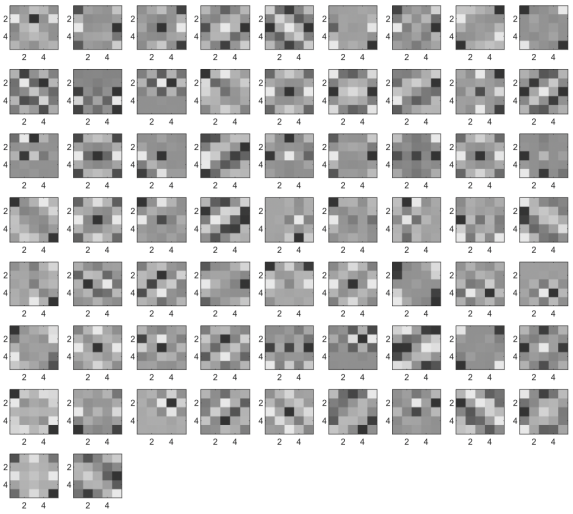

Fig. 4. Montage of the 65 filters (one per cluster) designed for the first layer.

very few images. We compare these results to those obtained via transfer learning[16] using VGG-16. Further, to show the advantages of our filter derivation methodology, we also use the filters of the 1st layer of VGG-16 in our base network (instead of the analytically derived filters) and compare the obtained results. ***DSIAC-ATR Infra-red dataset:*** A database of infra-red images of targets collected by US Army Night Vision Laboratories is available from DSIAC [17]. This dataset has medium wave infra-red (MWIR) images of 9 vehicle categories, depicted in Figure 2. These include 2S3 - Self-Propelled Howitzer, BMP2 -Armored Personnel Carrier, BRDM2 –Infantry Scout Vehicle, MTLB –Armored Reconnaissance Vehicle, BTR70 –Armored Personnel Carrier, SUV, PICKUP, T72 –Main Battle Tank and ZSU23-4 -Anti-Aircraft Weapon. For each target type, video was collected at ranges varying from 1km to 5km in increments of 500m in both day and night conditions. The original images are of size 480 x 640. Since range information is available, we resize them such that the targets appear to be at a range of 2km. We then use ground truth information to extract a 32 x 64 image patch containing just the target for training and testing purposes. We choose ranges of 1km,2km,3km,4km and 5 km for the train set, and ranges 1.5km,2.5km,3.5km and 4.5km for the test set. This is done to ensure training and testing images do not share the same background, and that the classifier must distinguish objects based on the object's shape and features only. In total, there are 15660 training chips (approx.1800 per class) and 12600 test chips which we then use in all our experiments. The T-SNE[18] visualization of the training data manifold is shown in Figure 3, where the colored dots represent the samples of the different classes. Using the k-means algorithm, we obtain 65 clusters whose centers are also indicated on the figure by the gray dots. As mentioned before, a given cluster may contain images from several classes. Although the number of clusters is chosen somewhat arbitrarily, the goal is to use sufficient clusters to capture the local structure of the manifold. Following the procedure in Section 2.1, we then design 5 x 5 filters, one for each of the 65 clusters as shown in Figure 4. Holding these weights fixed in the first layer of the network, we randomly initialize the rest of the network and train for 20 epochs using RMSprop [19]optimizer and a learning rate of 1e-3 for the initial 10 epochs followed by 1e-4 for the last 10 epochs. We use a batch size of 100. The first version of the network (Ver1) is trained on all 9 classes, and its performance is shown by the deep blue bars in Figure 5. For comparison, the light blue bars show the performance obtained when the first layer is initialized randomly, and trained from scratch with the rest of the network. It is clear that using the analytically designed filters improved the overall performance for all 9 classes. As shown in the second column of Table 1, the overall accuracy for this version (Ver1) for the HybridNet is 0.85, while training completely from scratch (including the first layer) achieves a lower accuracy of 0.75. Column 3 shows the performance on the BMP2 class to be greater for HybridNet than its conventionally trained counterpart. Next, our goal is to train another version of network (Ver2) to first recognize 8 classes using the first two steps in Figure 1, and then in the last step teach it the 9-th class using very few additional images. In this experiment, we remove the BMP2, from the initial training process, but then add it in the third step of Figure 1. To be clear, the BMP2 images are completely removed from the clustering and the low level filter design. After the model learns the 8 classes, we follow step 3 and hold all of its convolutional layers fixed, but use 10% of images for BMP2 (approx.180) to fine-tune the

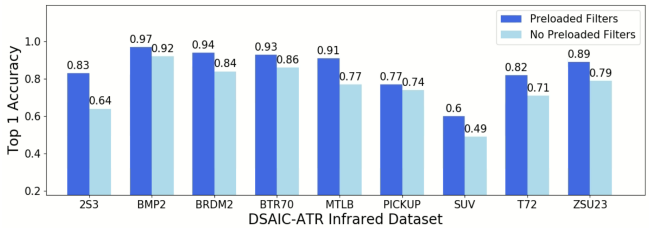

Fig. 5. Top-1 Accuracy for all classes in DSAIC ATR dataset for HybridNet (dark blue), and when all layers are randomly initialized (light blue) i.e. trained from scratch.

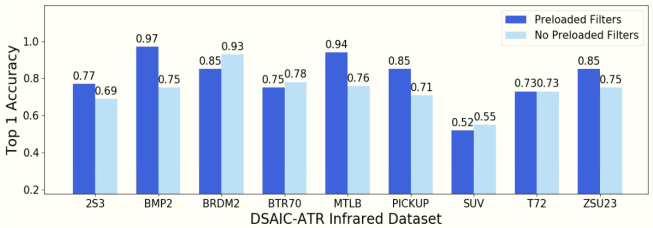

Fig. 6. Top-1 Accuracy for all classes in DSAIC ATR dataset when HybridNet (dark blue) and its conventionally trained counterpart (light blue) learn the BMP2 with 10% images.

fully connected layers. This enables the network to learn the new (9th) class very rapidly. To prevent class imbalance, 180 random images for other classes are also used. The network is again fine-tuned for 20 epochs. The results of this experiment are shown in Figure 6. For comparison, we also trained the 8-class network entirely from scratch with random initialization, and then added the BMP2 in the same manner as in step 3. The classification accuracy of the BMP2 for the HybridNet (dark blue) is seen to be higher than that obtained using random initialization and conventional training (light blue). As noted in Table 1 (under Ver2 overall column), the network trained from scratch achieved a lower overall recall of 0.74 compared the HybridNet which achieved 0.81. On the newly learned BMP2 class, the network trained from scratch achieves an F1-score and recall of 0.78 and 0.75, respectively, which are also lower than the HybridNet's F1-score and recall of 0.80 and 0.97,respectively. ***Comparison with VGG-16:*** VGG[13] and other deep classification models pre-trained on ImageNet[20] have been widely used for transfer learning. The question we ask is whether such networks (pre-trained on RGB data) can learn to classify IR image with relatively few images? To answer this question, we fine-tune the fully connected and last convolutional layer of VGG-16 on the DSIAC-ATR dataset to recognize all 9 classes (Ver1). We also create a second 8 class version (Ver2) similarly fine-tuned using training images of all classes excluding the BMP2. To teach this version the 9th class, it is later fine-tuned with just 10% of the BMP2 images. We use SGD optimizer[21] and a learning rate of 1e-3 with a batch size of 64. The results in Table 1

DSAIC-ATR	Ver1 Overall	Ver1 BMP2	Ver2 Overall	Ver2 BMP2(10%)
Scratch				
F1-score		0.91		0.78
Recall	0.75	0.92	0.74	0.75
HybridNET				
F1-score		**0.95**		**0.80**
Recall	**0.85**	**0.97**	**0.81**	**0.97**
$VGG-16$				
F1-score		0.80		0.68
Recall	0.75	0.76	0.76	0.69
$VGGLayer1$				
F1-score		0.82		0.60
Recall	0.77	0.88	0.76	0.85

Table 1. Comparison of the HybridNet to a conventionally trained network and VGG16. We also show the results obtained using 1st layer filters of VGG16 in our network.

show that the VGG-16 fine-tuned directly on 9 classes (Ver1) achieves an overall recall of 0.75. Specifically on the BMP2, it achieves F1 score and Recall of 0.8 and 0.76, respectively. In comparison, Ver2 achieved similar overall recall of 0.76. However, on the BMP2 (now learned with 10% images), it achieves lower F1 score and recall of 0.68 and 0.69, respectively, compared to Ver1 results for the same experiment. Finally, we ask the question whether the weights of the first layer of the VGG-16 network can be used in lieu of the analytically derived weights in the first layer of HybridNet? The results of this experiment are also shown in the fifth row of Table 1, which shows that the performance is much lower than that of HybridNet for both Ver1 and Ver2 cases. This demonstrates that the proposed method not only leads to better overall performance for the IR dataset, but also produces better results for learning a new class with few images.

4. CONCLUSION

We have introduced the HybridNet for infra-red object recognition and few shot learning using analytically designed filters in the first layer of a network which represent clusters in the data manifold. These filters not only produce features which generalize well across existing and new classes, but also provide strong priors for the learning process which enables the network to achieve better results than learning from scratch. This is particularly true for cases where the dataset is relatively small, and new classes must be learned using very few images. In particular, we demonstrated that the HybridNet i) achieves better performance than learning from scratch, ii) is better at learning a new class with 10% training images iii) works better than fine-tuning a standard network such as a pretrained VGG-16, and iv) is also better than using the VGG-16's layer-1 weights in lieu of the analytically derived filters.

5. REFERENCES

[1] Jake Snell, Kevin Swersky, and Richard S Zemel, "Prototypical networks for few-shot learning," *arXiv preprint arXiv:1703.05175*, 2017.

[2] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[4] Marcia Hon and Naimul Mefraz Khan, "Towards alzheimer's disease classification through transfer learning," in *2017 IEEE International conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2017, pp. 1166–1169.

[5] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun, "Accelerating very deep convolutional networks for classification and detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 1943–1955, 2015.

[6] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," *arXiv preprint arXiv:1411.1792*, 2014.

[7] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," *arXiv preprint arXiv:1606.04080*, 2016.

[8] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals, "Rapid learning or feature reuse? towards understanding the effectiveness of maml," *arXiv preprint arXiv:1909.09157*, 2019.

[9] Peter B Denton, Stephen J Parke, Terence Tao, and Xining Zhang, "Eigenvectors from eigenvalues: A survey of a basic identity in linear algebra," *arXiv preprint arXiv:1908.03795*, 2019.

[10] Philippos Mordohai and Gérard Medioni, "Dimensionality estimation, manifold learning and function approximation using tensor voting.," *Journal of Machine Learning Research*, vol. 11, no. 1, 2010.

[11] DSIAC, "ATR Algorithm Development Image Database," *Available: [Online]*, 2017.

[12] Hussam Qassim, Abhishek Verma, and David Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 169–175.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] Amir Ahmad and Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.

[15] Sachin Ravi and Hugo Larochelle, "Optimization as a model for few-shot learning," *arXiv preprint*, 2016.

[16] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[17] Maliha Arif and Abhijit Mahalanobis, "Multiple view generation and classification of mid-wave infrared images using deep learning," *arXiv preprint arXiv:2008.07714*, 2020.

[18] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[19] Yoshua Bengio and MONTREAL CA, "Rmsprop and equilibrated adaptive learning rates for nonconvex optimization," *corr abs/1502.04390*, 2015.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[21] Léon Bottou, "Stochastic gradient descent tricks," in *Neural networks: Tricks of the trade*, pp. 421–436. Springer, 2012.